Towards a Safety Mechanism for Cooperative Automated Driving

Ellen van Nunen[1], Dimitrios Tzempetzis[2], Gerald Koudijs[2], Henk Nijmeijer[2], Mark van den Brand[2]

Abstract— **Cooperative Automated driving has shown to be technically feasible, but safety aspects are still challenging. Wireless communication between vehicles allows to maintain reduced inter-vehicle distances, thereby improving traffic throughput and decreasing fuel consumption. As the driver can no longer be a backup at short inter-vehicle distances, the system needs to be fail-safe for both hazardous traffic situations as well as failures. In this paper, a scenario is defined which combines a hazardous traffic situation with a communication failure. First, the methodology for developing safety related functionality in automated driving is presented. This methodology combines aspects of the ISO26262 standard with the Harmony profile. Second, the safety mechanism to avoid a collision by braking is described. This ensures that a safe state can be reached for a set of use cases which are derived from the defined scenario. Finally, the proposed solution is tested in a simulation environment and is also implemented on test vehicles. The result of the simulations and experiments demonstrate the practical validity and show increased safety related functionality.**

I. INTRODUCTION

Automated driving is one of the promising solutions to make ground transportation safer, more comfortable and more environmentally friendly [1]. When wireless communication between vehicles is added to automated driving, short inter-vehicle distances become possible. Thereby traffic throughput increases [2], and for trucks, in particular, a significant decrease of fuel consumption is expected [3].

DARPA Grand Challenges and the Grand Cooperative Driving Challenge demonstrated the state-of-the-art technology of vehicle automation [4], [5]. In these prototypes the driver is often still considered as a backup.

Cooperative Automated Driving systems can be designed for automation level 3 or 4, according to SAE J3016 [6]. For automation level 3 the driver is assumed to be a backup, while for level 4 the driver is no longer relied upon as a backup. Legally, these systems were not allowed because of the 1968 Vienna Convention on Road traffic, which stated that a human driver has to drive the car him/her self. Recently, this convention has been adapted towards the statement that a driver must be present and able to take the steering wheel at any time. This is a huge step towards legal permission of cooperative automated driving.

When the driver has no longer a driving task, reaction times can go up to 10 s [7]. So, a reliable backup is needed (also for level 3 systems) in order to bring the vehicle to a safe state, both in (hazardous) traffic situations and in case of (hardware) failures. For this purpose, a safety mechanism for cooperative automated driving is required.

Advanced Driver Assistance Systems, [1], guarantee operational safety (the safety of the driver, the passenger and other traffic participants) by means of subsystems that take care of safety. Since the driver is still a backup here, false negatives are acceptable in these systems. Further, the timing of autonomous brake activation is based on human reaction times and is therefore conservative, [8]. These two aspects complicate the direct use of these subsystems in cooperative automated driving systems.

The development of these subsystems is among others based on the ISO26262 functional safety standard [9]. ISO26262 provides guidelines in order to prevent or mitigate a hazard that can be caused by hardware or software failures. The ISO26262 standard is designed for all the electrical/electronic automotive systems, not explicitly for automated driving systems. Although the standard assumes that the driver is a backup, it could still be used as a basis for development of automated driving systems.

A lot of research is presented in literature on the development of safety related functionality in automated driving systems. Fault-tolerant solutions, which aim to keep the system functional in case of hardware failures, are discussed in [10]. In [10] several reliability requirements are suggested. One of the conclusions is that the design of architecture is important in realizing safety. A number of papers focus on the architecture description for safety [11],[12], while others emphasize on a design of a system based on traffic rules, [13]. Moreover, fault-tolerance for wireless failures are considered in [14] and [15]. Particularly, [15] focuses on determining the safe state in case of wireless failures by calculating safe time gaps. However, the transition towards increasing the unsafe time gap to the safe time gap is not completely tested for all operating conditions. In case braking actions occur while the safe time gap is not reached yet, collisions might occur.

This paper aims to improve safety during the period between a wireless failure towards the moment at which the safe time gap is reached. The proposed approach incorporates fault-tolerance and takes into consideration the ISO26262 guidelines.

The outline of this paper is as follows. First, the problem is further defined in Section II. Then, the approach is described in Section III. Section IV and V present respectively the (software) architecture and detailed design of the cooperative automated driving system including safety related functionality. Finally the test results and conclusions are presented in Section VI and VII.

[1]Ellen van Nunen is with TNO, The Netherlands

[2]Dimitrios Tzempetzis, Gerald Koudijs, Henk Nijmeijer and Mark van den Brand are with TU/e, The Netherlands

II. PROBLEM STATEMENT

The main functionality of the cooperative automated driving system is short-distance vehicle following (longitudinal and lateral). Longitudinally the system is based on Cooperative Adaptive Cruise Control (CACC), [16], and lateral following functionality is added using a camera to track the lateral position of the lead vehicle.

Cooperative automated driving heavily depends on the availability of wireless communication. However, since wireless communication is inherently unreliable, a failure in wireless communication could result in a hazard, especially when the lead vehicle starts braking. This scenario is shown in Figure 1. The scenario can be detailed by means of several use cases, different moments of braking with respect to the time of failure, different durations of wireless failure, and different deceleration levels of the lead vehicle.

A subsystem, providing a safety mechanism, needs to be designed which, in case of a critical failure such as a communication failure, brings the vehicle to a predefined safe state. The vehicle shall remain in the safe state until the driver is able to take over the control.

One of the use cases could be a wireless communication failure and the lead vehicle continues with constant speed. This situation is not that hazardous, so a severe braking action should be avoided. In this paper, a situation is refered to as ”false-positive” when severe braking action is applied which is not hazardous. The safety mechanism should minimize false positives.

Further, system availability (the time for which the automated driving functionality is available) can be increased by applying fault tolerance. The vehicle following functionality should be maintained as much as possible (although at a larger time gap).

This paper aims to extend the cooperative automated driving system with a safety mechanism to cover fail-safety (safety only) and fault-tolerance in case of wireless communication failure, thereby following the guidelines provided by the ISO26262.

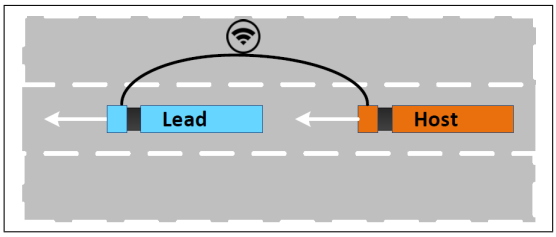

Fig. 1. Use Case: emergency braking in combination with wireless failures.

III. APPROACH

The approach for the development of the safety mechanism is based on system engineering techniques and includes the following steps, [17]:

- **System development process**

The ISO26262 standard follows the V-model. This model, shown in Figure 2, describes the development process which is followed in this paper.

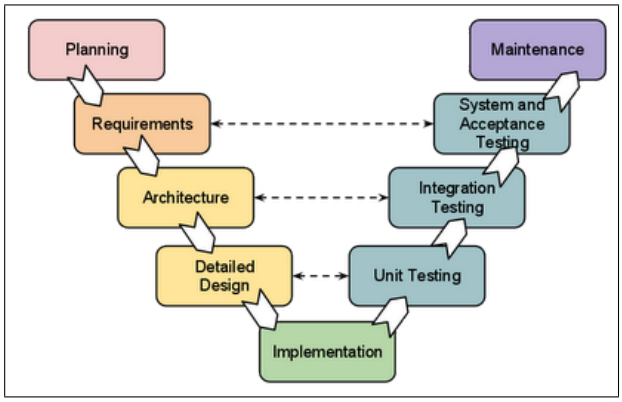

Fig. 2. V-model as system development process

- **Requirements**
  ISO26262 introduces two requirement types at the system level; the functional safety requirements (FSR) and the technical safety requirements (TSR). The FSRs are derived from the ”safety goals”. The safety goals are the outcome of the Hazard Analysis and Risk Assessment (HARA) and form one of the main work-products of the ISO26262 ”Concept phase” [9]. To transform the FSR and TSR into a modular design, system modeling languages, such as SysML or VHDL (VHSIC Hardware Description Language), can be used for this step.
  The selected methodology for the execution of the top-down phases of the V-model is based on the Harmony profile [18]. The Harmony profile supports model development per use case which enables parallel development of subsystems. This results in a modular and extensible design of the system. Furthermore, the development steps of Harmony are compatible with the diagrams of the system modeling language SysML.

The architecture and detailed design phases are further detailed in Sections IV and V respectively. The validation steps (systems and acceptance testing in Figure 2) are described in Section VI.

IV. ARCHITECTURE

The cooperative automated driving system consists of several hardware and software blocks, [16]. The design of the safety subsystem restricted to the update of the decision unit of the system: the Supervisor block. To allow duplication of the safety related functionality on a redundant hardware, we have chosen to concentrate the safety-related functionality in one block and separate this from the nominal functionality. Therefore, two decision units were introduced in the supervisor as shown in Figure 3: the safety decision unit (SDU) and the performance decision unit (PDU). The composition of the safety decision unit results in a block with centralized safety functionality, while the performance decision unit includes the nominal functionality. In accordance with the ISO26262

standard, the safety decision unit consists of the following components:

- The Safety Measures block monitors the system and traffic status.
- The Safety Mode Selection block decides upon the safe state.
- The Safety Algorithm block determines the action to reach the safe state.

The functions and the behaviour of these blocks are described in Section V.

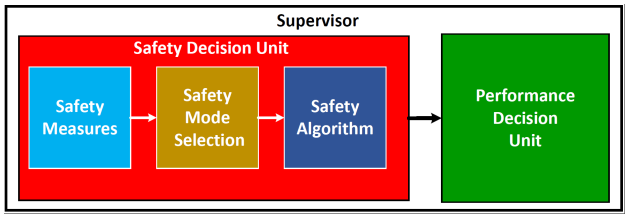

Fig. 3. The Supervisor block in the cooperative automated driving system.

V. Detailed Design

Safety Measures

The safety measures are needed as input for the decisions in the Safety Mode Selection block. A safety measure identifies a criticality index for a situation. The following assumptions are made for calculation of the safety measures:

- Maximum braking of the lead vehicle is assumed (a_{min}=-6 m/s^2).
- The lead and host vehicle are identical vehicles.

Under these assumptions, the following safety measures are defined, [19]:

- Brake Threat Number (BTN), [20] : the ratio of required acceleration (a_{req}) of the host vehicle upon minimum acceleration (a_{min}).

$$\text{BTN}(t) = \frac{a_{req}(t)}{a_{min}} \tag{1}$$

  The required acceleration is based on the calculation of braking paths of the lead vehicle and the host vehicle. The required acceleration calculation takes into account:
  - the braking system actuation time delay,
  - the braking system actuator dynamics, and
  - the distance at standstill, which is chosen to equal 0.5 m. This allows for margin to compensate for external disturbances, which are not modeled.
- Worse case impact speed v_{imp}: assuming maximum braking of the lead and host vehicle, the worse case impact speed is calculated. A low-impact collision which would not lead to injuries in the improbable worse case scenario might be acceptable, to improve comfort.
- Probability of lead vehicle braking P: this safety measure can adapt the system behaviour to the traffic conditions. When the lead vehicle has communicated its environmental perception, the last communicated values for the distance d and range rate $\dot{d}$ between the lead and its preceding vehicle can be used to determine a probability of braking. When a preceding vehicle is far away (and assuming that no other traffic participants will cut-in), the probability that the lead vehicle will brake is very small. It can be discussed how this probability should be derived, but here a weighted Poisson distribution with configurable weight w is proposed:

$$P(d,\dot{d}) = wP_{\dot{d}}(\dot{d}) + (1-w)P_d(d) \tag{2}$$

  with

$$P_{\dot{d}}(\dot{d}) = \lambda_{\dot{d}} e^{-\beta_{\dot{d}}(\dot{d}-\dot{d}_{min})}\Delta t \tag{3}$$

  and

$$P_d(d) = \lambda_d e^{-\beta_d(d-d_{min})}\Delta t \tag{4}$$

  Here, d_{min} and $\dot{d}_{min}$ are the minimum range and range rate respectively, below which a probability is chosen to be 1. Therefore, λ_d and $\lambda_{\dot{d}}$ are calculated such that $P_d = 1$ for $d < d_{min}$ and $P_{\dot{d}} = 1$ for $\dot{d} < \dot{d}_{min}$. Variable Δt is the time for which the probability is calculated and configurable parameters $\beta_{\dot{d}}$ and β_d are chosen to be 2 and 3.

Safety Mode Selection & Safety Algorithm

In the Safety Mode Selection different states are defined to activate the proper braking action in the Safety Algorithm. Nominal CACC refers to the state in which no hardware failures exist. The behaviour in case of a communication failure is determined by the Safety Algorithm block in Figure 3:

- The fault tolerant state represents a possible final safe state. This state adjusts the (longitudinal) CACC controller settings (time gap and standstill distance) based on a calculation of brake paths, as described in [15]. Steady initial conditions are assumed, which is not the case in the transition towards the safe state. By means of simulations it is found that the system can remain at this state up to communication failures with a duration up to 0.5 s without causing a collision.
- Intermediate state in which limited braking is applied based upon a maximum jerk and minimum acceleration according to the ISO15622 specifications for ACC [21]. This state is mainly required to avoid a maximum braking action when the lead vehicle does not brake (the "false-postive" use case). However, if the lead vehicle would brake, while in this state, it could still lead to a collision (with a certain maximum impact speed v_{imp}), so to avoid a collision in this situation, an additional state is introduced, the fail-safe state.
- Fail-safe state, which is based on a real-time calculation of the required acceleration to avoid a collision, discussed for the BTN calculation (1). The equation for the reference acceleration a_{ref} applied to the system yields:

$$a_{ref}(t) = a_{req}(t) + \lambda\ a_{min}, \quad \lambda > 0 \tag{5}$$

  The configurable parameter λ implies more braking is applied than the required acceleration, which ensures the BTN is reduced over an emergency braking event.

These states are in the Safety Mode Selection related via a state flow diagram, which is summarized in Figure 4. Next,

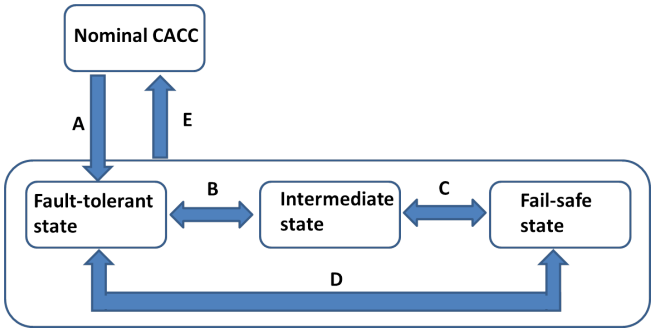

Fig. 4. The state flow diagram of the Safety Mode Selection.

the transitions (depicted by A, B, C, D and E in Figure 4) in the Safety Mode Selection can be further detailed:

- A: Detection of communication failure: in case no message is received for a period of 0.1 s, the fault-tolerance state is activated and the controller settings are adapted towards a larger desired time gap and standstill distance. This automatically results in a smooth braking action.
- B: If the duration of the communication failure is less than 0.5 s, the fault-tolerant state provides sufficient braking to avoid a collision. However, in case the duration of the communication failure is longer than 0.5 s AND BTN $>$ 0.85 AND the probability of the lead braking is less than a certain threshold, the intermediate state needs to be activated.
- C: If (BTN $>$ 0.95 AND range rate is negative) OR (v_{imp} $>$ 20 km/h) the fail-safe state needs to be activated.
- D: If the probability of the lead vehicle braking is above a certain threshold, the fail-safe state is immediately activated.
- E: CACC can only be activated in case the communication is recovered and when the initial conditions are safe:
  - The control errors, defined below, should be larger than zero:
    - the difference between the measured distance and the desired distance should be positive, and
    - its derivative with respect to time should be positive.
  - Furthermore, the relative acceleration should be larger than or equal to zero.

For a long duration of communication failure, the algorithm converges either to the fault-tolerant state or to the fail-safe state (which is represented as standstill at 0.5 m distance to its lead vehicle in case the lead vehicle has come to a full stop).

VI. TEST RESULTS

A. Test plan

A test plan is created to validate the developed design in case of wireless communication failure, both on simulation level as with experiments on a test track. The goal is to evaluate the ability of the host vehicle to avoid a collision with the lead vehicle in a two-vehicle platoon when the wireless communication fails.

For experiments on a test track, two test vehicles were available. However, these test vehicles were passenger cars. So, for the simulation study two vehicle types were modelled in order to simulate the cooperative automated driving system functionality: a truck and a passenger vehicle. The steps to create a test plan for the safety mechanism are described below:

- **Assumptions**
  A homogeneous platoon is assumed, so either both trucks or both passenger cars are used for the simulations. The platooning vehicles have the same braking capacity (up to a_{min}) and they drive initially with constant speed of 80 km/h at 0.3 s time gap.
- **Safety use case parameters**
  Furthermore, it is important to identify which parameters are critical when a communication failure occurs. These parameters, shown in Figure 5, are summarized below:
  1) The duration of the communication failure (t_{fail}): The time duration of the communication failure determines whether the vehicle will stay in the Fault-Tolerant state or it will enter the Intermediate state as described in V.
  2) The lead vehicle deceleration start time (t_{brake}): the moment a communication failure occurs, the behaviour of the lead vehicle is crucial. The worst case scenario is when the lead vehicle brakes exactly at the same time at which a communication failure occurs. Both this situation as well as the situation in which the brake action of the lead vehicle is 0.5 s later than the moment of communication failure is tested.
  3) The lead vehicle deceleration (a_{lead}): this parameter affects the host vehicle response as well. The values selected for the tests range from -6 m/s^2 to 0 m/s^2.
  4) The probability of the lead vehicle braking ($P(d, \dot{d})$): this parameter also influences the transition from Fault-tolerant state to Fail-safe state.
- **Test case derivation**
  The test cases are derived by combining the safety use case parameters. This leads to 80 test cases when all safety use case parameters are combined. Based on simulation, the critical combinations are determined. Besides, the test cases must also incorporate scenarios that are non-hazardous to evaluate false-positives. In total, 23 test cases are selected to validate the safety

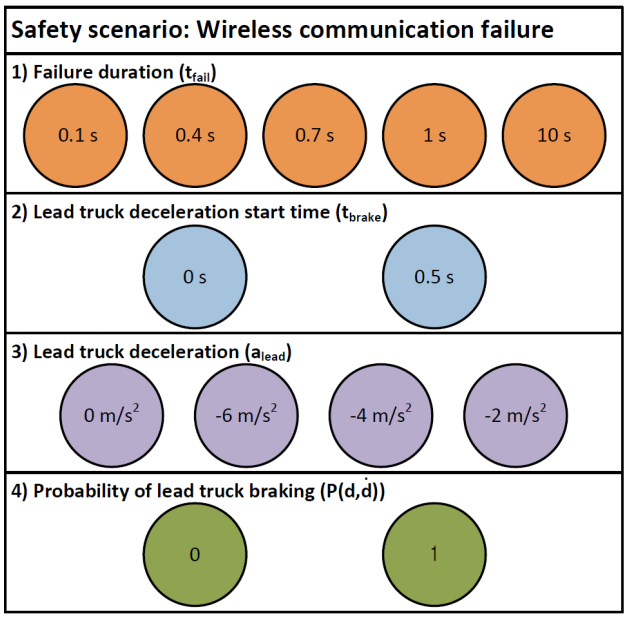

Fig. 5. Mapping of safety use case parameters.

mechanism. An example of a test case is a wireless failure with a duration of 10 s, and 0.5 s after the wireless communication failure occurred, the lead vehicle brakes with -4 m/s^2 till standstill.

B. Simulations

The 23 test cases have been simulated for both the passenger vehicle as well as the truck. To verify the safety improvement of the proposed safety mechanism, simulations with only CACC functionality were performed first. The simulations with the passenger vehicle showed that without the safety mechanism, 13 out of the 23 test cases resulted in a collision between the host and the lead vehicle. The simulations with the truck resulted in 14 possible collisions.

With the new design, which includes the safety mechanism, no collisions occur for all 23 test cases, for both vehicle types. Besides, in the false positive scenarios the system applies braking up to the ACC limit (-3.5 m/s^2), as defined in the "Intermediate" state. Therefore, the safety of the system improves for both vehicle types in case of wireless communication failure.

C. Experiments on a test track

After the simulations, the safety mechanism is tested in experiments on a test track. The main test goal is to verify the ability of the safety mechanism to avoid collisions between the following and lead vehicle in case of a wireless failure. The test vehicles, which are equipped with platooning functionality and with the safety mechanism, are two Toyota Prius III. All the 23 test cases are evaluated under the following operating conditions:

- Initially both vehicles drive with a constant velocity of 80 km/h.
- The spacing policy of the following vehicle is defined by the desired distance d_{des}:

$$d_{des} = hv + r \tag{6}$$

with h the time gap, equal to 0.3 s and r the standstill distance, which is chosen to equal 2.5 m.
The desired distance corresponds to approximately 10 m distance at 80 km/h. To ensure safety during the tests a virtual length of 15 m is added to the lead vehicle.
- The vehicle maximum deceleration is for both vehicles limited to -6 m/s^2.

The system showed consistent performance with the simulations. In all critical scenarios, which included braking at the moment of the failure, collision could be avoided. The final distance between the vehicles, which was aimed to equal at least 0.5 m, varied from 0.82 m to 4 m depending on the braking action of the lead vehicle.

However, the results of the false positive tests were less consistent. If the lead vehicle did not brake, in simulations only the intermediate state was activated and not the fail-safe state. This prevented a severe braking action in a non-hazardous situation. The minimum acceleration for these simulations is equal to -3.5 m/s^2. Within the experiments, due to sensor noise, the system switches from the intermediate state to the fail-safe state (transition C in Figure 4), resulting in a minimum acceleration of -5.3 m/s^2. This behavior is not desired and should be avoided in the future.

VII. CONCLUSIONS AND RECOMMENDATIONS

A methodology to introduce a safety mechanism for cooperative automated driving systems is proposed, based on system engineering methodology and the ISO26262 guideline. This methodology was used to extend the cooperative automated driving system with safety related functionality aiming at a system with automation level 3. The methodology was used to obtain a modular and extensible design at the supervisor level of a cooperative automated driving system based on a safety critical scenario of a wireless communication failure.

A safety mechanism to avoid collisions for this scenario is presented in this paper and is tested both on simulation level and with test vehicles. The experiments showed that collisions can be prevented in all 23 test cases. However, the false-positive situation in which the lead vehicle does not brake resulted in undesired severe braking. This is due to sensor noise, which resulted in an activation of the fail-safe state. This activation can be avoided by further tuning of the transitions to the fail-safe state, which is future work.

Also, heterogeneous platoons should be included. Other safety scenarios such as a cut-in or cut-through, and other (hardware and software) failures need further investigation. Both aspects are also part of future work.

REFERENCES

[1] A. Vahidi and A. Eskandarian, "Research advances in intelligent collision avoidance and adaptive cruise control," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 4, pp. 143–153, Sept 2003.

[2] S. E. Shladover, C. Nowakowski, D. Cody, and J. O'Connell, "Cooperative adaptive cruise control: Field testing of driver use and acceptance," in *Proceedings of the 16th World Congress & Exhibition on Intelligent Transport Systems and Services*, (Stockholm, Sweden), Sept. 21–25 2009.

[3] R. Ramakers, K. Henning, S. Gies, D. Abel, and H. Max, "Electronically coupled truck platoons on german highways," in *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, pp. 2409–2414, Oct 2009.

[4] J. Wille, F. Saust, and M. Maurer, "Stadtpilot: Driving autonomously on braunschweig's inner ring road," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pp. 506–511, June 2010.

[5] R. Kianfar, B. Augusto, A. Ebadighajari, U. Hakeem, J. Nilsson, A. Raza, R. Tabar, N. Irukulapati, C. Englund, P. Falcone, S. Papanastasiou, L. Svensson, and H. Wymeersch, "Design and experimental validation of a cooperative driving system in the grand cooperative driving challenge," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, pp. 994–1007, Sept 2012.

[6] SAE, "SAE J 3016:2014 Taxonomy And Definitions For Terms Related To On-Road Motor Vehicle Automated Driving Systems ," tech. rep., SAE, 2014.

[7] R. Kiefer and L. Angell, "A comparison of the effects of an analog versus digital speedometer on driver performance in a task environment similar to driving.," *Vision in vehicles*, no. 4, pp. 283–90, 1993.

[8] M. Brannstrom, E. Coelingh, and J. Sjoberg, "Model-based threat assessment for avoiding arbitrary vehicle collisions," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, pp. 658–669, Sept 2010.

[9] ISO, "ISO26262:2009 Road Vehicles - Functional Safety," tech. rep., ISO, 2009.

[10] A. Manzone, A. Pincetti, and D. De Costantini, "Fault tolerant automotive systems: an overview," in *On-Line Testing Workshop, 2001. Proceedings. Seventh International*, pp. 117–121, 2001.

[11] M. Horwick and K. Siedersberger, "Strategy and architecture of a safety concept for fully automatic and autonomous driving assistance systems," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pp. 955–960, June 2010.

[12] R. Debouk, B. Czerny, J. Dmbrosio, and J. Joyce, "Safety strategy for autonomous systems," in *29th International System Safety Conference 2011*, 2011.

[13] B. Vanholme, D. Gruyer, B. Lusetti, S. Glaser, and S. Mammar, "Highly automated driving on highways based on legal safety," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 14, pp. 333–347, March 2013.

[14] J. Ploeg, E. Semsar-Kazerooni, G. Lijster, N. van de Wouw, and H. Nijmeijer, "Graceful degradation of cacc performance subject to unreliable wireless communication," in *Intelligent Transportation Systems - (ITSC), 2013 16th International IEEE Conference on*, pp. 1210–1216, Oct 2013.

[15] E. van Nunen, J. Ploeg, A. Medina, and H. Nijmeijer, "Fault tolerancy in cooperative adaptive cruise control," in *Intelligent Transportation Systems - (ITSC), 2013 16th International IEEE Conference on*, pp. 1184–1189, Oct 2013.

[16] J. Ploeg, *Analysis and Design of Controllers for Cooperative and Automated Driving*. PhD thesis, Eindhoven University of Technology, Eindhoven, The Netherlands, Apr. 2014.

[17] D. Tzempetzis, "Towards a safety concept for cooperative automated driving," tech. rep., Eindhoven Unitversity of Technology, Stan Ackerman Institute, 2015.

[18] H.-P. Hoffmann, *Systems Engineering Best Practices with the Rational Solution for Systems and Software Engineering*. IBM Rational, release 3.1.2 ed., 2011.

[19] G. Koudijs, "Master thesis on control methods for fail-safety in cooperative automated driving," tech. rep., Department of Dynamics and Control, Eindhoven University of Technology, 2015.

[20] A. Eidehall, "Multi-target threat assessment for automotive applications," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pp. 433–438, Oct 2011.

[21] ISO, "ISO15622:2010 Intelligent transport systems - Adaptive Cruise Control Systems," tech. rep., ISO, 2010.