Interval-Valued Invehicular Latency Characterization for Risk/Fault Management Purposes

Nadhir Mansour Ben Lakhal [*,**] **Lounis Adouane** [***]
Othman Nasri [**] **Jaleleddine Ben Hadj Slama** [**]

[*] *Institut Pascal, UCA/SIGMA – UMR CNRS 6602, Clermont Auvergne University, 63178 Aubière cedex, France*
[**] *LATIS Lab, National Engineering School of Sousse (ENISo), University of Sousse, BP 264 Sousse Erriadh 1023, Tunisia*
[***] *Université de technologie de Compiègne, CNRS, Heudiasyc UMR 7253, 60203 Compiègne cedex, France*

Abstract: Guaranteeing the reliability of Intelligent Transportation Systems (ITSs) is an overlapping issue. Not only the in-road risks must be mastered, but also the internal threats that emerge from the ITS material aspects should be addressed carefully. Most of the safety verification solutions ignore the limitations issued from the ITS onboard embedded structure, such as the invehicular communication delays. In this work, a novel Response Time Analysis (RTA) method is introduced to bridge the gap between the risk/failure management issues and the material aspects (the invehicular communication delays in particular) for ITSs. Aside of its simplicity, the suggested RTA reveals all potential latency scenarios by returning an interval estimation of delays. The interval results enclose certainly the exact delay that may occur in run-time. As a proof of concept of the introduced RTA algorithm, the Hardware-in-the-Loop (HIL) validation technique is adopted. The HIL platform is realized thanks to a model of a modern industrial automotive system. The validation work proves the consistency and efficiency of the proposed RTA solution to ensure ITS safety assurance.

Keywords: Intelligent transportation systems, response time analysis, minimum/maximum invehicular latency, risk/fault management, hardware-in-the-loop, adaptive cruise control.

1. INTRODUCTION

Over the past decades, several solutions have been introduced by the Intelligent Transportation System (ITS) community to ensure the autonomous navigation (see Iberraken and Adouane (2022) and Ben Lakhal et al. (2022)). These large varieties of technologies have been integrated into modern ITSs to increase their autonomy and safety, e.g., artificial intelligence, data-driven/model-based solutions, as shown by Kolekar et al. (2021) and Nie et al. (2021). Due to the crucial context of ITSs, great attention has been paid for the autonomous driving safety/reliability assurance through various formal verification and risk management policies, see Liu et al. (2021). These latter are high-level software solutions that provide decisional safety guarantees for ITSs. Nonetheless, once these strategies are well defined, it is important to handle the material issues (i.e., the communication delays) related to the on-board equipment, where the high-level navigation algorithms are deployed, see Fan et al. (2021). Although the high-level approaches are sufficiently reliable, the ITS safety is still tightly linked to the proper operation of its embedded system. This issue is emphasized by the deep complexity of modern ITSs, where several strict operational timing requirements must not be violated, as stated by Zhu et al. (2021).

Nowadays, abundant number of components with completely distinct timing features are incorporated into modern ITSs. An arbitrary delay, loss or disorder in the communication packets can happen and may lead to a hazardous irregularity/discontinuity in the ITS functioning, see Ben Lakhal et al. (2020a). To overcome the reliability/safety issues emerging from the increase in the number of vehicular components, the ITS decision-making layer must be conscious of any potential communication delay. Yet, the ITS safety verification policies target only the in-road threats. However, the intelligent driving is not sufficiently reliable without warranting the ITS appropriate reaction to the environmental events. The in-vehicle delays are among factors that can unpredictably emphasize the risk level of a driving situation. In this sense, a comprehensive knowledge of all possible evolution of the invehicular latency, to guarantee that relevant decisions are made in good timing even under data freshness problems, is required (see Ben Lakhal et al. (June 2019)).

Not only the ITS risk management layer is sensitive to the onboard data transmission delays. Indeed, ITSs are enhanced by fault-tolerant control strategies to succeed the management of critical failures. To handle faults impacting sensors and perception tools, faulty observations are re-estimated analytically to carry on the navigation with reduced sensing capacities, see Zhang et al. (2021). In this case, the data may be recovered incorrectly if the reconstruction of the faulty measurement is proceeded with delayed data (cf. Section 2). Thus, adaptive and delay-aware decision making for ITSs will permit to face the aforementioned invehicular communication-related imperfections. Correspondingly, efficient methodologies to characterize the delays occurring in the ITS onboard communication for risk/failure management purposes are addressed in this work.

The existing Response Time Analysis (RTA) solutions in the literature rely generally on an uncertain probabilistic approximation of the invehicular delays or may estimate only the worst case values of these latencies. In this paper, the core of the RTA proposed in our previous work Nasri et al. (September 2019) (where the RTA was adopted only for design and schedulability purposes) is improved to consider the variation in the invehicular communication latencies. Contrarily to our previous RTA model and to the existing solutions that are applied as an offline analysis procedure, the suggested RTA is not limited to the computation of the worst-case of transmission latencies to validate the safety requirements. By adding the ability of assessing lower bounds of delays, an interval-value of latencies that encloses certainly the exact real delays is obtained. The interval findings can be exploited in runtime to report all possible states of data freshness and validity. In such a manner, the interval-valued RTA has great capacity to enhance the critical and time-sensitive components integrated in ITSs, such as the risk/failure management layers, where knowing the upper bound of delays is not sufficient (cf. Section 2). A proof of concept of the suggested RTA algorithm is presented by conducting experiments on an industrial automotive system within a diagnosis context. The evaluation of the diagnosis messages transmission time is indeed crucial, since the violation of the on-board diagnosis hard deadlines can be destructive.

The rest of the paper is organized as follows: Section 2 investigates limitations of existing RTA methods. It presents also the real motivation behind the characterization of all possible scenarios of the invehicular communication delays. Section 3 details the different steps from the suggested RTA model in this paper. Section 4 depicts the experimental validation work realized to prove the consistency of the introduced RTA model. Finally, Section 5 concludes this paper and discusses some future works.

2. MOTIVATION AND RELATED WORK

It is noteworthy that the invehicular communications may be extremely troublesome for ITSs. The freshness of the run-time data is mandatory to ensure the correctness and exactness of any navigation task. Indeed, the verification of the data freshness aims to inspect the data validity in the time space. All along the ITS functional lifetime, using expired samples (not corresponding to the current real-time sampling step) must be strictly avoided. In this context, recent studies about recognizing the data validity interval time acquire special relevancy, see Fu et al. (2019).

Several critical decisions made by the ITS depend on the analysis results of the data collected from the environment via different sensors. The proceeded data analysis counts roughly on the appropriate interpretation of the temporal dependency between the ITS variables and parameters, see Ben Lakhal et al. (July 2019a) and Ben Lakhal et al. (2020b). In many applications, such as data fusion, the invehicular latencies can corrupt the analysis of data that come especially from different communication pipelines. In order to evaluate the data freshness and then extract the correct temporal dependencies between data flows, knowing the worst case of latencies is not sufficient. Characterizing the minimum bound of latency that may happen is also needed to meet this goal.

Fault occurrence is another issue that emphasizes the undesired impacts of the invehicular delays on ITSs. Once a fault is detected by the diagnosis functions, safety countermeasures should be taken by the ITS through several predefined functional degradation modes. These modes play as backup strategies when proceeding the ITS nominal operation is impossible. Such a fault tolerant control aims to avoid aborting suddenly the navigation task. In general, the proposed fault tolerant control methods in the literature exploit the analytical redundancy characterizing the concerned system to re-estimate the faulty variables, see Zhang et al. (2020). A sort of temporal reconstruction of data is proceeded to master the fault-issued risk, as shown in Lee and Lee (2020). The accomplishment of the data reconstruction can be erroneous or impossible due to the communication latencies. Correspondingly, blocks providing the recovery data must consider all cases of the delays that may happen.

As explained, delay manifestation into the ITS embedded layout is an alarming issue. Thus, it is primordial to develop risk management and diagnosis approaches with great awareness of such latencies in regard to their role in warranting operational safety. Such delays may slow down reactions made by the ITS high-level safety verification techniques. In our previous work Ben Lakhal et al. (July 2019b), a communication latency-aware risk management for ITS was presented. However, there was no clear strategy permitting to quantify and characterize such invehicular delays. In this view, the present paper is dedicated to deal explicitly with the invehicular delays in relation with the risk/failure management context.

To characterize data propagation delays through the ITS embedded architecture, a large range of RTA models and professional software tools are used in the automotive market. CANoe, CANalyzer, SymTA/S and Rubus-ICE are among the software solutions that may assure latency analysis and invehicular network simulation, see Sun et al. (2019). However, the existing approaches and the commercial software do not provide facilities to evaluate all possible states of data freshness/validity. They assume that only the worst-case transmission latencies are relevant for the safety considerations. This may not always hold true, as explained in this section. As an alternative, interest is given in the sequel to a novel RTA algorithm that characterizes all possible latency states for the invehicular communication.

Generally, RTA are analytical models permitting the approximation of the end-to-end data transmission time through embedded systems. Indeed, a multitude of statistical RTA-based approaches were proposed to calculate a message transmission delays. These methods are efficient in performing RTA while filling gaps caused by incomplete information about the configurations and composition of the considered ITS, see Zeng et al. (2010). In a different way, several RTA models used probabilities to explore the system response times, see Gong et al. (2018). From this perspective, the probabilistic reasoning allowed standard RTA to predict events such erroneous data exchange, see Shah et al. (2016). Otherwise, Shuai et al. (2014) have recourse to Taylor series expansion to cope with RTA modeling imperfections.

Hereafter, the RTA is exploited for the first time for online safety verification and in-road risk management purposes. Conventionally, the RTA models provide reliability guarantees during an early design phase of a given embedded system to prohibit any unacceptable violation of on-board communication deadlines, see Nasri et al. (September 2019). Unlike the research line that relied on the uncertain statistical or stochastic RTA models, focus in this work is put on approaches providing certain thresholds of the minimum and maximum possible delays that may occur through the ITS. According to this point of view, RTA may serve as an informative support to avoid slow reactions of the risk management and safety assurance against any potential hazard in run-time.

3. RTA MODEL FOR INTERVAL TIME CHARACTERIZATION OF CAN RESPONSES

The RTA introduced in this paper is dedicated for ITSs where the CAN is the communication support. The ultimate concern of the current work is to estimate the invehicular communication minimum and maximum delays for risk management purposes. The CAN is only a communication protocol, which is selected to present an example of particular RTA case of study. Nonetheless, the proposed approach may be applied on any invehicular communication protocol.

3.1 Preliminaries

The first step from the proposed RTA is to distinguish all nodes, tasks and data flows of the ITS. A data flow refers to elements involved in the data transmission starting from the transmitter task, up to the recipient task. A node refers to an Electronic Control Unit (ECU), which is connected to the communication bus. Multiple tasks may be implemented on each node. Every task executes predefined functionalities. The manifestation of specific events may trigger the transmission of particular series of messages to a destination task. In this paper, all factors that may slow down a message transmission are examined. Contrarily to other methods, even delays that impact the message instance by the transmitter task are incorporated into the RTA to obtain precise results.

Consider an ITS, which is constructed from a network of n nodes. Each single node, denoted $N_{h=1..n}$, executes a finite number of tasks. Similarly, every task associated to N_h is denoted by $\Gamma_{h,i}$. Let us note by $S_{h,j}$ a series of consecutive messages (i.e., a message stream), which are initiated by a particular transmitter component/task. Since one node can include several tasks, the stream message should be assigned to a specific task instead of a node. In such a manner, a given flow φ_c consists of the complete transmission pathway, which conjoins data streams and tasks implicated into the end-to-end transmission. The key elements that should be verified via a deterministic RTA are:

- **Message worst-case broadcast time**: It is the maximum possible duration to transmit a CAN message. The calculation of such a period is accomplished with respect to the communication payload and to the duration to transmit an individual bit through the CAN-bus.
- **Maximum release jitter**: The jitter delays consists of the time extended between the message generation instant (due to an event) and the instant of its queuing.
- **Queuing delays**: Regarding to its criticality, a specific priority value is attributed to every possible message. Hence, a message can be blocked temporarily until delivering other higher priority messages. Otherwise, queuing delays may happen due an already initiated transfer of a lower priority message. As a consequence, both higher/lower priority messages that may make the CAN-bus temporarily unavailable should be considered to estimate latencies.

3.2 Proposed RTA model

To fulfill the RTA modeling, let us assume the following:

- Every task $\Gamma_{h,i}$ has a well-defined maximum execution time, denoted $C^{task}_{h,i}$.
- Every message stream $S_{h,j}$ is featured by a maximum duration to deliver the message. It is denoted in the sequel by $C^{message}_{h,j}$. The $C^{message}_{h,j}$ value does not consider any interference induced from other CAN messages.

$C^{task}_{h,i}$ and $C^{message}_{h,j}$ are important to define the final CAN response time. Let assume that $m_{h,j}$ is a CAN message initiated by the node N_h. Each CAN message possesses a unique identifier (ID). There are actually two distinct types of CAN messages relatively to the number of the ID bits, which may be 11 or 29 bits. The RTA introduced here focuses on the total number of bits included in the CAN frame. Thus, the worst-case broadcast time for a 11 bits ID frame is given by equation (1):

$$C^{message}_{h,j} = (55 + 10 \times lm_{h,j}) \times \tau_{bit} \qquad (1)$$

Likewise, counting the worst-case broadcast time for a 29 bits ID frame is feasible through equation (2):

$$C^{message}_{h,j} = (80 + 10 \times lm_{h,j}) \times \tau_{bit} \qquad (2)$$

Where $lm_{h,j}$ is the number of data bytes in the frame and τ_{bit} is the period to deliver a unique bit from this latter. τ_{bit} is set according to the CAN network baudrate and speed. The use of equations (1) and (2) is common in the literature. The additional bits implied by the stuffing mechanism should also be considered, see Lange et al. (2016). Indeed, the CAN frame includes a “Data Bytes” field that includes the message content. It incorporates also other fields ensuring the correct transmission of data. Without counting the content of the “Data Bytes”, the maximum number of bits in the rest of fields is respectively 55 and 80 for the 11 and 29 bits ID frames as shown in equations (1) and (2).

Thereafter, a Worst Case Response Time (WCRT) is calculated for every single message stream to obtain the end-to-end response-time for the whole flow. Let admit that $R_{h,j}$ is the WCRT of a given $m_{h,j}$ belonging to $S_{h,j}$. As already stated, not only the CAN frame transmission time should be taken into account, but also the maximum release jitter and the queuing block time. Correspondingly, equation (3) allows to estimate $R_{h,j}$:

$$R_{h,j} = J_{h,j} + C_{h,j}^{message} + W_{h,j} \tag{3}$$

where:

- $J_{h,j}$ presents the maximum queuing jitter, see Davis et al. (2007). $J_{h,j}$ is the main source of variability in the CAN transmission delays. It is tightly linked to the processing capacities of the node initiating the message transmission. In the sequel, every node from the CAN network is supposed to have a known value of maximum release jitter.
- $W_{h,j}$ is the message queuing delay. Davis et al. (2007) and Lange et al. (2016) presented the required algorithms to estimate $W_{h,j}$ by tackling all potential scenarios of CAN blocking time due to messages' priority issues.

The priority concept concerns also tasks. In this context, the WCRT assigned to a task is evaluated as:

$$R_{h,i} = I_{h,i} + C_{h,i}^{task} \tag{4}$$

Note that $I_{h,i}$ is the interference time caused by $hp(h,i)$ the set of tasks with a priority higher than $\Gamma_{h,i}$. Recursive algorithms to calculate $I_{h,i}$ are provided by Lange et al. (2016). Finally, the complete WCRT of φ_c consists of the sum of the response times related to the streams and tasks of this data flow.

A considerable literature is available on the approximation of the worst cases of CAN response times. Nonetheless, the minimum possible data propagation time through CAN was rarely discussed. With a slight modification in the proposed model to compute the WCRT of φ_c, the most optimistic evaluation of response times may be obtained as follows:

- Contrarily to equations (1) and (2), used to predict the maximum transmission time for a CAN frame, the stuffing bits should be ignored. Therefore, the shortest transmission time for CAN frame is given by equations (5) and (6), respectively for 11 and 29 bits ID frames:

$$C_{h,j}^{message} = (47 + 8 \times lm_{h,j}) \times \tau_{bit} \tag{5}$$

$$C_{h,j}^{message} = (67 + 8 \times lm_{h,j}) \times \tau_{bit} \tag{6}$$

- Both delays $W_{h,j}$ and $I_{h,i}$, defined in equations (3) and (4), should be neglected by ignoring the CAN blocking time due to higher or lower priority tasks/messages.

Finally, an interval-valued estimation of delays is derived from the introduced RTA model. The real CAN transmission time is definitely inside the interval bounds. In contrast to the existing methods that only tackle a probabilistic or a worst case of response times, the proposed method assesses all possible states of data freshness and validity. This is extremely valuable to make any risk/failure management scheme aware of all scenarios of the invehicular communication delays.

4. PROOF OF CONCEPT: APPLICATION ON SDK SYSTEM

To validate the suggested RTA approach, a high fidelity model of an industrial anti-crash system is employed. It simulates modern in-vehicular components and measurement devices. It is indeed developed and validated according to the industrial specifications (the model was elaborated by Renault Trucks/Volvo SAS and SERMA INGENIERIE under the DIAFORE project, see Nasri et al. (September 2019)).

The provided model has a great portability with the Hardware-In-the-Loop (HIL) experimental plants. Hence, a proof of concept for the proposed RTA scheme is presented thanks to the HIL real invehicular communication middleware. The functionalities supplied by this automotive system, named Smart Distance Keeping (SDK), are detailed in the sequel.

4.1 SDK description

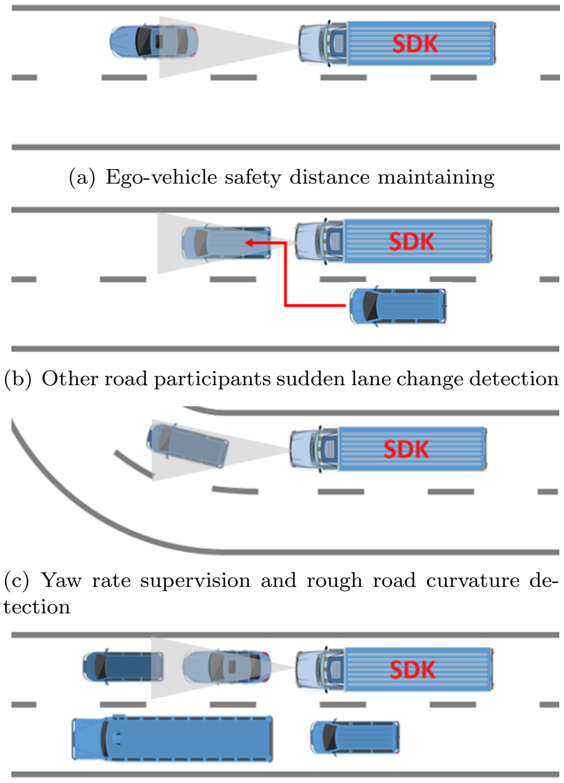
The SDK is an Adaptive Cruise Control (ACC) implemented on trucks. The long travel distances emphasize truck drivers distractions and then accidents. Thus, the SDK assists drivers in coping with hazardous situations in motorways. Apart from maintaining a safe distance from in-front vehicles, the SDK warns the driver in case of a sudden lane change performed by other vehicles to the SDK-equipped vehicle lane. Risks of unexpected hard breaks of other vehicles are also mastered via the SDK. Especially for night time travels, the SDK detects rough curvatures via monitoring the road yaw rate. The SDK deals also with traffic jams by maintaining an optimal distance from in-front objects. Figure 1 recapitulates all the SDK facilities.

Likewise, the SDK composition and its tasks are as follows:

- The "SDK controller" is in charge of the decision making process. A smooth control of the truck while mastering hazards is ensured by task $\Gamma_{1,1}$. Based on data provided by the rest of components, this task generates the convenient controls for the truck.
- A radar is mounted on the truck. In particular, the vehicle relative velocity and the spacing distance between the vehicle and other road participants are captured via task $\Gamma_{2,1}$. Due to interferences, the radar is prone to faults. Thus, a diagnosis task $\Gamma_{2,2}$ is also implemented into the radar.
- The SDK is devoted to assist drivers of six-wheel trucks. Hence, six sensors ensure the angular velocity measurement of wheels via the "wheel's ECU". The task $\Gamma_{3,1}$ checks continuously the difference between the angular velocities of wheels to estimates the truck longitudinal speed. To monitor the wheels' sensors, a model-based diagnosis task $\Gamma_{3,2}$, is executed, see Ben Lakhel et al. (2016) for details.

Table 1. Description of SDK data flows

Data flow	Description
$\varphi_1 = \{\Gamma_{5,2} + S_{5,1} + \Gamma_{1,1}\}$	Transfer recovery information and enable/disable the drive assistance
$\varphi_2 = \{\Gamma_{2,2} + S_{2,1} + \Gamma_{5,1}\}$	Transfer the diagnosis results of the radar device towards the supervisor
$\varphi_3 = \{\Gamma_{3,2} + S_{3,1} + \Gamma_{5,1}\}$	Inform the supervisor by the diagnosis outcomes given by the wheels' ECU
$\varphi_4 = \{\Gamma_{2,1} + S_{2,2} + \Gamma_{5,2}\}$	Inform the supervisor node by the truck velocity given by the radar
$\varphi_5 = \{\Gamma_{2,1} + S_{2,3} + \Gamma_{1,1}\}$	Provide the SDK controller with the truck velocity given by the radar
$\varphi_6 = \{\Gamma_{3,1} + S_{3,2} + \Gamma_{5,2}\}$	Inform the supervisor by the truck velocity given by the wheels' ECU
$\varphi_7 = \{\Gamma_{4,1} + S_{4,1} + \Gamma_{5,2}\}$	Provide the supervisor with the truck velocity given by the transmission block



(a) Ego-vehicle safety distance maintaining

(b) Other road participants sudden lane change detection

(c) Yaw rate supervision and rough road curvature detection

(d) Ego-vehicle's velocity control in traffic jam

Fig. 1. SDK capacities in truck drivers assistance

- The "Transmission ECU" determines, via task $\Gamma_{4,1}$, the truck longitudinal speed based on the crankshaft angular speed. For safety aims, the obtained value is compared with the one issued from the "wheel's ECU".
- A supervisor node receives diagnosis reports ensured by the remaining components. In this respect, task $\Gamma_{5,1}$ enables/disables the SDK based on a fault-tolerant control algorithm. Under faults, task $\Gamma_{5,2}$ substitutes if possible the faulty measurements by recovery data.

Table 1 and Figure 2 provide details about the SDK nodes, tasks and flows. The description of the SDK priority assignment of its messages/tasks can be found in Nasri et al. (September 2019). In the sequel, the response time intervals of flows φ_1, φ_2 and φ_3 are derived experimentally and then via the proposed RTA method. Focus is given for these flows due to the importance of the temporal dependency between the SDK variables to generate the diagnosis and recovery data. The obtained intervals of latencies will be exploited in a future work to present a recovery data estimation robust against delays.

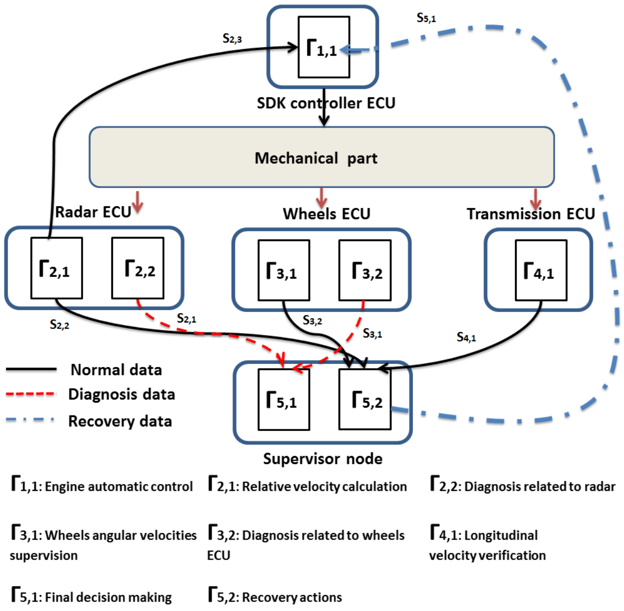

S2,3
S5,1
Γ1,1
SDK controller ECU
Mechanical part
Radar ECU
Wheels ECU
Transmission ECU
Γ2,1
Γ2,2
Γ3,1
Γ3,2
Γ4,1
S2,2
S2,1
S3,2
S3,1
S4,1
Normal data
Diagnosis data
Recovery data
Γ5,1
Γ5,2
Supervisor node
Γ1,1: Engine automatic control
Γ2,1: Relative velocity calculation
Γ2,2: Diagnosis related to radar
Γ3,1: Wheels angular velocities supervision
Γ3,2: Diagnosis related to wheels ECU
Γ4,1: Longitudinal velocity verification
Γ5,1: Final decision making
Γ5,2: Recovery actions

Fig. 2. SDK system tasks/flows

4.2 Experimental conditions and emulation environment

At this stage, experiments are tackled via a realistic HIL plan. The code of the supervisor (cf. Subsection 4.1) is implemented on a real electronic board. Hence, the CAN enables the communication between the SDK model and the supervisor. The truck dynamics are virtually simulated by the employed model. More precisely, a real electronic board is connected to the computer that runs the SDK simulated model. Hence, this first electronic node is responsible for the transmission/reception of messages to/from the supervisor electronic board. This latter is linked to an HMI executed on a second computer to keep the driver aware of emergencies.

The HIL is composed of two ARM Cortex-M4 electronic boards. The data exchange between both nodes via CAN is enabled through the $SN65HVD230$ transceivers. For all the realized tests, the CAN bit rate was fixed at $500\,Kbit/s$. Besides, all the messages are of 11 bits ID. Otherwise, the connection between each computer with its corresponding electronic board is established via an USART communication instead of using an expensive CAN emulator. Figure 3 shows the HIL platform.

Tests are tackled by triggering message streams after injecting faults in the SDK components. Without doubt, the SDK-truck is prone to fatal risks in case of important delays of diagnosis messages. In contrast to standard streams, the diagnosis messages are event-

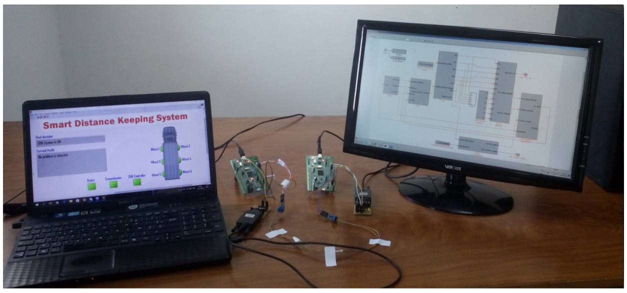


Fig. 3. HIL platform for realistic experimentation

triggered elements. Increasing the number of injected faults raises systematically the transmitted messages. The more overloaded the CAN, the more latency may occur, which permits to conduct tests under critical situations. Based on the proposed RTA (see Section 3), the minimum/maximum response times of φ_1, φ_2 and φ_3 (cf. Table 1) are presented in Tables 2, 3 and 4.

Table 2. Lower/upper response times of φ_1

Elements in flow φ_1	Lower response time (ms)	Upper response time (ms)
$\Gamma_{5,2}$	6.4	7
$S_{5,1}$	0.24	0.36
$\Gamma_{1,1}$	10.2	11
Total (ms)	16.84	18.36

Table 3. Lower/upper response times of φ_2

Elements in flow φ_2	Lower response time (ms)	Upper response time (ms)
$\Gamma_{2,2}$	6.6	7
$S_{2,1}$	0.21	0.52
$\Gamma_{5,1}$	9.3	10
Total (ms)	16.11	17.52

Table 4. Lower/upper response times of φ_3

Elements in flow φ_3	Lower response time (ms)	Upper response time (ms)
$\Gamma_{3,2}$	4.7	5
$S_{3,1}$	0.21	0.68
$\Gamma_{5,1}$	9.3	10
Total (ms)	14.21	15.68

To validate the theoretical results, the HIL experiments are carried out within distinct scenarios of fault injection. For each scenario, a particular number of diagnosis messages is transmitted. In such a manner, the number of injected faults in the predefined following scenarios is gradually raised and so is the CAN data traffic. As a result, the elaborated tests involve diverse cases of interferences between message streams and tasks. For each fault injection scenario, experiments have been repeated several times to obtain certain measurements of response times. Table 5 illustrates the timing performances measured from the HIL, presented by the Mean Response Time (MRT) of the considered flows.

Table 5. Experimental results

Scenarios	MRT of φ_1 (ms)	MRT of φ_2 (ms)	MRT of φ_3 (ms)
Scenario 1	18.272	17.141	15.176
Scenario 2	18.281	17.356	15.258
Scenario 3	18.309	17.427	15.409
Scenario 4	18.407	17.549	15.639
Scenario 5	18.456	17.670	15.760

Obviously, there is a great convergence between the MRTs recorded by the HIL and the theoretical RTA. In few occasions, the HIL results exceed slightly the RTA findings. This is noticed only for scenarios 4 and 5, where the busload is huge. This may be explained by the use of the USART communication as an interface between computers and the electronic boards. Although the USART delays are very small (which is not the main concern of this work), they are frequent. Therefore, the occasional violation of the predicted response times does not contradict the RTA efficiency in providing valuable informative support about bounds of the invehicular communication latencies.

The number of the SDK components is small compared to other ITSs. Therefore, the caught variation in latencies is slight (0.5 ms in φ_3 to 0.7 ms in φ_1). For larger scale ITSs, the delay variations would be more noticeable. Nonetheless, it is important to present a clear and relevant proof of concept for the proposed RTA. To meet this goal and ensure a better readability of the paper, the selection of a case of study with a simple structure (number and nature of tasks, message streams, flows, etc.) was required.

5. CONCLUSIONS

In this work, the link between the high-level solutions (risk management, diagnosis or any autonomous navigation tasks) and the material issues is established. The high-level software of safety-critical mechanisms, such as Adaptive Cruise Control (ACC) systems, may suffer from delayed responses to risks due to invehicular latencies. Thus, decisions made by the vehicle control/decision making layer should be sufficiently aware of such delays. Accordingly, a simple algorithm based on a Response Time Analysis (RTA) model is introduced to quantify these latencies. A proof of concept of the suggested RTA is presented via experiments on a Smart distance keeping (SDK) system. The realized experiments proved the proposed RTA efficiency in estimating precise minimum/maximum invehicular communication delays. Since it provides an interval characterization of delays and reports the data validity/freshness, the suggested RTA has promising perspectives in enhancing the recovery strategies in presence of faults. Besides, the in-road risk management policies can take advantage of it to make provably quick and reliable decisions. Topics for future work include the generalization of the suggested RTA to other automotive communication protocols. Besides, more advanced experiments must be realized on a larger-scale vehicular embedded system.

ACKNOWLEDGEMENTS

This work is supported by IMobS3 Laboratory of Excellence (ANR-10-LABX-16-01). This work also received the support of the Hauts-de-France CPER (Contrats de Plan État-Région), through the RITMEA (Recherche et Innovation en Transports et Mobilités Eco-responsables et Automatisées) project.

REFERENCES

Ben Lakhal, N.M., Adouane, L., Nasri, O., and Slama, J.B.H. (2020a). Controller area network reliability:

Overview of design challenges and safety related perspectives of future transportation systems. *IET Intelligent Transport Systems*, 14, 1727–1739.

Ben Lakhal, N.M., Adouane, L., Nasri, O., and Slama, J.B.H. (July 2019a). Interval-based solutions for reliable and safe navigation of intelligent autonomous vehicles. In *2019 12th International Workshop on Robot Motion and Control (RoMoCo)*, 124–130, Poznań , Poland.

Ben Lakhal, N.M., Adouane, L., Nasri, O., and Slama, J.B.H. (June 2019). Interval-based/data-driven risk management for intelligent vehicles: Application to an adaptive cruise control system. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, 239–244, Paris, France.

Ben Lakhal, N.M., Adouane, L., Nasri, O., and Ben Hadj Slama, J. (2020b). Reliable modeling for safe navigation of intelligent vehicles: Analysis of first and second order set-membership ttc. In *Proceedings of the 17th International Conference on Informatics in Control, Automation and Robotics - Volume 1: ICINCO,*, 545–552, Paris, France. SciTePress. doi:10.5220/0009890305450552.

Ben Lakhal, N.M., Adouane, L., Nasri, O., and Ben Hadj Slama, J. (2022). Safe and adaptive autonomous navigation under uncertainty based on sequential waypoints and reachability analysis. *Robotics and Autonomous Systems*, 152, 104065.

Ben Lakhal, N.M., Adouane, L., Nasri, O., and Slama, J.B.H. (July 2019b). Risk management for intelligent vehicles based on interval analysis of ttc. *IFAC-PapersOnLine*, 52(8), 338–343. 10th IFAC Symposium on Intelligent Autonomous Vehicles IAV 2019, Gdańsk, Poland.

Ben Lakhel, N.M., Nasri, O., Gueddi, I., and Slama, J.B.H. (2016). Sdk decentralized diagnosis with vertices principle component analysis. In *2016 International Conference on Control, Decision and Information Technologies (CoDIT)*, 517–522, St. Julian's, Malta.

Davis, R.I., Burns, A., Bril, R.J., and Lukkien, J.J. (2007). Controller area network (can) schedulability analysis: Refuted, revisited and revised. *Real-Time Systems*, 35(3), 239–272.

Fan, W., Li, P., Han, Z., Fan, J., He, J., Wang, Z., and Wang, R. (2021). Dynamic virtual network embedding of mobile cloud system based on global resources in internet of vehicles. *IEEE Transactions on Vehicular Technology*, 70(8), 8161–8174.

Fu, C., Liu, Q., Wu, P., Li, M., Xue, C.J., Zhao, Y., Hu, J., and Han, S. (2019). Real-time data retrieval in cyber-physical systems with temporal validity and data availability constraints. *IEEE Transactions on Knowledge and Data Engineering*, 31(9), 1779–1793.

Gong, H., Li, R., Bai, Y., An, J., and Li, K. (2018). Message response time analysis for automotive cyber–physicalsystems with uncertain delay: An m/ph/1 queue approach. *Performance Evaluation*, 125, 21–47.

Iberraken, D. and Adouane, L. (2022). Safe navigation and evasive maneuvers based on probabilistic multi-controller architecture. *IEEE Transactions on Intelligent Transportation Systems*, 23(9), 15558–15573. doi: 10.1109/TITS.2022.3141893.

Kolekar, S., Gite, S., Pradhan, B., and Kotecha, K. (2021). Behavior prediction of traffic actors for intelligent vehicle using artificial intelligence techniques: A review. *IEEE Access*, 9, 135034–135058.

Lange, R., de Oliveira, R.S., and Vasques, F. (2016). A reference model for the timing analysis of heterogeneous automotive networks. *Computer Standards & Interfaces*, 45, 13–25.

Lee, K. and Lee, M. (2020). Fault-tolerant stability control for independent four-wheel drive electric vehicle under actuator fault conditions. *IEEE Access*, 8, 91368–91378.

Liu, S., Wang, X., Hassanin, O., Xu, X., Yang, M., Hurwitz, D., and Wu, X. (2021). Calibration and evaluation of responsibility-sensitive safety (rss) in automated vehicle performance during cut-in scenarios. *Transportation Research Part C: Emerging Technologies*, 125, 103037.

Nasri, O., Ben Lakhal, N.M., Adouane, L., and Ben Hadj Slama, J. (September 2019). Automotive decentralized diagnosis based on can real-time analysis. *Journal of Systems Architecture*, 98, 249–258.

Nie, J., Yan, J., Yin, H., Ren, L., and Meng, Q. (2021). A multimodality fusion deep neural network and safety test strategy for intelligent vehicles. *IEEE Transactions on Intelligent Vehicles*, 6(2), 310–322.

Shah, M.B.N., Husain, A.R., Aysan, H., Punnekkat, S., Dobrin, R., and Bender, F.A. (2016). Error handling algorithm and probabilistic analysis under fault for can-based steer-by-wire system. *IEEE Transactions on Industrial Informatics*, 12(3), 1017–1034.

Shuai, Z., Zhang, H., Wang, J., Li, J., and Ouyang, M. (2014). Combined afs and dyc control of four-wheel-independent-drive electric vehicles over can network with time-varying delays. *IEEE Transactions on Vehicular Technology*, 63(2), 591–602.

Sun, X., Cai, Y., Wang, S., Xu, X., and Chen, L. (2019). Optimal control of intelligent vehicle longitudinal dynamics via hybrid model predictive control. *Robotics and Autonomous Systems*, 112, 190–200.

Zeng, H., Natale, M.D., Giusto, P., and Sangiovanni-Vincentelli, A. (2010). Using statistical methods to compute the probability distribution of message response time in controller area network. *IEEE Transactions on Industrial Informatics*, 6(4), 678–691.

Zhang, H., Liang, J., and Zhang, Z. (2020). Active fault tolerant control of adaptive cruise control system considering vehicle-borne millimeter wave radar sensor failure. *IEEE Access*, 8, 11228–11240.

Zhang, L., Wang, Z., Ding, X., Li, S., and Wang, Z. (2021). Fault-tolerant control for intelligent electrified vehicles against front wheel steering angle sensor faults during trajectory tracking. *IEEE Access*, 9, 65174–65186.

Zhu, Z., Adouane, L., and Quilliot, A. (2021). Flexible multi-unmanned ground vehicles (mugvs) in intersection coordination based on ε-constraint probability collectives algorithm. *International Journal of Intelligent Robotics and Applications*, 5(2), 156–175.