Risk reduction of experimental autonomous vehicles: The Safety-Bag approach

Manel Brini, Paul Crubillé, Benjamin Lussier, Walter Schon

Risk reduction of experimental autonomous vehicles: The Safety-Bag approach

Manel Brini*, Paul Crubillé*, Benjamin Lussier* and Walter Schön*
*Sorbonne université, Université de Technologie de Compiègne, CNRS,
Heudiasyc UMR 7253, CS 60 319, 60203 Compiègne Cedex France
Email: {manel.brini,paul.crubille,benjamin.lussier,walter.schon}@utc.fr

***Abstract*—This work presents a study concerning the dependability of experimental autonomous vehicles in the Heudiasyc laboratory. This study confirms that the use of these vehicles involves significant risks during experiments, and that integration of a Safety-Bag component in the vehicle architecture can significantly reduce these risks. In this paper, we define a severity scale and propose a FMEA (Failure Mode Effects Analysis) of an autonomous vehicle. We also present the implementation of our safety-bag component and how it can reduce risks.**

***Index Terms*—Safety-Bag, Dependability, Safety, FMEA, Autonomous vehicle, Fault tolerance.**

I. Introduction

Autonomous vehicles are fast and powerful mobile robots, able to accumulate a large amount of energy. Thus, they are a source of danger to their operator and their environment. Because they are complex systems including artificial intelligence software based on declarative mechanisms, their validation and verification are difficult [1].

In the case of experimental vehicles, the constant evolution of software, as well as shorter validation steps than for industrialized systems, make it an even more complex problem [10].

Moreover in academic research context, the vehicle is used to validate or invalidate new algorithms or methods. The software components are thus prone to faults and incorrect behaviors.

For the Heudiasyc laboratory's vehicles, we conducted a dependability study using two analysis techniques: a FMEA and fault trees [6]. This study shows that even with trained driver able to recover the system manually, the risks of accidents are high.

To ensure better safety for autonomous vehicles, we propose to integrate a Safety-Bag component into these systems architecture. The Safety-Bag oversees safety rules [8], [9]. It can also supervise software components liveness and timeliness, and detects some errors in sensors and actuators. In case of a detected error, it is able to alert the pilot quickly and put the vehicle in state permitting manual recovery in good conditions [7], [11].

Based on two embedded computers, our Safety-Bag component integrates sufficient redundancy to prevent failures caused by a single fault. Our first experimentations with the Safety-Bag confirm the efficiency of this approach.

II. Risk analysis of experimental autonomous vehicles

On Heudiasyc's experimental autonomous vehicles, a pilot is ready to resume manual control at any time. He oversees autonomous driving and is ready to react to failures such as software or sensors errors. It usually requires a few seconds for the driver to react, but depending on the vehicle's failure it can become uncontrollable in a very short time. We present in this section our autonomous vehicles architecture and the FMEA that we have realized.

A. Autonomous vehicle architecture

The Heudiasyc laboratory developed two types of automated cars. The first is based on Renault's FLUENCE model, and the second on Renault's Zoe model. In this study, we focus on a FLUENCE automated car. The solutions for the brake and acceleration are purely mechanical (controlled by analog signals). Although not a sport car, the automated vehicle is able to accelerate from 0 km/h up to 50 km/h in 4 seconds, can reach 130 Km/h and weighs almost a ton and a half (including pilot, calculators, sensors, etc.).

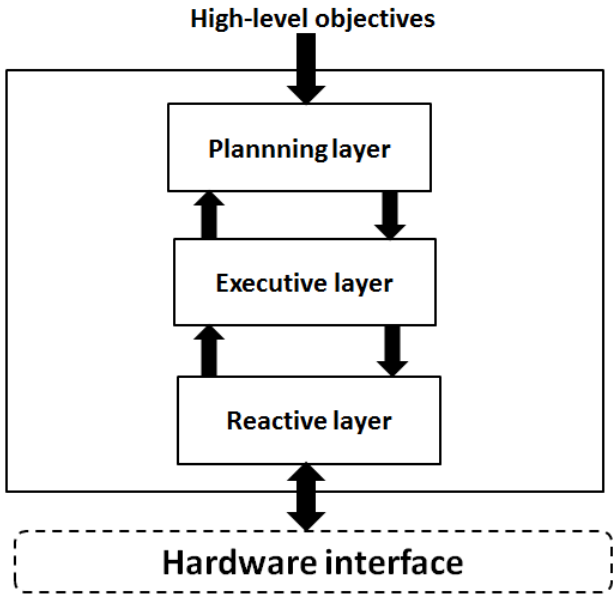

Fig. 1. A hierarchical architecture style

The vehicle's autonomy is implemented through the classic hierarchised architecture presented in Figure 1 [1]:

- A reactive layer (or functional layer): It is responsible for the execution of basic tasks requested by the upper layers. It performs a set of low-level actions and controls hardware actuators. This layer's frequency is around 100 Hz to allow real-time actions and reactions.
- An executive layer implementing situation categorization and reactive navigation: This layer supervises the functional layer and uses data derived from sensors to identify the vehicle's situation and to generate trajectories that the lower layer will reactively follow. This layer's frequency is around 10 Hz.
- A planning layer: this layer produces a high level plan (a succession of roads and intersections) that the vehicle will follow to go from its current position to its destination. As planning can take several seconds (or even minutes) to generate a plan, this layer's frequency is quite low and totally unadapted to real time constraints.

The Safety-Bag component aims to tolerate many faults from the software components of the three layers (valued or timely faults). It will also be able to detect some faults in sensors and actuators, depending on their redundancy.

We present in this section the architecture of a control application of an experimental autonomous vehicle. This architecture is presented in Figure 1 using an UML deployment diagram.

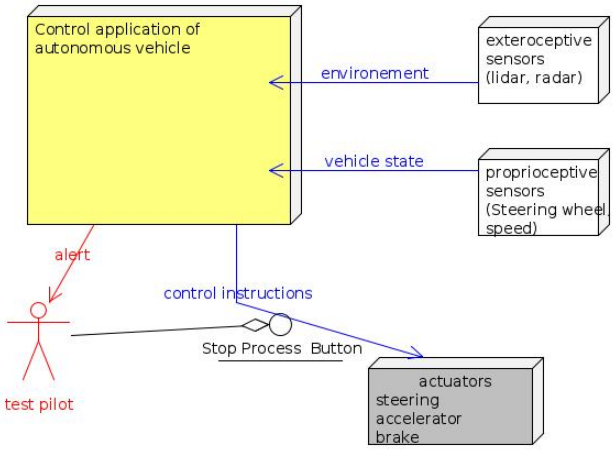

Fig. 2. Deployment without Safety-Bag

The vehicle is equiped with both proprioceptive sensors (ie. sensors giving information on the system's state; such as odometers, inertial measurement unit, etc.) and with exteroceptive sensors (ie. sensors that give information on the system's environment; such as GPS, lidar, camera, radar, etc.).

The exteroceptive sensors allow the identification of the vehicle environment and the driving situations. They are thus essential to decide road maneuvers and then trajectories. The proprioceptive sensors are mostly used to follow the defined trajectory and are essential to detect system's failures.

Acceleration, braking and steering commands are sent to the actuators through analog signals produced by the command application and D/A converters. A Stop Process Button allows the driver to disable the automatic control and regain manual control. Visual and sonic alarms can alert the pilot of the application's state and possible problems if the command application can produce a correct diagnosis.

B. FMEA analysis of an autonomous vehicle

The goal of the Failure Mode and Effects Analysis is to establish the list of failures of each component and determine their occurrence rate and the severity of their consequences on the system's behavior. This analysis is then used to define risk reduction measures, and particularly avoid catastrophic failures, that are failures that can cause catastrophic consequences in terms of human life or economical value.

1) Severity rating: The first step of the FMEA is to define the severity scale for the system's failures. This is used to grade each components failure in the analysis, but also to guide risk reduction approaches.

N° G	Severity scale
0	nominal operation
1	The automated control is near its functioning limits/ need to regain control in 10s
2	The automated control is in a degraded mode/need to regain control in 2s
3	The automated control has stopped/ need of immediate recovery with alarms warning
4	The automated control has stopped/ need of immediate recovery without alarms warning
5	The automated control produces aberrant commands.

Fig. 3. Severity scale

2) FMEA table: The FMEA is in the form of a table detailing for each component its different failure modes, and the effects associated, including their severity and occurrence rate. For our application, we have further differentiated between effects on the control system (Computing effects) and effects on the vehicle's behavior (Vehicle effects).

We also detail the possible detection and correction means, and the time scale that they will take. Figure 4 shows an extract of the FMEA analysis for the autonomous vehicle for only four components and their failures. This illustrative example shows that high-risk situations have high probabilities of occurrences.

The occurrence rate $\lambda(t)$ is the inverse of the MTBF and is difficult to obtain without making a large number of experiments. We chose pessimistic values in regard to hardware failure values, from our knowledge of the system and its components.

Elements	Types of failure	Computing effects	Vehicle effects	λ (t)	Detection		Action		Gravity of consequences		Ref
					means	t	means	t	G	Comments	
Control Application	Locked down	The last outputs are maintained	No control: acceleration, brake, steering torque unchanged.	~10^-3	Pilot	>2s	Pilot	+2s	5	Uncontrollable vehicle	1
	Voltage off	Outputs set to 0.	-No Acceleration, -No Autonomous brake, -Electric Power Steering is in default, dropped wheel.	~10^-5	Pilot	>2s	Pilot + Emergency Stop Button	+2s	4	The Electric Power Steering stays in failure state.	2
	Failure of Keyboard or screen	Loss of control by the operator	No effect	~10^-5	Operator	>4s	Pilot + Emergency Stop Button	+2s	1	—	3
Speed sensor	Sensor	If diagnostic is supported by application	Alert	~10^-5	Application	0.1s	Pilot	<2s	3	Is the diagnosis reliable?	4
		If there is no diagnosis from the application => aberrant command.	No alert and excessive speed	~10^-5	Pilot	4s	Pilot	2s	5	Excessive speed not adapted to the situation.	5
Accelerator actuator	blocked to 0	_	No acceleration	~10^-5	Pilot	2s	Pilot	+2s	2	This sort of incident occurs testing our vehicle.	6
	locked down to a value	_	Acceleration without request						5		
Brake actuator	Electric motor or wire used to activate the brakes	_	No braking	~10^-5	Pilot	2s	Pilot	+2s	5	Our solution to operate the brakes isn't as reliable as the automotive brakes.	7

Fig. 4. An extract of FMEA for autonomous vehicles without Safety-Bag

In our example, locked down failures of the control application or the actuators lead to an uncontrolable vehicle as the actuators commands are not updated. The severity of such failures is 5. Sensors failures can also cause dangerous situations if the software generates inappropriate orders due to their inputs.

In our experimental vehicle, actuators failures are more frequent than in a regular car. The components added to interface the actuators (such as the motor and the wire pulling on the brake pedal) are not as reliable as industrialized automotive equipment.

The software components used in our vehicles are developed in an experimental context and can not be tested extensively before being integrated into the complete system. We believe that their failure rates are probably orders of magnitude higher than what we wrote into the FMEA table. As part of a dependability analysis, a FMEA should be as complete as possible, and may include a large number of lines depending on the number of considered components. In our case, the table for the vehicle without safety-bag has 14 components and 11 additional elements for the Safety-Bag .

III. The Safety-Bag Component

To reduce the risks exhibited by the FMEA we implement a Safety-Bag that can reject dangerous automated commands and detect aberrant behaviors, raising alarms when necessary to allow the driver to take control as quickly and easily as possible.

A. Concept and state of the art

Independent from the functional system, a Safety-Bag or *Independant Safety Component* is responsible for supervising the system's commands and enforcing safety rules to avoid catastrophic failures of the system. To avoid common cause failures, it must be specified and developed independently from the functional systems, and have means of action and detection independent from the faults to be tolerated. It monitors the operational system, and in case of danger set the system on a safe state [1].

This approach was used effectively for critical applications such as: the ELEKTRA rail system [12], the SPAAS project for an autonomous spacecraft [2], the SPIN nuclear plant monitoring [13], the SMOF framework and a mobile robot excavator robot [9].

B. Implementation of the Safety-Bag component

Implementing a fault tolerant component introduces new possible faults and failures in the system. As a single fault should not cause failure of the whole Safety-Bag, it is composed of two computers monitoring each others. One is called the rules checker, and the other the supervisor.

The control command application sends now orders to the Safety-Bag rules checker. Thereby, the Safety-Bag rules checker is able to check and modify these orders before sending them to the actuators.

The Safety-Bag rules checker examines the liveliness of the control application and oversees its liveliness. It also enforces other safety rules, for example using speed sensors to ensure that the vehicle's speed is adequate. Moreover, it also verifies that the orders that it sends to the actuators have real and consistent effects.

The two computers monitor each other by the exchange of heartbeats. If the Safety-Bag supervisor no longer receives the signal from the Safety-Bag rules checker, it disables the vehicle actuators via a MOSFET. This has the same effect as the pilot pressing the Stop Process Button. If the Safety-Bag rules checker no longer receives the signal from the supervisor, it asks the pilot to stop the experiment even if there is no immediate risk, as the Safety-Bag is now vulnerable to a single fault, and as without it the system's behavior could no longer be guaranteed.

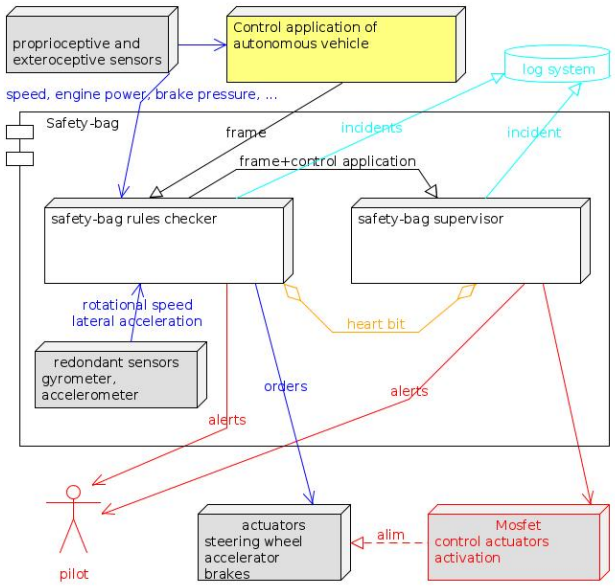

Fig. 5. Deployment with Safety-Bag

The Safety-Bag can raise visual and sonic alarms and log all safety related events for post experimentation examination.

IV. Conclusion

In our study, a Safety-Bag or Independent Safety Component is proposed to reduce the risks of experimental autonomous vehicles. This approach detects control failures such as locked outputs or dangerous commands as well as some actuators and sensors failures.

It also recovers from the detected errors, either by rejecting dangerous commands, enforcing actions to put the system in a safe state, or utltimately giving the control back to the operator. However, to avoid introducing new logical faults in the system, potentially causing catastrophic failures, the behavior of the Safety-Bag must be simple and easy to validate, which limits the expressiveness of safety rules that it provides.

To prevent failure in the Safety-Bag, it comports enough redundancy to detect and tolerate internal faults.

The elimination of the most serious and most immediate risk situations is done by alerting the pilot and switching to manual mode. Thus, trained and vigilant pilots remain essential in the vehicles, but the Safety-Bag ensures the detection of most dangerous errors and allows more response time to the driver.

Acknowledgment

This work was carried out and funded under the EQUIPEX ROBOTEX. It was supported by the French Government, through the program *Future Investment* managed by the National Research Agency. (Reference: ANR-10-EQPX).

References

[1] Lussier B. (2007). Tolérance aux fautes dans les systèmes autonomes.
[2] Blanquart J., Fleury S., &Hernek, M. (n.d.). Software Safety Supervision On-board Autonomous Spacecraft.
[3] David P. & Guiochet J. (2005). Etude et analyse de différents dispositifs externes de sécurité-innocuité de type safety bag.
[4] Baudin ., Blanquart J. P., Guiochet J. & Powell D. (2007). Independent Safety Systems for Autonomy.
[5] Powell D., & Thévenod-Fosse P. (2002, October). Dependability issues in ai-based autonomous systems for space applications. In 2nd IARP/IEEE-RAS Joint Workshop on Technical Challenge for Dependable Robots in Human Environments (pp.163-177).
[6] Amina Mekki-Mokhtar. Processus d'identication de propriétés de sécurité-innocuité vériables en ligne pour des systèmes autonomes critiques. Robotics.Université Paul Sabatier Toulouse III, 2012.French.
[7] Mekki Mokhtar A., Blanquart J.-P. & Guiochet J. (n.d.). Safety Trigger Conditions for Critical Autonomous Systems, 18th Pacific Rim Int. Symp. on Dependable Computing (PRDC 2012), Niigata, Japan, 2012.
[8] Pace C., & Seward D. (n.d.). An approach to safety for a robotic excavator.1-7.
[9] Pace C., Seward D., & Sommerville I. (n.d.). A Safety Integrated Architecture for an Autonomous Excavator. IEEE,Proc. 17th Int. Symp. on Automation and Robotics in Construction, (ISARC, 2000), Taiwan, 2000.
[10] Pietre-Cambacedes L. (2010). Des relations entre sureté et sécurité. Paris: Télécom ParisTech.
[11] Guiochet J., Powell ,Baudin E. & Blanquart J. P. (2008, May). Online safety monitoring using safety modes.In Workshop on Technical Challenges for Dependable Robots in Human Environments (pp.1-13).
[12] P. Klein, The Safety-Bag Expert System in the Electronic Railway Interlocking System Elektra, Expert System with Applications, 3(4):499-506, 1991.
[13] G. Guesnier, J. F. Hamelin, & J. M. Peyrouton, Centrale nucléaires N4: l'informatique au service d'une conduite plus sure, Epure, (56):59-74, 1997.