Case Study Report : Safety rules synthesis for an autonomous robot

Lola Masson, Jérémie Guiochet, Hélène Waeselynck

Case Study Report : Safety rules synthesis for an autonomous robot

Lola Masson, Jérémie Guiochet, Hélène Waeselynck
University of Toulouse, LAAS - CNRS
Toulouse, France
firstname.lastname@laas.fr

1. Introduction

As autonomous systems are evolving in complex and dynamic environments and in the vicinity of humans, the safety aspects are crucial. We use fault tolerance techniques, particularly monitoring techniques, to ensure safe operation despite the occurrence of faults or adverse situations. A safety monitor is a device responsible for safety only. It is in charge of observing the system via sensors and of intervening on it via actuators according to safety rules. While this approach is widespread in embedded systems [1] and robotics [2], [3], the identification of the safety rules to implement often lacks a rigorous approach.

This paper presents the process we use to define the safety rules implemented on the safety monitor. This approach is applied to the industrial case study presented in Section 2. We first perform a risk analysis presented in Section 3. From the list of hazards, we extract safety invariants, which are conditions to be met to preserve the system safety. The invariants are modelled as explained in Section 4. The safety invariants and available interventions are then combined to create safety rules. To automate this process we developed the SMOF tool [4], [5] applied in Section 5.

2. An industrial case study

The studied system is a robot from the French company STERELA (figure 1). Its mission is to control the lights along the airport runways.

The robot consists of a mobile platform and a commutable payload. The payload is a photometric sensor deported on the side of the platform. The deported sensor moves above the lights (15-20cm) with a maximum speed of 1.4m/s. A human operator is supervising the mission with a digital tablet from the extremity of the runway. As the robot operates at night and at long distances, the operator has no direct visual contact with it.

3. Risk analysis with HAZOP-UML

We use the method HAZOP-UML [6] to perform the risk analysis at an early stage of the system development, based on the UML diagrams that are created in the first steps of the project. In these diagrams every element is considered to determine which possible deviations can occur.

Figure 1. Robot from STERELA

Among 334 possible deviations, the analysis yields 38 deviations with high severity and 48 deviations with medium severity. Some of these hazards are gathered and we eventually obtain 7 hazards:

1) Collision with an obstacle (including lights);
2) Fall of the robot - the robot can fall on the ground or on an operator;
3) Deposit of debris on the runway;
4) Absence of power supply during operation;
5) Movement in an unauthorized zone - some areas of the airport are not allowed to the robot;
6) Inability for the operator to locate the robot;
7) Dangerous velocity - the robot can go too fast in terms of linear or angular velocity.

4. Modelling

After determining the list of hazards, we formalize the safety invariants that the monitor will consider. Not every hazard can be addressed by the monitor as observations or interventions may not be available. We retain the 5 following invariants:

1) The robot must not collide with an obstacle (static or mobile);
2) The robot must not fall;
3) The robot must not enter a prohibited zone;

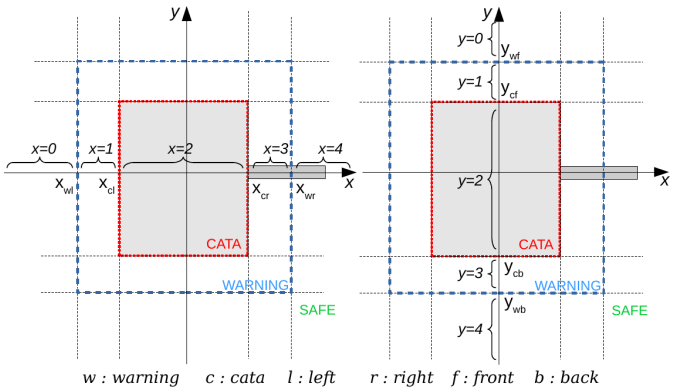

Figure 2. Robot (in grey) with thresholds for the obstacle avoidance

4) The robot must not travel a long distance if the communication with the operator is broken;
5) The robot must not exceed a certain velocity.

Here we present only the modelling of the first invariant. It represents an interesting problem because the robot must be allowed to move very close to some obstacles (the lights) so that its deported sensor passes above them. Still, it must not be allowed to move very close to humans or other vehicles. We decided to separate the obstacles in two types: high obstacles, including humans, which are too tall to pass under the sensor, and low obstacles (20 cm max), which can pass under the sensor. It imposes that the robot can make the difference, which is possible by means of simple laser sensors. The monitor then uses the following observations:
x : abscissa of the obstacle in the robot's referential
y : ordinate of the obstacle in the robot's referential
v : robot velocity
$type_{obst}$: type of the obstacle (high or low)
The variables x and y are discretized according to the concept of warning state. A warning state is entered when an obstacle gets into an area close to the robot (the thresholds depend on the type of obstacle). The dotted rectangular area in Figure 2 illustrates this for low obstacles. Velocity is also discretized: robot is standstill ($v = 0$) or moving ($v = 1$). According to this discretization, the catastrophic states violating the invariant can be defined as follows, for low obstacles ($type_{obst} = 0$):
$cata : x = 2 \wedge y = 2 \wedge v = 1 \wedge type_{obst} = 0$

To maintain the invariant (i.e., avoid $cata$), interventions must be applied when entering the warning states. Two interventions are available: the full stop which stops the robot ($full\_stop : next(v) = 0$), and the inhibition of the rotation that prevents the robot from turning on the side of the deported sensor ($inhib\_rotation : next(x)! = x - 1$). The latter intervention is effective only for fixed obstacles: we assume that low obstacles do not move.

5. Safety rule synthesis with SMOF

To determine which interventions should be applied in warning states, we used the tool SMOF (Safety MOnitoring Framework). This tool is based on the model-checker NuSMV [7], and synthesizes rules for assuring safety but also permissiveness (i.e all non-catastrophic states are reachables).

For the obstacle invariant detailed in the previous section, 64 valid strategies are generated. The synthesis took 51s on an Intel Core i5-3437U CPU @ 1.90 GHz x 4 with 16 GB of memory.

The developer has to make a choice among the 64 strategies. As the system is autonomous, we could suppose that the intervention inhibiting the rotation has to be favoured over the full stop which is very restrictive. Among the 64 strategies synthesised, we choose the strategy which does not use full stop when the obstacle is on the right so the robot can pass above lights: full stop is applied whenever the obstacle is in front, behind or left and inhibition of the rotation is applied when the obstacle is on the right.

6. Conclusion and perspectives

We propose in this paper to apply a rigorous process to assist the developer, from the risk analysis to the synthesis of safety strategies implemented on a safety monitor. SMOF provides a simple template for modelling, and automatically determines which strategy fulfills the safety and permissiveness requirements.

Our approach has some limitations. First, the synthesis of the rules is highly dependent on the choices made for the modelling steps (discretization, thresholds,...). Then, we would like to integrate several monitors with different integrity levels in order to use less restrictive interventions.

Acknowledgments

This project has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 644400 [8].

References

[1] R. Woodman, A. F. T. Winfield, C. Harper, and M. Fraser, "Building safer robots: Safety driven control," *The International Journal of Robotics Research*, vol. 31, no. 13, pp. 1603–1626, Nov. 2012.

[2] D. Crestani, K. Godary-Dejean, and L. Lapierre, "Enhancing fault tolerance of autonomous mobile robots," 2014.

[3] C. Pace, D. Seward, and I. Sommerville, "A Safety Integrated Architecture for an Autonomous Excavator," 2000.

[4] SMOF, "Safety Monitoring Framework," LAAS-CNRS Project, https://www.laas.fr/projects/smof, accessed 2016-07-01.

[5] M. Machin, F. Dufossé, J.-P. Blanquart, J. Guiochet, D. Powell, and H. Waeselynck, "Specifying Safety Monitors for Autonomous Systems Using Model-Checking," in *Computer Safety, Reliability, and Security*, Sep. 2014, no. 8666.

[6] J. Guiochet, "Hazard analysis of human–robot interactions with HAZOP–UML," *Safety Science*, vol. 84, pp. pp. 225–237, Apr. 2016.

[7] "NuSMV home page." [Online]. Available: http://nusmv.fbk.eu/

[8] CPSELabs, "Cyber-Physical Systems Engineering Labs," Project funded by the European Union, Horizon2020 Programme, www.cpse-labs.eu.