Safety-Bag pour les systèmes complexes

Manel Brini

Par **Manel BRINI**

Safety-Bag pour les systèmes complexes

Thèse présentée
pour l'obtention du grade
de Docteur de l'UTC

Soutenue le 23 novembre 2018
Spécialité : Technologies de l'Information et des Systèmes : Unité de recherche Heudyasic (UMR-7253)

D2444

Safety-Bag pour les systèmes complexes

Manel BRINI

Thèse soutenue le 23 novembre 2018 devant le jury composé de :

Président:

Véronique Cherfaoui
Professeur des universités
Univ de Technologie de Compiègne

Rapporteurs:

Jérémie GUIOCHET
Maître de conférences
Univ de Toulouse

Simon COLLART-DUTILLEUL
Directeur de recherches
IFSTTAR/ESTAS

Examinatrices:

Véronique CHERFAOUI
Professeur des universités
Univ de Technologie de Compiègne

Julie BEUGIN
Chargée de recherches
IFSTTAR/ESTAS

Directeurs de Thèse:

Walter SCHÖN
Professeur des universités
Univ de Technologie de Compiègne

Benjamin LUSSIER
Enseignant chercheur
Univ de Technologie de Compiègne

Université de Technologie de Compiègne

Laboratoire Heudiasyc UMR CNRS 7253

23 - 11 - 2018

Avant propos

Les travaux présentés dans ce mémoire ont été réalisés au Laboratoire Heudiasyc de l'Université de Technologie de Compiègne. Je remercie les directeurs successifs du laboratoire Heudiasyc, Messieurs Ali Charara et Philippe Bonnifait, de m'avoir accueillie dans ce laboratoire. Je remercie Madame Véronique Cherfaoui, Professeur à l'Université de Technologie de Compiègne, pour avoir présidé mon jury de thèse, ainsi que :

- Monsieur Jérémie Guiochet, Maître des conférences à l'Université de Toulouse,
- Monsieur Simon Collart-Dutilleul, Directeur de recherches à l'IFSTTAR/ESTAS,
- Madame Julie Beugin, Chargée de recherche à l'IFSTTAR/ESTAS,

pour avoir participé à ce jury. Je remercie en particulier Messieurs Jérémie Guiochet et Simon Collart-Dutilleul qui ont accepté la charge d'être rapporteurs.

Je remercie énormément mes encadrants de thèse : Benjamin Lussier et Walter Schön pour la confiance qu'il m'ont accordée pour réaliser ce travail, pour leurs conseils et leur soutien. Merci à Benjamin pour sa rigueur scientifique, ses conseils et ses relectures diligentes. Je remercie également Walter pour son dynamisme, sa disponibilité et son aide plus que précieuse.

Un grand merci à mon superviseur Paul Crubillé, qui m'a accompagné tout au long de ma thèse et qui a également suivi mes travaux, pour ses remarques toujours très pertinentes, ses conseils et aussi pour son humour légendaire. Je remercie aussi Yohan et Thierry pour leur aide sur le banc VILAD et la plateforme véhicule.

Merci aux membres de mon célèbre bureau 148, avec qui j'ai passé de nombreuses journées bien accompagnée. Merci en particulier aux anciens et nouveaux camarades Subeer, Lanting, Nicolas, Rayen et Augustin.

Je remercie bien évidemment mes parents Mohamed et Samira qui ont toujours cru en moi et qui ont fait en sorte que je sois là où je suis aujourd'hui. Une pensée à mon frère Houssem, mes sœurs Marwa et Sourour, mon cher Moetez, ma deuxième famille Crubillé (Dominique, Marlène, Raphaêl et Paul), mes deux chers oncles M'hammed et Mohamed et ma chère tante Dhekra. Le parcours que j'ai fais jusqu'à aujourd'hui n'aurait pas été possible sans leur soutien et leurs conseils.

Table des matières

Liste des figures

Liste des tableaux

Introduction

Les véhicules autonomes sont présentés aujourd'hui par les médias comme prêts à être commercialisés (dès 2020). Cependant, les incidents et les accidents des derniers mois sur les véhicules autonomes expérimentaux circulant sur le domaine public (Tesla, Uber, etc.) ont mis en évidence que la sûreté de fonctionnement n'a pas été suffisamment traitée et n'a pas encore de solutions satisfaisantes. La complexité des véhicules autonomes, des tâches de conduite et l'environnement ouvert et incertain dont lequel ils opèrent, constitue un défi pour appliquer les techniques de sûreté de fonctionnement.

De plus, la défaillance de ces véhicules exposent les passagers et les personnes présentes à proximité à des accidents graves. Des dégâts environnementaux et des pertes économiques sont également à craindre. Les véhicules autonomes sont ainsi des systèmes critiques, qui nécessitent de réduire les risques de leurs défaillances.

Des systèmes de conduite ou de pilotage automatiques existent et sont utilisés dans l'aviation et en ferroviaire. Cependant, à l'exception des métros automatiques de type *VAL* systèmes CBTC (Communication-Based Train Control) en conduite automatique intégrale, ces systèmes ne sont pas au niveau maximal d'autonomie. En effet, la plupart sont sous la supervision d'opérateurs humains professionnels formés et vigilants qui sont à même de reprendre la conduite ou le pilotage en manuel. Les systèmes automobiles existants qui sont actuellement de niveau d'autonomie 2 au maximum (c'est-à-dire des aides à la conduite évoluées), sont parfois mal utilisés ou détournés par les conducteurs. L'objectif actuel des constructeurs est ainsi de réaliser des systèmes de conduite pour l'automobile assurant les niveaux d'autonomie 4 et 5, c'est-à-dire d'avoir des systèmes capables de réaliser certaines tâches (ou toutes pour le niveau 5) sans intervention humaine.

Il faut aussi noter que pour le ferroviaire et pour l'aviation, la problématique de la perception des dangers est sensiblement plus simple que dans l'automobile. Les temps de réactions imposés pour le système ou pour un éventuel opérateur humain sont plus longs (10 secondes voire une minute pour le train, quelques minutes pour l'avion hors phase de décollage et d'atterrissage contre moins de deux secondes pour la voiture). Il est par ailleurs économiquement acceptable de doubler voire de tripler les parties critiques des systèmes de perception et de contrôle et de commande pour

un train ou pour un avion alors que ce n'est pas envisageable pour une automobile de grande diffusion.

Pour les véhicules autonomes, la tolérance aux fautes doit ainsi être assurée en limitant au maximum les équipements capteurs, calculateurs et actionneurs redondants. Pour ce faire, nous nous sommes intéressés à un dispositif *Composant Indépendant de Sécurité*, que nous appelons *Safety-Bag*. Ce dispositif peut être constitué typiquement de deux calculateurs et de quelques capteurs simples (accéléromètre, gyromètre, voltmètre et ampèremètre). Ce Safety-Bag permet de vérifier en ligne qu'un ensemble de nécessités de sécurité (c'est-à-dire des propriétés nécessaires pour assurer la sécurité-innocuité du système) sont respectées. Si l'une de ces nécessités de sécurité est violée, le Safety-Bag intervient en faisant le rétablissement soit en inhibant, soit en forçant une action de sécurité pour remettre le système dans un état sûr. Nous avons étudié dans nos travaux l'identification de ces nécessités de sécurité et leur apport pour la sécurité-innocuité du véhicule autonome.

Le premier chapitre décrit les concepts et la terminologie utilisés dans notre manuscrit, notamment les notions de la sûreté de fonctionnement, la problématique du véhicule autonome et les différentes méthodes d'analyse de risques. Pour analyser les risques dans notre approche, nous avons utilisé les deux techniques ascendantes d'analyse de risques AMDEC et HazOp-UML. L'AMDEC est une méthode classique et connue depuis de nombreuses années, l'HazOp-UML est une variante d'HazOp utilisé au laboratoire LAAS et appliquée à la robotique.

Le deuxième chapitre présente le dispositif de sécurité-innocuité Safety-Bag et ses différentes applications dans la recherche et l'industrie.

Le troisième chapitre entre dans le cœur de nos travaux de thèse en présentant les analyses de risques AMDEC et HazOp-UML des véhicules autonomes que nous avons effectuées et l'ensemble des exigences de sécurité qui en résulte.

Dans le quatrième chapitre, nous précisons les moyens d'implémentations de ces exigences de sécurité. Si le Safety-Bag fait partie de ces moyens d'implémentation, nous dérivons les nécessités de sécurité correspondantes. Une nécessité de sécurité est constituée d'une condition de déclenchement de sécurité et d'une intervention de sécurité. Pour que le dispositif de sécurité intervienne avant que le système ne sorte des états sûrs, nous déterminons pour chaque conditions de sécurité une valeur seuil de déclenchement, typiquement issue d'une valeur de danger et d'une marge de sécurité.

Dans le cinquième chapitre, nous complétons l'analyse de sûreté de fonctionnement par une nouvelle analyse AMDEC prenant en compte l'impact du composant Safety-Bag sur le comportement du véhicule. Nous réalisons également une analyse de sécurité portant sur le Safety-Bag lui-même pour nous assurer qu'il n'introduit

pas de nouvelle défaillance catastrophique pour le système. Enfin, nous validons le Safety-Bag proposé en testant 6 des nécessités de sécurité développées face à des injections de faute sur un banc Vehicle-in-the-Loop en utilisant le véhicule autonome expérimental IRIS du laboratoire Heudiasyc. Nous présentons également quelques expériences portant sur la vitesse de réaction de conducteurs avec et sans Safety-Bag dans ce même véhicule expérimental sur la piste de conduite SEVILLE.

Sûreté de fonctionnement et véhicules autonomes

Sommaire

Les systèmes autonomes critiques et en particulier les véhicules autonomes sont des systèmes complexes qui exigent un grand soin lors de leur conception, lors de leur implémentation et durant leur fonctionnement. Ce sont des systèmes critiques, car ils peuvent provoquer des dégâts catastrophiques pour les êtres humains qui les utilisent et l'environnement dans lequel ils opèrent tels que des blessures, des pertes de vie humaine ainsi que des pertes financières. Afin de prendre des décisions dans des environnements ouverts et complexes, les véhicules autonomes utilisent des mécanismes décisionnels et des logiciels d'intelligence artificielle basés sur les mécanismes déclaratifs . Cependant, ces mécanismes ont des comportements difficiles à prédire et leur bon fonctionnement n'est actuellement pas garanti.

Il nous paraît ainsi nécessaire d'appliquer aux systèmes autonomes les méthodologies de la sûreté de fonctionnement pour nous permettre d'avoir une confiance justifiée dans leur sécurité-innocuité et leur fiabilité.

Ce chapitre décrit la terminologie et les concepts autour desquels s'articulent nos travaux de thèse. Nous abordons tout d'abord les notions relatives aux systèmes sûrs de fonctionnement puis nous présentons les véhicules autonomes ainsi que les menaces auxquelles ils sont exposés. Nous concluons en présentant des travaux et des concepts autour de la notion de risque utilisés dans l'industrie pour des analyses de sécurité-innocuité.

1.1 La sûreté de fonctionnement

La sûreté de fonctionnement d'un système est la propriété qui permet à ses utilisateurs de placer une confiance justifiée dans le service que le système leur délivre, c'est-à-dire dans son comportement tel qu'ils le perçoivent [Avizienis et al., 2004] [Laprie et al., 1996]. Cette définition souligne que la confiance que nous avons en un système doit être justifiée par des analyses et des méthodes objectives et précises. Pour préciser cette notion, nous allons développer différents points dans la suite de cette section. Dans un premier temps, nous rappellerons la caractérisation de la sûreté de fonctionnement basée sur la définition d'*attributs*, d'*entraves* et de

moyens et ensuite, nous nous intéresserons dans le cadre de nos travaux à un moyen particulier : la *tolérance aux fautes*.

1.1.1 Concepts généraux de la sûreté de fonctionnement

Comme présenté dans la figure 1.1, la sûreté de fonctionnement peut être caractérisée par *ses attributs*, les propriétés attendues du système ; *ses entraves*, les comportements inacceptables du système qui sont causes ou conséquences de la non-sûreté de fonctionnement ; *ses moyens*, les méthodes qui permettent à un système d'être apte à accomplir correctement sa fonction en plaçant une confiance justifiée dans le service qu'il délivre.

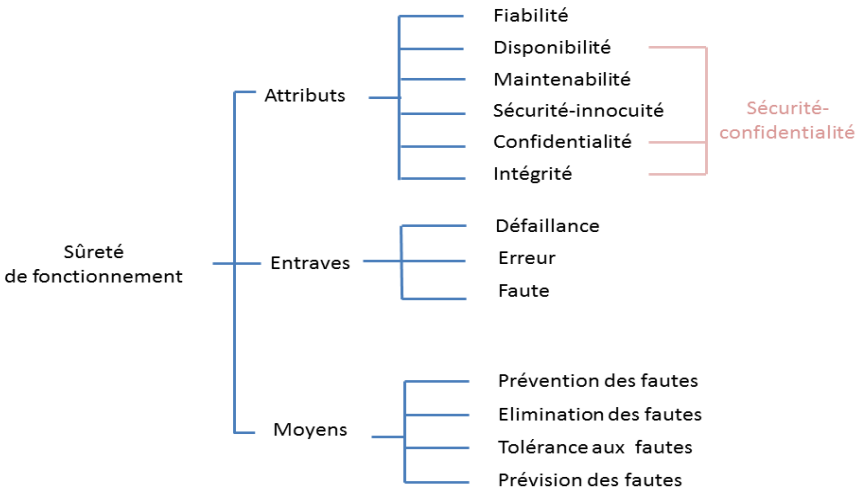

Figure 1.1 – Arbre des concepts de la sûreté de fonctionnement

1.1.1.1 Attributs de la sûreté de fonctionnement

La sûreté de fonctionnement peut être caractérisée par différentes propriétés complémentaires, qui sont *ses attributs*. Ces attributs sont :

- La fiabilité (ou *la continuité de service*) : la délivrance continuelle et correcte du service requis,
- La disponibilité : le fait qu'un système soit prêt à l'utilisation,
- La maintenabilité : l'aptitude aux réparations et aux évolutions.
- La sécurité-innocuité (ou *safety*) : l'absence de conséquences catastrophiques pour l'environnement et l'utilisateur du système,

- La confidentialité : l'absence de divulgations non autorisées de l'information,
- L'intégrité : l'absence d'altérations inappropriées de l'information,

  La sécurité-confidentialité (ou *security*) est un composé de confidentialité, d'intégrité et de disponibilité.

  Dans ce manuscrit, nous nous intéressons en particulier à la fiabilité et à la sécurité-innocuité (que nous appellerons simplement *sécurité* dans la suite de notre manuscrit). Ces deux objectifs sont en partie en opposition, puisque des mécanismes améliorant l'un risquent d'entraver l'autre. En effet, si le véhicule autonome est dans une situation potentiellement dangereuse, la sécurité peut exiger d'arrêter et d'évaluer la situation, par contre la fiabilité demanderait de continuer le service et donc de poursuivre le trajet.

1.1.1.2 Les entraves

Les entraves à la sûreté de fonctionnement sont les circonstances indésirables pour le système. Elles consistent en :

- La défaillance : elle survient lorsque le service dévie de l'accomplissement de la fonction du système (Exemple : dépassement d'une vitesse limite, collision avec un piéton, sortie de la route).
- L'erreur : elle est la partie erronée de l'état du système qui est susceptible d'entraîner une défaillance. Par propagation, une erreur peut créer de nouvelles erreurs (Exemple : une absence de détection d'obstacle peut se propager en la génération d'une trajectoire erronée, perte de message réseau soit suite à une entrée dans un tunnel ou un passage par une forêt, soit la distance est trop importante de la source, etc.).
- La faute : elle est la cause établie ou supposée d'une erreur. Suivant leur origine, leur nature, ou leur phase de création, les fautes sont classées en trois types principaux :
  - ◇ fautes de développement : qui incluent les fautes de conception (Exemple : un algorithme erroné ou bien un problème d'ordonnancement ne permettant pas de respecter les contraintes temporelles) et les fautes d'implémentation (Exemples : mauvaise variable ou opérateur utilisés, erreur de gestion mémoire, etc.),
  - ◇ fautes physiques : (Exemple : capteur défaillant, connectique oxydée),

◇ fautes d'interaction : (Exemples : conditions météo imprévues, interaction incorrecte avec l'utilisateur, etc.).

Une faute non activée est présente dans le système mais n'a pas de conséquences sur le système tant qu'elle ne devient pas active (exemple : Un système d'alarme défaillant peut avoir des fautes inactives tant qu'il n'est pas testé ou sollicité. Il n'y a pas de conséquences tant qu'un évènement ou une autre défaillance ne nécessite pas de déclencher cette alarme). Une faute activée produit une erreur, qui peut se propager dans un composant ou d'un composant à un autre jusqu'à provoquer une défaillance. Ce processus de propagation est représenté sur la figure 1.2.

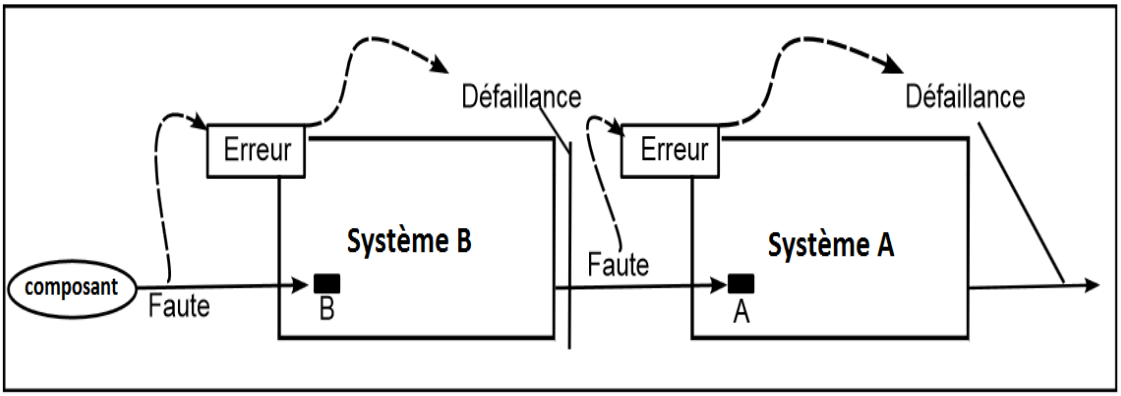

Figure 1.2 – Propagation d'erreurs dans un système de systèmes

Sur la figure 1.2, l'entrée du système A est la sortie du système B. Le service correct de A dépend ainsi du service correct de B. Pour le système complet, la défaillance de B est considérée comme une erreur interne, qui provoque éventuellement la défaillance du système.

1.1.1.3 Les moyens

Les moyens de la sûreté de fonctionnement sont les méthodes et techniques qui cherchent à rendre un système capable d'accomplir correctement sa mission, et à donner confiance dans cette aptitude. Afin d'aboutir à un système aussi sûr que possible, on peut avoir recours à une combinaison de ses moyens [Avizienis et al., 2004] qui sont :

- *prévention des fautes* : méthodes qui empêchent l'occurrence ou l'introduction de fautes. Elles peuvent être réalisées par des méthodes de spécification et de développement relevant de l'ingénierie des systèmes,
- *élimination des fautes* : méthodes qui réduisent le nombre et la sévérité des fautes dans le système. Elles peuvent être réalisées pendant la phase de

développement d'un système par vérification et correction, ou pendant sa phase opérationnelle par maintenance préventive et/ou mise à jour,

- *tolérance aux fautes* : méthodes qui permettent à un système de remplir correctement sa fonction en dépit des fautes. Ces méthodes sont principalement atteintes par la détection d'erreurs et le rétablissement du système. Elles nécessitent la redondance de certains composants du système,

- *prévision des fautes* : méthodes qui estiment la présence, les taux d'occurrence, et les possibles conséquences des fautes. Elles sont mises en œuvre par l'évaluation du comportement du système par rapport à l'occurrence des fautes et à leur activation.

Ces méthodes peuvent être classées en deux approches complémentaires [Lussier, 2007] pour concevoir un système sûr de fonctionnement :

- Les méthodes d'évitement des fautes regroupent la prévention et l'élimination des fautes et visent à réaliser des systèmes exempts de fautes,

- Les méthodes d'acceptation des fautes regroupent la tolérance et la prévision des fautes : elles partent du principe qu'il y a toujours des fautes que nous ne pouvons pas éviter, telles que certaines fautes matérielles, des fautes de développement résiduelles ou des fautes d'interactions, etc. Nous essayons alors d'évaluer leurs impacts sur le système et de réduire la sévérité des défaillances qu'elles peuvent causer (si possible jusqu'à la suppression des défaillances).

Dans ce manuscrit, nous nous intéressons en particulier à la tolérance aux fautes (mécanisme Safety-Bag) et à la prévision des fautes (analyses de sécurité).

1.1.2 La tolérance aux fautes

Dans cette section, nous abordons les notions et les méthodes de la tolérance aux fautes, qui est l'un de nos objectifs. Nous présentons ainsi les principes de base et le problème de la validation des mécanismes de la tolérance aux fautes.

1.1.2.1 Principe de la tolérance aux fautes

La tolérance aux fautes opère généralement dans la phase opérationnelle du système pour que ce dernier reste sûr ou/et fonctionnel en dépit de la présence des fautes. Sa mise en œuvre se fait généralement en deux étapes majeures la *détection d'erreurs* et le *rétablissement du système* :

- La détection d'erreurs : elle provoque la levée d'un signal ou d'un message d'erreur à l'intérieur du système. Les formes les plus courantes de détection d'erreurs sont : la *duplication et comparaison*, le *contrôle temporel* et le *contrôle de vraisemblance* [Lussier, 2007].

  - ◇ La duplication et comparaison : est basée sur une comparaison des résultats fournis par au moins deux unités redondantes indépendantes vis-à-vis des fautes à tolérer et fournissant le même service.

    Typiquement, les fautes physiques internes sont détectables par une redondance des composants matériels tels que les capteurs (par exemple, l'odométrie, la centrale inertielle et le GPS permettent d'avoir des sources redondantes et indépendantes pour la vitesse d'un véhicule). Par contre, les fautes logicielles et de conception sont détectables par la diversification des algorithmes (par exemple, les algorithmes de fusion de données).

  - ◇ Le contrôle temporel : détecte des erreurs temporelles dans le système, en vérifiant par exemple que son temps de réponse ne dépasse pas une valeur-seuil, ou surveille périodiquement l'activité des capteurs en utilisant un *chien de garde* ou encore en contrôlant l'horodatage des valeurs critiques.

  - ◇ Le contrôle de vraisemblance : détecte des erreurs en valeurs aberrantes pour le système. Typiquement, vérifier si la sortie appartient à une fourchette de valeurs possibles pour un capteur est un exemple de contrôle de vraisemblance.

- Le rétablissement (ou *recouvrement* du système) : il est généralement déclenché par un signal ou par un message. Il transforme l'état erroné détecté en un état jugé exempt d'erreurs et de fautes. Il peut être effectué par deux méthodes complémentaires : le *traitement d'erreurs* et le *traitement de fautes* [Lussier, 2007].

  - ◇ Le traitement d'erreurs : vise à corriger l'état erroné du système en lui substituant un état exempt d'erreurs. Cette substitution peut elle-même prendre trois formes :

    - ✓ La reprise : ramène le système dans un état correct survenu avant l'occurrence de l'erreur. Cela nécessite de capturer et de sauvegarder périodiquement l'état du système. Plusieurs difficultés sont liées à la reprise telle que la taille des sauvegardes et le sur-coût temporel nécessaire à leur établissement,
    - ✓ La poursuite : cherche un nouvel état à partir duquel le système peut continuer à fonctionner de façon acceptable, éventuellement en mode

dégradé (par exemple, si les performances du système de vision sont insuffisantes de fait de l'environnement ou du capteur lui-même, on peut imposer une faible vitesse). Une forme limite de cette technique est la mise du système dans un état sûr, par exemple en redonnant la main au conducteur ou en arrêtant le système (dans le cas d'un véhicule automobile, cet arrêt doit se faire dans un emplacement sûr tel que la bande d'arrêt d'urgence),

- ✓ La compensation d'erreurs : consiste à utiliser des redondances présentes dans le système pour fournir un service correct malgré les erreurs en transformant l'état erroné en un état correct.

◇ Le traitement de fautes : vise à empêcher la faute à l'origine de l'erreur d'être à nouveau activée. Cette méthode est composée de quatre phases successives :

- ✓ Le diagnostic : identifie et localise la faute activée responsable de l'état erroné du système,
- ✓ L'isolement : rend la faute dormante, c'est-à-dire, exclut la participation du composant erroné de la délivrance du service, par moyen physique ou logiciel,
- ✓ La reconfiguration : compense l'isolement du composant défaillant, soit en basculant sur des composants redondants, soit en réassignant ses tâches à d'autres composants,
- ✓ La ré-initialisation : vérifie et met à jour la nouvelle configuration du système.

Afin de mettre en œuvre la tolérance aux fautes, le principe de base consiste à utiliser la redondance. Dans un véhicule autonome, l'utilisation de plusieurs capteurs redondants de vitesse, par exemple, permet de détecter des incohérences de données. Cependant, les redondances matérielles et logicielles peuvent engendrer un surcoût financier ainsi qu'un temps de développement important.

1.1.2.2 Validation des mécanismes de tolérance aux fautes

Il est primordial de s'assurer que l'introduction des mécanismes de tolérance aux fautes ne provoquera pas de nouvelles fautes dans le système. Pour ce faire, deux approches complémentaires peuvent être réalisées afin d'évaluer et de valider ces mécanismes de tolérance aux fautes. La première est *la vérification formelle* et la deuxième est *l'injection de fautes*.

- La vérification formelle : permet de valider le comportement d'un système en présence de fautes, à condition de disposer de modèles formels qui décrivent les fautes considérées et leurs conséquences sur le comportement du système. Elle consiste à utiliser des techniques formelles telles que l'analyse statique, la preuve mathématique ou l'analyse de comportement.

- L'injection de fautes : permet d'évaluer les mécanismes de la tolérance aux fautes. [Arlat et al., 1993] décrivent une méthode d'évaluation basée sur l'injection de fautes. Leur étude consiste à réaliser des expériences surveillées par l'introduction volontaire des fautes dans un système pour tester et évaluer son comportement. L'objectif est de comparer le comportement nominal du système (un scénario de conduite normal d'un véhicule autonome dans notre cas sans injection des fautes), avec son comportement en introduisant des fautes injectées pendant l'exécution en ligne (par exemple, en bloquant l'application de contrôle-commande ou en dépassant la vitesse limite en ville) afin de révéler par la suite les déficiences éventuelles des mécanismes de tolérance aux fautes et les corriger.

1.2 Présentation générale des véhicules autonomes

Le développement d'ADAS (Aides à la conduite ou *Advanced Driver Assistance Systems*) automobile, de plus en plus sophistiqués (ABS, ESP, détection de piétons, systèmes d'aide au freinage et au maintien de trajectoire, etc.) semble mener naturellement à des véhicules autonomes. Cependant, passer d'une conduite humaine assistée à une conduite autonome est plus qu'une simple évolution aussi bien d'un point de vue technique et technologique que d'un point de vue social et juridique. Nous préciserons ce qu'implique l'autonomie pour un véhicule automobile, avant de présenter des architectures adaptées, ce qui nous amènera à poser la problématique de la sûreté de fonctionnement de tels systèmes.

1.2.1 Définitions et caractéristiques des véhicules autonomes

Cette section s'attache à présenter la notion d'autonomie et à définir les différents niveaux d'autonomie des véhicules automobiles autonomes.

1.2.1.1 Notion d'autonomie

Nous trouvons dans la littérature de nombreuses définitions de l'autonomie. Nous avons choisi deux définitions, les plus proches de notre domaine d'application :

E.Baudin et.al dans [Baudin et al., 2007] ont définit un système autonome comme suit : *Un système autonome est capable de raisonner et de prendre des décisions afin d'atteindre des objectifs en se basant sur ses connaissances et sa perception de l'environnement fluctuant dans lequel il évolue.* H.Huang et.al dans [Huang et al., 2005] ont définit à leur tour l'autonomie de la façon suivante : *L'autonomie est la capacité d'un système à percevoir, analyser, communiquer, planifier, établir des décisions et agir afin d'atteindre des objectifs assignés par un opérateur humain ; cette autonomie est mesurée par les aptitudes du système selon divers facteurs incluant la complexité de la mission, la difficulté de l'environnement et les interactions désirées ou non désirées avec différentes catégories d'êtres humains (opérateurs, co-équipiers, passants).*

Ces deux définitions mettent en avant les notions de perception, de prise de décision, et d'action dans un environnement dynamique.

Les mécanismes décisionnels peuvent être caractérisés par leurs capacités à enrichir l'autonomie du système. On peut distinguer cinq fonctions principales liées à l'autonomie :

- La planification : elle a pour but de choisir et d'organiser les tâches à entreprendre, en fonction de leur résultat prévu, afin d'atteindre un ou plusieurs objectifs.

- Le diagnostic : il sert à identifier un état erroné du système, généralement après une détection d'erreurs.

- Le contrôle d'exécution coordonne et supervise l'exécution de séquences de tâches, en décomposant chaque tâche en un ensemble de tâches plus simples qui seront mises en œuvre par une couche logicielle de niveau inférieur.

- La reconnaissance de situations a pour but d'observer un historique d'évènements, et d'en tirer des conclusions sur l'état actuel du système et de son environnement, et éventuellement sur les intentions des agents qui en font partie.

- L'apprentissage (hors ligne) : il permet d'améliorer les capacités d'un système en utilisant des informations liées à de précédentes exécutions. L'apprentissage est une fonction quelque peu à part, puisqu'elle est principalement utilisée pour développer les modèles, qui sont ensuite appliqués aux fonctions précédentes.

Un système autonome est capable de prendre des décisions grâce à des mécanismes développés dans le domaine de l'intelligence artificielle, et capable de prendre en

compte les incertitudes et l'évolution de l'environnement dans lequel le système évolue [Lussier, 2007].

Dans notre cas, les véhicules autonomes ont une autonomie opérationnelle puisque le pilote est automatique et les mécanismes d'asservissement sont bouclés pour contrôler la trajectoire. Ils ont également une autonomie décisionnelle puisque ces véhicules autonomes sont capables de percevoir et de distinguer les autres usagers de la route, parfois dans des conditions climatiques difficiles avec peu de visibilité et de décider de l'évitement, de l'arrêt, et éventuellement du changement d'itinéraire.

Donc dans une voiture autonome, en plus du contrôle par des techniques d'automatique (loi de commande par exemple) du moteur, du freinage, et de la direction pour suivre la trajectoire, on doit reconnaitre la situation de la conduite respectait le code de la route et les contraintes de l'environnement (profil de la route, autres véhicules, piétons, etc.). Ce sont ces fonctions de haut niveau qui caractérisent l'autonomie du véhicule. Ces capacités vont être réalisées en intégrant des techniques d'intelligence artificielle notamment des mécanismes de prise de décision, d'apprentissage et de planification.

1.2.1.2 Niveaux d'autonomie des véhicules autonomes

La SAE (Society of Autonomous Engineers)[Kyriakidis et al., 2015] puis l'OICA [OICA, 2014] (Organisation Internationale des Constructeurs Automobile) ont défini une classification des niveaux d'autonomie des véhicules autonomes. Les deux classifications sont similaires dans le principe, à quelques nuances près. Celle de l'OICA comprend 6 niveaux.

- Le niveau 0 (ou *no automation*) : correspond aux véhicules classiques sans fonctions autonomes.

- Le niveau 1 (ou *driver assistance*) : correspond aux véhicules équipés de fonctions d'assistance à la conduite telles que la régulation de vitesse.

- Le niveau 2 (ou *partial automation*) : correspond aux véhicules disposant d'une autonomie limitée à quelques situations particulières telles que le *park assist*, qui peuvent effectuer des créneaux automatiques. Le conducteur reste entièrement responsable de la conduite du véhicule.

- Le niveau 3 (ou *conditional automation*) : correspond aux véhicules capables de conduire de manière autonome sous la supervision d'un conducteur humain. Les situations de conduite gérées peuvent être peu nombreuses et les performances peuvent être limitées. Le véhicule doit être capable de

reconnaître ses limites et alors informer le conducteur en lui laissant un délai raisonnable pour reprendre le contrôle.

- Le niveau 4 (ou *high automation*) : diffère du niveau 3 par le fait que le système autonome doit assurer entièrement la conduite dans les cas d'utilisation prévus y compris si le conducteur ne réagit pas lors d'une demande de retour en conduite manuelle.

- Le niveau 5 (ou *full automation*) : correspond à des véhicules totalement autonomes dans toutes les circonstances. Ils ne nécessitent pas une supervision humaine.

Notre laboratoire dispose de trois véhicules (deux sont de type Renault Zoé et une de type Fluence) pour mener ses développements de véhicules autonomes et les expérimentations associées. Ces véhicules autonomes expérimentaux se placent dans les niveaux 3 à 4 de cette classification.

1.2.2 Architecture des véhicules autonomes

Après avoir introduit dans la section précédente la notion d'autonomie et les différents niveaux d'autonomie des véhicules autonomes, nous présentons ici les architectures et quelques exemples des véhicules autonomes réels.

1.2.2.1 Architecture des véhicules autonomes

D'aprés [Campbell et al., 2010], l'architecture de la plupart des véhicules autonomes se décompose en quatre sous-systèmes de base : *la détection*, *la perception*, *la planification*, et *le contrôle* (voir figure 1.3).

- La détection : est chargée de mesurer des données brutes afin d'orienter le véhicule (cela inclut le GPS, l'IMU (l'Unité de Mesure Inertielle, ou *Inertial Measurment Unit*), et les mesures d'odométrie. La détection est aussi chargée de détecter les éléments de l'environnement urbain statique et dynamique (ces mesures comprennent des télémètres laser, des radars et des caméras).

- La perception : crée des informations utilisables relatives au véhicule et à son environnement. L'estimation d'état du véhicule comprend généralement une mesure de « *pose* » (position inertielle, vitesse, attitude, fréquences) ainsi qu'une information relative aux cartes routières électroniques du système (par exemple, l'emplacement du véhicule dans une voie ou une carte). La perception utilise généralement des mesures de vision ou de laser afin d'aider à produire

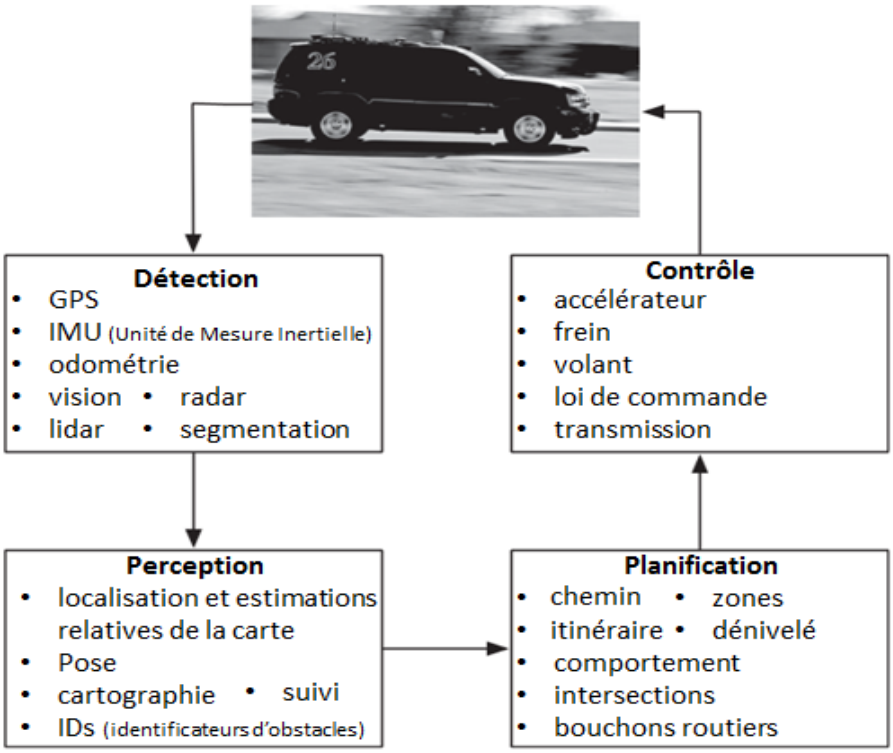

Figure 1.3 – Architecture de systèmes de haut niveau pour la conduite urbaine

des estimations relatives à la carte. L'estimation de l'état de l'environnement peut être réalisée de plusieurs façons, principalement en fonction de variations dans les capteurs, le calcul et les ressources.

- La planification : comprend les planificateurs de trajectoires, les planificateurs de routes (cartes) et les planificateurs de comportement.

- Le contrôle : comprend les actionneurs réels et commande la conduite de la voiture.

Dans notre manuscrit, nous allons considérer un modèle d'autonomie qui suit l'architecture hiérarchisée [Alami et al., 1998] [Lussier, 2007], ainsi que des exemples de mécanismes décisionnels.

L'architecture hiérarchisée (aussi appelée trois niveaux) a été proposée pour structurer le contrôle d'un système autonome en couches de niveaux d'abstraction différents (voir la figure 1.4). Elle comprend typiquement trois niveaux : une couche réactive, une couche de supervision, et une couche décisionnelle pour effectuer des processus décisionnels déclaratifs.

- Une couche réactive (ou *couche fonctionnelle*) : elle est chargée de l'exécution des tâches élémentaires du plan défini par les couches supérieures telles que

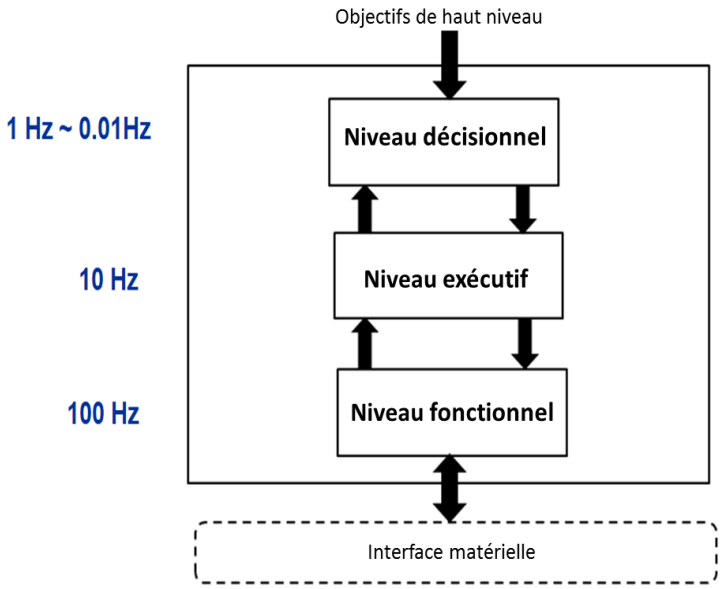

Figure 1.4 – Architecture hiérarchisée

le suivi de la trajectoire. Elle effectue un ensemble d'actions de bas niveau et contrôle les actionneurs et les capteurs. Son cycle d'exécution est généralement autour de 100 voire 1000 Hz.

- Une couche exécutive (ou *couche de fusion*) : c'est la couche intermédiaire qui découpe les activités de haut niveau du plan de la couche décisionnelle, et peut gérer des fonctionnalités telles que la reconnaissance de l'environnement ou le calcul de l'espace navigable. Généralement, la génération des trajectoires cinématiques du véhicule est réalisée à ce niveau, avec des fréquences d'exécution de l'ordre de 10 Hz.

- Une couche décisionnelle : c'est une couche de haut niveau d'abstraction, dont le cycle est de l'ordre de la seconde ou de la minute, et qui gère la planification de l'itinéraire et le changement de modes de conduite. Basée sur des mécanismes d'intelligence artificielle, elle comporte généralement un modèle, représentant un ensemble d'actions et de contraintes qui décrivent la façon dont le système peut atteindre ses objectifs. La recherche d'une solution peut être effectuée par l'utilisation d'heuristiques.

Nous constatons qu'il existe des points en commun entre ces deux architectures. En effet, la détection et le contrôle selon l'architecture de [Campbell et al., 2010] se trouvent dans la couche fonctionnelle et l'architecture hiérarchisée[Alami et al., 1998].

La perception peut se retrouver dans la couche fonctionnelle, suivant la vitesse de traitement de ses algorithmes. Enfin, la planification se trouve typiquement dans

la couche décisionnelle (planification de trajectoire de haut niveau). Cependant, par exemple dans le cas des véhicules autonomes, certaines tâches de planification telle que la planification de la trajectoire peuvent faire partie de la couche exécutive. L'architecture Campbell nous parait décrire d'abord la circulation des informations des capteurs jusqu'aux actionneurs à travers la détection, la perception, la planification, et le contrôle tandis que l'architecture LAAS est une architecture en couche d'abstraction.

1.2.2.2 Exemples des véhicules autonomes réels

Dans la dernière décennie, les recherches sur les véhicules automobiles autonomes ont progressé significativement. En 2003, l'Agence des Projets de Recherche Avancée de la Défense DARPA (ou *Defence Research Projects Agency*) a annoncé le premier Grand Challenge, qui vise à developper des véhicules autonomes capables de naviguer à grande vitesse sur les routes et sur les pistes du désert. Dans le DARPA Grand Challenge organisé en Mars 2004 [Urmson et al., 2004], quinze véhicules terrestres autonomes ont eu pour tâche de parcourir 228 kilomètres dans le désert Mojave, de Barstow en Californie jusqu'à Primm Nevada. Les deux compétiteurs les plus performants n'ont pas dépassé le douzième kilomètre.

En 2005, un second Grand Challenge s'est tenu à Primm, Nevada en mettant en jeu un prix de 2 millions de dollars. Le gagnant a été le véhicule de l'équipe d'ingénieurs de l'université Stanford en Californie, *Stanley*, un *Volkswagen Touareg* amélioré. Ce véhicule a parcouru avec succès 212 kilomètres du désert à une vitesse moyenne d'environ 30 km/h en seulement 6 heures, 53 minutes et 58 secondes. Le principal défi technologique dans le développement de Stanley était de construire un système très fiable, capable de rouler à des vitesses relativement élevées dans des environnements hors route divers et non structurés sans intervention manuelle, et de faire tout cela avec une grande précision. Le système logiciel de ce robot repose principalement sur des technologies d'intelligence artificielle, telles que l'apprentissage et le raisonnement probabiliste [Thrun et al., 2006].

[Urmson et al., 2008] décrit le gagnant de la troisième édition annoncée par DARPA Urbain Challenge 2007 : *Boss*, un *Chevrolet Chevy Tahoe 4x4*, est un véhicule autonome capable de rouler en toute sécurité dans un environnement urbain à des vitesses atteignant jusqu'à 48 km/h. Il utilise des capteurs à bord (système de positionnement global, lasers, radars et caméras) pour suivre les autres véhicules, détecter les obstacles statiques et se localiser par rapport à un modèle de route.

[Naufal et al., 2017] citent plusieurs entreprises, notamment, Google, Volvo, Tesla, Nissan, Audi, Toyota, les camions Mercedes, les motocycles Yamaha, et

Scania qui développent aujourd'hui des véhicules autonomes. Renault et PSA se sont également lancés en France dans le marché des véhicules autonomes. Cependant, les caractéristiques d'autonomie visées diffèrent sensiblement selon les projets. Cette diversité a nécessité une classification des niveaux d'autonomie.

1.2.3 Sûreté de fonctionnement des véhicules autonomes

Dans cette section, nous présentons les menaces contre la sûreté de fonctionnement sur l'architecture générale d'un système autonome, puis les vulnérabilités spécifiques pour les véhicules autonomes.

1.2.3.1 Menaces contre la sûreté de fonctionnement des véhicules autonomes

L'autonomie soulève des problématiques de sûreté de fonctionnement supplémentaires dues à l'environnement dynamique et ouvert dans lequel les véhicules autonomes opèrent. Dans cet environnement ouvert, qui peut être soumis à des menaces exogènes, en particulier les aléas de l'environnement (tels que la présence des piétons, des autres véhicules et des obstacles fixes et mobiles, etc.) ainsi que les conditions météorologiques environnementales (telles que la pluie, la neige, une route glissante, etc.), le système doit être capable d'assurer de manière autonome sa mise en état sûr. L'autonomie requiert aussi des logiciels de contrôle complexes (typiquement d'intelligence artificielle) pour lesquels il est difficile d'appliquer les techniques classiques utilisées pour valider les systèmes critiques. Par exemple, les techniques de *model-checking* sont difficilement praticables vu l'explosion du nombre d'états. Cette dernière est due à l'énorme diversité des situations puisque le système se trouve dans un environnement ouvert. Pour ces mêmes raisons, il est difficile d'évaluer la complétude de l'ensemble des tests effectués [Koopman and Wagner, 2016]. De plus, l'utilisation des heuristiques afin d'accélérer la recherche de solution induit une complexité supplémentaire ; le système décisionnel devient difficilement prédictible, ce qui rend le test et la vérification encore plus difficiles [Mekki Mokhtar, 2012]. Suivant Bandin et al. dans [Baudin et al., 2007] et en s'inspirant des travaux présentés dans [Powell and Thévenod-Fosse, 2002] et [Avizienis et al., 2004], on peut dire que les menaces contre la sûreté de fonctionnement des systèmes autonomes peuvent être classées comme indiqué dans la figure 1.5 :

- Les menaces endogènes, qui proviennent du système autonome lui-même, comprennent les fautes de développement (introduites lors de la modélisation du système ou lors du développement), et les fautes physiques opérationnelles

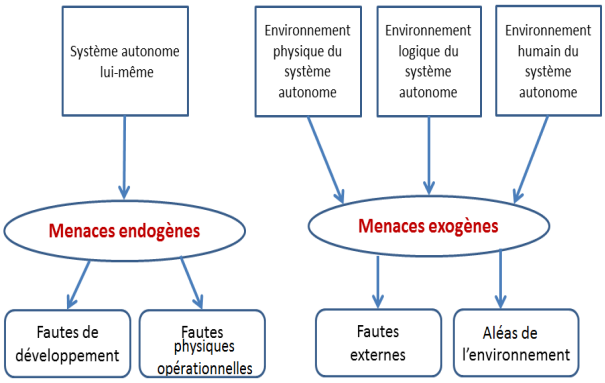

Figure 1.5 – Ensemble des menaces contre la sûreté de fonctionnement des systèmes

(se produisant pendant la phase opérationnelle et affectant les ressources physiques : capteurs, processeurs, sources d'énergie).

- Les menaces exogènes, qui proviennent de l'environnement physique, logique et humain du système autonome, comprennent les fautes externes (par exemple les interférences, les attaques malveillantes et des fautes de coopération inter-systèmes, dont les fautes d'interaction avec l'humain), la mauvaise perception de l'environnement (manque d'observabilité, capteurs imparfaits, etc.) ainsi que les aléas de l'environnement (événements incontrôlables, circonstances imprévues, etc.).

[Guiochet and Powell, 2005] citent quelques classes de menaces contre la sûreté de fonctionnement émergeant du contexte d'utilisation des robots, mais aussi de l'architecture LAAS en deux couches (fonctionnelle et décisionnelle). Parmi ces menaces, on trouve :

- Des menaces liées à la complexité interne du système telles que la complexité des mécanismes décisionnels et la complexité de la structure de l'application.
- Des menaces dues à la complexité externe du système telles que l'organisation de développement avec des acteurs multiples, et l'hétérogénéité des composants matériels, des modèles, des langages, etc.
- Des menaces liées à l'environnement ouvert.

1.2.3.2 État de l'art de la sûreté de fonctionnement pour les véhicules automobiles autonomes

Les véhicules automobiles autonomes doivent encore améliorer leur sûreté de fonctionnement avant d'être utilisés pour la conduite quotidienne. En particulier,

leur complexité, le recours à des mécanismes de l'intelligence artificielle, pourtant indispensables pour évoluer dans un environnement non structuré, ouvert et partiellement inconnu, soulèvent le problème de leur sécurité-innocuité.

Selon [Naufal et al., 2017], la sécurité est bien un obstacle technologique à l'industrialisation des véhicules automobiles autonomes. Pour répondre à ce problème, l'agence américaine NHTSA [NHTSA, 2016], chargée de la sécurité routière, exige aujourd'hui un processus qui devrait comporter une analyse de risques et une évaluation des exigences de sécurité. Ce processus devrait contenir, en particulier, des redondances de conception et des stratégies de sécurité capables de ramener le véhicule autonome dans un état sûr de fonctionnement en cas de situation dangereuse (une défaillance matérielle ou logicielle par exemple). Ce système de sécurité *Fall back* doit également faciliter la transition de la conduite autonome à la conduite manuelle lorsque le système ne peut pas fonctionner correctement.

Dans [Lussier et al., 2004], les auteurs ont fait la distinction entre la tolérance aux fautes et la robustesse. La robustesse se distingue de la tolérance aux fautes par les situations adverses que chacune cherche à tolérer.

- Les techniques de la tolérance aux fautes : visent à éviter les défaillances du système en présence de fautes affectant les ressources matérielles et logicielles du système (telles que les défaillances des capteurs, les fautes de conception du logiciel, etc.).

- Les techniques de robustesse : visent à éviter les défaillances du système en présence de fautes externes, d'incertitudes environnementales et des contingences.

En s'appuyant sur [Ingrand and Ghallab, 2017], J. Guiochet dans [Guiochet, 2015] indique que la tolérance aux fautes est rarement explicitement mentionnée dans la littérature sur les systèmes robotiques autonomes, où le concept de surveillance est préférée lorsqu'on se réfère à la planification. Bien que certaines techniques de détection d'erreurs (telles que le contrôle temporel par un *chien de garde*, la duplication et comparaison par redondance, et la surveillance basée sur un modèle de diagnostic) ou de rétablissement du système (confinement d'erreur, positionnement dan un état sûr, et reconfiguration matérielle et logicielle) sont assez communs, leur utilisation est loin d'être systématique, en partie parce que la plupart des systèmes autonomes sont encore des plateformes de recherche basées sur le développement de fonctions autonomes (et non pas sur leur tolérance aux fautes). Un autre point intéressant est que la tolérance aux fautes augmente significativement le coût de développement en termes d'espace physique ou d'autonomie de puissance, qui sont tous des enjeux critiques pour les systèmes embarqués, et a fortiori pour

les systèmes autonomes. Les techniques de la sûreté de fonctionnement utilisables pour les véhicules autonomes ne sont pas limitées seulement à la tolérance aux fautes et à la robustesse. [Guiochet, 2015] a cité quelques techniques d'élimination de fautes. Nous trouvons par exemple la vérification dynamique (notamment l'exécution de tests et la détection de fautes via un moniteur d'exécution ou une analyse de Log) et la vérification statique (notamment l'analyse statique, le model checking et la preuve de théorème). Néanmoins, comme l'ont mentionné Pecheur dans [Pecheur, 2000], Tiwara et Sinha dans [Tiwari and Sinha, 2003] et Menzie et Pecheur dans [Menzies and Pecheur, 2005], les problèmes classiques rencontrés par la vérification dans les systèmes de contrôle sont exacerbés dans le cas des systèmes autonomes.

Dans [Xu and Yuan, 2016], Xu et Yuan proposent un système de sécurité active pour les véhicules autonomes, qui a été identifié comme une technique efficace pour éviter les accidents. Il se compose d'un planificateur de mouvement et d'un contrôleur basé sur un contrôle robuste plutôt que sur la capacité à tolérer les fautes. Une autre approche robuste pour l'évaluation de la sécurité des trajectoires planifiées des véhicules autonomes est présentée dans [Althoff et al., 2009]. Cette approche tient compte de la dynamique du véhicule, de l'interaction entre les usagers du trafic et les changements de voie dans un trafic à multiples voies. Cependant, à notre connaissance, la tolérance aux fautes est rarement mentionnée dans les voitures autonomes.

Nous pouvons conclure ainsi que les travaux de recherche jusqu'à présent portent plus sur la robustesse que sur la tolérance aux fautes.

Le moyen de sûreté de fonctionnement le plus étudié, à notre connaissance, pour un véhicule autonome, est l'élimination des fautes par des tests. Selon la norme IEEE 829-2008 [IEEE, 2008], le test est défini comme étant : *une activité dans laquelle un système ou un composant est exécuté sous des conditions spécifiques ; les résultats sont observés ou enregistrés ; et l'évaluation de certains aspects du système ou du composant est effectuée.* Un test suivant [Mekki Mokhtar, 2012] est un ensemble de *cas* à tester. La même norme précédente a défini un cas à tester comme étant : des données d'entrées, des conditions d'exécution et les résultats attendues pour un objectif donné. Le test peut être classé selon le niveau de cycle de développement du logiciel dans lequel il est effectué : test composant (unitaire), test d'intégration, test système (ou test fonctionnel) et test d'acceptation.

Nous trouvons dans [Mekki Mokhtar, 2012], une autre classification du test selon sa caractéristique.

- test de performance : validation que les performances spécifiées du système sont respectées,

- test fonctionnel : vérification que les fonctionnalités répondent aux attentes,
- test de vulnérabilité : vérification de la sécurité du logiciel,
- test de robustesse : validation de l'aptitude du système à délivrer un service correct dans la durée, ou bien en présence d'entrées invalides ou de conditions environnementales stressantes. Ce type de test se trouve en détails dans [Chu, 2011].

Dans le cas des systèmes autonomes, des tests en simulation intensifs ont été réalisés en place sur l'architecture hiérarchisée développée par la NASA dans le cadre du projet DS1 (*Deep Space One*) [Bernard et al., 2000]. Six bancs de test ont été mis en place tout au long du développement, intégrant 600 tests. Bernard et al. dans [Bernard et al., 2000] soulignent la pertinence des tests intensifs, mais reconnaissent des difficultés particulières concernant les systèmes autonomes, notamment le problème d'oracle (c'est-à-dire, comment décider si un test a échoué ou est passé avec succès). Dans Micskei et al. [Micskei et al., 2012], une structure a été développée pour générer des cas de test de robustesse pour les systèmes autonomes mobiles. Il est basé sur un modèle de tâches système (représentées par des diagrammes de séquences UML), et sur une modélisation de l'environnement (provenant de l'ontologie, spécifiant tous les objets possibles de l'environnement et leurs propriétés). Toujours dans le domaine de la génération de cas du test, [Zou et al., 2014] présentent une approche basée sur la mutation génétique. L'objectif est de générer des cas pour tester l'évitement des collisions entre deux drones. Considérant que l'oracle est basé sur l'estimation d'une distance entre drones équivalente à une collision, la fonction de la remise en forme est facilement implémentée. Dans le cadre des véhicules autonomes, [Meltz and Guterman, 2014] présente la limitation de la fiabilité existante et les outils d'ingénierie de sécurité pour les véhicules terrestres autonomes sans pilote et propose une nouvelle méthodologie basée sur des tests statistiques dans un environnement simulé.

Cependant, [Koopman and Wagner, 2016] postulent que les tests seuls sont impraticables pour assurer une sûreté de fonctionnement suffisante dans les véhicules autonomes. En effet, les véhicules autonomes ne font que rendre ce problème plus difficile en automatisant les réponses aux situations environnementales très complexes et en introduisant des technologies telles que l'apprentissage, qui sont difficiles et coûteuses à tester. De plus, étant donné qu'une grande partie de la capacité d'autonomie doit avoir un niveau d'intégrité de la sécurité-innocuité automobile (ASIL) élevé en raison du manque de la surveillance de la conduite humaine, il semble difficile de faire assez de tests ordinaires de système pour obtenir

même un niveau d'assurance raisonnable. Une approche prometteuse pour aider à valider les caractéristiques d'autonomie consiste à effectuer une injection de fautes appliquée à plusieurs niveaux d'abstraction du composant.

1.3 Analyse de risques

Dans nos travaux, nous chercherons à identifier les différents modes de défaillances possibles induits par l'utilisation du système afin d'éliciter, exprimer et vérifier un ensemble de propriétés de sécurité permettant de les éviter. Il existe aujourd'hui de nombreux travaux de recherche pour cette identification, les méthodes d'analyse de risques apparaissant comme un moyen permettant d'identifier les défaillances et pour certaines d'évaluer leur gravité et de déterminer leur taux d'occurrence [Mekki Mokhtar, 2012]. Afin de mettre en application les activités de la gestion du risque, plusieurs techniques sont utilisées. En particulier, l'analyse du risque est en partie traitée par l'utilisation de techniques comme l'AMDEC, l'HazOp, l'APR, les arbres d'évènements ou les arbres de défaillances. Ces techniques largement utilisées dans d'autres domaines sont aujourd'hui appliquées à des dispositifs comme les véhicules autonomes. Nous présentons dans cette section des notions sur le risque, puis différentes méthodes d'analyse.

1.3.1 Le risque

Cette partie présente une sélection de travaux et de concepts autour de la notion de risque. Nous présentons tout d'abord les notions de risque, puis de gestion des risques. Nous finissons cette partie par présenter un exemple d'une matrice de risque et par définir le niveau d'intégrité de la sécurité-innocuité automobile.

1.3.1.1 Notion de risque

La norme ISO Guide 51 [Aven, 2011] définit la notion de risque comme étant la combinaison de la probabilité d'occurrence d'un dommage et de sa gravité. La notion de risque est étroitement liée à la sécurité-innocuité, cette dernière étant définie par la norme comme étant l'absence de risques inacceptables. De ce fait, un processus de gestion de risque est défini comme permettant d'identifier les risques potentiels, d'évaluer l'acceptabilité de ces risques et de permettre de statuer sur la nécessité de mesures correctives.

1.3.1.2 Gestion du risque et ses étapes

La gestion du risque est un processus itératif qui doit s'effectuer tout au long du cycle de vie du système. Les étapes du processus de gestion du risque se composent en trois parties :

- L'analyse de risque : elle est l'utilisation systématique des informations disponibles afin d'identifier les dangers et d'estimer le risque [ISO/IEC-Guide51, 1990].
- L'évaluation du risque : elle se base sur l'étape précédente pour évaluer les risques et déterminer si le risque est tolérables.
- La réduction de risque : elle permet, dans le cas où le risque est considéré comme inacceptable, de déployer les mesures nécessaires à la réduction des risques.

Ce processus est principalement mis en place dans l'industrie pour les systèmes critiques en s'appuyant sur des normes. Dans le ferroviaire, les normes sont EN 50126, EN 50128 et EN 50129. Dans le secteur automobile, la norme centrale est la ISO 26262. Pour les machines industrielles, la norme de référence est la EN 62061.

1.3.1.3 L'acceptabilité des risques

Le processus de gestion de risque permet d'identifier les risques potentiels dans un système étudié, d'évaluer leur acceptabilité et de prendre des mesures nécessaires correctives afin de réduire ces risques. Nous présentons dans la partie suivante un exemple d'une matrice de risque et nous définissons par la suite le niveau d'intégrité de la sécurité-innocuité automobile.

1.3.1.3.1 Matrice de risque Nous rappelons tout d'abord que le risque est combiné de la probabilité d'occurrence d'un dommage et de sa gravité. Nous trouvons dans [ISO/IEC-Guide51, 1990] un exemple de matrice de risque composée de deux paramètres *Fréquence* et *Gravité*. Dans le même document, chaque paramètre est dérivé en 6 niveaux. Les niveaux de la fréquence d'occurrence sont : *fréquente*, *probable*, *occasionnelle*, *rare*, *invraisemblable* et *impossible*. Les niveaux de la gravité sont : *fatale*, *critique*, *sévère*, *sérieuse*, *modéré*, *mineure*, et *néant*. Nous obtenons alors 36 combinaisons possible de la criticité du dommage. Plus la fréquence d'occurrence ou la gravité est importante, plus le risque est élevé. Comme montre la figure 1.6, les différentes combinaisons sont par la suite catégorisées en 4 niveaux de

risque : Haute (*High, H*), Intermédiaire (*Intermediate, I*), Basse (*Low, L*) et Nulle (*None, N*).

Les niveaux de risques de type *High (H)* sont considérés comme inacceptables. Tous le reste seront considérés comme acceptables.

Fréquence d'occurrence	Gravité						
	6 Fatale	5 Critique	4 Sévère	3 Sérieuse	2 Modérée	1 Mineure	0 Néant
Fréquente	H	H	H	H	H	I	N
Probable	H	H	H	H	I	L	N
Occasionnelle	H	H	H	I	L	N	N
Rare	H	H	I	L	N	N	N
Invraisemblable	I	I	L	N	N	N	N
Impossible	L	L	N	N	N	N	N

Figure 1.6 – Exemple d'une matrice de risque

Dans le cas où le risque est considéré comme inacceptable, des mesures nécessaires à la réduction des risques doivent être effectuées. Selon le guide ISO/IEC 51, les mesures qui consistent à réduire la fréquence d'occurrence du dommage sont appelées mesures de *prévention*, alors que les mesures qui consistent à réduire la gravité du dommage sont appelées mesures de *protection*. La figure 1.6 reste un exemple typique d'une matrice de risque. Nous pouvons alors modifier et utiliser d'autres dérivés et d'autres niveaux dépendant de notre système et de nos besoins.

1.3.1.3.2 Niveau d'intégrité de la sécurité-innocuité automobile (*Automotive Safety Integrity Level, ASIL*)

L'ISO-26262 est la norme de sécurité-innocuité fonctionnelle pour l'industrie des véhicules routiers. Elle se concentre sur les exigences, les processus et les méthodes faisant face aux effets des défaillances systématiques et des défaillances non systématiques [Dhouibi et al., 2014]. Cette norme exige un effort important de la part des constructeurs automobiles, qui ont besoin d'une part de gérer leurs coûts et d'autre part de mettre en place la sécurité-innocuité et se conformer aux nouvelles méthodologies standardisées. L'ISO-26262 a adapté différents concepts tels que la *traçabilité* : c'est la capacité de suivre les exigences de sécurité-innocuité à partir des caractéristiques de sécurité au niveau du système jusqu'à la conception détaillée du logiciel et du matériel [Azevedo et al., 2013]. L'ISO-26262 utilise le concept d'un niveau d'intégrité de sécurité-innocuité (*Safety Integrity Level, SIL*) qui a été redéfini comme le niveau d'intégrité de la sécurité-innocuité automobile (*Automotive Safety Integrity Level,*

ASIL) par la suite. L'ASIL représente le degré de rigueur qui doit être appliqué dans le développement, l'implémentation et la vérification d'une exigence afin de minimiser le risque résiduel dans le produit final [Dhouibi et al., 2014]. Les niveaux d'intégrité de sécurité-innocuité sont définis et classés par criticité de la façon suivante :

- 4 ASIL : de A (exigences moins strictes) jusqu'à D (exigences les plus strictes)

  Exemple de niveau D : la perte de contrôle du véhicule peut causer des conséquences catastrophiques.

- La spécification QM (*Quality Management*), qui n'implique aucune exigence de sécurité particulière. Par exemple : La perte des alertes (perte de supervision) dans un système autonome ne peut pas mener à une situation catastrophique.

1.3.2 Méthodes d'analyse de risque

Les méthodes d'évaluation des risques et de la fiabilité développées au début des années 1960 ont pour origine les programmes aérospatiaux et de missiles des États-Unis. Au début du projet Apollo, la question a été posée concernant la faisabilité d'envoyer avec succès des astronautes sur la Lune et les ramener en toute sécurité. Un calcul des risques potentiels a été effectué par arbre de défaillances et la probabilité de succès de la mission s'est avérée extrêmement faible. Ce résultat a découragé la NASA de poursuivre ce type d'analyse. La NASA s'est contentée d'effectuer des analyses des Modes de Défaillances, leurs Effets et leur Criticité (AMDEC, ou FMEA en anglais) jusqu'à l'accident de la navette spatiale Challenger en 1986. Après cet accident, l'importance du l'Evaluation Probabiliste de Risques (EPR, ou PRA en anglais) incluant une analyse par Arbre de Défaillances (AdD, ou FTA en anglais) dans l'analyse des risques des systèmes a été reconnue [Stamatelatos et al., 2002]. L'industrie nucléaire a commencé l'utilisation de l'évaluation probabiliste des risques pour évaluer la sécurité suite à l'accident de Three Mile Island en 1979.

Dans cette section, nous allons nous focaliser sur les méthodes d'analyse de risques : l'APR, qui se concentre sur les évènements redoutés, l'AMDEC, qui considère tous les composants du système, les arbres de fautes, qui prennent en compte les défaillances multiples, et finalement l'HazOp, qui assure une analyse méthodique des fonctionnalités du système.

1.3.2.1 Analyse Préliminaire des risques (APR)

L'analyse Préliminaire des Risques (APR) a été élaborée par l'armée américaine dans les années soixante et est décrite dans la première version du standard MIL-STD-882 [MIL-STD-882c, 1993] en 1969, sous le nom de *Preliminary Hazard Analysis (PHA)*. Ensuite, elle a été utilisée par un certain nombre d'industries, notamment l'aéronautique, la chimie et le nucléaire. L'Union des Industries Chimiques (UIC) diffuse un guide méthodologique adapté à l'industrie chimique et recommande son utilisation en France depuis le début des années 1980 [Flaus, 2013]. C'est aujourd'hui la pierre angulaire du Système de Management de la Sécurité (SMS) dans de nombreuses industries [Mazouni et al., 2008]. D'après [Rausand, 2013], l'analyse préliminaire des risques consiste à identifier tous les risques potentiels et les évènements pouvant entraîner un accident, classer les évènements accidentels identifiés en fonction de leur sévérité et identifier les contrôles de dangers requis et les actions de suivi. Plusieurs variantes d'APR sont utilisées, et parfois sous des noms différents comme : classement rapide des risques (*Rapid Risk Ranking*) ou identification des risques (*HAZard IDentification, HAZID*).

L'APR est une méthode d'analyse de risques qui peut être soit inductive comme l'AMDEC et l'HazOp ou bien déductive comme l'AdF. A partir de chaque mode de défaillance, on cherche à identifier les conséquences possibles sur le système et son environnement. Jean-Marie Flaus a mentionné dans son livre [Flaus, 2013] les différentes étapes utilisées pour réaliser la méthode d'analyse de risques APR :

- préparer l'analyse : cette étape consiste à définir le contexte,
- décrire et modéliser le système,
- identifier les risques et les différents évènements redoutés,
- analyser les situations dangereuses,
- analyser les conséquences,
- rechercher les barrières existantes,
- évaluer la sévérité et la fréquence ou vraisemblance, de façon qualitative, ou semi-qualitative,
- proposer de nouvelles barrières,
- rédiger un rapport d'analyse.

1.3.2.2 Analyse des Modes de Défaillances, leurs Effets et leurs Criticité (AMDEC)

L'Analyse des Modes de Défaillances, de leurs Effets et leur Criticité (AMDEC) (ou *Failure Mode, Effects and Criticality Analysis*) a été utilisée par la NASA afin d'évaluer la sécurité de systèmes spatiaux. Elle a également été utilisée dans l'industrie automobile. Selon la norme IEC61508-7 [IEC61508-7, 2000], l'objectif de l'analyse de risque AMDEC est *d'établir un ordre de criticité d'une dégradation du système par le biais de défaillances uniques, afin de déterminer quels composants peuvent nécessiter une attention particulière et des mesures de surveillance nécessaires pendant la conception ou l'exploitation.* L'AMDEC est composée de deux analyses qui sont l'AMDE et l'Evaluation de la Criticité.

La méthode AMDE est une méthode ascendante (*bottom-up*) qui examine les possibles modes de défaillances des composants du système, afin de déterminer les effets de telles défaillances sur les équipements et les performances du système [Mekki Mokhtar, 2012]. Les composants considérés peuvent être des éléments simples ou des sous-systèmes, qui peuvent être abstraits dans le cadre de cette analyse.

L'Evaluation de la Criticité des modes de défaillance passe habituellement par la prise en considération croisée de la probabilité d'occurrence du mode, la possibilité de mettre en place une détection, et la gravité des conséquences.

A. Mekki Mokhtar dans sa thèse [Mekki Mokhtar, 2012] a suivi les étapes suivantes pour réaliser la méthode AMDEC :

- Définir et partitionner le système en systèmes/sous-systèmes, ou bien sous-ensembles/pièces élémentaires, avec une description fonctionnelle de ces sous systèmes.
- Développer pour chaque élément de base, une liste complète des modes de défaillance.
- Déterminer et enregistrer les effets de chaque défaillance dans la matrice AMDEC.
- Déterminer et enregistrer les causes possibles de chaque défaillance dans la matrice AMDEC.
- Assigner et enregistrer pour chaque mode de défaillance de chaque élément de base sa gravité.
- Analyser la capacité du système à détecter et à signaler pour chaque composant.

- Calculer la criticité du mode de défaillance pour chaque composant à partir de différents paramètres (Exemples : indice de fréquence, indice de gravité, indice de détection , durée de la mission, etc.).

- Mettre en évidence les éléments et les défaillances critiques selon le niveau de criticité, afin de réduire les risques.

Dans le cas de notre étude des véhicules autonomes, nous avons introduit des colonnes supplémentaires dans la matrice AMDEC pour distinguer les effets sur le composant de contrôle-commande (le sous-système particulièrement ciblé par nos travaux) et les effets finaux sur le comportement du véhicule. Nous avons ajouté également les étapes suivantes :

- Définir le taux de défaillance λ (ou *failure probability per hour*) pour chaque élément défaillant.

- Déterminer les moyens de détection (pilote, application de contrôle-commande, ou Safety-Bag, etc.) et le temps de détection ainsi que les moyens d'action (le pilote par exemple, ou le Bouton d'Arrêt de Process (BAP)) et le temps d'action pour chaque élément.

- Identifier les exigences de sécurité associées à chaque type de défaillance de chaque élément afin d'extraire par la suite les contraintes de sécurité.

1.3.2.3 Arbre de fautes (AdF)

L'Analyse par Arbre de Fautes (ou *Fault Tree Analysis (FTA)*) est l'une des techniques logiques et probabilistes les plus importantes dans l'évaluation PRA. L'analyse des arbres de défaillances s'est popularisée au milieu des années soixante. En 1981, la NRC (Nuclear Regulatory Commission) aux Etats-Unis a publié un manuel des arbres de défaillances, [Vesely et al., 1981]. Ce document est devenu la principale source d'information technique sur la façon dont l'analyse des arbres de défaillances devrait être effectuée.

L'analyse par arbre de défaillances est une méthode descendante *top-down* : à partir d'un évènement redouté à la racine, on construit des nœuds en identifiant les combinaisons d'évènements susceptibles de provoquer cet évènement global. Ce processus est itéré jusqu'aux feuilles de l'arbre : les défaillances élémentaires des composants.

Cette technique nécessite d'identifier au préalable l'évènement indésirable (ou redouté) et d'avoir une grande connaissance sur le système afin de pouvoir déduire les causes de l'évènement redouté et de pouvoir raffiner au maximum jusqu'à arriver

à des évènements non décomposables. L'arbre de défaillances est alors un modèle graphique (dessiné à travers des portes logiques conventionnelles) des défaillances qui peuvent être des évènements associés à des défaillances matérielles, des erreurs humaines, des erreurs logicielles, ou tout autre évènement qui mènerait à l'évènement indésirable.

Une coupe est un ensemble d'événements pouvant provoquer l'événement indésirable situé à la racine de l'arbre. Une coupe d'un arbre est dite minimale lorsqu'elle ne contient aucune autre coupe de cet arbre [Ericson, 1999].

L'analyse par arbre de défaillances peut être utilisée pour identifier ces coupes minimales et pour évaluer la probabilité de l'évènement redouté s'il est possible d'associer des probabilités d'occurrence aux feuilles de l'arbre et que les évènements sont indépendants.

1.3.2.4 Etude de danger et d'opérabilité (HazOp)

Comme la méthode AMDEC, l'étude de danger et d'opérabilité (ou *Hazard and Operability analysis, HazOp*) est une méthode ascendante *bottom-up*, ayant pour but d'identifier des dangers d'un système pour la sécurité-innocuité, leurs causes possibles, leurs conséquences ainsi que les actions recommandées pour minimiser leurs probabilités d'occurrence [IEC61882, 2001]. A l'origine, HazOp a été développée dans les années 1970 par Imperial Chemical Industries (ICI) pour traiter les systèmes thermohydrauliques. La méthode HazOp a comme principe d'identifier les dangers d'un système de façon systématique en appliquant à chaque paramètre d'un modèle système un mot guide, produisant une déviation. La méthode HazOp a servi de base à plusieurs variantes orientées selon des domaines d'activité telles que SHARD (ou *Software Hazard Analysis and Resolution in Design*) adapté au systèmes logiciels, et HazOp-UML développé au laboratoire LAAS et utilisé dans les systèmes robotiques [Guiochet, 2016]. C'est cette dernière variante que nous avons adaptée pour les véhicules autonomes. En HazOp-UML, une modélisation en langage UML (Unified Modeling Langage), des diagrammes de cas d'utilisation et des diagrammes de séquences modélisent les interactions entre le système à étudier et l'environnement suivant le processus de chaque fonctionnalité du système. Cette modélisation est effectuée dès le début de la phase de conception et de développement. Elle permet de décrire l'utilisation du système et d'organiser les interactions possibles avec les *acteurs*. Cette première étape d'une analyse HazOp-UML aboutit à une liste de *cas d'utilisation*. Des attributs (ou éléments) doivent être associés pour chaque cas d'utilisation. Nous définissons ces attributs et nous illustrons ces définitions par l'exemple du cas d'utilisation *faire un café* en utilisant la machine Expresso :

- des pré-conditions : des conditions obligatoires au lancement correct du cas d'utilisation. Exemples :
  - ◇ Il y a une tasse,
  - ◇ Il y a du café,
  - ◇ Il a de l'eau dans le réservoir.
- des invariants : des conditions qui doivent toujours être vraies pendant le déroulement du cas d'utilisation. Exemples :
  - ◇ Il y a de l'eau dans le réservoir,
  - ◇ Il y a de l'électricité.
- des post-conditions : des conditions qui doivent être vraies à la fin du cas d'utilisation. Exemples :
  - ◇ Le café est dans la tasse.

L'étape suivante consiste à appliquer un ensemble de mots guides pour chaque attribut dans chaque cas d'utilisation. En considérant le premier attribut *Il y a une tasse*, les mots guides utilisés pour le cas d'utilisation sont :

- No/none : La condition n'est pas évaluée et peut avoir n'importe quelle valeur. Exemple : La présence de la tasse est inconnue.
- Other than : La condition est évaluée mais a une valeur incorrecte. Exemple : Il n'y a pas de tasse.
- As well as : La condition est correctement évaluée, mais d'autres conditions inattendues sont vraies. Exemple : La tasse est mal placée ou a une taille inadaptée.
- Part of : La condition est partiellement évaluée. Certaines conditions sont manquantes. Le part of est utilisé dans le cas des conditions multiples, qui n'est pas le cas dans notre exemple. Sous l'attribut *Il y a une tasse bleue*, un exemple de part of pourrait être qu'il y a bien une tasse, mais d'une autre couleur.
- Early : La condition est évaluée plus tôt que nécessaire et l'état actuel du système n'est ainsi pas évalué. Exemple : La tasse était en place auparavant.
- Late : La condition est évaluée plus tard que nécessaire. Exemple : La tasse est placée tard (le café coule déjà).

J. Guiochet dans [Guiochet et al., 2010], a présenté l'analyse HazOp-UML sous la forme d'un tableau détaillant :

- l'entité UML à laquelle appliquer l'analyse,
- l'attribut ou l'élément considéré,
- le mot guide appliqué,
- la déviation résultant de l'interprétation de la combinaison d'un attribut et d'un mot guide,
- l'effet au niveau du cas d'utilisation,
- l'effet possible sur le système,
- les causes possibles de chaque déviation (logiciel, matériel, environnemental ou suite à des erreurs humaines, etc.),
- l'exigence du niveau d'intégrité : la spécification préliminaire d'un niveau d'intégrité de sécurité pour certains éléments afin de spécifier des techniques adéquates de prévision et d'élimination de fautes pour éviter la déviation,
- les nouvelles exigences de sécurité : si la déviation ne peut être évitée, une nouvelle exigence de sécurité est spécifiée, ces exigences sont numérotées dans notre étude.
- les remarques : explication de l'analyse, interventions supplémentaires, etc.,
- le numéro du danger : les effets sur le système sont identifiés comme des dangers et un numéro est attribué à chacun d'entre eux.

Dans notre approche, nous avons ajouté une colonne supplémentaire désignant la gravité pour chaque déviation, selon la même classification de l'échelle de gravité faite pour la méthode d'analyse de risque AMDEC.

1.3.2.5 Récapitulatif

Le tableau 1.1 récapitule les méthodes présentées, qui se scindent en deux groupes :

- Les méthodes descendantes, qui cheminent des effets vers les causes et qui visent à identifier les origines au niveau composant, des défaillances du système (AdF ou l'APR).

- Les méthodes ascendantes, qui progressent des causes vers les effets et qui servent à déterminer les conséquences au niveau du système des défaillances des composants (AMDEC, HazOp, APR).

Dans ce tableau, on montre aussi les différentes caractéristiques des méthodes d'analyse de risque :

Nous constatons alors que certaines techniques d'analyse de risque ne couvrent pas tous les aspects du processus. Par exemple, l'AMDEC met l'accent sur les composants internes du système et ne considère pas les défaillances multiples tandis que HazOp se focalise sur les processus et les interactions de l'environnement et peut couvrir des défaillances multiples avec le mot clé as well as.

1.4 Conclusion

Les systèmes autonomes (en particulier les véhicules autonomes), de par leur nature, sont des systèmes critiques pouvant provoquer des dégâts catastrophiques sur l'humain, et l'environnement dans lequel ils opèrent.

Dans ce chapitre, nous avons présenté les notions de base et la terminologie de la sûreté de fonctionnement. Nous avons rappelé que la gravité des risques et les futures exigences réglementaires imposent d'identifier systématiquement les menaces sur la sûreté de fonctionnement des systèmes autonomes et les réponses pouvant leur être apportées. L'exposé des méthodes d'analyse de risques montre la nécessité d'en combiner plusieurs pour atteindre cet objectif. Ces méthodes d'analyses de risque apparaissent comme le moyen permettant d'identifier les dangers et pour certaines d'évaluer leur gravité ou/et leur probabilité d'occurrence. Dans le reste de ce document, nous allons étudier les deux méthodes d'analyse de risques AMDEC et HazOp-UML qui peuvent permettre de tolérer des fautes logicielles des mécanismes décisionnels très difficiles à valider.

Dans le chapitre suivant, nous présenterons le mécanisme Safety-Bag en s'appuyant sur quelques exemples d'architectures issues de la littérature. Nous définirons alors le dispositif de sécurité-innocuité Safety-Bag ainsi que les concepts de base des règles de sécurité exécutées en ligne par un tel système de sécurité. Ensuite, nous nous concentrerons sur les méthodes d'élicitation, d'expression et de vérification de ces règles. Nous présenterons finalement l'architecture d'un Safety-Bag générique en finissant par une illustration des différents états du système surveillé.

Méthode d'analyse de risque	**Sens de l'analyse**	**domaine d'origine**	**Phrase clé**	**Caractéristiques**
APR	ascendante ou descendante	militaire puis chimique	Analyser les conséquences de défaillances à partir l'identification des évènements redoutés,	• détermine les évènements redoutés pouvant provoquer des risques.
AMDEC	ascendante	spatial	Recenser les conséquences des défaillances et les évaluer	• est composé de AMDE et de l'évaluation de la Criticité, • AMDE vise à identifier les différents modes de défaillances du système causés par une défaillance unique d'un composant, • met l'accent sur les composants internes du système, • évalue et assigne pour chaque mode de défaillance de chaque élément du système sa gravité.
AdF	descendante	spatial et nucléaire	Organiser les éléments pouvant contribuer à un évènement redouté	• nécessite d'identifier au préalable l'évènement redouté et d'avoir, là aussi, une grande connaissance du système afin de pouvoir déduire les causes de l'évènement redouté et de pouvoir raffiner au maximum jusqu'à arriver à des évènements non décomposables, • quantifie les risques entrainés par des défaillances multiples, • calcule le taux d'occurrence de l'évènement redouté pour tout le système, • détermine les coupes minimales.
HazOp-UML	ascendante	chimique	Recenser les conséquences des défaillances et identifier les actions recommandées pour minimiser leurs probabilités d'occurrence	• met l'accent sur les interactions entre le système et son environnement, • identifie les dangers d'un système d'une façon systématique en appliquant à chaque attribut une liste de mots guide, produisant des déviations, • identifie les exigences de sécurité.

Tableau 1.1 – Caractéristiques des méthodes d'analyse de risque.

Safety-Bag pour les systèmes autonomes

Sommaire

La problématique posée dans le premier chapitre impose de trouver des solutions pour augmenter la sûreté de fonctionnement et la sécurité-innocuité des systèmes autonomes, et en particulier des véhicules autonomes. Ces systèmes autonomes, opérant dans un environnement ouvert, dynamique et incertain, doivent faire face à plusieurs menaces spécifiques. Les mécanismes décisionnels, cœur des systèmes autonomes, font partie de ces menaces comme une source potentielle de fautes.

Pour éviter les fautes de développement résiduelles et les situations dangereuses, des méthodes de vérifications *en ligne* doivent être développées. En effet d'aprés [Lussier, 2007], les méthodes de vérification *hors ligne* telles que le test, le model checking et la preuve de théorème ne sont pas suffisantes pour les systèmes autonomes et spécialement les mécanismes décisionnels. Ceci est du à leur complexité et leur évolution dans un contexte d'environnement quasi-infini, et donc la difficulté de vérifier un nombre d'états potentiellement au delà des limites réalisables. Elles restent cependant incontournables pour l'élimination de fautes avant la mise en œuvre des systèmes.

En revanche, la vérification *en ligne* est une méthode complémentaire aux méthodes précédentes, qui peut assurer la sécurité-innocuité malgré la présence de fautes. Une méthode de vérification en ligne recommandée par la norme [IEC61508-7, 2000] et de nombreux travaux pour les systèmes autonomes ([Schroeder, 1995], [Leucker and Schallhart, 2009] et [Goodloe and Pike, 2010]) est le recours à un dispositif de surveillance indépendant. Ce dispositif permet de surveiller le système en ligne, c'est-à-dire, pendant sa phase d'exécution et est capable d'intervenir en inhibant ou en déclenchant des actions de recouvrement pour remettre le système dans un état sûr.

Dans ce chapitre, nous définissons tout d'abord le composant Safety-Bag et les concepts de base relatifs à ses règles de sécurité avant d'introduire plusieurs exemples. Ensuite, nous présentons les approches existantes dans la littérature pour exprimer et vérifier un ensemble des règles de sécurité exploitables par un Safety-Bag. Finalement, nous proposons une architecture du Safety-Bag générique développée au laboratoire Heudiasyc ainsi que la description de son fonctionnement.

2.1 Mécanisme Safety-Bag

Le mécanisme Safety-Bag est un composant interne de sécurité visant à satisfaire le respect de règles de sécurité-innocuité préalablement définies. Dans cette section, nous introduisons le Safety-Bag et ses concepts avant de présenter ses applications dans la recherche et l'industrie.

2.1.1 Définitions du Safety-Bag

Il existe dans la littérature plusieurs appelations pour le composant indépendant de sécurité. Nous trouvons par exemple : *moniteur de sécurité* [IEC61508-7, 2000] ou *safety-manager* [Pace et al., 2000]. Ce dispositif peut également apparaître sous le nom de *checker* [Py and Ingrand, 2002], ou sous le nom d'*autonomous safety system* [Roderick et al., 2004]. Nous retenons le terme Safety-Bag utilisé par [Klein, 1991], [Lussier, 2007] et [IEC61508-7, 2000].

La norme IEC 61508-7 2000 [IEC61508-7, 2000] définit le Safety-Bag comme suit : *Un Safety-Bag est un moniteur externe de sécurité, mis en œuvre à partir d'un ordinateur indépendant selon une spécification différente qui, vise à assurer la protection du logiciel contre les anomalies de spécification et d'implémentation résiduelles susceptibles d'affecter la sécurité-innocuité. Ce système de surveillance permet de vérifier que l'ordinateur principal exécute des opérations sûres (pas nécessairement correctes). En effet, ce dispositif externe de sécurité surveille en permanence l'ordinateur principal afin de l'empêcher d'entrer dans un état dangereux. En cas de détection d'un état d'ordinateur potentiellement dangereux, il faut que le système soit ramené à l'état sûr par le dispositif externe de sécurité ou l'ordinateur principal.*

Nous considérons dans notre manuscrit le Safety-Bag comme répondant aux précisions apportées dans [Mekki Mokhtar, 2012] : *Le Safety-Bag est un système de sécurité logiciel et matériel qui doit être capable de détecter et éventuellement traiter toute évolution du système vers un état à risque.* Il est chargé de vérifier en ligne un ensemble de règles de sécurité. Si une condition de sécurité est violée, le Safety-Bag intervient en inhibant une action potentiellement dangereuse ou en forçant une action de sécurité afin de maintenir ou de ramener le système dans un état de sécurité. Dans certains cas, le Safety-Bag peut dégrader la disponibilité pour assurer la sécurité-innocuité.

2.1.2 Concept de règles de sécurité

Dans cette section, nous allons préciser les concepts concernant les règles de sécurité puis nous présenterons une classification proposée par le LAAS en états sûrs et en états catastrophiques qui permet de définir l'ensemble des états d'alerte associés à une marge de sécurité.

2.1.2.1 Définitions de base

Il existe dans la littérature une variété de termes liés à la sécurité. Nous trouvons par exemple : exigence de sécurité, contrainte de sécurité, propriété de sécurité, règle de sécurité, etc. En particulier, les termes *exigence de sécurité* et *contrainte de sécurité* ont des définitions divergentes :

- Exigence de sécurité : Dans la littérature, [Medikonda and Panchumarthy, 2009] ont défini l'exigence de sécurité comme suit : *L'exigence pour que le système logiciel soit sûr n'est pas qu'il ne défaille jamais, mais qu'il ne cause pas ou ne contribue pas à une violation de l'une des contraintes du système sur le comportement sûr.* Une autre définition donné par la norme ISO/IEC 9126-1 [ISO9126, 2001] dans le contexte de la robotique est la suivante : *Les exigences de sécurité sont des règles de sécurité pour assurer la sécurité-innocuité du personnel associé à l'utilisation d'un système robotique.* De nombreuses normes et réglementations industrielles et gouvernementales se concentrent sur la spécification des contraintes de sécurité plutôt que sur les exigences de sécurité.

- Contrainte de sécurité : [Medikonda and Panchumarthy, 2009] ont définit une contrainte de sécurité comme suit : *un risque caractérise un état du système qui, pour des raisons de sécurité, ne devrait pas se produire. Si cela est nié et que certaines marges de sécurité sont incluses, nous obtenons une contrainte de sécurité, c'est-à-dire une description d'une propriété que le système devrait posséder pour être sûr.*

  [Firesmith, 2004] a défini la notion de contrainte de sécurité comme suit : *Une contrainte de sécurité est toute contrainte qui spécifie une protection de sécurité spécifique (par exemple : un mécanisme de sécurité architectural, une caractéristique de conception de sécurité, une technique d'implémentation de sécurité). Les contraintes de sécurité comprennent généralement des éléments comme des dispositifs d'inter-verrouillage, des mesures de protection ou des panneaux d'alerte.*

  Nous constatons que ces deux définitions ne sont pas cohérentes. Suivant la première définition, une contrainte de sécurité est une condition suffisante qui,

compte tenu de la marge de sécurité, pourrait être temporairement violée. Par contre, selon la deuxième définition, une contrainte de sécurité est considérée comme une condition nécessaire (le non respect de cette condition implique que la protection de sécurité a échoué).

En se basant sur quelques définitions existantes dans l'industrie et imposées par certaines normes, [Mekki Mokhtar, 2012] a définit les sept termes suivants.

- Exigence de sécurité (ou *Safety Requirement SR*) : En se reposant sur les deux définitions précédentes, Mekki Mokhtar considère une exigence de sécurité comme un objectif de sécurité général. Une exigence de sécurité est donc une exigence générale de haut niveau d'abstraction de ce que signifie pour le système d'être sûr.

- Contrainte de sécurité (ou *Safety Constraint SC*) : Une contrainte de sécurité est une condition suffisante pour empêcher l'occurrence d'une situation dangereuse. Cette définition reprend donc celle donnée par [Medikonda and Panchumarthy, 2009].

- Invariant de sécurité (ou *Safety Invariant SI*) : est une condition nécessaire, c'est-à-dire que la violation d'un invariant de sécurité est intolérable en ce qu'elle implique un dommage et la violation d'une exigence de sécurité de haut niveau d'abstraction. Cette définition se substitue à celle donnée pour les contraintes de sécurité dans [Firesmith, 2004].

- Une condition de déclenchement de sécurité (ou *Safety Trigger Condition STC*) : est une condition qui, lorsqu'elle est évaluée à vrai, enclenche une action de sécurité de la part d'un composant Safety-Bag.

- Marge de sécurité (ou *Safety Margin*) : est la *distance* entre une condition de déclenchement de sécurité et la négation de l'invariant de sécurité. La marge de sécurité est définie de telle sorte qu'elle puisse assurer, lors de la violation d'une condition de déclenchement de sécurité, un délai suffisant avant la défaillance pour effectuer un rétablissement et repasser dans un état sûr.

- Action de sécurité (ou *Safety Action SA*) : est une activité réalisée explicitement pour remettre le système dans un état sûr. [Machin, 2015] a inclus l'action de sécurité dans un autre terme : *intervention de sécurité*. L'intervention de sécurité est la capacité du dispositif de sécurité à contraindre le comportement du système afin d'empêcher le système de violer un invariant

de sécurité. Une intervention de sécurité n'est efficace que lorsque ses pré-conditions sont remplies. [Machin, 2015] a distingué alors deux types d'interventions de sécurité les *inhibitions* et les *actions*.

- ◇ Une inhibition de sécurité : empêche un changement d'état du système (par exemple : verrouiller les roues ; d'un robot lorsqu'il est à l'arrêt).
- ◇ Une action de sécurité : déclenche un changement d'état du système (par exemple : appliquer un freinage d'urgence).

- Règle de sécurité (ou *Safety Rule SR*) : définit une façon de réagir en réponse à une situation dangereuse. Une règle de sécurité est exprimée comme une règle « Si-Alors ». Entre Mekki Mokhtar et Machin, il y a une nuance dans l'expression des règles de sécurité. En effet, dans [Mekki Mokhtar, 2012], la Règle de sécurité = Si [Condition de déclenchement de sécurité] alors [Action de sécurité], par contre, dans [Machin, 2015], la règle de sécurité est définie comme suit : Règle de sécurité = Si [Condition de déclenchement de sécurité] alors [Intervention de sécurité]

  La définition de Machin est donc plus générale, incluant celle de Mekki Mokhtar.

Afin d'éviter toute ambiguïté, nous avons proposé des définitions claires pour la liste des termes dont nous aurons besoin dans la suite de notre méthodologie. En particulier, nous proposons les définitions suivantes inspirées de la littérature :

- Exigence de sécurité : C'est une propriété nécessaire pour la sécurité-innocuité du système. Elle peut être de haut niveau d'abstraction ou raffinée successivement jusqu'à porter sur des composants élémentaires. Sa détermination et son expression dépendent des compétences et des expériences de l'analyste. Les exigences de sécurité sont largement utilisées dans l'industrie pour les applications critiques notamment dans le domaine de l'automobile (Renault) et dans le domaine ferroviaire (Alstom). L'exigence de sécurité est généralement rédigée de façon informelle au moins au plus haut niveau d'abstraction et est tirée conjointement d'analyses de sécurité et des spécifications du système. En sûreté de fonctionnement, il est fortement conseillé que les exigences de sécurité soient identifiées par des analystes de sécurité qui sont des personnes différentes des développeurs du système. Exemples :

  1. La vitesse doit rester inférieure à celle qui pourrait causer des accidents graves (pour les véhicules expérimentaux).

2. Il faut vérifier que l'application de contrôle commande fonctionne correctement. Cette exigence peut être dérivée en exigences plus précises telles que entre autres : la vivacité de l'application de contrôle-commande et la cohérence temporelle.
   - ◇ L'application de contrôle-commande n'est pas bloquée : les commandes sont mises à jour régulièrement, une fréquence nominale de commande $f_{nominaldecommande}$ est définie, mais les imprécisions du signal et des pertes éventuelles de message mènent à tolérer un délai allant jusqu'à 3/f entre deux échantillons (commandes) et la valeur moyenne entre 2 échantillons est de `1/f +/- 20%`.
   - ◇ L'application de contrôle-commande est temporellement cohérente : un horodatage de la commande doit être défini, qui est la date nominale de production de la commande. Une commande plus ancienne ne doit pas être reçue après une commande plus récente. L'implémentation concrète de cette vérification doit tenir compte du glissement entre les horloges de différents calculateurs.
3. Le véhicule doit respecter les limitations de vitesse légale.

- Nécessité de sécurité : C'est une propriété implémentée sur le composant Safety-Bag et nécessaire pour satisfaire/respecter les exigences de sécurité. Pour chaque exigence de sécurité, on recherchera les moyens de l'implémenter dans le Safety-Bag . Si elle est implémentable par le Safety-Bag, une nécessité de sécurité correspondante est rédigée. En effet, la nécessité de sécurité précise ce que le Safety-Bag doit observer, comment il détecte une défaillance à travers ses moyens d'observation et comment il doit réagir. La nécessité de sécurité est généralement rédigée de façon informelle. A partir d'une exigence de sécurité, une ou plusieurs nécessités de sécurité peuvent être dérivées pour être implémentées dans le Safety-Bag.

  Exemples : A partir des exemples considérés précédemment pour les exigences de sécurité, on dérive respectivement les nécessités de sécurité suivantes :

  1. La vitesse du véhicule ne doit pas dépasser 50 km/h (moins une marge de sécurité) en ville. Sinon, le Safety-Bag intervient en inhibant l'accélération.
  2. Le Safety-Bag doit vérifier pour l'application de contrôle-commande :
     - ◇ la vivacité (signal carré (*heartbeat*)) et la mise à jour d'informations clés clairement définies (comme sa vitesse, sa position, l'état de son environnement, etc.).

◇ la cohérence temporelle.

Si ces conditions ne sont pas vérifiées, Le Safety-Bag intervient en bloquant les commandes de l'application de contrôle commande, et en rendant la main au conducteur.

3. Cette exigence de sécurité se décomposerait en deux nécessités de sécurité, la seconde étant nécessaire à la première :

   ◇ Le véhicule doit respecter la vitesse légale qui a été déterminée par le moniteur de sécurité.

   ◇ Le moniteur de sécurité doit être capable de déterminer la vitesse légale.

   Or, cette seconde nécessité n'est pas implémentable par le Safety-Bag tel que nous l'envisageons (le Safety-Bag ne dispose pas actuellement d'une source d'informations fiable pour connaitre la vitesse légale à l'endroit où se trouve le véhicule). Cette exigence ne peut donc pas être implentée dans notre système par des nécessités de sécurité.

- Condition de déclenchement de sécurité (STC) : C'est une condition évaluable par le Safety-Bag permettant au Safety-Bag d'imposer le respect d'une nécessité de sécurité. Le Safety-Bag enclenche une action ou une inhibition de sécurité si la condition de déclenchement de sécurité est vraie. Elle peut être exprimée d'une façon informelle, mais doit être suffisamment précise pour pouvoir être implémentée sans ambiguïté sur le Safety-Bag.

  Exemples : A partir des exemples considérés précédemment, on exprime les conditions de déclenchement de sécurité suivantes :

  1. La borne de vitesse :

     ◇ Informellement : Le véhicule ne doit pas dépasser une vitesse limite. On choisit la vitesse du véhicule sur le CAN comme source de vitesse.

     ◇ Formellement : $V \leq V_{max}$

  2. La vivacité et la cohérence temporelle :

     ◇ La vivacité :

     ✓ Informellement : La période de la mise à jour des commandes par l'application de contrôle-commande est comprise entre deux valeurs seuil.

     ✓ Formellement :

     `Loop`

Commande reçue
$t_i = Now()$; %Now() : horloge du Safety-Bag
Check : $valeur\_seuil1 < t_i - t_{i-1} < valeur\_seuil2$;
%valeur_seuil1 : moitié de la valeur moyenne de la période de mise à jour de la commande et valeur_seuil2 : double de cette même valeur moyenne de la période.
Echec test => intervention de sécurité (mise en sécurité d'urgence)
End Loop

◇ La cohérence temporelle de l'application de contrôle-commande :

✓ Informellement : A chaque réception de données, on vérifie que le temps auquel elles ont été générées n'a pas dépassé un certain seuil de durée de validité.

✓ Formellement :
Now-dv < T_h < Now; Avec dv: étant le seuil de durée de validité de données et T_h: l'horodatage de la donnée reçue.

- Marge de sécurité : C'est la distance entre une condition de déclenchement de sécurité et la négation de la nécessité de Sécurité. La marge de sécurité est définie à partir d'une analyse de sécurité pour fixer la condition de déclenchement de sécurité. Une fois la marge de sécurité et la condition de déclenchement de sécurité définies, une intervention de sécurité devrait être spécifiée et définie de façon formelle. Cette intervention devra être enclenchée pour ramener le système dans un état sûr avant que la marge de sécurité ne soit dépassée. La détermination de la marge de sécurité nécessite une coopération entre les experts de sécurité et les experts du domaine d'application du système.

  Exemple : Dans l'exemple 1, en considérant une intervention de sécurité par inhibition d'accélération, nous avons choisi une marge de sécurité sur la vitesse de 3 km/h. En effet, si on force l'accélération à zéro, le véhicule ne devrait pas physiquement la dépasser, sauf si on est en pente. On garde alors une marge de sécurité de 3 km/h pour être prudent et pour prendre en compte la dynamique du véhicule entre le temps de la détection de la vitesse et le temps où le Safety-Bag réagit en mettant l'accélération à zéro.

- Intervention de sécurité : C'est l'intervention (action et/ou inhibition) explicite exercée par le Safety-Bag pour que le système reste dans un état sûr. Les interventions de sécurité soit assurent la « mise en état sûr » dans un mode

dégradé où une condition suffisante de sécurité est garantie (ultimement l'arrêt du véhicule ou le passage en conduite manuelle) soit permettent la poursuite de la mission en limitant les performances. Une intervention de sécurité doit être exprimée en langage formel. Exemples :

- ◇ Une action de sécurité : déclenche un changement d'état du système (exemple : freiner, `frein=5 Volts`) ;
- ◇ Une inhibition de sécurité : empêche un changement d'état du système (exemple : empêcher l'accélération, `rejeter toute commande d'accélération`).

- Règle de sécurité : Définit une façon de réagir en réponse à une situation dangereuse. Une règle de sécurité est exprimée comme une règle « Si-Alors ».

  Règle de Sécurité = Si [Condition de déclenchement de sécurité] Alors [Intervention de sécurité] Exemple : Si [la vitesse excède 47 km/h] Alors [empêcher l'accélération]

Nous avons maintenu dans notre approche les définitions telles que présentées dans [Mekki Mokhtar, 2012] en ce qui concerne la *condition de déclenchement de sécurité*, *la marge de sécurité* ainsi que la *règle de sécurité* formelle qui doit être complètement précisée. De plus, notre Safety-Bag intervient en faisant soit une inhibition, soit une action de sécurité. Pour cela, nous avons utilisé le terme de [Machin, 2015] *intervention de sécurité* au lieu du terme *action de sécurité* qui englobe l'inhibition et l'action de sécurité.

Cependant, nous avons défini différemment le terme *exigence de sécurité* et nous avons ajouté un nouveau terme *nécessité de sécurité* que nous trouvons primordial dans notre étude. Ce choix est basé sur notre besoin de mettre en œuvre les moyens d'implémentation des exigences de sécurité afin d'exprimer informellement l'intervention du Safety-Bag pour la vérification et la surveillance en ligne de certaines règles de sécurité.

Le reste des termes notamment la *contrainte de sécurité* et l'*invariant de sécurité* dans [Mekki Mokhtar, 2012] ne seront pas utilisés dans notre méthodologie.

2.1.2.2 Classification des différents états du système

Au LAAS, l'équipe de sûreté de fonctionnement a utilisé dans son approche une classification des états possibles du système fonctionnel [Mekki Mokhtar, 2012]. L'invariant de sécurité définit une partition de l'espace d'état du système en états *non-catastrophiques* et en états *catastrophiques*. L'ensemble des états non-catastrophiques peut à son tour être partitionné en ensemble des états *sûrs* et

en ensemble des états d'*alerte*, de sorte que tout chemin d'un état sûr à un état catastrophique traverse un état d'alerte.

La marge de sécurité est visualisée sur la figure 2.1 par la zone qui regroupe les états d'alerte.

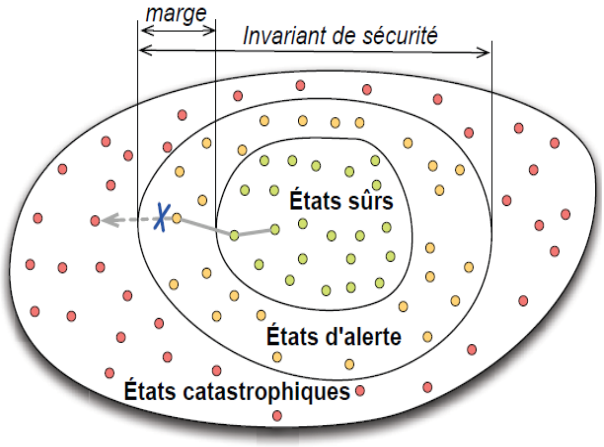

Figure 2.1 – Partition des états possibles du système fonctionnel

Les différents états sont définis comme suit :

- Les états sûrs (ou *safe*) : sont les états dans laquelle la sécurité-innocuité est garantie. Dans le cas du véhicule autonome, ce sont les états d'arrêt du véhicule, les états de conduite manuelle, et les états de conduite autonome dans lesquels les nécessités de sécurité sont respectées.

- Les états catastrophiques : sont les états dans lesquels il est possible à tout moment d'avoir une défaillance à conséquences catastrophiques. Si le système viole l'un des invariants de sécurité, il entre dans un état catastrophique.

- Les états d'alerte : sont des états dans lequel le système est en sécurité, mais qui ont des transitions vers des états catastrophiques.

Une condition de déclenchement de sécurité (STC) doit être exprimée lorsque le système passe d'un état sûr (par exemple, x_s dans la figure 2.2) à un état d'alerte (par exemple, x_w dans la figure 2.2).

Le moniteur de sécurité doit intervenir avant que le système soit dans un état catastrophique (x_c par exemple, dans la figure 2.2). Si le système est dans un état d'alerte, le moniteur de sécurité doit intervenir soit en déclenchant une action de sécurité, soit en inhibant une action afin d'empêcher l'évolution du système vers un état catastrophique. La marge de sécurité ne peut être définie que conjointement avec les actions de sécurité. Elle ne doit pas être trop restrictive, c'est-à-dire, qu'elle doit

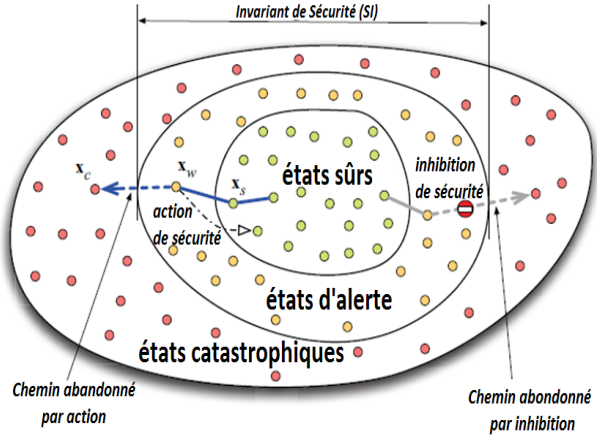

Figure 2.2 – Illustration des principaux concepts

être définie de telle façon que le système puisse accomplir sa mission. Le calcul de la marge de sécurité est une étape cruciale car un compromis doit être trouvé entre la sécurité-innocuité et la disponibilité. Cette étape nécessite une étroite collaboration entre les experts en sûreté de fonctionnement et les experts du domaine d'application du système [Mekki Mokhtar, 2012].

[Mekki Mokhtar, 2012] distingue les composants Safety-Bag suivant qui implémentent seulement des actions de sécurité ou des inhibitions de sécurité : respectivement le moniteur de sécurité et l'inter-verrouillage de sécurité. Ce sont donc des sous-ensembles particuliers de l'ensemble des composants Safety-Bag. Dans nos travaux, nous considérons un composant Safety-Bag qui peut effectuer une intervention de sécurité (à la fois des actions et des inhibitions).

2.1.3 Exemples de dispositifs de surveillance pour la sécurité-innocuité

Dans la littérature, nous trouvons le concept du Safety-Bag dans de nombreux systèmes de domaines critiques variés comme : les centrales nucléaires, le spatial (projet SPAAS [Blanquart et al., 2004] [Blanquart et al., 2003]), le médical (guardian agent [Fox and Das, 2000]), le ferroviaire (Elektra [Klein, 1991] [Erb, 1989]), la robotique (R2C [Py and Ingrand, 2004a] [Py and Ingrand, 2004b]), l'excavateur LUCIE ([Pace and Seward, 2000][Pace et al., 2000], DLR-Co-Worker [Haddadin et al., 2011], etc. Les mécanismes présentés effectuent la surveillance et les actions de mise en état sûr de différentes façons et à différents niveaux de l'architecture. Dans cette partie, nous présentons quelques dispositifs de sécurité-innocuité concernant les systèmes autonomes notamment le composant R2C dans le domaine de la robotique, le projet SPAAS dans le spatial et le système Elektra

dans le domaine ferroviaire.

2.1.3.1 R2C : Request and Resource Checker

Le projet R2C (ou *Request and Resource Checker*) a été développé par l'équipe Robotique et Intelligence Artificielle (RIA) du LAAS-CNRS. Frédéric Py et Félix Ingrand dans [Py and Ingrand, 2004a] et dans [Py and Ingrand, 2004b], ont présenté une nouvelle approche pour intégrer la couche de contrôle de l'exécution dans l'architecture des systèmes autonomes et comment une telle approche s'intègre dans leur architecture logicielle.

Dans leur architecture, l'implantation de la capacité d'autonomie est réalisée dans la couche décisionnelle. La complexité croissante des composants fonctionnels ainsi que la présence de composants d'autonomie deviennent cependant un obstacle à la sécurité-innocuité et la sûreté de fonctionnement du système. Pour résoudre ce problème, les auteurs dans [Py and Ingrand, 2004a] et [Py and Ingrand, 2004b] proposent l'intégration d'un composant de contrôle d'exécution (R2C) dans l'architecture logicielle LAAS, intégration représentée dans la figure 2.3.

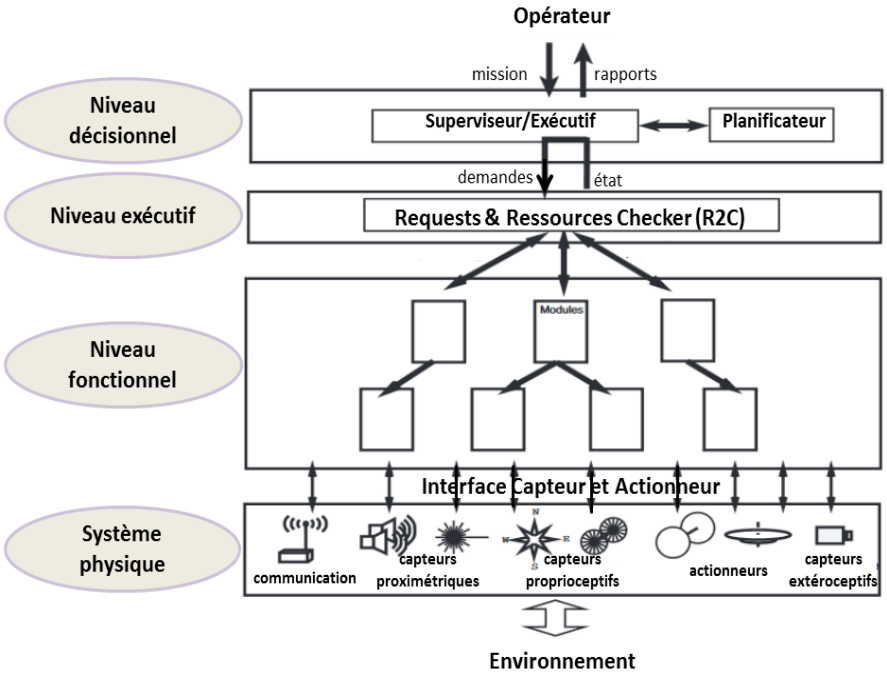

Figure 2.3 – Implémentation de R2C au sein de l'architecture trois niveaux du LAAS

Avant de détailler le composant R2C, nous présentons d'abord l'architecture LAAS [Alami et al., 1998], qui a été conçue à l'origine pour les robots mobiles autonomes. Cette architecture reste assez générale et s'appuie sur un ensemble

cohérent d'outils et de méthodologies, afin de concevoir, d'intégrer, de tester et de valider correctement un système autonome complexe. Cette architecture comporte les deux niveaux décisionnel et fonctionnel de la figure 2.3 :

- Un niveau **décisionnel** (ou *decision level*) : Ce niveau centralise les capacités décisionnelles de haut niveau du système. Il peut inclure un certain nombre de composants tels que (mais non limité à) : un planificateur, qui produit des plans de haut niveau pour atteindre les objectifs, et un superviseur/exécutant, qui décompose et raffine les plans en actions atomiques exécutables par des composants fonctionnels.

- Un niveau **fonctionnel** (*fonctional level*) : Ce niveau, contrôlé par le niveau décisionnel, comprend toutes les fonctionnalités de base du système autonome (capteurs, effecteurs, etc.) et de traitement (planification de mouvement, localisation, traitement d'image, etc.).

La couche de contrôle d'exécution contenant le R2C (*execution control level*, figure 2.3), est introduite dans l'architecture pour augmenter la sûreté de fonctionnement. Cette couche se trouve entre les deux couches fonctionnelle et décisionnelle. Elle est chargée de vérifier en ligne un ensemble de règles de sécurité à la fois sur les requêtes émanant de la couche décisionnelle (précisément du superviseur) et sur les requêtes échangées entre les modules de la couche fonctionnelle.

[Mekki Mokhtar, 2012] a décrit les tâches effectuées par le R2C (présenté dans 2.4) comme suit :

- Il capture tous les événements pouvant modifier l'état du système.

- Il met à jour la base de données des états du système.

- Il vérifie la validité des requêtes dans le composant *vérificateur d'état* de la figure 2.4. Ce composant intègre un *vérificateur par modèle*. A partir d'un graphe d'état modélisant le système, le vérificateur par modèle identifie des actions à entreprendre pour garder le système dans un état sûr.

- Il exécute les actions choisies et envoie des rapports aux deux couches voisines.

Nous devons noter que le vérificateur par modèle est le composant clé de la structure de R2C. Il doit vérifier des règles de sécurité exprimées sous formes d'assertions logiques.

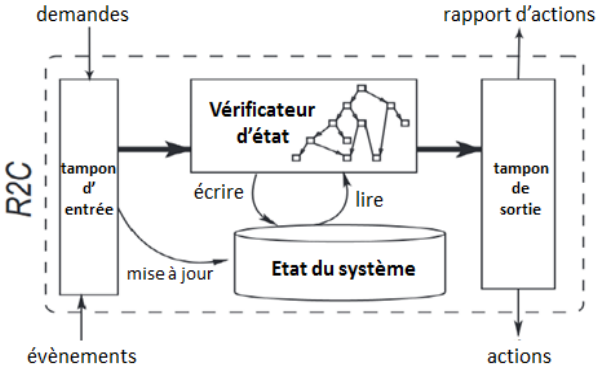

Figure 2.4 – Architecture de système de sécurité-innocuité R2C

2.1.3.2 SPAAS : Software Product Assurance for Autonomy on-board Spacecraft

Le projet spatial SPAAS (ou *Software Product Assurance for Autonomy on-board Spacecraft*) est un projet ESA auquel a collaboré le LAAS-CNRS. Dans le cadre de ce projet, l'équipe de sûreté de fonctionnement du LAAS était chargée de traiter la sûreté de fonctionnement des logiciels autonomes pour les satellites du futur. En effet, les satellites ou les robots spatiaux actuels ont des niveaux d'autonomie limités. Dans le cadre de ce projet, les auteurs ont montré l'importance d'un dispositif de type *Safety-Bag*. Un tel dispositif a été implanté à la fois sur le système de planification utilisé dans des applications spatiales embarquées dans des satellites, et sous la dénomination *plausibility checker* pour un système de contrôle depuis le sol des commandes envoyées à un satellite. Dans le cas du *Safety-Bag*, pour les systèmes embarqués ou du *plausibility checker*, pour le système de contrôle depuis le sol, la problématique est similaire. Le *plausibility checker* basé au sol, vise à soutenir et compléter la validation au sol du logiciel autonome, et en particulier, les procédures de contrôle embarquées avant le téléchargement et l'exécution réelle. Le *safety-bag* vise à surveiller en ligne un ensemble de règles de sécurité afin d'autoriser ou d'interdire l'exécution par l'engin spatial de commandes élaborées par les applications logicielles autonomes. Des règles de sécurité ont été implantées dans une architecture dont laquelle le module *Safety-Bag* est connecté entre les composants de l'application d'autonomie (*Application Under Surveillance*) et le composant TC (*télécommande*) services comme illustré dans la Figure 2.5. Précisons que, à notre connaissance, ce Safety-Bag et ce plausibility checker n'ont jamais été implantés sur un système réel.

TC désigne le module d'exécution des télécommandes selon la terminologie EADS Astrium. Le *Safety-Bag* effectue deux actions majeures [Guiochet and Powell, 2005] :

- Transition : en fonction de la TC et de l'état actuel du système, le *Safety-Bag*

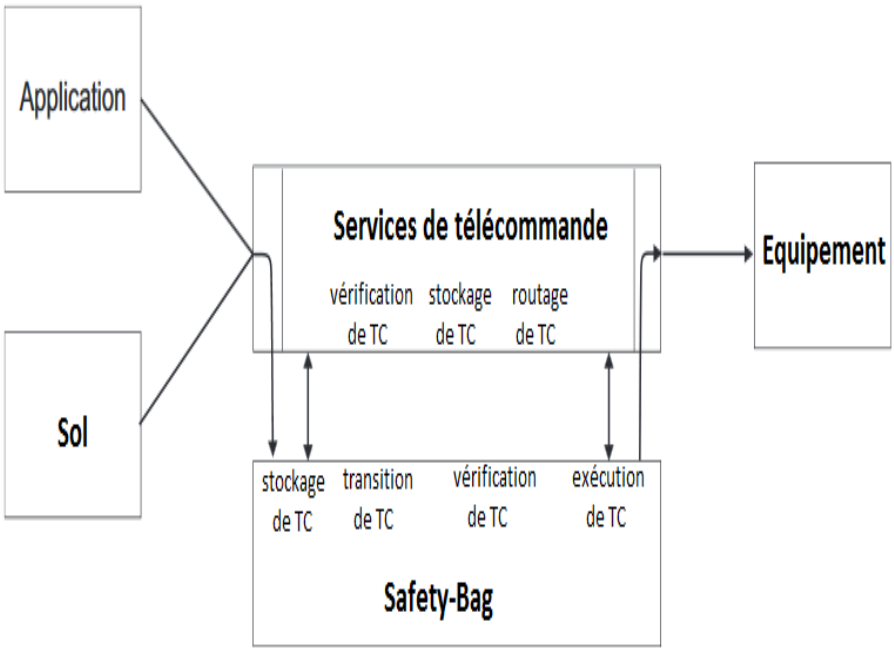

Figure 2.5 – Remote Agent Safety-Bag

détermine la valeur prévue de l'état suivant ;

- Vérification : Le *Safety-Bag* évalue si la valeur prévue de l'état suivant n'est pas un état à risque. Si c'est le cas, l'exécution est bloquée.

2.1.3.3 Elektra : Electronic Interlocking System

Elektra ou *Electronic Interlocking System* est un système d'aiguillage pour le transport ferroviaire, considéré comme le premier *Safety-Bag* développé par Alcatel Autriche [Klein, 1991]. En 2005, il était mis en place dans 80 installations en Autriche et 20 en Suisse. L'architecture du système *Safety-Bag* consiste comme le montre la figure 2.6 en deux canaux : un canal logique (*logic channel*), dédié à la réalisation des fonctions du système et un canal de sécurité (*safety channel*), assurant la sécurité du premier en vérifiant les commandes produites. Si la commande ne respecte pas certaines règles de sécurité, le Safety-Bag inhibera la commande. Le système de sécurité Elektra est composé d'un système expert, dont la tâche consiste à vérifier en ligne (en temps réel) certains calculs du canal logique.

Ce système expert comprend :

- Une base de connaissances, qui contient les règles de sécurité, ainsi que l'état courant du système,

- Un moteur d'inférence, qui choisit quelle règle est applicable puis décide de transmettre ou bloquer la requête.

Dans le contexte de l'application ferroviaire, les auteurs dans [Theuretzbacher, 1986] indiquent la présence d'erreurs potentiellement dangereuses menant à des situations catastrophiques sans les décrire. D'après [Kantz and Koza, 1995], le CENELEC (Comité Européen de la Normalisation Électrotechnique) préconise que le taux de défaillances critiques soit inférieur à 10^{-9} par heure pour la sécurité des petits systèmes d'enclenchement. Simultanément, pour respecter l'horaire des trains, éviter des pertes financières et maintenir la réputation du transport ferroviaire, la disponibilité et la fiabilité des systèmes d'enclenchement ferroviaires sont indispensables. *Austrian Federal Railways* exigent d'avoir moins d'une interruption de service en 10 ans (soit 1 interruption de service en 9* 10^4 heures). C'est pour atteindre cet objectif qu'a été mis en place un Safety-Bag dans le cadre de projet Elektra, qui vise à éviter toute défaillance pouvant engendrer une situation catastrophique telle qu'un écrasement de trains à titre d'exemple.

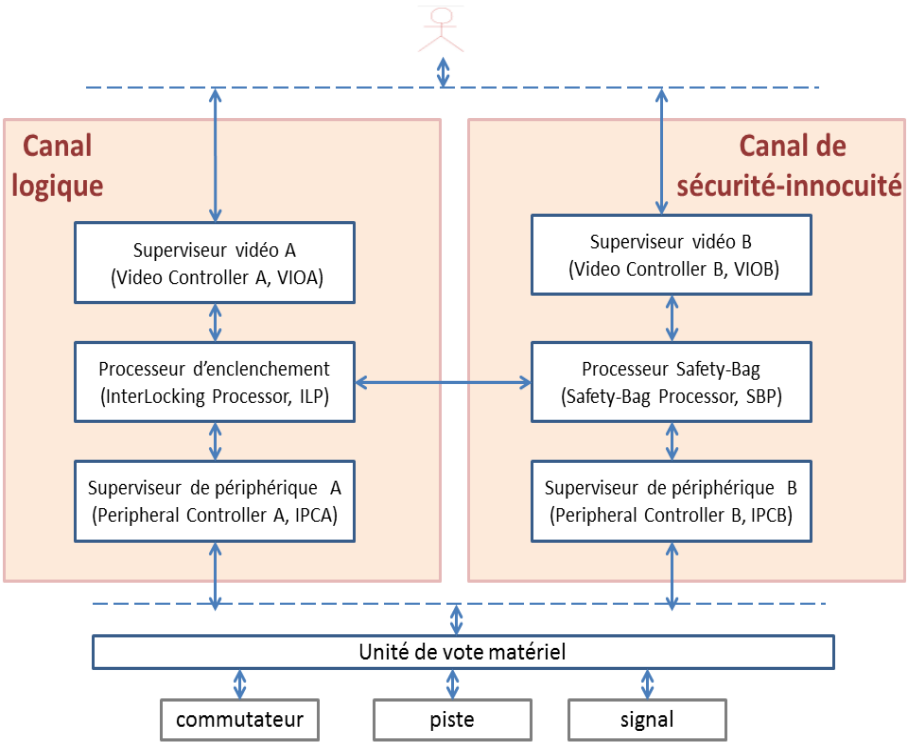

Figure 2.6 – Architecture du système de sécurité-innocuité Elektra

L'architecture Elektra a deux niveaux de redondance :

- Le niveau 1 est réalisé par le couple canal logique et canal de sécurité présenté précédemment.
- Le niveau 2 s'appuie sur une couche logicielle appelée VOTRICS (ou *Voting Triple-Modular-Redundancy Computer System*), assurant une redondance ma-

térielle pour chacun des deux calculateurs des deux canaux afin d'augmenter la disponibilité du système.

Dans la figure 2.6, le processeur principal ILP (ou *Inter-locking Processor*) élabore une commande et l'envoie au processeur du Safety-Bag (SBP), puis l'envoie à l'IPCA.

Comme dit précédemment, le SBP est composé d'un moteur d'inférence et d'une base de connaissances contenant les règles et une mémoire de travail. Le moteur d'inférence transforme d'abord la commande en un GOAL, qui est l'expression logique de l'autorisation de la commande. Puis, il lit les données relatives à l'état du système, et sélectionne la ou les règles applicables (avec un système de gestion des conflits entre ces règles). L'exécution de la règle consiste à transmettre la requête à l'IPCB ou à la bloquer.

Lorsque les calculs d'IPCA et d'IPCB sont effectués dans les deux processeurs, un vote matériel est réalisé. Il consiste à ne pas transmettre la commande calculée par le canal logique et à mettre le système dans un état sûr lorsque les deux canaux de calcul ne sont pas d'accord (comme par exemple mettre les feux à rouge et bloquer tous les trains en attendant que la situation évolue) [Guiochet and Powell, 2005].

2.1.4 Discussion

Un composant indépendant de sécurité, aussi appelé *Safey-Bag*, permet de détecter des erreurs de systèmes complexes. Il est responsable de la supervision des commandes produites par le système de contrôle et il impose le respect de conditions de sécurité pour prévenir les défaillances catastrophiques. Le Safety-Bag lui même ne doit pas avoir de modes de défaillance catastrophique. Afin d'éviter les causes communes de défaillances entre le système opérationnel et des composants du Safety-Bag, il doit être spécifié et développé indépendament. Il doit également disposer de moyens d'actions et de détection indépendants des fautes tolérés et donc au moins partiellement redondants [Lussier, 2007].

Dans la figure 2.7, nous présentons comment les différents dispositifs de sécurité des exemples s'insèrent dans les architectures de chaque système. Ceci permet de montrer leur capacité à être indépendant du reste du système.

- La couche de sécurité de R2C est insérée entre la couche décisionnelle et la couche fonctionnelle. L'indépendance de R2C concerne la définition des règles de sécurité.

- En plus de l'indépendance lors de la détermination des règles de sécurité par rapport au développement, SPAAS intègre deux systèmes complètement

indépendants : le système sol *plausibility checker* et le *Safety-Bag* intégré au satellite.

- Dans Elektra, il existes deux canaux de traitement (canal de commande et canal de sécurité) reliées grâce à différents canaux de communication.

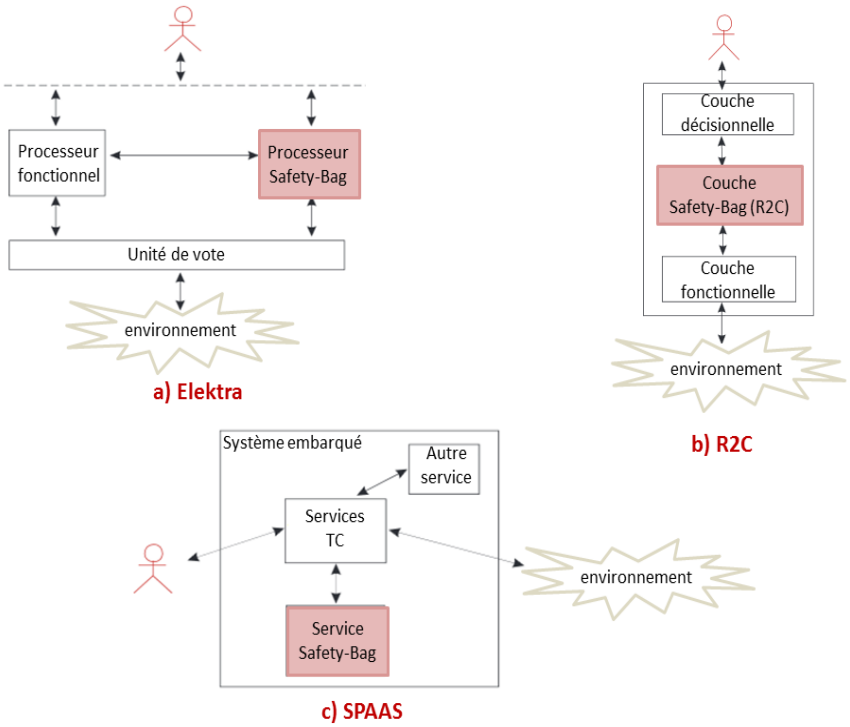

Figure 2.7 – Les trois architectures incluant les différentes approches du dispositif de sécurité [Guiochet and Powell, 2005]

Les trois dispositifs de sécurité évoqués dans les exemples introduisent une redondance par rapport au système fonctionnel, mais n'intègrent aucune redondance interne.

- La redondance de R2C consiste en sa capacité à vérifier des règles de sécurité indépendantes par rapport aux composants fonctionnels du système. Les règles de sécurité sont identifiées dans un processus d'analyse indépendant du processus de développement des composants fonctionnels. L'objectif de R2C est l'intégration de cette capacité dans l'architecture logicielle. La problématique de l'architecture matérielle et de la redondance du matériel n'a pas été abordée dans le projet.

- SPAAS est un composant supplémentaire de surveillance de la couche de communication (TC). Cependant, le service Safety-Bag (dans le satellite)

Dispositif de sécurité	Domaine d'application	Type d'intervention	Indépendance physique	Fautes traitées
R2C	robotique	interception et modification de requête entre la couche décisionnelle et la couche fonctionnelle	Non	• fautes de conception (Exemple : complexité des mécanismes décisionnels, etc.) • situation adverses non prévues de l'environnement ouvert
SPAAS	spatial	ordonnancement de commande/ pré-validation hors ligne	Oui	• fautes ou défaillances du matériel et du logiciel • situation adverses non prévues (séquences d'évènements inconnus liés à l'espace)
Elektra	ferroviaire	blocage de processus dangereux (inter-verrouillage)	Oui	• fautes de conception (Exemple : complexité de la topologie du réseau ferroviaire) • fautes physiques

Tableau 2.1 – Tableau récapitulatif caractérisant les dispositifs de sécurité.

n'offre pas de redondance matérielle. Par contre, les commandes sont également vérifiées par le plausibility checker au sol, ce qui est une redondance forte vis-à-vis des défaillances du Safety-Bag, mais inapplicable dans d'autres domaines comme l'automobile.

- Dans Elektra, les deux canaux de traitement assurent une redondance entre le canal de commande et le canal de sécurité. Il n'y a pas de redondance interne du canal de sécurité, mais grâce au système de vote, ses défaillances n'interrompent pas le système.

Le tableau 2.1 rappelle les spécificités de chaque dispositif de sécurité étudié dans les exemples.

Notre solution est très proche du composant R2C. Cependant, notre Safety-Bag intègre une redondance interne sans dispositif de vote.

2.2 Elicitation, expression et vérification des règles de sécurité

Dans cette section, nous présentons quelques pratiques décrites dans la littérature pour éliciter les règles de sécurité puis exprimer et vérifier formellement ces règles. Pour chacune de ces sous-sections, nous reprendrons brièvement les trois exemples courants R2C, SPAAS et Elektra avant de présenter l'approche formelle SMOF du LAAS.

2.2.1 Elicitation des règles de sécurité

La spécification des règles de sécurité dans le système de sécurité R2C [Py and Ingrand, 2004a] [Py and Ingrand, 2004b] n'a pas suivi de méthode particulière. Les règles de sécurité ont principalement été identifiées lors d'entretiens avec des spécialistes de la robotique et des modules fonctionnels utilisés. L'expertise des concepteurs est dans ce cas la garantie de la cohérence et de la complétude des règles de sécurité.

Dans le cas du projet spatial SPAAS, la détermination des règles de sécurité n'étant pas l'objectif du projet, il n'existe aucune section dédiée à ce sujet. L'élicitation des règles de sécurité, que ce soit au niveau du *Safety-Bag* ou du *plasibility checker* n'a suivi aucun processus systématique ou aucune démarche particulière.

Par contre, en ce qui concerne le Safety-Bag Elekra, [Erb, 1989] mentionne que des analyses de type AMDEC (Analyse des Modes de Défaillances, leurs Effets et leur Criticité) du système électromécanique (sans logiciel, uniquement avec des relais) ont été effectuées afin de déterminer des exigences de sécurité. A partir de ces exigences, des règles de sécurité ont été dérivées en utilisant les normes, en consultant les experts et grâce à certaines expériences des développeurs dans le domaine ferroviaire. En revanche, aucune méthode de détermination de ces règles n'est présenté.

Dans [Mekki Mokhtar, 2012], le cas d'étude est le projet de robot déambulateur MIRAS (Multimodal Interactive Robot for Assistance in Strolling) qui, vise à concevoir et à mettre au point un robot semi-autonome pour l'aide à la verticalisation et à la déambulation, utilisable dans les hôpitaux pour les personnes âgées qui souffrent de problèmes d'équilibre et d'orientation. L'objectif de ce robot est de rendre ces patients plus autonomes, en les aidant à se lever, à marcher et à s'asseoir.

Malgré qu'il n'y ait aucune implémentation réelle d'un système Safety-Bag dans cette étude, l'auteur a suivi un processus systématique pour générer un ensemble de règles de sécurité exécutables en ligne par un dispositif indépendant de sécurité.

Cette méthodologie, basée sur la méthode d'analyse de risques HazOP-UML commence par une modélisation en langage UML des diagrammes de cas d'utilisation et des diagrammes de séquences mettant respectivement l'accent sur les interactions envisageables entre le système et son environnement extérieur et sur les collaborations entre objets selon un point de vue temporel (à travers la chronologie des envois de message).

Pour chaque cas d'utilisation, un ensemble d'attributs (pré-conditions, post-conditions et invariants) est identifié et appliqué à une liste de mots guide afin de déterminer par la suite les déviations, qui sont les dangers potentiels. Quand la gravité de la déviation est non nulle (c'est-à-dire soit une gravité mineure, modérée, sérieuse, sévère, critique ou fatale), l'auteur a précisé les effets de cette déviation sur le cas d'utilisation ainsi que sur le monde réel pour chaque cas d'utilisation.

L'étape suivante consiste à éliciter les règles de sécurité à partir de la colonne *Déviations* de l'analyse de risque de MIRAS. Une génération des contraintes de sécurité est effectuée à partir de cette colonne. Ensuite, Mekki Mokhtar tente de spécifier les conditions de déclenchement de sécurité correspondantes.

Sa méthodologie proposée a aussi été utilisée pour analyser les risques du bras robotisé mobile du projet PHRIENDS (Physical Human-Robot Interaction Dependability and Safety). Ce bras motorisé opère dans un environnement dynamique en présence d'autres objets en mouvement notamment les humains, d'autres robots, etc. Dans cette étude, l'auteur identifie les contraintes de sécurité à partir des effets obtenus dans les trois colonnes suivantes : *Effet sur le cas d'utilisation*, *Effets sur le système* ainsi que *Déviations*.

Dans ces deux études, les auteurs montrent qu'il est généralement difficile de spécifier des contraintes de sécurité exploitables à partir des effets obtenus dans les colonnes *Effets sur le cas d'utilisation* et *Effets sur le système*. Cette difficulté est due à leur haut niveau d'abstraction. Par contre, la colonne *Déviations* permet, de façon plus pertinente que les deux autres, de spécifier des contraintes de sécurité exploitables.

2.2.2 Expression et vérification des règles de sécurité

Après la détermination des règles de sécurité, nous discutons de leur expression formelle qui permettent des vérifications formelles notamment de cohérence. Après l'examen des approches de R2C, SPAAS et Elektra, nous présentons l'évolution de l'expression manuelle des règles de sécurité dans [Mekki Mokhtar, 2012], à laquelle [Machin, 2015] ajoute la vérification par Model Checking, qui aboutit à l'approche SMOF pour synthétiser les stratégies de sécurité.

- Pour exprimer les règles de sécurité dans R2C, un langage de haut niveau d'abstraction a été développé. Ce langage d'expression est composé d'un ensemble d'opérateurs logiques tels que *always*, *never*, *running*, *last*, et des symboles logiques usuels (tels que & &, ||, =>). [Guiochet and Powell, 2005] ont constaté qu'il existe des règles de sécurité qui sont interprétables sans trop de difficultés pour les non experts du domaine, et d'autres qui nécessitent une connaissance très fine des modules fonctionnels utilisés. Leur lecture est alors complexe (règle 1 dans la figure 2.8 par exemple).

```
Check{
/* Règle 1 */
/* On ne peut attacher de Motion Estimator (ME) ou de Sensor Estimator (SE) */
/* à POM que si le modèle géométrique du robot a déjà été défini */

always : (running (pom.addME) | | running (pom.addSE))  =>  last (pom.SetModel);
                                                    .
                                                    .
                                                    .
/* Règle 9
* On ne peut communiquer via l'antenne si le robot bouge */

never: running (antenna.Comunicate) && running (rflex.TrackSpeedStart);

                                                    .
                                                    .
                                                    .
/* Règle 11
* Le robot ne doit pas dépasser la vitesse de 0.9 m/s */

never:  running (rflex.TrackSpeedStart with arg.name.value. v>0.9);
                                                    .
                                                    .
};
```

Figure 2.8 – Exemple de règles de sécurité dans le système R2C

Ces règles sont ensuite traduites par un compilateur développé au laboratoire LAAS : ExoGen, pour former un arbre logique (ou arbre binaire) décomposant chaque élément en tests binaires. Chaque nœud de l'arbre représente un test d'une variable d'état et se présente sous la forme d'un prédicat. Ces prédicats décrivent le contexte qui doit être vérifié pour permettre l'exécution d'un service. Une règle de sécurité en ExoGen exprime une contrainte de sécurité ainsi que l'action de sécurité enclenchée si cette contrainte est violée. Une vérification de ces règles sera effectuée après à travers un graphe OCRD (Ordered Constrained Rule Diagram).

- En ce qui concerne le projet spatial SPAAS, deux règles de sécurité ont été mises en œuvre dans le prototype du *Safety-Bag* : l'une concernant le contrôle de l'énergie totale du système et la deuxième que tout déclenchement de consommation de ressources sera suivi de l'arrêt de cette consommation. Ces deux règles ont été intégrées au code source sans langage informatique

spécifique. Le Safety-Bag lui-même a été programmé en tant qu'un service (et non comme un composant indépendant de sécurité) du système de gestion des données du contrôle de satellite. Par contre, sur le *plausibility checker*, un travail plus important a été effectué pour exprimer les règles de sécurité et les traduire automatiquement en langage machine et pour représenter l'état du système. En effet, des propositions ont été faites au niveau de l'expression des règles de sécurité basées sur une expression en BNF (Backus Naur Form)[Lecubin et al., 2001].

[Guiochet and Powell, 2005] concluent la présentation du projet SPAAS en insistant sur la nécessité d'une approche systématique de détermination des règles de sécurité et sur la nécessité de démontrer qu'une erreur du *Safety-Bag* ne peut entrainer de défaillance catastrophique. Grâce à un parser qui a été développé au sein du *plausibility checker*, les règles de sécurité ont été traduites automatiquement en langage machine exécutable. Aucun outil de vérification n'a été développé dans le *Safety-Bag*.

- Dans le domaine ferroviaire, le langage de programmation logique PAMELA, de haut niveau d'abstraction, a été utilisé pour définir les règles de sécurité exploitables par le Safety-Bag Elektra. Les articles ne donnent pas des exemples des règles de sécurité dans ce langage, mais uniquement une traduction en anglais (voir 2.9) de ces règles, exprimées sous la forme de IF [*condition*] THEN [*action*] [Klein, 1991].

IF there is a request to throw over a switch
AND the switch is either in position full left or full right
AND the switch is not locked or interlocked
THEN commit the request
IF there is a request to interlock a switch
AND the switch is in correct position
THEN commit the request
IF there is a request to perform the check to establish the route
AND all tracks and switches in the route are free
AND the start signal is not locked
AND the status of the route is locked
THEN commit the request
[...]
If a request could not be committed by one of these rules it is rejected
by a general rule with low priority

Figure 2.9 – Exemple de règles de sécurité dans le système Safety-Bag Elektra

Cependant, aucun outil de vérification de ces règles n'a été mentionné. Considérant les normes ferroviaires en vigueur, il est possible que des méthodes formelles aient été utilisées pour les valider.

Mekki Mokhtar dans son approche a exprimé manuellement les règles de sécurité. La première étape consiste à faire la négation de chaque déviation. Nous rappelons qu'une déviation résulte de l'interprétation de la combinaison d'un attribut et d'un mot guide effectuée afin d'identifier les situations conduisant à des risques potentiels pour la sécurité-innocuité du système et de son environnement. A travers cette négation, peuvent être extraites des contraintes de sécurité exprimées en langage naturel obtenues par l'analyse de risque du système. Ces contraintes seront exprimées par la suite en langage formel sous forme de prédicats en utilisant des variables pertinentes à la sécurité et observables par le moniteur de sécurité. Ensuite, en considérant chaque invariant de sécurité, une marge de sécurité est définie. Notons que les marges de sécurité sont spécifiées pour permettre au moniteur de sécurité de déclencher une action de sécurité afin de remettre le système dans un état sûr avant que ce dernier ne puisse arriver dans un état catastrophique. Un invariant de sécurité conduit alors à un ensemble d'états du système qui peuvent être divisés en un ensemble sûr et un ensemble d'alerte, avec une condition de déclenchement de sécurité. Après avoir défini les conditions de déclenchement de sécurité à partir les contraintes de sécurité, une action de sécurité sera déclenchée pour remettre le système dans un état sûr et une règle de sécurité est alors définie.

Différemment de l'approche précédente dite *par états d'alerte* dans [Mekki Mokhtar, 2012], qui vise à déterminer manuellement des règles de sécurité , [Machin, 2015] a choisi de synthétiser automatiquement des stratégies de sécurité sûres (la sécurité est caractérisée comme la non-atteignabilité des états catastrophiques) et permissives (la permissivité est la capacité de permettre au système d'effectuer ses tâches) à partir du modèle pour couvrir un invariant. Une stratégie de sécurité est définie comme : *un ensemble de règles de sécurité visant à garantir un invariant de sécurité, c'est-à-dire à empêcher tous les chemins menant à la violation d'un invariant. La stratégie a aussi pour but de ne pas brider la polyvalence du système [Machin, 2015].* Pour concevoir alors ces règles de sécurité, Machin a disposé de la spécification des moyens d'observation et d'interventions du moniteur ainsi que des invariants de sécurité, qui sont déjà identifiées par la méthode d'analyse de risque HazOp-UML puis formalisés en utilisant uniquement les variables observables. Afin de spécifier les stratégies de sécurité, Machin a recours à un processus décrit dans le schéma descriptif 2.10.

La première étape consiste à la construction d'un modèle pour chaque invariant. Il comprend toutes les informations nécessaires pour déterminer les stratégies de sécurité liées à cet invariant. Ensuite une stratégie est choisie pour chacun des invariants, en les traitant séparément les uns des autres. Finalement, les invariants sont réunis pour vérifier que l'ensemble des stratégies n'appliquent pas d'interventions conflictuelles.

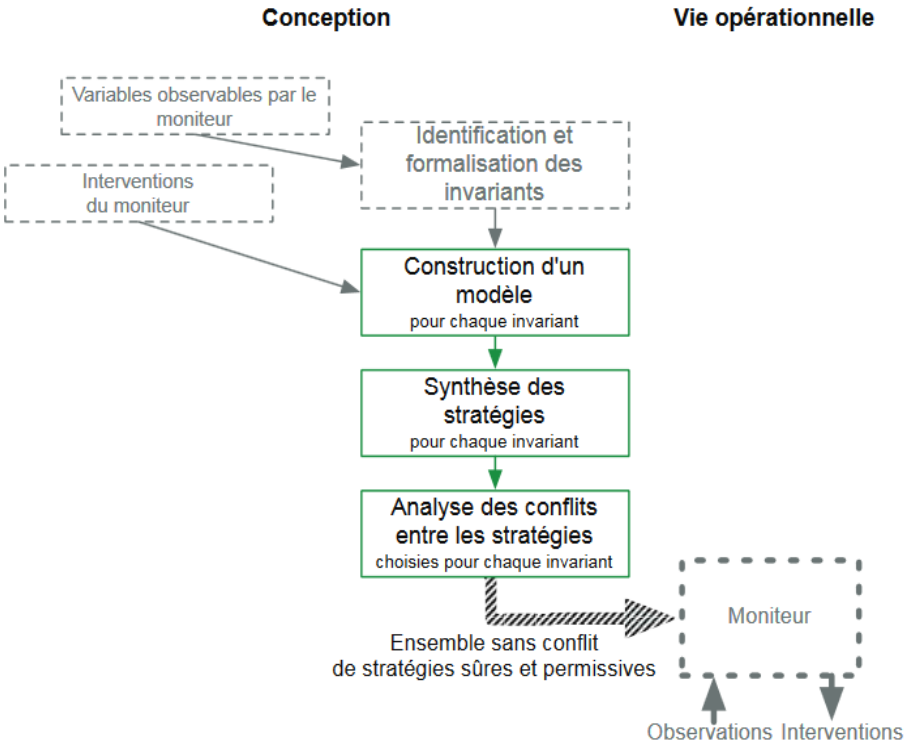

Figure 2.10 – Principales étapes pour spécifier les stratégies de sécurité [Machin, 2015]

[Machin, 2015] mentionne trois outils de vérification pour un modèle discret présents dans la littérature.

- SPIN [Holzmann, 1997] : est un système de vérification efficace pour les modèles de systèmes logiciels distribués. Il a été utilisé pour détecter des erreurs de conception dans des applications allant des descriptions de haut niveau d'algorithmes distribués au code détaillé pour le contrôle des échanges téléphoniques. Il permet de modéliser des systèmes communicants dans le langage Promela.

- UPPAAL [Bengtsson et al., 1996] : est un outil de vérification automatique des propriétés de sécurité et de vivacité bornée de systèmes temps-réel modélisés comme des réseaux d'automate temporisés. Ce vérificateur comprend : une interface graphique qui supporte les représentations graphiques et textuelles des réseaux d'automate temporisés, un compilateur qui transforme une certaine classe de systèmes hybrides linéaires en réseaux d'automates temporisés, et un vérificateur de modèle basé sur des techniques de résolution de contraintes.

- NuSMV [Cimatti et al., 2002] : vérifie des automates. D'après A.Cimatti et al., NuSMV est une plate-forme pour la vérification de modèles symboliques, conçue pour être applicable dans des projets de transfert de technologie.

[Machin, 2015] a choisi NuSMV pour encoder et vérifier son modèle du fait que le comportement du modèle est un automate très proche de celui généré par NuSMV lors de la déclaration des variables.

Cependant, pour l'utilisateur, écrire et vérifier manuellement les règles de sécurité dans NuSMV peut être fastidieux et source d'erreurs.

Pour résoudre ce problème, SMOF [Machin et al., 2016] (Safety MOnitoring Framework for Autonomous Systems), un projet de recherche élaboré au LAAS-CNRS, est apparu comme un outil de synthèse des règles de sécurité pour les systèmes autonomes basé sur l'analyse de risques HazOp-UML. Cette approche prend en considération à la fois la sécurité-innocuité (l'absence des conséquences catastrophiques) et la permissivité (la capacité d'assurer la sécurité sans réduire les fonctionnalités du système).

La figure suivante 2.11 présente une vue d'ensemble du processus utilisé dans le laboratoire LAAS [Guiochet, 2015]. Ce processus couvre deux moyens de la sûreté

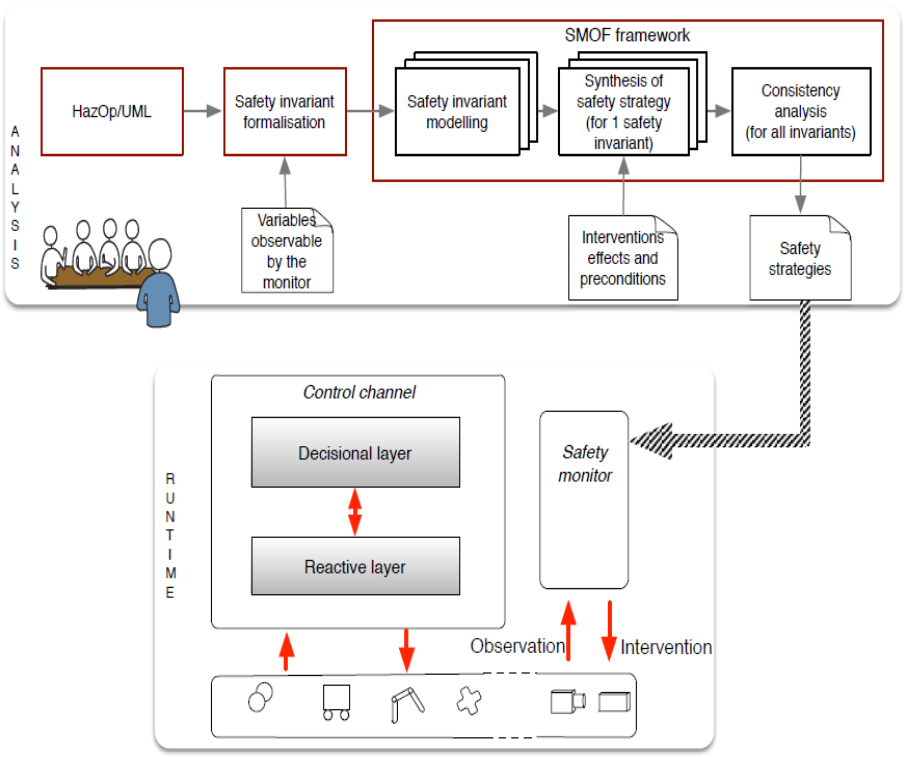

Figure 2.11 – Vus d'ensemble de processus

de fonctionnement : la prévision des fautes et la tolérance aux fautes. Il commence comme montre la figure par une analyse de risque HazOp-UML (prévision des fautes). L'invariant de sécurité est ensuite formellement exprimé avec des prédicats sur des variables observables par le moniteur. Ensuite, cet invariant est modélisé dans le modèle SMOF (tolérance aux fautes) afin de synthétiser les stratégies de sécurité.

[Masson et al., 2017] ont suivi la même démarche que celle considérée dans [Machin et al., 2016]. Ils ont commencé tout d'abord par une analyse de risque HazOp-UML pour dériver un ensemble des invariants de sécurité. Leur étude consiste à spécifier des règles de sécurité de surveillance active (en ligne) appliquée sur un système robotique autonome pour la maintenance du système d'éclairage des aérodromes, qui est développé par Sterela. Le moniteur de sécurité est chargé dans ce cas de maintenir le système dans des états satisfaisant ses invariants. Après une étape de formalisation, des règles de sécurité appropriées sont synthétisées par SMOF (Safety MOnitoring Framework), qui utilise le vérificateur de modèle NuSMV.

La structure SMOF est composée comme montre la figure 2.11 [Machin et al., 2016] :

- d'une modélisation de l'invariant de sécurité : Le modèle SMOF formalise une partie du système liée à un invariant de sécurité, observable par le moniteur de sécurité. Ce modèle rassemble toutes les informations nécessaires pour produire des stratégies de sécurité assurant cet invariant de sécurité :

  ◇ Comportement : automate du système en l'absence du moniteur, contenant tous les chemins vers les états catastrophiques. Le comportement contient alors les états d'alerte.

  ◇ Interventions : capacités du moniteur de sécurité à contraindre le comportement du système. Une règle de sécurité associe une intervention de sécurité (ou une combinaison d'interventions) à l'un de ces états d'alerte. Une stratégie est composée de potentiellement plusieurs règles de sécurité.

  ◇ Sécurité-innocuité et permissivité : propriétés souhaitées de l'action du moniteur. Elles sont attachées à une paire de comportement et de stratégie.

  Pour formaliser leur modèle, [Machin et al., 2016] ont choisi d'utiliser des langages et des outils disponibles dans la communauté du model-checking.

- d'une synthèse des stratégies de sécurité (pour un seul invariant de sécurité) : La synthèse des stratégies de sécurité est basée sur le model-checker NuSMV2. L'algorithme de synthèse prend en entrées le modèle de comportement d'un invariant, les interventions disponibles et les propriétés à assurer (la sécurité et la permissivité). Il produit un ensemble de stratégies de sécurité alternatives, chacune d'elles satisfaisant les propriétés. Conceptuellement, une stratégie assigne une combinaison d'interventions de sécurité à chaque état d'alerte.

Dispositif de sécurité	**domaine d'application**	**méthode de détermination des règles de sécurité**	**langage d'expression des règles de sécurité**	**vérification en ligne des règles de sécurité**
Elektra	ferroviaire	normes et experts en ferroviaire	PAMELA	moteur d'inférence du canal de sécurité
R2C	robotique	experts en robotique	ExoGen (parser/compilateur)	graphe OCRD
SPAAS	spatial	experts dans le domaine aérospatial	structures de données (BNF) + parser	_
_	robot déambulateur MIRAS	analyse de risque HazOp-UML	langage naturel	_
_	robot manipulateur Omnirob de Kuka	analyse de risque HazOp-UML	SMOF	vérificateur de modèle NuSMV
_	robot Sterela	analyse de risque HazOp-UML	SMOF	vérificateur de modèle NuSMV

Tableau 2.2 – Tableau récapitulatif décrivant les méthodes d'élicitation, d'expression et de vérification des règles de sécurité existantes dans la littérature.

- d'une analyse de cohérence (pour tous les invariants de sécurité) : L'analyse de la cohérence, ou l'analyse des conflits entre les stratégies, vise à vérifier si deux stratégies, synthétisées à partir d'invariants différents, appliquent des interventions incompatibles en même temps (par exemple, un freinage et une accélération en même temps). Le model-checker détecte facilement les cas grossièrement incohérents, comme l'augmentation et la diminution d'une variable. Mais, il pourrait y avoir des incohérences moins évidentes non capturées dans les modèles abstraits. L'utilisateur doit dans ce cas spécifier les combinaisons interdites. Nous rappelons que dans cette approche, chaque invariant de sécurité est modélisé séparément et dispose d'une stratégie de sécurité dédiée. Pour vérifier l'effet de la fusion des stratégies retenus pour chaque invariant, les modèles SMOF sont transformés en module NuSMV dans un modèle global.

2.2.3 Récapitulatif

Nous pouvons récapituler dans le tableau 2.2 les différentes méthodes utilisées dans la littérature afin de déterminer, d'exprimer et de vérifier les règles de sécurité exécutées en ligne par le dispositif de sécurité.

2.3 Architecture du Safety-Bag générique

Nous présentons dans cette section une architecture générique de Safety-Bag développé par l'équipe Plateformes du laboratoire Heudiasyc. La présentation de l'architecture sera suivie des diagrammes d'états décrivant le comportement du safety-bag.

2.3.1 Intégration du Safety-Bag dans un système autonome

Le Safety-Bag s'insère entre l'application de contrôle-commande et les actionneurs 2.12. Il assure le respect des règles de sécurité, lesquelles relient des conditions de déclenchement de sécurité à des interventions de sécurité. Le Safety-Bag intercepte les commandes puis les transmet, les modifie, les inhibe ou les remplace selon les interventions de sécurité associés aux conditions de sécurité qu'il surveille. En cas de défaillance, le Safety-Bag est capable de déclencher les alertes pour solliciter l'intervention de l'opérateur. Il assure également l'archivage dans un système d'enregistrement *log* des informations concernant tous les évènements impactant la sécurité.

Afin de vérifier que l'application de contrôle-commande respecte ou non les règles de sécurité, le Safety-Bag peut utiliser des informations qu'elle lui fournit (sans toutefois leur faire confiance), des informations issues des mêmes capteurs que ceux utilisés par celle-ci et des informations fournies par des capteurs qui lui sont propres, suivant les composants ciblés. Les capteurs propres au Safety-Bag permettent de détecter d'éventuelles défaillances des capteurs utilisés par l'application. Ces capteurs doivent privilégier la fiabilité et leurs défaillances doivent être détectables.

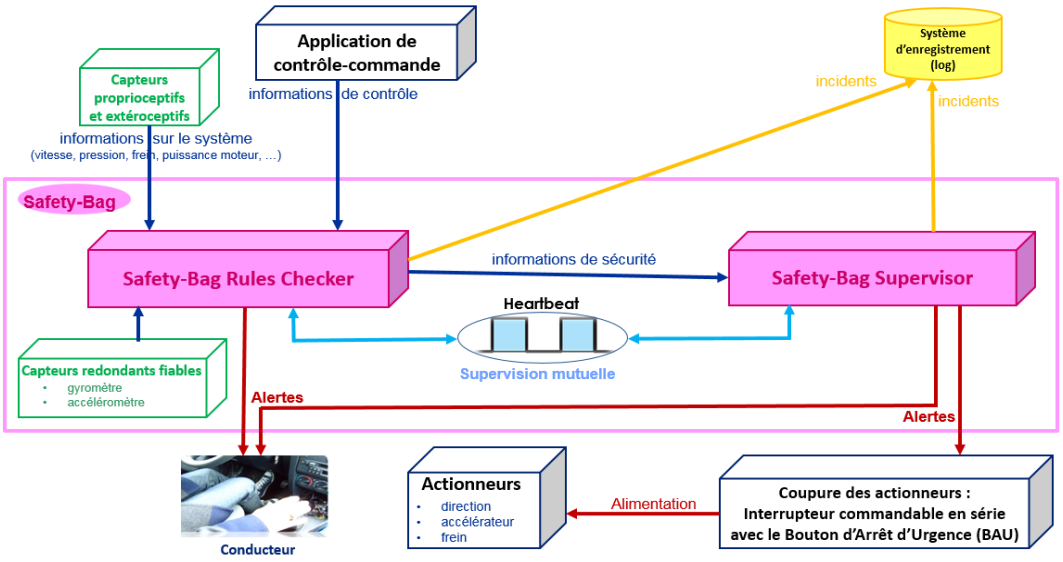

Figure 2.12 – Architecture du système étudié avec un Safety-Bag générique

2.3.2 Structure du système Safety-Bag générique

Pour réduire ses risques de défaillance, le Safety-Bag contient deux calculateurs diversifiés qui se surveillent et se supervisent entre eux : un *Safety-Bag Rules Checker* et un *Safety-Bag supervisor* (voir figure 2.12). Une défaillance simple, notamment d'un calculateur ne peut ainsi pas provoquer la défaillance de l'ensemble du Safety-Bag. Chaque calculateur peut activer les alarmes physiques s'il détecte un problème.

Les rôles de ces deux calculateurs sont les suivants :

- Le Safety-bag Rules Checker vérifie le respect des règles de sécurité-innocuité du système. Parmi celles-ci, certaines caractérisent la vivacité de l'application de contrôle-commande par une supervision des fréquences d'envoi des commandes. Il utilise les données capteurs du système tout autant que des données capteurs spécifiques en fonction des fautes que l'on veut tolérer. Par ailleurs, il vérifie également que les consignes qu'il envoie aux actionneurs ont des effets réels et cohérents.

- Le Safety-Bag Supervisor surveille de façon redondante avec le Safety-Bag Rules Checker une partie des nécessités de sécurité. Ne disposant pas de capteur propre, le Safety-Bag Supervisor surveillera de façon redondante les nécessités de sécurité liées uniquement aux données de l'application de contrôle-commande (vérification temporelle de vivacité, contrôle de vraisemblance, etc.)

Les deux calculateurs se co-surveillent par l'échange d'une supervision mutuelle.

- Si le Safety-Bag Supervisor ne reçoit plus le signal émis par le Safety-Bag Rules Checker, il désactive les actionneurs du système pour éviter que les actionneurs ne continuent à envoyer des données aberrantes. Cela a le même effet qu'un opérateur appuyant sur un Bouton d'Arrêt d'Urgence (BAU).

- Si le Safety-Bag Rules Checker n'est plus supervisé, c'est-à-dire ne reçoit plus un signal du Safety-Bag Supervisor, il demande au conducteur d'arrêter l'expérimentation même s'il n'y a pas de risque immédiat, car le Safety-Bag est maintenant vulnérable à une seule défaillance.

Chacun des calculateurs peut activer les alarmes physiques et connaître l'état du bouton d'arrêt d'urgence. En cas de défaillance détectée par le Safety-Bag, les alarmes visuelles et sonores sont levées et l'opérateur peut intervenir, ainsi que les fichiers d'enregistrement *log* permettront après l'expérience d'examiner le comportement du système.

2.4 Graphe d'états du Safety-Bag

Nous décrivons dans cette section les principaux blocs d'état du Safety-Bag dans les deux calculateurs Safety-Bag rules Checker et Safety-Bag Supervisor et nous concluons en faisant le lien avec les classes d'états définies dans [Mekki Mokhtar, 2012].

2.4.1 État global du calculateur Safety-Bag Rules Checker

Nous présentons dans la figure 2.13 un statechart décrivant l'état global du calculateur Safety-Bag Rules Checker. Nous distinguons alors trois blocs principaux :

- État *Mode autonome nominal* : Le système autonome se comporte dans ce cas de façon nominale. Le Safety-Bag Rules Checker reçoit et lit les trames venant de l'application de contrôle-commande et transmet les consignes aux actionneurs du système après vérification qu'aucune nécessité de sécurité n'est enfreinte. On a distingué dans le schéma :

  - ◇ Les nécessités de sécurité fonctionnelles dépendantes du comportement du système (vérification de contraintes dynamiques, de nécessités de sécurité propres au système, etc.) ;
  - ◇ Les nécessités de sécurité temporelles telles que la vérification de la vivacité de l'application de contrôle-commande (la mise à jour des informations envoyées par l'application de contrôle-commande), les horodatages d'informations critiques (généralement les variables de l'état perçues par le système autonome) et la cohérence temporelle de ces informations.

  Après étude des trames reçues :

  - ◇ Si les trames sont correctes et conformes aux nécessités de sécurité, le Safety-Bag Rule Checker reste dans l'état de fonctionnement normal.
  - ◇ Si le Rule Checker a identifié que des trames sont perdues, incorrectes ou incompatibles avec les nécessités de sécurité, on passe à l'état d'alerte.

- États d'*alerte* : Suivant une échelle de gravité croissante, trois types d'états sont distingués :

  - ◇ **Warning** : les commandes sont appliquées, mais l'opérateur est prévenu par des alarmes. Le fonctionnement autonome peut rester actif pendant au moins plusieurs minutes et le Safety-Bag reviendra en état de

fonctionnement nominal si la condition de sécurité est à nouveau vérifiée vraie.

On passe de l'état *mode autonome nominal* dans cet état notamment si le Safety-Bag Rules Checker détecte :

- ✓ La défaillance du superviseur
- ✓ Détection de *situations non nominales* de l'environnement telles que par exemple, une autonomie électrique très faible ou insuffisante.

◇ **Inhibitions** : le non respect de certaines nécessités de sécurité impose une intervention du Safety-Bag afin d'empêcher l'évolution vers un état catastrophique, mais cette intervention ne force pas le retour en état sûr. Ainsi, des *anomalies* dans la conduite du système telles que le non respect d'une limite posée comme condition de sécurité mène dans cet état dans lequel le Safety-Bag impose le respect de cette limite. De même, le Safety-Bag Rules Checker passe dans cet état lors de la détection de certaines *erreurs*, qui peuvent nécessiter de limiter les performances pour maintenir le système en sécurité.

Si la nécessité de sécurité causant les interventions de sécurité est à nouveau respectée, le système peut revenir vers un état nominal. Cette évolution s'effectue à travers le passage dans l'état warning. Ainsi, le Safety-Bag refera les vérifications réalisées dans cet état avant le retour en mode autonome nominal.

◇ **Actions** : Cet état regroupe les situations dans lesquelles l'application peut être incapable de contrôler le système ou de conduire le process. Ceci peut être du fait de défaillances de ses capteurs ou d'erreurs fonctionnelles comme des incohérences ou des dysfonctionnements temporels ou des problèmes de communication comme des informations de contrôle obsolètes diagnostiqués par le Safety-Bag.

Dans ces états, le Safety-Bag Rules Checker déclenche les actions nécessaires pour mettre le système en état sûr. Le Safety-Bag n'étant pas capable de contrôler le système ou de conduire le processus, cette mise en état sûr vise à arrêter le système ou à faciliter le passage dans un mode de contrôle manuel.

- ✓ Dans certains de ces états, le Safety-Bag peut constater que les nécessités de sécurité sont de nouveau vérifiées et le Safety-Bag Rules Checker sortira de l'état *actions* pour permettre le rétablissement du service. Par exemple, des pertes de messages peuvent retarder la mise à jour des consignes. Le Safety-Bag constate que ces

consignes sont obsolètes et passe dans l'état *situation rattrapable*. Après recouvrement du fonctionnement correct, il est acceptable de sortir de l'état *actions*. Dans la figure 2.13, nous avons fait le choix d'imposer un passage par l'état *inhibitions*.

✓ Dans d'autres états, la défaillance n'est pas récupérable et le retour en *fonctionnement nominal* est impossible. C'est le cas par exemple lorsque le Safety-Bag détecte une incohérence. On passe dans un état non rattrapable. Ces états sont des *états puits* pour la conduite autonome, dont la seule sortie possible et le passage en mode manuel ou l'arrêt du système.

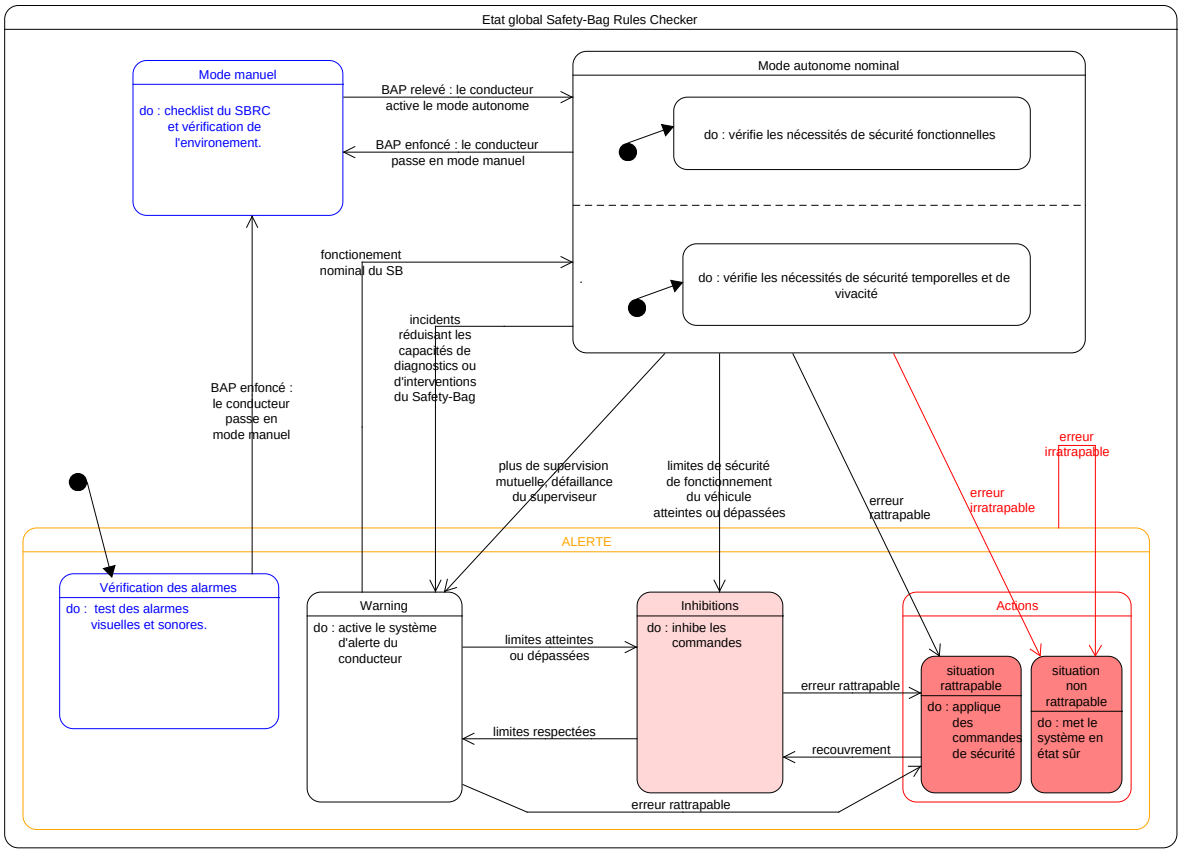

Figure 2.13 – État global du Safety-Bag Rules Checker

- État initial désigné comme *vérification des alarmes* : Au démarrage, le Safety-Bag Rules Checker est placé dans l'état *vérification des alarmes*. Cet état est intégré dans l'ensemble d'états *ALERTE* car les alertes visuelles et sonores y sont activées, ce qui permet au conducteur de vérifier leur fonctionnement avant la mise en opération du système.

  Le passage en mode autonome doit nécessiter une intervention manuelle. Pour

ce faire, un dispositif de passage en mode autonome doit être activé. Un bouton d'arrêt de process (BAP) de la forme d'un interrupteur « coup de poing » assure cette fonction. Le conducteur relève le bouton, ce qui ferme le circuit pour activer le mode autonome et l'enfonce, ce qui ouvre le circuit pour passer en mode manuel.

- État *Mode manuel* : Lorsque le conducteur enfonce le BAP, quelque soit l'état, le statechart du Safety-Bag Rules Checker passe dans l'état *mode manuel*. Dans cet état, le Safety-Bag Rules Checker peut vérifier que les pré-conditions nécessaires à la conduite autonome sont vérifiées avant d'être prêt au passage en mode autonome lorsque le conducteur relèvera le BAP.

2.4.2 État global du calculateur Safety-Bag Supervisor

Nous présentons dans la figure 2.14 l'état global du calculateur Safety-Bag Supervisor. L'automate d'état du Safety-Bag Supervisor a pour objectif de détecter des trames manquantes ou incohérentes et de superviser la vivacité du Safety-Bag Rules Checker. Il n'utilise pas de données capteurs, mais utilise seulement les valeurs fournies par le Safety-Bag Rules Checker.

Nous distinguons quatre blocs principaux :

- État *Mode autonome nominal* : dont lequel le Safety-Bag Supervisor n'a qu'un rôle de surveillance du Safety-Bag Rules Checker qui doit lui fournir les informations concernant son propre état et lui transmettre les informations que lui envoie l'application de contrôle-commande.

- État initial désigné comme *vérification des alarmes* (illustré en détails dans la figure 2.15) :

  Pour le superviseur, l'état initial est indépendant des autres blocs et on ne peut en sortir que lorsque le conducteur a enfoncé le BAP. Dans cet état, les alarmes visuelles et sonores sont activées pour permettre la vérification de leur fonctionnement par le conducteur.

- État *Mode manuel* : Le Safety-Bag Supervisor dispose d'une commande qui contrôle l'alimentation et l'amplification des signaux produits par le Safety-Bag Rules Checker vers les actionneurs du système. Cette commande est désignée par l'*interrupteur commandable*.

  Au démarrage du système, le Safety Bag Supervisor ouvre l'interrupteur commandable, empêchant toute commande autonome d'arriver aux actionneurs. Il exécute ensuite la vérification d'une checklist, comprenant entre autres

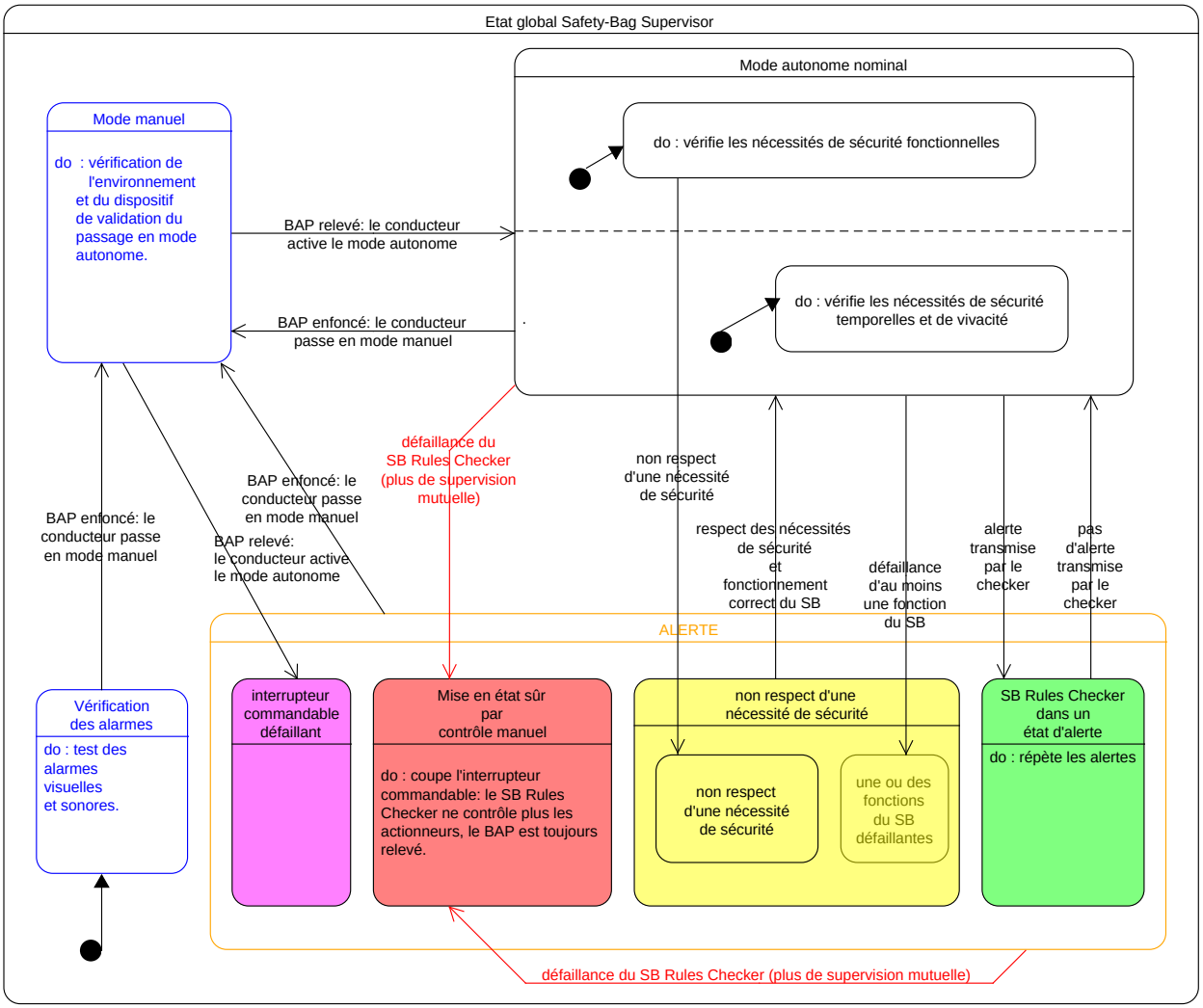

Figure 2.14 – État global du Safety-Bag Supervisor

la vérification du bon comportement de l'interrupteur commandable. Si la vérification de l'interrupteur commandable échoue, le statechart passe dans le sous état de l'état d'alerte *interrupteur commandable défaillant*. Dans cet état, les alarmes sonnent mais si l'interrupteur commandable est bloqué fermé et si le conducteur relève le BAP, le Safety-Bag Supervisor n'a plus aucun moyen d'empêcher le Safety-Bag Rules Checker de transmettre les commandes aux actionneurs. Une fois l'ensemble de la checklist vérifiée, le Safety-Bag Supervisor ferme l'interrupteur commandable pour autoriser le passage des commandes autonomes aux actionneurs, et passe en état *mode autonome nominal*.

- État d'*ALERTE* : L'état de défaillance est décomposé en quatre sous-état. En plus de l'état *interrupteur commandable défaillant* que nous avons décrit dans

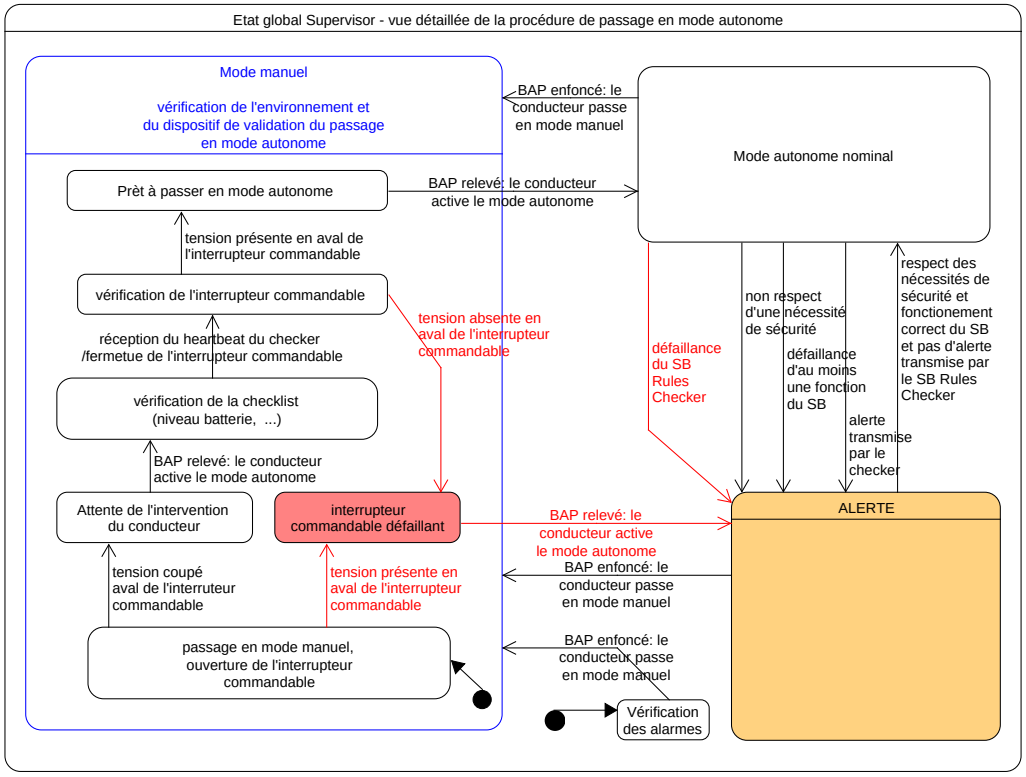

Figure 2.15 – Détails de l'état *Mode manuel* qui implémente la procédure de démarrage du Safety-Bag Supervisor

le paragraphe ci-dessus, nous avons trois autres sous états, qui sont atteints à partir de l'état de mode autonome nominal, si un problème est détecté. Ce sont :

◇ L'état *Safety-Bag Rules Checker en état d'alerte* : Dans ce cas, le Safety-Bag Supervisor n'a pas constaté lui-même la violation de la nécessité de sécurité qu'a remonté le Safety-Bag Rules Checker. Le Safety-Bag Supervisor active ses alarmes redondantes pour garantir que l'opérateur soit informé de l'incident.

◇ Dans l'état *non respect d'une nécessité de sécurité*, le Safety-Bag Supervisor a détecté la violation d'une nécessité de sécurité pour la vérification de laquelle il a été reconnu compétent lors de l'analyse. Il va alors déclencher les alertes pour informer le conducteur de la présence d'une anomalie. Il n'a cependant aucun moyen pour forcer ou inhiber une action. Les consignes appliquées aux actionneurs sont toujours transmises par le Safety-Bag Rules Checker. La vérification des nécessités de sécurité fournit une redondance pour la production des alertes destinées au conducteur. Certaines nécessités peuvent concerner le fonctionnement

du Safety-Bag lui-même et mènent alors dans le sous-état *fonctions du Safety-Bag défaillantes*.

- ◇ L'état *mise en état sûr par contrôle manuel* est déclenché lorsque le Safety-Bag Supervisor ne détecte plus le signal de vivacité (*supervision mutuelle*) provenant du Safety-Bag Rules Checker. Pour le superviseur, ceci est la défaillance la plus grave.
  - ✓ Il coupe les actionneurs en ouvrant l'interrupteur commandable. Le système est en mode manuel sans que le conducteur l'ais demandé.
  - ✓ Il alerte l'opérateur pour le prévenir de la défaillance du Safety-Bag.

En cas de défaillance du Safety-Bag Rules Checker, on arrive dans cet état que l'on soit dans le mode de fonctionnement nominal ou dans un état d'alerte.

2.4.3 Comparaison entre les états des statecharts et les états sûrs, d'alerte et catastrophiques des systèmes autonomes

Dans cette section, nous comparons les différents états des statecharts (2.13, 2.14 et 2.15) du Safety Bag avec la classification en états sûrs, dangereux ou catastrophiques de la Figure 2.2. Les états *mode autonome nominal* et les états de la *mode manuel* font partie des états sûrs.

Toutes les transitions qui entrent dans les états d'alerte correspondent à des conditions de déclenchement de sécurité et donc correspondent à des transitions qui font entrer dans les états dangereux de la frontière de la figure 2.2. L'ensemble des états d'alerte fait partie de la frontière à l'exception d'une partie des sous-états des *actions forcées mises en états sûr automatique* : les états (qui ne sont pas représentés explicitement sur le statechart) dont lesquels la mise en état sûr est terminée.

Les états catastrophiques n'apparaissent pas sur les statecharts. C'est en effet le but du Safety-Bag d'éviter ces états, et si le système s'y retrouve c'est que le Safety-Bag a failli a sa fonction.

2.5 Conclusion

Nous avons présenté dans ce chapitre les principes des composants Safety-Bag et les les concepts nécessaires pour définir les nécessités de sécurité. Ensuite, nous avons introduit plusieurs composants Safety-Bag existants dans la recherche et l'industrie et nous avons traité les méthodes de la détermination, l'expression et la vérification des nécessités de sécurité surveillées par ces dispositifs. Cette détermination fait

généralement appel à des experts du domaine et leur expression utilise souvent des langages logiques dédiés.

Nous avons ensuite présenté une architecture générique de Safety-Bag développé par des ingénieurs de recherche du laboratoire Heudiasyc. Cette architecture comporte deux calculateurs se surveillant l'un l'autre, pour éviter une défaillance sur faute unique, et se place entre l'application de contrôle commande du système et les actionneurs. Le Safety-Bag utilise des capteurs spécifiques ou ceux présents dans le système pour surveiller son état et vérifier la validité des nécessités de sécurité. En cas d'infraction, il pratique une intervention de sécurité en rejetant les commandes dangereuses ou générant ses propres commandes pour mettre le système dans un état sûr. L'expression de ces nécessités de sécurité est donc fondamental pour son bon comportement.

Dans le chapitre suivant, nous nous intéressons aux méthodes d'identification des dangers. Nous présentons ainsi un processus qui permet d'identifier les exigences et les conditions de sécurité qui permettront d'éliciter et de formaliser par la suite les nécessités de sécurité surveillées par notre système Safety-Bag.

Identification des exigences de sécurité à partir des analyses de sécurité

Sommaire

Comme nous l'avons évoqué précédemment, les véhicules autonomes sont des systèmes complexes et très hétérogènes faisant appel à de multiples capteurs, calculateurs et logiciels. Pour parer aux risques des expérimentations avec ces véhicules, nous proposons un dispositif Safety-Bag permettant d'assurer la sûreté de fonctionnement du système et en particulier sa sécurité-innocuité. Pour ce faire, nous avons recours à des mécanismes basés sur des méthodes de prévision de fautes (telles que les méthodes d'analyse de risques) et de la tolérance aux fautes (telles que la redondance). L'objectif de cette thèse est de définir un ensemble de nécessités de sécurité qui seront implémentées par le Safety-Bag. Nous dérivons cet ensemble de nécessités à partir des exigences de sécurité. Ces dernières sont le résultat de l'analyse de risques. Le système étudié est une voiture autonome expérimental, c'est-à-dire qu'un conducteur vigilant est toujours derrière le volant.

Dans ce chapitre, nous présentons l'analyse de risques effectuée dans le cadre de cette thèse. Nous spécifions un ensemble d'exigences de sécurité de façon systématique à partir des défaillances et des dangers potentiels induits par l'utilisation des composants élémentaires du système ainsi que les interactions du système avec les éléments de l'environnement routier. La détermination des exigences de sécurité est largement pratiquée dans l'industrie pour les applications critiques, notamment dans le domaine de l'automobile et dans le domaine ferroviaire. Nous nous sommes alors intéressés en particulier aux techniques permettant l'identification d'éventuels dangers. Pour cela, nous avons utilisé deux méthodes d'analyse de risques : *AMDEC* et *HazOp-UML*. L'AMDEC met l'accent sur les éléments internes du système, par contre l'HazOp-UML met l'accent sur le processus de conduite et l'environnement routier.

Dans la première section, nous présenterons le processus suivi pour analyser les risques et déterminer les exigences de sécurité ainsi que les nécessités de sécurité. Ensuite, nous présenterons l'analyse AMDEC pour les composants des véhicules autonomes. Nous étudierons ensuite l'analyse HazOp après avoir modélisé le système

en langage UML et identifié les cas d'utilisation et les attributs. Nous montrerons comment ces analyses permettent d'identifier des défaillances et des déviations possibles et comment dériver de celles-ci des exigences de sécurité dont une partie pourrait être implémentable par le Safety-Bag. Nous finirons par comparer les résultats obtenus par ces deux méthodes d'analyse de risques et montrer ainsi leur complémentarité.

3.1 Processus de détermination des exigences de sécurité

Afin d'identifier les exigences de sécurité, et donc les nécessités de sécurité par la suite, nous proposons un processus utilisant la diversification de méthodes d'analyse de risques pour obtenir le plus large spectre d'exigences de sécurité possibles. Ce processus sera présenté en détails en utilisant en exemple les techniques AMDEC et HazOp-UML. Nous définissons ensuite une échelle de niveau de gravité commune à toutes les méthodes d'analyse de risques et spécifique à notre application. Couplée à la fréquence d'occurrence du danger, cette échelle permettra de cibler les exigences de sécurité les plus importantes pour la sécurité-innocuité du système.

3.1.1 Description du processus

Nous proposons un processus de conception pour les exigences de sécurité basé sur des méthodes d'analyse de risques diversifiées, chacune étant implémentée par une équipe de sécurité différente si cela est possible. Ce processus est mis en œuvre pour deux raisons principales :

- Tout d'abord, comme les analyses de risques ne peuvent être effectuées que par des experts humains, l'utilisation de méthodes et équipes diversifiées est une technique recommandée pour réduire les erreurs et les oublis dans le résultat commun des analyses, du fait qu'une équipe ne fera pas forcément les mêmes erreurs que l'autre.

- De plus, chaque méthode d'analyse de risques se spécialise sur un aspect particulier, et l'utilisation de diverses analyses complémentaires nous permettra de produire un plus grand nombre d'exigences de sécurité portant sur tous ces différents aspects.

Dans ce manuscrit, nous avons choisi les deux méthodes d'analyse de risques AMDEC et HazOp-UML qui nous semblent les plus intéressantes afin d'identifier

respectivement les modes de défaillances potentiels des éléments internes dans notre système et les défaillances liées au processus de conduite et à l'environnement routier.

La figure 3.1 présente notre processus de conception en utilisant ces deux méthodes. D'autres méthodes complémentaires pourraient être utilisées, telles que les arbres de défaillance (AdD), qui permettent d'identifier les conséquences de défaillances multiples (contrairement à l'AMDEC) ainsi que l'analyse préliminaire des risques (APR), qui permet de déterminer les évènements redoutés pouvant provoquer des risques.

Dans l'AMDEC, nous identifions d'abord les composants et les sous-composants du système, puis nous déterminons toutes les défaillances de ces composants et ces sous-composants et leurs effets sur le système. Les exigences de sécurité sont déduites de ces informations par les experts de sécurité.

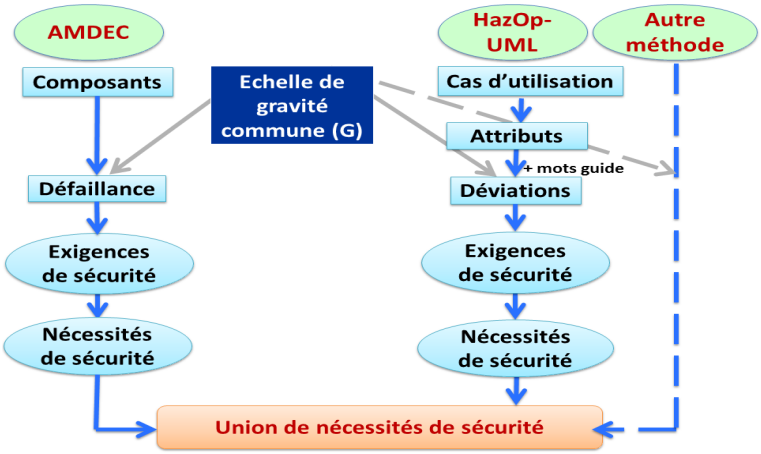

Figure 3.1 – Processus de conception pour les exigences de sécurité

L'HazOp-UML est une méthode basée sur la description du système et des acteurs pouvant interagir avec lui. Dans notre cas, les acteurs peuvent être par exemple : le conducteur, l'opérateur, les piétons, les autres véhicules, etc. L'analyse se concentre sur les processus du système et ses relations avec les différents acteurs. Tout d'abord, nous identifions un ensemble de cas d'utilisation, qui représentent les fonctionnalités du système. Pour chaque cas d'utilisation, nous déterminons ses attributs : les *pré-conditions*, les *invariants* et les *post-conditions*, qui définissent les limites fonctionnelles du cas d'utilisation (par exemple, la vitesse minimale et la vitesse maximale), mais aussi les exigences pour un comportement correct (par exemple, une entrée correcte d'un capteur particulier, ou l'absence d'un véhicule sur une voie particulière). Ensuite, nous appliquons la méthode HazOp en associant chaque attribut à une liste de mots guide pour guider le processus et identifier touts les dangers potentiels qui sont appelés *déviations*. Enfin, nous identifions les exigences de sécurité de chaque déviation. Les exigences de sécurité des deux

N°	Échelle de gravité
0	Fonctionnement nominal
1	Arrêt de la conduite autonome : arrêt automatique et/ou nécessité de reprise en manuel en 10s
2	Perte de contrôle du véhicule à moins d'une reprise en main par le conducteur en 4 à 5 secondes avec alerte
3a	Perte de contrôle du véhicule à moins d'une reprise en main immédiate par le conducteur en 2 secondes avec alerte ⇒ Les conditions de reprise en main sont favorables et rendent l'intervention du conducteur facile
3b	Perte de contrôle du véhicule à moins d'une reprise en main immédiate par le conducteur en 2 secondes avec alerte ⇒ Les conditions de reprise en main sont défavorables et rendent l'intervention du conducteur difficile s'il réagit immédiatement
4	Perte de contrôle du véhicule à moins d'une reprise en main immédiate malgré l'absence d'alerte ⇒ Les conditions permettent au conducteur de limiter les conséquences de la perte de contrôle
5	Perte de contrôle du véhicule irrattrapable par le conducteur

Tableau 3.1 – Echelle de gravité.

analyses sont raffinées en nécessités de sécurité, implémentables par le Safety-Bag. Les nécessités dérivées de chaque méthode sont rassemblées, et les nécessités équivalentes identifiées et éliminées. Ce raffinement des exigences en nécessités et leur fusion sont présentés en détails dans le chapitre 5.

3.1.2 Échelle de gravité

Nous présentons dans ce paragraphe la première étape d'analyse de risques qui consiste en la définition d'une table de niveaux de gravité (tableau 3.1). Chaque défaillance ou déviation sera caractérisée par un niveau de cette échelle. Celle-ci servira ainsi de guide pour la réduction des risques en permettant de se concentrer sur les défaillances ou déviations les plus dangereuses pour le système et son environnement.

- Le niveau 0 : correspond au fonctionnement nominal du système dans lequel le logiciel est capable de conduire correctement le véhicule tandis que le matériel applique correctement les décisions prises par le logiciel. Les autres niveaux nécessitent une intervention du conducteur et l'arrêt de l'expérimentation.

- Le niveau 1 (limité) : caractérise soit un comportement aux limites du système (vitesse/dynamique limite, obstacles évités de peu) ou un problème qui nécessite une reprise en main dans un délai confortable (par exemple la caméra tombe en panne, mais le lidar fonctionne, permettant encore de détecter les obstacles immédiats).

- Le niveau 2 (mineur) : correspond à un comportement inadéquat mais ne risquant pas d'avoir des conséquences catastrophiques immédiates : le conducteur a plusieurs secondes pour réagir. Par exemple, un dépassement de la vitesse adaptée, ou une trajectoire s'écartant lentement du centre de la

voie. Une défaillance de niveau de gravité 2 nécessite une intervention rapide du conducteur suite au déclenchement d'alertes et la possibilité de son occurrence impose une supervision vigilante de la part du conducteur.

- Le niveau 3 (majeur) : correspond à un comportement dangereux qui peut amener à une conséquence catastrophique en quelques secondes, mais qui est rattrapable par le conducteur. Au niveau 3, ce comportement dangereux a été détecté et des alarmes ont été levées pour amener le conducteur à réagir. Le niveau 3 peut encore être décomposé en deux sous-niveaux :
  - ◇ Niveau 3a : qui correspond à une évolution continue de la situation de conduite. Par exemple une trajectoire qui sort de la route, ou des commandes maintenues risquant de provoquer une collision ou une perte de contrôle. Le conducteur peut reprendre en main le véhicule s'il réagit à temps.
  - ◇ Niveau 3b : qui correspond à une évolution brutale de la situation de conduite, par exemple une brusque modification de l'angle au volant, ou une accélération soudaine. Bien que des alertes soient levées, le changement brutal de dynamique du véhicule peut rendre la reprise en main par le conducteur difficile.
- Le niveau 4 (sérieux) : correspond à une situation de conduite dangereuse qui peut amener à une conséquence catastrophique en quelques secondes, mais qui est rattrapable par le conducteur. Au niveau 4, les alarmes ne sont pas déclenchées car la situation situation dangereuse n'est pas détectée ou car le système d'alarmes est défaillance. Le conducteur doit s'apercevoir lui-même du problème avant de réagir, ce qui rend improbable une reprise en main avant une conséquence catastrophique.
- Le niveau 5 (catastrophique) : correspond à des défaillances irrattrapables par le conducteur, entrainant des commandes aberrantes de rotation du volant ou d'accélération causant une perte de contrôle du véhicule quasi-immédiate. Dans ce niveau, le conducteur peut ne pas avoir le temps de réagir avant qu'une conséquence catastrophique ne se produise. Même en cas de déclenchement d'alertes, cette défaillance reste toujours irrattrapable par le conducteur.

3.1.3 Discussion

Comme nous l'avons dit précédemment, le processus de conception dédié à la détermination des exigences de sécurité et ensuite des nécessités de sécurité n'est pas

limité par l'utilisation des deux méthodes d'analyse de risques AMDEC et HazOp-UML. Nous pouvons également utiliser par exemple la méthode d'analyse par l'arbre de défaillances, qui peut compléter ces deux analyses pour identifier les défaillances multiples dangereuses.

Notons également que l'échelle de gravité doit être adaptée au système étudié. Des niveaux peuvent être ajoutés ou modifiés en fonction du système.

3.2 AMDEC pour les véhicules autonomes

La méthode d'analyse AMDEC (ou Analyse des Modes de Défaillances, de leurs Effets et leur Criticité) est une méthode ascendante (*bottom-up*), qui permet d'identifier et de déterminer les conséquences des types de défaillances (matérielles, logicielles, etc.) des composants du système. Des exigences de sécurité cherchant à traiter ces défaillances peuvent être extraites de l'analyse par des experts de sécurité ayant des connaissances suffisantes sur l'architecture du système étudié.

Cette analyse se présente ainsi sous la forme d'un tableau détaillant chaque composant du système, ses différents types de défaillances et les effets associés, y compris leur gravité selon l'échelle de gravité définie dans le tableau 3.1.

Le tableau AMDEC comporte aussi le taux de défaillance de chaque composant (qui permet avec la gravité d'identifier les défaillances les plus risquées pour le système), les moyens de détection, d'action ou de correction et les exigences de sécurité qui nous permettront par la suite d'éliciter les nécessités de sécurité si et seulement si l'exigence de sécurité est implémentable par le dispositif Safety-Bag.

Afin d'effectuer une telle étude AMDEC, il faut tout d'abord avoir des connaissances suffisantes sur l'architecture et les composants du système. Pour ce faire, nous allons commencer par présenter l'architecture d'un véhicule autonome expérimental développé dans notre laboratoire. Ensuite, nous allons préciser la décomposition du tableau AMDEC et nous allons montrer comment nous avons traité chaque colonne jusqu'à l'identification de l'exigence de sécurité. Finalement, un récapitulatif sera présenté pour résumer cette étude.

3.2.1 Architecture des véhicules autonomes

Nous présentons dans ce paragraphe l'architecture de notre véhicule autonome, de type Renault-Fluence. Cette architecture est présentée sous forme d'un diagramme de déploiement UML (voir figure 3.2). Nous distinguons :

- l'application de contrôle-commande,

- les capteurs proprioceptifs (fournissant la vitesse, l'accélération, la position du volant, etc.), qui permettent de connaitre l'état du véhicule,
- les capteurs extéroceptifs (caméra lidars, radars, GPS, etc.), qui permettent d'obtenir des informations sur l'environnement.
- les actionneurs (accélérateur, frein et volant).

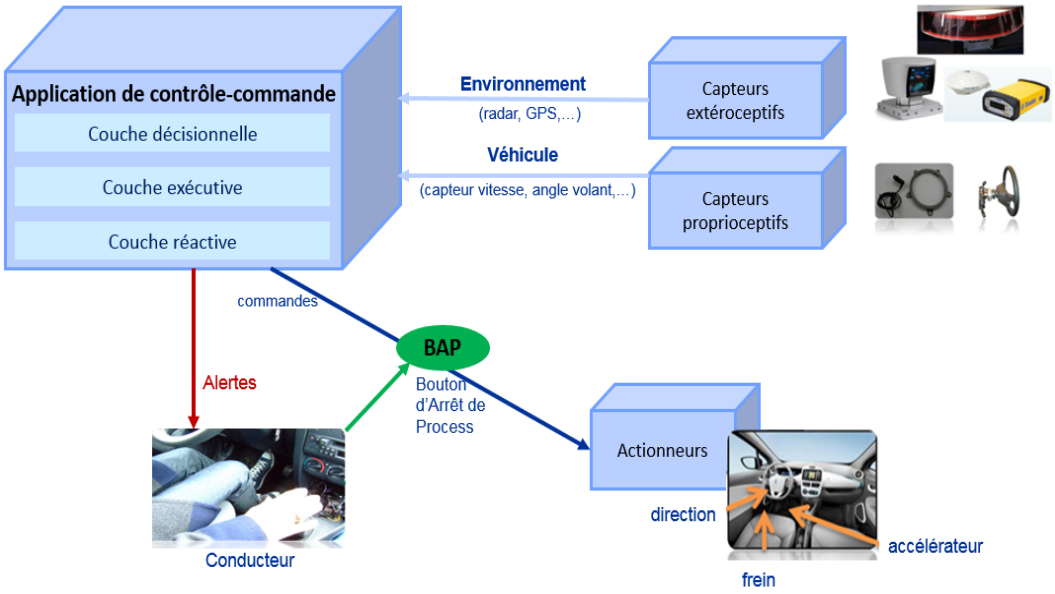

Figure 3.2 – Architecture de notre véhicule autonome expérimental

Nous avons mis en avant sur le schéma entre l'application de contrôle-commande et les actionneurs du véhicule (accélérateur, frein et volant) un Bouton d'Arrêt de Process (BAP), qui permet au conducteur de neutraliser la commande automatique pour reprendre le contrôle en manuel. On voit également qu'une alerte peut être émise à destination du conducteur. Ainsi que noté dans l'échelle de gravité, cette alerte peut être fondamentale pour le conducteur car elle le dispense de s'apercevoir par lui-même que l'application ne fonctionne plus correctement. Ceci peut lui faire gagner plusieurs secondes, cruciales pour la conduite d'un véhicule de forte puissance. Cependant, elle n'est vraiment utile que si une erreur a été détectée et remontée par un mécanisme du système. Il est important de noter que nous ne disposons pas dans ce véhicule d'application de contrôle-commande. Elle sera simulée dans nos expériences par un algorithme suivant une trajectoire définie au préalable.

3.2.2 Décomposition

Nous présentons dans cette section la matrice AMDEC (présentée dans les tableaux 3.2, 3.3, 3.4 et 3.5) des véhicules autonomes expérimentaux que nous étudions.

Chaque ligne de cette matrice comporte les éléments suivants :

- le composant défaillant considéré ; ce composant peut être un composant élémentaire (comme un calculateur particulier), ou un sous-composant de niveau d'abstraction plus élevé (comme l'application de contrôle commande),
- une défaillance possible du composant considéré ; toutes les défaillances du composant doivent être considérées,
- les effets informatiques logiciels, qui sont les conséquences sur le fonctionnement du système autonome,
- les effets sur le véhicule et sa trajectoire,
- le taux de défaillance $\lambda(h^{-1})$, c'est-à-dire la probabilité de défaillance par unité de temps (généralement l'heure), qui est l'inverse du MTBF (*Mean Time Between Failures*),
- le moyen de rétablissement : le conducteur est toujours présent pour reprendre le contrôle du véhicule, mais des systèmes automatiques de rétablissement du système peuvent être utilisés.
- le temps d'action (t) : le délai dans lequel le moyen d'action intervient pour ramener le véhicule dans un état sûr,
- le moyen de détection : le moyen peut être un mécanisme automatique ou humain,
- le temps de détection : le délai dans lequel le moyen de détection identifie la défaillance. S'il s'agit d'un dispositif de diagnostic, c'est le délai pour déclencher l'alerte. Si c'est un humain, c'est le délai estimé pour que celui-ci identifie la situation comme dangereuse et décide d'intervenir.
- la gravité des conséquences : l'indice dans l'échelle de gravité
- l'exigence de sécurité identifiée pour contrer cette défaillance.

3.2.3 Analyse AMDEC et identification des exigences de sécurité

Les tableaux (3.2, 3.3, 3.4 et 3.5) représentent une étude complète de l'analyse AMDEC du véhicule autonome expérimental dans notre laboratoire sans Safety-Bag.

Ces tableaux correspondent à 11 éléments et à leurs types de défaillances suivants :

- défaillance des composants matériels de l'application de contrôle-commande (ou AppCC) : ce type de défaillance comporte la défaillance du calculateur sur lequel fonctionne l'application de contrôle commande, pouvant causer des défaillances de gel de l'application avec des sorties bloquées, ou d'arrêt du calculateur (et donc de l'application) avec des sorties mises à zéro. Ce type de défaillance comporte aussi les défaillances d'entrée/sortie de l'application de contrôle-commande, c'est-à-dire l'écran et le clavier utilisé pour contrôler la conduite et/ou l'expérience.

- défaillance des composants logiciels de l'application de contrôle-commande : ce type de défaillance comporte un gel de l'application, amenant des commandes bloquées en sortie, ou des défaillances fonctionnelles soit en valeur (commandes aberrantes) soit temporelles (commandes envoyées en retard ou en avance),

- défaillance du réseau CAN : ce type de défaillance peut être causé par un problème de connectique (fil coupé ou déconnecté) qui va amener l'absence de transmission des messages sur le réseau, une altération des messages transmis (par exemple causée par des problèmes de compatibilité électromagnétique) ou une surcharge du réseau causant des collisions et la perte de certains messages.

- défaillance du convertisseur Numérique/Analogique (D/A) : ce type de défaillance peut amener des sorties de commandes bloquées si la carte du convertisseur est gelée, ou la mise à zéro des commandes dans certains cas de défaillance logicielle du pilote de la carte du convertisseur.

- défaillance du capteur angle volant : ce type de défaillance cause des valeurs de sortie aberrantes sur le capteur d'angle au volant, risquant d'amener un comportement défaillant de l'application de contrôle commande.

- défaillance des capteurs de vitesse et d'accélération longitudinales : de la même façon que précédemment, la défaillance de ces capteurs peut amener un comportement aberrant de l'application de contrôle commande.

- défaillance des capteurs proprioceptifs latéraux du véhicule (l'accéléromètre latéral et le gyromètre) : ce type de défaillance est comparable aux deux précédents.

- défaillance des capteurs extéroceptifs liés à l'environnement (tels que les lidars, les radars, les caméras, le GPS, etc.) : comme pour les capteurs proprioceptifs,

la défaillance des capteurs extéroceptifs peut amener un comportement aberrant de l'application de contrôle commande : non-détection d'obstacle, mauvaise identification des limites navigables du véhicule, localisation erronée, etc.

- défaillance de l'actionneur « *accélérateur* », : ce type de défaillance couvre un actionneur bloqué fermé (accélération nulle) et bloqué ouvert (accélération bloquée à une commande), typiques des actionneurs mécaniques,
- défaillance de l'actionneur « *frein* » : ce type de défaillance entraîne l'impossibilité de freiner, soit à cause d'une défaillance du moteur électrique du frein, soit d'un problème de connectique.
- défaillance de l'actionneur « *assistance de direction* » : ce type de défaillance cause l'impossibilité de contrôler latéralement le véhicule par moyens logiciels, causée par une défaillance de la direction assistée électrique (DAE). Notons que le conducteur peut toujours contrôler manuellement le véhicule latéralement en tournant le volant.

Notons que des défaillances des composants matériels (1a, 2a et 3a dans 3.2) ou des composants logiciels (4a et 5a dans 3.2) de l'application de contrôle-commande risquent d'amener des tensions de commandes incorrectes sur l'accélérateur et le volant, pouvant conduire très rapidement le véhicule dans un état où toute tentative de reprise en main (par un conducteur ou un rétablissement automatique) est illusoire. De plus, la perte des informations de vitesse (12a dans 3.4) et d'angle volant (11a dans 3.3) peut, en l'absence de diagnostic par l'application de contrôle-commande, amener une application à commander une trajectoire aberrante. Ces défaillances ont ainsi dans le pire cas (toujours considéré dans des analyses de sécurité-innocuité) une gravité de niveau 5. Ce sont des résultats inacceptables, considérant en particulier que certaines défaillances, notamment sur le logiciel, peuvent être assez fréquentes. Dans le tableau 3.5, nous remarquons par exemple qu'une panne de direction assistée se traduira par la désactivation d'assistance et peut avoir une gravité 4 puisque le véhicule ne suit plus sa trajectoire et va tout droit tandis que le conducteur n'est pas prévenu.

Cette étude met en évidence qu'un tel véhicule a de nombreux modes de défaillances induisant des niveaux de risques élevés et que ces évènements ont des probabilités d'occurrence significatives. Le taux de défaillance est difficile à obtenir sans effectuer de nombreuses expérimentations et dans la plupart des cas, nous ne connaissons pas leurs valeurs exactes. Les valeurs indiquées dans le tableau sont ainsi des valeurs empiriques obtenues à partir de ce que nous savons du système et

de ses composants. Nous les avons prises comme des ordres de grandeurs en ce qui concerne les valeurs de défaillances matérielles. Pour les défaillances logicielles, les composants expérimentaux utilisés sont développés dans un contexte de recherche et ne pourront être testés que sommairement avant d'être intégrés dans le système complet. Nous avons donné quelques estimations d'environ 10^{-3} ou 10^{-4} par heure. La pratique nous conduit à croire qu'ils sont significativement élevés.

Nous avons décidé de mettre des temps de détection et d'action d'un conducteur variables selon la difficulté de la détection et la complexité des actions à mettre en œuvre selon notre point de vue. Ces temps varient de 2 secondes (c'est le temps reconnu par le code de la route pour les détections et les actions simples) à 4 secondes si l'analyse de la situation est plus complexe ou si l'action est élaborée.

A partir de cette AMDEC, nous avons réussi à identifier les 11 exigences de sécurité (parmi les 18 types de défaillance définis) suivantes :

1. Exigence de sécurité 1 : « *Le système doit être capable de détecter les défaillances de gel ou d'arrêt inopiné de l'application de contrôle-commande et de mettre le système en état sûr au besoin (1a, 2a, 4a, 9a, 10a et 11a).* »

   Cette exigence correspond aux défaillances matérielles du calculateur de l'application de contrôle commande, à la défaillance logicielle de gel de l'application de contrôle-commande, à la défaillance de la carte du convertisseur numérique/analogique (D/A) et à la défaillance du capteur couple volant.

2. Exigence de sécurité 2 : « *Le conducteur doit pouvoir reprendre la conduite en manuel par un autre moyen que les interfaces de l'application de contrôle commande (3a).* »

   Notre système dispose d'un bouton d'arrêt de process permettant de repasser immédiatement en mode manuel, ce qui répond à cette exigence. De manière générale, le passage en mode manuel de cette façon reste un mode de rétablissement ultime pour toutes les défaillances de notre véhicule expérimental. Comme précisé dans l'échelle de gravité, il n'est cependant pas assuré que le conducteur puisse reprendre le contrôle dans toutes les situations.

3. Exigence de sécurité 3 : « *Le système doit détecter les erreurs fonctionnelles de l'application de contrôle-commande, telles que la désynchronisation des données, des commandes aberrantes, des erreurs de décision, et des mauvaises interprétations. En cas d'erreur, il faut assurer la mise en sécurité du système en levant des alarmes et en repassant en mode manuel (5a).* »

   Cette exigence de sécurité couvre toutes les défaillances fonctionnelles du logiciel, et demandera à être raffinée pour être traitée par différentes nécessités

de sécurité.

4. Exigence de sécurité 4 : « *Le système doit détecter et traiter les défaillances de communications sur le bus CAN : arrêt des communications ou pertes de données (6a et 8a).* »

5. Exigence de sécurité 5 : « *Des vérifications de cohérence entre les données CAN et les données de capteurs redondants du Safety-Bag pourraient être réalisées (7a).* »

   Cette exigence de sécurité correspond au cas d'altération des données transmises par le bus CAN.

6. Exigence de sécurité 6 : « *Le système doit vérifier que les capteurs proprioceptifs longitudinaux (vitesse et accélération) fonctionnent correctement (12a).* »

7. Exigence de sécurité 7 : « *Le système doit vérifier que les capteurs proprioceptifs latéraux (accéléromètre latéral et gyromètre) fonctionnent correctement (13a).* »

8. Exigence de sécurité 8 : « *Le système doit vérifier que les capteurs extéroceptifs ne sont pas défaillants (14a).* »

   En pratique, les techniques de robustesse telles que la fusion de données permettent de tolérer certaines fautes de capteurs, mais elles ne garantissent généralement pas la tolérance à toutes les fautes possibles, et ne sont donc pas suffisantes.

9. Exigence de sécurité 9 : « *Le système doit détecter les incohérences entre la commande d'accélération produite par l'application de contrôle-commande et la commande d'accélération effectivement réalisée par le moteur du véhicule (19a).* »

10. Exigence de sécurité 10 : « *Le système doit détecter les incohérences entre la commande de freinage produite par l'application de contrôle-commande et la commande effectivement réalisée par les freins du véhicule (20a).* »

11. Exigence de sécurité 11 : « *Le système doit détecter les incohérences entre la commande de direction produite par l'application de contrôle-commande et la commande effectivement réalisée par l'assistance à la direction (21a).* »

    Notons que pour un véhicule commercial, le moteur de direction doit être redondant en cas de défaillance d'un des moteurs. Ce n'est pas le cas pour notre véhicule, mais considérant son utilisation expérimentale, il faut au minimum

que le conducteur soit alerté de la défaillance pour reprendre la conduite en manuel.

Ces exigences de sécurité sont définies d'une manière informelle et seront implantées soit par notre système de sécurité Safety-Bag soit par un autre moyen, comme la redondance de composant pour des défaillances matérielles.

Nous rappelons que dans le cas où le moyen d'implémentation de l'exigence de sécurité est le Safety-Bag, une nécessité de sécurité sera définie à partir de l'exigence de sécurité ciblée.

Tableau 3.2 – AMDEC des composants véhicules autonomes (page1)

Élément	**Type de Défaillance**	**Effets Informatiques**	**Effets Véhicule**	$\lambda(h^{-1})$	**Détection**		**Action**		**Gravité des Conséquences**		**Exigence de Sécurité**	**Réf**
					moyen	t	moyen	t	G	Commentaires		
Matériel de l'AppCC	Panne bloquée	les sorties de convertisseurs sont maintenues	Pas de contrôle : L'accélération, le frein et le couple volant sont maintenus.	$\sim 10^{-3}$	conducteur	> 2s	conducteur	$+2$s	5	le véhicule est incontrôlable	Le système doit être capable de détecter les défaillances de gel ou d'arrêt inopiné de l'application de contrôle-commande et de mettre le système en état sûr au besoin.	1a
	Hors Tension	Les sorties de convertisseur restent à 0.	• pas d'accélération • pas de freinage autonome • la DAE est en défaillance	$\sim 10^{-5}$	conducteur	> 2s	conducteur et BAP	$+2$s	4	–	idem 1a	2a
	Clavier/ écran	Pertes des interactions par l'opérateur.	–	$\sim 10^{-5}$	opérateur	> 4s	conducteur et BAP	$+2$s	1	–	Le conducteur doit pouvoir reprendre la conduite en manuel par un autre moyen que les interfaces de l'application de contrôle-commande.	3a
Logiciel de l'AppCC	blocage du logiciel	idem 1a		$\sim 10^{-3}$	idem 1a							4a
	Erreur Fonctionnelle	Commandes aberrantes en valeur ou temporelles	comportement imprévisible	$\sim 10^{-3}$	conducteur	4s	conducteur	$+2$s	1 à 5	Les erreurs fonctionnelles peuvent causer une grande variété de dysfonctionnements qui seront très difficiles à détecter assez tôt pour que l'intervention du conducteur évite un accident.	Le système doit détecter les erreurs fonctionnelles de l'application de contrôle-commande, telles que la désynchronisation des données, des commandes aberrantes, des erreurs de décision, et des mauvaises interprétations. En cas d'erreur, il faut assurer la mise en sécurité du système en levant des alarmes et en repassant en mode manuel.	5a

Tableau 3.3 – AMDEC des composants véhicules autonomes (page 2)

Élément	**Type de Défaillance**	**Effets Informatiques**	**Effets Véhicule**	$\lambda(h^{-1})$	**Détection**		**Action**		**Gravité des Conséquences**		**Exigence de Sécurité**	**Réf**
					moyen	t	moyen	t	G	Commentaires		
CAN (défaillance matérielle)	déconnecté ou fil coupé	perte d'une partie des communications du système embarqué	Plusieurs alertes / déconnexion de la puissance de freinage et des systèmes de direction.	$\sim 10^{-9}$	système embarqué du véhicule	< 1s	conducteur	2 à 4s	3b	Le contrôle autonome n'est soudainement pas activé et les alertes sont envoyées au conducteur, mais, ce dernier récupère potentiellement le contrôle sans assistance au freinage et à la direction.	Le système doit détecter et traiter les défaillances de communications sur le bus CAN : arrêt des communications ou les pertes de données.	6a
	altération des données transmises	large type de dysfonctionnements des parties du système embarqué du véhicule	• valeur aberrante possible pour les données d'angle volant et de la vitesse et d'accélération latérales et longitudinales • de ce fait, les commandes générées à partir de ces données par l'AppCC peuvent également être aberrantes.	$\sim 10^{-10}$	conducteur	4s	conducteur	2 à 4s	5	–	Des vérifications de cohérences entre les données CAN et les données de capteurs redondants du Safety-Bag pourraient être réalisées.	7a
	CAN surchargé		idem 6a									8a
convertisseur D/A (défaillance matérielle)	carte bloquée	idem 1a										9a
	défaillance logicielle du pilote de la carte	idem 2a										10a
capteur couple volant (défaillance matérielle)	capteur défaillant	commande aberrante	pas d'alerte, trajectoire aberrante ou incontrôlable	$\sim 10^{-5}$	conducteur	4s	conducteur + BAP	+2s	5	Il est probable que le pilote ne puisse pas reprendre le contrôle.	idem 1a	11a

Tableau 3.4 – AMDEC des composants véhicules autonomes (page 3)

Élément	**Type de Défaillance**	**Effets Informatiques**	**Effets Véhicule**	$\lambda(h^{-1})$	**Détection**		**Action**		**Gravité des Conséquences**		**Exigence de Sécurité**	**Réf**
					moyen	t	moyen	t	G	Commentaires		
capteur de vitesse (défaillance matérielle)	capteur défaillant	commande dangereuse	pas d'alerte et risque de vitesse excessive	$\sim 10^{-5}$	conducteur	4s	conducteur	2s	5	vitesse non adaptée à la situation et peut être excessive	Le système doit vérifier que les capteurs proprioceptifs longitudinaux (vitesse et accélération) fonctionnent correctement.	12a
capteurs proprioceptifs (gyromètre + accéléromètre) (défaillance matérielle)	capteur défaillant	résultats incorrects	trajectoire non adaptée	$\sim 10^{-5}$	trajectoire aberrante	4s	contrôle manuel difficile	+2s	3a à 5	La gravité dépend de la réalisation de l'application, mais aussi de la dynamique du véhicule pendant l'expérimentation.	Le système doit vérifier que les capteurs proprioceptifs latéraux (accéléromètre latéral et gyromètre) fonctionnent correctement.	13a
capteurs extéroceptifs (défaillance matérielle)	capteur ou connectivité	mauvaise détection	contrôle incorrect mais dynamique limitée	_	conducteur	> 4s	conducteur	+2s	3a à 5	L'environnement n'est pas détecté (route, véhicules, piétons, etc.)	Le système doit vérifier que les capteurs extéroceptifs ne sont pas défaillants.	14a

Tableau 3.5 – AMDEC des composants véhicules autonomes (page 4)

Élément	**Type de Défaillance**	**Effets Informatiques**	**Effets Véhicule**	$\lambda(h^{-1})$	**Détection**		**Action**		**Gravité des Conséquences**		**Exigence de Sécurité**	**Réf**
					moyen	t	moyen	t	G	Commentaires		
actionneur « accéléra-teur » (dé-faillance matérielle)	bloqué à 0	_	Pas d'accélération	$\sim 10^{-5}$	conducteur	2s	conducteur	+2s	2	Ce genre d'incident s'est produit en testant notre véhicule.	Le système doit détecter les incohérences entre la commande d'accélération produite par l'application de contrôle-commande et la commande d'accélération effectivement réalisée par le moteur du véhicule.	15a
	verrouillé à une valeur	_	accélération sans demande	idem 1a					5	idem 1a		
actionneur « frein » (défaillance matérielle)	moteur électrique ou fil utilisé pour activer les freins	_	pas de freinage	$\sim 10^{-5}$	conducteur	2s	conducteur	+2s	5	Notre solution pour actionner les freins n'est pas aussi fiable que les systèmes de l'industrie automobile.	Le système doit détecter les incohérences entre la commande de freinage produite par l'application de contrôle-commande et la commande effectivement réalisé par les freins du véhicule.	16a
actionneur « assistance de direction » (défaillance matérielle)	défaillance de la direction assistée électrique	_	pas de direction	$\sim 10^{-5}$	conducteur	2s	conducteur	+2s	4	La DAE est approuvée pour un véhicule à conduite humaine mais n'a pas un niveau de fiabilité adapté à l'actionneur d'un véhicule commercial autonome.	Le système doit détecter les incohérences entre la commande de direction produite par l'application de contrôle-commande et la commande effectivement réalisée par l'assistance de direction.	17a

3.2.4 Récapitulatif

Le tableau 3.3 récapitule les résultats trouvés en effectuant l'analyse AMDEC pour les véhicules autonomes. L'approche AMDEC pour les composants des véhicules autonomes nous a permis d'identifier 18 types de défaillances parmi 11 composants dans le système et de spécifier 11 exigences de sécurité. Le niveau de gravité associé à la majorité des composants élémentaires du système est 5. Ce niveau correspond à des défaillances provoquant des commandes aberrantes, scénarios dans lesquels la perte du contrôle du véhicule peut être quasiment inévitable, même avec une reprise en main rapide. Un tel niveau de gravité aux taux de défaillances estimés est inacceptable pour un véhicule industrialisé ou expérimental.

AMDEC pour les véhicules autonomes	
Nombre de composants	*11*
Nombre de types de défaillances	*18*
Gravité maximale	*5*
Nombre d'exigences de sécurité	*11*

Figure 3.3 – Synthèse de l'analyse AMDEC des véhicules autonomes

3.3 Étude de danger et d'opérabilité (HazOp-UML)

Complémentairement à la méthode AMDEC qui vise à identifier toutes les défaillances possibles liées aux composants du système, la méthode d'analyse de risques HazOp-UML cherche à identifier les différents risques pour la sécurité-innocuité liés à l'environnement routier et le processus de conduite. La méthode HazOp permet d'identifier systématiquement les risques possibles du système en appliquant à chaque attribut une liste de mots guide produisant les déviations.

HazOp-UML est une extension de la méthode HazOp utilisant des diagrammes UML (Unified Modeling Language) [[Guiochet, 2016], [Martin-Guillerez et al., 2010]] pour étudier le comportement du système et ses interactions avec d'autres acteurs de l'environnement, notamment les autres usagers, les obstacles, etc. Cette étude utilise en particulier des cas d'utilisations et a généralement lieu au début de la conception du processus avant la phase de développement.

3.3.1 L'analyse de risque HazOp-UML

Dans cette partie, nous allons étudier la méthode HazOp-UML, extension de la méthode HazOp. Nous allons commencer par déterminer les cas d'utilisation poten-

tiels et mettre en avant leurs interactions avec les composants de l'environnement. Ensuite, nous allons associer à chaque cas d'utilisation un ensemble d'attributs afin de pouvoir éliciter les déviations possibles par leur association avec les mots clés HazOp.

3.3.1.1 Cas d'utilisation

Nous rappelons que les cas d'utilisation énumèrent toutes les interactions envisageables entre le système et son environnement extérieur. Nous avons défini les acteurs suivants (c'est-à-dire les entités externes interagissant avec le système) : l'opérateur, le conducteur, l'infrastructure communicante, les piétons, et les autres véhicules.

Notons que dans les acteurs des figures 3.4 et 3.5, nous différencions conducteur et opérateur. Le conducteur est la personne qui possède l'accès aux commandes de contrôle du véhicule (accélérateur, frein et volant). Il est capable de passer en mode autonome, ou en mode manuel en cas d'anomalie ou de situation dangereuse. Il peut suivre les alertes et reprendre la main sur le véhicule dans un certain délai et doit pour cela connaître l'état du véhicule autonome et de son environnement.

Par contre, l'opérateur est celui qui supervise l'état du véhicule autonome et est capable d'interagir avec l'application de contrôle-commande, par exemple pour générer un itinéraire ou gérer une trajectoire cinématique(comme le montre la figure 3.5). L'opérateur peut dans certains cas être la même personne que le conducteur : c'est le cas pour un véhicule autonome commercialisé, mais pas forcément pour nos véhicules autonomes expérimentaux.

Pour générer un itinéraire, l'opérateur fournit une destination finale au système, qui va générer un itinéraire permettant d'aller de sa position initiale jusqu'à cette destination (par exemple de la gare de Compiègne à la gare de Paris Nord). Un itinéraire est constitué d'une succession d'étapes, chacune correspondant à une section de route que doit parcourir le véhicule pour atteindre sa destination finale. Une étape est achevée à chaque fois que le véhicule arrive à un point de croisement, un rond-point, une intersection, une sortie ou un autre point particulier. Une étape est ainsi définie par son origine et par son objectif, chacun étant un point de repère (généralement une intersection) sur la même route de la carte routière. La première origine d'étape dans un itinéraire est le point de départ de l'itinéraire, et l'objectif d'étape de la dernière étape de l'itinéraire considéré est la destination de l'itinéraire. Typiquement, une étape a une longueur de quelques centaines de mètres en ville, et quelques dizaines de kilomètres sur autoroute.

Afin d'identifier un ensemble de *cas d'utilisation* pertinent pour effectuer une analyse HazOp d'un véhicule autonome, nous nous sommes appuyés sur la liste de

conditions citée pages 28 et 29 du document [NHTSA] [NHTSA, 2016], en attendant que les autorités européennes produisent des recommandations analogues.

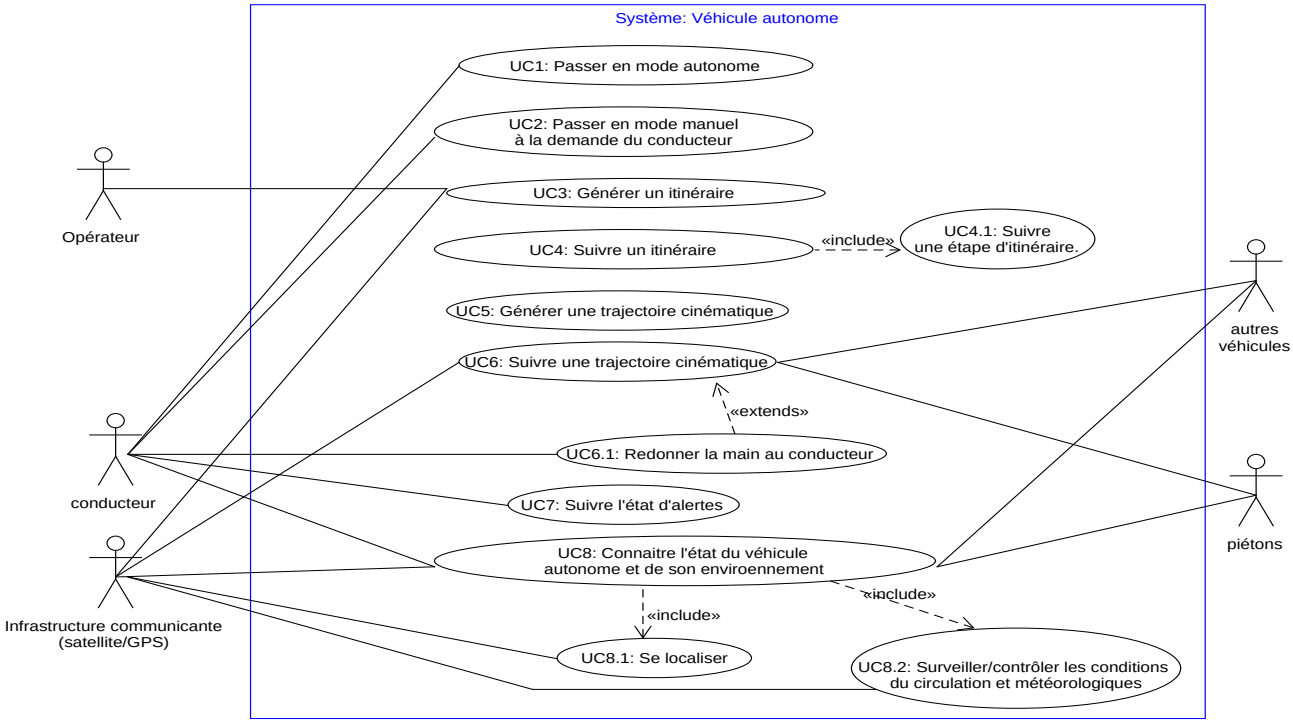

Figure 3.4 – Schéma UML des cas d'utilisation de notre véhicule autonome

Des 28 situations décrites dans ce document, nous avons retiré au total 26 cas d'utilisation comme le montrent les figures 3.4 et 3.5 : *8* cas d'utilisation généraux liés à notre véhicule autonome, *14* cas d'utilisation extensions du cas d'utilisation UC5, *1* cas d'utilisation dérivé du cas d'utilisation UC4, *2* cas d'utilisation inclus dans le cas d'utilisation UC8 et *1* cas d'utilisation qui étend le cas d'utilisation UC6.

Les 8 cas d'utilisation généraux des véhicules autonomes sont :

- UC1 : Passer en mode autonome : le conducteur peut passer en mode de conduite autonome lorsque le véhicule est à l'arrêt.
- UC2 : Passer en mode manuel à la demande du conducteur : le conducteur doit pouvoir reprendre la main sur le contrôle du véhicule à tout moment.
- UC3 : Générer un itinéraire : l'opérateur fournit une destination finale. A partir du positionnement du véhicule et d'informations (carte et infrastructure), l'application de contrôle-commande génèrera l'itinéraire à suivre.
- UC4 : Suivre un itinéraire : ce cas d'utilisation va s'assurer que le véhicule suit correctement les différentes étapes de l'itinéraire jusqu'à sa destination finale.

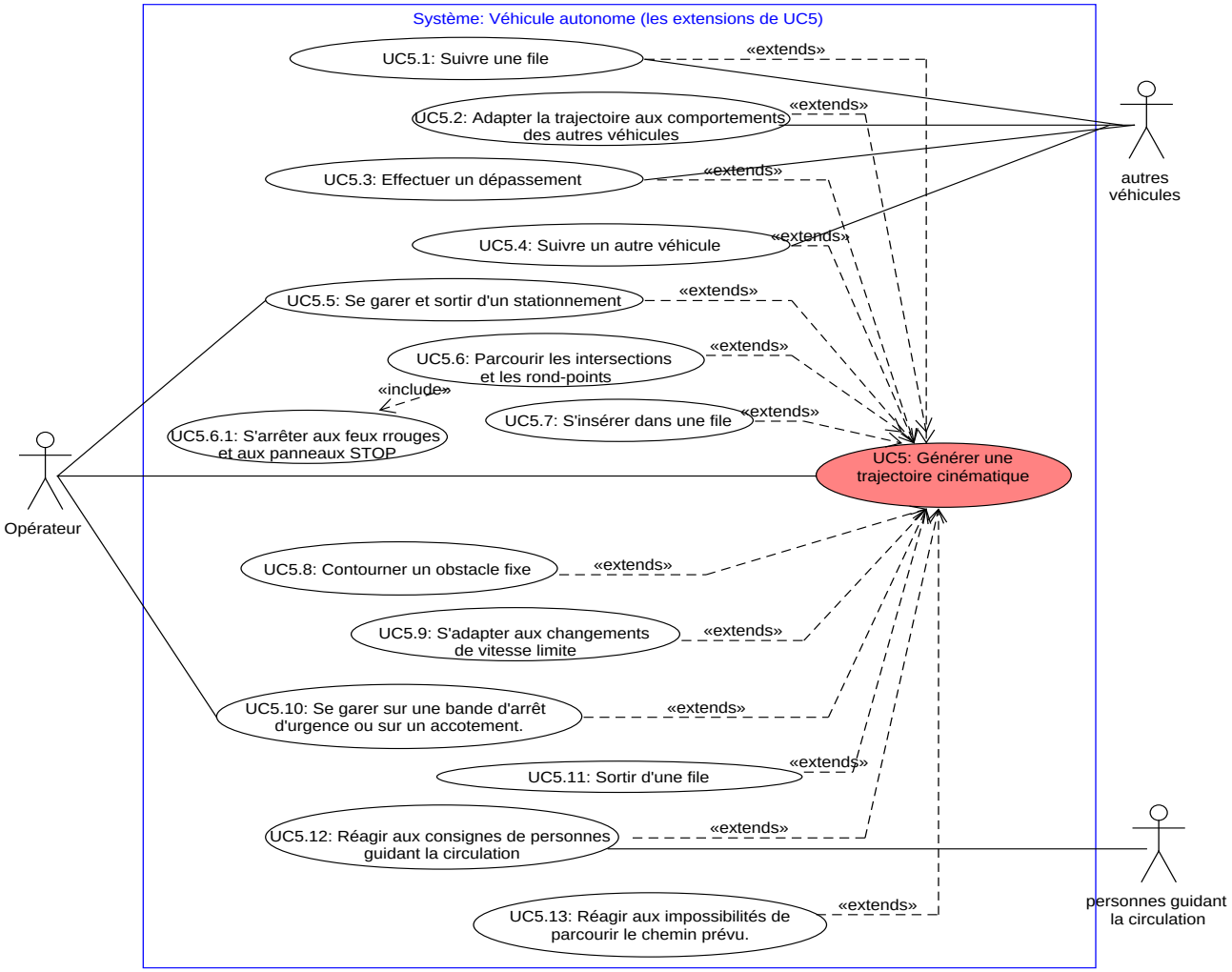

Figure 3.5 – Les extensions du cas d'utilisation UC5 « *Générer une trajectoire cinématique* »

- UC5 : Générer une trajectoire cinématique [1] : ce cas d'utilisation cherche à générer une trajectoire pour le véhicule, en fonction de la destination d'étape à atteindre, de l'état cinématique du véhicule et de l'environnement identifié (autres véhicules, piétons, signalisation routière, espace navigable, etc.)

- UC6 : Suivre une trajectoire cinématique : Ce cas d'utilisation détermine les commandes à appliquer aux actionneurs pour suivre la trajectoire générée au cas d'utilisation UC5.

- UC7 : Suivre l'état d'alertes : Le conducteur installé au poste de conduite supervise la conduite autonome, surveille les alertes et reprend le contrôle manuel en cas de besoin.

[1] trajectoire cinématique : voir définition ci-dessous

- UC8 : Connaitre l'état du véhicule autonome et de son environnement : ce cas d'utilisation consiste à constamment connaître l'état du véhicule (vitesses et accélérations latérales et longitudinales) et sa situation par rapport à l'environnement (localisation, espace navigable, etc.). Ces informations permettront la génération et le suivi de trajectoire (UC6 et UC5), le suivi d'itinéraire (UC4) et le suivi d'alertes (UC7).

L'application de contrôle-commande manipule des informations nécessaires sur les mouvements et la navigation du véhicule autonome. Nous avons considéré les données suivantes :

- La pose du véhicule : est constituée de la position géographique (x,y) du véhicule et de son orientation (cap).
- L'état cinématique estimé par le véhicule : est constitué de la pose du véhicule et de sa vitesse longitudinale (v). Il est donc composé de sa position géographique (x,y), de l'orientation (cap) et de la vitesse (v) du véhicule.
- La trajectoire cinématique : est l'ensemble des positions, orientation (cap) et la vitesse que l'application de contrôle-commande prévoit de faire suivre au véhicule dans un futur proche (de l'ordre 5 secondes).
- L'espace navigable : est l'ensemble des positions dans lesquelles peut physiquement se trouver le véhicule sans heurter des obstacles fixes ou mobiles.
- La zone navigable en sécurité (ZNS) : est l'espace dont sont exclues les marges de sécurité. Ces marges dépendent de la vitesse du véhicule et de la nature des obstacles fixes ou mobiles et de leurs positions.

3.3.1.2 Attributs et déviations

Après avoir identifié l'ensemble des cas d'utilisation, nous définissons les attributs qui représentent des propriétés physiques ou logiques des cas d'utilisation. Ces attributs sont utilisés dans la méthode HazOp-UML pour déterminer les déviations. Les attributs des cas d'utilisation choisis dans le contexte d'HazOp-UML sont :

- Les pré-conditions : les conditions préalables obligatoires au bon déroulement du cas d'utilisation considéré.
- Les invariants : les conditions qui doivent être vraies pendant le déroulement du cas d'utilisation considéré.

Mot guide	Interprétation
No/none	La condition n'est pas évaluée et peut avoir n'importe quelle valeur.
Other than	La condition est évaluée à une valeur incorrecte.
As well as	La condition est correctement évaluée, mais d'autres conditions inattendues sont vraies.
Part of	La condition est partiellement évaluée. Certaines conditions sont manquantes.
Early	La condition est évaluée plus tôt que nécessaire.
Late	La condition est évaluée plus tard que nécessaire.

Tableau 3.6 – Liste de mots guides.

- Les post-conditions : les conditions qui doivent être vraies après le déroulement du cas d'utilisation considéré.

Après avoir identifié les différents attributs, nous appliquerons à chaque attribut une liste composée des 6 guides présentés dans le tableau 3.6, de façon similaire à l'exemple donné section 1.3.2.4, afin de déterminer les déviations du système, qui serviront à leur tour à définir les exigences de sécurité.

Nous avons identifié les différents attributs associés aux trois cas d'utilisation suivants. Les autres use case n'ont pas été considérés, car trop centrés sur le fonctionnement détaillé de l'application de contrôle-commande, que nous considérons comme une boîte noire dans notre étude.

1. **UC 4.1 : Suivre une étape d'itinéraire (tableau 3.7)**

   Pour ce cas d'utilisation, nous avons défini 3 pré-conditions, 3 invariants et 3 post-conditions. Nous étudions ici la première pré-condition (PrC1) : La pose estimée du véhicule (composée de sa position géographique (x,y) et de son orientation (cap)) est à l'origine d'étape, c'est-à-dire, au début de l'étape de l'itinéraire. En appliquant la liste standard composée de 6 mots guides (définie dans le tableau 3.6), nous avons élicité 2 déviations et 3 autres, qui sont des sous-cas des 2 premières.

   - En appliquant la condition « *no/none* », qui signifie que la condition de l'attribut n'est pas évaluée et que nous n'avons donc aucune information sur cette pré-condition, nous avons défini la déviation associée : la pose estimée (X,Y, et cap) n'est pas connue ou pas comparée à l'origine de l'étape.
   - En appliquant le mot guide « *Other than* », l'attribut est évalué incorrectement et on obtient la déviation : la pose estimée est incorrecte.

- Nous n'avons pas appliqué le mot guide « *as well as* », puisque nous n'avons pas identifié de condition additionnelle qui causerait problème sur ce prédicat.
- En appliquant le mot clé « *part of* », qui signifie que l'attribut est partiellement correct, nous obtenons la déviation suivante : seule une partie de la pose du véhicule est correcte (seulement une partie de sa position et de son orientation).
- Les mots guide « *early* » et « *late* » signifient que la condition est évaluée trop tôt ou trop tard par rapport au moment désiré, et dans notre cas correspondent à une déviation de désynchronisation temporelle. Celle-ci soulèvera probablement une erreur dans le cas du mot guide « *early* », qui devra être traitée correctement par le système. Dans le cas du mot guide « *late* », les effets dépendent principalement du comportement de l'application de contrôle commande, que nous considérons comme une boîte noire dans notre étude, pour permettre son applicabilité à différentes applications.

Nous avons trouvé dans notre étude des invariants équivalents en étudiant les 2 cas d'utilisation UC4.1 et UC6, qui exigent que *les composants du véhicule ne défaillent pas.* Satisfaire cet invariant implique d'identifier tous les composants (capteurs, actionneurs ou composants de l'application de contrôle-commande, etc.) et toutes leurs défaillances. Ceci est l'objectif de la méthode d'analyse de risques AMDEC qui met l'accent sur les éléments internes du système. On peut considérer que cet invariant renvoie donc à l'AMDEC du véhicule autonome.

Notons qu'en associant certains mots guide (*As well as* par exemple), nous trouvons parfois des exigences de sécurité très larges qui doivent être traitées de façon complémentaires par une autre méthode.

Comme nous l'avons évoqué précédemment, les exigences de sécurité sont identifiées par l'analyse de sécurité. Nous les déterminons donc en nous appuyant sur nos connaissances sur le système et ses fonctionnalités en considérant les situations des déviations HazOp-UML identifiées.

2. **UC 6 : Suivre la trajectoire cinématique (tableau 3.9)**

   Pour ce cas d'utilisation, nous avons défini 5 pré-conditions, 3 invariants et 3 post-conditions. La première pré-condition (PrC1) consiste en ce que l'état cinématique estimé par le véhicule soit conforme à la réalité.

   - Comme précédemment, lui appliquer le mot guide « *no/none* » donne une

déviation indiquant que la conformité de l'état cinématique estimé par le véhicule à la réalité n'est pas vérifiée, ou que l'état cinématique estimé par le véhicule n'est pas connu.

- En appliquant le mot guide « *other than* », nous obtenons la déviation suivante : l'état cinématique est incorrect.
- En appliquant le mot guide « *as well as* », une déviation peut être trouvée même si l'évaluation est correctement réalisée, par exemple l'espace navigable généré par le système est incorrect. Le véhicule n'aura alors aucun moyen de générer la trajectoire cinématique correcte.
- En appliquant le mot guide « *part of* », cet attribut peut être partiellement évalué, c'est-à-dire, une ou plusieurs variables de l'état cinématique (x, y, cap et v) peuvent être incorrectes ou manquantes.
- Finalement en appliquant les mots clés « *early* » et « *late* », l'état cinématique estimé par le véhicule peut être temporellement désynchronisé, donc soit évalué plus tôt, soit évalué plus tard que nécessaire.

Nous avons étudié en détails ce cas d'utilisation UC6 dans la section suivante en appliquant la méthode d'analyse HazOp-UML.

3. **UC 8 : Connaître l'état du véhicule autonome et de son environnement (tableau 3.11)**

   Pour ce cas d'utilisation, nous avons défini 4 pré-conditions, 4 invariants et une seule post-condition. Nous avons trouvé des attributs qui sont à la fois une pré-condition et un invariant, et également un attribut qui est à la fois un invariant et une post-condition. En effet,

   - la pré-condition PrC2 et l'invariant I1 sont les mêmes : *les mécanismes de détection d'erreur et de diagnostic du véhicule fonctionnent correctement.* Ceci exprime le fait que cette condition doit être vraie avant et pendant le cas d'utilisation.
   - la pré-condition PrC4 et l'invariant I3 sont aussi les mêmes : *les conditions environnementales perçues par le véhicule correspondent à la réalité.*
   - et finalement, l'invariant I4 et la post-condition PoC1 sont les mêmes : *Les données du diagnostic et de la boîte noire du véhicule sont compréhensibles et enregistrées.* Cela exprime le fait que cette condition doit obligatoirement être vérifiée pendant et après le déroulement du cas d'utilisation.

Comme exemple, nous détaillons comment est appliquée la liste de mots guides à l'attribut « *les mécanismes de diagnostic du véhicule fonctionnent correctement* ».

- En appliquant le mot guide « *no/none* », nous obtenons la déviation suivante : nous n'avons aucune information sur l'état du système de diagnostic.
- En appliquant le mot guide « *other than* », nous obtenons la déviation : un ou plusieurs mécanismes de diagnostic du véhicule ne fonctionnent pas correctement.
- En appliquant le mot clé « *as well as* », nous obtenons la déviation : certaines défaillances dans le système peuvent être non diagnosticables, telles qu'une perturbation du GPS.
- En appliquant le mot guide « *part of* », la déviation est : une partie des mécanismes de diagnostic peuvent fonctionner correctement alors que d'autres peuvent donner un diagnostic imprécis. Par exemple, une dégradation du capteur d'une caméra est la cause de mauvaise image, mais cette cause peut être confondue avec de mauvaises conditions de visibilité.
- Dans cette analyse, un faux diagnostic d'erreur est évalué lors de l'application du mot guide « *early* » tandis qu'une non détection ou une détection tardive l'est lors de l'application du mot guide « *late* ».

Tableau 3.7 – UC 4.1 : Suivre une étape d'itinéraire

Use case 4.1 : Suivre une étape d'itinéraire	
• **Projet : Safety-Bag** • **Table Hazop** • **Entité Heudiasyc**	**Préconditions :** ◇ **PrC1 :** La pose estimée du véhicule est à l'origine d'étape. ◇ **PrC2 :** La pose estimée du véhicule est conforme à l'état cinématique réel. ◇ **PrC3 :** Le chemin allant de l'origine d'étape à l'objectif d'étape est défini. **Invariants :** ◇ **I1 :** Les composants du véhicule ne défaillent pas (les actionneurs, les capteurs, l'application de contrôle-commande, etc.). ◇ **I2 :** Le véhicule est en capacité de suivre le chemin jusqu'à l'objectif d'étape malgré des aléas environnementaux (mauvaise visibilité, neige, mauvaise adhérence de la route, etc.). ◇ **I3 :** Le chemin d'étape reste applicable (pas de route barrée, pas de route bloquée, etc.). **Post-conditions :** ◇ **PoC1 :** L'étape d'itinéraire suivante est définie ou le véhicule a atteint la destination. ◇ **PoC2 :** La pose du véhicule est à l'objectif d'étape. ◇ **PoC3 :** La pose estimée est conforme à la pose réelle.

PrC1 : La pose estimée du véhicule est à l'origine d'étape.	
Mot guide	**Déviation**
No/none	La pose estimée du véhicule est inconnue ou n'est pas vérifiée.
Other than	La pose estimée du véhicule n'est pas à l'origine d'étape.
As well as	_
Part of	Sous cas de « Other than »
Early	idem « Other than » (à réévaluer au moment opportun)
Late	idem « Other than » (défaillance temporelle de l'application de contrôle-commande)

PrC2 : La pose estimée du véhicule est conforme à l'état cinématique réel.	
Mot Guide	**Déviation**
No/none	La pose estimée du véhicules est inconnue ou n'est pas vérifiée.
Other than	La pose estimée n'est pas conforme à l'état cinématique réel.
As well as	_
Part of	Sous cas de « Other than »
Early	Déséquencement temporel de la pose
Late	idem « Early »

PrC3 : Le chemin allant de l'origine d'étape à l'objectif d'étape est défini.	
Mot guide	**Déviation**
No/none	L'origine d'étape n'est pas définie ou l'objectif d'étape n'est pas défini ou le chemin n'est pas défini, ou la condition n'est pas vérifiée.
Other than	Le chemin est mal défini.
As well as	L'objectif d'étape est inaccessible (Exemple : la destination est bloquée par des travaux)
Part of	_
Early	Problème temporel dans le suivi d'étape
Late	idem « Early »

I1 : Les composants du véhicule ne défaillent pas.	
Mot Guide	**Déviation**
No/none	Pas d'informations sur l'état d'un ou plusieurs composants du véhicule.
Other than	Un ou plusieurs composants sont défaillants.
As well as	défaillances multiples causant la défaillance des mécanismes de détection (par exemple, défaillance simultanée de capteurs redondants).
Part of	_
Early	Fausse détection de défaillance
Late	Une défaillance n'est pas détectée.

I2 : Le véhicule n'est pas en capacité de suivre le chemin jusqu'à l'objectif d'étape à cause des aléas environnementaux.	
Mot guide	**Déviation**
No/none	L'objectif d'étape est inconnu ou le chemin est inconnu.
Other than	L'objectif d'étape n'est pas atteignable pour ce véhicule.
As well as	_
Part of	_
Early	_
Late	_

I3 : Le chemin d'étape reste applicable (pas de route barrée, pas de route bloquée,etc.).	
Mot Guide	**Déviation**
No/none	Le chemin d'étape est inconnu ou la condition n'est pas vérifiée.
Other than	Le chemin d'étape n'est pas applicable (déviation, route barrée, etc.).
As well as	_
Part of	_
Early	_
Late	_

PoC1 : L'étape suivante est définie ou le véhicule a atteint la destination.	
Mot guide	**Déviation**
No/none	L'étape suivante est inconnue ou la position du véhicule est inconnue ou la condition n'est pas évaluée.
Other than	L'étape suivante n'est pas définie alors que le véhicule n'a pas atteint sa destination.
As well as	_
Part of	_
Early	La boucle de suivi d'étape est désynchronisée.
Late	La boucle de suivi d'étape est désynchronisée.

PoC2 : La pose du véhicule est à l'objectif d'étape.	
Mot Guide	**Déviation**
No/none	La pose estimée du véhicule est inconnue ou l'objectif d'étape est inconnu ou la condition n'est pas évaluée.
Other than	La pose du véhicule n'est pas à l'objectif d'étape.
As well as	_
Part of	Sous cas du « Other than » (seulement partie de la pose)
Early	La boucle de suivi d'étape est désynchronisée.
Late	La boucle de suivi d'étape est désynchronisée.

PoC3 : La pose estimée est conforme à la pose réelle.	
Mot guide	**Déviation**
No/none	La pose estimée est inconnue ou la condition n'est pas évaluée.
Other than	La pose estimée n'est pas conforme à la pose réelle.
As well as	_
Part of	_
Early	La boucle de suivi de trajectoire est temporellement désynchronisée.
Late	idem « Early »

Tableau 3.9 – UC 6 : Suivre une trajectoire cinématique

Use case 6 : Suivre une trajectoire cinématique	
• **Projet : Safety-Bag** • **Table Hazop** • **Entité Heudiasyc**	**Préconditions :** ◇ **PrC1 :** L'état cinématique estimé par le véhicule est conforme à la réalité. ◇ **PrC2 :** L'état cinématique estimé par le véhicule est au début de la trajectoire cinématique. ◇ **PrC3 :** La trajectoire cinématique ne doit pas causer la perte de contrôle du véhicule. ◇ **PrC4 :** L'espace navigable contient la trajectoire cinématique. ◇ **PrC5 :** L'espace navigable est estimé correctement. **Invariants :** ◇ **I1 :** L'emprise du véhicule en suivant la trajectoire cinématique est dans la zone navigable en sécurité. ◇ **I2 :** L'état cinématique estimé par le véhicule ne dépasse pas une distance limite par rapport la trajectoire cinématique. ◇ **I3 :** Les composants du véhicule ne défaillent pas. **Post-conditions :** ◇ **PoC1 :** Le véhicule connait la trajectoire cinématique suivante. ◇ **PoC2 :** L'état cinématique du véhicule est au début de la trajectoire cinématique suivante. ◇ **PoC3 :** L'état cinématique du véhicule est conforme à l'état cinématique réel.

PrC1 : L'état cinématique estimé par le véhicule est conforme à la réalité.	
Mot guide	**Déviation**
No/none	L'état cinématique est inconnu ou la condition n'est pas évaluée.
Other than	L'état cinématique est incorrect.
As well as	Les cartes routières embarquées sont incorrectes.
Part of	Une ou plusieurs variables de l'état cinématique (x,y,cap,v) sont manquantes ou incorrectes.
Early	L'état cinématique est temporellement désynchronisé.
Late	idem « Early »

PrC2 : L'état cinématique estimé par le véhicule est au début de la trajectoire cinématique.	
Mot Guide	**Déviation**
No/none	L'état cinématique est inconnu ou la condition n'est pas évaluée.
Other than	L'état cinématique n'est pas au début de la trajectoire cinématique.
As well as	_
Part of	Une partie des variables de l'état cinématique est manquante ou ne correspond pas au début de la trajectoire cinématique.
Early	L'état cinématique estimé par le véhicule atteint le début de la trajectoire cinématique trop tôt : le véhicule n'a pas encore de trajectoire suivante. Il faut repasser en mode manuel.
Late	L'état cinématique estimé par le véhicule atteint le début de la trajectoire cinématique trop tard : l'application de contrôle-commande devra resynchroniser la nouvelle trajectoire.

PrC3 : La trajectoire cinématique ne doit pas causer la perte de contrôle du véhicule.	
Mot guide	**Déviation**
No/none	La trajectoire cinématique est inconnue ou la condition n'est pas évaluée.
Other than	La trajectoire cinématique peut causer la perte de contrôle du véhicule.
As well as	Les conditions environnementales ne sont pas celles attendues. La perte de contrôle du véhicule est possible dans les limites de conduite usuelle de l'application de contrôle-commande.
Part of	_
Early	L'espace navigable peut avoir évolué et la trajectoire doit être rafraichie.
Late	idem « Early »

PrC4 : L'espace navigable contient la trajectoire cinématique.	
Mot Guide	**Déviation**
No/none	L'espace navigable est inconnu ou la condition n'est pas évaluée.
Other than	La trajectoire cinématique n'est pas incluse dans l'espace navigable.
As well as	Un obstacle dynamique interfère avec la trajectoire cinématique.
Part of	_
Early	La boucle de contrôle du suivi de trajectoire est désynchronisée temporellement.
Late	idem« Early »

PrC5 : L'espace navigable est estimé correctement.	
Mot guide	**Déviation**
No/none	L'espace navigable est inconnu ou la condition n'est pas évaluée.
Other than	L'espace navigable n'est pas estimé correctement.
As well as	Les données des capteurs ne permettent pas de calculer l'espace navigable.
Part of	Une partie de l'espace navigable est calculée incorrectement ou manquante.
Early	L'espace navigable est désynchronisé temporellement.
Late	idem « Early »

I1 : L'emprise du véhicule en suivant la trajectoire est dans la zone navigable en sécurité.	
Mot Guide	**Déviation**
No/none	L'espace navigable est inconnu ou la trajectoire cinématique est inconnue ou la condition n'est pas évaluée.
Other than	L'emprise du véhicule en suivant la trajectoire sort de la zone navigable.
As well as	La zone navigable en sécurité (ZNS) est trop près des obstacles (marges insuffisantes).
Part of	_
Early	La boucle de contrôle de suivi de trajectoire est désynchronisée temporellement.
Late	idem « Early »

I2 : L'état cinématique estimé par le véhicule ne dépasse pas une distance limite par rapport la trajectoire cinématique.	
Mot guide	**Déviation**
No/none	L'état cinématique est inconnu ou la trajectoire cinématique est inconnue ou la condition n'est pas évaluée.
Other than	L'état cinématique estimé par le véhicule s'écarte de manière significative de la trajectoire cinématique.
As well as	Un obstacle dynamique interfère avec la trajectoire cinématique.
Part of	_
Early	La boucle de suivi d'étape est désynchronisée.
Late	idem « Early »

I3 : Les composants du véhicule ne défaillent pas.	
Mot Guide	**Déviation**
No/none	Pas d'informations sur l'état d'un ou plusieurs composants du véhicule ou la condition n'est pas évaluée.
Other than	Un ou plusieurs composants sont défaillants.
As well as	_
Part of	_
Early	Fausse détection de défaillance
Late	Une défaillance n'est pas détectée

PoC1 : Le véhicule connaît la trajectoire cinématique suivante.	
Mot guide	**Déviation**
No/none	La trajectoire cinématique n'est pas connue ou la condition n'est pas évaluée.
Other than	La trajectoire cinématique n'est pas connue.
As well as	_
Part of	Sous cas du« *Other than* »
Early	La boucle de suivi d'étape est temporellement désynchronisée.
Late	idem « Early »

PoC2 : L'état cinématique du véhicule est au début de la trajectoire cinématique suivante.	
Mot guide	**Déviation**
No/none	L'état cinématique du véhicule n'est pas connu ou la condition n'est pas évaluée.
Other than	L'état cinématique du véhicule n'est pas à la fin de la trajectoire cinématique précédente.
As well as	_
Part of	Seule une partie de l'état cinématique estimé par le véhicule est à la fin de la trajectoire cinématique précédente.
Early	Le système n'a pas terminé la trajectoire cinématique précédente.
Late	Le système a déjà fini la trajectoire cinématique, sans que l'application de contrôle-commande ait déjà donné la suivante : le système n'a plus de trajectoire et il faut repasser en mode manuel.

PoC3 : L'état cinématique du véhicule est conforme à l'état cinématique réel.	
Mot guide	**Déviation**
No/none	L'état cinématique est inconnu ou la condition n'est pas évaluée.
Other than	Le véhicule ne suit pas la trajectoire cinématique.
As well as	_
Part of	Sous-cas du « *Other than* »
Early	L'état cinématique du système est temporellement désynchronisé.
Late	idem « *Early* »

Tableau 3.11 – Use case 8 : Connaitre l'état du véhicule autonome et de son environnement

Use case 8 : Connaitre l'état du véhicule autonome et de son environnement	
• **Projet : Safety-Bag** • **Table Hazop** • **Entité Heudiasyc**	**Préconditions :** ◇ **PrC1 :** Les capteurs du véhicule fonctionnent correctement. ◇ **PrC2 :** Les mécanismes de diagnostic du véhicule fonctionnent correctement. ◇ **PrC3 :** Le véhicule est correctement localisé sur la carte. ◇ **PrC4 :** Les conditions environnementales perçues par le véhicule correspondent à la réalité. **Invariants :** ◇ **I1 :** Les mécanismes de diagnostic du véhicule fonctionnent correctement. ◇ **I2 :** Le véhicule est correctement localisé sur la carte. ◇ **I3 :** Les conditions environnementales perçues par le véhicule correspondent à la réalité. ◇ **I4 :** Les informations de diagnostics de la boite noire sont compréhensibles et enregistrées. **Post-conditions :** ◇ **PoC1 :** Les informations de diagnostics de la boite noire sont compréhensibles et enregistrées.

PrC1 : Les capteurs du véhicule fonctionnent correctement.	
Mot guide	**Déviation**
No/none	L'état de fonctionnement des capteurs est inconnu ou la condition n'est pas évaluée.
Other than	Un ou plusieurs capteurs du véhicule ne fonctionnent pas.
As well as	_
Part of	_
Early	Faux positif (détection d'erreur intempestive)
Late	Absence de détection (erreur non détectée)

PrC2 : Les mécanismes de diagnostic du véhicule fonctionnent correctement.	
Mot Guide	**Déviation**
No/none	On ne connait pas l'état du système de diagnostic ou son état n'est pas évalué.
Other than	Un ou plusieurs mécanismes de diagnostic du véhicule ne fonctionnent pas correctement.
As well as	Autre défaillance dans le système (qui par conséquent n'est pas diagnosticable)
Part of	Certains mécanismes de diagnostic fonctionnent correctement.
Early	Faux positif
Late	Absence de détection

PrC3 : Le véhicule est correctement localisé sur la carte.	
Mot guide	**Déviation**
No/none	La localisation n'est pas connue ou la condition n'est pas évaluée.
Other than	Le véhicule n'est pas localisé correctement sur la carte.
As well as	La carte est erronée.
Part of	Sous cas de « Other than »
Early	Désynchronisation de la localisation
Late	idem « Early »

PrC4 : Les conditions environnementales perçues par le véhicule correspondent à la réalité.	
Mot Guide	**Déviation**
No/none	Les conditions environnementales perçues par le véhicule sont inconnues ou la condition n'est pas évaluée.
Other than	Les conditions environnementales perçues par le véhicule ne correspondent pas à la réalité.
As well as	Des capteurs ou fonctionnalités manquent pour détecter certaines conditions spécifiques.
Part of	Sous cas de « Other than »
Early	Désynchronisation de la perception de l'environnement
Late	« idem Early »

I1 : Les mécanismes de diagnostic du véhicule fonctionnent correctement.	
Mot guide	**Déviation**
No/none	On ne connait pas l'état du système de diagnostic ou son état n'est pas évalué.
Other than	Un ou plusieurs mécanismes de diagnostic du véhicule ne fonctionnent pas correctement.
As well as	autre défaillance dans le système
Part of	Certains mécanismes de diagnostic fonctionnent correctement.
Early	Faux positif
Late	Absence de détection

I2 : Le véhicule est correctement localisé sur la carte.	
Mot Guide	**Déviation**
No/none	La localisation n'est pas connue ou la condition n'est pas évaluée.
Other than	Le véhicule n'est pas localisé correctement sur la carte.
As well as	La carte est erronée.
Part of	Sous cas de « Other than »
Early	Désynchronisation de la localisation
Late	idem « Early »

I3 : Les conditions environnementales perçues par le véhicule correspondent à la réalité.	
Mot guide	**Déviation**
No/none	Les conditions environnementales perçues par le véhicule sont inconnues ou la condition n'est pas évaluée.
Other than	Les conditions environnementales perçues par le véhicule ne correspondent pas à la réalité.
As well as	Des capteurs ou fonctionnalités manquent pour détecter certaines conditions spécifiques.
Part of	Sous cas de « Other than »
Early	Désynchronisation de la perceptions de l'environnement
Late	idem « Early »

I4 : Les informations de diagnostic et de la boîte noire sont compréhensibles et enregistrées.	
Mot Guide	**Déviation**
No/none	Les informations de diagnostic et de la boîte noire sont inconnues ou la condition n'est pas évaluée.
Other than	Les informations de diagnostic et de la boîte noire ne sont pas suffisantes ou le support des informations est endommagé.
As well as	_
Part of	Sous cas du « Other than »
Early	Désynchronisation dans les informations de diagnostic
Late	idem « Early »

PoC1 : Les informations de diagnostic et de la boite noire sont compréhensibles et enregistrées	
Mot guide	**Déviation**
No/none	Les informations de diagnostic et de la boîte noire sont inconnues ou la condition n'est pas évaluée.
Other than	Les informations de diagnostic et de la boîte noire ne sont pas suffisantes ou le support des informations est endommagé.
As well as	_
Part of	Sous cas du « Other than »
Early	Désynchronisation dans les informations de diagnostic
Late	idem « Early »

3.3.2 L'analyse de risque HazOp et l'identification des exigences de sécurité

Après avoir défini les attributs de chaque use case et avoir obtenu les déviations possibles du système en ayant associé les mots guide à chacun, il nous reste à définir pour chaque déviation ses conséquences (les effets), ses causes possibles (composant matériel ou logiciel, par exemple), sa gravité (selon le tableau 3.1) et enfin l'exigence de sécurité qui permet de la traiter, que nous dériverons dans le chapitre suivant en nécessité de sécurité.

L'entête de la table de déviations de l'analyse HazOp-UML contient les champs suivants :

- le cas d'utilisation considéré,
- l'identifiant de la déviation, pour pouvoir la tracer et la référencer,
- l'attribut considéré,
- le mot guide appliqué (voir 3.6) ,
- les effets sur le système : ce sont des conséquences de la déviation sur les composants du système,
- les effets sur le cas d'utilisation : ce sont des conséquences finales pour l'utilisateur et l'environnement,
- la gravité (G) selon l'échelle de gravité,
- les causes possibles de la déviation (logiciel, matériel, environnemental, etc.),
- l'exigence de sécurité qui permet de la traiter,
- des commentaires : explication de l'analyse, recommandations supplémentaires, etc.
- le numéro d'exigence de sécurité (NES).

Nous avons détaillé en particulier le cas d'utilisation UC6 « *Suivre une trajectoire cinématique* ». Nous pensons que les autres cas d'utilisation sont semblables et donnent des résultats similaires. Comme ce cas d'utilisation est d'un niveau d'abstraction proche du fonctionnel et fournit des commandes aux actionneurs, contrairement à d'autres cas d'utilisation plus abstraits comme « *générer un itinéraire* », nous pourrons comparer par la suite les résultats obtenus dans la

détermination de ces exigences et ces nécessités de sécurité avec ceux fournis par la méthode d'analyse de risques AMDEC.

Pour le cas d'utilisation choisi, nous proposons une liste composée de 11 attributs (comme montré dans le tableau 3.13 : 5 pré-conditions, 3 invariants et 3 post-conditions).

Les tableaux (3.13, 3.14, 3.15, 3.16, 3.17, 3.18 et 3.19) représentent l'étude HazOp complète de ce cas d'utilisation. Au début, nous avons associé à chaque attribut une liste de mots guides afin de déterminer les déviations. Ensuite, nous avons associé à chaque déviation les conséquences et les causes possibles, la gravité et finalement les exigences de sécurité indispensables pour extraire par la suite les nécessités de sécurité. En étudiant le cas d'utilisation *Suivre une trajectoire cinématique*, nous avons réussi à identifier 46 déviations. Nous avons associé dans la majorité des cas le niveau de gravité le plus élevé 5, vu que ces déviations peuvent avoir des effets graves sur le cas d'utilisation et sur le système, donc des conséquences dangereuses respectivement pour l'utilisateur et l'environnement et sur les composants du système.

Par exemple, si l'espace navigable, c'est-à-dire l'ensemble des positions dans laquelle peut physiquement se trouver le véhicule sans heurter les obstacles fixes ou mobiles, n'est pas estimé correctement à cause d'une défaillance logicielle ou d'une condition environnementale imprévue, cela peut se traduire par une trajectoire cinématique incorrecte et des collisions. La gravité dans ce cas est 5 et nous exigeons que le système soit capable de vérifier que l'espace navigable est correct (ligne 20 dans le tableau 3.16). De plus, si la zone navigable en sécurité (ZNS), c'est-à-dire l'espace dont sont exclues les marges de sécurité, est trop près des obstacles à cause d'une défaillance logicielle, le suivi correct de la trajectoire cinématique estimée peut amener le véhicule à heurter un obstacle, ce qui conduit à des effets similaires. L'exigence de sécurité dans ce cas est que le système soit capable de prendre des marges suffisantes pour la zone navigable en sécurité (ligne 26 dans le tableau 3.17).

Tableau 3.13 – HaZop (page 1)

• **Projet : Safety-Bag** • **Table Hazop** • **Entité**	**Use case 6 : Suivre la trajectoire cinématique** **Pré-conditions (5) :** ◇ PrC1 : L'état cinématique estimé par le véhicule est conforme à la réalité. ◇ PrC2 : L'état cinématique estimé par le véhicule est au début de la trajectoire cinématique. ◇ PrC3 : La trajectoire cinématique ne doit pas causer la perte de contrôle du véhicule. ◇ PrC4 : L'espace navigable contient la trajectoire cinématique. ◇ PrC5 : L'espace navigable est estimé correctement. **Invariants (3) :** ◇ I1 : L'emprise du véhicule en suivant la trajectoire cinématique est dans la zone navigable en sécurité. ◇ I2 : L'état cinématique estimé par le véhicule ne dépasse pas une distance limite de la trajectoire cinématique. ◇ I3 : Les composants du véhicule ne défaillent pas. **Post-conditions (3) :** ◇ PoC1 : Le véhicule connait la trajectoire cinématique suivante. ◇ PoC2 : L'état cinématique du véhicule est au début de la trajectoire cinématique suivante. ◇ PoC3 : L'état cinématique du véhicule est conforme à l'état cinématique réel.	• **Date** • **Préparé par : Manel Brini**

N°	**Attribut**	**Mot Guide**	**Déviation**	**Effets sur le cas d'utilisation**	**Effets sur le système**	**G**	**Causes possibles**	**Exigences de sécurité**	**Commentaires**	**NES**
1	Pré-condition 1	No/none	L'état cinématique estimé par le véhicule est inconnu ou la condition n'est pas évaluée.	L'application de contrôle-commande renvoie une erreur ou n'importe quoi, le système peut faire n'importe quoi.	Le véhicule ne peut plus suivre la trajectoire (risque de collision).	5	défaillance matérielle (capteur, connectique) ou défaillance logicielle	Le système doit vérifier ou garantir la vivacité de l'état cinématique estimé par le véhicule.	_	1
2		Other than	L'état cinématique estimé par le véhicule est incorrect.	L'application de contrôle-commande suit une trajectoire décalée sans renvoyer de diagnostic.	Le véhicule suit une trajectoire décalée (risque de collision).	5	défaillance matérielle (capteur de localisation) ou défaillance logicielle (fusion des données)	Le système doit vérifier ou garantir que l'état cinématique estimé par le véhicule est correct.	redondance sur le mécanisme de localisation (plus difficile à diagnostiquer que l'exigence de sécurité 1)	2
3		As well as	Les cartes routières embarquées sont incorrectes.	L'application de contrôle-commande produit/suit une trajectoire incorrecte.	Le véhicule suit une mauvaise trajectoire (risque de collision).	5	défaillance logicielle ou modification environnementale non répercutée sur les cartes (travaux, effondrement,etc.)	Le système doit s'assurer de la conformité de ce que perçoit le véhicule avec les cartes routières embarquées.	dépend de l'utilisation des cartes routières dans la génération de trajectoire	3
4		Part of	Une ou plusieurs variables de l'état cinématique estimé par le véhicule (x,y,cap,v) sont incorrectes ou manquantes.	idem 2						2
5		Early/late	L'état cinématique estimé par le véhicule est temporellement désynchronisé.	Les commandes appliquées sont inadaptées à la situation.	La trajectoire cinématique peut être aberrante (risque de collision).	5	défaillance logicielle temporelle	Le système doit vérifier la cohérence temporelle de l'état cinématique estimé par le véhicule.	_	4

Tableau 3.14 – HaZop (page 2)

N°	**Attribut**	**Mot Guide**	**Déviation**	**Effets sur le cas d'utilisation**	**Effets sur le système**	**G**	**Causes possibles**	**Exigences de sécurité**	**Commentaires**	**NES**
6	Pré-condition 2	No/none	L'état cinématique estimé par le véhicule est inconnu ou la condition n'est pas évaluée.	idem 1						1
7		Other than	L'état cinématique estimé par le véhicule n'est pas au début de la trajectoire cinématique.	impossible de suivre la trajectoire cinématique	Le véhicule ne suit pas la commande appliquée ou essaye de rejoindre brutalement la trajectoire.	4 ou 5	défaillance d'actionneur ou défaillance logicielle	Le système doit vérifier que les actionneurs et l'application de contrôle-commande fonctionnent correctement.	_	5
8		Part of	Une partie des variables de l'état cinématique estimé par le véhicule est manquante ou ne correspond pas au début de la trajectoire cinématique.	impossible de suivre la trajectoire cinématique	trajectoire cinématique inadaptée	4 ou 5	idem 5			5b
9		Early	L'état cinématique estimé par le véhicule atteint le début de la trajectoire cinématique trop tôt : le véhicule n'a pas encore de trajectoire suivante. Il faut repasser en mode manuel.	idem 5b		5	défaillance logicielle	idem 5		5c
10		Late	L'état cinématique estimé par le véhicule atteint le début de la trajectoire cinématique trop tard : l'application de contrôle-commande devra resynchroniser la nouvelle trajectoire.	idem 5b		5	défaillance logicielle	idem 5		5d

Tableau 3.15 – HaZop (page 3)

N°	Attribut	Mot Guide	Déviation	Effets sur le cas d'utilisation	Effets sur le système	G	Causes possibles	Exigences de sécurité	Commentaires	NES
11	Pré-condition 3	No/none	La trajectoire cinématique est inconnue ou la condition n'est pas évaluée.	pas de trajectoire à suivre	Le comportement du véhicule est imprévisible.	5	défaillance logicielle	Le système doit vérifier la vivacité de la trajectoire cinématique.	_	6
12		Other than	La trajectoire cinématique peut causer la perte de contrôle du véhicule.	Le véhicule suit une trajectoire cinématique dangereuse.	possible perte de contrôle du véhicule	4 ou 5	défaillance logicielle	Le système doit détecter une trajectoire cinématique dangereuse.	_	7
13		As well as	Les conditions environnementales ne sont pas celles attendues, risquant la perte de contrôle du véhicule dans les limites de conduite usuelle de l'application de contrôle commande.	La trajectoire cinématique n'est plus adaptée aux conditions environnementales.	risque de perte de contrôle du véhicule	5	défaillance matérielle ou défaillance logicielle	Les conditions environnementales doivent être correctement identifiées par le système.	nécessite une redondance, mais difficile à mettre en œuvre	8
14		Early/late	L'espace navigable peut avoir évolué et la trajectoire doit être rafraichie.	La trajectoire cinématique rencontre un obstacle dynamique.	risque de collision avec un obstacle dynamique	5	environnement dynamique	Le système doit vérifier que l'espace navigable est mis à jour à une fréquence suffisante.	vérifier avant d'appliquer toute commande de conduite que l'espace navigable est toujours libre	9
								La trajectoire cinématique doit être vérifiée et éventuellement mise à jour à chaque rafraichissement de l'espace navigable.		10

Tableau 3.16 – HaZop (page 4)

N°	Attribut	Mot Guide	Déviation	Effets sur le cas d'utilisation	Effets sur le système	G	Causes possibles	Exigences de sécurité	Commentaires	NES
15	Pré-condition 4	No/none	L'espace navigable est inconnu ou la condition n'est pas évaluée.	La trajectoire cinématique ne peut être calculée ou sera incorrecte.	Le véhicule suit la trajectoire, mais cette trajectoire peut être dangereuse.	5	défaillance logicielle	1. idem 6 2. Le système doit vérifier que l'espace navigable est déterminé.	mise en sécurité : passage en mode manuel en urgence	11
16		Other than	La trajectoire cinématique n'est pas incluse dans l'espace navigable.	En suivant la trajectoire cinématique, le véhicule peut heurter des obstacles en dehors de l'espace navigable.		5	défaillance logicielle	idem 10	_	10
17		As well as	Un obstacle dynamique interfère avec la trajectoire cinématique.	risque de collision avec d'autres véhicules	Le véhicule suit sa trajectoire, mais cette dernière ne s'adapte pas à la dynamique d'un autre véhicule.	5	défaillance externe	Le système doit anticiper les mouvements des véhicules et piétons et garder une distance de sécurité avec eux.	problème de robustesse	12
18		Early/late	La boucle de contrôle du suivi de trajectoire est temporellement désynchronisée.	mauvais suivi de la trajectoire cinématique	risque de perdre le contrôle du véhicule, et risque de heurter des obstacles statiques ou mobiles	5	défaillance logicielle	idem 6	_	6
19	Pré-condition 5	No/none	L'espace navigable est inconnu ou la condition n'est pas évaluée.	idem 11 (exigence de sécurité N° 2)						11
20		Other than	L'espace navigable n'est pas estimé correctement.	La trajectoire cinématique est incorrecte.	Le comportement du véhicule est imprévisible.	5	défaillance logicielle ou condition environnementale imprévue	Le système doit vérifier que l'espace navigable est correct.	mise en sécurité : il faut un diagnostic	13
21		As well as	Les données des capteurs ne permettent pas de calculer l'espace navigable.	idem 3			défaillance externe environnementale	Le système doit disposer de suffisamment de capteurs redondants pour tolérer suffisamment leurs défaillances matérielles. Une étude basée sur leurs taux de défaillance peut être nécessaire pour identifier les redondances nécessaires.	problème de robustesse	14
22		Part of	Une partie de l'espace navigable est calculée incorrectement ou manquante.	idem 14			défaillance matérielle ou logicielle	idem 14	mise en sécurité ; repassage en manuel en urgence	14
23		Early/late	L'espace navigable est désynchronisé temporellement.	L'espace navigable est incorrect.	La trajectoire cinématique sera incorrecte.	5	défaillance logicielle	idem 11 (exigence de sécurité N° 2)	_	11

Tableau 3.17 – HaZop (page 5)

N°	Attribut	Mot Guide	Déviation	Effets sur le cas d'utilisation	Effets sur le système	G	Causes possibles	Exigences de sécurité	Commentaires	NES
24	Invariant 1	No/none	L'espace navigable est inconnu ou la condition n'est pas évaluée.	idem 11 (exigence de sécurité N° 2)						11
			La trajectoire cinématique est inconnue.	idem 6						6
25		Other than	L'emprise du véhicule en suivant la trajectoire sort de la zone navigable.	Le véhicule heurte un obstacle.	Le véhicule frôle ou heurte un obstacle en suivant la trajectoire.	5	défaillance matérielle (problème mécanique) ou défaillance logicielle	Le système doit vérifier que l'emprise connue du véhicule reste dans la Zone Navigable en Sécurité (ZNS).	mise en sécurité	15
26		As well as	La ZNS est trop près des obstacles (marges insuffisantes).	Le suivi correct de la trajectoire cinématique estimée peut amener le véhicule à heurter un obstacle.	Risque de heurter un obstacle ou sortir de la route	5	défaillance logicielle	Il faut prendre des marges suffisantes pour la ZNS.	mise en sécurité	16
27		Early/late	La boucle de contrôle du suivi de trajectoire est temporellement désynchronisée.	idem 6						6
28	Invariant 2	No/none	L'état cinématique estimé par le véhicule est inconnu ou la condition n'est pas évaluée.	idem 1						1
			La trajectoire cinématique est inconnue.	idem 6						6
29		Other than	L'état cinématique estimé par le véhicule s'écarte de manière significative de la trajectoire cinématique.	La trajectoire cinématique suivie par l'application de contrôle-commande ne respecte pas la trajectoire cinématique demandée.	Le véhicule n'arrive pas à suivre la trajectoire cinématique demandée.	5	défaillance matérielle (problème d'actionneurs) ou défaillance logicielle (le contrôle fonctionne mal) ou problème d'environnement (glissement)	Le système doit vérifier ou garantir que l'on ne s'éloigne pas trop de la trajectoire cinématique demandée.	–	17
30		As well as	Un obstacle dynamique interfère avec la trajectoire cinématique.	idem 12						12
31		Early/late	La boucle de contrôle du suivi de trajectoire est temporellement désynchronisée.	idem 6						6

Tableau 3.18 – HaZop (page 6)

N°	**Attribut**	**Mot Guide**	**Déviation**	**Effets sur le cas d'utilisation**	**Effets sur le système**	**G**	**Causes possibles**	**Exigences de sécurité**	**Commentaires**	**NES**
32	Invariant 3	No/none	On n'a pas d'informations sur l'état d'un ou plusieurs composants du véhicule ou la condition n'est pas évaluée.	Des composants du véhicules peuvent être défaillants sans qu'on le sache.	On ne peut plus avoir confiance dans le comportement du véhicule.	1 à 5	défaillance du mécanisme de diagnostic	Un composant critique auto-testable doit détecter toutes ses erreurs. De plus, tous les composants critiques doivent être auto-testables ou validés formellement de façon à n'avoir aucune faute interne.	_	18
33		Other than	Un ou plusieurs composants sont défaillants.		Le véhicule peut avoir n'importe quel comportement.	4 ou 5	défaillance matérielle (actionneur défaillant)	Il faut tolérer les défaillances des composants ou garantir que les composants ne défaillent pas.	_	19
34		Early	On a une fausse détection de défaillance.	activation intempestive d'alarmes et de mécanismes de tolérance aux fautes	perte de fiabilité	1 voire 3b	défaillance du mécanisme de diagnostic	idem 19	Il est important de conserver un niveau de fiabilité suffisant.	19
35		Late	une défaillance n'est pas détectée	idem 19						19
36	Post-condition 1	No/none	La trajectoire cinématique n'est pas connue ou la condition n'est pas évaluée.	idem 6						6
37		Other than	La trajectoire cinématique suivante n'est pas connue.	idem 6						6
38		Early/late	La boucle de contrôle du suivi de trajectoire est temporellement désynchronisée.	idem 6						6

Tableau 3.19 – HaZop (page 7)

N°	Attribut	Mot Guide	Déviation	Effets sur le cas d'utilisation	Effets sur le système	G	Causes possibles	Exigences de sécurité	Commentaires	NES
39	Post-condition 2	No/none	L'état cinématique du véhicule n'est pas connu ou la condition n'est pas évaluée.	idem 6						6
40		Other than	L'état cinématique du véhicule n'est pas au début de la trajectoire cinématique précédente.	Il y a eu une erreur pendant le suivi de la trajectoire.	Le véhicule n'a pas été contrôlé comme prévu (freinage inefficace, dérapage, etc.).	5	défaillance du mécanisme de diagnostic	Le système doit vérifier régulièrement lors du suivi de trajectoire la conformité de l'évolution dans le temps de la position du véhicule avec le suivi de trajectoire prévue. Si une incohérence apparaît, une nouvelle trajectoire doit être générée.	_	20
41		Part of	Seule une partie de l'état cinématique du véhicule est à la fin de la trajectoire cinématique précédente.	idem 18						18
42		Early	Le système a fini la trajectoire cinématique, sans que l'application de contrôle-commande ait donné la suivante : le système n'a plus de trajectoire et il faut repasser en mode manuel.	pas de trajectoire à suivre	Le véhicule n'a plus de trajectoire et son comportement est imprévisible.	5	défaillance logicielle	idem 20	_	20
43		Late	Le système n'a pas terminé la trajectoire cinématique précédente.	On ne peut pas commencer la prochaine trajectoire cinématique.	L'application de contrôle-commande doit resynchroniser le suivi de trajectoire et prévoir une nouvelle trajectoire à partir de l'état courant.	0 ou 1	défaillance logicielle	_	Le véhicule n'a pas réussi à arriver où il doit aller. Une nouvelle trajectoire pourrait être générée.	_
44	Post-condition 3	No/none	L'état cinématique est inconnu.	idem 1						1
45		Other than	Le véhicule ne suit pas la trajectoire cinématique.	Le véhicule a un comportement imprévisible.	risque d'accidents	5	défaillance logicielle	idem 18		18
46		Early/late	L'état cinématique du système est temporellement désynchronisé.	idem 4						4

A partir de la méthode d'analyse de risques HazOp-UML, nous avons réussi à identifier les 20 exigences de sécurité (parmi les 46 déviations définies) suivantes :

- Exigence de sécurité 1 : « *Le système doit vérifier ou garantir que l'état cinématique estimé par le véhicule est déterminé.* »

  Cette exigence correspond à la déviation « *L'état cinématique estimé par le véhicule est inconnu* » provoquée par une défaillance matérielle (un problème connectique ou une défaillance d'un capteur) ou par une défaillance logicielle.

- Exigence de sécurité 2 : « *Le système doit vérifier ou garantir que l'état cinématique estimé par le véhicule est correct.* »

  Cette exigence correspond au cas où l'état cinématique ou une partie des variables de l'état cinématique (x,y,cap,v) soient incorrects suite à une défaillance matérielle du capteur de localisation par exemple ou un problème logiciel lié à la fusion des données.

- Exigence de sécurité 3 : « *Le système doit s'assurer de la conformité de ce que perçoit le véhicule avec les cartes routières embarquées.* »

  Cette exigence couvre le cas où les cartes routières embarquées sont incorrectes suite à une modification dans les éléments environnementaux non répercutée sur les cartes routières (nouveaux travaux ou effondrement par exemple) ou des erreurs lors de la digitalisation des cartes.

- Exigence de sécurité 4 : « *Le système doit vérifier la cohérence temporelle de l'état cinématique estimé par le véhicule.* »

  Cette exigence vise à éviter la déviation portant sur la désynchronisation temporelle de l'état cinématique suite à une défaillance logicielle temporelle (comme une surcharge du calculateur causé par trop d'activités ou de calculs).

- Exigence de sécurité 5 : « *Le système doit vérifier que les actionneurs et l'application de contrôle-commande fonctionnent correctement.* »

  En effet, une défaillance au niveau d'un actionneur peut provoquer une incohérence de l'état cinématique du véhicule par rapport à l'état cinématique souhaité par le véhicule en empêchant le véhicule d'atteindre physiquement la position désirée. Également, une défaillance logicielle de l'application de contrôle commande, notamment dans la génération ou le suivi de trajectoire, peut aboutir au même résultat.

- Exigence de sécurité 6 : « *Le système doit vérifier que la trajectoire cinématique est déterminée.* »

Cette exigence de sécurité correspond à la déviation « *la trajectoire cinématique n'est pas vivante (n'est pas mise à jour) régulièrement* » ou « *la boucle de contrôle du suivi de trajectoire cinématique est temporellement désynchronisée* ». Cela peut être dû à une défaillance logicielle des éléments de l'application de contrôle-commande ou des problèmes de communication entre composants.

- Exigence de sécurité 7 : « *Le système doit détecter une trajectoire cinématique dangereuse.* »

  Nous exprimons cette condition de sécurité afin d'éviter la perte de contrôle du véhicule qui peut être causée par une trajectoire cinématique mal adaptée.

- Exigence de sécurité 8 : « *Les conditions environnementales doivent être correctement identifiées par le système.* »

  Nous avons identifié cette exigence de sécurité pour prévenir la déviation « *les conditions environnementales inattendues risquent de provoquer la perte de contrôle du véhicule dans les limites de conduite usuelle de l'application de contrôle-commande* ». Dans ce cas, la trajectoire cinématique prévue pour des conditions environnementales différentes risque de ne pas être aux conditions environnementales courantes.

- Exigence de sécurité 9 : « *Le système doit vérifier que l'espace navigable est mis à jour à une fréquence suffisante.* »

  Nous avons identifié cette exigence de sécurité pour prévenir la déviation *L'espace navigable peut avoir évolué et la trajectoire doit être rafraichie.*, provoquée par une évolution imprévue de l'environnement dynamique.

- Exigence de sécurité 10 : « *La trajectoire cinématique doit être vérifiée et éventuellement mise à jour à chaque rafraichissement de l'espace navigable.* »

  En effet, la mise à jour de la trajectoire cinématique alors que l'espace navigable n'est pas libre, c'est-à-dire qu'il y a présence des piétons ou d'autres véhicules, peut causer des dégâts catastrophiques en heurtant par exemple un obstacle dynamique non prévu. Il faut donc d'une part que l'espace navigable soit mis à jour régulièrement pour prendre en compte ces obstacles dynamiques, et d'autre part qu'une vérification soit faite à chaque mise à jour de l'espace navigable que la trajectoire cinématique prévue reste sur le nouvel espace navigable calculé.

- Exigence de sécurité 11 : « *Le système doit vérifier que l'espace navigable et la trajectoire cinématique sont déterminés.* »

Cette exigence de sécurité est mise en place au cas où l'espace navigable et la trajectoire cinématique ne sont plus rafraichis ou ont été assignés à des valeurs quelconques.

- Exigence de sécurité 12 : « *Le système doit anticiper les mouvements des véhicules et piétons et garder une distance de sécurité avec eux.* »

  Notons que garantir cette exigence de sécurité est à l'heure actuelle impossible pour des systèmes autonomes routiers.

- Exigence de sécurité 13 : « *Le système doit vérifier que l'espace navigable est correct.* »

  En effet, l'espace navigable peut être estimé incorrectement suite à une défaillance logicielle imprévue ou à une condition environnementale inattendue.

- Exigence de sécurité 14 : « *Le système doit disposer de suffisamment de capteurs redondants pour tolérer leurs défaillances matérielles.* »

  Une étude basée sur leurs taux de défaillance peut être nécessaire pour identifier les redondances nécessaires. Cette condition de sécurité est déduite du fait que les données des capteurs peuvent être incorrectes ou insuffisantes pour calculer l'espace navigable. La trajectoire cinématique est donc incorrecte et le comportement du véhicule est imprévisible.

- Exigence de sécurité 15 : « *Le système doit vérifier que l'emprise connue du véhicule reste dans la Zone Navigable en Sécurité (ZNS).* »

  Nous introduisons cette exigence de sécurité pour éviter que l'emprise du véhicule sur la chaussée, c'est-à-dire la place qu'il prend sur la route, reste dans la zone navigable, moins une marge de sécurité. Le non-respect de cette exigence peut être due à de nombreuses fautes : matérielles sur les capteurs ou actionneurs, logicielles sur l'application de contrôle commande.

- Exigence de sécurité 16 : « *Il faut prendre des marges suffisantes pour la zone navigable en sécurité (ZNS).* »

  Cette condition de sécurité a été identifiée pour éviter que la zone navigable en sécurité, qui borne les positions permises au véhicule, soit trop près des obstacles dynamiques.

- Exigence de sécurité 17 : « *Le système doit vérifier ou garantir que l'on ne s'éloigne pas trop de la trajectoire cinématique demandée.* »

  Nous identifions cette exigence pour éviter un écart significatif de l'état cinématique estimé par le véhicule par rapport à la trajectoire cinématique.

- Exigence de sécurité 18 : « *Un composant critique auto-testable doit détecter toutes ses erreurs. Pour le véhicule autonome, tous les composants critiques doivent être auto-testables ou validés formellement de façon à n'avoir aucune faute interne.* »

  En effet, si nous n'avons aucune information sur l'état d'un ou plusieurs composants du véhicule, nous ne pouvons plus avoir confiance dans le comportement du véhicule. Les erreurs non détectées peuvent provoquer la perte du contrôle du véhicule.

- Exigence de sécurité 19 : « *Il faut tolérer les défaillances des composants ou garantir que les composants ne défaillent pas.* »

  Cette exigence a été identifiée pour remédier au cas d'une défaillance non détectée ou mal détectée de composants du véhicule.

- Exigence de sécurité 20 : « *Le système doit vérifier régulièrement lors du suivi de trajectoire la conformité de l'évolution dans le temps de la position du véhicule avec le suivi de trajectoire prévue. Si une incohérence apparaît, la trajectoire cinématique est tronquée et une nouvelle trajectoire doit être générée.* »

  Cette condition de sécurité permet de vérifier le véhicule suit correctement la trajectoire prévue. Si ce n'est pas le cas, pour des raisons environnementales, une nouvelle trajectoire doit immédiatement être calculée.

3.3.3 Récapitulatif

Le tableau 3.6 récapitule les résultats trouvés en effectuant la méthode d'analyse de risques HazOp-UML pour le cas d'utilisation *Suivre une trajectoire cinématique.* Nous avons identifié 46 déviations parmi 11 attributs et spécifié 20 exigences de sécurité. Le niveau de gravité associé à la majorité des déviations dans le système est 5. Dans ce niveau, des défaillances provoquent des commandes aberrantes et la perte du contrôle du véhicule peut être inévitable, même avec une reprise en main rapide.

3.4 Comparaison AMDEC/HazOp-UML vis-à-vis des exigences de sécurité

Dans cette partie, nous comparons les résultats obtenus par les deux méthodes étudiées et nous analysons les similitudes et différences observées. En effet :

- Les analyses AMDEC et HazOp-UML ont globalement le même objectif d'identifier respectivement les types de défaillances et les dangers possibles (les déviations) du système afin de les réduire par des techniques de sûreté de fonctionnement. Nous avons pu identifier les deux listes d'exigences de sécurité suivantes :
  - ◇ En appliquant l'AMDEC (11 exigences de sécurité) :
    - ✓ A1 : Le système doit être capable de détecter les défaillances de gel ou d'arrêt inopiné de l'application de contrôle-commande et de mettre le système en état sûr au besoin.
    - ✓ A2 : Le conducteur doit pouvoir reprendre la conduite en manuel par un autre moyen que les interfaces de l'application de contrôle-commande.
    - ✓ A3 : Le système doit détecter les erreurs fonctionnelles, telles que la désynchronisation des données, des commandes aberrantes, des erreurs de décision, et des mauvaises interprétations. En cas d'erreur, il faut assurer la mise en sécurité du système en levant des alarmes et en repassant en mode manuel.
    - ✓ A4 : Le système doit détecter et traiter les défaillances de communications sur le bus CAN : arrêt des communications ou pertes de données.
    - ✓ A5 : Des vérifications de cohérence entre les données CAN et les données de capteurs redondants du Safety-Bag pourraient être réalisées.

HazOp-UML	
Nombre d'attributs	*11*
Nombre de déviations	*46*
Gravité maximale	*5*
Nombre d'exigences de sécurité	*20*

Figure 3.6 – Synthèse de la méthode d'analyse de risques HazOp-UML sur le cas d'utilisation UC6 : « *Suivre une trajectoire cinématique* »

- ✓ A6 : Le système doit vérifier que les capteurs proprioceptifs longitudinaux (vitesse et accélération) fonctionnent correctement.
- ✓ A7 : Le système doit vérifier que les capteurs proprioceptifs latéraux (accéléromètre latéral et gyromètre) fonctionnent correctement.
- ✓ A8 : Le système doit vérifier que les capteurs extéroceptifs ne sont pas défaillants.
- ✓ A9 : Le système doit détecter les incohérences entre la commande d'accélération produite par l'application de contrôle-commande et la commande d'accélération effectivement réalisée par le moteur du véhicule.
- ✓ A10 : Le système doit détecter les incohérences entre la commande de freinage produite par l'application de contrôle-commande et la commande effectivement réalisée par les freins du véhicule.
- ✓ A11 : Le système doit détecter les incohérences entre la commande de direction produite par l'application de contrôle-commande et la commande effectivement réalisée par l'assistance à la direction.

◇ En appliquant l'HazOp (20 exigences de sécurité) :

1. H1 : Le système doit vérifier la vivacité de l'état cinématique estimé par le véhicule.
2. H2 : Le système doit vérifier ou garantir que l'état cinématique estimé par le véhicule est correct.
3. H3 : Le système doit s'assurer de la conformité de ce que perçoit le véhicule avec les cartes routières embarquées.
4. H4 : Le système doit vérifier la cohérence temporelle de l'état cinématique estimé par le véhicule.
5. H5 : Le système doit vérifier que les actionneurs et l'application de contrôle-commande fonctionnent correctement.
6. H6 : Le système doit détecter une trajectoire cinématique dangereuse.
7. H7 : Les conditions environnementales doivent être correctement identifiées par le système.
8. H8 : Le système doit vérifier que l'espace navigable est mis à jour à une fréquence suffisante.
9. H9 : Le système doit vérifier que la trajectoire cinématique est mise à jour à chaque rafraichissement de l'espace navigable.
10. H10 : Le système doit vérifier la vivacité de la trajectoire cinématique.
11. H11 : Le système doit vérifier la vivacité de l'espace navigable.

12. H12 : Le système doit anticiper les mouvements des véhicules et piétons et garder une distance de sécurité entre eux.
13. H13 : Le système doit vérifier que l'espace navigable est correct.
14. H14 : Le système doit disposer de suffisamment de capteurs redondants pour tolérer leurs défaillances matérielles. Une étude basée sur leurs taux de défaillance peut être nécessaire pour identifier les redondances nécessaires.
15. H15 : Le système doit vérifier que l'emprise connue du véhicule reste dans la Zone Navigable en Sécurité (ZNS).
16. H16 : Il faut prendre des marges suffisantes pour la zone navigable en sécurité (ZNS).
17. H17 : Le système doit vérifier ou garantir que l'on ne s'éloigne pas trop de la trajectoire cinématique demandée.
18. H18 : Un composant critique auto-testable doit détecter toutes ses erreurs. De plus, tous les composants critiques doivent être auto-testables ou validés formellement de façon qu'on va avoir aucune faute interne.
19. H19 : Il faut tolérer les défaillances des composants ou garantir que les composants ne défaillent pas.
20. H20 : Le système doit vérifier régulièrement lors du suivi de trajectoire la conformité de l'évolution dans le temps de la position du véhicule avec le suivi de trajectoire prévue. Si une incohérence apparaît, une nouvelle trajectoire doit être générée.

- La méthode AMDEC permet d'analyser les conséquences des défaillances des composants logiciels expérimentaux assurant le contrôle et la commande des véhicules autonomes robotisés, ainsi que des défaillances des éléments du véhicule. Les composants de l'application de contrôle-commande sont des composants inconnus dont nous ne savons rien sur leurs fonctionnalités ni leurs spécificités. Nous ne pouvons par conséquent les considérer que comme des boîtes noires. La méthode HazOp-UML est plus adaptée aux systèmes autonomes, du fait qu'elle met l'accent principalement sur le processus de conduite et sur les composants de l'environnement routier.

  Cela explique les deux conséquences suivantes :

  - ◇ Tout d'abord, il existe des exigences de sécurité similaires entre les deux techniques, qui aboutissent à des vérifications similaires ou équivalentes

telles que la vérification de la vivacité de l'application de contrôle-commande et de la cohérence temporelle.

En particulier l'exigence de sécurité N° 5 définie par l'HazOp « *il faut vérifier que l'application de contrôle-commande et les actionneurs (accélérateur, frein et direction assistée) fonctionnent correctement* » englobe les 5 exigences de sécurité suivantes identifiées en appliquant l'AMDEC :

- ✓ Le système doit détecter la défaillance de l'application de contrôle-commande et mettre le système en état sûr au besoin (exigence de sécurité N° 1 de l'AMDEC).
- ✓ Le système doit détecter les erreurs fonctionnelles de l'application de contrôle-commande (exigence de sécurité N° 3 de l'AMDEC).
- ✓ Le système doit détecter la cohérence entre la consigne produite par l'application de contrôle-commande et les actionneurs du véhicule (exigence de sécurité N° 9 de l'AMDEC liée à la défaillance de l'accélérateur).
- ✓ Le système doit détecter les incohérences entre la commande du frein et la puissance du freinage (exigence de sécurité N° 10 de l'AMDEC liée à la défaillance du frein).
- ✓ Le conducteur doit être alerté pour reprendre la conduite en manuel (exigence de sécurité N° 11 de l'AMDEC liée à la défaillance de la direction assistée).

◇ Les autres exigences de sécurité apparaissent seulement dans l'une des études et pas dans l'autre, bien que l'HazOp-UML demande une analyse de défaillance des composants critiques, et peut donc être d'un certain point de vue considérée comme englobant l'AMDEC.

- Il faut également noter que les résultats de l'analyse HazOp dépendent du choix de représentation et d'expression de la conception, et donc de la personne qui la réalise. En effet, le résultat peut varier suivant la manière dont sont choisis et exprimés les cas d'utilisation, leurs attributs et leurs dérivations. En particulier, nous ne pouvons pas garantir d'avoir identifié dans notre étude toutes les déviations possibles selon cette méthode.

- L'analyse AMDEC nous apparaît plus systématique, mais en contrepartie plus pauvre du point de vue déroulement de processus et les deux techniques nous semblent donc complémentaires.

3.5 Conclusion

Pour pallier les risques et certaines défaillances, nous avons étudié deux méthodes d'analyse de risques AMDEC et HazOp-UML qui permettent de spécifier les exigences de sécurité nécessaires pour la sécurité-innocuité du système. Ces méthodes analysent le système selon deux points de vue différents et nous paraissent complémentaires pour identifier le plus grand nombre possible d'exigences de sécurité. L'AMDEC permet d'identifier les différents modes de défaillances des composants internes du système, alors l'HazOp-UML se concentre sur le processus de conduite et l'environnement routier.

Dans le dossier de sécurité, l'analyse AMDEC doit être aussi complète que possible, et peut comprendre un très grand nombre de lignes suivant le niveau d'abstraction des composants étudiés. L'analyse HazOp quant à elle identifie un nombre considérable de déviations à partir de la grande diversité des situations de conduite et de l'environnement. Cependant, il est à noter que les résultats de ces analyses (et notamment de l'analyse HazOp-UML) dépendent grandement des compétences de l'analyste de sécurité qui doit être de préférence une autre personne que le développeur du système.

Nous rappelons que notre objectif dans ce manuscrit est d'identifier un nombre maximal des nécessités de sécurité. Ces nécessités de sécurité sont identifiées à partir des exigences de sécurité définies dans ce chapitre par les deux méthodes d'analyse de risques considérées, et seront implémentées et vérifiées en ligne par le Safety-Bag.

Le chapitre suivant précise comment sont obtenues les nécessités de sécurité à partir des exigences de sécurité.

Expression des nécessités de sécurité à partir des exigences de sécurité

Sommaire

Comme présenté dans les chapitres précédents, un composant Safety-Bag pourrait permettre d'améliorer significativement la sécurité-innocuité de véhicules autonomes. Ce dispositif détecte les anomalies des composants logiciels et matériels du système, et permet d'inhiber ou de forcer des commandes afin de remettre le système en état sûr. De telles actions peuvent se traduire par exemple par le rejet des commandes dangereuses, ou un repli en mode manuel accompagné de signaux d'alerte pour l'opérateur. Il est cependant difficile de prime abord de déterminer ses nécessités de sécurité, c'est-à-dire ce que ce composant Safety-Bag doit surveiller et comment il doit réagir.

Dans la première section de ce chapitre, nous allons tout d'abord préciser les moyens de détection, donc les sources et les natures des informations et des commandes à surveiller par le Safety-Bag, ainsi que les moyens d'action et d'inhibition de sécurité.

Ensuite, nous allons préciser dans la deuxième section les moyens d'implémentation des exigences de sécurité identifiées dans le chapitre précédent par les méthodes AMDEC et HazOp-UML. Si le Safety-Bag en fait partie, une nécessité de sécurité correspondante doit être rédigée. Cette dernière précise ce que le Safety-Bag doit observer, et comment il réagit face aux erreurs détectées.

L'identification des nécessités de sécurité nous mènera à identifier formellement les conditions de déclenchement de sécurité et à déterminer les marges de sécurité dans la section 3. Dans les sections 4 et 5, nous allons préciser respectivement la liste des interventions de sécurité, c'est-à-dire les inhibitions et les actions de sécurité enclenchées par le Safety-Bag pour chaque nécessité de sécurité, et la liste des nécessités de sécurité que nous pouvons implémenter sur notre véhicule autonome expérimental pour évaluer le comportement de notre système de sécurité Safety-Bag.

4.1 Analyse des moyens de détection et d'intervention du Safety-Bag

Le Safety-Bag est un dispositif indépendant de sécurité qui sert à détecter et tolérer des erreurs à un niveau systémique plutôt qu'en étudiant séparément chaque composant. Il est chargé de filtrer ou générer des commandes sur différents actionneurs et capable de comparer des informations provenant de différents capteurs. Il vérifie ainsi le respect des conditions de déclenchement de sécurité et détecte les anomalies en faisant le rétablissement soit en inhibant des commandes potentiellement dangereuses, soit en forçant des actions de sécurité pour remettre le système dans un état sûr. L'ensemble des inhibitions et des actions de sécurité est appelé *interventions de sécurité*.

Dans cette section, nous commençons tout d'abord par décrire les sources d'informations que peut obtenir le Safety-Bag pour pouvoir surveiller le système, détecter les anomalies et intervenir si nécessaire. Ensuite, nous identifions les différentes actions et inhibitions de sécurité que le Safety-Bag est susceptible d'effectuer sur le système.

4.1.1 Sources et natures d'informations fournies au Safety-Bag

Le Safety-Bag obtient généralement des informations sur le système de trois sources différentes :

- Des informations provenant des composants logiciels de l'application de contrôle-commande :
  - ◇ Les commandes envoyées par l'application de contrôle-commande : ce sont des informations sur lesquelles le Safety-Bag fera différents contrôles temporels et de vraisemblance pour s'assurer de leur intégrité.
  - ◇ L'état cinématique : De façon similaire, le Safety-Bag fera des contrôles de vraisemblance d'évolution géométrique.
  - ◇ Les horodatages des commandes, de l'état cinématique, de la trajectoire cinématique et de l'espace navigable : Le Safety-Bag utilisera ces informations pour vérifier la cohérence temporelle et la vivacité des données.

  Ces différentes données peuvent être erronées pour deux raison principales : la défaillance potentielle des capteurs à partir desquels elles sont générées et

les fautes internes à l'application de contrôle-commande qui va les manipuler et ou les générer. Le Safety-Bag permet de renforcer l'intégrité de ces données en réalisant différents contrôles temporels et de vraisemblance. Comme nous le verrons dans l'identification des nécessités de sécurité, il faut noter que certaines données vérifiées par le Safety-Bag ne sont pas les sorties de base de l'application de contrôle-commande, mais sont des sorties additionnelles nécessaires pour respecter les exigences de sécurité. Cela implique quelques modifications lors de la conception de l'application de contrôle-commande notamment concernant les flux de données du système.

- Des informations de capteurs proprioceptifs du véhicule, acheminées par le bus CAN [1] :
  - ◇ La vitesse du véhicule
  - ◇ l'accélération
  - ◇ la pression du frein
  - ◇ l'angle et la vitesse du volant

  Ces informations peuvent être considérées comme des informations fiables pour le cas de notre véhicule autonome expérimental, qui effectue de courtes expériences sous l'attention du conducteur. Par contre, pour un véhicule commercial, il faudra davantage assurer leur intégrité, par exemple en redondant les capteurs considérés.

- Des informations propres au Safety-Bag à partir de ses propres capteurs :
  - ◇ données de capteurs proprioceptifs redondants :
    - ✓ accélération latérale (accéléromètre)
    - ✓ vitesse de rotation (gyromètre)
    - ✓ vitesse et accélération longitudinales

    Ces informations permettront de détecter des défaillances dans les capteurs ou dans d'autres composants du véhicule en comparant leurs sorties.

    Notons aussi que notre véhicule expérimental ne dispose pas de capteurs redondants de vitesse et d'accélération longitudinales.

[1] Bus CAN : Le nombreux calculateurs et capteurs intelligents des véhicules modernes sont interconnectés par un ensemble de réseaux informatiques. Le réseau CAN est le plus fréquemment utilisé.

Dans nos véhicules, les réseaux CAN permettent d'accéder à l'ensemble des informations sur l'état du véhicule et de commander l'éclairage et les fonctions de confort (chauffage, climatisation, etc.).

◇ données sur l'état des actionneurs :

✓ la puissance appliquée sur le moteur électrique (ampèremètre et voltmètre)

✓ la puissance appliquée sur le moteur de la direction assistée électrique (ampèremètre et voltmètre)

Ces informations sont nécessaires pour diagnostiquer rapidement les défaillances des actionneurs. Notons cependant que nous ne disposons pas de ces informations sur notre véhicule expérimental.

4.1.2 Actions et inhibitions de sécurité possibles par le Safety-Bag

Afin de remettre le système dans un état sûr en cas d'anomalie, le Safety-Bag déclenche toujours des alertes sonores et/ou visuelles en même temps que toute autre intervention de sécurité. Généralement, le Safety-Bag combine les deux interventions de sécurité (action et inhibition) pour faire le rétablissement.

Dans notre étude, nous avons pu identifier les interventions de sécurité suivantes :

- Inhibitions de sécurité possibles sur les actionneurs du véhicule :

  ◇ Inhiber le freinage du véhicule. Bien que cette intervention puisse réduire les conséquences de défaillance dans de rares cas (arrêt inopportun causant des collisions arrière ou sur le côté), nous considérons que freiner permet dans la majorité des situations de réduire le danger (en particulier les conséquences d'un accident). Nous ne considérons donc pas cette intervention dans la suite de nos travaux.

  ◇ Inhibition de l'accélération (la consigne de l'accélérateur est mise à 0 Volts). Cette action empêche physiquement le véhicule d'augmenter sa vitesse, à moins qu'il ne se trouve dans une pente.

  ◇ Inhibition de la modification de l'angle au volant. Cette inhibition peut être dangereuse car elle empêche à la conduite autonome de modifier l'axe des roues du véhicule, ce qui risque fortement de l'amener à quitter la voie au bout d'un moment. Elle n'est donc pas considérée dans la suite de nos travaux.

- Actions de sécurité possibles sur les actionneurs du véhicule :

  ◇ Mettre en état sûr le véhicule : Afin de rester simple et sûr de fonctionnement, le Safety-Bag ne peut pas mettre en place des manœuvres

complexes. En particulier, générer et suivre une trajectoire pour s'arrêter sur le bas côté tenant compte de l'espace navigable de la route et de l'évolution des obstacles dynamiques est une fonctionnalité actuellement trop complexe pour être gérée de façon sûre de fonctionnement. Pour répondre à ces problèmes, nous pouvons imaginer un composant dédié à des manœuvres simples de rétablissement (ralentissement et déplacement sur la bande d'arrêt d'urgence) auquel le Safety-Bag donnerait la main à la place de l'application de contrôle-commande en cas de défaillance grave de cette dernière. Ce composant de rétablissement serait une version simplifiée et diversifiée de l'application de contrôle-commande, capable de conduire le véhicule de 10 à 20 secondes pour le mettre en état sûr. Cette simplification et diversification permettrait de limiter les fautes de développement de ce composant, et le Safety-Bag surveillerait le composant de mise en état sûr de la même façon qu'il surveille l'application de contrôle-commande.

◇ Mettre en état sûr le véhicule par les actions de pilote (passage en mode manuel) : Cette intervention peut être considérée comme une inhibition ultime du véhicule, puisqu'elle inhibe d'une certaine façon toutes les commandes de l'application de contrôle-commande.

◇ Déclenchement des alertes : Les alertes doivent être émises de manière sûre, cette émission est donc diversifiée par des alarmes visuelles et sonores. De plus, le Safety-Bag Supervisor relaie les alertes émises par le Safety-Bag Rules Checker pour diversifier la transmission des alertes en cas de défaillance de la connectique. Enfin, le Safety-Bag peut également émettre ses propres alertes notamment en cas de défaillances de Safety-Bag Rules Checker. La modulation du signal d'alerte sonore émis permet au conducteur d'évaluer la gravité de l'incident.

✓ En cas d'action du Safety-Bag, le Safety-Bag Rules Checker active l'alerte sonore 0.8 secondes par cycle de 0.9 secondes, tandis que le Safety-Bag Supervisor active le son 0.4 secondes par cycle de 1 seconde. Le signal perçu par le conducteur est la combinaison de ces deux sons. Lorsque les deux signaux couvrent toute la période, le signal est continu pendant quelques secondes.

✓ En cas d'inhibition par le Safety-Bag, le Safety-Bag Rules Checker active le son 0.5 secondes par cycle de 0.9 secondes, tandis que le Safety-Bag Supervisor l'active 0.3 secondes par cycle de 1 seconde. Le signal varie en durée entre 0.3 et 0.8 secondes séparé de silence de

durée variable entre 0.1 et 0.4 secondes.

✓ Si le Safety-Bag Supervisor détecte la défaillance du Safety-Bag Rules Checker par l'absence de heartbeat, il déclenche une alerte sonore pendant 3 secondes toutes les 6 secondes.

On remarque que quand la fréquence de répétition du signal est inférieure à une seconde, il y a une défaillance soit de l'un des calculateurs du Safety-Bag, soit de la transmission des alertes par l'un des calculateurs.

◇ Accélérer : Il est important de mentionner cette action de sécurité qui reste pertinente dans certains cas (par exemple pour éviter une collision arrière ou de côté avec un autre véhicule). Cependant, les situations qui mèneraient le véhicule à accélérer sont trop complexes à l'heure actuelle pour que le Safety-Bag soit capable de les traiter (tel est le cas des nécessités de sécurité H7 et H12 dans le tableau 4.5), et nous n'avons donc pas une telle intervention dans les nécessités de sécurité implémentables par le Safety-Bag.

◇ Freiner : Le Safety-Bag peut appliquer deux commandes de freinage, $f_{urgence}$ et f_{safe}, définies comme suit :

✓ $f_{urgence}$: C'est une tension de freinage utilisée par le Safety-Bag, pour ralentir immédiatement le véhicule. Sa valeur est un compromis pour ralentir rapidement la voiture sans risquer de provoquer une perte de contrôle, par exemple si ce freinage a lieu lors d'un virage (objectif de valeur maximale d'environ $0.4\vec{g}$).
Sur notre véhicule autonome expérimental, $f_{urgence} = 5.35Volts$ et assure une décélération de $2.5m.s^{-2}$. Cette décélération n'est pas énorme en regard de notre objectif, mais nous avons déterminé cette valeur de façon à éviter la mise en défaillance de la direction qui se produit fréquemment sur le simulateur si on applique un freinage plus important sur notre véhicule expérimental.

✓ f_{safe} : C'est une tension de freinage utilisée par le Safety-Bag pour arrêter progressivement le véhicule sans impact important sur le confort du passager (objectif de valeur maximale d'environ $0.2\vec{g}$).
Sur notre véhicule, $f_{safe} = 5Volts$ et assure une décélération entre 2 et $2.5m.s^{-2}$.

Si le Safety-Bag réagit à une nécessité de sécurité en appliquant un freinage f_{safe}, mais que l'application de contrôle-commande demande un freinage plus fort, c'est cette dernière commande qui sera appliquée. En effet, nous considérons dans cette situation que freiner plus fort, en

réduisant davantage l'énergie cinétique du véhicule, réduira probablement le risque.

- ◇ maintenir ou appliquer le couple volant : Comme il n'existe pas à l'heure actuelle de moyens sûrs pour garantir l'espace navigable du véhicule, et encore moins une trajectoire résultante, le Safety-Bag n'a que deux actions possibles de réduction des risques sur le couple volant :
  - ✓ Maintenir la position d'angle volant : Le véhicule continue tout droit ou suit un arc de cercle, suivant la commande originelle.
  - ✓ Réduire progressivement jusqu'à zéro la valeur du couple volant : C'est la seule mesure que peut prendre le Safety-Bag s'il ne connaît plus l'angle au volant (en cas de défaillance du capteur ou du bus CAN, par exemple).
    Ces interventions ne limitent les risques que pour une très courte période insuffisante pour que le Safety-Bag arrête le véhicule de manière autonome. Le conducteur doit donc reprendre le contrôle en mode manuel très rapidement.
  - ✓ Appliquer les commandes de l'application de contrôle commande pendant quelques secondes avant de maintenir la position volant. Par exemple, si la trajectoire cinématique n'est plus mise à jour, les consignes du couple volant demeurent néanmoins pertinentes pendant environ une à deux secondes. Elles sont donc appliquées par le Safety-Bag avant qu'il ne bloque la position d'angle volant.

4.2 Moyens d'implémentation des exigences de sécurité et élicitation informelle des nécessités de sécurité

Une fois que les exigences de sécurité et les moyens d'observation et d'action du Safety-Bag ont été identifiés, l'objectif est de spécifier des nécessités de sécurité informelles à partir des exigences de sécurité trouvées. Ces nécessités de sécurité sont composées des propriétés surveillées par le Safety-bag qui s'appellent les *conditions de déclenchement de sécurité*, ainsi que des comportements de rétablissement à réaliser qui s'appellent les *interventions de sécurité*.

Afin de pouvoir mieux comparer les deux méthodes utilisées, nous allons spécifier séparément dans cette section les nécessités de sécurité dérivées de l'analyse AMDEC

(dans les trois tableaux 4.1, 4.2 et 4.3), et celles dérivées de l'analyse HazOp (dans les trois tableaux 4.4, 4.5 et 4.6).

4.2.1 Nécessités de sécurité dérivées de l'AMDEC

Avant de déterminer les nécessités de sécurité, nous commençons par rechercher les moyens d'implémentation de chaque exigence de sécurité. Si l'exigence de sécurité est implémentable par le Safety-Bag, une nécessité de sécurité correspondante sera rédigée. En particulier, la nécessité de sécurité précise ce que le Safety-Bag doit observer, comment il identifie les défaillances à partir de ses moyens d'observation (ses propres capteurs ou ceux du système par exemple) et ce qu'il doit faire une fois qu'une défaillance est détectée.

4.2.1.1 Liste des exigences de sécurité implémentables par le Safety-Bag et nécessités de sécurité associées

Nous avons pu déduire 11 nécessités de sécurité à partir de 8 exigences de sécurité parmi les 16 identifiées dans le chapitre précédent avec l'analyse de risques AMDEC.

Ci-dessous, nous précisons pour chacune de ces 8 exigences de sécurité quelles sont les nécessités de sécurité dérivées :

- L'exigence de sécurité (A1 dans 4.1) : « *Le système doit être capable de détecter les défaillances de gel ou d'arrêt inopiné de l'application de contrôle-commande et de mettre le système en état sûr au besoin.* » est implémentable par le Safety-Bag en imposant à l'application de contrôle-commande de signaler périodiquement qu'elle a effectivement recalculé de nouvelles commandes.

  En cas de non-réponse de l'application de contrôle-commande, le Safety-Bag intervient en déclenchant les alarmes, en inhibant les dernières commandes de l'application de contrôle-commande telles que l'accélération et/ou en forçant une action de sécurité telle que le freinage du véhicule. Nous pouvons donc rédiger la nécessité de sécurité suivante : *Le Safety-Bag doit vérifier la vivacité de l'application de contrôle-commande. Si celle-ci n'envoie plus de commande, le Safety-Bag doit prévenir le conducteur et repasser en mode manuel ou arrêter le véhicule en allumant les feux de détresse.*

- L'exigence de sécurité (A3 dans 4.1) : « *Le système doit détecter les erreurs fonctionnelles, telles que la désynchronisation des données, des commandes aberrantes, des erreurs de décision, et des mauvaises interprétations.* » est partiellement implémentable par le Safety-Bag à partir des tests de cohérence de l'application de contrôle-commande. Nous avons élicité les trois nécessités

de sécurité suivantes, mais d'autres défaillances peuvent être trop complexes pour être analysées, détectées ou traitées par le Safety-Bag :

- ◇ *Le Safety-Bag doit vérifier la cohérence temporelle des commandes à appliquer. En cas d'incohérence temporelle, le Safety-Bag doit assurer la mise en sécurité du système en levant les alarmes et en repassant en mode manuel.*
- ◇ *Le Safety-Bag doit vérifier les bornes de vitesse, c'est-à-dire que la vitesse du véhicule ne dépasse pas une vitesse de sécurité de 50 km/h. Dans le cas contraire, le Safety-Bag déclenche les alarmes, bloque les commandes d'accélération de l'application de contrôle-commande et assure la reprise en mode de conduite manuelle.*

  Les incohérences temporelles peuvent notamment être caractérisées par :

  - ✓ des horodatages qui ne respectent pas l'ordre d'envoi des commandes,
  - ✓ des différences trop grandes entre l'horaodatage de génération de la commande et le temps de réception de celle-ci,
  - ✓ des horodatages dans le futur

  Actuellement, notre Safety-Bag effectue une vérification seulement pour des horodatages qui ne respectent pas l'ordre d'envoi croissant des commandes, mais les autres vérifications ne sont pas complexes à implémenter.
- ◇ *Le Safety-Bag doit vérifier les limites dynamiques, c'est-à-dire que l'accélération latérale reste inférieure à une valeur seuil. Dans le cas contraire, le Safety-Bag doit déclencher les alarmes et assurer la reprise en mode manuel.*

  Cette condition est aisément testable avec un capteur tel qu'un accéléromètre. Cependant pour nos expérimentations (chapitre 5, section 5.4.2), le véhicule ne bouge pas réellement et ce capteur est inopérant.

- L'exigence de sécurité (A4 dans 4.1) : « *Le système doit détecter et traiter les défaillances de communication sur le bus CAN :*
  - ◇ *arrêt des communications,*
  - ◇ *perte de données, etc.* »

  n'est implémentable partiellement par le Safety-Bag qu'en cas d'arrêt de communication. Le Safety-Bag peut dans ce cas vérifier régulièrement la circulation des trames sur le bus CAN. Pour cela, nous définissons la nécessité

de sécurité suivante : *Le Safety-Bag doit vérifier la circulation des trames sur le bus CAN. En cas d'absence de communication pendant une durée trop importante, le Safety-Bag doit alerter le conducteur et repasser en mode manuel.*

Pour les deux autres cas (perte ou altération des données), le Safety-Bag n'est pas capable de détecter ces défaillances. D'autres mécanismes de tolérance aux fautes pour les réseaux doivent être utilisés si les taux de défaillance liés à ces défaillances est trop élevé.

- L'exigence de sécurité (A5 dans 4.2) : « *Le système doit vérifier que les capteurs proprioceptifs longitudinaux (vitesse et accélération) fonctionnent correctement.* » est implémentable par le Safety-Bag. Nous élicitons alors la nécessité de sécurité comme suit :*Le Safety-Bag doit vérifier que les capteurs proprioceptifs longitudinaux fonctionnent correctement. En cas d'incohérence, le Safety-Bag doit freiner et alerter le conducteur.*

  Le Safety-Bag doit disposer ainsi de capteurs redondants pour vérifier la cohérence des données de vitesse et d'accélération longitudinales.

  Cependant, notre véhicule expérimental ne dispose pas de ces capteurs et ne peut donc pas implémenter cette nécessité de sécurité.

- L'exigence de sécurité (A7 dans 4.2) : « *Le système doit vérifier que les capteurs proprioceptifs latéraux (accéléromètre latéral et gyromètre) fonctionnent correctement.* » est implémentable par le Safety-Bag. Ce dernier dispose de capteurs redondants (l'accéléromètre et le gyromètre) par rapport aux capteurs liés à l'application de contrôle-commande. Nous élicitons alors la nécessité de sécurité comme suit : *Le Safety-Bag doit vérifier que les informations des capteurs provenant de l'application de contrôle-commande et les valeurs de vitesse angulaire et d'accélération latérale sont compatibles. Dans le cas contraire, le Safety-Bag déclenche les alarmes, inhibe l'accélération, freine modérément et applique la valeur d'angle volant provenant de l'application de contrôle-commande.*

- L'exigence de sécurité (A9 dans 4.2) : « *Le système doit détecter les incohérences entre la commande d'accélération produite par l'application de contrôle-commande et la commande d'accélération effectivement réalisée par le moteur du véhicule.* » est implémentable par le Safety-Bag à partir d'un diagnostic sur le moteur électrique du véhicule. Nous avons ainsi élicité la nécessité de sécurité suivante : *Le Safety-Bag doit vérifier que l'intensité dans le moteur électrique correspond à la valeur de la commande d'accélération fournie par l'application*

de contrôle commande. Dans le cas contraire, une alarme est levée et le Safety-Bag inhibe l'accélération et freine modérément.

Notre véhicule n'a pas été équipé pour l'instant des capteurs nécessaires pour tester cette nécessité de sécurité. Nous ne l'avons pas donc implémentée.

- L'exigence de sécurité (A10 dans 4.2) : « *Le système doit détecter les incohérences entre la commande de freinage produite par l'application de contrôle-commande et la commande effectivement réalisée par les freins du véhicule.* » est implémentable par le Safety-Bag à partir d'un diagnostic sur le circuit hydraulique du freinage. Nous avons défini alors la nécessité de sécurité comme suit : *Le Safety-Bag doit vérifier que la pression ne s'écarte pas de la valeur d'une fonction donnée de la commande du frein. Dans le cas contraire, le Safety-Bag déclenche les alarmes, empêche les commandes d'accélération et freine modérément.*

- L'exigence de sécurité (A11 dans 4.3) : « *Le système doit détecter les incohérences entre la commande de direction produite par l'application de contrôle-commande et la commande effectivement réalisée par l'assistance à la direction.* » est implémentable par le Safety-Bag. En effet, ce dernier est capable de détecter un problème au niveau de l'assistance de direction. De plus, il peut surveiller la cohérence des commandes et des mouvements du volant à travers un diagnostic sur le moteur de la direction. Deux nécessités de sécurité ont été définies dans ce cas :

  - ◇ *Le Safety-Bag doit vérifier que les angles volant évoluent dans le sens prévu selon le couple volant appliqué. Dans le cas contraire, une alerte doit être déclenchée vers le conducteur et le Safety-Bag doit mettre le véhicule en état sûr en passant en mode de conduite manuelle.*
  - ◇ *Le Safety-Bag doit vérifier que l'intensité sur le moteur de direction correspond aux consignes envoyées par l'application de contrôle-commande. En cas de non-correspondance, le Safety-Bag doit alerter le conducteur et doit mettre le système en état sûr en redonnant la main au conducteur.*

4.2.1.2 Liste des exigences de sécurité non implémentables par le Safety-Bag

Toutes les exigences de sécurité ne sont pas implémentables par le Safety-Bag. Certaines nécessitent un autre moyen pour leur réalisation. Ces exigences de sécurité sont les suivantes :

- L'exigence de sécurité (A2 dans 4.1) : « *Le conducteur doit pouvoir reprendre la conduite en manuel par un autre moyen que les interfaces de l'application de contrôle-commande.* » n'est pas implémentable par le Safety-Bag. En effet, le Safety-Bag n'est pas capable de détecter une défaillance matérielle liée par exemple à l'écran ou au clavier. Dans ce cas, cette exigence de sécurité peut être réalisée par un autre moyen. Un bouton d'arrêt de process est par exemple utilisé dans notre véhicule expérimental pour pallier ce problème.

- L'exigence de sécurité (A5 dans 4.2) : « *Des vérifications de cohérence entre les données CAN et les données de capteurs redondants du Safety-Bag pourraient être réalisées.* » n'est pas implémentables par le Safety-Bag. En effet, le Safety-Bag n'est pas capable de détecter ces incohérences de données transmises par le bus CAN. Les capteurs permettant ces vérifications de cohérence ne sont pas installés sur notre véhicule.

- L'exigence de sécurité (A8 dans 4.2) : « *Le système doit vérifier que les capteurs extéroceptifs ne sont pas défaillants.* » n'est pas implémentable par le Safety-Bag. En effet, ce dernier n'est pas compétent pour détecter une défaillance au niveau des capteurs extéroceptifs liés à l'environnement. En effet, pour surveiller les éléments de l'environnement et pour détecter la présence d'obstacles, de piétons ou de panneaux de signalisation, il aurait besoin de mécanismes de reconnaissance de situations, qui le rendraient très complexe et diminueraient la confiance que l'on peut porter dans le développement et la validation d'un comportement correct. Dans ce cas, l'exigence de sécurité nécessite un mécanisme complémentaire pour être assurée. Dans notre cadre expérimental, nous proposons que le conducteur soit là pour contrôler ce qui se passe dans l'environnement et soit capable de prendre la main en cas de situation dangereuse. Il faut cependant noter que pour les niveaux d'autonomie 4 et 5, cette solution ne peut pas convenir, et il n'en existe actuellement pas d'autre satisfaisante à notre connaissance.

Tableau 4.1 – Exigences et nécessités de sécurité issues de l'analyse AMDEC (page1)

Élément	**Type de Défaillance**	**Exigences de sécurité**	**Moyens d'implémentation d'exigence de sécurité**	**Nécessité de sécurité en langage naturel**	**Commentaires**	**Réf**
Matériel de l'AppCC	Panne bloquée	Le système doit être capable de détecter les défaillances de gel ou d'arrêt inopiné de l'application de contrôle-commande et de mettre le système en état sûr au besoin.	Implémentable par le Safety-Bag, qui : • filtre les commandes entre l'AppCC et les actionneurs. • est informé des mises à jour des commandes (heartbeat, réception des messages réseau, etc.). En cas de non-réponse de l'AppCC, le Safety-Bag peut : • déclencher les alarmes et les feux de détresse et redonner la main au pilote, • inhiber les commandes appliquées par l'AppCC (ne pas accélérer, ne pas tourner d'avantage le volant, etc.), • freiner si pertinent/possible.	Le Safety-Bag doit vérifier la vivacité de l'application de contrôle-commande. Si celle-ci n'envoie plus de commande, le Safety-Bag doit prévenir le conducteur et repasser en mode manuel et arrête le véhicule lentement sur le côté de la route en allumant les feux de détresse.	Sans Safety-Bag, cette défaillance peut provoquer un comportement catastrophique car les consignes sont maintenues.	A1
	Clavier/écran	Le conducteur doit pouvoir reprendre la conduite en manuel par un autre moyen que les interfaces de l'application de contrôle-commande.	Un bouton (BAP) dédié permet le passage en mode manuel.	–	Le Safety-Bag ne dispose pas des informations nécessaires pour surveiller ce périphérique, et la présence du bouton de passage en mode manuel est suffisante.	A2
Logiciel de l'AppCC	Erreur Fonctionnelle	Le système doit détecter les erreurs fonctionnelles, telles que la désynchronisation des données, des commandes aberrantes, des erreurs de décision, et des mauvaises interprétations. En cas d'erreur, il faut assurer la mise en sécurité du système en levant des alarmes et en repassant en mode manuel.	Mécanismes de diagnostic de l'AppCC et tests de cohérence par le Safety-Bag	Le Safety-Bag doit vérifier : • la cohérence temporelle des commandes à appliquer et certaines informations de l'AppCC. En cas d'incohérence temporelle, le Safety-Bag doit assurer la mise en sécurité du système en levant les alarmes et en repassant en mode manuel. • Les bornes de vitesse : il doit vérifier si la vitesse du véhicule ne dépasse pas une vitesse de sécurité de 50 km/h. Dans le cas contraire, le Safety-Bag déclenche les alarmes, bloque les commandes d'accélération de l'application de contrôle-commande et assure la reprise en mode de conduite manuelle. • les limites dynamiques : l'accélération latérale doit rester inférieure à une valeur seuil. Dans le cas contraire, le Safety-Bag doit déclencher les alarmes et assurer la reprise en mode manuel.	Le Safety-Bag ne pourra détecter qu'un sous ensemble des défaillances de l'AppCC ou plus tardivement leurs conséquences sur la dynamique du véhicules.	A3
CAN	déconnecté ou fil coupé	Le système doit détecter et traiter les défaillances de communication sur le bus CAN : arrêt des communications, pertes de données, altération des données, ou désynchronisation.	Le Safety-Bag peut être connecté sur le CAN véhicule. Il peut alors vérifier la circulation régulière des trames les plus importantes. Un fois que le diagnostic véhicule détecte une anomalie, une alarme véhicule est déclenchée.	Le Safety-Bag doit vérifier la circulation des trames du bus CAN. En cas d'absence de communication pendant une durée trop importante, le Safety-Bag doit alerter le pilote et repasser en mode manuel.	Le Safety-Bag ne peut détecter ainsi que l'arrêt des communications sur le bus CAN. Si nécessaire, d'autres mécanismes doivent être utilisés pour les deux autres types de défaillance.	A4

Tableau 4.2 – Exigences et nécessités de sécurité issues de l'analyse AMDEC (page2)

Élément	**Type de Défaillance**	**Exigences de sécurité**	**Moyens d'implémentation d'exigence de sécurité**	**Nécessité de sécurité en langage naturel**	**Commentaires**	**Réf**
CAN	altération des données transmises	Des vérifications de cohérence entre les données CAN et les données de capteurs redondants du Safety-Bag pourraient être réalisées.	Une redondance des capteurs permet d'effectuer ces vérifications de données.	–	Les capteurs permettant ces vérifications de cohérence des données transmises par le bus CAN ne sont pas installés actuellement dans notre véhicule.	A5
Capteurs longitudinaux du véhicule	capteur de vitesse longitudinale défaillant	Le système doit vérifier que les capteurs proprioceptifs longitudinaux (vitesse et accélération) fonctionnent correctement.	Le Safety-Bag doit disposer de capteurs de vitesse longitudinale et d'accélération longitudinale redondants pour pouvoir faire un test de cohérence avec les données des capteurs du véhicule.	Le Safety-Bag doit vérifier que les capteurs proprioceptifs longitudinaux (capteur de vitesse et accélération longitudinales) fonctionnent correctement. En cas d'incohérence, le Safety-Bag doit freiner et alerter le conducteur.	Il faut noter que notre véhicule expérimental ne dispose pas de ces capteurs et ne peut donc pas implémenter cette nécessité de sécurité.	A6
Capteurs proprioceptifs (gyromètre + accéléromètre) du véhicule	capteur défaillant	Le système doit vérifier que les capteurs proprioceptifs latéraux (accéléromètre latéral et gyromètre) fonctionnent correctement.	Si le Safety-Bag dispose d'accéléromètre et de gyromètre redondants, il peut vérifier les données des capteurs proprioceptifs défaillants.	Le Safety-Bag doit vérifier que les capteurs proprioceptifs latéraux (accélération latérale et gyromètre) fonctionnent correctement. En cas d'incohérence, le Safety-Bag doit freiner et alerter le conducteur.	Dans notre simulateur VIL VILAD, les capteurs (gyromètre et accéléromètre) ne donnent pas d'informations ce qui ne permet pas de valider cette nécessité de sécurité.	A7
Capteurs extéroceptifs	capteur ou connectivité	Le système doit vérifier que les capteurs extéroceptifs ne sont pas défaillants.	Surveillance conducteur	–	Notre Safety-Bag n'est pas compétent pour assurer cette exigence de sécurité.	A8
actionneur « accélérateur »	bloqué à 0	Le système doit détecter les incohérences entre la commande d'accélération produite par l'application de contrôle-commande et la commande d'accélération effectivement réalisée par le moteur du véhicule.	Un ampèremètre sur le moteur électrique du véhicule permet d'effectuer ce diagnostic.	Le Safety-Bag doit vérifier que l'intensité dans le moteur électrique correspond à la valeur de la commande d'accélération fournie par l'application de contrôle commande. Dans le cas contraire, une alarme est levée et le Safety-Bag inhibe l'accélération et freine modérément.	Ce capteur n'est pas encore installé sur notre véhicule.	A9
actionneur « frein »	circuit de freinage et/ou actionnement de la pédale du frein	Le système doit détecter les incohérences entre la commande de freinage produite par l'application de contrôle-commande et la commande effectivement réalisée par les freins du véhicule.	Un capteur sur le circuit hydraulique permet d'assurer le diagnostic.	Le Safety-Bag doit vérifier que la pression ne s'écarte pas de la valeur d'une fonction donnée de la commande du frein. Dans le cas contraire, le Safety-Bag déclenche les alarmes, empêche les commandes d'accélération et freine modérément.	Bien que le frein soit redondé sur les voitures, l'actionneur robotisé du frein sur notre véhicule expérimental n'est ni redondé ni particulièrement fiable et devrait être surveillé avec attention.	A10

Tableau 4.3 – Exigences et nécessités de sécurité issues de l'analyse AMDEC (page3)

Élément	**Type de Défaillance**	**Exigences de sécurité**	**Moyens d'implémentation d'exigence de sécurité**	**Nécessité de sécurité en langage naturel**	**Commentaires**	**Réf**
actionneur « assistance de direction »	défaillance de la DAE	Le système doit détecter les incohérences entre la commande de direction produite par l'application de contrôle-commande et la commande effectivement réalisée par l'assistance à la direction.	Le Safety-Bag surveille la cohérence de mouvements de volant et les consignes. Par ailleurs, on envisage un capteur de l'intensité de courant sur le moteur de la direction.	• Le Safety-Bag doit vérifier que les angles volant évoluent dans le sens prévu selon le couple volant appliqué. Dans le cas contraire, une alerte doit être transmise au conducteur et le Safety-Bag doit mettre le véhicule en état sûr en passant en mode de conduite manuelle. • Le Safety-Bag doit vérifier que l'intensité sur le moteur de direction correspond aux consignes envoyées par l'application de contrôle-commande. En cas de non-correspondance, le Safety-Bag doit alerter le conducteur et doit mettre le système en état sûr en redonnant la main au conducteur.	• Le capteur d'intensité n'est pas installé. • La DAE standard de la Fluence a une loi de commande appliquée et que nous ne connaissons pas. Ce qui ne nous permet pas d'implanter concrètement la deuxième nécessité de sécurité.	A11

4.2.2 Nécessités de sécurité dérivées de l'analyse HazOp

De la même manière que pour les exigences de sécurité dérivées de l'AMDEC, nous identifions tout d'abord à partir de chaque exigence de sécurité dérivée de l'HazOp quels moyens de sûreté de fonctionnement sont capables de l'assurer. Si le Safety-Bag en fait partie, nous élicitons une nécessité de sécurité correspondante. Nous présentons dans cette section tout d'abord les exigences de sécurité amenant à une nécessité de sécurité, avant de détailler quelles exigences de sécurité ne peuvent pas être implémentées par le Safety-Bag dans notre système.

4.2.2.1 Liste des exigences de sécurité implémentables de sécurité par le Safety-Bag et nécessités de sécurité dérivées

Nous avons pu déduire 14 nécessités de sécurité à partir des 20 exigences de sécurité identifiées par l'analyse de risques HazOp. Les 14 nécessités de sécurité sont les suivantes :

- L'exigence de sécurité (H1 dans 4.4) : « *Le système doit vérifier la vivacité de l'état cinématique estimé par le véhicule.* » est implémentable par le Safety-Bag qui doit vérifier que l'état cinématique est régulièrement rafraichi. Nous avons donc défini la nécessité de sécurité suivante : *Le Safety-Bag doit vérifier la vivacité (la mise à jour) de l'état cinématique régulièrement. Si l'état cinématique n'est pas mis à jour, le Safety-Bag intervient en déclenchant les alarmes, en inhibant des commandes de l'accélération et/ou en forçant une action de sécurité telle que le freinage du véhicule et le maintien de l'angle volant après un certain délai.*

  Cela implique que l'application de contrôle-commande informe régulièrement le Safety-Bag de son estimation de l'état cinématique, et donc une modification du flot de données dans la conception du système.

- L'exigence de sécurité (H2 dans 4.4) : « *Le système doit vérifier ou garantir que l'état cinématique estimé par le véhicule est correct.* » est implémentable par le Safety-Bag à partir des tests de cohérence sur l'état cinématique.

  Nous rappelons que l'état cinématique estimé par le véhicule est constitué de sa vitesse (v), de sa position géographique (x,y) et de son orientation (cap). Les tests de cohérence consistent à vérifier que chaque variable (x, y, v et cap) est réaliste par rapport aux valeurs précédentes d'après un modèle simplifié de la dynamique du véhicule. La nécessité de sécurité définie dans ce cas est la suivante : *Le Safety-Bag doit vérifier la cohérence de l'évolution de l'état cinématique, c'est-à-dire la cohérence des valeurs liées à chaque*

variable de l'état cinématique par rapport aux valeurs précédentes. En cas d'incohérence de l'état cinématique, le Safety-Bag intervient en levant les alarmes, en inhibant l'accélération et en freinant modérément le véhicule. Il doit également maintenir l'angle volant.

- L'exigence de sécurité (H4 dans 4.4) : « *Le système doit vérifier la cohérence temporelle de l'état cinématique estimé par le véhicule.* » est implémentable par le Safety-Bag. Ce dernier doit vérifier l'horodatage de l'état cinématique. Nous avons élicité ainsi la nécessité de sécurité suivante : *Le Safety-Bag doit vérifier la cohérence temporelle de l'état cinématique. Si l'état cinématique n'est pas temporellement cohérent, le Safety-Bag doit lever les alarmes, inhiber l'accélération, freiner modérément le véhicule et maintenir l'angle volant.*

- L'exigence de sécurité (H5 dans 4.4) : « *Le système doit vérifier que les actionneurs et l'application de contrôle-commande fonctionnent correctement.* » est partiellement implémentable par le Safety-Bag qui doit vérifier que l'application de contrôle-commande et les actionneurs ne défaillent pas. Dans ce cas nous avons défini 7 nécessités de sécurité que le Safety-Bag peut assurer :

  - ◇ 4 nécessités de sécurité liées au bon fonctionnement de l'application de contrôle-commande :

    - ✓ *Le Safety-Bag doit vérifier la vivacité de l'application de contrôle-commande. Si celle-ci n'est pas vivante, le Safety-Bag doit prévenir le conducteur et repasser en mode manuel et arrête le véhicule en allumant les feux de détresse* (même nécessité de sécurité A1 définie dans 4.1).
    - ✓ *Le Safety-Bag doit vérifier la cohérence temporelle de l'application de contrôle-commande. En cas d'incohérence temporelle, le Safety-Bag doit assurer la mise en sécurité du système en levant les alarmes et en repassant en mode manuel* (même nécessité de sécurité A3 définie dans 4.1).
    - ✓ *Le Safety-Bag doit vérifier les bornes de vitesse. Dans le cas contraire,le Safety-Bag déclenche les alarmes, bloque les commandes d'accélération de l'application de contrôle-commande et assure la reprise en mode manuel* (même nécessité de sécurité A3 définie dans 4.1).
    - ✓ *Le Safety-Bag doit vérifier le contrôle dynamique du système : Il doit vérifier que l'accélération latérale reste inférieure à une valeur seuil. Dans le cas contraire, le Safety-Bag doit déclencher les alarmes et*

assurer la reprise en mode manuel (même nécessité de sécurité A3 définie dans dans 4.1).

◇ 3 nécessités de sécurité liées au bon fonctionnement des actionneurs :

✓ *Le Safety-Bag doit vérifier que l'accélérateur fonctionne correctement. En cas de défaillance, le Safety-Bag lève une alarme, inhibe l'accélération et freine modérément* (même nécessité de sécurité A8 définie dans 4.2).

✓ *Le Safety-Bag doit vérifier que le frein fonctionne correctement. En cas de défaillance, le Safety-Bag déclenche les alarmes, empêche les commandes d'accélération et freine modérément* (même nécessité de sécurité A9 définie dans 4.2).

✓ *Le Safety-Bag doit vérifier que les composants matériels contrôlant l'axe de direction du véhicule fonctionnent correctement. Dans le cas contraire, le Safety-Bag déclenche une alerte vers le conducteur et doit mettre le véhicule en état sûr en passant en mode de conduite manuelle.*

Cette nécessité de sécurité est générale en analysant l'analyse de risques HazOp, mais a été raffinée dans l'étude AMDEC en deux nécessités de sécurité (A10 dans 4.3).

Les 7 nécessités de sécurité dans ce cas concernent les composants internes de notre système. De manière logique, une analyse des défaillances de l'application de contrôle commande et des actionneurs nous redonnent la partie des nécessités de sécurité déterminées à partir de l'AMDEC sur ces composants.

- L'exigence de sécurité (H6 dans 4.5) : « *Le système doit vérifier la vivacité de la trajectoire cinématique.* » est implémentable par le Safety-Bag qui peut vérifier la mise à jour régulière de la trajectoire cinématique. Cela demande une modification du flux de données de l'application contrôle commande, qui doit donc envoyer cette trajectoire au Safety-Bag. Nous avons élicité dans ce cas la nécessité de sécurité suivante : *Le Safety-Bag doit vérifier la vivacité (la mise à jour) de la trajectoire cinématique. Dans le cas contraire, une alarme sera déclenchée et le Safety-Bag inhibe l'accélération, freine et maintient la position d'angle volant après un certain temps.*

- L'exigence de sécurité (H9 dans 4.5) : « *Le système doit vérifier que l'espace navigable est mis à jour à une fréquence suffisante.* » est implémentable par le Safety-Bag qui doit vérifier que l'espace navigable est régulièrement généré par l'application de contrôle-commande. Pour cela, nous avons défini la nécessité

de sécurité suivante : *Le Safety-Bag doit vérifier la vivacité (la mise à jour) de l'espace navigable régulièrement. Dans le cas contraire, le Safety-Bag inhibe l'accélération, freine et maintient la position d'angle volant après un certain temps.*

- L'exigence de sécurité (H11 dans 4.5) : « *Le système doit vérifier que la trajectoire cinématique est mise à jour à chaque rafraichissement de l'espace navigable.* » est implémentable par le Safety-Bag qui peut vérifier que la trajectoire cinématique est mise à jour après chaque évolution de l'espace navigable. La nécessité de sécurité élicitée dans ce cas est la suivante : *Le Safety-Bag doit vérifier que l'horodatage de la trajectoire cinématique est plus récent que l'horodatage de l'espace navigable. Dans le cas contraire, le safety-Bag déclenche les alarmes, inhibe l'accélération et force des actions de sécurité telles que le freinage et le maintien d'angle volant.*

4.2.2.2 Liste des exigences de sécurité non implémentables par le Safety-Bag

Dans le cas de l'analyse HazOp-UML, les exigences de sécurité qui ne peuvent pas être assurées par le Safety-Bag sont les suivantes :

- L'exigence de sécurité (H3 dans 4.4) : « *Le système doit s'assurer de la conformité de ce que perçoit le véhicule avec les cartes routières embarquées.* » n'est pas implémentable par le Safety-Bag . La mise en place de cette exigence de sécurité dans le Safety-Bag est trop complexe pour assurer la confiance en son bon fonctionnement. En fait, nous ne connaissons pas à l'heure actuelle de moyens sûrs de fonctionnement pour assurer cette exigence.

- L'exigence de sécurité (H7 dans 4.5) : « *Le système doit détecter une trajectoire cinématique dangereuse.* » n'est pas implémentable par le Safety-Bag. Ce dernier ne dispose pas d'informations sur l'environnement suffisamment sûres pour pouvoir prendre des décisions. Un composant logiciel diversifié de l'application de contrôle commande et chargé de vérifier l'innocuité de ses trajectoires pourrait être une solution alternative pour implémenter ces exigences.

- L'exigence de sécurité (H8 dans 4.5) : « *Les conditions environnementales doivent être correctement identifiées par le système.* » n'est pas implémentable par le Safety-Bag. Ce dernier n'a pas la capacité pour détecter les conditions liées à l'environnement. Encore une fois, il n'existe à notre connaissance aucun

mécanisme sûr de fonctionnement permettant d'implémenter la fonctionnalité de reconnaissance de situation.

- L'exigence de sécurité (H12 dans 4.5) : « *Le système doit anticiper les mouvements des véhicules et piétons et garder une distance de sécurité avec eux.* » n'est pas implémentable par le Safety-Bag. Assurer cette exigence de sécurité nécessite des informations que le Safety-Bag ne peut pas obtenir avec un niveau de sécurité suffisant. A notre connaissance, il n'existe actuellement pas de solution pour traiter la détection d'obstacles par un véhicule autonome avec un niveau de sécurité (SIL) suffisant.

- Les 4 exigences de sécurité :

  ◇ *« H13 dans 4.5 : Le système doit vérifier que l'espace navigable est correct.*

  ◇ *H15 dans 4.6 : Le système doit vérifier que l'emprise connue du véhicule reste dans la Zone Navigable en Sécurité (ZNS).*

  ◇ *H16 dans 4.6 : Il faut prendre des marges suffisantes pour la zone navigable en sécurité (ZNS).*

  ◇ *et H20 dans 4.6 : Le système doit vérifier régulièrement lors du suivi de trajectoire la conformité de l'évolution dans le temps de la position du véhicule avec le suivi de trajectoire prévue. Si une incohérence apparaît, une nouvelle trajectoire doit être générée.* »

  ne sont pas implémentables par le Safety-Bag. Ces exigences doivent être réalisées par un composant logiciel diversifié de l'application de contrôle-commande. Ce composant pourrait effectuer certaines vérifications trop complexes à mettre en œuvre par le Safety-Bag.

- L'exigence de sécurité (H14 dans 4.5) : « *Le système doit disposer de suffisamment de capteurs redondants pour tolérer leurs défaillances matérielles. Une étude basée sur leurs taux de défaillance peut être nécessaire pour identifier les redondances nécessaires.* » est également non implémentable par le Safety-Bag. Cette exigence est typiquement liée à la redondance et la diversification de capteurs, d'ailleurs nécessaire au Safety-Bag pour certaines de ses nécessités de sécurité.

- L'exigence de sécurité (H17dans 4.6) :« *Le système doit vérifier ou garantir que l'on ne s'éloigne pas trop de la trajectoire cinématique demandée.* » n'est pas implémentable par le Safety-Bag. Ce dernier n'a pas les compétences suffisantes pour pouvoir garantir la localisation du véhicule et donc sa distance à la

trajectoire cinématique. Connaître précisément la localisation d'un véhicule autonome reste un verrou technologique dans certaines situations, où par exemple un GPS centimétrique n'arrive pas à donner des résultats (tunnel, présence de grands immeubles ou d'arbres, etc.).

- L'exigence de sécurité (H18 dans 4.6) : « *Un composant critique auto-testable doit détecter toutes ses erreurs. De plus, tous les composants critiques doivent être auto-testables ou validés formellement de façon à n'avoir aucune faute interne.* » n'est pas implémentable par le Safety-Bag. Ce dernier doit cependant être auto-testable. L'analyse de sécurité du chapitre suivant montre dans notre cas qu'il est capable de détecter une simple erreur dans son état et d'en alerter le conducteur.

- L'exigence de sécurité (H19 dans 4.6) : « *Il faut tolérer les défaillances des composants ou garantir que les composants ne défaillent pas.* » n'est pas implémentable par le Safety-Bag.

  Cette exigence de sécurité est trop générale, bien qu'elle rappelle la nécessité pour tous les composants critiques, y compris le Safety-Bag, d'être tolérants aux fautes. Le Safety-Bag n'est pas compétent pour vérifier pour l'instant cette exigence de sécurité. En effet, cela nécessite soit de réaliser une analyse AMDEC des parties critiques du véhicule autonome soit de détailler davantage les cas d'utilisation, en particulier les attributs (conditions/invariants). Il s'agit donc d'une méta-nécessité de sécurité derrière laquelle peut-être exigée d'autres analyses de risque complémentaires, telles que des arbres de fautes. Ce problème montre bien la complémentarité nécessaire de ces différentes analyses, point principal de ce mémoire.

Tableau 4.4 – Exigences et nécessités de sécurité issues de l'analyse HazOp (page1)

Réf	**Exigences de sécurité**	**Moyens d'implémentation d'exigence de sécurité**	**Nécessité de sécurité en langage naturel**	**Commentaires**
H1	Le système doit vérifier la vivacité de l'état cinématique estimé par le véhicule.	Le Safety-Bag peut vérifier que l'état cinématique est mis à jour.	Le Safety-Bag doit vérifier la vivacité (la mise à jour) de l'état cinématique régulièrement. Si l'état cinématique n'est pas mis à jour, le Safety-Bag intervient en déclenchant les alarmes, en inhibant des commandes de l'accélération et/ou en forçant une action de sécurité telle que le freinage du véhicule et le maintien de l'angle volant après un certain délai.	L'état cinématique fourni au Safety-Bag doit être rafraichi régulièrement, c'est-à-dire que l'application de contrôle-commande informe régulièrement le Safety-Bag de son estimation de l'état cinématique, et donc une modification du flot de données dans la conception du système.
H2	Le système doit vérifier ou garantir que l'état cinématique estimé par le véhicule est correct.	Redondance sur les mécanismes et/ou contrôle de cohérence dans le Safety-Bag.	Le Safety-Bag doit vérifier la cohérence de l'évolution de l'état cinématique, c'est-à-dire la cohérence des valeurs liées à chaque variable de l'état cinématique par rapport aux valeurs précédentes. Le Safety-Bag doit réaliser alors les tests de cohérence suivants sur l'état cinématique : • La vitesse par rapport à la vitesse précédente est réaliste. • La position par rapport à la position précédente est réaliste. • L'orientation par rapport à l'orientation précédente est réaliste. Dans le cas d'incohérence dimensionnelle de l'état cinématique, le Safety-Bag intervient en levant les alarmes, en inhibant l'accélération et en freinant modérément le véhicule. Il doit également maintenir l'angle volant.	• Redondance des mécanismes de localisation coûteuse. Des vérifications de cohérence sont plus facile à mettre en œuvre.
H3	Le système doit s'assurer de la conformité de ce que perçoit le véhicule avec les cartes routières embarquées.	Comparaison entre ce que perçoivent le véhicule et les cartes routières embarquées.	–	Trop complexe pour une mise en place dans le Safety-Bag
H4	Le système doit vérifier la cohérence temporelle de l'état cinématique estimé par le véhicule.	Cette exigence de sécurité est implémentable par le Safety-Bag.	Le Safety-Bag doit vérifier la cohérence temporelle de l'état cinématique. Si l'état cinématique n'est pas temporellement cohérent, le Safety-Bag doit lever les alarmes, inhiber l'accélération, freiner modérément le véhicule et maintenir l'angle volant.	L'état cinématique doit être horodaté.
H5	Le système doit vérifier que les actionneurs et l'application de contrôle-commande fonctionnent correctement.	Mécanismes de diagnostic des actionneurs et de l'AppCC et mise en sécurité. Une partie est réalisée par le Safety-Bag.	1. Le Safety-Bag doit vérifier pour les actionneurs que : • le frein fonctionne • l'accélérateur fonctionne • la direction fonctionne 2. Le Safety-Bag doit vérifier pour l'application de contrôle-commande : • la vivacité • la cohérence temporelle • les bornes de vitesse • les limites dynamiques du véhicule Dans le cas contraire, le Safety-Bag déclenche les alertes, inhibe l'accélération et force des actions de sécurité soit en freinant ou en maintenant l'angle volant.	1. idem A9, A10 et A11 2. Le Safety-Bag ne pourra détecter qu'un sous-ensemble des défaillances de l'AppCC.

Tableau 4.5 – Exigences et nécessités de sécurité issues de l'analyse HazOp (page2)

Réf	**Exigences de sécurité**	**Moyens d'implémentation d'exigence de sécurité**	**Nécessité de sécurité en langage naturel**	**Commentaires**
H6	Le système doit vérifier la vivacité de la trajectoire cinématique.	Le Safety-Bag peut vérifier que la trajectoire cinématique est mise à jour et il peut faire des tests de cohérence de la trajectoire cinématique.	Le Safety-Bag doit vérifier la vivacité (la mise à jour) de la trajectoire cinématique. Dans le cas contraire, une alarme est déclenchée et le Safety-Bag inhibe l'accélération, freine et maintient la position d'angle volant après un certain temps.	La vérification par le Safety-Bag est efficace pour la vivacité.
H7	Le système doit détecter une trajectoire cinématique dangereuse.	Nécessité d'un modèle logiciel qui vérifie à chaque pas que la trajectoire cinématique reste dans la partie sûre de l'espace navigable.	_	Ce module logiciel est nécessaire, mais il parait compliqué de l'implanter dans le Safety-Bag. En effet, ce dernier travaille sur les commandes et non sur la trajectoire cinématique, sinon il serait trop complexe.
H8	Les conditions environnementales doivent être correctement identifiées par le système.	Redondance des capteurs d'environnement et/ou communication avec l'infrastructure.	_	Le Safety-Bag n'est pas compétent pour résoudre ce problème d'environnement, mais il peut enregistrer certaines informations d'environnement.
H9	Le système doit vérifier que l'espace navigable est mis à jour à une fréquence suffisante.	Le Safety-Bag peut vérifier que l'espace navigable est mis à jour.	Le Safety-Bag doit vérifier la vivacité (la mise à jour) de l'espace navigable régulièrement. Dans le cas contraire, une alarme est déclenchée et le Safety-Bag inhibe l'accélération, freine et maintient la position d'angle volant après un certain temps.	L'application de contrôle-commande doit transmettre au Safety-Bag la date de la mise à jour de l'espace navigable.
H10	Le système doit vérifier la vivacité de l'espace navigable.	idem 9		
H11	Le système doit vérifier que la trajectoire cinématique est mise à jour à chaque rafraichissement de l'espace navigable.	Le Safety-Bag peut vérifier que la trajectoire cinématique est mise à jour après chaque évolution de l'espace navigable.	Le Safety-Bag doit vérifier que l'horodatage de la trajectoire cinématique est plus récent que l'horodatage de l'espace navigable. Dans le cas contraire, le safety-Bag déclenche les alarmes, inhibe l'accélération et force des actions de sécurité telles que le freinage et le maintien d'angle volant.	_
H12	Le système doit anticiper les mouvements des véhicules et piétons et garder une distance de sécurité entre eux.	La redondance des capteurs de fusion de données dans l'application peut permettre au Safety-Bag de vérifier la distance par rapport aux véhicules suivis.	_	Cette exigence de sécurité nécessité des informations que le Safety-Bag ne peut pas obtenir de manière suffisamment fiable.
H13	Le système doit vérifier que l'espace navigable est correct.	Un composant logiciel diversifié de l'application de contrôle-commande pourrait effectuer certaines vérifications trop complexes à mettre en œuvre par le Safety-Bag.	_	_
H14	Le système doit disposer de suffisamment de capteurs redondants pour tolérer leurs défaillances matérielles. Une étude basée sur leurs taux de défaillance peut être nécessaire pour identifier les redondances nécessaires.	Redondance des capteurs (caméra, télémètre, radar, etc.)	_	_

Tableau 4.6 – Exigences et nécessités de sécurité issues de l'analyse HazOp (page3)

Réf	**Exigences de sécurité**	**Moyens d'implémentation d'exigence de sécurité**	**Nécessité de sécurité en langage naturel**	**Commentaires**
H15	Le système doit vérifier que l'emprise connue du véhicule reste dans la Zone Navigable en Sécurité (ZNS).	idem H13	–	–
H16	Il faut prendre des marges suffisantes pour la zone navigable en sécurité (ZNS).	idem H13	–	Le Safety-Bag n'est pas capable de réaliser cette exigence de sécurité.
H17	Le système doit vérifier ou garantir que l'on ne s'éloigne pas trop de la trajectoire cinématique demandée.	• Redondance logicielle : pour garantir un contrôle qui fonctionne bien, • Redondance matérielle : actionneurs, moteur, freinage, direction, etc. • Redondance des capteurs d'environnement	–	–
H18	Un composant critique auto-testable doit détecter toutes ses erreurs. De plus, tous les composants critiques doivent être auto-testables ou validés formellement de façon à n'avoir aucune faute interne.	Cette exigence de sécurité est implémentable par un composant critique auto-testable.	–	On doit avoir une information sur l'état des composants du véhicule.
H19	Il faut tolérer les défaillances des composants ou garantir que les composants ne défaillent pas.	• Redondance des capteurs, • Le Safety-Bag doit intégrer une tolérance à ses propres fautes : ◇ Redondance de ses calculateurs, ◇ Redondance du mécanisme de journalisation (log), ◇ Au moins un moyen ultime de mise en sécurité	–	–
H20	Le système doit vérifier régulièrement lors du suivi de trajectoire la conformité de l'évolution dans le temps de la position du véhicule avec le suivi de trajectoire prévue. Si une incohérence apparaît, une nouvelle trajectoire doit être générée.	idem H13	–	–

4.2.3 Comparaison AMDEC/ HazOp-UML vis-à-vis des nécessités de sécurité

La figure 4.1 représente les résultats quantitatifs obtenus en effectuant les deux méthodes d'analyse de risques AMDEC et HazOp-UML. Cependant, nous ne sommes concentrés sur le cas d'utilisation qui génèrerait le pus de nécessités de sécurité pour le Safety-Bag. En effet, un cas d'utilisation tel que « *UC3 : générer un itinéraire* » n'aboutirait probablement qu'à peu de nécessités de sécurité comme il n'y a peu de commandes ou d'informations sortant de l'application de contrôle-commande. Nous avons obtenu 11 nécessités de sécurité à partir de 11 exigences de sécurité et 14 nécessités de sécurité à partir de 20 exigences de sécurité en appliquant respectivement les deux méthodes AMDEC et HazOp. Ainsi que détaillé dans les sections précédentes, il est possible d'avoir moins de nécessités de sécurité que d'exigences de sécurité, car certaines nécessités ne peuvent pas être actuellement implémentées sur le Safety-Bag. En effet, des nécessités de sécurité portant sur des questions complexes et de haut niveau d'abstraction (l'espace navigable est-il dégagé par exemple) ne peuvent pas être vérifiées sans données intègres (ce que ne fournissent pas actuellement les capteurs pour ces fonctions) et nécessitent même dans certains cas des mécanismes décisionnels complexes (que le Safety-Bag ne peut pas contenir pour garantir sa sûreté de fonctionnement). Notons également que certaines exigences de sécurité peuvent aboutir à plusieurs nécessités de sécurité.

	AMDEC	**HazOp** *(1 cas d'utilisation)*
Nombre de défaillances/déviations	*20*	*46*
Exigences de sécurité	*11*	*20*
Nécessités de sécurité au total	*11*	*14*
Nécessités de sécurité similaires à l'autre méthode	*8*	*7*
Exigences de sécurité non implémentées	*3*	*12*

Figure 4.1 – Résultats quantitatifs des deux méthodes d'analyse de risque AMDEC et HazOp

Dans notre étude, nous avons remarqué qu'il existe des éléments qui apparaissent dans l'une des deux méthodes et pas dans l'autre. En effet, les 3 nécessités de sécurité suivantes correspondent à des nécessités de sécurité issues de l'analyse AMDEC et ne sont pas mentionnées dans l'HazOp :

1. A1 : Le Safety-Bag doit vérifier la circulation des trames du bus CAN. En cas d'absence de communication pendant une durée trop importante, le Safety-Bag doit alerter le conducteur et repasser en mode manuel.

2. A2 : Le Safety-Bag doit vérifier que les capteurs proprioceptifs latéraux

(accélération latérale et gyromètre) fonctionnent correctement. Dans le cas contraire, le Safety-Bag déclenche les alarmes, inhibe l'accélération, freine modérément le véhicule et applique la valeur d'angle volant venant de l'application de contrôle-commande.

3. A3 : Le Safety-Bag doit vérifier que les capteurs proprioceptifs longitudinaux (capteurs de vitesse et d'accélération longitudinales) fonctionnent correctement. Dans le cas contraire, le Safety-Bag déclenche les alarmes, inhibe l'accélération, et freine modérément le véhicule.

Inversement, certaines nécessités qui sont identifiées dans l'HazOp ne le sont pas dans l'AMDEC. Il s'agit des nécessités de sécurité suivantes :

1. H1 : Le Safety-Bag doit vérifier la vivacité de l'état cinématique. Si l'état cinématique n'est pas mis à jour, le Safety- Bag intervient en déclenchant les alarmes, en inhibant des commandes de l'accélération et/ou forçant une action de sécurité telle que le freinage du véhicule et le maintien de l'angle volant après un certain délai.

2. H2 : Le Safety-Bag doit vérifier la cohérence de l'évolution de l'état cinématique. En cas d'incohérence de l'état cinématique, le Safety-Bag intervient en levant les alarmes, en inhibant l'accélération et en freinant modérément le véhicule. Il doit également maintenir l'angle volant.

3. H3 : Le Safety-Bag doit vérifier la cohérence temporelle de l'état cinématique. Si l'état cinématique n'est pas temporellement cohérent, le Safety-Bag doit lever les alarmes, inhiber l'accélération, freiner modérément le véhicule et maintenir l'angle volant.

4. H4 : Le Safety-Bag doit vérifier la vivacité de la trajectoire cinématique. Dans le cas contraire, le Safety-Bag doit déclencher les alarmes, inhiber l'accélération, freiner et maintenir la position d'angle volant après un certain temps.

5. H5 : Le Safety-Bag doit vérifier la vivacité de l'espace navigable. Dans le cas contraire, le Safety-Bag déclenche une alarme, inhibe l'accélération, freine et maintient l'angle volant après un certain temps.

6. H6 : Le Safety-Bag doit vérifier que l'horodatage de la trajectoire cinématique est plus récent que l'horodatage de l'espace navigable. Dans le cas contraire, le Safety-Bag déclenche les alarmes, inhibe l'accélération et force des actions de sécurité telles que le freinage et le maintien d'angle volant.

Il existe des éléments communs entre les deux techniques, qui aboutissent à des vérifications similaires ou équivalentes. Parmi ces éléments en commun, nous avons identifié les 8 nécessités de sécurité similaires suivantes :

1. U1 : Le Safety-Bag doit vérifier la vivacité de l'application de contrôle-commande. Si celle-ci n'envoie plus de commande, le Safety-Bag doit prévenir le conducteur et repasser en mode manuel et arrête le véhicule en allumant les feux de détresse.

2. U2 : Le Safety-Bag doit vérifier la cohérence temporelle de l'application de contrôle-commande. En cas d'incohérence temporelle, le Safety-Bag doit assurer la mise en sécurité du système en levant les alarmes et en repassant en mode manuel.

3. U3 : Le Safety-Bag doit vérifier les bornes de vitesse maximale du véhicule. Dans le cas contraire, le Safety-Bag déclenche les alarmes, bloque les commandes d'accélération de l'application de contrôle-commande et assure la reprise en mode manuel.

4. U4 : Le Safey-Bag doit vérifier le contrôle dynamique du véhicule : il doit vérifier que l'accélération latérale reste inférieure à une valeur seuil. Si ces bornes ne sont pas respectées, le Safety-Bag doit déclencher les alarmes et assurer la reprise en mode manuel.

5. U5 : Le Safety-Bag doit vérifier que l'accélérateur fonctionne correctement. Il doit vérifier que l'intensité dans le moteur électrique correspond à la valeur de la commande d'accélération fournie par l'application de contrôle-commande. Dans le cas contraire, le Safety-Bag lève une alarme, inhibe l'accélération et freine modérément.

6. U6 : Le Safety-Bag doit vérifier que le frein fonctionne correctement. Il doit vérifier que la pression ne s'écarte pas de la valeur d'une fonction donnée de la commande du frein. Dans le cas contraire, le Safety-Bag déclenche les alarmes, empêche les commandes d'accélération et freine modérément.

7. U7 : Le Safety-Bag doit vérifier que la direction fonctionne correctement : Il doit vérifier que les angles volant évoluent dans le sens prévu selon le couple volant appliqué. Dans le cas contraire, une alerte doit être déclenchée vers le conducteur et le Safety-Bag doit mettre le véhicule en état sûr en passant en mode de conduite manuelle.

8. U8 : Le Safety-Bag doit vérifier que la direction fonctionne correctement : il doit vérifier que l'intensité sur le moteur de direction correspond aux consignes envoyées par l'application de contrôle-commande. En cas de non correspondance, le Safety-Bag doit alerter le conducteur et doit mettre le système en état sûr en redonnant la main au conducteur.

Que nous ayons trouvé 7 nécessités HazOp qui soient similaires à celles de l'AMDEC et 8 nécessités AMDEC qui soient similaires à celles de l'HazOp peut être justifié par le fait que la nécessité de sécurité identifiée par la méthode AMDEC « *A11 : Le Safety-Bag doit vérifier que la direction fonctionne correctement. Dans le cas contraire, le Safety-Bag déclenche les alarmes, empêche les commandes d'accélération et freine modérément.* » a été raffinée en deux nécessités de sécurité :

- Le Safety-Bag doit vérifier que la direction fonctionne correctement : Il doit vérifier que les angles volant évoluent dans le sens prévu selon le couple volant appliqué. Dans le cas contraire, une alerte doit être déclenchée vers le conducteur et le Safety-Bag doit mettre le véhicule en état sûr en passant en mode de conduite manuelle.

- Le Safety-Bag doit vérifier que la direction fonctionne correctement : Il doit vérifier que l'intensité sur le moteur de direction correspond aux consignes envoyées par l'application de contrôle-commande. En cas de non correspondance, le Safety-Bag doit alerter le conducteur et doit mettre le système en état sûr en redonnant la main au conducteur.

Dans l'HazOp, la nécessité de sécurité A11 est restée sous la forme générale suivante : « *H5 : Le Safety-Bag doit vérifier que la direction fonctionne correctement. Dans le cas contraire, le Safety-Bag déclenche les alarmes, empêche les commandes d'accélération et freine modérément.* »

La cause principale des différences observées entre les nécessités de sécurité des deux analyses est que l'HazOp-UML étudie explicitement les processus du système et leur évolution dans le temps (en se basant sur les cas d'utilisation et l'évolution de leurs invariants), tandis que l'AMDEC se concentre sur les défaillances des composants du système à un instant donné.

Par conséquent, ces deux techniques nous apparaissent complémentaires.

Les principaux verrous technologiques identifiés dans ces analyses concernent la garantie de la conformité de ce que perçoit le véhicule par rapport aux cartes routières embarquées, la détection des trajectoires dangereuses, des zones navigables en sécurité, et des distances de sécurité. En effet, Le Safety-Bag n'a pas les moyens pour traiter de façon fiable la détection d'obstacles pour un véhicule autonome et

ne peut pas identifier les conditions environnementales (verglas, pluie, etc.). Vérifier précisément la localisation effectuée par le système de conduite autonome du véhicule est encore trop difficile pour le Safety-Bag. Des écarts entre les positions prévisibles et les positions calculées par le système de localisation peuvent être importants sans qu'il y ait de dysfonctionnement.

Pour certaines exigences de sécurité, ces verrous technologiques peuvent être levés en introduisant par exemple un composant logiciel diversifié de l'application de contrôle-commande, qui pourrait effectuer certaines vérifications trop complexes à mettre en œuvre par le Safety-Bag. Un tel module logiciel peut détecter et vérifier les exigences de sécurité suivantes :

- La trajectoire cinématique reste dans une partie sûre de l'espace navigable (H7).
- L'espace navigable est correct (H13).
- L'emprise connue du véhicule reste dans la zone navigable en sécurité (ZNS) (H15).
- Il faut prendre des marges suffisantes pour la zone navigable en sécurité (H16).
- Il faut vérifier la conformité de l'évolution dans le temps de la position du véhicule avec le suivi de trajectoire prévue (H20).
- Il faut vérifier que les capteurs extéroceptifs liés à l'environnement fonctionnent correctement (A7).

Ce module logiciel serait donc un composant de sécurité effectuant des opération de fusion de données et de calcul sur les trajectoires. Bien que plus simples à valider que des mécanismes de reconnaissance de situation, ces composants logiciels restent actuellement trop complexes pour que les moyens d'élimination des fautes soient suffisants pour garantir leur bon fonctionnement, contrairement à la partie logicielle du Safety-Bag.

L'exigence de sécurité H18 « *Un composant critique auto-testable doit détecter toutes ses erreurs. De plus, tous les composants critiques doivent être auto-testables ou validés formellement de façon à n'avoir aucune faute interne.* » impose notamment la redondance des composants critiques du Safety-Bag. Le dispositif que nous avons proposé n'est pas totalement suffisant, notamment parce que le Safety-Bag Supervisor, redondance du Safety-Bag Rules Checker, ne peut pas inhiber ou générer de commandes sur les actionneurs, au contraire du second (voir chapitre 5, section 5.1.2.1).

Du fait de ces limitations, le Safety-Bag reste une solution prometteuse mais partielle pour des véhicules autonomes industriels/commerciaux. La vérification permanente lors de la conduite autonome de ces exigences de sécurité constituent ainsi un verrou technologique pour le développement des véhicules autonomes.

4.3 Identification des conditions de déclenchement de sécurité et détermination des marges de sécurité

La condition de déclenchement de sécurité (CdS) est une condition évaluable par le Safety-Bag lui permettant de détecter une erreur et de déclencher l'intervention de sécurité associée.

La condition de déclenchement de sécurité est généralement une variable à surveiller par le Safety-Bag comparée à une valeur seuil. La valeur seuil peut être définie empiriquement par des expériences ou plus formellement par la définition d'une valeur de danger tirée d'une analyse de sécurité dont est soustraite une marge de sécurité.

La valeur de danger représente la frontière avec les états catastrophiques dans lesquels le système est susceptible d'arriver suite à une défaillance. La marge de sécurité fixe la condition de déclenchement de sécurité pour que l'intervention de sécurité ramène le système dans un état sûr avant que le système n'arrive dans un état catastrophique.

Une condition de déclenchement de sécurité est ainsi généralement assimilable à la forme :

`variable à observer` $\geq$ `valeur seuil (empirique ou formelle)`

ou encore

`variable à observer` $\geq$ `valeur de danger - marge de sécurité`

La valeur seuil, la valeur de danger, et les marges de sécurité peuvent être déterminées de trois manières différentes :

- Formellement : cette approche nécessite de pouvoir calculer la valeur de danger ou la marge de sécurité à partir de modèles physiques ou mathématiques.
- Empiriquement : Le choix de la valeur seuil est effectué sur la base de résultats expérimentaux.
- A travers un jugement d'expert : C'est une approche qui n'est justifiée ni

par des descriptions formelles ni par des résultats expérimentaux, mais par des arguments basés sur la connaissance du système et l'expérience. C'est la méthode donnant le moins de confiance. Elle est souvent utilisée pour des systèmes expérimentaux sans processus industrialisé qui ne dispose pas de données d'étude suffisantes pour en tirer des valeurs empiriques ou des modèles plus formels, et qui sont souvent destinés à subir de nombreuses modifications en opération (nouveaux algorithmes, nouveaux composants matériels, etc.).

Dans la suite de cette section, nous présentons formellement et informellement les conditions de déclenchement des nécessités de sécurité présentées dans la section précédente. Ces résultats sont également présentés dans les tableaux (4.7, 4.8, 4.9,4.10 et 4.11).

4.3.1 Détermination des conditions de déclenchement de sécurité et leurs marges de sécurité

Dans cette section, nous ajoutons une expression formelle pour chaque condition de déclenchement de sécurité définie informellement. Nous justifions également les valeurs seuil, les valeurs de danger et les marges de sécurité que nous avons attribuées pour chaque condition de déclenchement de sécurité. La justification des interventions de sécurité sera étudiée dans la section 4.4.

1. Pour la nécessité de sécurité « *Le Safety-Bag doit vérifier la vivacité de l'application de contrôle-commande. Si celle-ci n'envoie plus de commande, le Safety-Bag doit prévenir le conducteur pour qu'il repasse en mode manuel et arrête le véhicule en allumant les feux de détresse.* », nous avons exprimé la condition de déclenchement de sécurité suivante :

   - Informellement : Le Safety-Bag doit vérifier que la période de la mise à jour des commandes par l'application de contrôle-commande doit être toujours inférieure à une valeur seuil.
   - Formellement : Le Safety-Bag implémente l'algorithme suivant :
     `Loop`
     `Commande reçue`
     $t_{i-1} = t_i; t_i = Now()$; `%Now() : horloge du Safety-Bag`
     $Check : t_i - t_{i-1} < valeur\_seuil$; `% L'assertion check correspond à la condition vérifiée par le Safety-Bag si elle est enfreinte, une intervention de sécurité est déclenchée; %` t_{i-1} `: horloge du Safey-Bag lors de la réception de la précédente commande.`

`End Loop`

Nous obtenons dans ce cas une valeur seuil de 20ms, une valeur de danger de 50ms et une marge de sécurité de `30 ms` en considérant que :

- ◇ La commande du véhicule doit être mise à jour au moins toutes les 50 ms (20 hz), pour que la conduite soit correcte (c'est le minimum reconnu par la communauté automobile).
- ◇ L'envoi des informations de contrôle par l'application de contrôle-commande peut être effectué toutes les 10 ms (100 hz). Ceci est un compromis entre la charge informatique (la quantité de calcul fait par les calculateurs de l'application de contrôle-commande) et réseau d'une part et la précision du contrôle de la voiture (la dynamique de la voiture) d'autre part.
- ◇ Donc, nous pouvons tolérer la perte de quatre commandes du véhicule avant d'atteindre un état dangereux. Le choix de la marge de sécurité dépend du compromis entre sécurité-innocuité et disponibilité que l'on veut atteindre. Considérant comme critique la non-réponse de l'application de contrôle commande, et peu probable la perte de deux commandes successives entre cette application et le Safety-Bag, nous choisissons de déclencher la détection d'erreur suite à l'absence de commande pendant 20 ms (deux commandes successives perdues). Nous avons ainsi une marge de 30 ms par rapport au passage dans l'état dangereux.

2. Pour la nécessité de sécurité : « *Le Safety-bag doit vérifier la cohérence temporelle des commandes à appliquer et certaines informations de l'application de contrôle-commande. En cas d'incohérence temporelle, le Safety-Bag doit assurer la mise en sécurité du système en levant les alarmes et en repassant en mode manuel.* », nous avons exprimé la condition de déclenchement de sécurité :

   - Informellement : A chaque réception de données fournies par l'application de contrôle-commande, le Safety-Bag vérifie qu'il n'y a pas d'incohérence temporelle, c'est-à-dire, que :
     - ◇ le temps auquel la commande a été générée est postérieur à celui de la commande précédente,
     - ◇ l'horodatage des données de la commande reçue n'est pas trop ancien,
     - ◇ l'horodatage des données de la commande reçue n'est pas dans le futur.

- Formellement :
  `Check :` $h_{ci} \geq h_{ci-1}$ `et` $NOW - \varepsilon < h_{ci} \leq NOW$ `; avec` h_{ci} `l'horodatage de la commande numéro i.`
  Cette condition d'ordre est une condition logique pour laquelle il n'y a pas une marge de sécurité.

3. Pour la nécessité de sécurité : « *Le Safety-Bag doit vérifier les bornes de vitesse : il doit vérifier si la vitesse du véhicule ne dépasse pas une vitesse de sécurité de 50 km/h. Dans le cas contraire, le Safety-Bag déclenche les alarmes, bloque les commandes d'accélération de l'application de contrôle-commande et assurer la reprise en mode de conduite manuelle.* », nous avons exprimé la condition de déclenchement de sécurité comme suit :

   - Informellement : Le Safety-Bag doit vérifier que le véhicule ne dépasse pas une vitesse limite de sécurité. La valeur de la vitesse du véhicule provient du réseau CAN.
   - Formellement : $V \leq 50km/h -$ `marge de sécurité ; la marge de sécurité étant prise à` $3km/h$.

   La valeur de danger dans ce cas vaut `50 km/h` et la marge de sécurité est égale à `3 km/h`. La première valeur est une valeur réglementée dans le code de la route, au-delà de laquelle les distances d'arrêt et vitesses de collision deviennent inacceptables en ville. La marge de sécurité est déterminée par jugement d'expert, considérant que l'action associée du Safety-Bag, l'inhibition de l'accélération, ne peut pas physiquement amener à une augmentation de la vitesse du véhicule à moins que celui-ci soit en pente.

   Nous préférons garder une marge de sécurité de 3 km/h pour être prudent et pour prendre en compte la dynamique du véhicule entre le temps de la détection d'une vitesse incorrecte et le temps où le Safety-Bag réagit en inhibant l'accélération (donc en mettant l'accélération à zéro). Considérant une pente faible, la marge choisie borne la somme des éléments suivants :

   - l'incertitude sur la mesure de la voiture (1 km/h),
   - le délai entre le dépassement effectif de la vitesse et sa détection par le Safety-Bag (inférieur à 2 cycles du Safety-Bag, soit 0.02 s, donc une augmentation de la vitesse de moins de 0.3 km/h, considérant que le véhicule considéré peut accélérer de 0 à 50 km/h en 4s),
   - et surtout le délai que nous ne connaissons pas précisément entre le changement de la consigne « *pédale d'accélération* » et sa prise en compte

par le calculateur de puissance du moteur électrique (mais que nous estimons à moins de 0.1 s, ce qui peut provoquer une augmentation de la vitesse jusqu'à 1.5 km/h, ce qui est confirmé par les observations que nous avons pu faire en coupant soudainement l'accélération du véhicule).

4. Pour la nécessité de sécurité : « *Le Safety-Bag doit vérifier les limites dynamiques : l'accélération latérale doit rester inférieure à une valeur seuil. Dans le cas contraire, le Safety-Bag doit déclencher les alarmes et assurer la reprise en mode manuel.* », nous avons introduit la condition de déclenchement de sécurité suivante :

   - Informellement : La condition exigeant que le Safety-Bag doit vérifier que le véhicule ne dépasse pas une accélération latérale limite est traduite pour nos expérimentations sur le banc VILAD par une limite sur le produit de la vitesse du véhicule par l'angle volant.
   - Formellement : $(vitesse_{vehicule} - 15) * |angle_{volant}| \leq 2000$
     Dans des conditions idéales, l'accélération latérale maximale de notre véhicule expérimental est environ $10m.s^{-2}$. Empiriquement sur le banc VILAD et avec le contrôle latéral actif, il est possible de contrôler le véhicule pour une accélération latérale maximale de $5m.s^{-2}$. La formule que nous avons définie empiriquement provoque l'intervention du Safety-Bag vers $4m.s^{-2}$. Il y a ainsi une marge de sécurité de $1m.s^{-2}$. Le Safety-Bag n'anticipant pas la trajectoire à venir, l'angle volant peut augmenter plus rapidement que la vitesse ne diminue. Cependant dans nos expérimentations, on approche mais ne dépasse pas les $5m.s^{-2}$. L'intervention du Safety-Bag n'empêche cependant pas toujours la sortie de la route.
     Les causes de ces sorties de route seront discutées dans la section 5.4.2.

5. Pour la nécessité de sécurité : « *Le Safety-Bag doit vérifier la circulation des trames du bus CAN. En cas d'absence de communication pendant une durée trop importante, le Safety-Bag doit alerter le pilote et repasser en mode manuel.* », nous avons exprimé la condition de déclenchement de sécurité suivante :

   - Informellement : Les deux trames (`Steering_wheel` et `Front_wheel_-speed`) sont émises au moins toutes les 20ms. La trame « *Steering-wheel* » contient l'information de la position et de la vitesse du volant. Et la trame « *Front-wheel-speed* » contient l'information de vitesse de chaque roue avant et leur moyenne.

- Formellement : Le Safety-Bag implémente le même algorithme que pour le cas de la nécessité de sécurité 1 « *Le Safety-Bag doit vérifier la vivacité de l'application de contrôle-commande. Si celle-ci n'envoie plus de commande, le Safety-Bag doit prévenir le conducteur et repasser en mode manuel ou arrêter le véhicule lentement sur le côté de la route en allumant les feux de détresse.*

  Toutes les informations contenues dans ces trames sont indispensables à l'application de contrôle-commande pour qu'elle conduise correctement le véhicule. Leur non-observation sur le réseau véhicule par le Safety-Bag indique probablement qu'elles ne sont pas accessibles pour l'application. La valeur de danger est la même que pour la nécessité de sécurité N° 1. Nous avons choisi le même seuil de 20ms et donc la marge de sécurité est également la même, soit 30ms. Cependant, les trames étant émises toutes les 20ms avec cette valeur seuil, le Safety-Bag déclenchera une détection d'erreur dés qu'une valeur est absente, sans accepter la moindre perte de données. La fiabilité du système risque donc d'en être affaiblie.

6. Pour la nécessité de sécurité : « *Le Safety-Bag doit vérifier que la vitesse de rotation et l'accélération latérale sont compatibles avec les informations redondantes et indépendantes par les capteurs du Safety-Bag. En cas d'incohérence, le Safety-Bag doit freiner et alerter le conducteur.* », nous avons introduit la condition de déclenchement de sécurité suivante :

   - Informellement : Le Safety-Bag vérifie que la vitesse de rotation connue par l'application de contrôle-commande est incluse dans un intervalle de sécurité comprenant la vitesse connue par le Safety-Bag.
   - Formellement :
     $\omega_{AppCC} \notin [\omega_{SB} - \varepsilon; \omega_{SB} + \varepsilon]$

   Pour déterminer la marge de sécurité (ε), il serait nécessaire de faire des expérimentations et des mesures sur une piste réelle.

   Nous n'avons pas implémenté cette nécessité de sécurité car l'environnement du banc VILAD est statique (la voiture reste fixée au banc, sans mouvements latéraux ou longitudinaux) et ne permet pas ces mesures même avec la présence des capteurs. De plus, comme dit précédemment, nous considérons que les données calculées par le simulateur ne sont pas générées à une fréquence suffisante pour être utilisable par le Safety-Bag.

7. Pour la nécessité de sécurité : « *Le Safety-Bag doit vérifier que la vitesse*

longitudinale et l'accélération longitudinale sont compatibles avec les informations redondantes et indépendantes par les capteurs du Safety-Bag. En cas d'incohérence, le Safety-Bag doit freiner et alerter le conducteur. », nous avons introduit la condition de déclenchement de sécurité suivante :

- Informellement : Le Safety-Bag vérifie que la vitesse longitudinale connue par l'application de contrôle-commande est incluse dans un intervalle de sécurité comprenant la vitesse longitudinale connue par le Safety-Bag.
- Formellement :
  $V_{AppCC} \notin [V_{SB} - \varepsilon; V_{SB} + \varepsilon]$

De même que précédemment, pour déterminer la marge de sécurité (ε), il serait nécessaire de faire des expérimentations et des mesures sur une piste réelle et sur la piste virtuelle simulée par le banc VILAD. En ce qui concerne la redondance des capteurs, nous disposons déjà des capteurs proprioceptifs du véhicule (vitesse de rotation de roues). Il faudrait rajouter un capteur redondant permettant de déterminer la vitesse du véhicule pour pouvoir les comparer.

8. Pour la nécessité de sécurité : « *Le Safety-Bag doit vérifier que l'intensité dans le moteur électrique correspond à la valeur de la commande d'accélération fournie par l'application de contrôle commande. Dans le cas contraire, une alarme est levée et le Safety-Bag inhibe l'accélération et freine modérément.* », nous avons introduit la condition de déclenchement de sécurité suivante :

   - Informellement : Le Safety-Bag doit vérifier que l'intensité dans le moteur électrique est proche de l'intensité nominalement associée à la valeur de la commande.
   - Formellement :
     `Loop`
     `%` $i_{vehicule}$ `l'intensité observée par un capteur dans le moteur électrique du véhicule`
     `Test`
     $check : I(acc) - \varepsilon < i_{vehicule} < I(acc) + \varepsilon$; `% I(acc): fonction calculant l'intensité nominale dans le moteur électrique pour une consigne d'accélération.`
     `End Loop`

     Le calcul de l'intensité nominale dans le moteur électrique dépend des spécifications de la machine électrique et de la loi de commande de

la machine électrique choisie par le constructeur. Ces informations ne sont pas facilement accessibles. Pour cela, nous ne sommes pas capables de déterminer ni le seuil ni la marge de sécurité et nous n'avons pas expérimenté cette nécessité de sécurité.

Notons que si le véhicule est équipé d'un système antipatinage et que celui-ci intervient, il peut perturber l'évolution de cette condition.

9. Pour la nécessité de sécurité : « *Le Safety-Bag doit vérifier que la pression ne s'écarte pas de la valeur d'une fonction donnée de la commande du frein. Dans le cas contraire, le Safety-Bag déclenche les alarmes, empêche les commandes d'accélération et freine modérément.* », nous avons introduit la condition de déclenchement de sécurité suivante :

   - Informellement : Le Safety-Bag doit vérifier que la pression du frein est proche de la pression nominalement associée à la commande.
   - Formellement :
     `Loop`
     p_{frein}
     `Test`
     check : $P(f)-\varepsilon < p_{frein} < P(f)+\varepsilon$`; % P(f): fonction calculant la pression nominale du frein en fonction de la commande du frein`
     `End Loop`

     Cette fonction dépend des spécifications du véhicule auxquelles nous n'avons pas accès. Cette fonction doit intégrer également les interventions des systèmes telles que l'ABS, l'ESP et l'amplification de freinage qui modifient la consigne de base.

10. Pour la nécessité de sécurité : « *Le Safety-Bag doit vérifier que l'intensité sur le moteur de direction correspond aux consignes envoyées par l'application de contrôle-commande. En cas de non-correspondance, le Safety-Bag doit alerter le conducteur et doit mettre le système en état sûr en redonnant la main au conducteur.* », nous avons introduit la condition de déclenchement de sécurité suivante :

    - Informellement : Le Safety-Bag vérifie que l'intensité sur le moteur de direction est proche de l'intensité nominalement associée à la commande.
    - Formellement :
      `Loop`
      `%` $i_{direction}$ `sur le moteur de direction`

```
Test
```
$check : I(direction) - \varepsilon < i_{direction} < I(direction) + \varepsilon$; `% I(direction) : fonction calculant l'intensité nominale dans la direction en fonction de la commande appliquée à la direction`
```
End Loop
```

La fonction I(direction) dépend comme c'est le cas pour la fonction donnant la pression du frein et la fonction donnant l'intensité du moteur électrique des spécifications du véhicule qui ne sont pas facilement accessibles. De ce fait, nous n'avons pas pu précisé la valeur seuil et la marge de sécurité.

11. Pour la nécessité de sécurité : « *Le Safety-Bag doit vérifier que les angles volant évoluent dans le sens prévu selon le couple volant appliqué. Dans le cas contraire, une alerte doit déclenchée vers le conducteur et le Safety-Bag doit mettre le véhicule en état sûr en passant en mode de conduite manuelle.* », la condition de déclenchement de sécurité est exprimée informellement puis formellement comme suit :

    - Informellement : Les angles volant doivent évoluer dans le sens prévu selon le couple volant appliqué.
    - Formellement :
      On détermine empiriquement une valeur seuil en dessous de laquelle la commande de la direction n'est pas efficace ($couple\_min$) :
      $|couple\_volant| < couple\_min$
      Le sens correct de l'évolution du volant est vérifiée par :
      $couple\_volant * vitesse\_volant > 0$

    Déterminer la valeur couple_min demanderait des expérimentations dans des conditions très variées (chaussées irrégulières, flaques d'eau, etc.) que nous ne pouvons pas nous permettre dans ces travaux. Cette nécessité de sécurité ne sera donc pas implémentée sur le véhicule.

12. Pour la nécessité de sécurité : « *Le Safety-Bag doit vérifier la vivacité de l'état cinématique. Si l'état cinématique n'est pas mis à jour, le Safety-Bag intervient en déclenchant les alarmes, en inhibant des commandes de l'accélération et/ou en forçant une action de sécurité telle que le freinage du véhicule et le maintien de l'angle volant après un certain délai.* », nous avons exprimé la condition de déclenchement de sécurité suivante :

- Informellement : la période de la mise à jour de l'état cinématique doit être toujours inférieure à une valeur seuil.
- Formellement :
  `Loop`
  `Pour l'état cinématique et pour sa` ***période*** `de mise à jour faire`
  $t_{i-1} = t_i$; $t_i = Now()$; `%Now() : horloge du Safety-Bag`
  $check : t_i - t_{i-1} < valeur\_seuil$
  `End Loop`

Nous obtenons par jugement d'expert une valeur seuil de 0.2s, une valeur de danger de 0.5s et une marge de sécurité de 0.3s en considérant que :

- L'état cinématique doit être mis à jour au moins toutes les 0.5s (2 hz), ce qui correspond pour un conducteur humain à fermer les yeux pendant une demi seconde. Cela représente déjà presque 7m à la vitesse de 50km/h. Cette valeur est aussi de l'ordre de grandeur proposé par [Houenou et al., 2013] pour la durée de validité d'une estimation de trajectoire basée sur un état. Ce papier a pour objectif de prédire les trajectoires des autres usagers de la circulation pour un véhicule autonome, afin d'éviter les collisions sur sa trajectoire prévue. Notons que cela ne peut pas être fait par notre Safety-Bag puisqu'il n'est pas capable de connaitre les décisions prises par l'application de contrôle-commande. La prédiction de la trajectoire basée sur l'état cinématique (x,y,v,cap + l'accélération et la vitesse de rotation) permet une estimation correcte pendant un « *short time* » tandis que la prédiction basée sur une estimation de la trajectoire est valable sur un « *long time* ». Le *long time* est de « plusieurs secondes » typiquement de 2 à 6 secondes. De là on peut en conclure que le *short time* est inférieur à une seconde. Pour nous, 0.5s nous parait raisonnable.
- La mise à jour de l'état cinématique par l'application de contrôle-commande est effectuée toutes les 50ms (20 hz). Ceci correspond également à la fréquence de plusieurs capteurs utiles pour déterminer l'état cinématique (le GPS et les caméras par exemple).
- De façon analogue à la nécessité de sécurité « *le Safety-Bag doit vérifier la vivacité de l'application de contrôle-commande* », on tolère que l'état cinématique ne soit pas mis à jour 3 fois avant de déclencher l'intervention de sécurité, donc après 0.2 secondes. Cette intervention de sécurité est déclenchée 0.3 s (la marge de sécurité) avant que l'état cinématique estimé

par le véhicule ne soit plus utilisable.

13. Pour les deux nécessités de sécurité : « *Le Safety-Bag doit vérifier la vivacité de la trajectoire cinématique. Dans le cas contraire, une alarme sera déclenchée et le Safety-Bag inhibe l'accélération, freine et maintient la position d'angle volant après un certain temps.* » et « *Le Safety-Bag doit vérifier la vivacité de l'espace navigable. Dans le cas contraire, une alarme est déclenchée et le Safety-Bag inhibe l'accélération, freine et maintient la position d'angle volant après un certain temps.* », nous avons exprimé la condition de déclenchement de sécurité suivante :

    - Informellement : la période de la mise à jour de la trajectoire cinématique ou de l'espace navigable doit être toujours inférieure à une valeur seuil.
    - Formellement :
      `Loop`
      `Pour la trajectoire cinématique ou l'espace navigable et pour leur` ***`période`*** `de mise à jour respective faire`
      $t_{i-1} = t_i$; $t_i = Now()$; `%Now() : horloge du Safety-Bag`
      $check : t_i - t_{i-1} < valeur\_seuil$
      `End Loop`

    Pour ces deux nécessités, nous choisissons une valeur seuil de 0.4 s, une marge de 1.6 s et une valeur de danger de 2s en considérant que :

    - La trajectoire cinématique et l'espace navigable doivent être mis à jour au moins toutes les 2s (0.5 hz), ce qui correspond au délai avant que la situation de conduite n'évolue de façon sensible. Cette fréquence est de l'ordre de grandeur donné dans les travaux sur la génération de trajectoire cinématique publiés dans [Chebly et al., 2017] qui décrivent la planification des manœuvres avec des tentacules clothoïdes pour la planification locale des trajectoires cinématiques.
    - La mise à jour de la trajectoire cinématique ou de l'espace navigable par l'application de contrôle-commande est effectuée toutes les 0.1 s (10 hz) dans le même papier [Chebly et al., 2017]. Les auteurs ont considéré que l'algorithme de planification locale est exécuté toutes les 100ms et qu'une tentacule est non navigable si un obstacle est détecté à une distance de sécurité correspondant à 2 secondes de réaction par rapport à la vitesse du véhicule. Si l'obstacle est au-delà de cette distance de sécurité, le tentacule est classé comme navigable.

- Comme dans le cas de l'état cinématique, on tolère que la trajectoire cinématique et l'espace navigable ne soient pas mis à jour 3 fois avant de déclencher l'intervention de sécurité, donc après 0.4s. Cette dernière est ainsi déclenchée 1.6 s (la marge de sécurité) avant que la trajectoire cinématique ne soit considérée obsolète. Notons que lors de l'intervention de sécurité, les commandes de volant par l'application de contrôle-commande seront appliquées pendant les 1.6s de la marge de sécurité avant de maintenir à la même valeur la position du volant : les 1.6s de la marge de sécurité permettent ainsi de ralentir sensiblement le véhicule tandis qu'il est encore contrôlé par les commandes de l'application de contrôle-commande générées sur la dernière trajectoire obtenue avant la détection d'erreur.

14. Pour la nécessité de sécurité : « *Le Safety-Bag doit vérifier la cohérence de l'évolution de l'état cinématique. En cas d'incohérence de l'état cinématique, le Safety-Bag intervient en levant les alarmes, en inhibant l'accélération et en freinant modérément le véhicule. Il doit également maintenir l'angle volant.* », nous avons exprimé la condition de déclenchement de sécurité suivante :

    - Informellement : Le Safety-Bag doit vérifier que la nouvelle position de l'état cinématique est cohérente avec l'ancienne en considérant l'évolution prévue du véhicule.
    - Formellement :
      $prediction_t = pose_{t-\Delta t} + \overrightarrow{v}.\Delta t$
      $distance(pose_t, prediction_t) \leq k$; où Δt est la différence temporelle entre l'horodatage de l'état cinématique et l'état cinématique précédent ; et k : est une constante, seuil du déclenchement de la condition de sécurité
      Nous estimons la valeur de danger pour cette contrainte à environ 50cm : la distance à laquelle un véhicule devrait passer à côté d'un autre objet en ville. Malheureusement, les limites actuelles des mécanismes de perception et les simplifications du modèle physique réalisées dans cette contrainte nous amène à une valeur seuil empirique k d'environ 1 mètre (la distance latérale de sécurité en ville selon le code de la route). Être capable de garantir une diminution de cette imprécision de localisation reste encore un verrou technologique pour les véhicules autonomes.

15. Pour la nécessité de sécurité : « *Le Safety-Bag doit vérifier la cohérence temporelle de l'état cinématique du véhicule. Si l'état cinématique n'est*

pas temporellement cohérent, le Safety-Bag doit lever les alarmes, inhiber l'accélération, freiner modérément le véhicule et maintenir l'angle volant. », nous avons exprimé la condition de déclenchement de sécurité suivante :

- Informellement : A chaque réception de l'état cinématique, le Safety-Bag vérifie que le temps auquel l'état cinématique a été généré est postérieur à celui de la réception précédente, le Safety-Bag vérifie que l'horodatage de l'état cinématique n'est pas trop ancien et le Safety-Bag vérifie que l'horodatage de l'état cinématique n'est pas dans le futur.
- Formellement :
  `Check` : $h_{ci} \geq h_{ci-1}$ `et` $NOW - \varepsilon < h_{ci} \leq NOW$ `; avec` h_{ci} `l'horodatage de l'état cinématique numéro i.`

Cette condition d'ordre est une condition logique pour laquelle il n'y a pas une marge de sécurité.

16. Pour la nécessité de sécurité : « *Le Safety-Bag doit vérifier que l'horodatage de la trajectoire cinématique est plus récent que l'horodatage de l'espace navigable. Dans le cas contraire, le safety-Bag déclenche les alarmes, inhibe l'accélération et force des actions de sécurité telles que le freinage et le maintien d'angle volant.* », nous avons défini la condition de déclenchement de sécurité suivante :

    - Informellement : Le Safety-Bag doit vérifier que l'horodatage de l'espace navigable est plus récent que l'horodatage de la trajectoire cinématique. Le Safety-Bag doit vérifier que l'horodatage de l'espace navigable n'est pas trop ancien et le Safety-Bag doit vérifier que l'horodatage de l'espace navigable n'est pas dans le futur.
    - Formellement :
      check : $t_{HEN} \geq t_{HTC}$ `et` $NOW < t_{HEN} + \varepsilon$ `et` $NOW \geq t_{HTC}$ `; avec` t_{HEN} `: Horodatage de l'espace navigable et` t_{HTC} `: Horodatage de la trajectoire cinématique`

    Cette condition d'ordre est une condition logique pour laquelle il n'y a pas une marge de sécurité.

4.3.2 Discussion

Pour pouvoir fixer les conditions de déclenchement de sécurité, nous avons déterminé les valeurs de danger, qui représentent des valeurs frontière de l'état catastrophique conduisant à la défaillance du système et les marges de sécurité en analysant les processus physiques à l'œuvre et les incertitudes liées à l'acquisition et aux traitements des données.

Dans notre étude, nous avons déterminé les valeurs de danger et les marges de sécurité en nous basant principalement sur le jugement d'expert. Nous n'avons généralement pas considéré l'approche formelle du fait que nous n'avons pas de descriptions assez précises du véhicule pour pouvoir appliquer cette approche pour les nécessités de sécurité issues de l'AMDEC. Les choix des marges de sécurité prises pour le cas de la vivacité de l'application de contrôle-commande, et la vivacité de l'état cinématique sont basés sur des valeurs empiriques justifiées par des arguments et confirmées par des expérimentations.

Nous pouvons remarquer dans ce contexte que dans le cas de la vivacité de l'application de contrôle-commande (NS1) et la vivacité de l'état cinématique (NS11), les marges de sécurité (respectivement 0.03s et 0.3s) permettent au Safety-Bag d'intervenir avant que le véhicule ne soit plus contrôlé, mais est trop courte pour permettre au conducteur de reprendre la conduite en manuel avant que ce délai ne soit dépassé. Par contre, dans le cas de la trajectoire cinématique et de l'espace navigable, les 1.6 secondes sont proches du délai nécessaire pour qu'un conducteur vigilant ait repris la conduite et le contrôle en manuel.

Certaines de nos marges de sécurité ont été validées empiriquement, c'est-à-dire en s'appuyant dans certains cas sur des résultats expérimentaux. En effet, pour ne pas dépasser la vitesse limite, nous devons garder une marge de sécurité de $3km/h$ afin de prendre en compte la dynamique du véhicule entre le temps de détection de la vitesse et le temps où le Safety-Bag réagit en inhibant l'accélération. Cette valeur a été testée et validée en faisant des expérimentations sur le véhicule autonome réel dans notre laboratoire.

Pour ne pas dépasser l'accélération latérale limite, nous devons garder une marge de sécurité déterminée empiriquement de $1m.s^{-2}$.

Nous pouvons dire aussi que plus ces valeurs de marges de sécurité sont contraignantes, plus nous détecterons au plus tôt les erreurs pour augmenter la sécurité-innocuité du système, mais plus il y aura des faux positifs. Cela peut dans plusieurs cas dégrader la disponibilité du système en interrompant le mode autonome du véhicule et en passant de façon inopportune en conduite manuelle.

4.4 Identification des interventions de sécurité

Pour chaque nécessité de sécurité et après chaque violation de condition de déclenchement de sécurité (CdS), nous définissons les interventions nécessaires imposées par le Safety-Bag pour remettre le système dans un état sûr. Pour cela, nous identifions dans cette section les interventions attribuées à chaque nécessité de sécurité (définies dans les tableaux 4.7, 4.8, 4.9, 4.10 et 4.11) pour que le Safety-Bag puisse effectuer le rétablissement et la mise en état sûr du notre système en cas du non respect de ces nécessités de sécurité.

Tableau 4.7 – interventions de sécurité issues de l'analyse AMDEC

N°	**Nécessité de sécurité en langage naturel**	**Condition de déclenchement de sécurité**	**Marge de sécurité**	**Interventions de sécurité**		**Fin de l'intervention**
				Action de sécurité	Inhibition de sécurité	
A1	Le Safety-Bag doit vérifier la circulation des trames du bus CAN. En cas d'absence de communication pendant une durée trop importante, le Safety-Bag doit alerter le conducteur et repasser en mode manuel.	Les deux trames (`Steering_wheel` et `Front_wheel_speed`) sont émises au moins toutes les 20ms.	marge de sécurité = 30ms	• déclencher les alarmes, • freiner modérément à $f_{safe} = 5Volts$,	• inhiber l'accélération, • couple volant mis à 0 progressivement.	Pas de retour en conduite autonome
A2	Le Safety-Bag doit vérifier que les capteurs proprioceptifs latéraux (accélération latérale et gyromètre) fonctionnent correctement. Dans le cas contraire, le Safety-Bag déclenche les alarmes, inhibe l'accélération, freine modérément le véhicule et applique la valeur d'angle volant venant de l'application de contrôle-commande.	La ω_{AppCC} latérale n'est pas dans l'intervalle $[\omega_{SB} - \varepsilon\ ,\ \omega_{SB} + \varepsilon]$	–	• déclencher les alarmes, • freiner modérément à $f_{safe} = 5Volts$, sauf si l'application de contrôle-commande freine plus fort.	• inhiber l'accélération jusqu'à ce que l'opérateur reprenne le contrôle du véhicule.	Pas de retour en conduite autonome
A3	Le Safety-Bag doit vérifier que les capteurs proprioceptifs longitudinaux (vitesse longitudinale et accélération longitudinale) fonctionnent correctement. Dans le cas contraire, le Safety-Bag déclenche les alarmes, inhibe l'accélération et freine modérément le véhicule.	La V_{AppCC} longitudinale n'est pas dans l'intervalle $[V_{longitu} - \varepsilon\ ,\ V_{longitu} + \varepsilon]$	–	• déclencher les alarmes, • freiner modérément à $f_{safe} = 5Volts$, sauf si l'application de contrôle-commande freine plus fort.	• inhiber l'accélération jusqu'à ce que l'opérateur reprenne le contrôle du véhicule.	Pas de retour en conduite autonome

Tableau 4.8 – interventions de sécurité issues de deux analyses AMDEC et HazOp-UML (page 1)

N°	**Nécessité de sécurité en langage naturel**	**Condition de déclenchement de sécurité**	**Marge de sécurité**	**Interventions de sécurité**		**Fin de l'intervention**
				Action de sécurité	Inhibition de sécurité	
U1	Le Safety-Bag doit vérifier la vivacité de l'application de contrôle-commande. Si celle-ci n'envoie plus de commande, le Safety-Bag doit prévenir le conducteur et repasser en mode manuel et arrête le véhicule en allumant les feux de détresse.	Le Safety-Bag doit vérifier que la période de la mise à jour des commandes par l'application de contrôle-commande doit être toujours inférieure à une valeur seuil.	marge de sécurité = 30 ms	• déclencher les alarmes, • freiner modérément à $f_{safe} = 5Volts$, • maintenir l'angle volant.	• inhiber l'accélération,	Pas de retour en conduite autonome : Si les commandes sont de nouveau mises à jour, le couple volant et le freinage $f > f_{safe}$ sont appliqués.
U2	Le Safety-Bag doit vérifier la cohérence temporelle de l'application de contrôle-commande. En cas d'incohérence temporelle, le Safety-Bag doit assurer la mise en sécurité du système en levant les alarmes et en repassant en mode manuel.	A chaque réception de données fournies par l'application de contrôle-commande, le Safety-Bag vérifie que le temps auquel elles ont été générées est postérieur à celui de la réception précédente, le Safety-Bag vérifie que l'horodatage des données n'est pas trop ancien et le Safety-Bag vérifie que l'horodatage des données n'est pas dans le futur.	–	• déclencher les alarmes, • freiner modérément à $f_{safe} = 5Volts$, • maintenir l'angle volant.	• inhiber l'accélération,	Pas de retour en conduite autonome : Les commandes de l'application de contrôle-commande ne sont plus appliquées.
U3	Le Safety-Bag doit vérifier les bornes de vitesse. Dans le cas contraire, le Safety-Bag déclenche les alarmes, bloque les commandes d'accélération de l'application de contrôle-commande et assure la reprise en mode manuel.	$V \leq 50km/h-$ marge de sécurité	la marge de sécurité = 3 km/h	• déclencher les alarmes,	• inhiber l'accélération.	Retour en conduite autonome après que la vitesse soit devenue inférieure à 44 km/h.
U4	Le Safey-Bag doit vérifier le contrôle dynamique du véhicule : Il doit vérifier que l'accélération latérale reste inférieure à une valeur seuil. Dans le cas contraire, le Safety-Bag doit déclencher les alarmes et assurer la reprise en mode manuel.	$(vitesse_{vehicule} - 15) * \lvert angle_{volant} \rvert \leq 2000$	la marge de sécurité = $1m.s^{-2}$; l'accélération latérale maximale est de $5m.s^{-2}$	• déclencher les alarmes, • freiner fort à $f_{urgence} = 7Volts$.	• inhiber l'accélération. On note que la commande du volant est toujours appliquée.	Retour en conduite autonome dès que la condition de sécurité est à nouveau vérifiée
U5	Le Safety-Bag doit vérifier que l'accélérateur fonctionne correctement. Il doit vérifier que l'intensité dans le moteur électrique correspond à la valeur de la commande d'accélération fournie par l'application de contrôle-commande. Dans le cas contraire, le Safety-Bag lève une alarme, inhibe l'accélération et freine modérément.	Le Safety-Bag doit vérifier que l'intensité dans le moteur électrique est proche de l'intensité nominalement associée à la valeur de la commande.	–	• déclencher les alarmes, • freiner modérément à $f_{safe} = 5Volts$.	• inhiber l'accélération.	Pas de retour en conduite autonome

Tableau 4.9 – Interventions de sécurité issues de deux analyses AMDEC et HazOp-UML (page 2)

N°	**Nécessité de sécurité en langage naturel**	**Condition de déclenchement de sécurité**	**Marge de sécurité**	**Interventions de sécurité**		**Commentaires**
				Action de sécurité	Inhibition de sécurité	
U6	Le Safety-Bag doit vérifier que le frein fonctionne correctement. Il doit vérifier que la pression ne s'écarte pas de la valeur d'une fonction donnée de la commande du frein. Dans le cas contraire, le Safety-Bag déclenche les alarmes, empêche les commandes d'accélération et freine modérément.	Le Safety-Bag doit vérifier que la pression du frein est proche de la pression nominalement associée à la commande.	–	• déclencher les alarmes, • freiner modérément à $f_{safe} = 5Volts$,	• inhiber l'accélération.	Pas de retour en conduite autonome
U7	Le Safety-Bag doit vérifier que la direction fonctionne correctement : Il doit vérifier que les angles volant évoluent dans le sens prévu selon le couple volant appliqué. Dans le cas contraire, une alerte doit être déclenchée vers le conducteur et le Safety-Bag doit mettre le véhicule en état sûr en passant en mode de conduite manuelle.	Le volant tourne à l'envers par rapport à la commande envoyée ou ne tourne pas avec une commande qui devrait être suffisante.	–	• déclencher les alarmes, • freiner modérément à $f_{safe} = 5Volts$.	• inhiber l'accélération.	Pas de retour en conduite autonome
U8	Le Safety-Bag doit vérifier que la direction fonctionne correctement : Il doit vérifier que l'intensité sur le moteur de direction correspond aux consignes envoyées par l'application de contrôle-commande. En cas de non correspondance, le Safety-Bag doit alerter le conducteur et doit mettre le système en état sûr en redonnant la main au conducteur.	Le Safety-Bag vérifie que l'intensité sur le moteur de direction est proche de l'intensité nominalement associée à la commande.	–	• déclencher les alarmes, • freiner modérément à $f_{safe} = 5Volts$.	• inhiber l'accélération.	Pas de retour en conduite autonome

Tableau 4.10 – Interventions de sécurité issues de l'analyse HazOp (page 1)

N°	Nécessité de sécurité en langage naturel	Condition de déclenchement de sécurité	Marge de sécurité	Interventions de sécurité		Fin de l'intervention
				Action de sécurité	Inhibition de sécurité	
H1	Le Safety-Bag doit vérifier la vivacité de l'état cinématique. Si l'état cinématique n'est pas mis à jour, le Safety- Bag intervient en déclenchant les alarmes, en inhibant des commandes de l'accélération et/ou forçant une action de sécurité telle que le freinage du véhicule et le maintien de l'angle volant après un certain délai.	la période de la mise à jour de l'état cinématique doit être toujours inférieure à une valeur seuil.	marge de sécurité = 0.3 s	• déclencher les alarmes, • freiner modérément à $f_{safe} = 5Volts$, sauf si l'AppCC freine plus fort, • maintenir l'angle volant après 1/4 secondes.	• inhiber l'accélération.	Pas de retour en conduite autonome
H2	Le Safety-Bag doit vérifier la cohérence de l'évolution de l'état cinématique. En cas d'incohérence dimensionnelle de l'état cinématique, le Safety-Bag intervient en levant les alarmes, en inhibant l'accélération et en freinant modérément le véhicule. Il doit également maintenir l'angle volant.	Le Safety-Bag doit vérifier que la nouvelle position de l'état cinématique est cohérente avec l'ancienne en considérant l'évolution prévue du véhicule.	–	• déclencher les alarmes, • freiner modérément à $f_{safe} = 5Volts$, • maintenir l'angle.	• inhiber l'accélération.	Pas de retour en conduite autonome
H3	Le Safety-Bag doit vérifier la cohérence temporelle de l'état cinématique. Si l'état cinématique n'est pas temporellement cohérent, le Safety-Bag doit lever les alarmes, inhiber l'accélération, freiner modérément le véhicule et maintenir l'angle volant.	A chaque réception de l'état cinématique, le Safety-Bag vérifie que le temps auquel l'état cinématique a été généré est postérieur à celui de la réception précédente, le Safety-Bag vérifie que l'horodatage de l'état cinématique n'est pas trop ancien et le Safety-Bag vérifie que l'horodatage de l'état cinématique n'est pas dans le futur.	–	• déclencher les alarmes, • freiner modérément à $f_{safe} = 5Volts$, • maintenir l'angle.	• inhiber l'accélération.	Pas de retour en conduite autonome
H4	Le Safety-Bag doit vérifier la vivacité de la trajectoire cinématique. Dans le cas contraire, le Safety-Bag doit déclencher les alarmes, inhiber l'accélération, freiner et maintenir la position d'angle volant après un certain temps.	la période de la mise à jour de la trajectoire cinématique doit être toujours supérieure à une valeur seuil.	marge de sécurité = 1.6 s	• déclencher les alarmes, • freiner modérément à $f_{safe} = 5Volts$, sauf si l'AppCC freine plus fort, • maintenir la position d'angle volant après 1.6 secondes.	• inhiber l'accélération.	Pas de retour en conduite autonome
H5	Le Safety-Bag doit vérifier la vivacité de l'espace navigable. Dans le cas contraire, le Safety-Bag déclenche une alarme, inhibe l'accélération, freine et maintient l'angle volant après un certain temps.	la période de la mise à jour de l'espace navigable doit être toujours supérieure à une valeur seuil.	marge de sécurité = 1.6 s	• déclencher les alarmes, • freiner modérément à $f_{safe} = 5Volts$, sauf si l'AppCC freine plus fort, • maintenir la position d'angle volant après 1.6 secondes.	• inhiber l'accélération.	Pas de retour en conduite autonome

Tableau 4.11 – Interventions de sécurité issues de l'analyse HazOp page (2)

N°	**Nécessité de sécurité en langage naturel**	**Condition de déclenchement de sécurité**	**Marge de sécurité**	**Interventions de sécurité**		**Fin de l'intervention**
				Action de sécurité	Inhibition de sécurité	
H6	Le Safety-Bag doit vérifier que l'horodatage de la trajectoire cinématique est plus récent que l'horodatage de l'espace navigable. Dans le cas contraire, le Safety-Bag déclenche les alarmes, inhibe l'accélération et force des actions de sécurité telles que le freinage et le maintien d'angle volant.	Le Safety-Bag doit vérifier que l'horodatage de l'espace navigable est plus récent que l'horodatage de la trajectoire cinématique. Le Safety-Bag doit vérifier que l'horodatage de l'espace navigable n'est pas trop ancien et le Safety-Bag doit vérifier que l'horodatage de l'espace navigable n'est pas dans le futur.	–	• déclencher les alarmes, • freiner modérément à $f_{safe} = 5Volts$, • maintenir la position d'angle volant.	• inhiber l'accélération.	Pas de retour en conduite autonome

1. En cas de violation de la CdS : « *Le Safety-Bag doit vérifier la vivacité de l'application de contrôle-commande* », le Safety-Bag :

   - déclenche les alarmes,
   - inhibe l'accélération (la consigne de l'accélérateur est mise à zéro),
   - et force deux actions de sécurité :
     - ◇ Il freine d'abord modérément à $5Volts$ (c'est une tension de freinage utilisée par le Safety-Bag pour arrêter le véhicule),
     - ◇ et maintient la position d'angle volant qui est la dernière position correcte connue.

   Dans cette intervention, il n'y a pas de retour possible en conduite autonome.

   Si les commandes de l'application de contrôle-commande sont de nouveau mises à jour après la détection d'erreur, elle peut contrôler le véhicule. Puisque l'accélérateur est inhibé et le frein déclenché dès la détection d'erreur, seul le couple volant peut cependant être commandé par l'application.

2. En cas de violation des 4 CdS suivantes :

   - *« Le Safety-Bag doit vérifier la cohérence temporelle de l'application de contrôle-commande*
   - *Le Safety-Bag doit vérifier la cohérence de l'évolution sur l'état cinématique,*
   - *Le Safety-Bag doit vérifier la cohérence temporelle de l'état cinématique,*
   - *Le Safety-Bag doit vérifier que l'horodatage de la trajectoire cinématique est plus récent que l'horodatage de l'espace navigable* »,

   le Safety-Bag :

   - active les alarmes,
   - inhibe l'accélération (la consigne de l'accélérateur est mise à zéro),
   - et impose deux actions de sécurité :
     - ◇ il freine modérément à $f_{safe} = 5Volts$,
     - ◇ et maintient la position d'angle volant au cas où de futures commandes seraient aberrantes.

   Ici, il n'y aura pas de retour en conduite autonome et les commandes de l'application de contrôle-commande ne sont plus appliquées. L'opérateur doit passer alors en mode manuel.

3. En cas de non-respect de la CdS : « *Le Safety-Bag doit vérifier les bornes de vitesse* », le Safety-Bag :

   - active les alarmes,
   - et inhibe toute commande d'accélération de l'application de contrôle-commande.

   Dans ce cas, la conduite autonome pourra être rétablie une fois que la vitesse du véhicule sera à nouveau descendue en-dessous de la vitesse limite de sécurité $50km/h$ diminuée de la marge de sécurité 3 km/h et diminuée en plus d'une valeur hystérésis (environ 3 km/h) donc une vitesse inférieure à 44 km/h.

   La valeur d'hystérésis évite que, à la limite de déclenchement de la nécessité, le Safety-Bag oscille entre les deux états d'activation et d'inhibition de l'accélération.

4. En cas de non-respect de la CdS : « *Le Safety-Bag doit vérifier la circulation des trames CAN* », le Safety-Bag :

   - déclenche les alarmes,
   - inhibe l'accélération ($acc = 0$ Volts),
   - inhibe le couple volant qui doit être mis à zéro progressivement,
   - et force l'action du freinage modérément à $f_{safe} = 5$ Volts.

     Si l'application de contrôle-commande freine plus fort, c'est-à-dire $f > f_{safe}$, la consigne du frein sera appliquée alors à cette valeur.

   Dans cette situation, nous ne pouvons pas reprendre la conduite autonome. Si le CAN est défaillant, l'application de contrôle-commande n'a plus accès à des données proprioceptives, et n'est plus capable de commander correctement le véhicule.

5. En cas de non-respect de la CdS : « *Le Safety-Bag doit vérifier que les capteurs latéraux (vitesse de rotation et accélération latérale) fonctionnent correctement.* », le Safety-Bag :

   - lève les alarmes,
   - inhibe l'accélération,
   - et freine modérément à $f_{safe} = 5$ Volts (ou plus si demandé par l'application de contrôle-commande).

Dans ce cas, la dernière position du volant (considérée comme correcte puisque établie avant la détection de la défaillance) est maintenue tandis que le Safety-Bag arrête le véhicule en appliquant un freinage jusqu'à l'arrêt ou la reprise en mode manuel.

6. En cas de non-respect de la CdS : « *Le Safety-Bag doit vérifier que les capteurs longitudinaux (vitesse longitudinale et accélération longitudinale) fonctionnent correctement.* », le Safety-Bag :

   - lève les alarmes,
   - inhibe l'accélération,
   - et freine modérément à $f_{safe} = 5$ Volts (ou plus si demandé par l'application de contrôle-commande).

   Dans ce cas, le contrôle du volant reste confié à l'application de contrôle commande, tant que le Safety-Bag ne détecte pas d'autres incohérences.

7. En cas de non-respect de la CdS : « *Le Safety-Bag doit vérifier les limites dynamiques en passant un virage* », le Safety-Bag :

   - déclenche les alarmes,
   - inhibe l'accélération,
   - et force un freinage fort d'urgence $f_{urgence} = 5.35$ Volts pour réduire la vitesse.

   On note que la commande du volant est toujours appliquée dans ce cas et que nous pouvons reprendre la conduite autonome dès que la condition de déclenchement de sécurité est à nouveau vérifiée. Dès que la vitesse est compatible avec le passage du virage, le freinage est relâché.

8. En cas de non-respect de 4 CdS suivantes :

   - *« Le Safety-Bag doit vérifier que l'intensité dans le moteur électrique correspond à une fonction de la commande d'accélération*
   - *Le Safety-Bag doit vérifier que la pression correspond à une fonction de la commande du frein,*
   - *Le Safety-Bag doit vérifier que les angles volant évoluent dans le sens prévu selon le couple volant appliqué,*
   - *Le Safety-Bag doit vérifier que l'intensité sur le moteur de direction correspond aux consignes qu'il a envoyé* »,

le Safety-Bag :

- lève les alarmes,
- inhibe l'accélération,
- et freine modérément à $f_{safe} = 5$ Volts.

Dans les quatre cas, nous appliquons le couple volant et nous actionnons le frein. Cependant, cette version du Safety-Bag a peu de moyens de s'assurer que ces actions seront suivies d'effets, et la réaction du conducteur peut ainsi être un aspect important du rétablissement. Pour les quatre cas, il n'y a pas un retour en conduite autonome.

9. En cas de non-respect de la CdS : « *Le Safety-Bag doit vérifier la vivacité de l'état cinématique* », le Safety-Bag :

   - déclenche les alarmes,
   - inhibe l'accélération
   - et force les deux actions de sécurité suivantes :
     - ◇ freiner modérément à $f_{safe} = 5$ Volts, sauf si l'application de contrôle-commande freine plus fort,
     - ◇ maintenir la position de l'angle volant après un délai de 0.3 secondes (marge de sécurité).

   Nous pouvons alors laisser l'application de contrôle-commande diriger le véhicule pendant un court délai. Ce délai correspond à la marge de sécurité fixée à 0.3 secondes pour la mise à jour de l'état cinématique. Pour obtenir cette valeur, détaillée dans la section précédente, nous avons considéré la fréquence minimale de mise à jour de l'état cinématique permettant de conduire un véhicule et retranché le délai de la détection de la défaillance.

   En effet, l'état cinématique étant correct, il peut servir à l'application de contrôle-commande pour contrôler le véhicule pendant encore une fraction de seconde. Même si l'accélérateur est inhibé et le freinage est déclenché dès la détection de défaillance, le couple volant peut continuer à être appliqué pendant cette fraction de seconde (0.3s). Ensuite, le couple volant sera maintenu jusqu'à l'arrêt du véhicule ou le retour en conduite manuelle. Dans ce cas, nous ne pouvons pas reprendre la conduite autonome.

10. En cas de non-respect des deux CdS : « *Le Safety-Bag doit vérifier la vivacité de la trajectoire cinématique* » et « *Le Safety-Bag doit vérifier la vivacité de l'espace navigable* », le Safety-Bag :

- déclenche les alarmes,
- inhibe l'accélération,
- et impose les deux actions de sécurité suivantes :
  - ◇ freiner modérément à $f_{safe} = 5$ Volts, sauf si l'application de contrôle-commande freine plus fort,
  - ◇ maintenir la position de l'angle volant après un délai de 1.6 secondes (marge de sécurité).

Nous pouvons alors laisser l'application de contrôle-commande diriger le véhicule pendant une courte durée de temps. Nous avons fixé ce délai à 1.6 secondes pour la trajectoire cinématique et l'espace navigable. Pour calculer cette valeur et comme le cas pour l'état cinématique, nous avons considéré la fréquence minimale de mise à jour de la trajectoire cinématique ou de l'espace navigable permettant de conduire un véhicule et nous avons retranché le délai de détection de la défaillance.

En effet, la trajectoire cinématique et l'espace navigable étant corrects, ils peuvent être utilisés pour conduire le véhicule pendant environ 2 secondes moins le délai de détection de leur non mise à jour de 0.4 secondes, le couple volant produit par l'application de contrôle-commande, peut continuer à être appliqué pendant 1.6 secondes. Dans ce cas, nous ne pouvons pas reprendre la conduite autonome. L'opérateur doit passer le véhicule en mode de conduite manuel.

Nous décrivons maintenant le comportement du Safety-Bag Rules Checker via-à-vis des différentes interventions de sécurité indiquées précédemment. Le tableau 4.12 récapitule les différents types d'interventions de sécurité que le Safety-Bag est susceptible d'effectuer. Il illustre le fait que le Safety-Bag tente de faire aux mieux avec les informations encore disponibles et fiables. Dans toutes les interventions de sécurité (indiquées dans le tableau récapitulatif 4.12), l'accélération est inhibée. Les actions de sécurité incluent toujours un freinage.

Tableau 4.12 – Tableau récapitulatif des interventions de sécurité

État	**Nécessité de sécurité (N°)**	**couple volant**	**accélérateur**	**frein**	**Condition de transition d'état**	**État suivant**
a	U1	maintenir la position	$0Volts$	$5Volts$	Application de contrôle-commande active	e
b	U2 + H2 +H3 + H6	maintenir la position	$0Volts$	$5Volts$	_	b
c	U3	couple appliqué	$0Volts$	frein appliqué	$V < 44km/h$	retour en fonctionnement nominal, conduite autonome
d	A1	mise à 0 progressivement	$0Volts$	$5Volts$	_	d
e	A2 + A3	couple appliqué	$0Volts$	$max(f_{safe}, f_{AppCC})$	_	e
f	U4	couple appliqué	$0Volts$	$5.35Volts$	condition de sécurité U4 vraie	retour en fonctionnement nominal, conduite autonome
g	U5 + U6 + U7 + U8	couple appliqué	$0Volts$	$5Volts$	_	g
h	H1	couple appliqué	0 Volts	$max(f_{safe}, f_{AppCC})$	$0.3s$	j
i	H4 + H5	couple appliqué	$0Volts$	$max(f_{safe}, f_{AppCC})$	$1.6s$	j
j	_	maintenir la position	$0Volts$	$5Volts$	L'horodatage de l'état cinématique, l'horodatage de la trajectoire cinématique et l'horodatage de l'espace navigable sont récents.	e

Les *états* référencés de « a » à « j » correspondent aux sous-états de l'état d'alerte du statechart décrivant le comportement des composants du Safety-Bag Rules Checker 4.2 en imposant les différentes interventions de sécurité sur les actionneurs du véhicule.

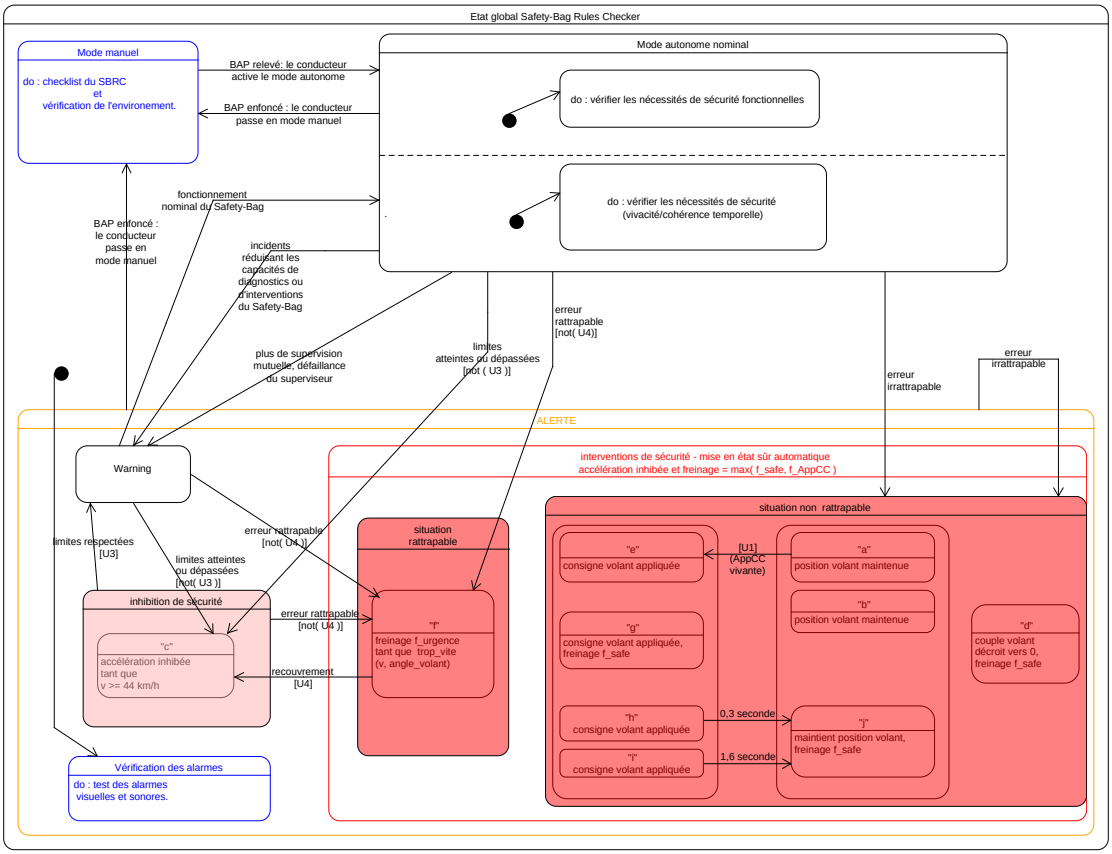

Figure 4.2 – Statechart décrivant le comportement du Safety-Bag Rules Checker

Nous avons introduit l'état « j », qui ne correspond à aucune nécessité de sécurité, mais qui décrit les actions à effectuer à la suite des états « h » et « i » une fois dépassé le délai pendant lequel nous pouvons faire confiance à l'application de contrôle-commande.

Remarquons que lorsque toutes les données (l'état cinématique, la trajectoire cinématique et l'espace navigable) ont été mises à jour, nous pourrions envisager la transition vers l'état « e » à partir de l'état « j » et permettre ainsi au couple volant transmis par l'application de contrôle-commande de commander à nouveau le véhicule par la Direction Assistée Électrique (DAE).

Cependant, nous ne sommes pas sûrs que l'application de contrôle-commande soit réellement revenue dans un état cohérent. Comme nous freinons fortement le véhicule, une commande aberrante sur l'angle au volant pourrait amener une perte de contrôle, et nous préférons donc ne pas appliquer les commandes volant

potentiellement fournies par l'application de contrôle commande.

Remarque

Les interventions du Safety-Bag peuvent associer des actions prédéfinies du Safety-Bag (valeur de freinage, maintenir la position d'angle volant, etc.) et des commandes produites par l'application de contrôle-commande. Par exemple, le freinage commandé par l'application de contrôle-commande est appliqué s'il est plus puissant que celui demandé par le Safety-Bag.

4.5 Liste des nécessités de sécurité à tester expérimentalement sur notre véhicule autonome expérimental

Nous présentons dans cette section les nécessités de sécurité que nous pouvons implémenter sur le Safety-Bag de notre véhicule expérimental afin de les valider et d'évaluer le comportement du Safety-Bag en cas de défaillances et suite à une violation de ces nécessités de sécurité. Le tableau 4.13 décrit les nécessités de sécurité possibles à évaluer ainsi que les nécessités de sécurité que nous avons implémentées et expérimentées réellement dans le chapitre suivant.

Dans notre étude, nous n'allons tester que les six nécessités de sécurité suivantes que nous trouvons les plus simples à implémenter dans le Safety-Bag et pour lesquelles nous sommes capables d'injecter des fautes représentatives :

- A1 : Le Safety-Bag doit vérifier la circulation des trames du bus CAN. En cas d'absence de communication pendant une durée trop importante, le Safety-Bag doit alerter le conducteur et repasser en mode manuel.
- U1 : Le Safety-Bag doit vérifier la vivacité de l'application de contrôle-commande. Si celle-ci n'envoie plus de commande, le Safety-Bag doit prévenir le conducteur pour qu'il repasse en mode manuel et arrête le véhicule en allumant les feux de détresse.
- U3 : Le Safety-Bag doit vérifier les bornes de vitesse. Dans le cas contraire, le Safety-Bag déclenche les alarmes, bloque les commandes d'accélération de l'application de contrôle-commande et assure la reprise en mode manuel.
- U4 : Le Safey-Bag doit vérifier le contrôle dynamique du véhicule : Il doit vérifier que l'accélération latérale reste inférieure à une valeur seuil. Dans le

cas contraire, le Safety-Bag doit déclencher les alarmes et assurer la reprise en mode manuel.

- H1 : Le Safety-Bag doit vérifier la vivacité de l'état cinématique. Si l'état cinématique n'est pas mis à jour, le Safety- Bag intervient en déclenchant les alarmes, en inhibant des commandes de l'accélération et/ou forçant une action de sécurité telle que le freinage du véhicule et le maintien de l'angle volant après un certain délai.

- H4 : Le Safety-Bag doit vérifier la vivacité de la trajectoire cinématique. Dans le cas contraire, le Safety-Bag doit déclencher les alarmes, inhiber l'accélération, freiner et maintenir la position d'angle volant après un certain temps.

Cependant, certaines nécessités de sécurité élicitées en s'appuyant sur les deux méthodes d'analyse de risques AMDEC et HazOp-UML ne peuvent pas être testées pour plusieurs raisons :

- Les 4 nécessités de sécurité :

  - ◇ *« U1 : Le Safety-Bag doit vérifier la cohérence temporelle de l'application de contrôle-commande. En cas d'incohérence temporelle, le Safety-Bag doit assurer la mise en sécurité du système en levant les alarmes et en repassant en mode manuel.*
  - ◇ *H2 : Le Safety-Bag doit vérifier la cohérence de l'évolution de l'état cinématique. En cas d'incohérence de l'état cinématique, le Safety-Bag intervient en levant les alarmes, en inhibant l'accélération et en freinant modérément le véhicule. Il doit également maintenir l'angle volant.*
  - ◇ *H3 : Le Safety-Bag doit vérifier la cohérence temporelle de l'état cinématique. Si l'état cinématique n'est pas temporellement cohérent, le Safety-Bag doit lever les alarmes, inhiber l'accélération, freiner modérément le véhicule et maintenir l'angle volant.*
  - ◇ *H6 : Le Safety-Bag doit vérifier que l'horodatage de la trajectoire cinématique est plus récent que l'horodatage de l'espace navigable. Dans le cas contraire, le Safety-Bag déclenche les alarmes, inhibe l'accélération et force des actions de sécurité telles que le freinage et le maintien d'angle volant. »*

  ont un comportement similaire à la nécessité de sécurité « *U1 : Le Safety-Bag doit vérifier la vivacité de l'application de contrôle-commande. Si celle-ci n'envoie plus de commande, le Safety-Bag doit prévenir le conducteur et repasser*

Nécessité de sécurité N°	**Implémentation possible**	**Implémentation effectuée**	**Explication**	**Réf**
A1	oui	oui	• implémentation de la nécessité de sécurité dans le Safety-Bag, • injection de faute identifiée et réalisable, • La faute est injectée manuellement par l'expérimentateur en basculant l'interrupteur qui bloque la passerelle CAN entre le réseau véhicule et l'application de contrôle-commande.	1
A2	oui	non	• Il est facile de vérifier la cohérence des données sur une piste réelle (lorsque le véhicule est mobile). • Le véhicule ne se déplace pas réellement sur le banc VILAD. Les capteurs (accéléromètre et gyromètre) ne peuvent pas détecter l'accélération latérale et la vitesse angulaire.	–
A3	oui	non	• Il est facile de faire des tests de cohérence sur la piste réelle et sur la piste virtuelle simulée par le banc VILAD. • Le Safety-Bag devrait disposer des sources indépendantes pour pouvoir vérifier la cohérence des données (de la vitesse) avec les valeurs fournies par les capteurs proprioceptifs du véhicule, mais ce n'est pas le cas dans notre étude.	–
U1	oui	oui	• implémentation de la nécessité de sécurité dans le Safety-Bag, • injection de faute identifiée et réalisable, • les résultats nous semblent intéressants afin d'évaluer le comportement du Safety-Bag et de valider la nécessité de sécurité.	2
U3	oui	oui	• implémentation de la nécessité de sécurité dans le Safety-Bag, • injection de faute identifiée et réalisable.	3
U4	oui	oui	• implémentation de la nécessité de sécurité dans le Safety-bag • injection de faute identifiée et réalisable, • cette expérience vise à tester la position du volant en passant un virage	4
U5/U6/U7/U8	non	non	• Le véhicule n'est pas équipé respectivement des capteurs d'intensité dans le moteur électrique du véhicule, dans le moteur électrique de la DAE et de capteur de la pression du frein.	–
H1	oui	oui	• implémentation de la nécessité de sécurité dans le Safety-Bag, • injection de faute identifiée et réalisable,	5
U2/H2/H3/H6	oui	non	• Les injections de fautes vont être introduites de la même manière que la nécessité de sécurité U1. Le Safety-Bag déclenchera donc les mêmes interventions de sécurité. • La vérification est simple, mais nous n'aurons pas les valeurs de l'état cinématique, de la trajectoire cinématique et de l'espace navigable par Scanner. • On ne va pas expérimenter ces nécessités de sécurité vu qu'elles n'apporteront que peu d'informations supplémentaires par rapport à la nécessité U1.	–
H4	oui	oui	• implémentation de la nécessité de sécurité dans le Safety-Bag, • injection de faute identifiée et réalisable, • On peut obtenir des résultats intéressants par rapport à d'autres scénarios du fait que les commandes restent valides et exploitables pendant un certain délai. Les scénarios devront intégrer cette caractéristique.	6
H5	oui	non	• implémentation de la nécessité de sécurité facile dans le Safety-Bag, • nous n'avons pas un environnement adéquat pour effectuer des expériences de validation pour cette nécessité de sécurité	–

Tableau 4.13 – Nécessités de sécurité expérimentées et évaluées en réalité.

en mode manuel et arrête le véhicule en allumant les feux de détresse. » et « *H1 :Le Safety-Bag doit vérifier la vivacité de l'état cinématique. Si l'état cinématique n'est pas mis à jour, le Safety- Bag intervient en déclenchant les alarmes, en inhibant des commandes de l'accélération et/ou forçant une action de sécurité telle que le freinage du véhicule et le maintien de l'angle volant après un certain délai.* » Considérant que nous utilisons un simulateur scripté d'application de contrôle-commande d'une part afin de mieux contrôler les risques d'expérimentation et d'autre part parce que nous n'avons pas actuellement d'application de contrôle commande fonctionnant avec notre véhicule, ces nécessités de sécurité nous semblaient moins intéressantes comme les fautes injectées sur des variables internes d'une application de contrôle commande ne seront pas représentatives de fautes réelles avec notre simulateur scripté. Nous avons ainsi préféré nous concentrer sur d'autres nécessités.

- Le véhicule n'est pas équipé de certains capteurs (tels que les capteurs d'intensité dans le moteur électrique lié à la commande d'accélérateur, sur le moteur électrique de la direction assistée électrique ainsi que la pression liée à la commande du frein, etc.).

  Sur le banc VILAD, les capteurs inertiels ne peuvent pas fonctionner car le véhicule est immobile et nous avons choisi de ne pas utiliser les valeurs calculées par le banc VILAD car celles-ci ne sont pas émises avec une fréquence suffisante par rapport au fréquence de travail du Safety-Bag. Nous utilisons le produit de l'angle volant du véhicule par sa vitesse.

  Cela explique que nous ne pouvons pas implémenter et tester les 4 nécessités de sécurité issues de l'AMDEC suivantes :

  - ◇ *« A2 : Le Safety-Bag doit vérifier que la vitesse de rotation et l'accélération latérale sont cohérentes avec les données de capteurs liés au Safety-Bag. Dans le cas contraire, le Safety-Bag déclenche les alarmes, inhibe l'accélération, freine modérément le véhicule et applique la valeur d'angle volant venant de l'application de contrôle-commande.*
  - ◇ *U5 : Le Safety-Bag doit vérifier que l'intensité dans le moteur électrique correspond à une fonction de la commande d'accélération. Dans le cas contraire, le Safety-Bag lève une alarme, inhibe l'accélération et freine modérément.*
  - ◇ *U6 : Le Safety-Bag doit vérifier que la pression correspond à une fonction de la commande du freins. Dans le cas contraire, le Safety-Bag déclenche les alarmes, empêche les commandes d'accélération et freine modérément.*

◇ *U8 : Le Safety-Bag doit vérifier que l'intensité sur le moteur de direction correspond aux consignes envoyées par l'application de contrôle-commande. En cas de non correspondance, le Safety-Bag doit alerter le conducteur et doit mettre le système en état sûr en redonnant la main au conducteur. »*

- La nécessité de sécurité issue de l'AMDEC : « *U7 : Le Safety-Bag doit vérifier que la direction fonctionne correctement : Il doit vérifier que les angles volant évoluent dans le sens prévu selon le couple volant appliqué. Dans le cas contraire, une alerte doit être déclenchée vers le conducteur et le Safety-Bag doit mettre le véhicule en état sûr en passant en mode de conduite manuelle.* » est possible à implémenter et vérifier. En revanche, pour injecter des fautes correspondantes, nous devrions insérer un calculateur supplémentaire qui filtrerait les commandes analogiques envoyées par le Safety-Bag Rules Checker et permettrait de les modifier lorsqu'on le souhaite. Nous n'avons pas réalisé ce développement.

4.6 Conclusion

Comme nous l'avons évoqué précédemment, le but de notre composant Safety-Bag est de réduire les risques dans un véhicule autonome expérimental. Il est chargé de surveiller en ligne un ensemble des nécessités de sécurité et de détecter les défaillances logicielles et matérielles en surveillant des propriétés systémiques du véhicule. Le Safety-Bag peut ensuite rétablir le système en rejetant les commandes dangereuses, en appliquant des actions de sécurité (telles que le freinage, l'inhibition d'accélération et le maintien de volant), ou finalement en redonnant le contrôle à l'opérateur quand on ne peut plus avoir confiance dans l'application de contrôle commande.

Nous avons présenté dans notre étude les deux méthodes d'analyse de risques AMDEC et HazOp-UML. Ces méthodes sont utilisées pour spécifier les nécessités de sécurité requises pour détecter et réagir aux problèmes. Ils analysent le système de deux points de vue différents, qui nous apparaissent complémentaires pour identifier les nécessités de sécurité. Cependant, il convient de noter que les résultats de ces analyses (en particulier l'analyse HazOp-UML) dépendent des compétences et des choix de modélisation de l'analyste. De plus, toutes les exigences de sécurité ne sont pas nécessairement applicables par le Safety-Bag. En effet, ce dernier doit d'une part rester suffisamment simple pour être facilement validé, et d'autre part ne pas impliquer de décisions basées sur des composants peu sûrs de fonctionnement (comme les mécanismes de fusion de données ou d'intelligence artificielle).

Chaque nécessité de sécurité est composée d'une condition de déclenchement de sécurité et d'une intervention de sécurité (action ou inhibition) imposée par le Safety-Bag.

Nous avons spécifié dans ce chapitre les différentes conditions de déclenchement de sécurité évaluables par le Safety-Bag. Chaque condition de déclenchement de sécurité est généralement un contrôle de vraisemblance (temporelle ou sur les données) ou un respect d'une valeur seuil, déterminée empiriquement ou avec une analyse de sécurité par une valeur de danger dont est exclue la marge de sécurité. De plus, nous avons associé pour chaque nécessité de sécurité les inhibitions et les actions de sécurité possibles imposées par le Safety-Bag afin de remettre le système dans un état sûr. Généralement, le Safety-Bag combine les deux types d'intervention de sécurité (action et inhibition) pour faire le rétablissement. Il déclenche également les alertes physiques, inhibe l'accélération et force une action soit en freinant, soit en maintenant l'angle volant.

En raison des contraintes de temps et à cause de la non disponibilité de certains capteurs dans notre véhicule (tels que les capteurs d'intensité dans le moteur électrique lié à la commande d'accélérateur par exemple), nous n'allons tester que 6 nécessités de sécurité parmi l'ensemble des nécessités de sécurité définies par les deux méthodes d'analyse de risques AMDEC et HazOp-UML.

Dans le chapitre suivant, nous allons montrer les impacts du Safety-Bag sur le véhicule expérimental étudié (en analysant l'AMDEC et l'arbre de défaillances du véhicule autonome expérimental en introduisant le Safety-Bag). Nous présenterons ensuite les expérimentations effectuées sur le le banc VILAD (Vehicle in The Loop for Autonomous Driving) sans intervention humaine et sur la piste réelle Seville avec intervention humaine afin de valider certaines nécessités de sécurité et afin d'évaluer le comportement de notre Safety-Bag en injectant des fautes.

Validations expérimentales des nécessités de sécurité du composant Safety-Bag

Sommaire

Nous présentons maintenant l'implémentation de notre système de sécurité-innocuité Safety-Bag dans le véhicule autonome expérimental de type Fluence IRIS du laboratoire Heudiasyc. Nous rappelons que ce dispositif de sécurité permet de détecter des erreurs présentes dans le système, notamment en identifiant la violation de certaines nécessités de sécurité et d'intervenir soit en inhibant des commandes soit en forçant une action de sécurité pour remettre le système dans un état sûr.

L'objectif de ce chapitre est d'évaluer le comportement du Safety-Bag suite à la détection d'une défaillance et de valider certaines nécessités de sécurité vérifiables et implémentées par le Safety-Bag.

Pour cela, nous allons commencer par décrire la robotisation du véhicule IRIS et par présenter l'intégration de notre système de sécurité Safety-Bag dans ce véhicule autonome expérimental. Nous détaillerons la structure matérielle, l'électronique nécessaire et mettrons en avant l'approche tolérance aux fautes.

Ensuite, nous allons présenter une analyse de risques du composant Safety-Bag pour nous assurer que le composant n'ajoute pas au système des défaillances simples catastrophiques. Dans la troisième section, nous décrirons les moyens d'expérimentation utilisés, notamment le banc VILAD, les pistes d'essais ainsi que le joueur de scénarios y compris le pilote virtuel pour le contrôle latéral.

Finalement, nous présenterons les expérimentations et les résultats de validation du Safety-Bag obtenus sur le banc VILAD ainsi que les expérimentations d'interaction homme-machine sur la piste réelle Seville respectivement dans les section 4, 5 et 6.

5.1 Le véhicule expérimental IRIS

Le laboratoire Heudiasyc dispose de trois véhicules utilisés pour mener des développements de véhicules autonomes et les expérimentations associées. Deux des véhicules sont des Renault Zoé qui disposent d'une version prototype d'une solution de robotisation développée par Renault. Dans cette solution de robotisation, Renault a inséré un calculateur désigné « *Micro-Autobox* » qui permet de commander numériquement les actionneurs du véhicule, et assure une fonction assez proche du composant Safety-Bag en implémentant des vérifications du système telles que la vitesse maximale. Cependant, ce système est complexe, difficile à valider et partiellement boîte noire. Il ne nous apparaît donc pas suffisamment digne de confiance pour assurer des fonctions de tolérance aux fautes. De plus, le modifier ou y ajouter des composants demande un travail de développement conséquent. Nous avons donc choisi d'utiliser le véhicule de type Fluence IRIS du laboratoire (voir la figure 5.1) pour lequel nous avons développé une solution de robotisation que nous maîtrisons et sur lequel nous avons installé notre propre système de sécurité-innocuité Safety-Bag.

Figure 5.1 – Véhicule autonome expérimental de type Fluence IRIS

Ce véhicule est capable d'accélérer de 0 km/h à 50 km/h en 4 secondes et peut atteindre 130 km/h.

Dans cette section, nous présenterons l'architecture du véhicule IRIS et nous montrerons l'intégration et les impacts de notre système de sécurité Safety-Bag sur ce véhicule expérimental.

5.1.1 Robotisation du véhicule Fluence IRIS

Le schéma suivant 5.2 présente la robotisation de la Fluence. L'application de contrôle-commande et le Safety-Bag commandent le véhicule en envoyant des signaux analogiques aux trois actionneurs (accélérateur, frein et volant). Le bouton d'arrêt de process permet au conducteur d'activer ou d'inhiber l'ensemble des actionneurs pour repasser sur une conduite uniquement manuelle.

En commutant les trois boutons (*switch frein*, *switch direction* et *switch acc*), l'opérateur ou le conducteur peut activer ou désactiver la commande autonome de seulement certains actionneurs. Nous pouvons ainsi lors des expérimentations contrôler manuellement le volant pendant que l'application de contrôle-commande contrôle le frein et l'accélérateur ou inversement.

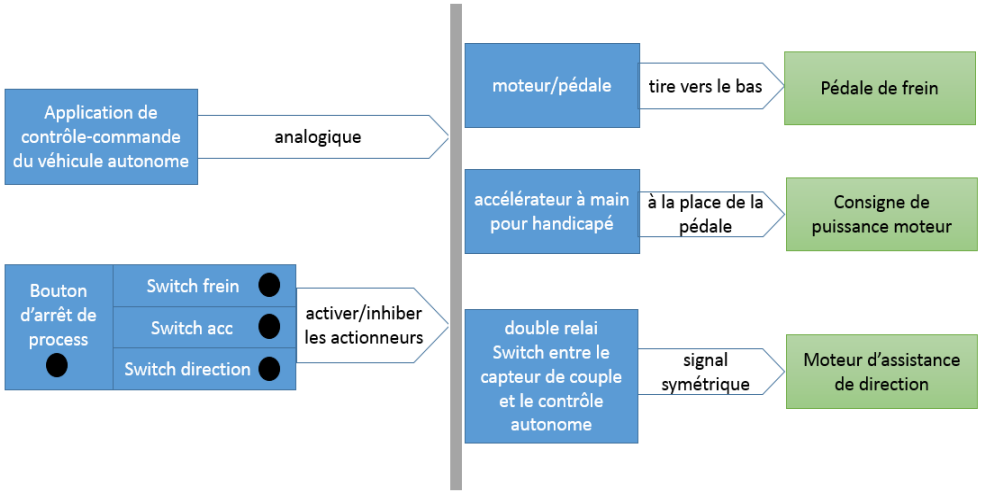

Figure 5.2 – Robotisation de la Fluence-IRIS

Les actionneurs sont commandés comme suit :

- Le frein est actionné par un moteur qui tire sur la pédale du frein. La puissance de freinage maximale est atteinte en fournissant 7 Volts au moteur ce qui correspond, sur le banc, à une décélération entre 3 et $3.5 m.s^{-2}$. Cette décélération est relativement faible à cause de la limite de puissance du moteur et également du réglage du banc qui simule une inertie un peu trop importante du véhicule. Lorsqu'on freine manuellement (en appuyant sur la pédale avec le pied), on atteint une décélération d'environ $5 m.s^{-2}$, ce qui est nettement moins performant que le freinage réel du véhicule.

- La consigne d'accélération est commandée via un accélérateur à main pour handicapé. Les valeurs de la consigne d'accélération sont dans l'intervalle [0.7 ; 1.7 Volts]. La valeur de 1.7 Volts correspond à l'accélération maximale du véhicule.

- La direction est commandée en remplaçant le signal du capteur de couple de la direction assistée électrique. Pour des raisons de sécurité, deux signaux symétriques sont fournis par le capteur de couple : ils sont centrés sur 2.5 Volts et selon une amplitude maximale de ± 0.45 Volts. Ainsi, pour fournir une consigne de 0.1 Volt, il faut produire deux signaux symétriques respectivement de 2.6 et 2.4 Volts. A l'opposé, pour fournir une consigne de -0.1 Volts, il faut produire des signaux de 2.4 et de 2.6 Volts.

  L'inconvénient de cette solution est que le calculateur de la direction assistée électrique interprète cette consigne en tenant compte de la vitesse, de l'angle volant, mais aussi d'autres critères et nous ne connaissons pas exactement la fonction qu'elle applique pour obtenir le couple appliqué sur la direction. Nous avons cependant constaté que plus la vitesse est élevée et plus l'angle volant est important, moins le couple appliqué est important.

5.1.2 Safety-Bag sur le véhicule autonome expérimental IRIS

Dans cette partie, nous présentons l'intégration du dispositif de sécurité-innocuité Safety-Bag dans le véhicule autonome expérimental IRIS. Nous allons commencer par présenter l'architecture du véhicule autonome avec Safety-Bag. Ensuite, nous allons décrire les composants constituant le système Safety-Bag. Nous rappellerons finalement comment la tolérance aux fautes est permise par le Safety-Bag.

5.1.2.1 Architecture du véhicule autonome en intégrant le Safety-Bag

Dans l'architecture du système avec Safety-Bag, le Safety-Bag s'insère entre l'application de contrôle-commande et les actionneurs du véhicule. Notre Safety-Bag est intégré à l'architecture à trois niveaux, précisément entre la couche réactive de plus bas niveau et l'application de contrôle-commande, pour intercepter les commandes envoyées aux actionneurs, et vérifier leurs cohérences avec des nécessités de sécurité. Si les conditions de déclenchement de ces nécessités sont enfreintes, il peut générer des alertes, refuser les commandes envoyées, forcer des actions de sécurité ou rendre la main à l'opérateur.

Le Safety-Bag est constitué de deux calculateurs qui se surveillent l'un l'autre par l'échange d'un signal carré (*heartbeat*). Chacun peut également activer les alarmes physiques et connaître l'état du bouton d'arrêt de process.

Le MOSFET est commandé par le Safety-Bag Supervisor pour activer/désactiver le mode autonome du véhicule en servant d'interrupteur entre l'application de contrôle-commande et les actionneurs. Ce MOSFET est en série avec le bouton d'arrêt de process.

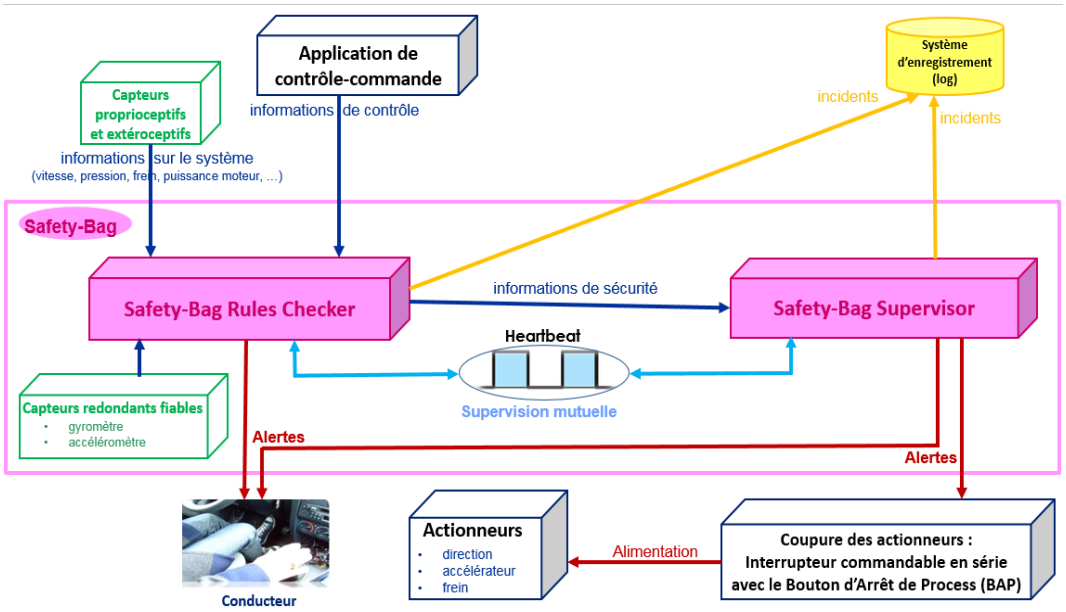

Figure 5.3 – Architecture du système étudié avec Safety-Bag

Le système de *log* de sécurité regroupe tous les changement d'état des deux calculateurs Safety-Bag Supervisor et Safety-Bag Rules Checker. Il comprend ainsi les démarrages du Safety-Bag, les basculements du bouton d'arrêt de process et toutes les détections de défaillances dans le système. Des capteurs (un gyromètre et un accéléromètre) utilisés seulement par le Safety-Bag lui fournissent des informations indépendantes par rapport au système de conduite (dont l'application de contrôle-commande). D'autres capteurs pourraient être nécessaires pour vérifier la cohérence d'autres informations comme par exemple la vitesse longitudinale.

Ces informations permettent de détecter des décisions erronées prises par l'application de contrôle-commande notamment si ces erreurs sont la conséquence de défaillance des capteurs utilisés par l'application de contrôle-commande. Les capteurs redondants du Safety-Bag devront être fiables et leur défaillance peut être détectée par comparaison avec les capteurs du système ou incohérence avec les informations fournies par l'application de contrôle-commande. En cas de défaillance détectée par le Safety-Bag, les alarmes sont levées et le conducteur intervient.

La figure 5.3 montre l'architecture des véhicules autonomes avec le système de sécurité-innocuité Safety-Bag.

5.1.2.2 Description des sous-composants du Safety-Bag

Le Safety-Bag s'assure que les commandes produites par l'application de contrôle-commande respectent des nécessités de sécurité avant de les transmettre aux actionneurs du véhicule. Pour réduire ses risques de défaillances, il contient deux

calculateurs diversifiés qui se surveillent l'un l'autre : le Safety-Bag Rules Checker et le Safety-Bag Supervisor. Le Safety-Bag est ainsi implanté physiquement sur deux calculateurs de technologies matérielles différentes et qui remplissent des fonctions complémentaires chacun comprenant un composant logiciel de diagnostic et un calculateur.

- Le Safety-Bag Rules Checker : est constitué d'un calculateur *CUBE*, basé sur un PowerPC sous le système d'exploitation Linux/Xenomai. Sur ce calculateur une gamme de cartes d'acquisition (carte *AO308* et carte *AI217*) et de sorties analogiques ou digitales est utilisée pour relier le Safety-Bag aux autres composants (capteurs, actionneurs, application de contrôle-commande).

- Le Safety-Bag Supervisor : est constitué d'un calculateur *IGEP*, calculateur basé sur un processeur ARM sous le système d'exploitation Linux/Xenomai. Ses possibilités d'entrées/sorties sont plus limitées que pour le Safety-Bag Rules Checker. Nous utilisons les GPIOs (General Purpose Input/Output) pour générer un heartbeat, recevoir celui du Safety-Bag Rules Checker, vérifier la présence de tensions et contrôler les alarmes visuelles et sonores.

L'électronique du composant Safety-Bag est répartie sur 2 cartes, une regroupant tous les composants de puissance et une carte principale regroupant tous les signaux faibles. Dans la carte de puissance, nous trouvons :

- Le connecteur du bouton d'arrêt de process,

- Le MOSFET (mentionné dans la sous-section précédente),

- les alarmes puissantes : un buzzer (alarme sonore) relativement bruyant et une led rouge de haute luminosité (alarme visuelle) pour alerter le conducteur,

- 2 convertisseurs DC/DC : un en 12 Volts pour fournir une alimentation stable aux composants de la carte principale et un en 5 Volts pour alimenter le calculateur IGEP du Safety-Bag Supervisor.

Dans la carte principale, nous trouvons :

- L'ensemble des connecteurs d'entrée et de sortie vers le calculateur CUBE du Safety-Bag Rules Checker et vers les capteurs et les actionneurs du véhicule. Un convertisseur digital/analogique (D/A) relie les sorties de la carte AO308 du calculateur CUBE du Safety-Bag Rules Checker vers les actionneurs du véhicule. De plus, un convertisseur analogique/digital (A/D) relie les capteurs du véhicule (en particulier les capteurs proprioceptifs) et les capteurs propres au Safety-Bag vers la carte d'acquisition AI217 du calculateur CUBE.

Une carte CAN est connectée au bus CAN du véhicule fournissant les valeurs de vitesse, d'accélération, d'accélération latérale, de vitesse de rotation, la position et la vitesse du volant, etc. Pour faire des expériences sur le banc VILAD, nous n'utilisons que la vitesse longitudinale donnée par l'odométrie, la position simulée par le simulateur intégré au banc VILAD et la vitesse du volant du fait que le véhicule ne bouge pas réellement.

- Un circuit avec 4 portes logiques *NAND*, qui assure l'union des alarmes émises par le calculateur CUBE du Safety-Bag Rules Checker et par le calculateur IGEP du Safety-Bag Supervisor. Ce composant prend en entrées les signaux d'alarmes produits par l'IGEP ou par le CUBE et active tous les dispositifs physiques d'alerte (led, buzzer, etc.) quand une alerte est levée par l'un des deux calculateurs.

- Le relais DAE, qui permet de commander le volant en dirigeant vers le calculateur de la direction assistée électrique les signaux émis par le calculateur CUBE du Safety-Bag Rules Checker.

5.1.2.3 Tolérance aux fautes internes du composant Safety-Bag et tolérance aux fautes apportée par le système Safety-Bag

Tout mécanisme de tolérance aux fautes nécessite d'une façon ou d'une autre de la redondance. La tolérance aux fautes interne à notre composant Safety-Bag ne déroge pas à cette règle. En effet, le Safety-Bag est composé comme nous l'avons évoqué dans la section 5.1.2.1 de deux calculateurs redondants qui se surveillent l'un l'autre par l'échange d'un signal carré. Chacun de ces calculateurs peut également activer les alarmes physiques. Le Safety-Bag dispose ainsi d'un système d'alerte intégré et redondant. Si le Safety-Bag Rules Checker est défaillant, le Safety-Bag Supervisor déclenche les alarmes sonores et visuelles et coupe la transmission des consignes aux actionneurs. Au contraire, si le Safety-Bag Supervisor est défaillant, le Safety-Bag Rules Checker active les alarmes visuelles mais les consignes continuent à être appliquées. En effet, il n'a y aucun indice de défaillance de l'application de contrôle-commande. Pour la sécurité du véhicule, il est préférable de laisser à l'application de contrôle-commande le contrôle du véhicule jusqu'au passage en conduite manuelle à l'initiative du conducteur.

De plus, le Safety-Bag utilise des capteurs supplémentaires et redondants par rapport aux capteurs de l'application de contrôle-commande.

En ce qui concerne la tolérance aux fautes que le Safety-Bag apporte au système, sa détection d'erreurs est basée sur :

- La duplication et comparaison : Le Safety-Bag est capable de comparer des informations fournies par des capteurs redondants vis-à-vis des fautes à tolérer et fournissant presque le même service. Ceci peut permettre au Safety-Bag de faire la comparaison de ces informations, de détecter les fautes physiques et d'intervenir en cas d'incohérence en faisant le rétablissement du système.

- Le contrôle temporel : Le Safety-Bag peut détecter des erreurs temporelles dans le système. Par exemple, le Safety-bag peut vérifier la vivacité et la cohérence temporelle de l'application de contrôle-commande et d'autres informations fournies par celle-ci telles que les horodatages de l'état cinématique estimé par le véhicule, de sa trajectoire cinématique ainsi que de son espace navigable.

- le contrôle de vraisemblance : Le Safety-Bag peut détecter des erreurs de vraisemblance dans le système par exemple en vérifiant la cohérence de certaines données telles que la limite de vitesse dangereuse, la limite d'accélération latérale, la différence entre la position du système et celle connue précédemment, etc.

En cas de détection d'erreur, le Safety-Bag peut intervenir en faisant un rétablissement du système. Ce rétablissement est réalisé par des interventions de sécurité mises en place par le Safety-Bag Rules Checker. Le Safety-Bag transforme ainsi l'état erroné détecté en un état jugé exempt d'erreurs et de fautes. Ce fonctionnement est celui d'un « *traitement d'erreur par poursuite* » : en détectant certaines défaillances dangereuses, le Safety-Bag peut intervenir en arrêtant le véhicule et en redonnant la main au conducteur. Par exemple, si le Safety-Bag détecte un dépassement d'une vitesse limite (50 km/h en ville), il intervient en freinant le véhicule et en repassant en conduite manuelle.

Comme nous l'avons dit précédemment, le but du Safety-Bag est d'améliorer la sécurité-innocuité du système, ce qui peut avoir un impact négatif sur sa disponibilité. Des compromis doivent être trouvés par les experts sécurité lors de la définition des conditions de déclenchement de sécurité pour permettre une tolérance suffisante aux fautes du système tout en soulevant aussi peu de fausses alertes que possible.

La bonne définition des nécessités de sécurité et le bon fonctionnement du Safety-Bag sont validés dans la suite de ce chapitre par des expériences d'injection de fautes. L'objectif de ces expériences est de comparer le comportement nominal du système (un scénario de conduite normale de notre véhicule autonome expérimental sans

injection des fautes) avec son comportement en introduisant des fautes injectées pendant l'exécution en ligne (par exemple, en bloquant l'application de contrôle-commande ou en dépassant une vitesse limite en ville), à la fois avec et sans le composant Safety-Bag, afin de valider le Safety-Bag et ses nécessités de sécurité, et d'identifier l'amélioration de sécurité-innocuité qu'il apporte au système.

5.2 Analyse de risques pour les véhicules autonomes avec Safety-Bag

Dans cette section, nous présentons une analyse AMDEC de notre composant Safety-Bag et nous étudions les nécessités de sécurité identifiées. Nous présentons ensuite un exemple d'arbre de défaillances montrant les limites bien connues de ce type d'analyses. Nous finissons cette section par une analyse AMDEC du véhicule autonome avec le composant Safety-Bag, montrant qualitativement l'apport de ce composant pour la sécurité-innocuité du système.

5.2.1 Analyse de sécurité AMDEC du composant Safety-Bag

Le composant Safety-Bag, constitué de composants matériels et logiciels supplémentaires par rapport au système original, apporte inévitablement de nouvelles possibilités de fautes dans le système. Il est donc nécessaire d'étudier ces fautes pour s'assurer qu'elles ne génèrent pas à leur tour de risques de défaillances inacceptables. Cela conduit à créer des nouvelles exigences de sécurité, qui doivent être garanties et vérifiées par le système. Dans le cas de l'étude AMDEC des composants du Safety-Bag, nous avons ajouté une colonne supplémentaire *« Détection et recouvrement »*, qui décrit comment la défaillance a été détectée et le moyen considéré pour la recouvrir.

Les tableaux (5.1, 5.2, 5.3, 5.4 et 5.5) représentent l'étude AMDEC concernant les éléments supplémentaires introduits pour implanter le Safety-Bag. Ces tableaux correspondent à 14 éléments et à leurs types de défaillances :

- Les composants matériels du Safety-Bag Rules Checker :
  1. Les convertisseurs numériques/analogiques (D/A) de la carte électronique AO308 du Safety-Bag Rules Checker peuvent être défaillants pour plusieurs raisons :
     - ◇ La carte AO308 ou le convertisseur d'émission du signal carré (heartbeat) sont bloqués. Dans ce cas, les sorties ne sont plus mises à jour et le signal

carré échangé par les deux calculateurs du Safety-Bag n'est plus généré. Le Safety-Bag Supervisor détecte alors l'absence du signal carré et active des alertes. Il active également le bouton d'arrêt du process (en coupant le MOSFET) et donc repasse la conduite du véhicule en mode manuel.

- ◇ Le convertisseur des alarmes (soit l'alarme sonore ou l'alarme visuelle) est bloqué.
- ◇ Le convertisseur d'accélérateur est bloqué à zéro ou verrouillé à une valeur précise, ou le convertisseur du frein est bloqué. Dans ces deux cas, en mode autonome, nous n'avons respectivement plus d'accélération, ou plus de freinage.
- ◇ L'un des convertisseurs de la direction assistée électrique est bloqué. Dans ce cas, le véhicule ne peut plus tourner en mode autonome.

2. Les convertisseurs analogiques/numériques (A/D) de la carte électronique AI217 du Safety-Bag Rules Checker peuvent défaillir suite à un problème connectique ou si la carte elle-même est bloquée. Dans ce cas, le Safety-Bag Rules Checker ne peut pas acquérir certaines informations utiles telles que le signal carré émis par le Safety-bag Supervisor, les valeurs des capteurs redondants liés au Safety-Bag (tels que le capteur gyromètre, le capteur accéléromètre), les valeurs des capteurs liés au véhicule tels que le capteur angle volant, la tension entre le bouton d'arrêt de process et le MOSFET, la tension batterie, etc. Dans cette situation, le Safety-Bag ne peut plus détecter certaines défaillances, et le conducteur doit être prévenu pour qu'il arrête les expérimentations.
3. Le calculateur du Safety-Bag Rules Checker peut défaillir suite à un problème de connexion ou une perte de tension. Cette défaillance peut provoquer les mêmes effets sur le Safety-Bag et sur le véhicule que dans le cas d'une défaillance des convertisseurs D/A de la carte AO308 du Safety-Bag Rules Checker.

- Les composants logiciel du Safety-Bag Rules Checker :
  4. La tâche temps réel du Safety-Bag Rules Checker peut défaillir à cause d'un bug logiciel ou un blocage du cœur Xenomai. Les conséquences suite à cette défaillance sont très proches de celles présentées dans le cas de la défaillance du convertisseur D/A de la carte AO308 du Safety-Bag Rules Checker.
  5. La tâche réseau du Safety-Bag Rules Checker peut défaillir suite à un bug logiciel ou à la déconnexion du réseau. Suite à cette défaillance, les données ne sont plus mises à jour dans la mémoire du Safety-Bag Rules Checker et les

trames émises par le Safety-Bag Rules Checker ne seront plus envoyées vers le Safety-Bag Supervisor. Dans ce cas, le véhicule n'accélère plus et la consigne de couple volant sera progressivement relâchée.

- Les composants matériels du Safety-Bag Supervisor :
  6. Les composants matériels du Safety-Bag Supervisor peuvent défaillir suite au blocage du calculateur Safety-Bag Supervisor, à une perte de tension ou bien à un convertisseur DC/DC (5 Volts) défaillant. Dans ce cas, le Safety-Bag Supervisor ne peut plus générer le signal carré transmis vers le Safety-Bag Rules Checker. Ce dernier ne peut plus alors être supervisé, mais il est capable de détecter l'absence du signal carré produit par le Safety-Bag Supervisor et de déclencher l'alarme visuelle pour alerter le pilote en lui demandant de reprendre le contrôle du véhicule manuellement.

- Les composants logiciels du Safety-Bag Supervisor :
  7. La tâche temps réel du Safety-Bag Supervisor : Comme dans le cas du Safety-Bag Rules Checker, cette défaillance peut survenir suite à un bug logiciel ou à un blocage au niveau du cœur Xenomai. Dans le second cas, le Safety-Bag Rules Checker détectera l'absence de signal carré et pourra alerter l'utilisateur. Pour contrer le premier cas, le Safety-Bag et ses nécessités de sécurité sont volontairement suffisamment simples pour être analysées et vérifiées facilement par des tests, des relectures croisées, etc.
  8. La tâche réseau du Safety-Bag Supervisor : Cette défaillance peut être le résultat d'un bug logiciel du Safety-Bag, un blocage du système d'exploitation ou une déconnexion réseau. Suite à cette défaillance, le Safety-Bag Supervisor ne reçoit plus de trames. La défaillance de la tâche réseau du Safety-Bag Supervisor ne provoque pas des conséquences dangereuses sur le système, et nous n'arrêtons pas la conduite autonome. Cependant, nous alertons l'opérateur par des alarmes visuelles et sonores pour qu'il soit au courant du problème et puisse arrêter rapidement l'expérience puisque le système n'est plus supervisé par le Safety-Bag.

- Les composants de supervision :

  9. Les alarmes visuelles (*LED rouge*) : Cette défaillance peut être produite suite à la défaillance de la carte électronique ou du composant *LED* lui-même. Dans ce cas, le conducteur ne sera plus informé par les alarmes visuelles, il ne pourra alors compter que sur les alarmes sonores.
  10. Les alarmes sonores (le *buzzer*) : Cela peut être le résultat de la défaillance de la carte électronique ou du composant *buzzer* lui-même. Le conducteur ne

sera plus informé en cas d'un incident grave par les alarmes sonores. Seule l'alarme visuelle sera perceptible.

11. Défaillance simultanée des différentes alarmes : La défaillance des alarmes physiques peut être provoquée à cause de la défaillance du composant de regroupement des alarmes *NAND* ou de la carte électronique ou bien suite à la défaillance du convertisseur DC/DC (12 Volts). La défaillance des alarmes seule n'apporte pas un danger très important sur le système et ne peut pas causer sa défaillance. En revanche, la présence d'une autre défaillance dangereuse (défaillance de l'actionneur du frein par exemple) sans signal d'alarme transmis au conducteur peut provoquer des dégâts catastrophiques.

- Blocage du système en mode manuel :

  12. Le relais DAE est bloqué en position automatique. Lors de cette défaillance, le Safety-bag peut intervenir en inhibant les commandes autonomes sur les actionneurs : le conducteur conduira alors normalement en mode manuel mais ne pourra plus profiter de la direction assistée électrique.

  13. Le MOSFET peut être défaillant à cause d'un problème connectique ou d'une mauvaise soudure. Ce Mosfet peut être défaillant en deux états différents : soit *ouvert* ce qui bloque le véhicule en mode de conduite manuelle et peut conduire le conducteur à une reprise nécessaire des commandes sans alarmes, soit *fermé* ce qui bloque le véhicule en mode de conduite autonome tant que le bouton d'arrêt de process est relevé et génère des alertes pour le conducteur.

  Comme nous n'avons pas de moyens de rétablissement autre que les alertes face à cette défaillance, le bon fonctionnement du MOSFET est vérifié par le Safety-Bag avant tout passage en mode autonome.

- Alimentation du Safety-Bag :

  14. Le fusible peut éventuellement fondre. Dans ce cas, le Safety-Bag Supervisor n'est plus alimenté et le Safety-Bag Rules Checker détecte cette perte d'alimentation par l'absence du signal carré. Le véhicule passe alors automatiquement en mode de conduite manuelle mais sans les alarmes sonores et visuelles qui ne peuvent pas être activées.

Tableau 5.1 – Composants Safety-Bag (page 1)

Élément		Type de Défaillance	Effets Safety-Bag	Détection et recouvrement	Effets Véhicule	$\lambda(h^{-1})$	Détection		Action		Gravité des Conséquences		Exigence de Sécurité	Réf
							moyen	t	moyen	t	G	Commentaires		
Composants matériels du SBRC	Convertisseurs D/A de la carte AO308 du SBRC	carte bloquée ou le convertisseur d'émission du heartbeat est bloqué	• les sorties ne sont plus mises à jour • le signal carré n'est plus généré	le SBS détecte l'absence du signal carré, coupe le MOSFET et donc repasse la conduite en mode manuel et déclenche les alarmes	Le contrôle autonome du véhicule est soudainement désactivé. Les alarmes visuelles et sonores sont déclenchées.	$\sim 10^{-5}$	SBS	3ms	conducteur	$<$ 2s	3b	réduction significative de la gravité par rapport au blocage des convertisseurs D/A du calculateur de commande	Le système doit vérifier que les sorties du convertisseur ne restent pas bloquées et doit garantir le passage en mode manuel.	1c
		le convertisseur soit de l'alarme visuelle ou de l'alarme sonore est bloqué	–	–	Une alarme non commandée peut éventuellement être levée.	$\sim 10^{-7}$	conducteur	$>$ 2s	conducteur	$>$ 2s	0	La transmission redondante des alarmes assure que les alarmes levées sont transmises au conducteur malgré cette défaillance.	Le système doit vérifier que les alarmes sont transmises au conducteur. Leur fonctionnement doit être vérifiée avant chaque mission.	2c
			L'alarme ne peut pas être activée par le SBRC.	Le SBS active l'alarme de façon redondante.	–									
		le convertisseur accélérateur ou frein est bloqué	détection par la vérification de la puissance consommée par le moteur du véhicule	–	accélération inhibée + alertes visuelles et sonores	$\sim 10^{-5}$	SB	$<$ 0.5s	conducteur	$<$ 2s	2	La détection de cette défaillance nécessite un capteur d'intensité sur l'alimentation du moteur électrique du véhicule.	Le système doit détecter les incohérences entre la commande d'accélération/de freinage produite par l'AppCC et la commande d'accélération effectivement réalisée par le moteur/les freins du véhicule.	3c
			détection par la vérification de la pression actuelle dans le système de freinage hydraulique	–	alertes visuelles et sonores + pas d'accélération supplémentaire	$\sim 10^{-5}$	SB	0.1s	conducteur	$<$ 2s	3b	La redondance du système de freinage devra être repensée et renforcée pour les véhicules autonomes commerciaux.		
		l'un des convertisseurs de la DAE est bloqué	détection pour la vérification de la vitesse de volant	–	La DAE passe en défaillance + alertes sonores et visuelles	$\sim 10^{-5}$	SBRC	1s	conducteur	$<$ 2s	3b	Le pilote reprend la main sans assistance de direction.	Le système doit détecter les incohérences entre la commande de direction produite par l'AppCC et la commande d'accélération effectivement réalisée par l'assistance à la direction du véhicule.	4c

Tableau 5.2 – Composants Safety-Bag (page 2)

	Élément	Type de Défaillance	Effets Safety-Bag	Détection et recouvrement	Effets Véhicule	$\lambda(h^{-1})$	Détection		Action		Gravité des Conséquences		Exigence de Sécurité	Réf
							moyen	t	moyen	t	G	Commentaires		
Composants matériels du SBRC	Convertisseurs A/D de la carte AI217 du SBRC	carte bloquée ou problème de connectique	Le SBRC ne peut acquérir ni les valeurs des capteurs redondants, ni la tension, ni le signal carré produit par le SBS.	Il détecte les capteurs redondants qui fournissent des valeurs incohérentes. Les sorties de l'accélérateur et de couple volant ne sont appliquées et les alarmes visuelles et sonores sont déclenchées.	Le SB ne peut plus superviser les actions du véhicules et les alarmes sonores sont levées.	$\sim 10^{-5}$	SBRC	4ms	conducteur	$<$ 2s	0	• Il n'y a pas de défaillance correspondante dans le véhicule sans SB. • Ce problème ne cause pas de défaillances du système, mais toute défaillance suivante risque de ne pas être détectée et ne pourra pas être remontée au conducteur.	Le système doit diagnostiquer la défaillance de l'ensemble des entrées analogiques du SBRC.	5c
	Calculateur du SBRC	hors tension ou déconnexion	idem 1c											6c
Composants logiciels du SBRC	tâche temps réel du SBRC	bug logiciel ou Xenomai bloqué	• idem 1c • faute logicielle dans le code ou les nécessités de sécurité qui amène un mauvais comportement du SB			$\sim 10^{-6}$	idem 1c				• 3b • 0 à 5	• idem 1c • un comportement incorrect du SB et particulièrement des nécessités de sécurité peut causer une défaillance catastrophique du système si le SB envoie des commandes aberrantes sur les actionneurs.	• Le système doit garantir le passage en mode manuel en cas de défaillance de la tâche temps réel du SBRC. • Les composants logiciels du SB doivent être fortement validés : le code hors nécessité de sécurité peut être validé formellement ou validé par lecture croisée, tandis que les nécessités de sécurité doivent être relues par des experts de sécurité.	7c

Tableau 5.3 – Composants Safety-Bag (page 3)

Élément		Type de Défaillance	Effets Safety-Bag	Détection et recouvrement	Effets Véhicule	$\lambda(h^{-1})$	Détection		Action		Gravité des Conséquences		Exigence de Sécurité	Réf
							moyen	t	moyen	t	G	Commentaires		
Composants logiciels du SBRC	tâche réseau du SBRC	déconnexion réseau, bug logiciel	• Les données ne sont plus mises à jour dans la mémoire du SBRC. • Les trames ne sont plus transmises vers le SBS.	Les tâches temps réel du SBRC et du SBS détectent l'absence de nouvelles trames et passent les automates dans l'état défaillant associé.	Le véhicule n'accélère plus, puis la consigne de couple volant sera progressivement relâchée. Les alarmes visuelles et sonores s'allument et sonnent.	$\sim 10^{-4}$	SB/tâche temps réel du SBRC	< 0.5ms	conducteur	< 2s	3a	Les commandes peuvent n'être relâchées que progressivement, la reprise en manuel est donc assez aisée.	Le système doit détecter la perte de la communication avec l'AppCC et repasser la conduite en mode manuel.	8c
Composants matériels du SBS	–	calculateur SBS gelé ou hors tension ou convertisseur DC/DC 5V défaillant	Le signal carré n'est plus produit par le SBS et le SBRC n'est plus supervisé.	Le SBRC détecte l'absence de l'heartbeat produit par le SBS. Il lève l'alarme visuelle pour demander au pilote de reprendre le contrôle en manuel.	L'alarme visuelle s'allume. Plusieurs options sont possibles	$\sim 10^{-5}$	SB/tâche temps réel du SBRC	< 5ms	conducteur	2s	0	La situation n'est pas dangereuse sauf en cas de défaillance « simultanée » du SBRC, mais l'expérimentation doit être interrompue pour régler le problème du SBS.	Le système doit tolérer l'arrêt inopiné du du SBS.	9c
Composants logiciels du SBS	tâche temps réel du SBS	bug logiciel ou Xenomain bloqué	idem 9c			$\sim 10^{-6}$	idem 9c							10c
	tâche réseau du SBS	déconnexion réseau, bug logiciel SB ou LINUX défaillant	Le SBS ne reçoit plus de trames.	La tâche temps réel du SBS détecte la perte des trames.	L'alarme s'allume, la conduite autonome n'est pas interrompue.	$\sim 10^{-4}$	SB/tâche temps réel du SBS	< 5ms	conducteur	2s	0	La situation n'est pas dangereuse, mais l'expérimentation doit être interrompue pour régler le problème du SBS.	Le système doit tolérer la perte de communication du SBS.	11c

Tableau 5.4 – Composants Safety-Bag (page 4)

Élément		Type de Défaillance	Effets Safety-Bag	Détection et recouvrement	Effets Véhicule	$\lambda(h^{-1})$	Détection		Action		Gravité des Conséquences		Exigence de Sécurité	Réf
							moyen	t	moyen	t	G	Commentaires		
Supervision	alarmes	défaillance du composant NAND ou de la carte électronique / défaillance du convertisseur DC/DC 12V	pas de signal d'alarme en cas de défaillance	_	pas de signal en cas de défaillance	$\sim 10^{-8}$	conducteur	$> 2s$	conducteur	$> 2s$	0	Panne non active : en l'absence de signal, le conducteur mettra beaucoup de temps pour réagir. La gravité des divers incidents augmente.	• Les alarmes doivent être transmises au conducteur. • Exigence exportée : leur fonctionnement doit être vérifié avant chaque mission.	12c
	alarmes visuelles (led rouge)	carte électronique ou LED	pas de signal visuel en cas de défaillance	_	pas d'alarme sur les incidents les moins graves	$\sim 10^{-8}$	conducteur	$> 2s$	conducteur	$> 2s$	0	Défaillance non active : la gravité des divers incidents augmente, mais ne concerne pas les situations les plus graves.	idem 12c	13c
	alarmes sonores (buzzer)	carte électronique ou buzzer	pas de signal sonore en cas de défaillance	_	seule l'alarme visuelle est perceptible sur les incidents graves.	$\sim 10^{-8}$	conducteur	$> 2s$	conducteur	$> 2s$	0	Défaillance non active : le pilote peut être surpris que la situation soit pire que celle indiquée par l'alarme. Cependant, il est prévenue et informé par l'alarme visuelle et doit réagir.	idem 12c	14c

Tableau 5.5 – Composants Safety-Bag (page 5)

Élément		Type de Défaillance	Effets Safety-Bag	Détection et recouvrement	Effets Véhicule	$\lambda(h^{-1})$	Détection		Action		Gravité des Conséquences		Exigence de Sécurité	Réf
							moyen	t	moyen	t	G	Commentaires		
Passage forcé du mode automatique en mode manuel	relais DAE	le relais « colle » en position automatique	_	Le Safety-Bag mettra l'actionneur à 0 s'il n'y a plus de commande cohérente.	Le conducteur a perdu la DAE.	$\sim 10^{-5}$	opérateur (panne non active)	_	_	_	5	défaillance inadmissible, BAP enfoncé, le SBRC doit recopier la valeur du capteur de couple volant en sortie.	Il ne faut pas que l'actionneur de direction (DAE) du véhicule reçoive des consignes autonomes après le passage en mode manuel.	15c
	Mosfet physique	connectique ou soudure	Ouvert : pas de fonctionnement en mode autonome	Le SB ne sort pas de la procédure « Init ».	Ouvert : les actionneurs sont désactivés sans alertes.	$\sim< 10^{-8}$	conducteur	$> 2s$	conducteur	$> 2s$	4	Ouvert : La conduite autonome est brutalement désactivée sans alertes.	Il est toujours possible de reprendre le contrôle en manuel. Exigence exportée : le bon fonctionnement du Mosfet doit être vérifié avant chaque mission.	16c
			Fermé : le SBS ne peut plus couper les actionneurs en cas de défaillance du SBRC.		Fermé : le SBS ne peut plus couper les actionneurs en cas de défaillance du SBRC.						0	Fermé : la supervision du SBRC par le SBS n'est alors plus efficace et une défaillance catastrophique du SBRC peut mener à un évènement de gravité 5.		
Alimentation du SB	fusible	fusible fond	Le SBS n'est plus alimenté. Le SBRC voit des tensions très basses et ne voit plus le signal carré du SBS.	Le SBRC détecte la perte d'alimentation et l'absence du signal carré.	Les alertes ne sont ni visibles, ni audibles. Le véhicule passe en mode manuel.	_	conducteur	$> 2s$	conducteur	$> 2s$	4	Le recouvrement en manuel sans alertes peut être périlleux.	Le fusible doit être amplement dimensionné de façon à ce que son taux de défaillance soit inférieur à 10^{-8}.	17c

Cette AMDEC nous permet d'identifier les composants critiques du Safety-Bag tels que le Mosfet physique (Réf. 16c dans le tableau 5.5) et le fusible (Réf. 17c dans le tableau 5.5). Pour ces deux défaillances, le niveau de gravité est 4, puisque les fonctionnalités autonomes s'arrêtent sans que les alertes ne soient déclenchées, mais la défaillance reste rattrapable par le conducteur. En s'assurant que ces défaillances sont suffisamment rares (en utilisant un Mosfet avec un faible taux de défaillance, et en dimensionnant correctement le fusible), les risques associés peuvent être considérés comme acceptables, surtout en considérant la réduction du risque qu'apporte le Safety-Bag sur les défaillances des autres composants du véhicule. Cependant, le niveau de gravité de la défaillance du relais DAE du Safety-Bag (Réf. 15c dans le tableau 5.5) est toujours 5. En effet, cette défaillance reste irrattrapable par le conducteur qui peut ne pas avoir le temps de réagir avant qu'une conséquence catastrophique ne se produise.

L'AMDEC des composants du Safety-Bag nous a permis également d'identifier plusieurs composants dont les fautes ont des conséquences dont la gravité est nulle. Ce niveau de gravité correspond au fonctionnement nominal de notre véhicule nominal expérimental dans lequel le logiciel est capable de conduire correctement le véhicule tandis que le matériel applique correctement les décisions prises par le logiciel. Cependant, des fonctionnalités du Safety-Bag (comme les alarmes) ne fonctionnent pas, restant dans un état d'erreur latente tant qu'une autre erreur du système traitée par le Safety-Bag ne demande pas de les déclencher. En particulier :

- Si l'un des deux convertisseurs de l'alarme visuelle ou de l'alarme sonore de la carte AO308 du Safety-Bag Rules Checker est bloqué, la transmission redondante des alarmes permettra d'assurer que les alarmes levées sont transmises au conducteur malgré cette défaillance.

- Si une ou toutes les alarmes sont défaillantes, il s'agit dans ce cas d'une erreur latente, c'est-à-dire présente dans le système mais qui n'a pas de conséquences sur le système tant qu'elle ne devient pas active. Ce système d'alarme défaillant reste dans un état d'erreur latent tant qu'il n'est pas testé ou sollicité. Il n'y a pas de conséquences tant qu'une autre défaillance ne nécessite pas de déclencher cette alarme. Pour cela, nous devons tester les alarmes avant chaque mission.

Les valeurs de taux de défaillance $\lambda(h^{-1})$ restent toujours difficiles à obtenir sans effectuer de nombreuses expérimentations. Les valeurs que nous avons mentionnées dans les tableaux (5.1, 5.2, 5.3, 5.4 et 5.5) sont des valeurs empiriques. Pour certains composants comme le calculateur du Safety-Bag Rules Checker et ses deux cartes AO803 et AI217, leur MTBF (*Mean Time Between Failures*) est de l'ordre de 100

milles heures, et le taux de défaillance dans ce cas est donc $10^{-5}(h^{-1})$. La défaillance du Mosfet ou des alarmes physique reste peu probable, de l'ordre de 10^{-8} par heure. Un système d'alerte fiable peut permettre éventuellement au conducteur d'être informé plus rapidement de la défaillance du système et donc de passer en mode manuel.

Comme pour le cas des composants logiciels du véhicule IRIS, les composants logiciels du Safety-Bag sont développés dans un contexte de recherche et ne sont pas testés aussi intensivement que des composants industrialisés avant d'être intégrés dans le système complet. Nous avons donné un taux de défaillance de l'ordre d'environ 10^{-6} par heure pour la tâche temps réel de chacun des deux calculateurs Safety-Bag Rules Checker et Safety-Bag Supervisor ou 10^{-4} par heure pour la tâche réseau de chacun de ces deux calculateurs.

A partir de l'AMDEC, nous avons réussi à identifier les 12 exigences de sécurité suivantes parmi les 17 types de défaillances définies :

- Exigence de sécurité 1 : En considérant la défaillance de blocage de la carte AO308 ou du convertisseur d'émission du signal carré (Réf. 1c dans 5.1) ou la défaillance du calculateur du Safety-Bag Rules Checker (Réf. 6c dans 5.2), *le système doit vérifier que les sorties du convertisseur ne sont pas restées bloquées et doit garantir le passage en mode manuel.*

  Cette exigence de sécurité est implémentable par le Safety-Bag. En effet, s'il n'y a plus l'échange d'un heartbeat émis par le Safety-Bag Rules Checker, le Safety-Bag Supervisor peut déclencher des alarmes pour prévenir le conducteur et déconnecter les actionneurs afin de mettre le véhicule en mode manuel.

- Exigence de sécurité 2 : En considérant la défaillance des convertisseurs des alarmes (alarme visuelle ou alarme sonore) (Réf. 2c dans 5.1), *le système doit vérifier que les alarmes sont transmises au conducteur.*

  Cette exigence de sécurité est implémentable par le Safety-Bag soit en relisant les sorties des convertisseurs, soit en disposant d'un capteur (un micro et une photo diode) pour contrôler les émissions physiques des signaux.

  Pour des raisons de temps et de complexité, nous n'avons pas implémenté ces vérifications. Cependant, nous avons spécifié la vérification du bon fonctionnement des alarmes comme faisant partie du processus à réaliser avant toute expérimentation.

- Exigence de sécurité 3 : En considérant la défaillance de blocage de la carte AI217 (Réf. 5c dans 5.2), *le système doit vérifier que l'ensemble des entrées analogiques du Safety-Bag Rules Checker sont mises à jour correctement.*

Cette exigence de sécurité est implémentable par le Safety-Bag. Ce dernier doit vérifier l'heartbeat émis par le Safety-Bag Supervisor vers le Safety-Bag Rules Checker. De plus, le Safety-bag doit vérifier que les signaux émis par les capteurs redondants (gyromètre et accéléromètre) sont dans l'intervalle de valeur de ces capteurs. Dans le cas contraire, le Safety-bag doit déclencher les alarmes.

- Exigence de sécurité 4 : En considérant la défaillance de la tâche temps réel du Safety-Bag Rules Checker (Réf. 7c dans 5.2), *le système doit garantir le passage en mode manuel.*

  Cette exigence de sécurité est un sous-ensemble de l'exigence de sécurité N° 1, implémentée par le Safety-Bag.

- Exigence de sécurité 5 : En considérant la défaillance de la tâche temps réel du Safety-Bag Rules Checker (Réf. 7c dans 5.2), nous avons pu identifier une deuxième exigence de sécurité : *Les composants logiciels du Safety-Bag doivent être fortement validés : le code (sans les nécessités de sécurité) peut être validé formellement ou validé par lecture croisée, tandis que les nécessités de sécurité doivent être relues par des experts de sécurité ou par des techniques formelles telles que la méthode SMOF [Machin et al., 2016].*

  Cette exigence de sécurité n'est pas implémentable par le Safety-Bag.

- Exigence de sécurité 6 : En considérant la défaillance de la tâche réseau du Safety-Bag Rules Checker (Réf. 8c dans 5.3), *le système doit détecter la perte de la communication avec l'application de contrôle-commande et doit repasser la conduite en mode manuel.*

  Cette exigence de sécurité est implémentable par le Safety-Bag. En effet, la tâche temps réel du calculateur Safety-Bag Rules Checker est capable de vérifier que la tâche réseau fonctionne correctement et est activée. Pour cela, la tâche temps réel du Safety-Bag Rules Checker doit déclencher les alertes et doit inhiber ou forcer des actions en cas de non fonctionnement de la tâche réseau. Cette exigence de sécurité est vérifiable par le même mécanisme que celui qui vérifier la vivacité de l'application de contrôle-commande.

- Exigence de sécurité 7 : En considérant la défaillance des composants matériels du Safety-Bag Supervisor (Réf. 9c dans 5.3) ou la défaillance de la tâche temps-réel du Safety-Bag Supervisor (Réf. 10c dans 5.2), *le système doit tolérer la défaillance matérielle du Safety-Bag Supervisor.*

Cette exigence de sécurité est implémentable par le Safety-Bag. En effet, le Safety-Bag Rules Checker peut détecter la défaillance du Safety-Bag Supervisor. Il vérifie la bonne réception du signal hearbeat émis par le Safety-Bag Superviser et déclenche des alertes quand le signal n'est plus reçu. Bien qu'il n'y ait pas de danger immédiat, le conducteur doit interrompre la conduite autonome jusqu'à correction du problème car le système n'est plus correctement supervisé par le Safety-Bag.

- Exigence de sécurité 8 : En considérant la défaillance de la tâche réseau du Safety-Bag Supervisor (Réf. 11c dans 5.3), *le système doit tolérer la perte de communication du Safety-Bag Supervisor.*

  Cette exigence de sécurité est implémentable par le Safety-Bag. La tâche temps réel du Safety-Bag Supervisor peut vérifier que la tâche réseau fonctionne et est activée. La tâche temps réel du Safety-Bag Supervisor doit vérifier ainsi la vivacité de la tâche réseau et doit déclencher les alarmes si elle est bloquée.

- Exigence de sécurité 9 : En considérant la défaillance des alarmes (Réf. 12c dans 5.4), *le système doit vérifier que les alarmes sont transmises au conducteur. Leur fonctionnement doit être vérifié avant chaque expérience.*

  Cette exigence inclut l'exigence de sécurité 2 mentionnée précédemment. Elle lui ajoute la vérification des alarmes avant chaque expérience, que nous avions déjà proposée pour l'exigence de sécurité N° 2.

- Exigence de sécurité 10 : En considérant la défaillance du relais de la direction assistée électrique (Réf. 15c dans 5.5), *le système doit vérifier que l'actionneur de direction du véhicule ne reçoit pas des commandes de l'application de contrôle-commande après le passage en mode manuel.*

  Cette exigence de sécurité est implémentable par le Safety-Bag. En effet, le Safety-Bag Rules Checker peut appliquer des tensions de sortie pour mettre la direction assistée électrique en défaut. Il suffit pour cela que les signaux émis ne soient pas symétriques. Le conducteur reprend alors la conduite, mais sans assistance de direction, ce qui n'est vraiment gênant qu'à basse vitesse notamment pour les manœuvres.

- Exigence de sécurité 11 : En considérant la défaillance du Mosfet (Réf. 16c dans 5.5), *le système doit permettre au conducteur de reprendre la conduite manuelle et le bon fonctionnement du Mosfet doit être vérifié avant chaque mission.*

Cette exigence de sécurité est implémentable par le Safety-Bag. Ce dernier peut vérifier que le MOSFET fonctionne correctement au démarrage du système et le bouton d'arrêt de process permet le passage en mode manuel puisqu'il s'agit d'un interrupteur monté en série avec le MOSFET.

- Exigence de sécurité 12 : En considérant la défaillance du fusible (coupé ou fondu) (Réf. 17c dans 5.5), *le fusible doit être amplement dimensionné de façon à ce que son taux de défaillance soit inférieur à* 10^{-8}.

  Cette exigence de sécurité n'est pas implémentable par le Safety-Bag. En effet, ce dernier ne dispose pas de capteurs propres lui permettant de détecter cette défaillance. Une vérification de dimensionnement par une maintenance préventive tous les deux ans peut être une solution pertinente pour pallier ce problème.

Certaines défaillances nécessitent des exigences de sécurité qui sont déjà identifiées et présentées dans l'étude AMDEC pour les véhicules autonomes. Nous citons par exemple :

- Exigence de sécurité 3 : En considérant la défaillance de l'un des convertisseurs accélérateur ou frein (Réf. 3c dans 5.1), *le système doit vérifier respectivement le bon fonctionnement de l'accélérateur et du frein.* Cette exigence de sécurité est la même exigence que celle identifiée pour le cas d'une défaillance d'accélérateur ou du frein dans l'étude d'analyse de risque AMDEC pour les véhicules autonomes.

- Exigence de sécurité 4 : En considérant la défaillance de l'un des deux convertisseurs de la direction assistée électrique (Réf. 4c dans 5.1), *le système doit vérifier le bon fonctionnement de l'assistance de direction.* Cette exigence de sécurité est la même que celle identifiée en étudiant l'AMDEC des véhicules autonomes pour le cas d'une défaillance de l'actionneur de direction.

5.2.2 Limites de l'AMDEC

L'AMDEC du véhicule autonome expérimental avec Safety-Bag montre une diminution significative de la gravité des incidents dans de nombreux cas.

Cependant, cette AMDEC ne considère pas les possibilités de *plusieurs défaillances d'éléments indépendants*. Pour prendre ces situations en compte, la technique des arbres de défaillances doit être utilisée pour compléter l'analyse AMDEC afin d'identifier les défaillances multiples dangereuses.

Dans cette section, nous présentons un exemple d'arbre de défaillance associé à l'évènement redouté « *défaillance matérielle du Safety-Bag* », dû à la défaillance des deux calculateurs (Safety-Bag Rules Checker et Safety-Bag Supervisor) en moins de deux cycles de surveillance mutuelle. Les deux évènements considérés (E010 et E011 dans la figure 5.4) provoquent l'événement redouté si ils ont lieu tous les deux dans l'intervalle du temps du cycle de surveillance mutuelle (soit environ 20 ms) autrement le Safety-Bag est capable de diagnostiquer la défaillance d'un des calculateurs.

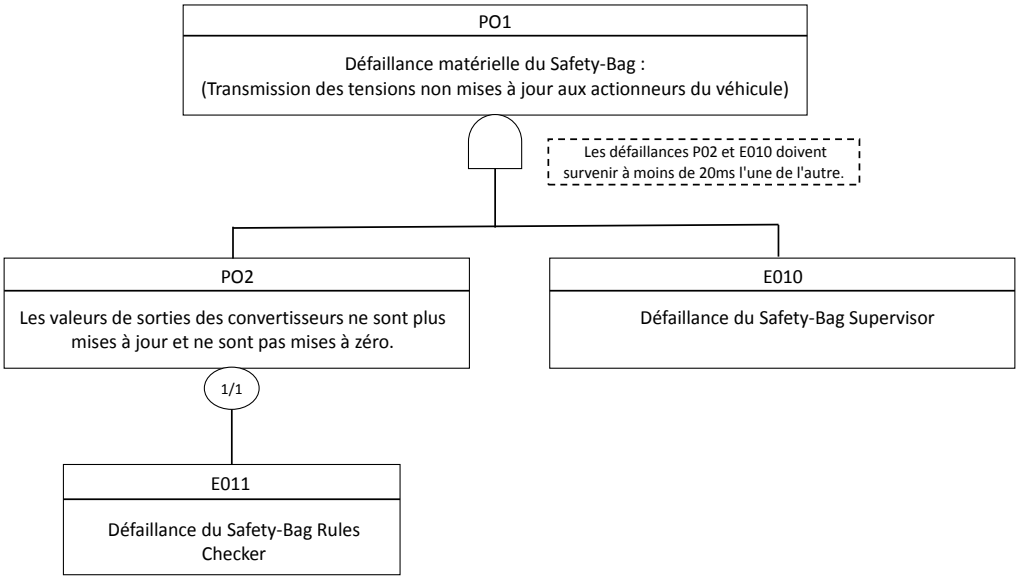

Figure 5.4 – Arbre de défaillances du Safety-Bag

Sur la figure 5.4, nous pouvons en effet voir que la défaillance de ces deux calculateurs du Safety-Bag mène à une situation de perte de contrôle. En effet, une défaillance aussi rapide du second calculateur empêcherait de diagnostiquer la défaillance du premier, provoquant une défaillance sans alerte du Safety-Bag, et des conséquences possibles de gravité maximale.

La probabilité d'une telle succession d'évènements aussi rapprochés est cependant extrêmement faible en estimant que le taux de défaillance $\lambda(h^{-1})$ du Safety-Bag Rules Checker est inférieur à $\lambda_1 = 10^{-3}\ h^{-1}$ et inférieur à $\lambda_2 = 10^{-4}\ h^{-1}$ pour le

Safety-Bag Supervisor.

Pour trouver le taux de défaillance de l'évènement redouté, nous effectuons le calcul suivant :

$\lambda_{evenement\_redoute}(h^{-1}) = \lambda_1 * \lambda_2 * T$; étant donné T : le temps de réaction du Safety-Bag que nous considérons environ 20 ms, soit environ 5.5 micro heures.

Le taux de défaillance de l'évènement redouté est ainsi environ $5.5 * 10^{-13}\ h^{-1}$.

L'analyse de risques par arbres de défaillances permet de détecter des défaillances multiples que l'AMDEC ne saurait prévoir, mais nécessite d'identifier au préalable l'évènement redouté et d'avoir une grande connaissance du système afin de pouvoir identifier les causes possibles de l'évènement redouté. Comme l'AMDEC et contrairement à l'HazOp-UML, l'arbre de défaillances ne permet pas non plus d'identifier un enchaînement d'évènements pouvant mener à une situation dangereuse, ce qui est fréquent dans un système complexe tel que notre système autonome.

5.2.3 Impacts du Safety-Bag sur l'AMDEC du véhicule

Nous présentons dans cette partie une AMDEC des véhicules expérimentaux en prenant en compte le composant Safety-Bag pour analyser qualitativement son impact sur la sécurité-innocuité du système.

Dans cette AMDEC, le tableau est complété en ajoutant les deux colonnes : *« Effets Safety-Bag »*, qui permet de montrer les effets de la défaillance sur le Safety-Bag lui même et *« Nécessité de sécurité considérée »*, qui indique quelle nécessité de sécurité sera déclenchée sur la défaillance considérée. Les nécessités de sécurité identifiées dans le chapitre précédent sont rappelées dans le tableau 5.6. En considérant la même échelle de gravité que précédemment, nous présentons la matrice AMDEC des véhicules autonomes avec Safety-Bag, de la même façon que précédemment. Le taux de défaillance et les exigences de sécurité restent évidemment les mêmes pour les deux cas des véhicules autonomes sans et avec Safety-Bag.

Les fautes identifiées dans le chapitre avec des niveaux de risques considérables ont dans ce nouveau système une gravité bien plus acceptable. Le conducteur est prévenu dans la majorité des cas par les alarmes visuelles et/ou sonores lui demandant d'intervenir rapidement, et dans de nombreux cas critiques, les commandes dangereuses ne sont pas transmises aux actionneurs.

Les tableaux (5.7, 5.8, 5.9, 5.10 et 5.11) constituent l'étude AMDEC après l'ajout d'un système de sécurité-innocuité Safety-Bag. Ils sont constitués de 11 éléments et reprennent les éléments identifiés dans 5.2.3 dans le cas du véhicule autonome sans Safety-Bag.

Réf	Nécessité de sécurité
A1	Le Safety-Bag doit vérifier la circulation des trames du bus CAN. En cas d'absence de communication pendant une durée trop importante, le Safety-Bag doit alerter le conducteur et repasser en mode manuel.
A2	Le Safety-Bag doit vérifier que les capteurs proprioceptifs longitudinaux (capteurs de vitesse et d'accélération longitudinales) fonctionnent correctement. Dans le cas contraire, le Safety-Bag déclenche les alarmes, inhibe l'accélération, freine modérément le véhicule et applique la valeur d'angle volant venant de l'application de contrôle-commande.
A3	Le Safety-Bag doit vérifier que les capteurs proprioceptifs latéraux (accélération latérale et gyromètre) fonctionnent correctement. Dans le cas contraire, le Safety-Bag déclenche les alarmes, inhibe l'accélération, freine modérément le véhicule et applique la valeur d'angle volant venant de l'application de contrôle-commande.
U1	Le Safety-Bag doit vérifier la vivacité de l'application de contrôle-commande. Si celle-ci n'envoie plus de commande, le Safety-Bag doit prévenir le conducteur et repasser en mode manuel et arrête le véhicule en allumant les feux de détresse.
U2	Le Safety-Bag doit vérifier la cohérence temporelle de l'application de contrôle-commande. En cas d'incohérence temporelle, le Safety-Bag doit assurer la mise en sécurité du système en levant les alarmes et en repassant en mode manuel.
U3	Le Safety-Bag doit vérifier les bornes de vitesse maximale du véhicule. Dans le cas contraire, le Safety-Bag déclenche les alarmes, bloque les commandes d'accélération de l'application de contrôle-commande et assure la reprise en mode manuel.
U4	Le Safey-Bag doit vérifier le contrôle dynamique du véhicule : il doit vérifier que l'accélération latérale reste inférieure à une valeur seuil. Si ces bornes ne sont pas respectées, le Safety-Bag doit déclencher les alarmes et assurer la reprise en mode manuel.
U5	Le Safety-Bag doit vérifier que l'accélérateur fonctionne correctement. Il doit vérifier que l'intensité dans le moteur électrique correspond à la valeur de la commande d'accélération fournie par l'application de contrôle-commande. Dans le cas contraire, le Safety-Bag lève une alarme, inhibe l'accélération et freine modérément.
U6	Le Safety-Bag doit vérifier que le frein fonctionne correctement. Il doit vérifier que la pression ne s'écarte pas de la valeur d'une fonction donnée de la commande du frein. Dans le cas contraire, le Safety-Bag déclenche les alarmes, empêche les commandes d'accélération et freine modérément.
U7	Le Safety-Bag doit vérifier que la direction fonctionne correctement : Il doit vérifier que les angles volant évoluent dans le sens prévu selon le couple volant appliqué. Dans le cas contraire, une alerte doit être déclenchée vers le conducteur et le Safety-Bag doit mettre le véhicule en état sûr en passant en mode de conduite manuelle.
U8	Le Safety-Bag doit vérifier que la direction fonctionne correctement : il doit vérifier que l'intensité sur le moteur de direction correspond aux consignes envoyées par l'application de contrôle-commande. En cas de non correspondance, le Safety-Bag doit alerter le conducteur et doit mettre le système en état sûr en redonnant la main au conducteur.
H1	Le Safety-Bag doit vérifier la vivacité de l'état cinématique. Si l'état cinématique n'est pas mis à jour, le Safety-Bag intervient en déclenchant les alarmes, en inhibant des commandes de l'accélération et/ou forçant une action de sécurité telle que le freinage du véhicule et le maintien de l'angle volant après un certain délai.
H2	Le Safety-Bag doit vérifier la cohérence de l'évolution de l'état cinématique. En cas d'incohérence de l'état cinématique, le Safety-Bag doit lever les alarmes, inhiber l'accélération et freiner modérément le véhicule. Il doit également maintenir l'angle volant.
H3	Le Safety-Bag doit vérifier la cohérence temporelle de l'état cinématique. Si l'état cinématique n'est pas temporellement cohérent, le Safety-Bag doit lever les alarmes, inhiber l'accélération, freiner modérément le véhicule et maintenir l'angle volant.
H4	Le Safety-Bag doit vérifier la vivacité de la trajectoire cinématique. Dans le cas contraire, le Safety-Bag doit déclencher les alarmes, inhiber l'accélération, freiner et maintenir la position d'angle volant après un certain temps.
H5	Le Safety-Bag doit vérifier la vivacité de l'espace navigable. Dans le cas contraire, le Safety-Bag doit déclencher les alarmes, inhiber l'accélération, freiner et maintenir la position d'angle volant après un certain temps.
H6	Le Safety-Bag doit vérifier que l'horodatage de la trajectoire cinématique est plus récent que l'horodatage de l'espace navigable. Dans le cas contraire, le Safety-Bag déclenche les alarmes, inhibe l'accélération et force des actions de sécurité telles que le freinage et le maintien d'angle volant.

Tableau 5.6 – Rappel des nécessités de sécurité identifiées dans le chapitre précédent.

Tableau 5.7 – AMDEC des composants véhicules autonomes avec l'intervention du système Safety-Bag (page 1)

Élément	**Type de Défaillance**	**Effets SB**	**Nécessité de sécurité considérée**	**Effets Véhicule**	$\lambda(h^{-1})$	**Détection**		**Action**		**Gravité des Conséquences**		**Exigence de Sécurité**	**Réf**
						moyen	t	moyen	t	G	Commentaires		
Matériel de l'AppCC	Panne bloquée	communication perdue	U1	Pas de contrôle autonome : • pas d'accélération • pas de direction • un freinage modéré est appliqué Le pilote reprend le contrôle total du véhicule.	$\sim 10^{-3}$	SB	4 ms	conducteur	< 2s	3a	le Safety-Bag limite la gravité et déclenche les alarmes	Le système doit être capable de détecter les défaillances de gel ou d'arrêt inopiné de l'application de contrôle-commande et de mettre le système en état sûr au besoin.	1b
	Hors Tension	idem 1b	U1	idem 1b	$\sim 10^{-5}$	SB	4 ms	conducteur	+2s	3a	idem 1b	idem 1b	2b
	Clavier/ écran	–	–	Le véhicule est contrôlé par l'application, mais l'opérateur ne peut pas interagir avec l'application.	$\sim 10^{-5}$	opérateur	> 4s	conducteur	+2s	1	Un BAP associé au Safety-Bag fournit le moyen exigé dans ce cas.	Le conducteur doit pouvoir reprendre la conduite en manuel par un autre moyen que les interfaces de l'application de contrôle-commande.	3b
Logiciel de l'AppCC	blocage du logiciel ou surcharge des calculateurs	perturbation de la communication	U1	idem 1b	$\sim 10^{-3}$	SB	4 ms	conducteur	2s	3a	idem 1b	idem 1b	4b
	erreur fonctionnelle	L'application demande au Safety-Bag d'appliquer des consignes inappropriées.	U2/U3/U4	trajectoire incorrecte mais le SB peut éviter la perte de contrôle du véhicule.	$\sim 10^{-3}$	SB + conducteur	> 2s	conducteur	+2s	2 à 5	Le SB tolère plusieurs erreurs logicielles, causant notamment des vitesses trop élevées ou une dynamique dangereuse en virage, mais aucune fonctionnalité d'évitement d'obstacles ou de maintien dans l'espace navigable n'est à l'heure actuelle traitée par le SB.	Le système doit détecter les erreurs fonctionnelles, telles que la désynchronisation des données, des commandes aberrantes, des erreurs de décision, et des mauvaises interprétations. En cas d'erreur, il faut assurer la mise en sécurité du système en levant des alarmes et en repassant en mode manuel.	5b

Tableau 5.8 – AMDEC des composants véhicules autonomes avec l'intervention du système Safety-Bag (page 2)

Élément	**Type de Défaillance**	**Effets SB**	**Nécessité de sécurité considérée**	**Effets Véhicule**	$\lambda(h^{-1})$	**Détection**		**Action**		**Gravité des Conséquences**		**Exigence de Sécurité**	**Réf**
						moyen	t	moyen	t	G	Commentaires		
CAN	déconnexion ou fil coupé	–	A1	• plusieurs alertes • risque de perturbation du contrôle de la direction	$\sim 10^{-9}$	système embarqué du véhicule	< 1s	conducteur	2 à 4s	3b	le contrôle automatique est soudainement désactivé et les alertes sont envoyées au pilote mais, lorsque le pilote reprend le contrôle, le freinage et la direction assistée sont peut être désactivés. Le SB est capable de relayer les alertes de l'application au conducteur.	Le système doit détecter et traiter les défaillances de communications sur le bus CAN : arrêt des communications, pertes de données, altération des données, ou désynchronisation.	6b
	altération des données transmises	–	–	• valeur aberrante possible pour les données d'angle volant et de la vitesse et d'accélération latérales et longitudinales • de ce fait, les commandes générées à partir de ces données par l'AppCC peuvent également être aberrantes.	$\sim 10^{-10}$	conducteur	4s	conducteur	2 à 4s	5	–	Des vérifications de cohérences entre les données CAN et les données de capteurs redondants du Safety-Bag pourraient être réalisées.	7b
	CAN surchargé	–	A1	idem 6b									8b
convertisseur D/A	carte bloquée	cet élément est remplacé par les convertisseurs intégrés au Safety-Bag (voir tableau 5.1)											9b
	défaillance logicielle du pilote de la carte	cet élément est remplacé par les convertisseurs intégrés au Safety-Bag (voir tableau 5.1)											10b

Tableau 5.9 – AMDEC des composants véhicules autonomes avec l'intervention du système Safety-Bag (page 3)

Élément	**Type de Défaillance**	**Effets SB**	**Nécessité de sécurité considérée**	**Effets Véhicule**	$\lambda(h^{-1})$	**Détection**		**Action**		**Gravité des Conséquences**		**Exigence de Sécurité**	**Réf**
						moyen	t	moyen	t	G	Commentaires		
capteur d'angle volant	défaillance du capteur	diagnostic par Safety-Bag en utilisant ses capteurs redondants	A2	• accélération inhibée par le SB • freinage modéré appliqué par le SB • pas de couple appliqué à la direction	$\sim 10^{-5}$	SB	0.1s à 1s	conducteur + freinage par le Safety-Bag	2s	3b	Le Safety-Bag utilise des capteurs gyromètre et accélérométre pour vérifier la cohérence de la direction et de la vitesse. Il faut un peu de temps pour pouvoir détecter la défaillance.	Le système doit vérifier que le capteur d'angle volant fonctionne correctement.	11b
capteur de vitesse longitudinale	défaillance du capteur	diagnostic par Safety-Bag en utilisant ses capteurs redondants	A3	La direction et le frein sont toujours contrôlés par l'AppCC.	$\sim 10^{-5}$	idem 12a				2	Le Safety-Bag doit disposer d'un capteur de vitesse longitudinal redondant. Dans notre véhicule expérimental, nous n'avons pas actuellement installé de capteur redondant. L'exigence de sécurité n'est pas implémentable par le Safety-Bag et la gravité ne peut donc pas être réduite et reste 4.	Le système doit vérifier que le capteur de vitesse longitudinale fonctionne correctement.	12b

Tableau 5.10 – AMDEC des composants véhicules autonomes avec l'intervention du système Safety-Bag (page 4)

Élément	**Type de Défaillance**	**Effets SB**	**Nécessité de sécurité considérée**	**Effets Véhicule**	$\lambda(h^{-1})$	**Détection**		**Action**		**Gravité des Conséquences**		**Exigence de Sécurité**	**Réf**
						moyen	t	moyen	t	G	Commentaires		
capteurs proprioceptifs (accéléromètre + gyromètre)	capteur ou connectique	Le SB juge uniquement par la dynamique du véhicule si la dynamique est dangereuse.	A2	après alerte, inhibition de l'accélération	$\sim 10^{-5}$	SB	_	conducteur	+2s	3a	Le Safety-Bag dispose d'un gyromètre et d'un accéléromètre redondants par rapport aux capteurs du véhicule. Sur le banc, ces deux capteurs ne servent à rien car le véhicule est immobile.	Le système doit vérifier que les capteurs proprioceptifs (accéléromètre latéral et gyromètre) fonctionnent correctement.	13b
capteurs extéroceptifs	capteur ou connectique	Le Safety-Bag empêche une trajectoire dangereuse et peut déclencher des alertes dès que nécessaire.	_	Si alerte, inhibition de l'accélération	_	idem 5b					Le Safety-Bag n'est pas compétent pour détecter cette défaillance. La défaillance de ces capteurs provoquent des défaillances fonctionnelles de l'application de contrôle-commande. Ce cas est donc un sous cas du cas d'erreurs fonctionnelles de l'application de contrôle-commande.	Le système doit vérifier que les capteurs extéroceptifs ne sont pas défaillants.	14b

Tableau 5.11 – AMDEC des composants véhicules autonomes avec l'intervention du système Safety-Bag (page 5)

Élément	**Type de Défaillance**	**Effets SB**	**Nécessité de sécurité considérée**	**Effets Véhicule**	$\lambda(h^{-1})$	**Détection**		**Action**		**Gravité des Conséquences**		**Exigence de Sécurité**	**Réf**
						moyen	t	moyen	t	G	Commentaires		
actionneur « accélérateur »	actionneur	détection par la vérification de la puissance consommée par le moteur du véhicule	U6	accélération inhibée + alertes visuelles et sonores	$\sim 10^{-5}$	SB	< 0.5s	conducteur	< 2s	2	La détection de cette défaillance nécessite un capteur d'intensité sur l'alimentation du moteur électrique du véhicule.	Le système doit détecter les incohérences entre la commande d'accélération produite par l'application de contrôle-commande et la commande d'accélération effectivement réalisée par le moteur du véhicule.	15a
actionneur « frein »	moteur électrique ou fil utilisé pour activer les freins	détection par la vérification de la pression effective dans le système de freinage hydraulique	U7	alertes visuelles et sonores + pas d'accélération supplémentaire	$\sim 10^{-5}$	SB	0.1s	conducteur	< 2s	3b	La redondance du système de freinage devra être repensée et renforcée pour les véhicules autonomes commerciaux.	Le système doit détecter les incohérences entre la commande de freinage produite par l'application de contrôle-commande et la commande effectivement réalisée par les freins du véhicule.	16b
actionneur « assistance de direction »	DAE	détection par la vérification de la vitesse de volant	U8	alertes visuelles et sonores + pas d'accélération supplémentaire	$\sim 10^{-5}$	SB	1s	conducteur	< 2s	3b	idem 16b. Les véhicules autonomes devront disposer d'un actionnement redondant de la direction. Certains équipementiers proposent déjà des doubles moteurs de la direction assistée électrique.	Le système doit détecter les incohérences entre la commande de direction produite par l'application de contrôle-commande et la commande effectivement réalisée par l'assistance à la direction.	17b

Nous précisons que les convertisseurs D/A sont maintenant intégrés au Safety-Bag (dans la carte AO308 du Safety-Bag Rules Checker), mais la gravité de leurs défaillances est réduite par les redondances intégrées au Safety-Bag, notamment la supervision assurée par le Safety-Bag Supervisor.

Nous pouvons constater que dans plusieurs cas que quand le Safety-Bag intervient dans le système, le niveau de gravité diminue. Nous détaillons ci-dessous la liste des défaillances qui montrent la réduction du niveau de gravité grâce à l'intervention du Safety-Bag :

- En cas de défaillance du capteur de vitesse longitudinale du véhicule (Réf. 12b dans le tableau 5.9), le Safety-Bag déclenche les alarmes et inhibe l'accélérateur. Il réduit la gravité de 5 à 2.

  Dans notre véhicule autonome expérimental, pour des raisons de coûts et de complexité qui devraient être faciles à résoudre sur un véhicule conçu pour être autonome, nous n'avons pas installé de tels capteurs redondants.

- En cas d'une panne bloquée ou une perte de tension de l'application de contrôle-commande (Réf. 1b et 2b dans le tableau 5.7), d'un gel des calculateurs de l'application de contrôle-commande (Réf. 4b dans le tableau 5.7), ou en cas de défaillance des capteurs proprioceptifs (accéléromètre et gyromètre) du véhicule (Réf. 13b dans le tableau 5.10), le Safety-Bag intervient en déclenchant les alarmes, en freinant et en maintenant l'angle volant. Néanmoins, le véhicule ne suit plus une trajectoire de conduite correcte. Le Safety-Bag réduit ainsi la gravité de 5 à 3a. Cette défaillance est donc rattrapable par le conducteur qui doit reprendre la conduite en manuel immédiatement.

- En cas de défaillance de l'actionneur de frein (Réf. 16b dans le tableau 5.11), le Safety-Bag peut détecter la défaillance de l'actionnement de freinage, il peut déclencher les alarmes, mais il ne dispose pas de moyens pour arrêter le véhicule. La gravité est également réduite de 5 à 3b.

- En cas de défaillance du CAN (Réf. 6b et 8b dans le tableau 5.8), de défaillance du capteur d'angle volant du véhicule (Réf. 11b dans le tableau 5.9) et de défaillance de l'actionneur de la direction (Réf. 17b dans le tableau 5.11), le Safety-Bag est capable de détecter le problème et de rendre la main au conducteur, réduisant la gravité de 5 à 3b. Cette réduction de risque n'est cependant valable que si l'absence de l'information d'angle volant est bien détectée par le Safety-Bag sur le bus CAN, que les capteurs sont redondés et que l'intensité dans le moteur de l'assistance de la direction est vérifiée.

Dans notre véhicule autonome expérimental, nous n'avons pas installé un capteur d'angle volant redondant, ni un capteur d'intensité sur le moteur de la direction assistée électrique.

- Les erreurs fonctionnelles de la partie logicielle de l'application de contrôle-commande (Réf. 5b dans le tableau 5.7) peuvent être extrêmement diverses et une analyse AMDEC ne pourrait être basée que sur les composants réels de cette application.

  A défaut, nous nous appuyons sur les résultats de l'analyse HazOp, qui identifie des exigences de sécurité. Certaines exigences de sécurité sont vérifiables par le Safety-Bag et d'autres non. Pour celles qui ne sont pas implémentables par le Safety-Bag, aucune réduction de risque ne peut être assurée par le Safety-Bag. D'autres méthodes complémentaires doivent être utilisées. La gravité du non respect de ces exigences de sécurité, notamment (H7, H12, H13, H15, H16 et H20 dans les tableaux HazOp du chapitre précédent) reste 5.

  Pour les exigences de sécurité H1, H2 et H4, la gravité passe de 5 à 3a ou 3b. L'angle volant dans ce cas est maintenu et le freinage est modéré.

  Pour les exigences de sécurité H6, H9 et H11, la gravité passe de 5 à 2. Le véhicule freine, mais continue à suivre la trajectoire courante tant qu'elle n'est pas obsolète, soit pendant 1.6 secondes. La vitesse a déjà diminué et le conducteur est en train de reprendre le contrôle en mode manuel.

Remarque

Sur notre véhicule autonome expérimental, la direction assistée électrique tombe souvent en défaillance si les commandes ou les réactions du banc ne correspondent pas à son modèle de conduite. Dans ce cas, la conduite autonome ne peut plus contrôler le volant et la conduite en mode manuel s'effectue sans assistance de direction. Si le Safety-Bag implémentait la vérification de bon fonctionnement de l'actionneur de direction, les alarmes seraient activées et le conducteur serait averti rapidement. Notons que la fréquence significative de ce problème est dû à une robotisation d'un véhicule existant, et devrait être réduite significativement sur un véhicule conçu dés le départ pour être autonome.

5.3 Description des moyens d'expérimentation

Nous décrivons dans cette section les moyens utilisés pour effectuer les différentes expérimentations présentées dans la suite de ce chapitre. Nous présentons tout d'abord la piste SEVILLE sur laquelle nous avons effectué certaines expériences d'interactions homme-machine face à des défaillances de véhicule autonome. Nous présentons ensuite le banc VILAD, une plateforme Vehicle In the Loop (VIL) permettant de faire évoluer un véhicule réel dans un environnement simulé. Finalement, nous décrivons le joueur de scénarios et le suiveur de trajectoire qui nous ont servi dans nos campagnes d'injection de fautes sur le banc VILAD.

5.3.1 La piste d'essai Seville

La piste SEVILLE 5.5 est une route de 60 mètres composée d'une ligne droite avec un rond-point à chaque extrémité.

Figure 5.5 – Piste Seville

5.3.2 Présentation du banc VILAD

Le banc VILAD (Vehicle in The Loop for Autonomous Driving) 5.6 permet de contrôler un véhicule réel (ici une Fluence électrique) dans un environnement virtuel. Des moteurs électriques, reliés à l'emplacement des roues, fournissent l'inertie du véhicule dûs à sa masse. Le poids des moteurs ne permet pas de faire tourner les

roues latéralement, et les mouvements du volant et de la crémaillère de direction sont rendus possibles par la déconnexion des biellettes de direction, ce qui les déconnecte des roues et leur donne la possibilité de tourner indépendamment de ces dernières. Un moteur relié à la crémaillère à la place d'une des biellette (la droite) fournit les efforts correspondants au frottement des pneus sur la chaussée. Côté véhicule, ce sont les systèmes standards moteur et frein qui régulent la vitesse de rotation des roues pour donner la vitesse longitudinale du véhicule, et la direction assistée électronique qui détermine le comportement latéral du véhicule.

Figure 5.6 – La Fluence sur le banc VILAD

La figure 5.8 présente l'architecture du banc VILAD. Le logiciel Scanner Studio simule le comportement du véhicule dans le monde virtuel. Le comportement dynamique du véhicule est calculé par le logiciel Callas, qui utilise des paramètres détaillés de sa structure mécanique. Scanner Studio obtient de Callas les efforts à appliquer aux roues et à la direction. Ces efforts sont transmis au logiciel de commande des moteurs Rotronics. Lors de la simulation, des *logs* complets sont produits par Scanner. Trois PC assurent le fonctionnement du banc comme montre la figure 5.7.

Le logiciel Scanner Studio peut simuler des capteurs du véhicule : les capteurs inertiels, les lidars, les radars, des caméras, le GPS, etc.

Le banc VILAD permet de conduire manuellement le véhicule avec un ressenti assez crédible. Ses intérêts sont multiples. Premièrement, le comportement du système autonome sur le banc VILAD est bien plus réaliste que toute simulation : les interactions entre les différents composants du véhicule (matériels et logiciels) sont réelles, mis à part les efforts dynamiques latéraux et certains capteurs simulés. Deuxièmement, il permet de rejouer la même expérience avec une reproductibilité significative. Enfin, il offre un cadre expérimental sûr permettant de réaliser des expérimentations (notamment d'injection de fautes) bien trop dangereuses à mettre

en place sur de vraies pistes.

Le banc VILAD a eu un développement progressif au laboratoire Heudiasyc sur la même période que nos travaux de validation. La première version du banc, disponible en 2016, ne disposait pas des capacités de commande latérale du véhicule. Des travaux sur le banc en novembre 2017 puis finalisés en avril 2018 ont permis de rajouter cette fonctionnalité. Nos premières expérimentations, présentées en section 5.4.1 ont été réalisées sur la première version du banc, tandis que les expérimentations présentées en section 5.4.2 ont été réalisées sur la deuxième version.

Figure 5.7 – La salle de contrôle

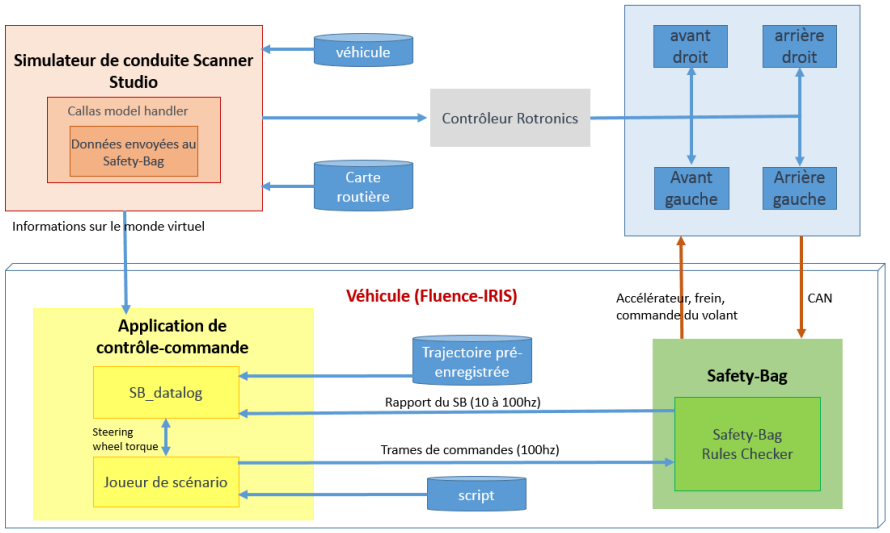

Figure 5.8 – Banc de simulation VILAD

5.3.3 Joueur de scénarios

La figure 5.8 présente les interactions du véhicule incluant le Safety-Bag, notre application de contrôle-commande scriptée, et le simulateur Scanner Studio contrôlant

les moteurs reliés aux roues. Scanner Studio dispose d'une base de données des caractéristiques du véhicule et d'une description des environnements virtuels sur lesquels nous allons effectuer nos expérimentations. Le simulateur commande alors les efforts à appliquer aux roues par l'intermédiaire du contrôleur développé par Rotronics.

Il est important de noter que pendant les expériences effectuées dans nos travaux, nous avons utilisé des commandes scriptées plutôt qu'une véritable architecture de commande pour trois raisons principales.

- Premièrement, nous n'avons pas encore développé l'application de contrôle-commande complète sur notre véhicule autonome expérimental.

- Deuxièmement, l'injection de fautes est plus facile à effectuer avec des commandes scriptées, pour des raisons de reproductibilité et de facilité d'injection. En effet, le comportement d'un script est bien plus prédictible et reproductible qu'une application contenant des mécanismes d'intelligence artificielle, et les injections de fautes peuvent être réalisées simplement en modifiant des lignes du script.

- Troisièmement, même si la réalisation d'expériences sur notre banc d'essai est beaucoup plus sûre que de faire rouler le véhicule expérimental sur des pistes réelles, injecter des fautes dans un tel système autonome reste toujours une tâche dangereuse, surtout pour l'opérateur. En effet, le simulateur emploie des moteurs de grande puissance dont des utilisations limites peuvent provoquer des dangers pour le matériel et les opérateurs éventuels. Ainsi, l'utilisation des commandes scriptées donne plus de contrôlabilité et donc de sécurité-innocuité pendant nos expériences.

La commande de la Fluence est générée sur un calculateur de type « *CUBE* » (défini dans 5.1.2.2), qui envoie des commandes sous forme de signal analogique pour les freins, l'accélérateur et la direction en simulant le capteur de couple de la direction assistée électrique. Pour des raisons de limitations des ressources et de simplification de l'architecture, ce calculateur supporte également le composant logiciel du Safety-Bag Rules Checker.

Pour tester les nécessités de sécurité identifiées précédemment à travers les deux méthodes d'analyse de risques AMDEC et HazOp-UML et pour évaluer le comportement du Safety-Bag en cas de défaillances, un « *joueur de scénarios* » (5.8), a été développé par Paul Crubillé, Ingénieur de recherche au laboratoire Heudiasyc. Ce composant est un logiciel s'exécutant sur un PC sous LINUX et produisant des

trames de commandes, permettant de commander le véhicule. Ce *joueur de scénarios* lit une description du scénario dans un fichier texte qui précise :

- les séquences de trames à envoyer au Safety-Bag Rules Checker,
- les valeurs des commandes pour les actionneurs,
- et la fréquence des trames.

Le *joueur de scénarios* permet de jouer et de rejouer des scénarios (incluant des phases d'accélération et des phases de freinage) sur le banc VILAD avec une bonne répétabilité bien que la commande soit réalisée en boucle ouverte, sans mécanismes supplémentaires, c'est-à dire sans tenir compte du résultat de la commande par un retour d'informations (provenant généralement de capteurs). Cette approche en boucle ouverte ne permet pas un contrôle correct de la direction, puisqu'un retour d'information est nécessaire aux fonctionnalités de suivi de trajectoire.

Dans l'exemple simplifié ci dessous, le CUBE émetteur qui joue le rôle de l'application de contrôle-commande, commence par émettre 100 trames à la fréquence de 100 Hz dans lesquelles sont précisées que l'application n'est pas en défaut (« $STATE\_OK$ ») et les actionneurs à 0. Ensuite, au bout donc d'une seconde, il envoie 1000 trames à la même fréquence avec une accélération modérée (800 mVolts). Pour finir, il envoie des trames avec une consigne de couple sur la direction. Exemple de script :

```
100 frames rate 100 hz
```
$producer\_state = STATE\_OK$, $couple\_direction = 0$,
$acceleration = 0$, $brake = 0$;
```
1000 frames rate 100 hz
```
$acceleration = 800$;
```
1000 rate 100 hz
```
$couple\_direction = -120$;

5.3.4 Suiveur de trajectoire

Pour commander le volant, un module de suiveur de trajectoire a été intégré au « *SB_ datalog* » (voir figure 5.8).

Ce dernier obtient de Scanner les informations sur le monde virtuel, notamment l'état cinématique (position, vitesse et cap). Le suiveur de trajectoire calcule alors l'angle des roues et l'angle volant pour suivre une trajectoire pré-enregistrée. A partir des informations de position et de vitesse du volant disponibles sur le bus CAN du véhicule, une consigne de couple est calculée. Cette consigne de couple est mise à disposition du joueur de scénarios dans une zone du *mémoire partagée*.

Cette mémoire partagée permet également de mettre à disposition du joueur de scénarios l'état cinématique du véhicule calculé par Scanner, les horodatages de la trajectoire cinématique et de l'état cinématique.

Notons que l'horodatage de la trajectoire cinématique est la date du calcul effectué par le suiveur de trajectoire, par contre l'horodatage de l'état cinématique est la date fournie par Scanner et correspondant à la date de l'échantillon de données du monde virtuel.

Utilité d'une nécessité de sécurité pour les campagnes d'expérimentations sur le banc VILAD

La nécessité de sécurité : « *Le Safety-Bag doit vérifier la vivacité de l'état cinématique. Si l'état cinématique n'est pas mis à jour, le Safety- Bag intervient en déclenchant les alarmes, en inhibant des commandes de l'accélération et/ou forçant une action de sécurité telle que le freinage du véhicule et le maintien de l'angle volant après un certain délai.* » a été implémentée dans le Safety-Bag. Elle ne fait pas partie des nécessités de sécurité ciblée dans notre campagne d'expérimentations, mais nous a malgré tout été utile lors des expériences. En effet, son implémentation amène à ce que le Safety-Bag intervienne si Scanner cesse d'envoyer les informations nécessaires au bon fonctionnement du système (vitesse, position et cap dans le monde virtuel).

Ainsi, en cas de défaillance de Scanner ou lors d'une activation du bouton d'arrêt d'urgence du banc VILAD, cette nécessité de sécurité sera violée et le Safety-Bag arrêtera le véhicule très rapidement sans besoin d'intervention humaine. Sans cette nécessité de sécurité, le moteur tournerait à une vitesse risquant d'endommager le banc, ce qui demande une intervention immédiate d'un opérateur.

5.3.5 Injection de fautes possibles

Le langage dans lequel sont exprimés les scénarios permet d'introduire des fautes diverses (arrêt des émissions, fréquence erronée et données incohérentes, etc.), qui peuvent être interprétées comme une défaillance par le Safety-Bag. L'écriture de scripts permet les injections de fautes suivantes :

- **Arrêt inopiné des commandes :**

  L'arrêt inopiné des commandes permet de simuler un gel de l'application de contrôle-commande (logiciel ou matériel), et les dernières commandes envoyées sur les actionneurs seront maintenues par ceux-ci. Cela permet de valider par exemple a nécessité de sécurité U1 : « *Le Safety-Bag doit vérifier la vivacité de l'application de contrôle-commande. Si celle-ci n'envoie plus de commande, le*

Safety-Bag doit prévenir le conducteur et repasser en mode manuel et arrête le véhicule en allumant les feux de détresse. ».

L'arrêt inopiné de commandes se fait simplement en tronquant le script à partir de l'instant désiré.

Sans détection de cette faute, le véhicule peut avoir un comportement dangereux, même sur le banc VILAD, en poussant par son comportement les moteurs Rotronics liés aux roues au-delà de leur limites de fonctionnement.

Pour limiter les risques, nous rajoutons des trames 10 secondes après l'arrêt inopiné pour couper l'accélération et freiner le véhicule.

- **Fréquence insuffisante :**

  Une fréquence insuffisante simule des fautes temporelles comme un ralentissement de l'application de contrôle-commande dû à de nombreux calculs ou à des problèmes de communication.

  Cette injection de faute est réalisée en modifiant la valeur rate qui indique la fréquence de l'envoi de la trame. Cela permet de valider par exemple la nécessité de sécurité U2 : « *Le Safety-Bag doit vérifier la cohérence temporelle de l'application de contrôle-commande. En cas d'incohérence, le Safety-Bag doit assurer la mise en sécurité du système en levant les larmes et en repassant en mode manuel.* ».

  Sans détection de cette faute injectée, le contrôle du véhicule peut être médiocre voire même présenter des risques y compris sur le banc VILAD. L'expérimentation doit être menée en dégradant progressivement le comportement et l'opérateur doit rester vigilant.

- **Données d'état du système incohérentes :**

  Des données d'état du système incohérentes simulent des fautes logicielles de l'application de contrôle-commande concernant la mise à jour et le calcul de l'état du véhicule, ou des fautes logicielles et matérielles des composants de perception. Cela permet par exemple de valider les nécessités de sécurité A2 : « *Le Safety-Bag doit vérifier que les capteurs proprioceptifs longitudinaux (capteurs de vitesse et d'accélération longitudinales) fonctionnent correctement. Dans le cas contraire, le Safety-Bag déclenche les alarmes, inhibe l'accélération, freine modérément le véhicule et applique la valeur d'angle volant venant de l'application de contrôle-commande.* » et A3 : « *Le Safety-Bag doit vérifier que les capteurs proprioceptifs latéraux (accélération latérale et gyromètre) fonctionnent correctement. Dans le cas contraire, le Safety-Bag déclenche les*

alarmes, inhibe l'accélération, freine modérément le véhicule et applique la valeur d'angle volant venant de l'application de contrôle-commande. ».

Cette injection de faute est réalisée en précisant dans le script de scénario des valeurs qui vont écraser les valeurs correspondant au déroulement de l'expérimentation dans le monde virtuel (speed = 1000 dans l'exemple ci-dessous remplace la valeur calculée et transmise par Scanner Studio). Ces valeurs seront ainsi transmises au Safety-Bag (si il est activé). Nous pouvons notamment modifier tous les champs de l'état cinématique.

Exemple de script :

```
500 frames rate 100 hz
```
$speed = 1000,\ acceleration = 1100,\ brake = 0;$

- **Perte de trames ou trames désordonnées :**

  Les pertes de trames simulent les problèmes de communication, notamment les pannes réseau. Cela permet de valider les nécessités de sécurité U1 : « *Le Safety-Bag doit vérifier la vivacité de l'application de contrôle-commande. Si celle-ci n'envoie plus de commande, le Safety-Bag doit prévenir le conducteur et repasser en mode manuel et arrête le véhicule en allumant les feux de détresse.* » et U2 : « *Le Safety-Bag doit vérifier la cohérence temporelle de l'application de contrôle-commande. En cas d'incohérence, le Safety-Bag doit assurer la mise en sécurité du système en levant les larmes et en repassant en mode manuel.* ».

  Cette injection de faute est réalisée soit en supprimant une trame pour la perte de trames, soit en modifiant le numéro de séquence comme illustré dans l'exemple ci-dessous pour les trames désordonnées.

  Exemple de script :

  ```
  500 frames from frame 101 rate 100 hz
  ```
  $acceleration = 1100,\ brake = 0;$

- **Trajectoire cinématique non mise à jour :**

  Pour simuler un arrêt du calcul ou de la mise à jour de la trajectoire cinématique du système du fait de fautes logicielles, l'horodatage de la trajectoire cinématique cesse d'être incrémenté par le suiveur de trajectoire au moment désiré. Cela permet de valider la nécessité de sécurité H4 : « *Le Safety-Bag doit vérifier la vivacité de la trajectoire cinématique. Dans le cas*

contraire, le Safety-Bag doit déclencher les alarmes, inhiber l'accélération, freiner et maintenir la position d'angle volant après un certain temps. »

- **Commandes aberrantes :**

  Le joueur de scénario permet d'introduire les commandes aberrantes en spécifiant des valeurs de commandes pour l'accélération, le freinage ou le couple volant pour la durée et la fréquence désirée. Pour l'accélération et le freinage, ces commandes modifiées sont substituées aux commandes du scénario nominal à un instant choisi. Pour le couple volant et quand le suiveur de trajectoire est actif, les consignes de couple volant présentes dans le fichier de scénarios remplacent les consignes de couple volant calculées par le suiveur de trajectoire. Cela permet de valider les nécessités de sécurité U3 : « *Le Safety-Bag doit vérifier les bornes de vitesse maximale du véhicule. Dans le cas contraire, le Safety-Bag déclenche les alarmes, bloque les commandes d'accélération de l'application de contrôle-commande et assure la reprise en mode manuel.* » et U4 : « *Le Safety-Bag doit vérifier le contrôle dynamique du véhicule : il doit vérifier que l'accélération latérale reste inférieure à une valeur seuil. Si ces bornes ne sont pas respectées, le Safety-Bag doit déclencher les alarmes et assurer la reprise en mode manuel.* ».

  Nous avons utilisé cette approche pour les scénarios avec accélération incorrecte sans contrôle latéral. Néanmoins, nous n'avons pas injecté de fautes de cette manière pour notre campagne avec le contrôle latéral.

5.4 Validation des nécessités de sécurité

Nous décrivons maintenant les expérimentations que nous avons effectuées sur le banc VILAD sans et avec contrôle latéral.

La figure 5.9 représente l'environnement simulé par Scanner Studios et sur lequel nous avons effectué les différentes expériences.

5.4.1 Expérimentations sur le banc VILAD : scénario sans contrôle latéral

Nous testons dans cette section les deux nécessités de sécurité suivantes :

1. U1 : Le Safety-Bag doit vérifier la vivacité de l'application de contrôle-commande. Si celle-ci n'envoie plus de commande, il doit prévenir le conducteur et repasser en mode manuel et arrête le véhicule en allumant les feux de détresse.

Figure 5.9 – L'environnement simulé et utilisé dans les expérimentations

2. U3 : Le Safety-Bag doit vérifier les bornes de vitesse. Dans le cas contraire, le Safety-Bag déclenche les alarmes, bloque les commandes d'accélération de l'application de contrôle-commande et assure la reprise en mode manuel.

5.4.1.1 Scénario nominal

La figure 5.10 représente la trajectoire de référence que nous demandons au système de suivre. Cette trajectoire est nommée *trajectoire de consigne*. Le scénario nominal cherche donc à suivre cette trajectoire, du point de départ jusqu'au point d'arrivée.

La trajectoire de consigne part donc du *point de départ* et consiste en une ligne droite de plus de 200 mètres suivie de 2 virages à droite (1 et 2 dans 5.10) puis au croisement (3) le véhicule tourne à gauche. Après 150 mètres, le véhicule aborde à nouveau deux virages (4 et 5) à droite suivis d'un virage à gauche (6) qui est presque aussi serré qu'un tourne-à-gauche dans un croisement. Le véhicule revient finalement sur la ligne droite sur laquelle se trouve le *point d'arrivée*.

La figure 5.11 présente la commande appliquée pour le contrôle longitudinal (accélération et freinage) du véhicule pendant le scénario nominal. Une valeur positive comprise entre 0 et 1,8 volts commande l'accélération de la Fluence. Une valeur négative comprise entre 0 et -5 volts commande le système de freinage. Bien qu'il soit possible d'appliquer simultanément des consignes sur ces deux actionneurs, ce n'est pas le cas dans cette campagne d'expérimentations. Sur la figure 5.10, on voit que le véhicule accélère avec une consigne d'accélération fixée à environ 1 Volt pour maintenir une vitesse d'environ 35 km/h. Ensuite, le véhicule freine pour s'arrêter

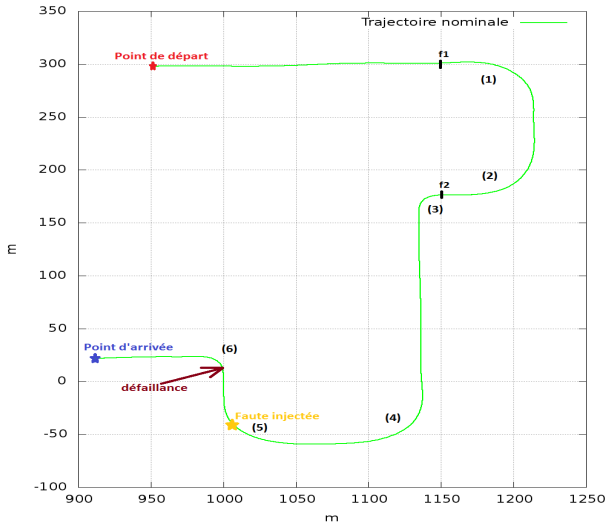

Figure 5.10 – Trajectoire de consigne suivie par le véhicule

au premier feu f1, puis redémarre pour parcourir les deux courbes à droite (1) et (2) avant de s'arrêter au feu f2 juste avant de s'arrêter au feu f2 juste avant le virage (3). Le véhicule redémarre à 1 Volt d'accélération pour réaliser le reste du trajet. Vers 120 secondes, l'accélération est relâchée pour modérer la vitesse pour le passage du virage (6) avant de remonter à 1 Volt après le virage.

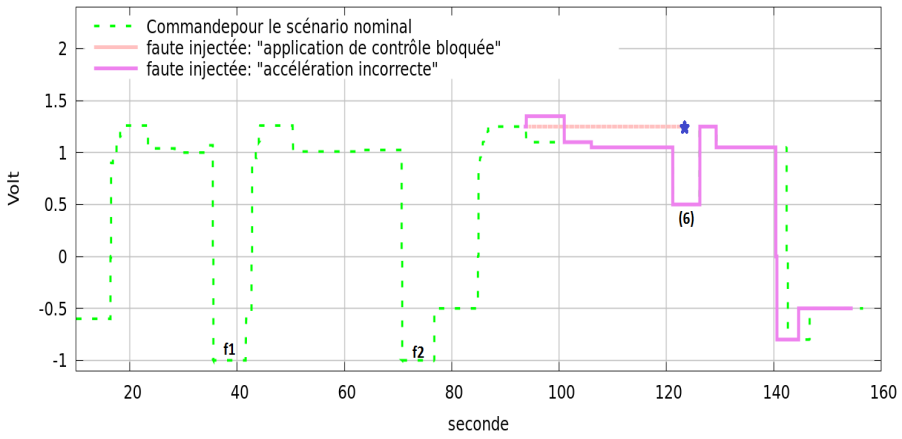

Figure 5.11 – Commandes nominales et fautes injectées à partir du bloc de contrôle-commande

5.4.1.2 Campagne d'injection de fautes

Nous injectons deux fautes différentes, chacune dans une expérience, en modifiant les commandes envoyées par notre application de contrôle-commande simulée. La première faute injectée consiste à simuler une application verrouillée qui ne modifie pas la dernière commande appliquée, tandis que la seconde faute injectée consiste à

envoyer une commande d'accélération erronée, supérieure à l'accélération nominale. Les fautes injectées peuvent être vues sur la figure 5.11.

Les fautes sont injectées précisément à 93 secondes pendant les expériences (sur le virage à droite indiqué sur la figure 5.10), mais leurs effets ne se font pas sentir sur le comportement du système avant le prochain virage à gauche. Notons que lors de la première injection (application de contrôle commande bloquée), l'opérateur interrompt la navigation autonome à 123 secondes car le comportement du véhicule s'avère dangereux. Lors de la deuxième injection (accélération incorrecte), nous avons ajouté une modification du script à 140 secondes pour freiner car le véhicule était plus loin sur la route en raison de l'augmentation erronée injectée sur l'accélération.

5.4.1.3 Résultats et analyses

Cette section présente les résultats des deux fautes injectées sur notre banc d'essai VILAD. Pour chaque faute, les figures 5.12 et 5.13 montrent les commandes longitudinales appliquées aux actionneurs de la voiture (accélération et freinage) dans trois cas :

1. le cas nominal,
2. le cas avec faute injectée sans Safety-Bag,
3. le cas avec faute injectée en présence du Safety-Bag.

La différence entre les commandes appliquées dans les cas 2 et 3 est bien entendu seulement due aux mesures de sécurité prises par le Safety-Bag après qu'une nécessité de sécurité ait été identifiée comme violée.

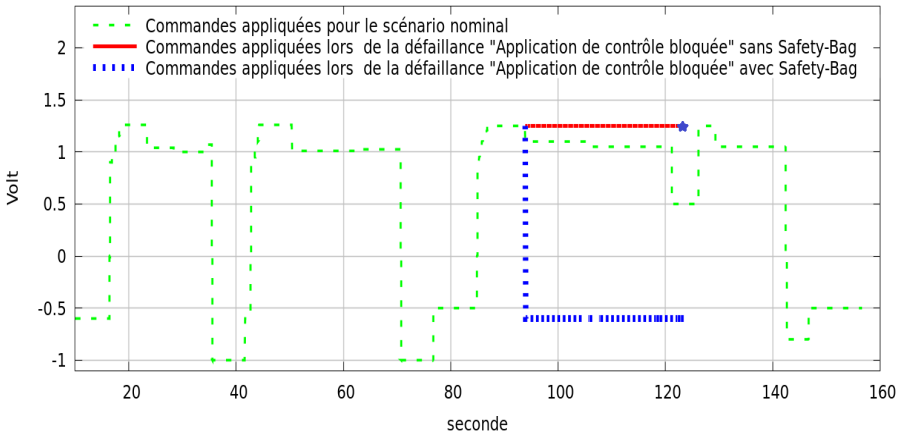

Figure 5.12 – Commandes appliquées aux actionneurs pour la défaillance : blocage de l'application de contrôle-commande

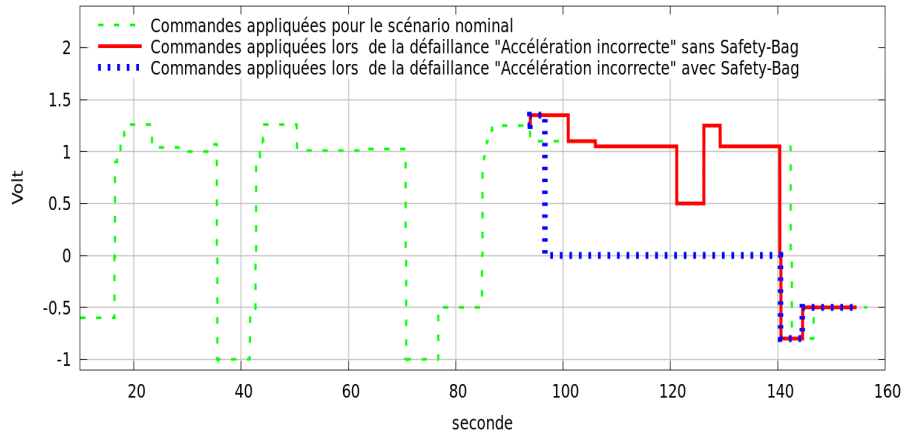

Figure 5.13 – Commandes appliquées aux actionneurs pour la défaillance : accélération erronée

a) Faute injectée : Application de contrôle-commande bloquée Nous testons ici la nécessité de sécurité U1 : « *Le Safety-Bag doit vérifier la vivacité de l'application de contrôle-commande. Si celle-ci n'envoie plus de commande, il doit prévenir le conducteur et repasser en mode manuel et arrête le véhicule en allumant les feux de détresse.* »

Sans Safety-Bag, on voit sur les figures 5.14 et 5.15 que le véhicule accélère jusqu'à 80 km/h avant le virage. Le ralentissement brutal suivant est dû au fait que le véhicule a glissé sur la route avant d'être arrêté par l'opérateur. La distance à la trajectoire saute à près de 6 mètres, ce qui montre que le contrôle du véhicule a clairement été perdu.

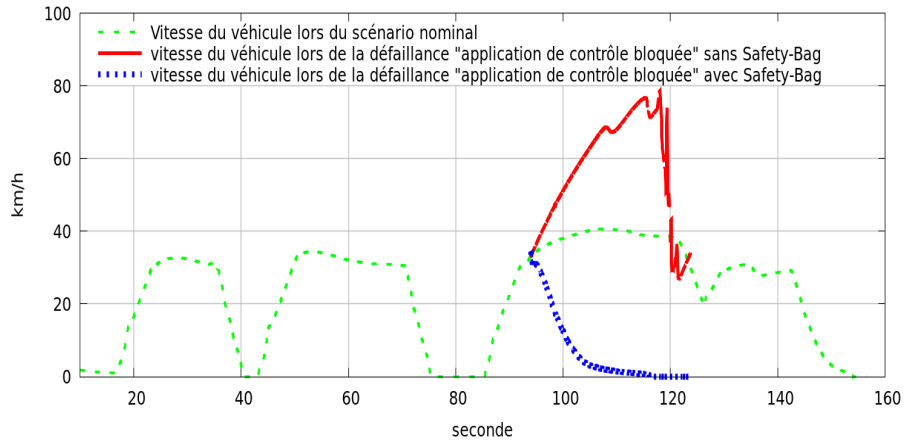

Figure 5.14 – Vitesse du véhicule en cas de défaillance : application de contrôle-commande bloquée

Avec Safety-Bag, on voit sur les mêmes figures que le gel de l'application de contrôle-commande est détecté en quelques cycles (quelques dizaines de millise-

condes). Ceci viole la nécessité de sécurité stipulant que la vivacité de l'application de contrôle-commande doit être assurée, et applique l'action de sécurité consistant à rejeter toutes les commandes d'accélération de l'application de contrôle-commande tout en alertant l'opérateur de sa défaillance. Le système s'arrête donc à 117 secondes sans intervention de l'opérateur, 24 secondes après la détection de fautes.

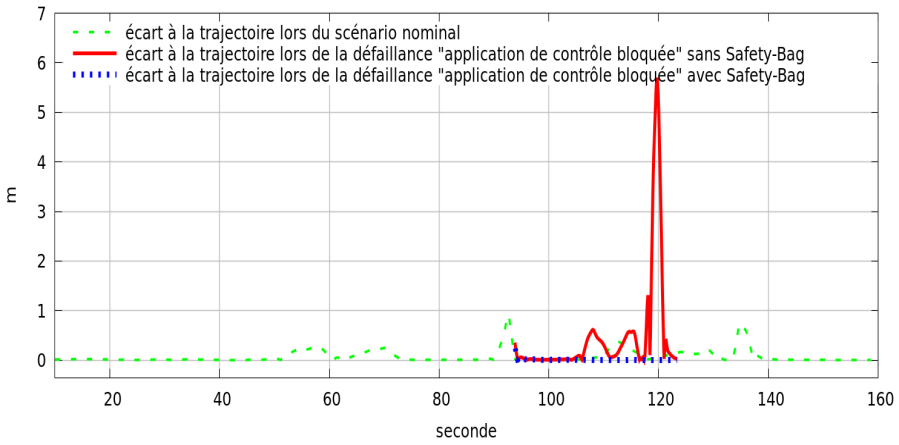

Figure 5.15 – Écart à la trajectoire du véhicule dans le cas de la défaillance : application de contrôle-commande bloquée

b) Faute injectée : Accélération incorrecte Nous testons ici la nécessité de sécurité U3 : « *Le Safety-Bag doit vérifier les bornes de vitesse. Dans le cas contraire, le Safety-Bag déclenche les alarmes, bloque les commandes d'accélération de l'application de contrôle-commande et assure la reprise en mode manuel.* »

Sans Safety-Bag, nous voyons sur la figure 5.16 que le véhicule accélère jusqu'à près de 70 km/h avant le virage à gauche. Les conséquences ne sont ainsi pas aussi graves que la faute injectée précédemment, mais la figure 5.17 montre encore des écarts significatifs par rapport à la trajectoire nominale, ce qui peut être dangereux en particulier sur une route à double sens.

Avec Safety-Bag, on voit que le véhicule ne dépasse pas 50 km/h. En effet, l'accélération erronée viole la nécessité de sécurité stipulant que le véhicule ne doit pas dépasser 47 km/h et l'action de sécurité inhibe toutes les commandes d'accélération en déclenchant une alarme pour l'opérateur, de la même façon que lors de l'injection précédente. Notons que dans notre implémentation actuelle, le Safety-Bag passe ici à un état dégradé duquel il ne peut pas sortir, mais qu'il serait possible de retourner à un mode de fonctionnement nominal une fois la vitesse repassée en-dessous de la valeur seuil de la nécessité de sécurité.

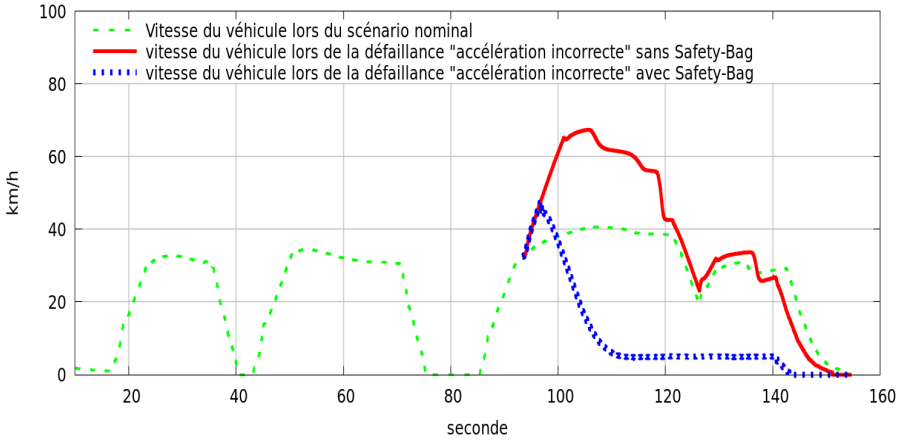

Figure 5.16 – Vitesse du véhicule en cas de défaillance : accélération incorrecte

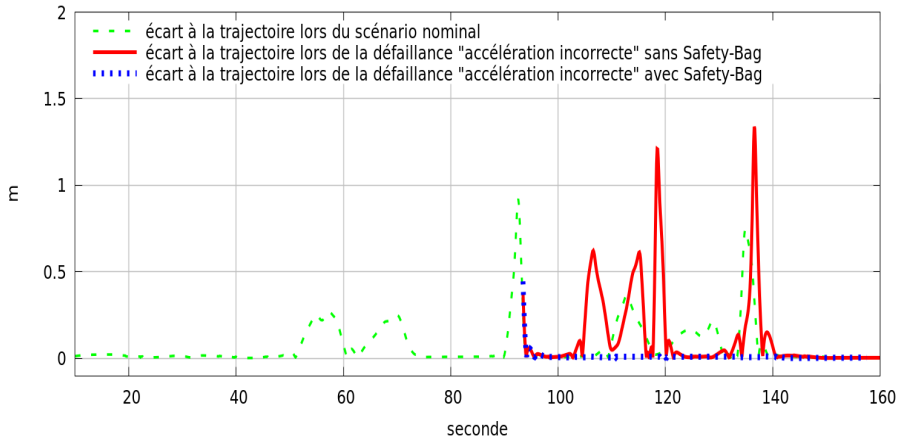

Figure 5.17 – Écart à la trajectoire du véhicule dans le cas de la défaillance : accélération incorrecte

5.4.2 Expérimentations sur le banc VILAD : scenario avec contrôle latéral

Nous testons dans cette section les quatre nécessités de sécurité suivantes avec le contrôle latéral du véhicule :

1. U1 : Le Safety-Bag doit vérifier la vivacité de l'application de contrôle-commande. Si celle-ci n'envoie plus de commande, il doit prévenir le conducteur et repasser en mode manuel et arrête le véhicule en allumant les feux de détresse.

2. A1 : Le Safety-Bag doit vérifier la circulation des trames du bus CAN. En cas d'absence de communication pendant une durée trop importante, le Safety-Bag doit alerter le conducteur et repasser en mode manuel.

3. U4 : Le Safety-Bag doit vérifier le contrôle dynamique du véhicule : Il doit

vérifier que l'accélération latérale reste inférieure à une valeur seuil. Dans le cas contraire, le Safety-Bag doit déclencher les alarmes et assurer la reprise en mode manuel.

4. H4 : Le Safety-Bag doit vérifier la vivacité de la trajectoire cinématique. Dans le cas contraire, le Safety-Bag doit déclencher les alarmes, inhiber l'accélération, freiner et maintenir la position d'angle volant après un certain temps.

5.4.2.1 Scénario nominal

Afin d'obtenir une trajectoire de consigne pour le suivi de trajectoire latéral, nous avons préalablement effectué un parcours en conduisant manuellement sur le banc VILAD. Lors du fonctionnement autonome du véhicule, le suiveur de trajectoire, qui s'exécute sur le CUBE, commande latéralement le véhicule pour rester au plus prés de la trajectoire, tandis que le joueur de scénario commande longitudinalement le véhicule en fonction du script de commande qui a été construit.

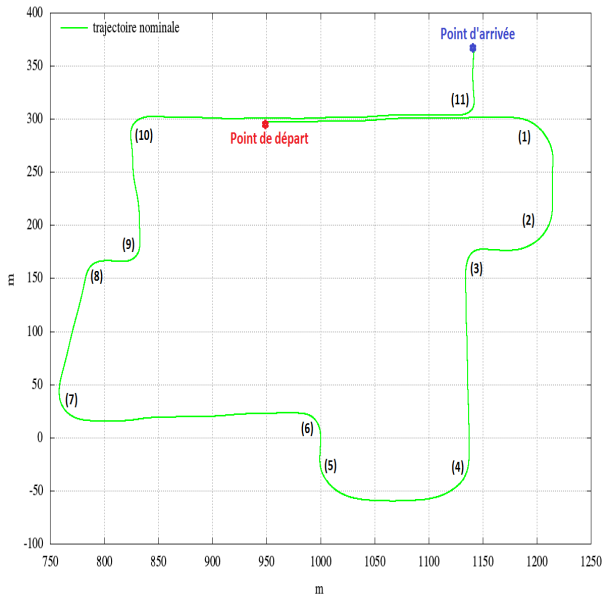

Figure 5.18 – Trajectoire pré-enregistrée suivie par le véhicule à basse vitesse

Cette trajectoire de consigne part du *point de départ* et consiste en une ligne droite de plus de 200 mètres suivie de 2 virages à droite (1 et 2 dans 5.18) puis au croisement (3) le véhicule tourne à gauche. Après 150 mètres, le véhicule aborde à nouveau deux virages (4 et 5) à droite suivis d'un virage à gauche (6) qui est presque aussi serré qu'un tourne-à-gauche dans un croisement. Après 200 mètres, le véhicule aborde un virage serré à droite (7) suivi d'un virage (8) moins serré qui précède un croisement(9) où le véhicule tourne à gauche. Après 100 mètres, le véhicule tourne à droite au croisement (10). Le véhicule revient alors sur la ligne droite sur laquelle se

trouve le point de départ et tourne à gauche au carrefour (11) au bout de cette ligne. Cette trajectoire est en fait une version plus longue de la trajectoire de consigne des expérimentations sans contrôle latéral.

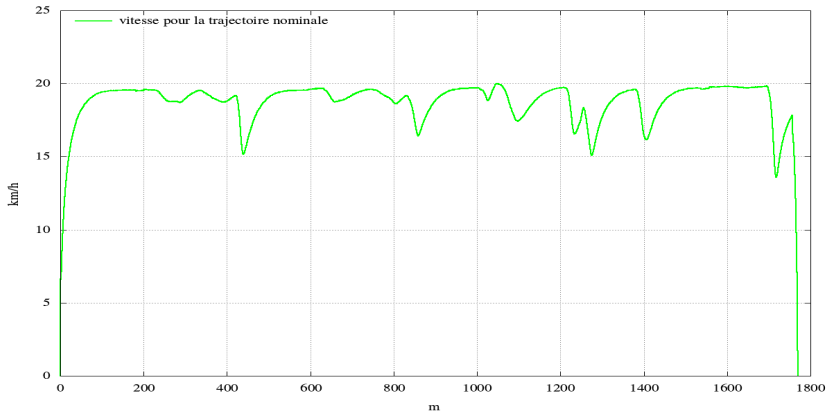

Figure 5.19 – Vitesse nominale considérée sans injection de fautes

Pour suivre au plus près la trajectoire nominale, nous avons fait des expériences dans lesquelles la voiture conduit très lentement (à moins de 20 km/h). Cette vitesse n'est pas explicitement définie dans le script des commandes puisque c'est l'accélération longitudinale est commandée plutôt que la vitesse. La commande maximale d'accélération donnée par le script est 0.99 Volts. Dans chaque virage, la résistance à l'avancement augmente et la vitesse diminue avant de retrouver progressivement une valeur juste inférieure à 20 km/h (voir figure 5.19).

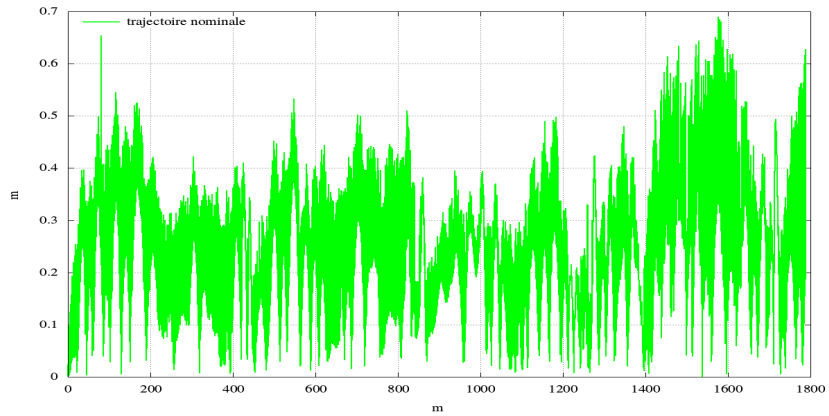

Figure 5.20 – Écart de la trajectoire nominale du véhicule par rapport à la trajectoire pré-enregistrée à basse vitesse lors du suivi à environ 20 km/h

L'écart à la trajectoire sur la courbe 5.20 est inférieur à 70 cm. Le suiveur de trajectoire n'est donc pas très précis et pourrait être amélioré. Notons que deux mêmes expériences donnent des résultats similaires, bien qu'imparfaitement reproductibles. Pour cette raison, en pratique, nous avons effectué 5 fois chaque

expérience pour avoir des résultats plus représentatifs. Nous présentons ainsi des résultats de plusieurs mêmes expériences dans les figures de nos figures. Il faut également souligner qu'avec le suiveur de trajectoire, la vitesse de passage en courbe du véhicule est très inférieure à ce qu'elle est dans les expériences précédentes où le contrôle latéral était laissé à Scanner parce que la direction assistée électrique que nous avons utilisée pour le contrôle latéral est lente à réagir.

Pour les expérimentations, la vitesse est trop faible (à peine 20 km/h) pour que les défaillances du système aient des effets significatifs. Nous avons choisi de retenir deux niveaux de consigne d'accélération constants pour mener les expériences. Le premier correspond à une accélération de 1.05 Volts, qui stabilise la voiture à une vitesse inférieure à 35 km/h. Le deuxième correspond à une accélération de 1.12 Volts, qui stabilise le véhicule à une vitesse d'environ 50 km/h.

L'utilisation des scénarios à consignes d'accélération constantes présente des inconvénients. Cependant, tenter d'asservir la vitesse, ou moduler l'accélération en fonction du temps, rend les interprétations encore plus incertaines.

5.4.2.2 Campagne d'injection de fautes

Nous avons retenu 20 expériences parmi celles que nous avons menées. Certaines n'étaient pas exploitables du fait d'incidents concernant soit le véhicule, soit Scanner, soit des interventions inopinées du conducteur. La moitié de ces expériences ont été effectuées avec le Safety-Bag actif et l'autre moitié avec une version du logiciel du CUBE qui n'effectue aucune intervention et n'émet aucune alarme. En fait, ce logiciel sans Safety-Bag est pour le reste, en tous points identiques au Safety-Bag et en particulier, ses états enregistrés dans les fichiers de données (logs) permettent d'identifier les instants d'injection de fautes.

Pour chaque nécessité de sécurité, nous présentons les fautes injectées correspondantes :

- Pour tester la nécessité de sécurité U1 : « *Le Safety-Bag doit vérifier la vivacité de l'application de contrôle-commande. Si celle-ci n'envoie plus de commande, il doit prévenir le conducteur et repasser en mode manuel et il arrête le véhicule en allumant les feux de détresse.* », nous introduisons la défaillance à 42 secondes en arrêtant l'envoi de trames de l'application de contrôle-commande au Safety-Bag.

- Pour tester la nécessité de sécurité A1 : « *Le Safety-Bag doit vérifier la circulation des trames sur le bus CAN. En cas d'absence de communication pendant une durée trop importante, le Safety-Bag doit alerter le conducteur et repasser en mode manuel.* », nous ne pouvons pas introduire la défaillance en modifiant

le scénario. La défaillance est injectée manuellement par l'expérimentateur en basculant l'interrupteur qui bloque la passerelle CAN qui communique les informations des capteurs véhicule à l'application de contrôle-commande et au Safety-Bag.

- Pour tester la nécessité de sécurité U4 : « *Le Safety-Bag doit vérifier le contrôle dynamique du véhicule : Il doit vérifier que l'accélération latérale reste inférieure à une valeur seuil. Dans le cas contraire, le Safety-Bag doit déclencher les alarmes et assurer la reprise en mode manuel.* », nous avons mené deux séries d'expériences en appliquant dans les deux cas une consigne d'accélération constante après la phase de démarrage, qui amène le système à violer les contraintes seuil de la nécessité de sécurité à plusieurs points lors du suivi de trajectoire. La première avec la consigne de 1.05 Volts (voir 5.28). La seconde avec une consigne de 1.12 Volts (voir figure 5.29).

- Pour tester la nécessité de sécurité H4 : « *Le Safety-Bag doit vérifier la vivacité de la trajectoire cinématique. Dans le cas contraire, le Safety-Bag doit déclencher les alarmes, inhiber l'accélération, freiner et maintenir la position d'angle volant après un certain temps.* », l'injection de faute consiste à tronquer la trajectoire pré-enregistrée et à arrêter de mettre à jour l'horodatage de la trajectoire cinématique envoyée au Safety-Bag quelques secondes avant la troncature.

  En effet, cette situation est représentative d'un gel dans la génération de la trajectoire cinématique : la dernière trajectoire générée, pouvant encore être suivie jusqu'à son extrémité.

5.4.2.3 Résultats et analyses

Pour chaque expérimentation, nous comparerons la courbe de la trajectoire, la courbe du profil de vitesse et la courbe des écarts à la trajectoire dans les deux cas sans et avec l'intervention du Safety-Bag.

Pour toutes les courbes, les trajectoires nominales et les trajectoires avant l'injection des fautes sont représentées avec un tracé discontinu.

a) Faute injectée : Application de contrôle-commande bloquée Sur les expérimentations que nous montrons, cette défaillance a lieu environ 240 mètres après le point de départ. Nous constatons sur la figure 5.21 que sans intervention du Safety-Bag, le véhicule sort complètement de la route.

Avec Safety-Bag, le véhicule s'arrête à proximité de la trajectoire nominale 5 secondes après la défaillance et la distance à la trajectoire reste entre 1.1 et 1.4 mètres.

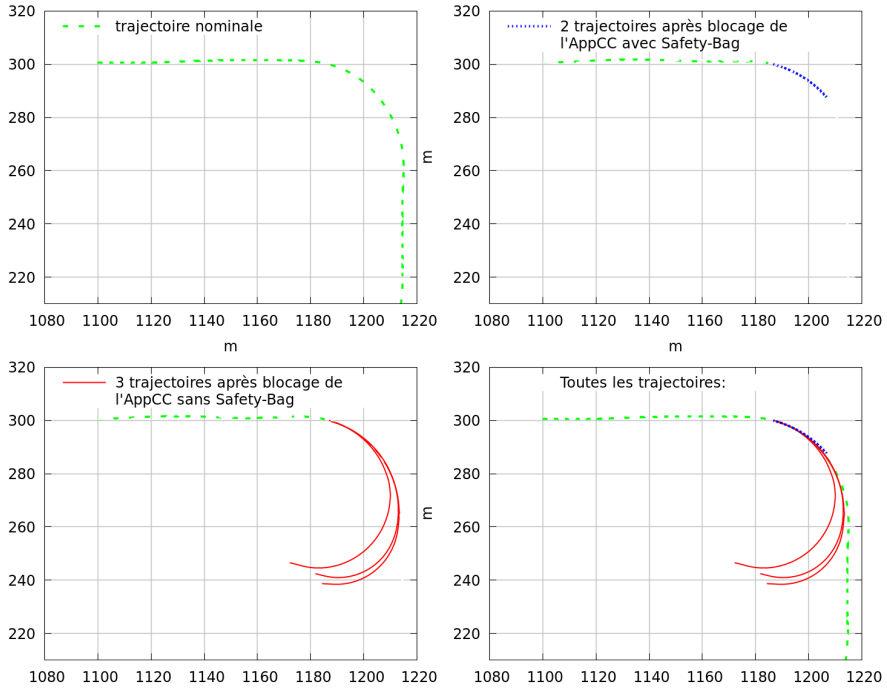

Figure 5.21 – Trajectoire du véhicule dans le cas de la défaillance : application de contrôle-commande bloquée

On voit sur la figure 5.22 que le Safety-Bag intervient en freinant le véhicule et l'arrête rapidement sur environ 25 mètres.

Sans Safety-Bag, le véhicule n'est pas freiné, la chute de vitesse est due d'une part au passage de la route goudronnée à l'herbe et d'autre part au braquage important de la direction.

Sur la figure 5.23, on observe plus précisément l'écart à la trajectoire dans ces différentes expérimentations. Sans Safety-Bag, pendant environ 3 secondes, le véhicule reste très près de la trajectoire puis il s'écarte de plus en plus, tandis qu'avec l'intervention du Safety-Bag, le véhicule s'arrête à moins de 1.5 mètres de la trajectoire de consigne.

Notons que dans cette expérience avec Safety-Bag, le véhicule s'arrête à la limite de la route, sur la voie de gauche. Cette intervention du Safety-Bag est pertinente et suffisante pour le véhicule expérimental, mais cette position d'arrêt serait évidemment dangereuse pour un véhicule commercial.

En effet, le Safety-Bag maintient la dernière position du volant avant le gel de l'application, ce qui permet de garder un virage ou une ligne droite pendant le

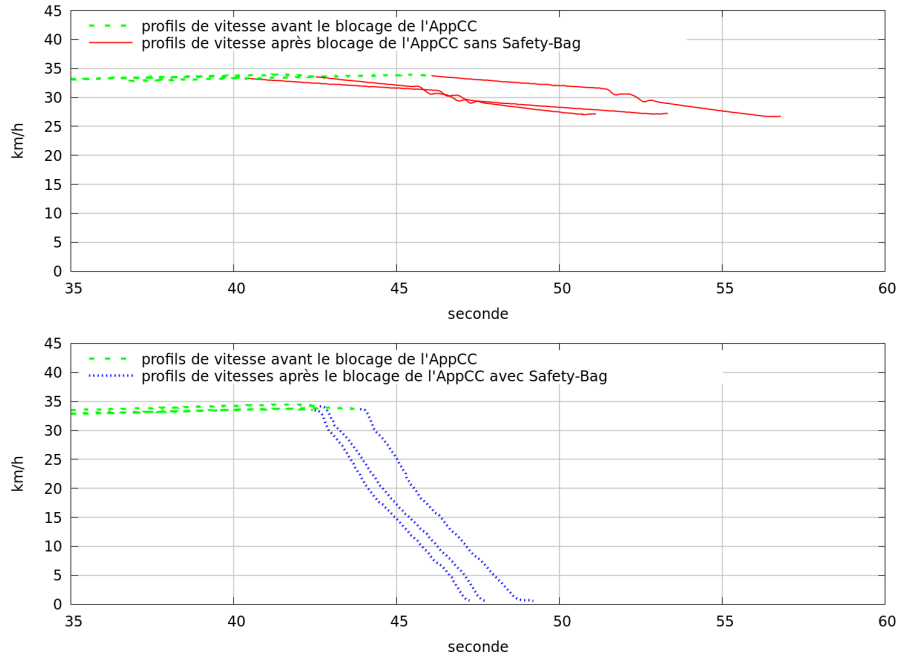

Figure 5.22 – Vitesse du véhicule en cas de défaillance : application de contrôle-commande bloquée

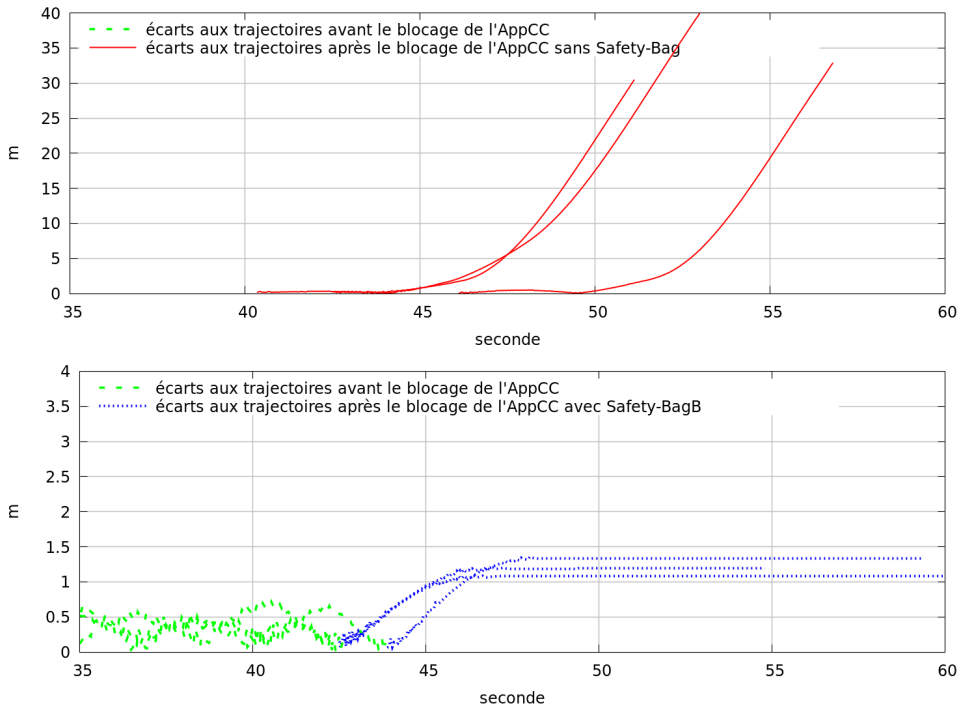

Figure 5.23 – Écart à la trajectoire du véhicule en cas de défaillance : application de contrôle-commande bloquée

freinage du véhicule. Dés que le conducteur reprend la main, cette consigne n'est plus maintenue.

Si l'orientation de la route change brutalement, l'action du Safety-Bag peut ne

plus être pertinente et la réaction du conducteur est d'autant plus nécessaire.

b) Faute injectée : CAN bloqué Les défaillances n'apparaissent pas exactement au même endroit ou moment, mais entre 210 et 340 mètres, soit entre 36 et 46 secondes à cause du déclenchement manuel de l'injection de fautes.

Sans Safety-Bag, le véhicule se met à zigzaguer sur la route car l'application de contrôle-commande ne connait plus la position d'angle volant et réagit seulement en fonction de son écart à la trajectoire par rapport au dernier angle au volant connu.

Nous voyons également dans la figure 5.24 qu'avec l'intervention du Safety-Bag le véhicule s'arrête à proximité de la trajectoire.

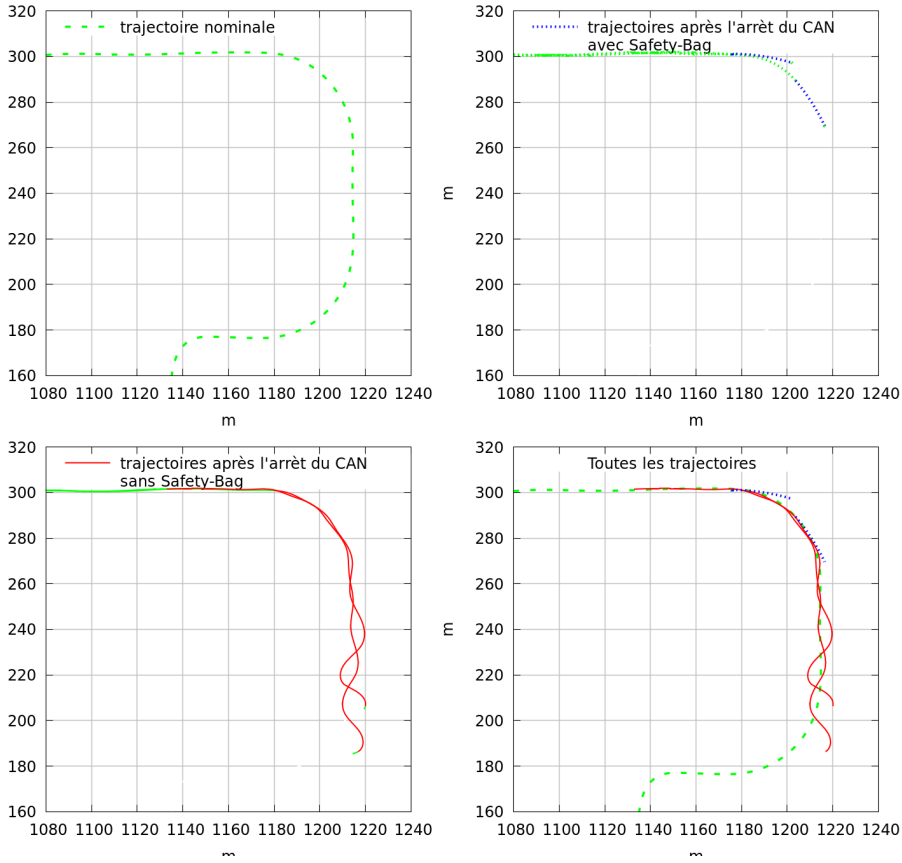

Figure 5.24 – Trajectoire du véhicule en cas de défaillance : CAN bloqué

Dans la figure 5.25, nous pouvons observer que sans Safety-Bag, le véhicule ne s'arrête pas, la vitesse diminue seulement du fait des oscillations de plus en plus brutales du véhicule que nous voyons sur la figure 5.26.

Avec Safety-Bag, dans le premier cas, le Safety-Bag détecte la défaillance vers 44 secondes, et arrête le véhicule rapidement vers 48.7 secondes. Par contre dans le deuxième cas, le Safety-Bag détecte la défaillance vers 46.2 secondes, et arrête le véhicule vers 50.7 secondes.

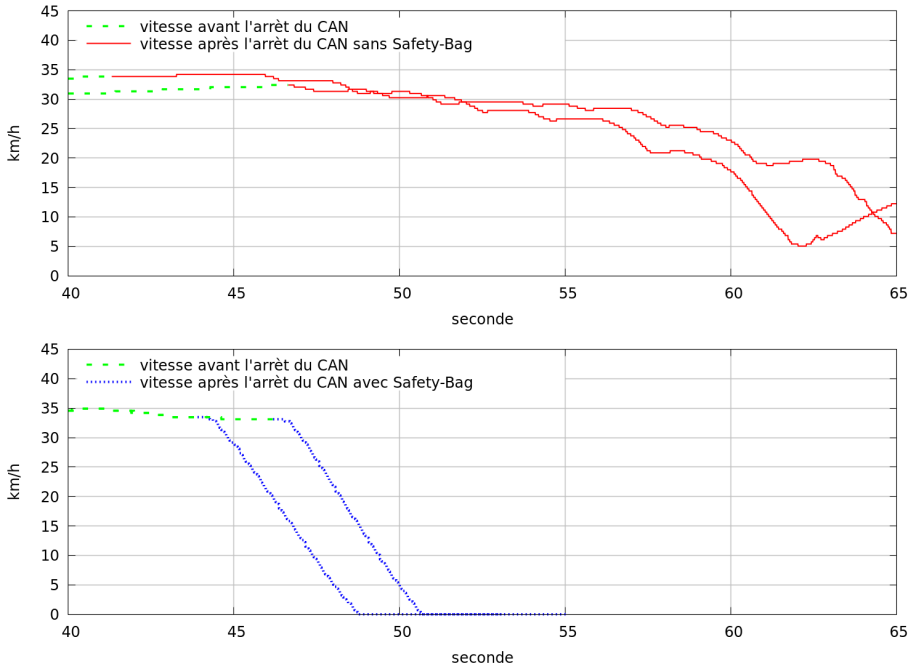

Figure 5.25 – Vitesse du véhicule en cas de défaillance : CAN bloqué

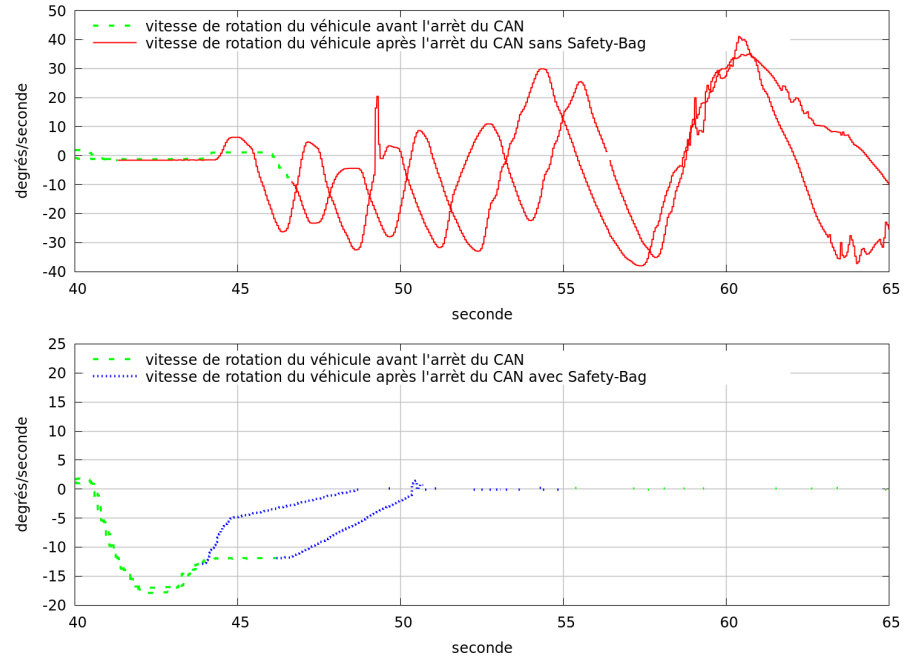

Figure 5.26 – Vitesse de rotation du véhicule en cas de défaillance : CAN bloqué

Dans la figure 5.26, nous constatons que sans Safety-Bag, les vitesses de rotation du véhicule sont très importantes et non seulement sont inconfortables, mais créent une situation dans laquelle la reprise de contrôle en mode manuel est difficile.

Sur la figure 5.27, l'écart à la trajectoire sans le Safety-Bag augmente de plus en plus, dépasse 3 mètres, 5 mètres puis 10 mètres, etc. Dans le véhicule, le conducteur

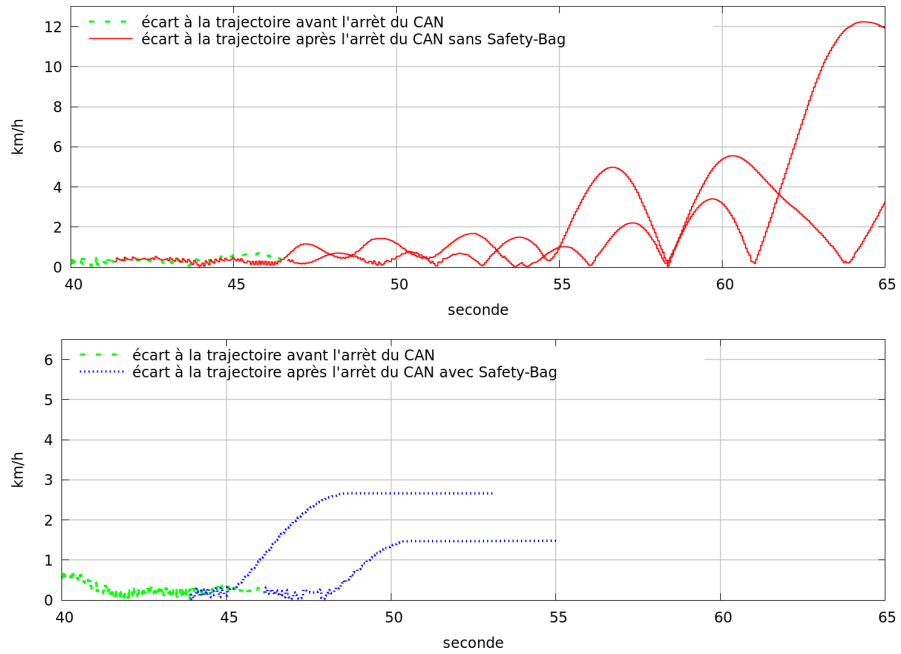

Figure 5.27 – Écart à la trajectoire du véhicule en cas de défaillance : CAN bloqué

constate que le volant va jusqu'à la butée, dans un sens puis dans l'autre.

Par contre quand le Safety-Bag intervient, le volant n'est pas maintenu par la direction assistée électronique et le conducteur peut reprendre facilement le contrôle du volant.

Contrairement à l'expérimentation précédente, l'intervention de sécurité ne maintient pas ici la commande sur l'angle au volant, mais la ramène progressivement à zéro. Pour cette raison, le véhicule s'écarte davantage de la trajectoire jusqu'à 3 mètres, car il ne suit pas la courbe du virage. Dans le monde virtuel Scanner, le véhicule s'arrête ainsi au milieu de la voie de gauche, ce qui ne serait évidemment pas acceptable pour un véhicule commercial.

c) Faute injectée : Vitesse en virage excessive Nous présentons deux séries d'expériences dans cette partie : un suivi de la trajectoire avec une consigne d'accélération de 1.05 Volts, puis un autre avec une consigne de 1.12 Volts. Ces deux séries montrent les résultats du comportement du véhicule avec et sans Safety-Bag.

Pour ces expérimentations, les courbes de vitesse et d'écart à la trajectoire sont présentées par rapport à la distance parcourue le long de la trajectoire de consigne et non par rapport au temps afin de mieux mettre en évidence le comportement du véhicule lors du passages des différents virages.

Nous constatons sur la figure 5.28 que dans le cas d'une accélération à 1.05 Volts, la vitesse est du même ordre de grandeur avec ou sans Safety-Bag. Nous

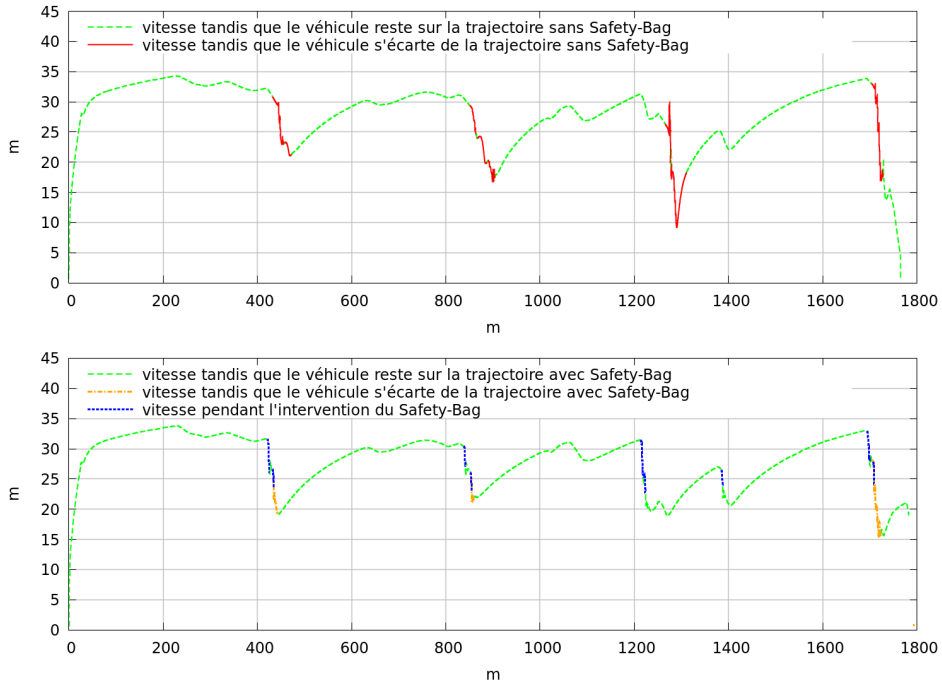

Figure 5.28 – Vitesse du véhicule en cas de défaillance : vitesse en virage excessive (expérience avec consigne d'accélération = 1.05 Volts)

pouvons remarquer cependant qu'on est à la limite d'activation de la condition de déclenchement puisque le Safety-Bag intervient régulièrement (courbe en pointillé).

D'ailleurs, la figure 5.30 montre que sans Safety-Bag le véhicule effectue des sorties de route dans les virages N° 3, 6, 9 et 11 (courbe en continu), tandis qu'avec Safety-Bag, le véhicule tient mieux la route, touchant cependant le trottoir dans les virages N° 3 et 6 et sortant vraiment de la route seulement dans le virage N° 11 (courbe jaune tiret-point).

Par contre, avec la consigne d'accélération 1.12 Volts, l'impact du Safety-Bag est évident. On peut d'ailleurs voir sur la figure 5.31, sans intervention du Safety-Bag, que le véhicule est incapable de suivre la trajectoire. Avec l'intervention du Safety-Bag, nous constatons deux sorties de route significatives dans les virages 3 et 11 et une perte temporaire de contrôle du véhicule après le virage N° 6.

Les écarts à la trajectoire avec la consigne d'accélération à 1.05 Volts (5.32) confirment ces anomalies. Dans le premier cas sans l'intervention du Safety-Bag, les écarts à la trajectoire sont significatifs, dépassant plusieurs fois 2 mètres et atteignant 7 mètres dans le dernier virage. Par contre avec l'intervention du Safety-Bag, les écarts à la trajectoires ne dépassent pas les 2 mètres sauf dans le dernier virage (5 mètres).

Ce dernier virage est un tourne-à-gauche dans un croisement à angle droit après une grande ligne droite. Le véhicule atteint sa vitesse maximale stable (de 33

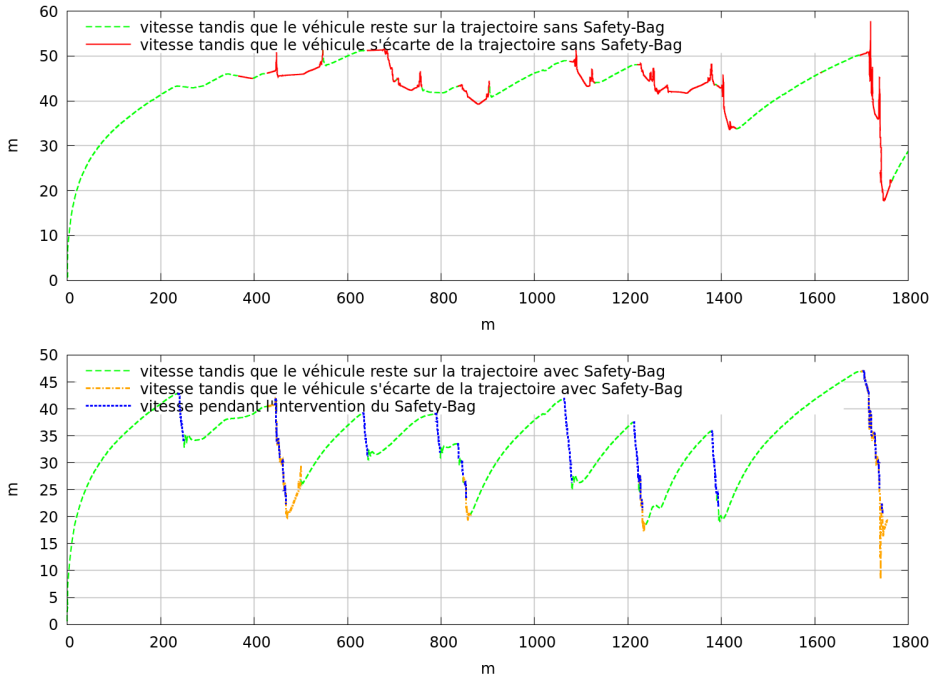

Figure 5.29 – Vitesse du véhicule en cas de défaillance : vitesse en virage excessive (expérience avec consigne d'accélération = 1.12 Volts)

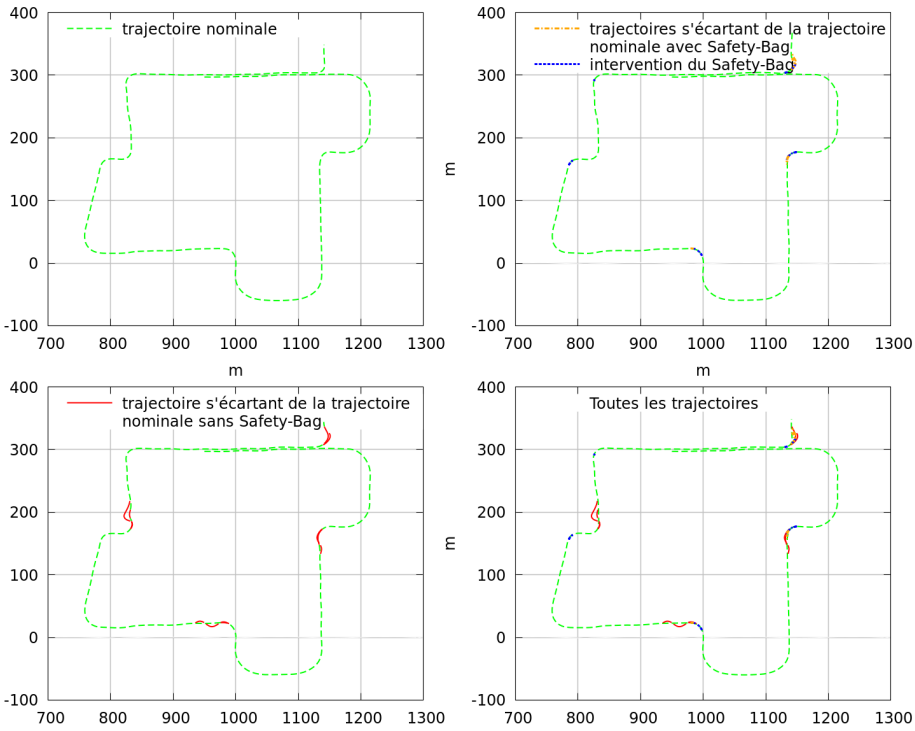

Figure 5.30 – Trajectoire du véhicule en cas de défaillance : vitesse en virage excessive (expérience avec consigne d'accélération = 1.05 Volts)

km/h). La direction assistée ne réagit pas rapidement à la commande du couple. De plus, l'intervention du Safety-Bag n'a lieu que lorsque le volant a déjà tourné

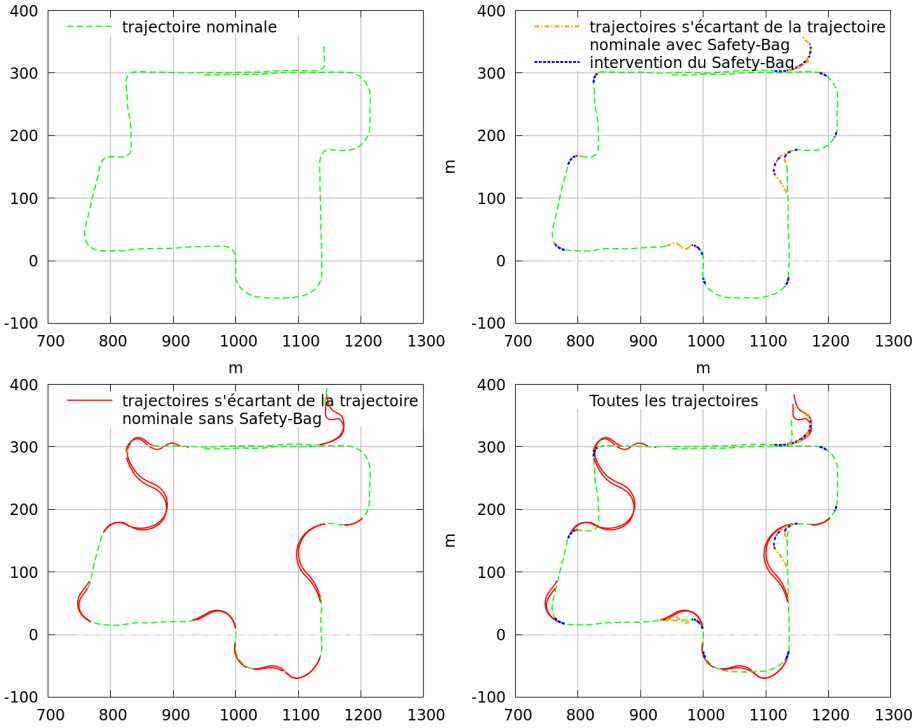

Figure 5.31 – Trajectoire du véhicule en cas de défaillance : vitesse en virage excessive (expérience avec consigne d'accélération = 1.12 Volts)

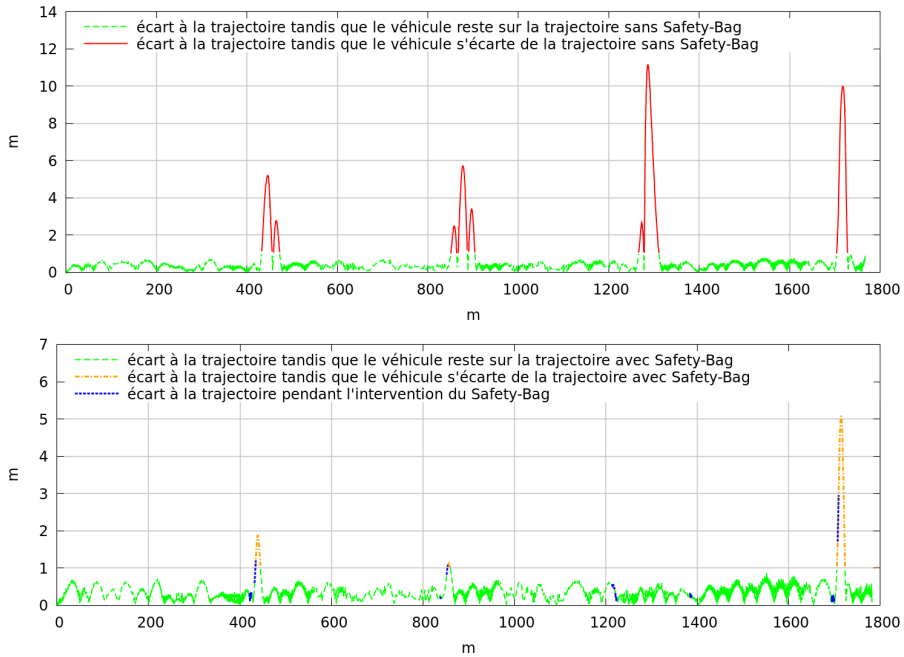

Figure 5.32 – Écart à la trajectoire du véhicule en cas de défaillance : vitesse en virage excessive (expérience avec consigne d'accélération = 1.05 Volts)

significativement : il est en fait déjà trop tard pour reprendre correctement le virage.

Les écarts à la trajectoire avec la consigne d'accélération 1.12 Volts (figure 5.33)

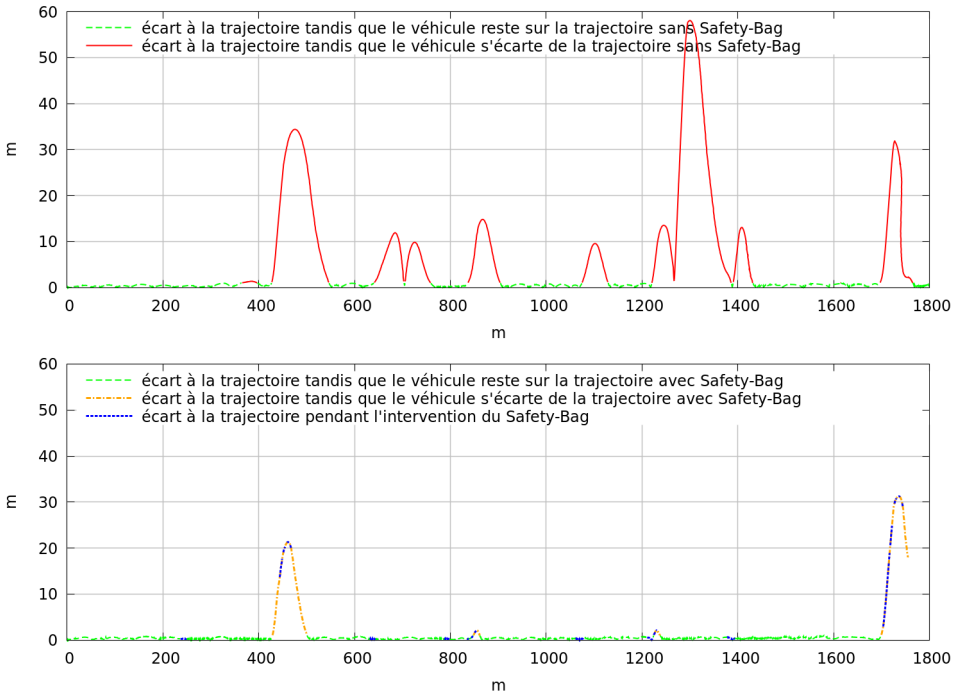

Figure 5.33 – Écart de la trajectoire du véhicule en cas de défaillance : vitesse en virage excessive (expérience avec consigne d'accélération = 1.12 Volts)

confirment l'importance des déviations sans Safety-Bag. Le Safety-Bag permet de réduire significativement ces déviations et d'éviter plusieurs sorties de route, se révélant particulièrement efficace dans les grandes courbes N° 5, 6, 7 et 8. Par contre, dans les virages serrés (N° 3 et 11), le Safety-Bag ne parvient pas à éviter la sortie de route.

Pour cette nécessité de sécurité, l'efficacité de l'intervention du Safety-Bag est limitée par le fait qu'il ne peut pas anticiper la trajectoire que compte suivre l'application de contrôle-commande et l'accélération qu'elle compte effectuer. Ceci est aggravé par la lenteur de la réponse du volant à la commande de couple envoyé par le Safety-Bag. Le Safety-Bag réduit cependant significativement les conséquences des fautes injectées par rapport à un système sans Safety-Bag. Les alarmes qu'il soulève permettent également une reprise en main plus rapide par le conducteur, qui n'est volontairement pas intervenu dans ces différentes expériences.

d) Faute injectée : Trajectoire cinématique non mise à jour Pour des raisons liées à l'injection de faute réalisée (arrêt de l'horodatage de la trajectoire désirée), nous n'avons pas la courbe d'écarts à la trajectoire pour cette expérience. Nous présentons par contre la courbe de vitesse du véhicule par rapport au temps et par rapport à la distance parcourue.

Sur la figure 5.34, on voit que dans les deux expériences avec Safety-Bag,

le véhicule s'arrête à 1.2 mètres de la trajectoire de consigne (notée trajectoire nominale en vert discontinu dans les figures). Par contre, sans Safety-Bag, le véhicule suit la trajectoire de consigne le plus longtemps possible, puis tourne brutalement et se dirige vers une destination indéterminée. Ce comportement est dû à la façon de programmer le générateur de trajectoire (ou le suiveur de trajectoire dans notre véhicule) et d'autres comportements auraient pu être observés, comme éventuellement la rotation autour de la dernière coordonnée donnée.

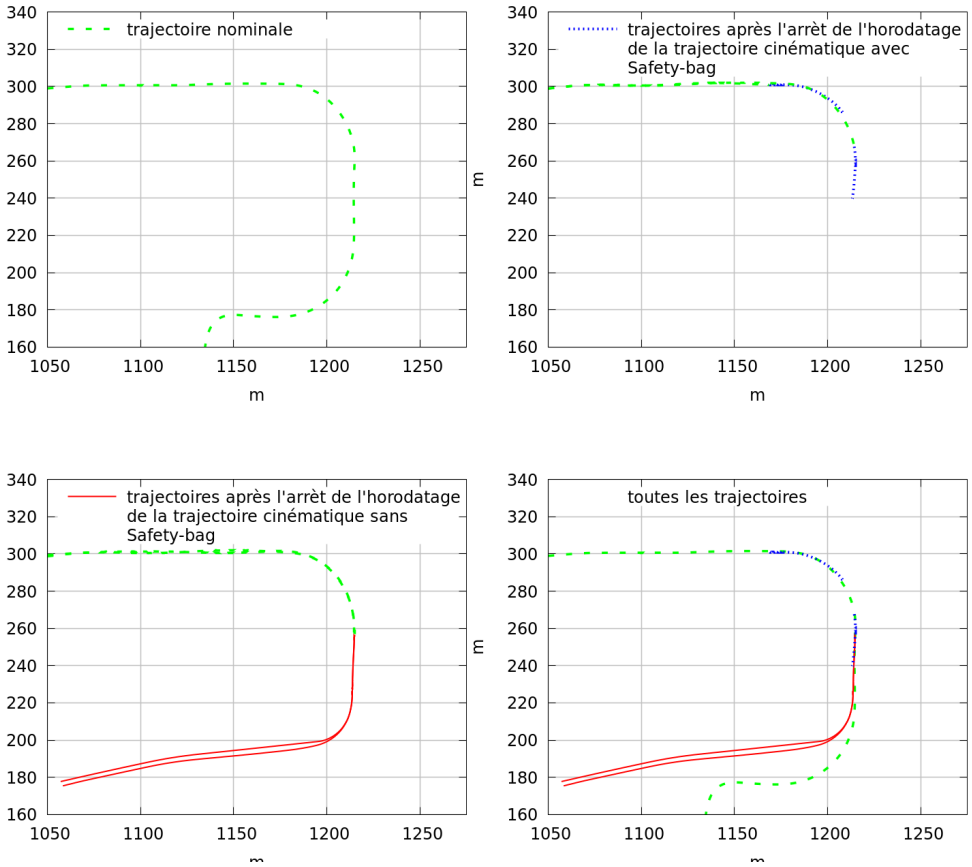

Figure 5.34 – Trajectoire du véhicule en cas de défaillance : Trajectoire cinématique bloquée

Sur la figure 5.36, on voit que le Safety-Bag freine le véhicule dès qu'il a constaté que la trajectoire cinématique n'est plus mise à jour et il arrête le véhicule en 7 secondes pour la première expérience et en 5.5 secondes pour la deuxième expérience (la vitesse lors de début de freinage de cette deuxième expérience est plus faible). Sans l'intervention du Safety-Bag, la vitesse est maintenue alors que le véhicule est complètement sorti de la route.

Sur la figure 5.35 qui présente l'évolution de la vitesse par rapport à la distance, nous avons fait apparaitre les deux phases d'intervention du Safety-Bag. Dans

la première phase (en continu), le Safety-Bag applique les consignes volant de l'application de contrôle-commande et donc le véhicule suit la trajectoire pendant le Safety-Bag le freine. Cette phase dure 1.6 secondes.

Dans la seconde phase (courbe en pointillé après la phase représentée en trait continu), l'angle volant est maintenu par le Safety-Bag pendant la fin du freinage.

Suivre la trajectoire lors de la première phase a certainement aidé le véhicule à rester au final sur la route.

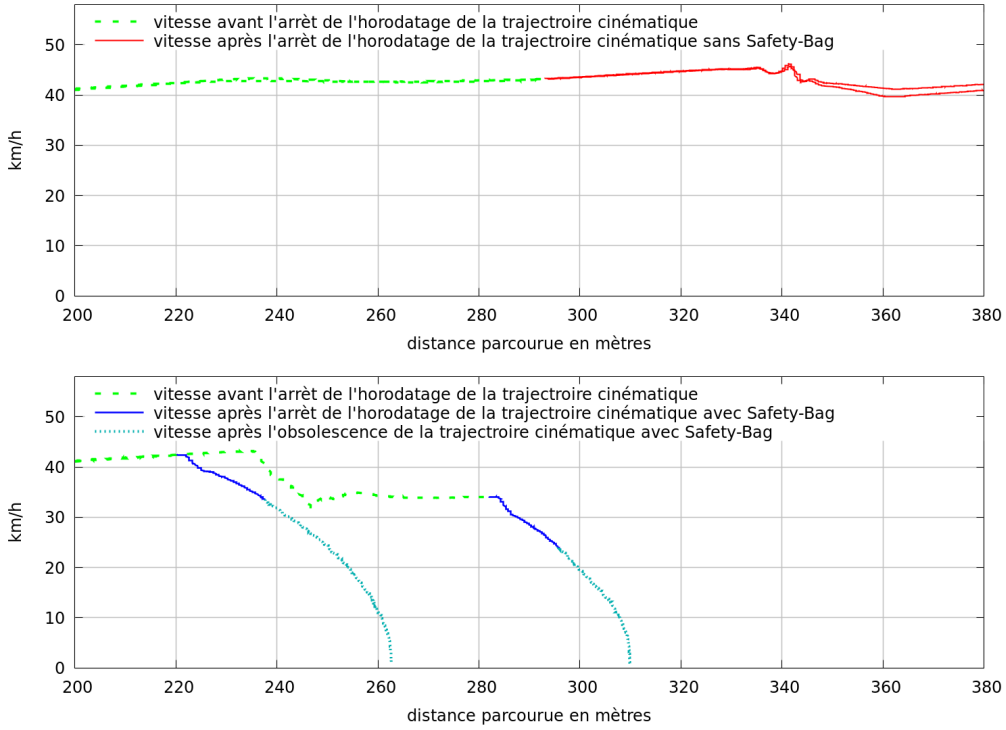

Figure 5.35 – Vitesse du véhicule (en fonction de la distance parcourue) en cas de défaillance : Trajectoire cinématique bloquée

Discussion

Les résultats que nous avons trouvés dans les deux sections 5.4.1 et 5.4.2 montrent que le Safety-Bag réduit significativement les risques et donc améliore considérablement le comportement du système dans beaucoup des cas étudiés. En effet, le Safety-Bag freine et arrête dans certains cas le véhicule en évitant qu'il ne sorte complètement de la route même si ce dernier s'arrête au milieu de la route (comme le cas pour l'application de contrôle-commande bloquée).

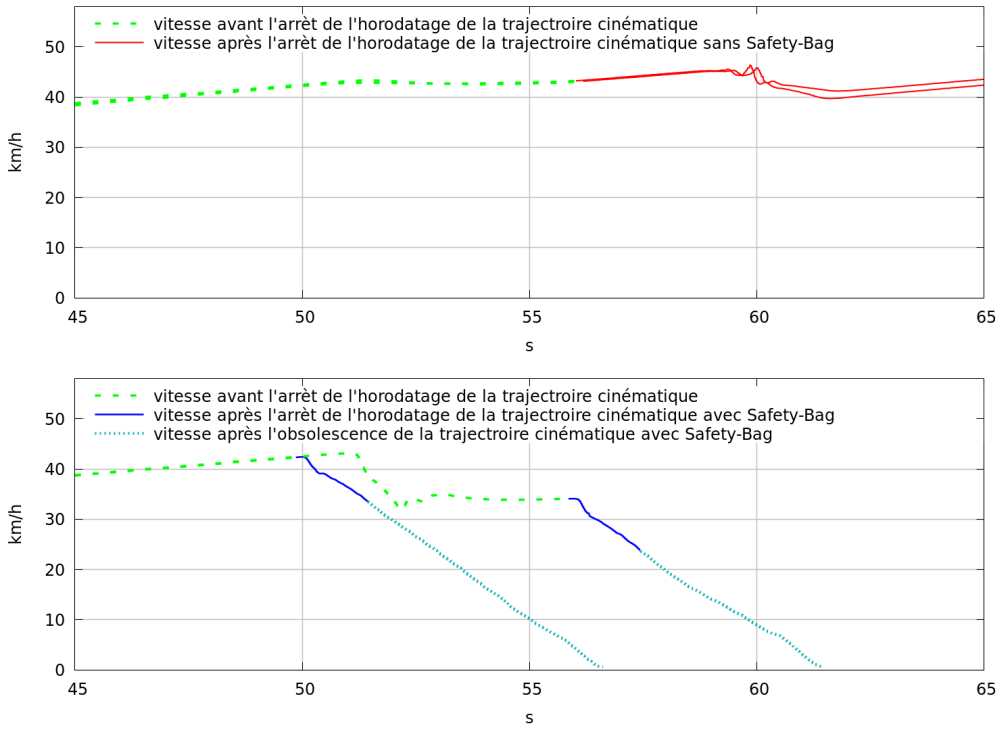

Figure 5.36 – Vitesse du véhicule (en fonction du temps) en cas de défaillance : Trajectoire cinématique bloquée

La reprise en main par un conducteur, non considérée dans nos expériences, améliorerait encore le rétablissement dans de nombreux cas, notamment en ramenant le véhicule dans la bonne file et/ou en l'arrêtant sur le bord de la route. Il est également nécessaire de reprendre le véhicule en main pour les nombreuses nécessités de sécurité qui interdisent le retour en un mode de conduite autonome pour des raisons de sécurité-innocuité. Néanmoins, nos expériences montrent que notre Safety-Bag a réussi à augmenter considérablement la sécurité-innocuité dans notre système.

Nous n'avons pas testé de scénarios visant à évaluer l'activation simultanée de plusieurs nécessités de sécurité. Le statechart du Safety-Bag (figure 2.14) montre que dans ce cas l'intervention liée à l'état le plus grave serait appliquée.

5.5 Expérimentations d'interactions homme/machine sur la piste Seville

Nous avons effectué des expériences sur la piste réelle Seville. L'objectif principal était d'évaluer l'impact des systèmes d'alarme du Safety-Bag sur le conducteur en mesurant son temps de réaction avec et sans Safety-Bag. Pour ces tests, nous avons demandé à différents conducteurs de réagir quand le comportement du véhicule

s'éloignait d'un scénario nominal qu'on leur avait décrit. Ils ont d'abord expérimenté le fonctionnement nominal sans injection de faute, puis ont expérimenté la procédure de reprise en conduite manuelle en actionnant le bouton d'arrêt de process. Nous leur avons précisé qu'une alarme pouvait retentir ou ne pas retentir en cas d'anomalies.

Nous avons mis en place des nécessités de sécurité spécifiques aux fautes que nous avons injectées : en effet, ces fautes sont bénignes, pour éviter tout risque au conducteur, et nous avons donc dû définir des nécessités de sécurité spécifiques pour qu'elles puissent déclencher les alarmes du Safety-Bag.

Pendant l'expérience, l'accélération et le freinage sont contrôlés de façon autonome par le véhicule. Par contre, les conducteurs maintiennent le contrôle du volant de direction. Nous pouvons noter que dans cette situation, la durée très courte de chaque scénario et la fréquence des scénarios nécessitant l'intervention du conducteur favorisent un haut niveau de vigilance.

5.5.1 Scénario nominal

Chaque conducteur conduit la voiture pendant presque 15 expérimentations sur une ligne droite du circuit Seville. Chaque expérimentation suit un scénario nominal simple et court : *la voiture accélère pendant 5 secondes jusqu'à 20 km/h, puis elle maintient sa vitesse pendant 5 secondes et enfin elle freine et s'arrête de façon autonome.*

Les conducteurs doivent réagir si la situation réelle s'écarte de ce scénario nominal en activant le bouton d'arrêt de process et en arrêtant le véhicule. Bien que le contrôle latéral du véhicule soit donné au conducteur, le véhicule est en ligne droite et aucune action spécifique n'est nécessaire sur cet aspect. De plus, pour éviter tout danger, aucune faute n'est injectée sur le comportement latéral du véhicule.

5.5.2 Campagne d'injection de fautes

Le scénario nominal est présenté aux conducteurs, mais ils sont prévenus à l'avance que des défaillances peuvent se produire et que ces défaillances se traduiront par le non-respect de ce scénario. Ces défaillances sont détectées par le Safety-Bag (qui émet alors une alerte sonore) quand il est activé. Avec ou sans activation du Safety-Bag, le conducteur doit réagir à une déviation du scénario en appuyant sur le bouton d'arrêt de process puis arrêter manuellement le véhicule.

Nous avons injecté les fautes suivantes :

- Scénario « *Acc_ sans_ alerte* » : Le véhicule maintient une accélération pendant 10 secondes au lieu de 5 secondes.

Scénarios déviants	**Comportement du véhicule, sans intervention du Safety-Bag**	**Comportement du véhicule, avec intervention du Safety-Bag**
Accélération plus longue et vitesse plus élevée	Le véhicule accélère plus de 5 secondes et dépasse 20 km/h	Dès que le scénario s'écarte du scénario nominal, l'alarme sonne et le véhicule ralentit et s'arrête.
Maintien la vitesse à 20 km/h	Le véhicule accélère jusqu'à 20 km/h puis maintient sa vitesse.	
Le véhicule redémarre après l'arrêt	Le véhicule suit le scénario nominal jusqu'à l'arrêt, puis redémarre.	Le véhicule suit le scénario nominal, puis l'alarme sonne alors que le véhicule est arrêté. Le conducteur doit interrompre le mode autonome en appuyant sur le bouton d'arrêt de process pour éviter toute défaillance ultérieure.

Tableau 5.12 – Scénarios déviants.

- Scénario « *Vitesse_ maintenue* » : Le véhicule conserve sa vitesse durant 10 secondes au lieu de 5 secondes.
- Scénario « *Redémarre* » : Le véhicule redémarre une demi seconde après son arrêt. En cas d'intervention du Safety-Bag, le véhicule ne démarre pas, les alarmes sonnent. Puisque le véhicule est en mode autonome et que l'alarme sonne, le conducteur doit interrompre le mode autonome en appuyant sur le bouton d'arrêt de process pour éviter toute défaillance ultérieure.

Dans le tableau 5.12, nous précisons le comportement du véhicule sans et avec l'intervention du Safety-Bag pour chacun des scénarios déviants.

5.5.3 Résultats et analyses

Les temps de réaction des conducteurs sont mesurés en utilisant les fichiers logs du Safety-Bag Rules Checker, dans lesquels les évènements sont datés avec une grande précision, en particulier l'activation du bouton d'arrêt de process.

A travers ces tests, nous avons pu comparer les comportements des conducteurs avec et sans intervention du Safety-Bag. Cette analyse comporte une comparaison quantitative avec la mesure de temps de réaction et une comparaison qualitative dans laquelle les conducteurs donnent leurs avis après les expériences.

Les quatre conducteurs du groupe de test appartiennent à l'équipe technique de notre laboratoire. Ils ont une connaissance générale des véhicules automones, mais, aucun n'a jamais conduit le véhicule avec le Safety-Bag, de sorte qu'ils ne sont pas formés pour réagir à ses interventions.

Dans ce contexte, sans intervention du Safety-Bag et pour 25 % des comportements incorrects, les conducteurs ne réagissent pas. Dans les autres cas, ils réagissent

Temps de réaction du conducteur	Temps de réaction moyen	Déviation standard	Min	Max
Sans Safety-Bag	4.1	1.4	2	6
Avec Safety-Bag	2.2	1.3	0.7	5

Tableau 5.13 – Temps de réaction moyen sans et avec Safety-Bag.

dans un délai d'environ 4 secondes. Ils expliquent que pour eux il n'y a pas de risque immédiat.

Néanmoins, si le Safety-Bag intervient, il détecte la défaillance et transmet des signaux d'alarmes au conducteur lui permettant de réagir plus rapidement en moyenne en 2 secondes, comme indiqué dans le tableau 5.13.

Notez qu'avec le Safety-Bag, le temps de réaction maximal est de 5 secondes. Dans ce cas, le conducteur avait en fait freiné et arrêté le véhicule avant d'appuyer sur le bouton d'arrêt de process.

Les conducteurs disent qu'il est difficile de décider quand ils doivent récupérer le contrôle sans le signal explicite alors que les alarmes du Safety-Bag leur donnent un signal clair auquel il est plus facile de réagir.

Bien qu'élémentaires, ces expériences montrent pour nous l'intérêt des alarmes du Safety-Bag en complément de ses autres interventions de sécurité : en envoyant un message clair au conducteur qu'un problème survient, il diminue significativement son temps de réaction.

Si nous avions disposé de plus de temps dans la thèse, nous aurions pu reproduire ces expériences sur le banc expérimental VILAD avec des scénarios plus complexes et dans des situations plus proches des conditions de conduite réelles.

5.6 Conclusion

Nous avons présenté dans cette section plusieurs analyses et expériences ayant pour but de montrer l'apport du composant Safety-Bag sur la sécurité-innocuité de notre véhicule autonome expérimental. En premier lieu, l'analyse de risques AMDEC introduisant le composant Safety-Bag montre que ce dernier permet d'améliorer significativement la sécurité-innocuité des véhicules autonomes. En effet, ce dispositif permet de réduire la gravité d'un grand nombre des défaillances catastrophiques identifiées. Grâce à ses redondances, son introduction n'ajoute que peu de risques de défaillance.

De plus, l'enregistrement des changements d'états du Safety-Bag fournit un outil précieux de diagnostic des incidents de sécurité, offrant des fonctionnalités proches de la boîte noire d'un avion.

Certaines exigences de sécurité implémentables par le Safety-Bag ne sont pas implémentées dans notre véhicule, soit parce que l'investissement n'a pas été effectué (redondance des capteurs par exemple), soit du fait des limitations des entrées/sorties de calculateurs que nous avons choisis (le CUBE ne dispose que de 8 sorties analogiques, l'IGEP n'a ni entrée ni sortie analogique).

Notre Safety-Bag est développé d'abord pour un véhicule autonome expérimental. En cas d'anomalie de ce Safety-Bag, on assure autant que possible la mise en sécurité du véhicule sans chercher à conserver la disponibilité éventuellement dans un mode dégradé.

Les expérimentations que nous avons menées confirment l'intérêt du Safety-Bag et l'efficacité de ses interventions. Cependant sans intervention du conducteur, le Safety-Bag diminue grandement les risques de mauvais comportement, en s'arrêtant rapidement, mais sans garantie d'être dans un état sûr : sans reprise en main du conducteur, il s'arrête généralement au milieu de la route, parfois même sur la voie de gauche. Ce composant est pour nous un premier pas indispensable pour augmenter la sécurité-innocuité des véhicules autonomes. Il n'est cependant pas suffisant du fait de la simplicité de ses règles, nécessaires pour permettre une grande confiance dans leur validation.

Conclusion et perspectives

Sommaire

6.1 Démarche suivie et leçons apprises

La méthodologie présentée dans le cadre de cette thèse propose de mettre en place un système de sécurité-innocuité *Safety-Bag*, chargé de filtrer des commandes et capable de surveiller en ligne un ensemble de nécessités de sécurité dépendantes de l'application. Si une de ces nécessités de sécurité est violée, le Safety-Bag intervient généralement en combinant des inhibitions et des actions de sécurité. La définition de ces nécessités est donc fondamentale dans le comportement du composant Safety-Bag, et la sécurité-innocuité qu'il apporte à son système.

Pour définir les nécessités de sécurité, nous proposons une méthode basée sur l'utilisation de méthodes d'analyse de risques diversifiées pour obtenir des ensembles d'exigences de sécurité portant sur le système. Une étude de ces exigences de sécurité, basée sur les capacités de perception et d'action possibles par le Safety-Bag, permet de déterminer lesquelles peuvent être traduites en nécessités de sécurité. Dans ce mémoire, nous avons utilisé les deux méthodes ascendantes d'analyse de risques AMDEC et HazOp-UML. Ces deux méthodes ont globalement le même objectif d'identifier respectivement les modes de défaillances et des dangers possibles (les déviations) du système afin de les réduire par des techniques de sûreté de fonctionnement.

En analysant ces deux méthodes, nous avons conclu que l'AMDEC et l'HazOp-UML sont complémentaires. Cela peut être expliqué par le fait que l'AMDEC met l'accent sur les composants internes matériels et logiciels du système, alors que l'HazOp-UML se concentre principalement sur le processus de conduite et sur les

composants de l'environnement routier. De ce fait, certaines exigences de sécurité apparaissent dans l'une et pas dans l'autre et inversement. De plus, nous avons trouvé des exigences de sécurité similaires entres ces deux techniques, qui aboutissent à des vérifications équivalentes (la vivacité ou la cohérence temporelle de l'application de contrôle-commande par exemple).

Il faut noter que les résultats de l'analyse HazOp-UML peuvent fortement varier selon la façon dont sont choisis et exprimés les cas d'utilisation, leurs attributs et leurs déviations. Cela est probablement dû au fort lien existant entre l'approche HazOp-UML et le processus de conception, pour lequel de nombreux résultats différents mais également corrects peuvent être envisagés.

Après avoir identifié les exigences de sécurité en analysant les deux méthodes AMDEC et HazOp-UML, nous avons élicité les nécessités de sécurité dans le cas où l'exigence de sécurité était implémentable par le Safety-Bag. Chaque nécessité de sécurité est la combinaison d'une condition de déclenchement de sécurité et d'une intervention de sécurité.

Certaines exigences de sécurité portant sur des questions complexes et de haut niveau d'abstraction (détection des trajectoires dangereuses par exemple) ne peuvent pas être vérifiées par notre Safety-Bag car nécessitent des mécanismes complexes (mécanismes de fusion de données ou décisionnels). En effet, le Safety-Bag doit rester un composant suffisamment simple pour être validé facilement (par tests, analyse de code ou méthodes formelles) puisqu'il doit garantir des règles de sécurité-innocuité pour le système. Dans le cas de véhicules autonomes, il ne peut pas fournir de fonctionnalités complexes comme la détection d'obstacles, la localisation, l'identification de situations et de conditions environnementales, ou le calcul de trajectoire ou d'espace navigable.

Pour remédier à certains de ces verrous technologiques, nous pourrions introduire dans certains cas un composant logiciel diversifié de l'application de contrôle-commande, pouvant effectuer certaines vérifications trop complexes à mettre en œuvre par le Safety-Bag.

Pour pouvoir fixer les conditions de déclenchement de sécurité, nous avons déterminé les valeurs de danger, qui représentent des valeurs frontières de l'état catastrophique provoquant la défaillance du système, et les marges de sécurité, qui représentent l'écart entre ces valeurs frontières et le comportement considéré sûr du véhicule. La détermination des valeurs de danger et des marges de sécurité est basée principalement sur le jugement d'expert ou sur des valeurs empiriques justifiées par des arguments et confirmées par des expérimentations (tel est le cas de la vivacité de l'application de contrôle-commande ou la vivacité de l'état cinématique).

De plus, nous avons associé à chaque nécessité de sécurité les interventions

de sécurité (inhibitions et actions) possibles imposées par le Safety-Bag afin de remettre le système dans un état sûr. Généralement le Safety-Bag combine les deux types d'intervention de sécurité pour faire le rétablissement. Il déclenche ainsi les alarmes, inhibe l'accélération et force une action de sécurité soit en freinant, soit en maintenant l'angle volant. Il faut noter que dans de nombreux cas la reprise en main du véhicule par un conducteur est importante pour la sécurité-innocuité du système, du fait que le Safety-Bag seul ne peut généralement qu'amener le véhicule à s'arrêter, sans garantir sa position sur la chaussée. De nombreux verrous technologiques sont encore à lever de notre point de vue avant de pouvoir disposer de véhicules autonomes de niveau d'autonomie 4 et 5, cibles actuels de l'industrie automobile.

En raison de contraintes de temps et à cause de la non disponibilité de certains capteurs dans notre véhicule, nous n'avons testé que 6 nécessités de sécurité définies par les deux méthodes d'analyse de risques AMDEC et HazOp-UML. Les résultats de nos expérimentations effectuées sur le banc VILAD et sur la piste réelle Seville montrent que le Safety-Bag réduit significativement les risques et donc améliore considérablement le comportement du système. En effet, le Safety-Bag déclenche les alarmes, freine et arrête dans certains cas le véhicule en lui évitant de sortir complètement de la route. Il faut noter que notre Safety-Bag est conçu pour un véhicule autonome expérimental et que son but principal est d'augmenter la sécurité-innocuité même si cela dégrade significativement la disponibilité du système. Les expérimentations effectuées montrent l'efficacité des interventions du Safety-Bag et surtout sa rapidité pour alerter le conducteur. La vitesse, l'énergie et donc la gravité des défaillances potentielles sont ainsi significativement réduites. Nous pouvons dire que le Safety-Bag est une solution satisfaisante pour des véhicules autonomes expérimentaux avec un conducteur vigilant.

Ces travaux ont donné lieu à plusieurs publications, notamment dans la conférence internationale EDCC 2018 à Iasi en Roumanie [Brini et al., 2018], et dans plusieurs événements nationaux comme la conférence QUALITA 2017 [Brini et al., 2017] à Bourges et plusieurs Workshops ([Brini et al., 2016], etc.).

6.2 Perspectives

Nos travaux de recherche s'inscrivent dans le contexte de la vérification en ligne des nécessités de sécurité par le Safety-Bag. Lors de ces travaux, nous avons identifié la possibilité de les étendre et les compléter des manières suivantes :

- A court terme :
  - ◇ Nous pourrons tester notre composant Safety-Bag et nos nécessités de sécurité sur une réelle application de contrôle-commande plutôt qu'un joueur de scénario.
  - ◇ Nous pourrons tester et valider les nécessités de sécurité qui restent en rajoutant les capteurs et mécanismes nécessaires sur le véhicule.
  - ◇ Nous pourrons détailler d'autres cas d'utilisation afin d'éliciter des nécessités de sécurité que nous trouverons peut-être intéressantes à mettre en place dans le Safety-Bag.
- A long terme :
  - ◇ Pour les exigences de sécurité non implémentables par le Safety-Bag, nous pourrons chercher et étudier en détails d'autres solutions adéquates, comme par exemple un composant diversifié de mise en état sûr par arrêt sur le bas-côté, ou un composant diversifié surveillant le calcul de la trajectoire et de l'espace navigable.
  - ◇ Nous pourrons dériver ce Safety-Bag générique pour d'autres domaines d'application.

Bibliographie

[Alami et al., 1998] Alami, R., Chatila, R., Fleury, S., Ghallab, M., and Ingrand, F. (1998). An architecture for autonomy. *In Proceedings of The International Journal of Robotics Research, SAGE Publications, April 1998*, 17(4) :315–337.

[Althoff et al., 2009] Althoff, M., Stursberg, O., and Buss, M. (2009). Safety assessment of driving behavior in multi-lane traffic for autonomous vehicles. In *Proceedings of the IEEE Intelligent Vehicles Symposium, 3-5 Jun 2009, Xi'an, China*, pages 893–900.

[Arlat et al., 1993] Arlat, J., Costes, A., Crouzet, Y., Laprie, J.-C., and Powell, D. (1993). Fault injection and dependability evaluation of fault-tolerant systems. *In Proceedings of the IEEE Transactions on Computers, (TC 1993)*, 42(8) :913–923.

[Aven, 2011] Aven, T. (2011). On the new iso guide on risk management terminology. *In Reliability engineering & System safety*, 96(7) :719–726.

[Avizienis et al., 2004] Avizienis, A., Laprie, J.-C., Randell, B., and Landwehr, C. (2004). Basic concepts and taxonomy of dependable and secure computing. *In Proceedings of the IEEE Transactions on Dependable and Secure Computing, (Jan.-March 2004)*, 1(1) :11–33.

[Azevedo et al., 2013] Azevedo, L. S., Parker, D., Walker, M., Papadopoulos, Y., and Araujo, R. E. (2013). Automatic decomposition of safety integrity levels : Optimization by tabu search. In *SAFECOMP Workshop CARS (2nd Workshop on Critical Automotive applications : Robustness & Safety) of the 32nd International Conference on Computer Safety, Reliability and Security, 14-27 September 2013, Toulouse, France.*

[Baudin et al., 2007] Baudin, E., Blanquart, J., Guiochet, J., and Powell, D. (2007). Independant safety systems for autonomy. Technical report, LAAS-CNRS, Toulouse, France.

[Bengtsson et al., 1996] Bengtsson, J., Larsen, K., Larsson, F., Pettersson, P., and Yi, W. (1996). Uppaal—a tool suite for automatic verification of real-time systems. In *Proceedings of the DIMACS/SYCON workshop on Hybrid systems III : verification and control : verification and control*, pages 232–243. Springer-Verlag New York, ISBN : 354-0-611-55-X.

[Bernard et al., 2000] Bernard, D. E., Gamble, E., Rouquette, N. F., Smith, B., Tung, Y.-W., Muscettola, N., Dorias, G. A., Kanefsky, B., Kurien, J., Millar, W., et al. (2000). Remote agent experiment ds1 technology validation report. *In Ames Research Center and JPL, 8-9 February 2000, California, USA*.

[Blanquart et al., 2004] Blanquart, J., Fleury, S., Hernerk, M., Honvault, C., Ingrand, F., Poncet, J., Powell, D., Strady-Lécubin, N., and Thévenod, P. (2004). Software safety supervision on-board autonomous spacecraft. experimentation of spaas reusable components. In *2nd European Congress on Embedded Real Time Software (ERTS-2), Toulouse, France, 2004*, volume 11p.

[Blanquart et al., 2003] Blanquart, J.-P., Fleury, S., Hernek, M., Honvault, C., Ingrand, F., Poncet, J.-C., Powell, D., Strady-Lécubin, N., and Thévenod, P. (2003). Software product assurance for autonomy on-board spacecraft. In *Proceedings of DASIA 2003 (ESA SP-532). 2-6 June 2003, Prague*, volume 532.

[Brini et al., 2016] Brini, M., Crubillé, P., Lussier, B., and Schon, W. (2016). Risk reduction of experimental autonomous vehicles : The safety-bag approach. In *CARS 2016 4th International Workshop on Critical Automotive Applications : Robustness & Safety*.

[Brini et al., 2017] Brini, M., Crubille, P., Lussier, B., and Schön, W. (2017). Contraintes de sécurité pour le safety-bag d'un véhicule autonome : méthodes amdec et hazop. In *12th International Pluridisciplinary Congress on Quality, Dependability and substainability (QUALITA 2017)*.

[Brini et al., 2018] Brini, M., Crubille, P., Lussier, B., and Schon, W. (2018). Validation of safety necessities for a Safety-Bag component in experimental autonomous vehicles. In *14th European Dependable Computing Conference (EDCC 2018)*, pages 33–40, Iasi, Romania.

[Campbell et al., 2010] Campbell, M., Egerstedt, M., How, J. P., and Murray, R. M. (2010). Autonomous driving in urban environments : approaches, lessons and

challenges. *In Philosophical Transactions of the Royal Society of London A : Mathematical, Physical and Engineering Sciences*, 368(1928) :4649–4672.

[Chebly et al., 2017] Chebly, A., Talj, R., and Charara, A. (2017). Maneuver planning for autonomous vehicles, with clothoid tentacles for local trajectory planning. In *Proceedings of the IEEE of the 20th International Conference on Intelligent Transportation (ITSC 2017), Oct 2017, Yokohama, Japan.*

[Chu, 2011] Chu, H.-N. (2011). *Test and Evaluation of the Robustness of the Functional Layer of an Autonomous Robot.* PhD thesis, Institut National Polytechnique de Toulouse-INPT, Toulouse, France.

[Cimatti et al., 2002] Cimatti, A., Clarke, E., Giunchiglia, E., Giunchiglia, F., Pistore, M., Roveri, M., Sebastiani, R., and Tacchella, A. (2002). Nusmv 2 : An opensource tool for symbolic model checking. In *Proceedings of the 14th International Conference on Computer Aided Verification*, pages 359–364. Spring, ISBN : 354-0-439-97-8.

[Dhouibi et al., 2014] Dhouibi, M. S., Perquis, J.-M., Saintis, L., and Barreau, M. (2014). Automatic decomposition and allocation of safety integrity level using system of linear equations. In *Proceedings of the 4th International Conference on Performance, Safety and Robustness in Complex Systems and Applications (PESARO 2014), Nice, Fance*, pages 1–5.

[Erb, 1989] Erb, A. (1989). Safety measures of the electronic interlocking system "elektra". *In IFAC/IFIP Workshop on Safety of Computer Control Systems 1989 (SAFECOMP '89), 5-7 December, Vienna, Austria*, 22(19) :49–52.

[Ericson, 1999] Ericson, C. A. (1999). Fault tree analysis. In *Erb17th International System Safety Conference, Seattle, Washington, 1999.*

[Firesmith, 2004] Firesmith, D. (2004). Engineering safety requirements, safety constraints, and safety-critical requirements. *In the Journal of object technology (March 2004)*, 3(3) :27–42.

[Flaus, 2013] Flaus, J.-M. (2013). *Analyse des risques des systèmes de production industriels et de services : Aspects technologiques et humains*, volume 359. Lavoisier, ISBN : 978-2-746-23919-7.

[Fox and Das, 2000] Fox, J. and Das, S. K. (2000). *Safe and sound : artificial intelligence in hazardous applications*, volume 1. AAAI Press/MIT Press Menlo Park, CA/Cambridge. ISBN : 978-0-262-06211-4.

[Goodloe and Pike, 2010] Goodloe, A. E. and Pike, L. (2010). Monitoring distributed real-time systems : A survey and future directions. Technical Report NASA/CR-2010-216724, NASA Langley Research Center (July 2010).

[Guiochet, 2015] Guiochet, J. (2015). *Trusting robots : Contributions to dependable autonomous collaborative robotic systems*. Habilitation à diriger des recherches en mathématiques et en informatique, Université III Paul Sabatier, Toulouse, France.

[Guiochet, 2016] Guiochet, J. (2016). Hazard analysis of human–robot interactions with hazop–uml. *In Safety science*, 84 :225–237.

[Guiochet et al., 2010] Guiochet, J., Martin-Guillerez, D., and Powell, D. (2010). Experience with model-based user-centered risk assessment for service robots. In *High-Assurance Systems Engineering (HASE), 2010 IEEE 12th International Symposium on*, pages 104–113. IEEE.

[Guiochet and Powell, 2005] Guiochet, J. and Powell, D. (2005). Etude et analyse de différents dispositifs externes de sécurité-innocuité de type safety bag. Technical report, LAAS-CNRS, Toulouse, France.

[Haddadin et al., 2011] Haddadin, S., Suppa, M., Fuchs, S., Bodenmüller, T., Albu-Schäffer, A., and Hirzinger, G. (2011). Towards the robotic co-worker. In *The 14th International Symposium on Robotics Research (ISRR2011), edited by C. Pradalier, R. Siegwart, and G. Hirzinger*, pages 261–282. Springer.

[Holzmann, 1997] Holzmann, G. J. (1997). The model checker spin. *In IEEE Transactions on software engineering*, 23(5) :279–295.

[Houenou et al., 2013] Houenou, A., Bonnifait, P., Cherfaoui, V., and Yao, W. (2013). Vehicle trajectory prediction based on motion model and maneuver recognition. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2013), Tokyo, Japan*, pages 4363–4369.

[Huang et al., 2005] Huang, H.-M., Pavek, K., Novak, B., Albus, J., and Messin, E. (2005). A framework for autonomy levels for unmanned systems (alfus). In

Proceedings of the AUVSI's Unmanned Systems North America Conference, 28-30 June 2005, Baltimore, USA, pages 849–863.

[IEC61508-7, 2000] IEC61508-7 (2000). Functional safety of electrical/electronic/-programmable electronic safety-related systems, part 7 : Overview of techniques and measures. *In International Organization for Standardization and International Electrotechnical Commission, 2000.*

[IEC61882, 2001] IEC61882 (2001). Hazard and operability studies (hazop studies). application guide. international electrotechnical commission, 2001.

[IEEE, 2008] IEEE (2008). Standard for software and system test documentation. *In IEEE Std 829-2008, pages 1–118, 2008.*

[Ingrand and Ghallab, 2017] Ingrand, F. and Ghallab, M. (2017). Deliberation for autonomous robots : A survey. *In Artificial Intelligence (June 2017)*, 247 :10–44.

[ISO9126, 2001] ISO9126 (2001). Software product evaluation–quality characteristics and guidelines for their use. *In ISO/IEC Standard*, 2001.

[ISO/IEC-Guide51, 1990] ISO/IEC-Guide51 (1990). Safety aspects–guidelines for their inclusion in standards. international organization for standardization and international electrotechnical commission, 1990.

[Kantz and Koza, 1995] Kantz, H. and Koza, C. (1995). The elektra railway signalling system : field experience with an actively replicated system with diversity. In *Proceedings of the IEEE Twenty-Fifth International Symposium on Fault-Tolerant Computing. Digest of Papers., 27-30 June 1995, Pasadena, CA, USA*, pages 453–458.

[Klein, 1991] Klein, P. (1991). The safety-bag expert system in the electronic railway interlocking system elektra. *In Expert Systems with Applications*, 3(4) :499–506.

[Koopman and Wagner, 2016] Koopman, P. and Wagner, M. (2016). Challenges in autonomous vehicle testing and validation. *In SAE International Journal of Transportation Safety*, 4(2016-01-0128) :15–24.

[Kyriakidis et al., 2015] Kyriakidis, M., Happee, R., and de Winter, J. C. (2015). Public opinion on automated driving : Results of an international questionnaire

among 5000 respondents. *In Transportation research part F : traffic psychology and behaviour, June 2015*, 32 :127–140.

[Laprie et al., 1996] Laprie, J.-C., Arlat, J., Blanquart, J.-P., Costes, A., Crouzet, Y., Deswarte, Y., Fabre, J.-C., Guillermain, H., Kaâniche, M., Kanoun, K., et al. (1996). Guide de la sûreté de fonctionnement. cépaduès, 2e édition. Technical report, ISBN : 978-2-854-28382-2.

[Lecubin et al., 2001] Lecubin, N., Poncet, J., Powell, D., and Thévenod, P. (2001). Spaas : Software product assurance for autonomy on-board spacecraft. Technical report, 01267, July 2001, LAAS-CNRS, Toulouse, France.

[Leucker and Schallhart, 2009] Leucker, M. and Schallhart, C. (2009). A brief account of runtime verification. *In the Journal of Logic and Algebraic Programming*, 78(5) :293–303.

[Lussier, 2007] Lussier, B. (2007). *Tolérance aux fautes dans les systèmes autonomes*. PhD thesis, Institut National Polytechnique de Toulouse, Toulouse, France.

[Lussier et al., 2004] Lussier, B., Chatila, R., Ingrand, F., Killijian, M.-O., and Powell, D. (2004). On fault tolerance and robustness in autonomous systems. *In Proceedings of the 3rd IARP-IEEE/RAS-EURON joint workshop on technical challenges for dependable robots in human environments, 7-9 September, 2004, Manchester, Great Britain*, pages 351–338.

[Machin, 2015] Machin, M. (2015). *Synthèse de règles de sécurité pour des systèmes autonomes critiques*. PhD thesis, Université III-Paul Sabatier, Toulouse, France.

[Machin et al., 2016] Machin, M., Guiochet, J., Waeselynck, H., Blanquart, J.-P., Roy, M., and Masson, L. (2016). Smof : A safety monitoring framework for autonomous systems. *In Proceedings of the IEEE Transactions on Systems, Man, and Cybernetics : Systems, 2016*, 48(5) :702–715.

[Martin-Guillerez et al., 2010] Martin-Guillerez, D., Guiochet, J., Powell, D., and Zanon, C. (2010). A uml-based method for risk analysis of human-robot interactions. In *Proceedings of the 2nd International Workshop on Software Engineering for Resilient Systems (SERENE 2010), April 2010, London, United Kingdom*, pages 32–41.

[Masson et al., 2017] Masson, L., Guiochet, J., Waeselynck, H., Desfosses, A., and Laval, M. (2017). Synthesis of safety rules for active monitoring : application to an airport light measurement robot. In *Proceedings of the IEEE International Conference on Robotic Computing (IRC 2017), April 2017, Taichung, Taiwan*, pages 263–270.

[Mazouni et al., 2008] Mazouni, M. H., Aubry, J.-F., et al. (2008). Méthode systémique et organisationnelle d'analyse préliminaire des risques basée sur une ontologie générique. In *Workshop Surveillance, Sûreté et Sécurité des Grands Systèmes, 3SGS'08, Jun 2008, Troyes, France*.

[Medikonda and Panchumarthy, 2009] Medikonda, B. S. and Panchumarthy, S. R. (2009). An approach to modeling software safety in safety-critical systems. *In the Journal of computer Science*, 5(4) :311.

[Mekki Mokhtar, 2012] Mekki Mokhtar, A. (2012). *Processus d'identification de propriétés de sécurité-innocuité vérifiables en ligne pour des systèmes autonomes critiques*. PhD thesis, Université III-Paul Sabatier, Toulouse, France.

[Meltz and Guterman, 2014] Meltz, D. and Guterman, H. (2014). Verification of safety for autonomous unmanned ground vehicles. In *Proceedings of the IEEE 28th Conference on Convention of Electrical & Electronics Engineers in Israel (IEEEI 2014), 3-5 December 2014, Eilat, Israel*, pages 1–5.

[Menzies and Pecheur, 2005] Menzies, T. and Pecheur, C. (2005). Verification and validation and artificial intelligence. *In Advances in computers*, 65 :153–201.

[Micskei et al., 2012] Micskei, Z., Szatmári, Z., Oláh, J., and Majzik, I. (2012). A concept for testing robustness and safety of the context-aware behaviour of autonomous systems. In *Agent and Multi-Agent Systems. Technologies and Applications*, pages 504–513. Springer, ISBN : 978-3-642-30947-2.

[MIL-STD-882c, 1993] MIL-STD-882c (1993). System safety program requirements. *In MIL-STD-882c, US Department of Defense, USA*.

[Naufal et al., 2017] Naufal, J. K., Camargo, J. B., Vismari, L. F., de Almeida, J. R., Molina, C., González, R. I. R., Inam, R., and Fersman, E. (2017). A^2cps : A vehicle-centric safety conceptual framework for autonomous transport systems. In *Proceedings of the IEEE International Confernece on Intelligent Transportation*

Systems Society (ITSC 2017), 16-19 October 2017, Yokohama, Japan, pages 1925–1939.

[NHTSA, 2016] NHTSA (2016). Federal automated vehicles policy : Accelerating the next revolution in roadway safety. *In US Department of Transportation Washington, DC, September 2016. ISBN : 978-1-539-00221-5.*

[OICA, 2014] OICA (2014). Automated driving : Definition for levels of automation. Technical report, OICA, Paris, France.

[Pace et al., 2000] Pace, C., Seward, D., and Sommerville, I. (2000). A safety integrated architecture for an autonomous excavator. In *Proceedings of the IEEE 17th IAARC/CIB/IEEE/IFR International Symposium on Automation and Robotics in Construction (ISARC 2000), 18-20 September 2000, Taipei, Taiwan.*

[Pace and Seward, 2000] Pace, C. J. and Seward, D. W. (2000). An approach to safety for a robotic excavator. In *Proceedings of the 17th IAARC/CIB/IEEE/IFR International Symposium on Automation and Robotics in Construction : (ISARC 2000), 18-20 September 2000, Taipei, Taiwan*, pages 415–420.

[Pecheur, 2000] Pecheur, C. (2000). Verification and validation of autonomy software at nasa. Technical report, Ames Research Center, Moffett Field, California, USA.

[Powell and Thévenod-Fosse, 2002] Powell, D. and Thévenod-Fosse, P. (2002). Dependability issues in ai-based autonomous systems for space applications. In *Proceedings of the 2nd IARP-IEEE/RAS joint workshop on Technical Challenge for Dependable Robots in Human Environments, Toulouse, France*, page 163.

[Py and Ingrand, 2002] Py, F. and Ingrand, F. (2002). Online execution control checking for autonomous systems. In *Proceedings of the 7th International Conference on Intelligent Autonomous Systems (IAS-7), 25-27 March 2002, Marina del Rey, California*, page 273. IOS Press.

[Py and Ingrand, 2004a] Py, F. and Ingrand, F. (2004a). Dependable execution control for autonomous robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS 2004), Sendai, Japan*, volume 2.

[Py and Ingrand, 2004b] Py, F. and Ingrand, F. (2004b). Real-time execution control for autonomous systems. In *Proceedings of the 2nd European Congress ERTS, Embedded Real Time Software (ERTS-2), Toulouse, France*, pages 21–23.

[Rausand, 2013] Rausand, M. (2013). *Risk assessment : theory, methods, and applications*, volume 115. John Wiley & Sons, ISBN : 978-1-118-28110-9.

[Roderick et al., 2004] Roderick, S., Roberts, B., Atkins, E., and Akin, D. (2004). The ranger robotic satellite servicer and its autonomous software-based safety system. *In IEEE Intelligent Systems*, 19(5) :12–19.

[Schroeder, 1995] Schroeder, B. A. (1995). On-line monitoring : A tutorial. *In IEEE Computer Society*, 28(6) :72–78.

[Stamatelatos et al., 2002] Stamatelatos, M., Vesely, W., Dugan, J., Fragola, J., Minarick, J., and Railsback, J. (2002). Fault tree handbook with aerospace applications. *In NASA Headquarters Office of Safety and Mission Assurance, NASA Langley Research Center, August 2002, Washington DC, USA.*

[Theuretzbacher, 1986] Theuretzbacher, N. (1986). Using ai-methods to improve software safety. In *Proceedings of the Fifth IFAC Workshop on Safety of Computer Control Systems 1986 (Safecomp'86) : Trends in Safe Real Time Computer Systems, 14-17 October 1986, Sarlat, France*, number 71, page 99.

[Thrun et al., 2006] Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., et al. (2006). Stanley : The robot that won the darpa grand challenge. *In the Journal of field Robotics*, 23(9) :661–692.

[Tiwari and Sinha, 2003] Tiwari, A. and Sinha, P. (2003). Issues in v&v of autonomous and adaptive systems. In *Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering (CCECE 2003), 4-7 May 2003, Montreal, Quebec, Canada*, volume 2, pages 1339–1342.

[Urmson et al., 2008] Urmson, C., Anhalt, J., Bagnell, D., Baker, C., Bittner, R., Clark, M., Dolan, J., Duggins, D., Galatali, T., Geyer, C., et al. (2008). Autonomous driving in urban environments : Boss and the urban challenge. *In the Journal of Field Robotics*, 25(8) :425–466.

[Urmson et al., 2004] Urmson, C., Anhalt, J., Clark, M., Galatali, T., Gonzalez, J. P., Gowdy, J., Gutierrez, A., Harbaugh, S., Johnson-Roberson, M., Kato, H., et al. (2004). High speed navigation of unrehearsed terrain : Red team technology for grand challenge 2004. Technical report, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, CMU-RI-04-37.

[Vesely et al., 1981] Vesely, W., Haasl, D., Goldberg, F., and Roberts, N. (1981). Fault tree handbook. *In Systems and Reliability Research Office of Nuclear Regulatory Research (U.S. Nuclear Regulatory Commission Washington, D.C. 20555).*

[Xu and Yuan, 2016] Xu, T. and Yuan, H. (2016). Autonomous vehicle active safety system based on path planning and predictive control. In *Proceedings of the IEEE 35th Chinese Control Conference (CCC 2016), 27-29 July 2016, Chengdu, China.*

[Zou et al., 2014] Zou, X., Alexander, R., and McDermid, J. (2014). Safety validation of sense and avoid algorithms using simulation and evolutionary search. In *33rd International Conference on Computer Safety, Reliability, and Security (SAFECOMP 2014), 10-12 September, Florence, Italy*, pages 33–48.