Contraintes de sécurité pour le Safety-Bag d'un véhicule autonome : méthodes AMDEC et HazOp

Manel Brini, Paul Crubille, Benjamin Lussier, Walter Schön

Contraintes de sécurité pour le *Safety-Bag* d'un véhicule autonome : méthodes AMDEC et HazOp

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

***Résumé*—L'utilisation des véhicules autonomes robotisés comporte des risques importants en cas de défaillances, notamment dans le cas de plateformes expérimentales qui n'ont pas suivi le cycle de certification requis par les systèmes industriels. Une des techniques permettant d'augmenter la sécurité-innocuité (*safety*) et la confiance dans ces systèmes autonomes est la mise en place d'un composant *Safety-Bag*, capable de surveiller leur bon fonctionnement et d'intercepter des commandes dangereuses. Cependant, la définition des contraintes de sécurité, qui permettent un tel comportement n'est pas facile, et peut dépendre de l'architecture du système et de ses composants. Cet article présente une étude de sûreté de fonctionnement utilisant les techniques d'analyse de risques AMDEC et HazOp pour définir ces contraintes de sécurité, et compare les résultats obtenus.**

***Index Terms*—*Safety-Bag*, Sûreté de fonctionnement, AMDEC, HAZOP, Véhicule autonome.**

I. Introduction

Les véhicules automobiles autonomes sont des robots mobiles évoluant dans un environnement ouvert dans lequel ils doivent respecter des règles strictes du code de la route, mais aussi être capables de réagir face à des situations. Selon leur niveau d'autonomie, ils doivent être capables d'assurer leur mission dans des contextes variés et en réalisant des tâches complexes, ce qui nécessite des mécanismes d'intelligence artificielle pour le traitement de la perception et des décisions. Or, ces mécanismes sont très difficiles à valider expérimentalement ou formellement. De plus, ces véhicules rapides et capables d'une dynamique importante, peuvent occasionner en cas de défaillance des dégâts considérables en termes économiques ou de vies humaines. Leur sécurité-innocuité (*safety*) doit donc être garantie. La sûreté de fonctionnement des véhicules automobiles autonomes devient ainsi un verrou scientifique et technologique clé pour leur industrialisation. Les composants indépendants de sécurité [3], aussi appelés *Safety-Bag*, ont été développés pour permettre de tolérer des fautes dans un système malgré une complexité interne difficile à maîtriser, et nous semble une solution prometteuse pour les véhicules autonomes [4]. Leur fonctionnement est cependant basé sur des contraintes de sécurité, dépendant de l'application et du système, et difficiles à énoncer [6].

Nous nous intéressons dans cet article à l'application des méthodes classiques d'analyses de risques AMDEC [10] et HazOp/UML [11] à l'un des véhicules autonomes expérimentaux du laboratoire Heudiasyc. Nous montrons comment ces analyses permettent d'extraire des exigences de sécurité d'un système réel et comment dériver de celles-ci des contraintes de sécurité implémentables dans un *Safety-Bag*.

La deuxième section de cet article présente un bref contexte sur l'architecture des systèmes autonomes, le composant *Safety-Bag*, ainsi que les méthodes d'analyse de sûreté de fonctionnement AMDEC et HazOp. La troisième section présente rapidement le véhicule autonome expérimental sur lequel nous travaillons, ainsi que l'implémentation de son *Safety-Bag*. La quatrième section présente une échelle de gravité nécessaire aux analyses de risque AMDEC et HazOp, respectivement présentées en section V et VI. La section VII décrit comment nous sommes passés des exigences de sécurité issues des analyses de sécurité, à des contraintes utilisables par le *Safety-Bag*. La section VIII présente une comparaison entre les résultats de ces deux méthodes pour la génération de contraintes de sécurité nécessaires au développement du *Safety-Bag*. Finalement, l'article se termine par des conclusions et nos perspectives pour la suite de ces travaux.

II. Contexte/Etat de l'art

Nous présentons dans cette partie l'architecture en trois niveaux des véhicules autonomes. Nous définissons par la suite l'approche *Safety-Bag* en citant quelques exemples de dispositifs de sécurité-innocuité existants dans la littérature. Nous finissons cette partie en introduisant les deux techniques d'analyse de risques AMDEC et HazOp.

A. Véhicules automobiles autonomes

La SAE (Society of Automative Engineers) [16] et l'OICA (Organisation Internationale des Constructeurs Automobile) [17] ont défini des classifications des véhicules en fonction de leur autonomie. Nous utilisons dans nos travaux la classification de l'OICA, qui comprend 6 niveaux.

- Le niveau 0 correspond aux véhicules classiques sans fonctions autonomes.
- Le niveau 1 correspond aux véhicules équipés de fonctions d'assistances à la conduite telles que la régulation ou la limitation de vitesse.
- Le niveau 2 correspond aux véhicules qui disposent d'une autonomie limitée à quelques situations particulières telles que le *park assist*. Dans ce cas, le conducteur reste entièrement reponsable de la conduite du véhicule.
- Le niveau 3 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 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 correspond à des véhicules totalement autonomes dans toutes les circonstances. Ils ne nécessitent pas de supervision humaine.

Nos véhicules autonomes expérimentaux au laboratoire Heudiasyc se placent dans les niveaux 3 à 4 de cette classification. Dans notre cas de plateforme expérimentale, une mise en état sûr peut toujours se concrétiser par un retour au mode manuel, même si d'autres actions peuvent être nécessaires pour éviter des collisions ou pertes de contrôle très rapides. Dans leur développement, les systèmes autonomes complexes utilisent massivement l'architecture hiérarchisée présentée dans la Figure 1 [14]. Elle consiste généralement en trois couches :

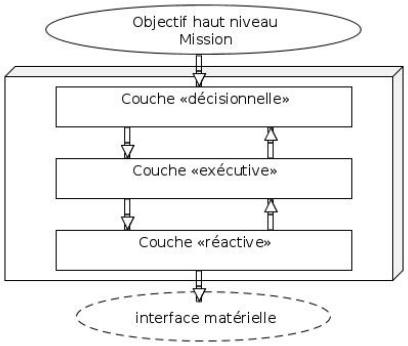

FIGURE 1. Architecture des trois niveaux

— 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.
— Une couche exécutive (ou couche de supervision) : c'est la couche intermédiaire qui découpe les activités 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 réactive (ou couche fonctionnelle) : elle est chargée de l'exécution des tâches élémentaires du plan défini par les couches supérieures telles que le suivi de la trajectoire cinématique [1]. Elle effectue un ensemble d'actions de bas niveau et contrôle les actionneurs. Son cycle d'exécution est généralement autour de 100 voire 1000 Hz.

B. Safety-bag

Un composant indépendant de sécurité, aussi appelé *Safety-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 règles de sécurité pour prévenir les défaillances catastrophiques.

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 et donc au moins partiellement redondants [1].

Cette approche a été effectivement utilisée pour des applications critiques telles que : le système ferroviaire ELEKTRA [8], le projet de module spatial autonome SPAAS [2], le système SPIN pour la supervision nucleaire [9], l'environnement SMOF et un robot excavateur [7].

C. AMDEC

L'analyse des Modes de Défaillances, de leurs Effets et leur Criticité (AMDEC) (ou Failure Mode, Effects and Criticallity Analysis, FMECA) est souvent utilisée dans l'industrie automobile et spatiale. Selon la norme IEC61508-7 [13], 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*. La méthode AMDEC 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 [5].

L'analyse AMDEC se présente ainsi sous la forme d'un tableau détaillant chaque composant, ses différents types de défaillances et les effets associés, y compris leur gravité. Le tableau AMDEC comporte aussi le taux de défaillances de chaque composant, et les moyens de détection, d'action ou de correction.

D. HazOp

La méthode HazOp (Hazard and Operability analysis) a 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 [12]. A l'origine, l'HazOp a été développée dans les années 1970 par Imperial Chemical Industries (ICI) pour traiter les systèmes thermohydrauliques. La méthode HazOp

1. La trajectoire cinématique : est l'ensemble des positions, cap et vitesse que parcourt le véhicule. Pour le véhicule autonome, les trajectoires cinématiques à parcourir sont calculées localement et constituent la mission à court terme.

a l'avantage d'identifier les dangers d'un système de façon systématique en appliquant à chaque paramètre d'un modèle du système un mot guide, produisant une déviation [5].

Afin d'appliquer cette technique, une modélisation en langage UML (Unified Modeling Langage), adaptée à l'analyse de risques HazOp, des diagrammes de cas d'utilisation et des diagrammes de séquences doit être faite dès le début dans les phases de conception et de développement. Cette modélisation permet de décrire l'utilisation du système et d'organiser les interactions possibles avec les acteurs. Des attributs sont associés à chaque cas d'utilisation pour préciser :

— des pré-conditions : qui sont des conditions préalables obligatoires au bon déroulement du cas d'utilisation.
— des invariants : qui sont des conditions qui doivent être vraies pendant le déroulement du cas d'utilisation.
— des post-conditions : qui sont des conditions vérifiées à la fin du cas d'utilisation.

L'étape suivante consiste à appliquer un ensemble de mots guides pour chaque attribut de chaque cas d'utilisation. Parmi ces mots guides, on trouve en particulier :

— No/none : La condition n'est pas évaluée et peut avoir n'importe quelle valeur.
— Other than : La condition est évaluée vraie alors qu'elle est fausse ou inversement.
— Early : La condition est évaluée plut tôt que nécessaire.
— Late : La condition est évaluée plus tard que nécessaire.

L'analyse HazOp se présente ainsi sous la forme d'un tableau détaillant pour chaque attribut de chaque cas d'utilisation et pour chaque mot guide la déviation associée au mot guide, l'effet de cette déviation sur le cas d'utilisation, l'effet de cette déviation sur le système, sa gravité, les causes possibles de chaque déviation (logicielle, matérielle, erreur externe, environnement, etc.), ainsi que l'exigence de niveau d'intégrité et la contrainte de sécurité (laquelle est également numérotée).

Les déviations obtenues par la méthode d'analyse de risque HazOp sont utilisées pour extraire et pour exprimer une liste des contraintes de sécurité en langage naturel qui seront par la suite exprimées en langage formel.

III. *Safety-Bag* POUR LES VÉHICULES AUTONOMES

Nous présentons maintenant l'architecture d'un véhicule autonome expérimental de type Fluence développé dans le laboratoire Heudiasyc. Le diagramme de déploiement UML figure 2 décrit cette architecture.

Le véhicule possède des capteurs proprioceptifs (capteurs donnant des informations sur l'état du véhicule) et des capteurs extéroceptifs (capteurs qui fournissent des informations sur l'environnement). Les capteurs de vitesse, d'angle volant, les capteurs inertiels font partie des capteurs proprioceptifs. Les capteurs extéroceptifs installés dans le véhicule comprennent un radar, plusieurs lidars et des caméras dont certaines observent le conducteur ainsi que plusieurs récepteurs GPS.

Les commandes d'accélération, de freinage et de la direction, calculées par l'application de contrôle-commande, sont envoyées aux actionneurs sous forme de signaux analogiques produits par des convertisseurs Digitaux/Analogiques (D/A).

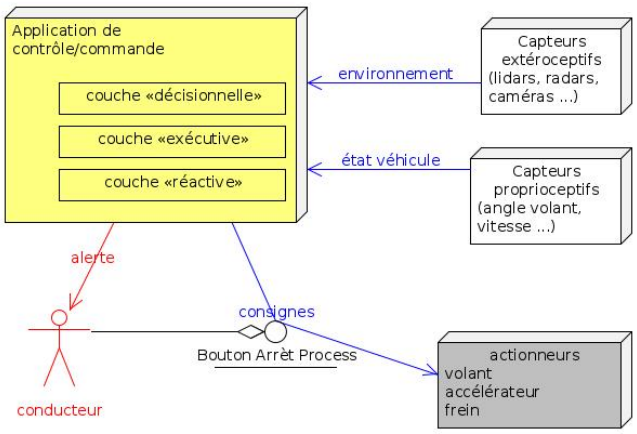

FIGURE 2. Architecture véhicule autonome sans Safety-bag

L'accélération est commandée en se substituant à la pédale, le frein est actionné par un moteur couple et la direction en se substituant au capteur de couple de la DAE (Direction Assistée Electrique).

Un Bouton d'Arrêt de Process permet au conducteur de désactiver le contrôle automatique et de reprendre le contrôle en manuel. Les alarmes sonores et les alarmes visuelles peuvent alerter le conducteur pour lui demander de reprendre la conduite en manuel si l'application de contrôle-commande estime ne plus être capable de conduire le véhicule soit que l'environnement de conduite n'est plus adapté, soit parce que une défaillance a été détectée.

Un composant *Safety-Bag* a été conçu et développé sur ce véhicule robotisé.

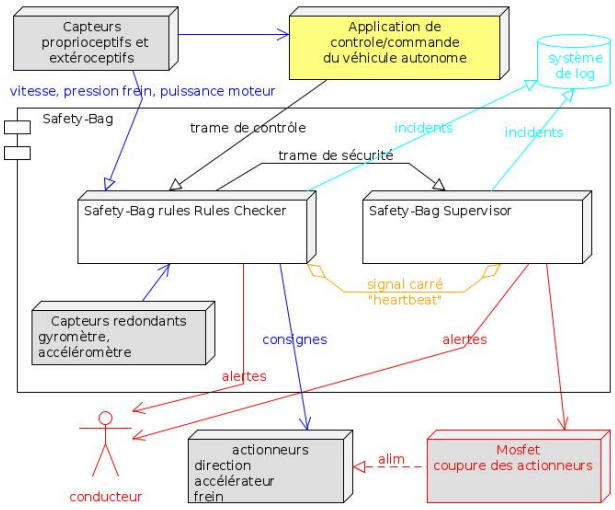

FIGURE 3. Architecture de l'application de contrôle-commande du véhicule autonome avec Safety-Bag

Une défaillance simple, notamment d'un calculateur ne doit pas provoquer la défaillance de l'ensemble du *Safety-Bag*. Il

est ainsi composé de deux calculateurs qui se surveillent l'un l'autre. Comme le montre la figure 3, l'un est appelé *Safey-Bag Rules Checker*, et l'autre *Safety-Bag Supervisor*. Chacun des deux calculateurs émet vers l'autre un *heartbeat* sous la forme d'un signal carré. Si le *Safety-Bag Supervisor* ne reçoit plus le signal du *Safety-Bag Rules Checker*, il désactive les actionneurs du véhicule via un MOSFET. Cela a le même effet que le pilote appuyant sur le Bouton d'Arrêt de Process. Si le *Safety-Bag Rules Checker* ne reçoit plus le signal du *Safety-Bag Supervisor*, il demande au pilote 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.

Le *Safety-Bag* peut déclencher des alarmes visuelles et sonores. Il enregistre tous les évènements liés à la sécurité pour permettre l'analyse post-expérimentation. Intercalé entre l'application de contrôle-commande et les actionneurs, il est capable d'analyser les sorties de l'application et de réaliser des mécanismes de rétablissement en fonction de ses contraintes de sécurité (vérification de la vivacité de l'application, de la cohérence des informations, du respect de règles comme une vitesse maximale, etc.). La définition de ces contraintes est l'objet de cet article.

IV. Echelle de gravité

Nous présentons ici la première étape d'analyse de risques qui consiste en la définition d'une table de niveaux de gravité (voir figure 4). Chaque défaillance identifiée sera placé sur cette échelle. Cette échelle est ainsi un guide pour la réduction des risques.

- 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 pilote et l'arrêt de l'expérimentation.
- Le niveau 1 : C'est 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 on a toujours le lidar qui fonctionne).
- Le niveau 2 : correspond à une conduite incorrecte par exemple une trajectoire cinématique qui pourrait heurter un obstacle du fait d'une non détection. Une défaillance de niveau de gravité 2 nécessite une intervention rapide du pilote et la possibilité de son occurrence impose une supervision vigilante de la part du pilote.
- Les niveaux 3 et 4 : correspondent à un arrêt brutal du contrôle du véhicule, ou à des commandes incorrectes demandant un ajustement immédiat. La reprise en manuel doit être immédiate. La différence entre les deux niveaux correspond à la présence ou non d'une alarme explicite à laquelle le pilote doit être entraîné à réagir.
- Le niveau 5 : correspond à des défaillances provoquant des commandes aberrantes de rotation du volant et d'accélération. Dans ce cas, une perte du contrôle du véhicule est quasiment inévitable, même avec une reprise en main rapide.

N°	Echelle de gravité
0	Fonctionnement normal
1	Arrêt de l'expérimentation automatique ou nécessité de reprise en manuel en 10s.
2	Commande maintenue devenant obsolète → Alerte et nécessité de reprise en manuel en 2s
3	Commande non maintenue → Alerte et nécessité de reprise en manuel facile immédiate **avec** alerte.
4	Commande non maintenue → nécessité de reprise en manuel facile immédiate **sans** alerte
5	Commande aberrante maintenue avec conséquence catastrophique

Figure 4. Echelle de gravité

V. Analyse de risques AMDEC des véhicules autonomes

Dans le cas des véhicules autonomes, nous avons introduit des colonnes suplémentaires pour distinguer les effets sur le composant de contrôle-commande et les effets finaux sur le comportement du véhicule. Nous avons ajouté également une colonne décrivant les exigences de sécurité associées à chaque défaillance afin d'extraire par la suite les contraintes de sécurités. De plus, nous avons introduit dans la table des sous colonnes pour indiquer l'ordre de grandeur des délais de détection et de réaction du système ou de l'opérateur.

La Figure 5 est un extrait de la table d'analyse AMDEC du véhicule autonome expérimental du laboratoire Heudiasyc sans système de réduction de risque. Cet extrait correspond à quatre composants et à leurs défaillances. Cet exemple 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'occurence significatives.

Le taux de défaillance est difficile à obtenir sans effectuer de nombreuses expérimentations. Les valeurs données dans la table sont ainsi des ordres de grandeurs obtenus à partir de ce que nous savons du système et de ses composants.

Dans notre exemple, la première ligne informe que les défaillances de type *panne bloquée* de l'application de contrôle du véhicule laissent inchangées les consignes transmises aux actionneurs et que le véhicule risque alors d'exécuter des commandes aberrantes, comme d'accélérer tandis que le couple volant est maintenu. L'accident peut devenir inévitable avant que le conducteur ne puisse reprendre la conduite en manuel. Ainsi, la gravité de cette défaillance est 5. Dans le cas de cette défaillance, nous exigeons que le système soit capable de détecter la défaillance de l'application de contrôle-commande et d'assurer la mise en sécurité. De même, les défaillances des capteurs peuvent provoquer des situations dangereuses si le logiciel génère des consignes inapropriées. Nous identifions également des dangers dues aux défaillances des actionneurs.

Le but de la méthode AMDEC est d'être exhaustif et d'identifier tous les modes de défaillance de tous les composants du système. Dans notre exemple (l'étude d'un véhicule autonome expérimental sans détailler l'architecture logicielle ni les capteurs extéroceptifs), la table AMDEC considère les

Eléments	Types de défaillance	Effets informatique	Effets véhicule	λ	Détection		Action		Gravité des conséquences		Exigence de sécurité	Réf
					Moyen	t	Moyen	t	G	commentaires		
Matériel de l'application de contrôle-commande	Panne bloquée	Les sorties de convertisseur sont maintenues	Pas de contrôle : L'accélération, le frein et le couple volant sont maintenus	~10^-3	Conducteur	>2s	Conducteur	+2s	5	Le véhicule est incontrôlable.	Le système doit être capable de détecter la défaillance de CCA* et de mettre le système en état sûr au besoin.	1
	Hors tension	Les sorties de convertisseur restent à 0.	-pas d'accélération -pas de freinage autonome -La Direction Assistée Electrique est en défaillance	~10^-5	Conducteur	>2s	Conducteur et Bouton d'Arrêt de Process	+2s	4	La DAE reste en état de défaillance même après le retour en manuel.	Sous cas du «panne bloquée»	2
	Clavier/ écran	Perte des interactions par l'opérateur	_	~10^-5	opérateur	>4s	Conducteur et BAP	+2s	1	_	Le conducteur doit pouvoir reprendre la conduite en manuel par un autre moyen.	3
...												
Capteur de vitesse	Capteur	S'il y a un diagnostic par le CCA*, ce dernier doit adapter la conduite ou demander le retour en conduite manuel.	Alerte générée par le CCA	~10^-5	CCA	0.1s	Conducteur	<2s	3	Le diagnostic est-il fiable ?	Il faut effectuer une vérification de cohérence des données de vitesse. En cas d'incohérence, il faut assurer la mise en sécurité du système en repassant en mode manuel et en levant des alarmes.	13
		S'il n'y a pas de diagnostic par le CCA ➔ la commande deviendra dangereuse	Pas d'alerte et risque de vitesse excessive	~10^-5	conducteur	4s	conducteur	2s	5	Vitesse non adaptée à la situation et peut être excessive.		14
...												
Actionneur frein	Circuit de freinage et/ou actionnement de la pédale du frein	_	Pas de freinage	~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.	Il faut détecter les incohérences entre la commande du frein et la puissance de freinage.	20

* : Application de Contrôle-Commande

FIGURE 5. Extrait du tableau AMDEC pour les véhicules autonomes sans Safety-Bag

modes de défaillances de 14 composants. La détermination des exigences de sécurité est présentée dans la section VII.

VI. ANALYSE DE RISQUES HAZOP DES VÉHICULES AUTONOMES

Afin d'identifier un ensemble de *use case* 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 [15], en attendant que les autorités européennes produisent des recommandations analogues. Des 28 situations décrites dans ce document, nous avons retiré au total 25 *use case* : 8 *use case* généraux liés à notre véhicule autonome, 14 *use case* extensions de use case 5, 2 *use case* inclus dans le *use case* 8 et un *use case* qui étend le *use case* 6. Les 8 *use case* généraux des véhicules autonomes sont :

- UC1 : Passer en mode autonome
- UC2 : Passer en mode manuel à la demande de pilote
- UC3 : Générer un itinéraire [2]
- UC4 : Suivre un itinéraire
- UC5 : Générer une trajectoire cinématique
- UC6 : Suivre une trajectoire cinématique et une extension
- UC7 : Suivre des alertes
- UC8 : Connaitre l'état du véhicule autonome et de son environnement et deux use case inclus dans UC8

2. Un itinéraire : est une liste d'étapes qui permet de rejoindre la destination choisie par l'opérateur.

Dans la suite de cet article, nous détaillons en particulier le use case «*Suivre une trajectoire cinématique*». La liste des attributs associée est la suivante :

- Pré-conditions :
  - l'état cinématique [3] estimé par le véhicule est conforme à l'état cinématique réel.
  - l'état cinématique estimé par le véhicule n'a pas une valeur éronnée.
  - la trajectoire cinématique ne doit pas causer la perte de contrôle du véhicule.
  - L'espace navigable [4] contient la trajectoire cinématique.
- Invariants :
  - L'emprise du véhicule [5] en suivant la trajectoire cinématique est dans la zone navigable.
  - L'état cinématique estimé par le véhicule reste à une distance limite de la trajectoire cinématique.
  - Les composants du véhicule ne défaillent pas.
- Post-conditions :
  - Le véhicule connait la prochaine trajectoire cinématique.

3. L'état cinématique du véhicule : est constitué de sa position, son orientation ainsi que sa vitesse. Cet état est estimé par rapport au repère géographique absolu mais aussi par rapport à la carte routière embarquée et par rapport à ses cartes locales.

4. L'espace navigable : est l'ensemble des positions dans laquelles peut physiquement se trouver le véhicule sans heurter les obstacles (statiques ou mobiles).

5. L'emprise du véhicule : est la projection du véhicule sur le sol.

— L'état cinématique est à la fin de la trajectoire cinématique précédente.
— L'état cinématique du véhicule est conforme à l'état cinématique réel.

Dans la table HazOp figure 6, nous avons associé à chaque déviation les conséquences et les causes possibles, la gravité et enfin les nouvelles exigences de sécurité indispensables pour extraire par la suite les contraintes de sécurité.

Contrairement à la méthode initiale de [5] présentée section II, nous ne déterminons pas directement les contraintes de sécurité du *Safety-Bag*, mais commençons par identifier les exigences de sécurité liées à la déviation de comportement du véhicule, de façon similaire aux analyses de sécurité réalisées par des ingénieurs sur la méthode AMDEC. Cette étape nous semble nécessaire pour savoir précisément ce que doit faire le *Safety-Bag*, et déterminer s'il est capable de le faire ou si une autre méthode doit être utilisée.

VII. Détermination des contraintes de sécurité à partir des exigences de sécurité AMDEC et HazOp

Une fois l'analyse HazOp/UML du système effectuée, l'objectif est de spécifier des contraintes de sécurité à partir des exigences de sécurité trouvées. Ces contraintes de sécurité serviront par la suite à spécifier des propriétés surveillées par le *Safety-Bag*.

Cependant, toutes les exigences de sécurité ne sont pas implémentables par le *Safety-Bag* comme par exemple la deuxième ligne de la table 7 ou la troisième ligne 3 de la table 8. D'autres méthodes peuvent être nécessaires (comme la redondance de composants matériels critiques). Les vérifications à effectuer peuvent être trop complexes pour les réaliser au niveau du *Safety-Bag*. En effet, celui ci doit rester suffisament simple pour être validé facilement.

Toutes les exigences de sécurité doivent donc être parcourues et étudiées, et des exemples du résultat de cette analyse sont présentés dans les tables 7 et 8. Pour chaque exigence, les moyens de son implémentation sont détaillés, et si le *Safety-Bag* en fait partie, une contrainte de sécurité est rédigée. Cette dernière précise ce que le *Safety-Bag* doit observer, et comment il détecte une erreur à partir de ces observations. Les contraintes sont d'abord formulées en langage naturel, avant d'être exprimées en langage formel, de même que dans la méthode HazOp/UML de [5]. Par souci de simplicité, nous n'avons détaillé ici que la formulation en langage naturel.

Nous avons obtenu 21 exigences de sécurité sur 42 attributs à partir de l'analyse HazOp du use case *Suivre une trajectoire cinématique*. De ces 21 exigences de sécurité, nous avons extrait 10 contraintes de sécurité. L'approche AMDEC nous a permis de déterminer 11 contraintes de sécurité à partir de 16 exigences de sécurité.

VIII. Comparaison AMDEC/HazOp vis à vis des exigences de sécurité

Dans cette partie, nous comparons les résultats obtenus entre les deux méthodes étudiées et nous tentons de faire une synthèse. En effet, les analyses AMDEC et HazOp ont globalement le même objectif d'identifier les risques ou défaillances du système afin de les réduire par des techniques de sûreté de fonctionnement. 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 et des composants du *Safety-bag*. En fait, ces composants ne sont pas figés, et nous ne pouvons les considérer que comme des boîtes noires. Tandis que, la méthode HazOp, étant plus adaptée aux systèmes autonomes, met l'accent principalement sur le processus et l'environnement.

Cela explique le fait qu'il existe des éléments qui apparaissent dans l'un mais pas dans l'autre. Par exemple, les exigences de sécurité issues des lignes 1 et 2 de l'HazOp ne correspondent pas à des exigences de sécurité issues de l'analyse AMDEC telle que nous l'avons menée. Inversement, certains composants qui sont mentionnés dans l'AMDEC (ligne 2 par exemple) ne sont pas pris en compte dans l'HazOp. Nous devons mentionner également qu'il existe des éléments communs entre les deux techniques, qui aboutissent à des vérifications similaires ou équivalentes. Nous citons dans ce cas les deux exemples de la ligne 1 AMDEC et de la ligne 5 HazOp, et celui de la ligne 5 AMDEC et de la ligne 5 HazOp.

L'exigence de sécurité issue de la ligne 21 de l'Hazop est trop générale et large pour être traduite directement en contraintes de sécurité. En fait, les déterminer nécessite soit une analyse AMDEC des parties critiques du véhicule autonome ou de détailler davantage les use case, en particulier les attributs (conditions/invariants).

Enfin, il faut noter que les résultats de l'analyse HazOp dépendent beaucoup de choix de représentation et d'expression de la conception, et donc de la personne qui la réalise. En effet de notre point de vue, le résultat final peut varier suivant comment sont choisis et exprimés les cas d'utilisation, leurs contraintes et leurs invariants. L'analyse AMDEC nous apparaît plus systématique, même si plus pauvre du point de vue déroulement des processus, et les deux techniques nous semblent donc plutôt complémentaires.

IX. Conclusion et perspectives

De notre point de vue, un composant *Safety-Bag* permet d'améliorer significativement la sécurité-innocuité de véhicules autonomes. Ce dispositif détecte les anomalies des composants du système, notamment les composants logiciels basés sur l'intelligence artificielle, 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 le conducteur. Il est cependant difficile de prime abord de déterminer ce que ce composant *Safety-Bag* doit surveiller et face à quelles situations il doit agir.

Nous présentons dans cet article deux méthodes d'analyse de risque AMDEC et HazOp qui permettent de déterminer les contraintes de sécurité nécessaires au *Safety-Bag*. Ces méthodes analysent le système selon deux points de vue

Projet Safety-Bag
Table HazOp
Entité :

Use case 1 : Suivre la trajectoire cinématique

Date :
Préparé par :

Les préconditions :
- **PrC1 :** L'état cinématique estimé par le véhicule est conforme à la réalité.
- **PrC2 :** L'état cinématique estimé par le véhicule est incorrect.
- **PrC3** : La trajectoire cinématique ne doit pas causer la perte de contrôle du véhicule.
- **PrC4 :** L'espace navigable contient la trajectoire cinématique.

Les invariants :
- **I1 :** L'emprise du véhicule en suivant la trajectoire cinématique est dans la zone navigable.
- **I2 :** L'état cinématique estimé du véhicule reste à une distance limite de la trajectoire cinématique.
- **I3 :** Les composants des véhicules ne défaillent pas.

Les post-conditions :
- **PoC1 :** Le véhicule connait la trajectoire cinématique suivante.
- **PoC2 :** L'état cinématique du véhicule est à la fin de la trajectoire cinématique précédente.
- **PoC3 :** L'état cinématique du véhicule est conforme à l'état cinématique réel.

N°	Elément (attribut)	Mot guide	Déviation	Effet sur le système *	Effet sur le cas d'utilisation **	G	Causes possibles	ENI	Nouvelles exigences de sécurité	Remarques	NES
1	PrC1	No / none	L'état cinématique estimé par le véhicule est inconnu.	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.	5	Défaillance matérielle (capteurs, connectique) ou défaillance logicielle	D	Le système doit vérifier ou garantir que l'état cinématique est déterminé.	_	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.	5	Défaillance logicielle (fusion de données) ou matérielle (capteur de localisation)	D	Le système doit vérifier ou garantir que l'état cinématique est correct.	Plus difficile à diagnostiquer que l'exigence de sécurité 1.	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 réelle peut être aberrante.	5	Défaillance logicielle temporelle	D	Le système doit vérifier la cohérence temporelle de l'état cinématique estimé par le véhicule.	_	4
7	PrC2	Other than	L'état cinématique estimé par le véhicule n'est pas au début de la trajectoire cinématique.	Le véhicule ne sait pas quelle commande appliquer ou essaie de rejoindre brutalement la trajectoire.	Impossible de suivre la trajectoire cinématique	4 ou 5	Défaillance d'actionneur ou défaillance logicielle	D	Il faut vérifier que les actionneurs et le CCA fonctionnent correctement.	_	5

* : conséquences de la déviation sur les composants du système ** : conséquences finales pour l'utilisateur et l'environnement ENI : Exigence de Niveau d'Intégrité NES : Numéro d'Exigence de Sécurité

FIGURE 6. Extrait de l'analyse HaZop

Eléments	Type de défaillance	Exigence de sécurité	Moyens d'implémentation d'exigence de sécurité	Contrainte de sécurité en langage naturel	commentaires	N°
l'application de contrôle-commande (CCA)	Panne bloquée	Le système doit être capable de détecter la défaillance de l'application de contrôle commande et de mettre le système en état sûr au besoin.	Implémentable par le SB, qui : • filtre les commandes entre CCA et actionneurs • est informé des mises à jour des commandes (heartbeat, message réseau, etc.) En cas de non-réponse de l'application de contrôle- commande, le SB peut : • déclencher des alarmes et feux de détresse et redonner la main au pilote, • inhiber les commandes appliquées par le CCA (ne pas accélérer, ne pas tourner d'avantage le volant). • freiner si pertinent/possible	Si la dernière mise à jour de commande est trop ancienne, le SB déclenche les alarmes pilotes et véhicules, et bloque les commandes de l'application de contrôle.	Pour assurer la redondance, le SB est constitué de deux calculateurs qui s'échangent un heartbeat.	1
	Clavier/ Ecran	Le conducteur doit pouvoir reprendre la conduite en manuel par un autre moyen.	Un bouton dédié permet le passage en mode manuel.	_	Le SB ne dispose pas des informations nécessaires pour surveiller ce périphérique, et la présence du bouton de passage en mode manuel est suffisante.	3
	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 du CCA et tests de cohérence par le SB.	Le SB peut et doit vérifier : -la cohérence temporelle des commandes à appliquer, et de certaines informations -les bornes de vitesse…	Le SB ne pourra détecter qu'un sous-ensemble des défaillances fonctionnelles du CCA ou plus tardivement leurs conséquences sur la dynamique du véhicule.	5
Capteur de vitesse	La commande est basée sur de fausses informations, et peut devenir aberrante.	Il faut effectuer une vérification de cohérence des données de vitesse. En cas d'incohérence, il faut assurer la mise en sécurité en repassant en mode manuel et en levant des alarmes.	Les capteurs du SB (gyromètre et accéléromètre latéral) permettent un test de cohérence.	La vitesse de rotation et l'accélération latérale doivent être cohérentes avec la vitesse du véhicule.	Le test proposé ne permet pas un diagnostic immédiat si le véhicule roule en ligne droite pour la vitesse latérale.	12
…						
Actionneur Accélérateur	Moteur électrique du véhicule	Il faut détecter les incohérences entre la commande d'accélération et la puissance fournie par le moteur et mettre le véhicule en sécurité.	Un ampèremètre sur le moteur électrique du véhicule permet d'effectuer ce diagnostic.	L'intensité dans le moteur électrique doit correspondre à une fonction de la commande d'accélérateur.	_	17
Actionneur frein	Circuit de freinage et/ou actionnement de la pédale du frein	Il faut détecter les incohérences entre la commande du frein et la puissance de freinage.	Un capteur sur le circuit hydraulique permet d'assurer le diagnostic.	La pression doit correspondre à une fonction de la commande du frein.	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 doit donc être surveillé avec attention.	18
…						

FIGURE 7. Exigences et contraintes issues de l'analyse AMDEC

Numéro d'exigence de sécurité	Exigence de sécurité	Moyen d'implémentation d'exigence de sécurité	Contrainte de sécurité en langage naturel	commentaires
1	Le système doit vérifier ou garantir que l'état cinématique est déterminé.	Le SB peut vérifier que l'état cinématique est mis à jour.	Vérifier que l'état cinématique est mis à jour.	L'état cinématique fourni au SB doit être rafraichi régulièrement.
2	Le système doit vérifier ou garantir que l'état cinématique est corerct.	Redondance sur les mécanismes et/ou contrôle de cohérence dans le safety-bag.	On réalise 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.	➢ Redondance des mécanismes de localisation couteuse. Des vérifications de cohérence (donc non complète) sont plus faciles à mettre en œuvre. ➢ Les 3 premiers points sont des vérifications de cohérence de l'état cinématique, par contre le dernier point utilise le capteur interne du SB pour une redondance de l'information.
3	S'assurer de la conformité de ce que perçoivent le véhicule et des cartes routières embarquées.	Comparaison entre ce que perçoivent le véhicule et les cartes routières embarquées.	–	Trop complexe pour mise en place dans le Safety-bag.
4	Le système doit vérifier la cohérence temporelle de l'état cinématique estimé par le véhicule.	Safety-bag	Vérifier que l'état cinématique a été généré récemment.	L'état cinématique doit être horodaté.
5	Il faut vérifier que les actionneurs et l'application de C.C fonctionnent correctement.	Mécanismes de diagnostic des actionneurs et de CCA et mise en sécurité. Une partie est réalisée par le SB.	**a)** Le safety-bag doit vérifier pour les actionneurs que : • le frein fonctionne • l'accélérateur fonctionne • la direction fonctionne **b)** Le safety-bag doit vérifier pour CCA • la vivacité • la cohérence temporelle • des bornes de vitesse	**a)** Les fonctions du frein, de l'accélération et de la direction doivent être déterminées expérimentalement. La direction est certainement très compliquée. Le SB peut obtenir directement l'intensité dans le moteur électrique du véhicule et dans le moteur d'assistance de direction. **b)** Le safety bag ne pourra détecter qu'un sous-ensemble des défaillances de CCA.
			...	
21	Il faut tolérer des composants défaillants ou garantir que les composants ne défailliront pas.	-redondance des capteurs -Le SB doit intégrer une tolérance à ses propres fautes : ➔Redondance de ces calculateurs ➔Redondance du mécanisme de journalisation (log) ➔Au moins un moyen ultime de mise en sécurité	➢ En cas de défaillance, on doit assurer la mise en sécurité du véhicule. Pour cela, ce qui est nécessaire pour mettre en œuvre cette mise en sécurité doit être redondé. ➢ En cas de défaillance, on peut : -Redonner la main au conducteur en déclenchant une alarme. -Freiner en se décalant sur la droite (en voie rapide) -Freiner et arrêter le véhicule (en ville)	➢ Redonner la main au conducteur nécessite que celui-ci soit vigilant. Les autres mesures nécessitent que le SB connaisse les mesures à prendre et que celles-ci soient adaptées à la situation. ➢ Des défaillances de système redondant ou de sécurité provoquent des alarmes qui seront perçues par l'utilisateur comme une fausse alarme et qui réduiront la disponibilité.

FIGURE 8. Exigences et contraintes issues de l'analyse HazOp

différents et nous paraissent complémentaires pour identifier le maximum de contraintes de sécurité possibles. Cependant, il est à noter que le résultat de ces analyses (et notamment de l'analyse HazOP/UML) dépendent grandement des compétences de l'analyste. De plus, toutes les exigences de sécurité ne sont pas forcément implémentables par le *Safety-Bag*, qui doit rester suffisamment simple pour être vérifiable, et ne doit pas comporter des décisions basées sur des composants à risque (comme des mécanismes de fusion de données ou d'intelligence artificielle).

Dans la suite de nos travaux, nous comptons analyser à quel point les contraintes de sécurité retenues permettent d'améliorer la sûreté de fonctionnement de notre plateforme expérimentale.

ACKNOWLEDGMENTS

Ce travail a été réalisé et financé par EQUIPEX ROBOTEX. Il a été soutenu par le gouvernement français, à travers le programme *Investissements d'avenir* géré par l'Agence Nationale de la Recherche. (Référence : ANR-10-EQPX).

RÉFÉRENCES

[1] Lussier B. (2007). Tolérance aux fautes dans les systèmes autonomes.

[2] Blanquart J., Fleury S., &Hernek, M. (n.d.). Software Safety Supervision On-board Autonomous Spacecraft.

[3] David P. & Guiochet J. (2005). Etude et analyse de différents dispositifs externes de sécurité-innocuité de type safety bag.

[4] Baudin E., Blanquart J. P., Guiochet J. & Powell D. (2007). Independent Safety Systems for Autonomy.

[5] Mekki Mokhtar A. (2012). Processus d'identification de propriétés de sécurité-innocuité vérifiables en ligne pour des systèmes autonomes critiques. Toulouse : Université de Toulouse.

[6] Mekki Mokhtar A., Blanquart J.-P. & Guiochet J. (n.d.). Safety Trigger Conditions for Critical Autonomous Systems, 18th Pacific Rim Int. Symp. on Dependable Computing (PRDC 2012), Niigata, Japan, 2012.

[7] Pace C., Seward D., & Sommerville I. (n.d.). A Safety Integrated Architecture for an Autonomous Excavator. IEEE,Proc. 17th Int. Symp. on Automation and Robotics in Construction, Taiwan, 2000.

[8] P. Klein, The Safety-Bag Expert System in the Electronic Railway Interlocking System Elektra, Expert System with Applications, 1991.

[9] G. Guesnier, J. F. Hamelin, & J. M. Peyrouton, Centrale nucléaires N4 : l'informatique au service d'une conduite plus sure, Epure, 1997.

[10] MIL-STD-1629A 24 November 1980 : Procedures for performing a Failure Mode, Effects, and Criticality Analysis

[11] Lawley, H. G.,(1974) Chemical Engineering Progress, vol 70, no 4 page 45 "Operability studies and hazard analysis"

[12] IEC61882. Hazard and operability studies (HAZOP studies) : Application Guide. International Electrotechnical Commission, 2001. [IEC10]

[13] IEC61508-7. Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems, Part 7 : Overview of techniques and measures. International Organization for Standardization and International Electrotechnical Commission, 2010.

[14] C. Urmson & all, Environments : Boss and the Urban Challenge, Journal of Field Robotics 25(8), 425–466 (2008)

[15] Accelerating the next revolution in roadway safety, NHTSA, Septembre 2016

[16] SAE J3016 : Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems

[17] OICA Automated Driving Definition for Levels of Automation Informal document No. WP.29-162-20 - 14 March 2014