Tolérance aux Fautes par Fusion de Données : Un Cas d'Étude

Kaci Bader, Benjamin Lussier, Walter Schön

Tolérance aux Fautes par Fusion de Données : Un Cas d'Étude

BADER Kaci, LUSSIER Benjamin, SCHÖN Walter
Université de Technologie de Compiègne
UMR CNRS 7253, Heudiasyc BP 20529
Email : {kaci.bader, benjamin.lussier, walter.schon}@hds.utc.fr

***Résumé*—Les systèmes de perception multi-capteurs commencent à apparaître en applications critiques, comme les systèmes ADAS dans l'automobile. Ces systèmes, complexes et basés sur des méthodes d'intelligence artificielle, sont difficiles et coûteux à valider. Dans cet article, nous étudions une méthode alternative de la sûreté de fonctionnement : la tolérance aux fautes. Nous étudions la tolérance aux fautes telle qu'elle est directement permise par la fusion de données, puis nous proposons des mécanismes de détection et de recouvrement adaptés à la perception multi-capteurs. Par analyse de certains paramètres liés à la fusion de données, nous proposons différents services de sûreté de fonctionnement des systèmes de perception, tels que la détection, le rétablissement par compensation, et le masquage de fautes.**

I. Introduction

La perception est une entrée fondamentale de tout système robotique, mais ses données significatives sont souvent complexes, et sujettes à des incertitudes et des imprécisions importantes.

Pour remédier à ces problèmes, l'approche multi-capteurs reçoit les données provenant de capteurs multiples et complémentaires, et utilise leur redondance pour accroître la précision des systèmes perceptifs par filtrage des bruits de mesures, l'élimination de certaines données aberrantes, et l'extraction des connaissances complexes de l'environnement observé.

Mais multiplier les capteurs et les algorithmes sous-jacents pour fusionner les données augmente en conséquence les risques d'erreurs matérielles et logicielles. En outre, la validation d'une telle approche est complexifiée par l'environnement ouvert auquel les systèmes robotiques complexes sont confrontés, qui génère un contexte d'exécution quasiment infini et fait du test une tâche difficile et coûteuse. Dans ce papier, nous proposons des mécanismes de tolérance aux fautes comme alternative à la validation : puisqu'il est difficile d'éliminer toutes les fautes du système, nous allons chercher à limiter leur impact sur son fonctionnement.

Ainsi, nous étudions comment la tolérance aux fautes peut être mise en place directement par les mécanismes de fusion de données après une étude formelle, illustrant notre démarche par un cas d'étude théorique simple. En particulier, nous proposons des mécanismes de détection, de rétablissement, et de masquage d'erreur matérielle.

Cet article est organisé comme suit : après cette introduction, la section 2 présente les principales méthodes de tolérance de fautes. La section 3 introduit les principes de base de la fusion de données et détaille une de ses théories : la théorie des fonctions de croyance. La section 4 expose un état de l'art sur la tolérance aux fautes dans la fusion de données. La section 5 présente les principes de notre étude de cas : les objectifs de l'étude, les mécanismes étudiés et les fautes considérées. Les résultats de l'analyse de fusion de données sont présentés dans la section 6, cette dernière détaillant aussi les différents services de tolérance aux fautes offerts par la fusion de données. Enfin, l'article se termine par une conclusion et des perspectives pour les travaux futurs.

II. Tolérance aux fautes

La tolérance aux fautes [10] [1] est un moyen de la sûreté de fonctionnement, cette dernière caractérisant la confiance justifiée que l'on place dans un système. Le but de la tolérance aux fautes est de maintenir la délivrance correcte des services d'un système en dépit des fautes. Sa mise en œuvre s'effectue généralement par deux méthodes complémentaires : la détection d'erreur et le rétablissement du système.

1) **La détection d'erreur :** Elle constitue un pré-requis indispensable à la mise en œuvre de solution de tolérance aux fautes, en permettant de détecter l'activation d'une faute avant qu'elle ne se propage en défaillance. Il existe trois principales méthodes de détection d'erreurs :
   - *La duplication et comparaison :* Elle consiste à comparer les résultats fournis par au moins deux unités redondantes indépendantes vis-à-vis des fautes à tolérer, fournissant le même service.
   - *Le contrôle temporel et d'exécution :* Il consiste à contrôler la défaillance d'un périphérique en contrôlant que son temps de réponse ne dépasse pas une valeur maximale (par exemple par chien de garde).
   - *Le contrôle de vraisemblance :* Il cherche à détecter des erreurs en valeurs aberrantes pour le système.
2) **Le rétablissement du système :** Il consiste à substituer l'état erroné détecté par un état jugé exempt d'erreurs.
   - *Le recouvrement d'erreur par reprise :* Il consiste à capturer et sauvegarder périodiquement l'état du système, en créant un ensemble de points de reprise (ou checkpoints) possibles, de façon à pouvoir ramener le système dans un état antérieur à l'erreur détectée.
   - *Le recouvrement d'erreur par poursuite :* Il consiste, après avoir détecté une erreur, à rechercher un nouvel

état acceptable du système à partir duquel il peut continuer à fonctionner.

- *Le recouvrement d'erreur par compensation :* Il nécessite la présence de redondances dans le système lui permettant de fournir un service correct malgré les erreurs qui pourront l'affecter.

III. Fusion de données

La fusion d'informations (ou fusion de données) consiste à combiner des informations issues de plusieurs sources afin d'améliorer la prise de décision [2]. Nous entendons, par source d'informations, tout système, des capteurs physiques à l'informateur humain, observant la situation réelle ou fournissant une information a priori sur les événements possibles.

En fusion de données multi-capteurs, les entrées de plusieurs capteurs sont combinées pour former une représentation complète de l'environnement observé : *le modèle de l'environnement*. La fusion de données cherche à tirer profit de toutes les informations disponibles sur un problème donné pour contrer les imperfections de chacune des sources (imprécision, incertitude, incomplétude, ambiguïté et conflit). Elle peut également être utilisée pour résoudre des problèmes concernant l'hétérogénéité des sources d'information, et la fiabilité (connue ou inconnue) des différents capteurs. Les capteurs utilisés peuvent cibler les mêmes informations de l'environnement (capteurs compétitifs), ou des informations complémentaires (capteurs coopératifs).

La mise en œuvre d'un processus de fusion de données comprend trois étapes principales :

- ***Modélisation :*** Le choix d'un formalisme mathématique pour représenter l'information provenant de chaque source.
- ***Combinaison :*** Le choix d'un opérateur compatible avec le formalisme adopté dans le but de combiner les différentes informations.
- ***Décision :*** Le choix d'un critère de décision sur le résultat de la combinaison des informations.

Cette mise en œuvre est généralement appliquée dans l'un de trois grands cadres théoriques, à savoir la théorie des probabilités, la théorie des possibilités [8] et la théorie des fonctions de croyance [7][12]. Dans le cadre de ce travail nous nous intéressons à cette dernière théorie, qui peut être vue comme un sur-ensemble des deux autres [6], et notre étude peut donc théoriquement être étendue à celles-ci.

A. Théorie des fonctions de croyance

La théorie des fonctions de croyance est issue des travaux de Dempster en 1967 [7], repris par Shafer [12] sous le nom de théorie de l'évidence. Cette méthode permet à la fois la modélisation et l'utilisation de données incertaines et imprécises, ainsi que de données qualitatives et quantitatives.

Cette théorie est bien connue pour prendre en compte ce qui reste inconnu, tout autant que ce qui est déjà connu. Seuls les résultats essentiels sont rappelés ci-dessous.

Soit ω une variable définie sur un domaine fini Ω appelé cadre de discernement supposé exhaustif (hypothèse du monde fermé : la solution est dans Ω) et exclusif (la solution est unique). La croyance d'une source concernant la valeur prise par ω est représentée par une masse qui est une fonction de l'ensemble 2^Ω des parties de Ω dans [0,1], telle qu'on ait : $\sum_{A\subseteq\Omega} m(A) = 1$, avec la quantité $m(A)$ représentant la croyance que l'on met exactement sur la proposition A.

Les éléments A tels que $m(A) \neq 0$ sont appelés les éléments focaux. À partir de cette fonction de masse, d'autres fonctions de croyance peuvent être définies. La fonction de crédibilité représente la croyance minimale d'une source définie à partir des masses élémentaires de croyance portées par les éléments focaux. Elle est définie pour tout $A \in 2^\Omega$ par :

$$Bel(A) = \sum_{\emptyset \neq B \subseteq A} m(B) \quad \forall A \subseteq \Omega \tag{1}$$

La plausibilité (toujours plus grande que la crédibilité) est la fonction duale de la fonction de crédibilité ; elle mesure le degré maximal susceptible d'être alloué a A. Elle est définie comme suit :

$$Pl(A) = \sum_{A \cap B \neq \emptyset} m(B) \quad \forall A \subseteq \Omega \tag{2}$$

La combinaison des fonctions de masse issues des différentes sources Sj, fondamentale pour la fusion de données, peut être réalisée suivant plusieurs opérateurs. La première règle proposée par Dempster (1967) et reprise par Shafer (1976) est la règle orthogonale normalisée (ou combinaison conjonctive normalisée). La combinaison de deux fonctions de masses indépendantes m_1 et m_2 effectuée par cette règle est donnée par :

$$m_{1\oplus 2}(A) = \frac{\sum_{B\cap C=A} m_1(B)m_2(C)}{1-k} \quad \forall A \subseteq \Omega \tag{3}$$

Avec : $k = \sum_{U\cap V=\emptyset, \forall U,V\subseteq\Omega} m_1(U)m_2(V)$

k reflète la masse de croyance conflictuelle existant entre les deux fonctions de croyance.

Une fois que tous les éléments d'information ont été recueillis et toutes les croyances modélisées, mises à jour et combinées, le système doit prendre une décision sur la valeur de ω, donc choisir un singleton dans Ω, selon une règle donnée. Trois critères de décision se retrouvent couramment dans la littérature : le maximum de crédibilité, le maximum de plausibilité, et le maximum de probabilité pignistique. Si la décision prise par le maximum de crédibilité peut être trop pessimiste, la décision issue du maximum de plausibilité est généralement trop optimiste. Le maximum de la probabilité pignistique, introduit par Smets [13], reste le compromis le plus employé. Cette mesure est obtenue en répartissant la masse de croyance $m(B)$ à parts égales entre les éléments de B.

Cette probabilité peut être également définie sur les sous ensembles de Ω qui ne sont pas nécessairement des singletons :

$$BetP(A) = \sum_{B\subseteq\Omega} \frac{|B\cap A|}{|B|} \frac{m(B)}{(1-m(\emptyset))} \tag{4}$$

Où |B| est le cardinal de B.

IV. Travaux antérieurs

A notre connaissance, peu de travaux existent sur la tolérance aux fautes dans la fusion de données. Les approches que nous avons trouvées dans la littérature utilisent principalement la tolérance aux fautes par duplication/comparaison pour tolérer des fautes physiques. La duplication peut être basée sur un modèle analytique du processus de fusion, comme dans [9], où un filtre de Kalman est utilisé comme modèle mathématique pour proposer une architecture tolérante aux fautes. Les fautes transitoires sont détectées en utilisant une technique de résidus, et les fautes persistantes sont détectées par une méthode de vote présentée dans [3]. Dans [14], les auteurs proposent une architecture de tolérance aux fautes matérielles sur les capteurs, en combinant des données de capteurs redondants et un modèle mathématique pour évaluer les données et réaliser une estimation de position plus fiable.

Une autre approche est basée sur l'analyse de certains paramètres internes, tels que le conflit. Dans [11], les auteurs présentent la détection d'un dysfonctionnement de capteurs par l'analyse temporelle de conflits résultant de la fusion de données. Ces travaux sont basés sur le Modèle des Croyances Transférables (MCT) de Smets. Dans [5], un algorithme est proposé pour détecter une source défectueuse. Cet algorithme analyse des indices de confiance associés à chaque source, estimés à partir de la divergence de valeur entre les sources par la méthode de seuillage. Dans [4], une approche similaire a été proposée par les mêmes auteurs, dans le contexte spécifique de la théorie des possibilités.

De notre point de vue, les approches proposées se concentrent sur la tolérance aux fautes matérielles, en utilisant directement les mécanismes de fusion de données, sans garantie que ces derniers soient exempts de fautes. Or, ces mécanismes sont justement complexes à développer et à valider, et il nous paraît difficile d'avoir une confiance justifiée dans leur service, ce qui peut mettre en danger le bon fonctionnement des méthodes précédentes.

La solution que nous proposons se base sur une analyse formelle exhaustive du comportement du système en regard de certains paramètres de modélisation du problème, justifiant pour nous une plus grande sûreté de fonctionnement. Le prix de cette garantie est que l'analyse peut être complexe, voire irréalisable dans certains cas.

V. Principe de l'étude

Le système considéré dans cette étude est un simple cas théorique considéré pour analyser les services de tolérance aux fautes pouvant être fournis par la fusion de données. Sa fonction consiste à détecter la présence d'une personne sur une chaise. Pour atteindre cet objectif deux capteurs sont utilisés : un capteur de pression placé sous la chaise, et une caméra installée à l'avant de la chaise pour détecter une silhouette sur celle-ci.

Cette application a été reprise de [11], où elle avait été proposée dans le cadre d'une architecture de perception de contexte pour l'habitat communicant et le maintien à domicile des personnes âgées.

Dans ce scénario, nous considérons la sortie des capteurs qui n'ont pas d'erreur comme non bruitée, et sans faux positif : la sortie de la caméra et du capteur de pression est de 1 si une personne est effectivement sur la chaise, et 0 sinon. En d'autres termes, nous nous concentrons sur les fautes internes au système de perception plutôt que sur les aléas environnementaux : l'étude se concentre sur la tolérance aux fautes plutôt que la robustesse.

A. Terminologie et notations

Nous introduisons dans la table I les différentes notations et terminologies utilisées dans la suite de cet article.

Notation	Description
P_j C_j F_j avec $j \in$ {1,2}	Capteur de pression j Caméra j Bloc de fusion j
$threshold$	Seuil de décision
p^i_{trust} $p_i(\exists)$ avec $i \in$ {P,C}	Facteur d'affaiblissement associé au capteur i Sortie de capteur i avec P pour pression et C pour caméra

Table I
Notations

B. But de l'étude

Notre objectif dans cette étude est d'analyser la tolérance aux fautes fournie par la fusion de données (figure 1). Nous cherchons à déterminer les critères de détection et de recouvrement d'erreurs dans ce système de perception. Les fautes visées sont les fautes matérielles liées aux capteurs (caméra et pression), amenant une sortie correcte à devenir erronée. Les deux erreurs possibles sont ainsi : soit une personne est assise sur la chaise et un capteur erroné délivre 0 en sortie, soit il n'y a personne sur la chaise et un capteur erroné délivre 1 en sortie.

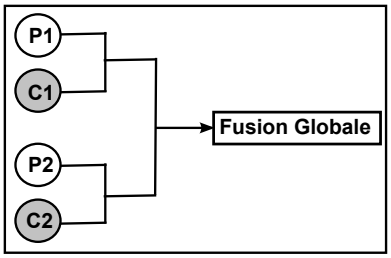

Figure 1. Architecture de fusion de données tolérante aux fautes

Avec cette architecture, nous proposons trois services de tolérance aux fautes :

- *Détection :* L'architecture permet de détecter une erreur dans les quatre capteurs.
- *Recouvrement par compensation :* Après la détection d'erreur, la sortie correcte est déduite des capteurs et de l'algorithme de fusion sans erreur.

- *Masquage :* L'architecture peut tolérer une erreur matérielle : en dépit d'un capteur erroné, le système donne le résultat correct. L'erreur n'est pas détectée, mais masquée.

Il est à noter que cette architecture ne peut détecter qu'une seule erreur a la fois présente dans le système. Comme d'habitude en tolérance aux fautes, plus de redondances permettrait cependant de tolérer plus d'erreurs simultanées.

VI. Analyse de la fusion de données

Notre analyse se découpe en deux parties : un premier cas simple sans redondance, représentant une seule branche de l'architecture proposée et permettant simplement la détection d'erreurs. Nous étudions ensuite l'architecture complète, et les trois services de tolérance aux fautes offerts.

A. Cas simple sans redondance

Dans un premier temps, nous étudions un seul jeu de capteurs (caméra, pression). Ce premier cas représente une branche de l'architecture de la figure 1.

- **Cadre de discernement**
  Le cadre de discernement est constitué de deux hypothèses binaires : soit une personne est assise sur la chaise ($\exists$), soit personne n'est assis sur la chaise ($\nexists$). Le cadre de discernement est alors composé uniquement des deux hypothèses ($\exists$) et ($\nexists$), tel que : $\Omega = \{\exists, \nexists\}$.
  A partir de ce cadre de discernement, nous construisons le référentiel de définition constitué de l'ensemble 2^Ω des parties de Ω : $2^\Omega = \{\{\exists\}, \{\nexists\}, \{\exists, \nexists\}, \emptyset\}$
- **Affectation des masses de croyance**
  Considérant $p_i(\exists)$ la sortie du capteur i ($i \in \{P, C\}$), nous avons alors :

$$\begin{cases} p_i(\exists) = 1 : \text{Si quelqu'un est assis sur la chaise.} \\ p_i(\exists) = 0 : \text{Autrement.} \end{cases} \quad (5)$$

  Pour chaque capteur i est associé un facteur d'affaiblissement $p^i_{trust} \in [0, 1]$. Les masses affectées aux hypothèses $\exists$, et $\nexists$ sont les sorties du capteur $p_i(\exists)$ pondérées par leur facteur d'affaiblissement p^i_{trust} comme indiqué dans (6) et (7).

$$\begin{cases} m_i(\{\exists\}) = p^i_{trust}.p_i(\exists) \\ m_i(\{\nexists\}) = p^i_{trust}.[1 - p_i(\exists)] \end{cases} \quad (6)$$

  D'où

$$\begin{cases} m_i(\{\exists\}) = \begin{cases} p^i_{trust} & si\ \ p_i(\exists) = 1 \\ 0 & si\ \ p_i(\exists) = 0 \end{cases} \\ et \\ m_i(\{\nexists\}) = \begin{cases} p^i_{trust} & si\ \ p_i(\exists) = 0 \\ 0 & si\ \ p_i(\exists) = 1 \end{cases} \end{cases} \quad (7)$$

  Une fois que les masses $m_i(\{\exists\})$ et $m_i(\{\nexists\})$ sont affectées, la masse restante est associée au cadre de discernement Ω ce qui modélise l'ignorance :

$$m_i(\{\exists, \nexists\}) = 1 - [m_i(\{\exists\}) + m_i(\{\nexists\})] \quad (8)$$

- **Combinaison des masses**
  Nous fusionnons les masses des trois hypothèses possibles obtenues à partir des capteurs P et C avec la règle de Dempster comme décrit dans (9). Nous obtenons ainsi la croyance globale sur les éléments de 2^Ω :

$$\begin{array}{l} m_{P\oplus C}(\{\exists\}) = \frac{m_P(\{\exists\}).m_C(\{\exists\}) + m_P(\{\exists\}).m_C(\Omega) + m_P(\Omega).m_C(\{\exists\})}{1 - [m_P(\{\exists\})m_C(\{\nexists\}) + m_P(\{\nexists\})m_C(\{\exists\})]} \\ m_{P\oplus C}(\{\nexists\}) = \frac{m_P(\{\nexists\}).m_C(\{\nexists\}) + m_P(\{\nexists\}).m_C(\Omega) + m_P(\Omega).m_C(\{\nexists\})}{1 - [m_P(\{\exists\})m_C(\{\nexists\}) + m_P(\{\nexists\})m_C(\{\exists\})]} \\ m_{P\oplus C}(\Omega) = 1 - [m_{P\oplus C}(\{\exists\}) + m_{P\oplus C}(\{\nexists\})] \end{array} \quad (9)$$

- **Décision**
  La décision est basée sur la probabilité pignistique $BetP$, calculée comme suit :

$$\begin{array}{l} BetP(\exists) = \frac{2m_{P\oplus C}(\{\exists\}) + m_{P\oplus C}(\Omega)}{2} \\ BetP(\nexists) = \frac{2m_{P\oplus C}(\{\nexists\}) + m_{P\oplus C}(\Omega)}{2} \end{array} \quad (10)$$

  Après avoir calculé la probabilité pignistique des deux hypothèses $\exists$ et $\nexists$, la décision est prise en fonction d'une valeur seuil ($threshold$) selon les deux contraintes suivantes :
  - Si $BetP(\exists) \geq threshold$ alors, nous décidons qu'une personne est assise.
  - Si $BetP(\nexists) \geq threshold$ alors, nous décidons que personne n'est assis.

Il est facile de montrer que pour un seuil supérieur à 0,5 :

$$1\text{-}threshold < BetP(\exists) < threshold \Leftrightarrow 1\text{-}threshold < BetP(\nexists) < threshold$$

1) Analyse des résultats:

a) Principe de détection: La détection de fautes dans ce système de perception est basée sur l'analyse des trois paramètres impliqués au niveau de la fusion de données : les facteurs d'affaiblissement p^P_{trust}, et p^C_{trust}, ainsi que le seuil de décision $threshold$.

Notre principe de détection des fautes est le suivant : si les deux capteurs ont la même valeur, aucune erreur n'est présente et nous voulons toujours arriver à une décision (la valeur correcte retournée par les capteurs). Si les deux capteurs ont des valeurs différentes, une erreur est présente et nous voulons toujours être incapables de décider. Ainsi, si le système décide, cela signifie qu'aucune erreur n'est présente, et s'il ne décide pas cela signifie qu'une erreur est présente. Pour respecter ce principe, les quatre conditions suivantes doivent êtres assurées.

1) En l'absence de fautes
   - **Condition 1 :** Quand il y a effectivement une personne assise sur la chaise, on veut que le système décide toujours qu'il y a bien une personne assise sur la chaise.

$$\begin{cases} P_P(\exists) = 1 \\ P_C(\exists) = 1 \end{cases} \Leftrightarrow BetP(\exists) \geq threshold$$

   - **condition 2 :** Quand il n'y a personne sur la chaise, on veut que le système décide toujours qu'il n'y a personne sur la chaise.

$$\begin{cases} P_P(\exists) = 0 \\ P_C(\exists) = 0 \end{cases} \Leftrightarrow BetP(\nexists) \geq threshold$$

2) En présence de fautes
   - **Condition 3 :** Quand il y a une personne assise sur la chaise et que les deux capteurs donnent des résultats contraires, on veut que le système ne prenne pas de décision.

$$\begin{cases} P_P(\exists) = 1 \\ P_C(\exists) = 0 \\ \text{avec C défaillant} \\ ou \\ P_P(\exists) = 0 \\ P_C(\exists) = 1 \\ \text{avec P défaillant} \end{cases} \Leftrightarrow \begin{array}{l} 1 - threshold < BetP(\exists) \\ < threshold \end{array}$$

   - **Condition 4 :** Quand il n'y a personne sur la chaise, et que les deux capteurs donnent des résultats contraires, on veut que le système ne prenne pas de décision.

$$\begin{cases} P_P(\exists) = 1 \\ P_C(\exists) = 0 \\ \text{avec P défaillant} \\ ou \\ P_P(\exists) = 0 \\ P_C(\exists) = 1 \\ \text{avec C défaillant} \end{cases} \Leftrightarrow \begin{array}{l} 1 - threshold < BetP(\nexists) \\ < threshold \end{array}$$

b) Service de détection: Pour pouvoir détecter des fautes dans le système, les quatre conditions citées précédemment doivent êtres satisfaites simultanément. Dans les figures qui suivent, nous allons présenter la probabilité pignistique associée aux deux hypothèses $\{\exists\}$, $\{\nexists\}$ en fonction des facteurs d'affaiblissement p^P_{trust},p^C_{trust}, afin d'étudier le domaine des facteurs d'affaiblissement vérifiant ces quatre conditions.

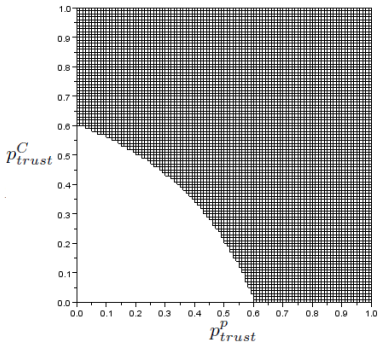

FIGURE 2. Couples p^i_{trust} satisfaisant les conditions 1 et 2

Sur la figure 2, l'axe des X représente le facteur d'affaiblissement p^P_{trust} associé au capteur de pression, et l'axe des Y le facteur d'affaiblissement p^C_{trust} associé au capteur caméra. La surface grisée représente la probabilité pignistique $BetP(\exists) \geq 0.8$ quand les deux capteurs délivrent la même sortie 1, et respectent ainsi la première condition.

Par symétrie de notre problème la surface satisfaisant la seconde condition est identique à celle satisfaisant la première condition (figure 2).

La figure 3 présente les couples de facteurs d'affaiblissement (p^P_{trust}, p^C_{trust}) des capteurs qui respectent la troisième condition.

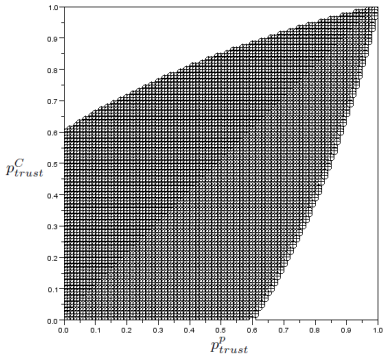

FIGURE 3. Couples p^i_{trust} satisfaisant les conditions 3 et 4

La zone grisée délimite les facteurs d'affaiblissement (p^P_{trust}, p^C_{trust}) pour lesquelles $0.2 < BetP(\exists) < 0.8$.

De même que précédemment, la surface satisfaisant la quatrième condition est identique à la figure 3.

La zone blanche en haut à gauche représente la zone pour laquelle la sortie du capteur caméra décide le résultat de la fusion : la confiance associée à la caméra est telle que sa sortie impose le résultat de la fusion quelle que soit la sortie du capteur de pression.

De même, la zone en bas à droite représente la zone pour laquelle la sortie du capteur pression décide seul le résultat de la fusion.

Finalement, les couples (p^P_{trust}, p^C_{trust}) qui permettent de détecter une erreur pour une valeur $threshold = 0.8$ sont ceux qui respectent les quatre conditions précédentes, et résultent donc de l'intersection des surfaces des figures 2 et 3. Ces couples sont représentés dans la figure 4.

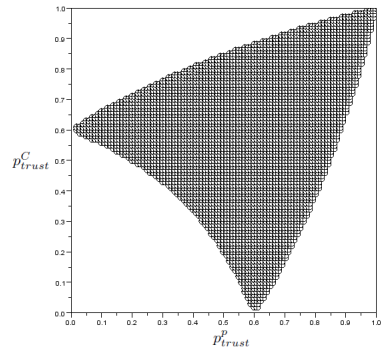

FIGURE 4. Service de détection pour un seuil 0.8

De la même façon que nous avons étudié les couples (p^P_{trust}, p^C_{trust}) permettant le service de détection proposé pour un $threshold = 0.8$, nous pouvons considérer d'autres $threshold$, et proposer les triplets (p^P_{trust}, p^C_{trust}, $threshold$) permettant le service de détection. Ces triplets sont présentés sur la figure 5. Seuls des paramètres de modélisation et de décision ont été présentés dans cet article, mais des paramètres de combinaison pourraient également être étudiés, comme

différents opérateurs de combinaison.

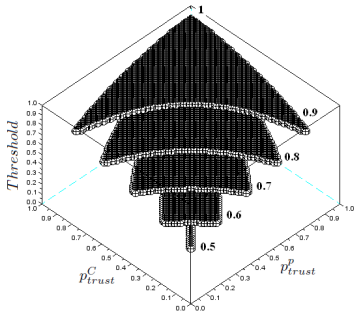

FIGURE 5. Services de détection pour des seuils variables

B. Architecture complète

Nous avons vu dans la sous section VI-A comment assurer la détection d'erreurs avec un seul jeu de capteurs compétitifs. Pour offrir d'autres services, tels que le rétablissement par compensation, ou le masquage, une duplication au niveau matériel (capteurs) est nécessaire. Dans ce qui suit nous allons étudier l'architecture de la figure 1, et nous allons analyser la tolérance aux fautes fournie par cette configuration en particulier.

Dans cette configuration, le cadre de discernement reste le même que dans le cas simple, et les masses de croyance associées aux hypothèses du problème sont les résultats des deux blocs de fusion F_1, et F_2. La combinaison de ces résultats se fait au niveau du bloc *fusion globale* avec la règle de Demspter selon les équations suivantes :

$$
\begin{aligned}
&m_{F1\oplus F2}(\{\exists\}) = \\
&\quad \frac{m_{F1}(\{\exists\}).m_{F2}(\{\exists\})+m_{F1}(\{\exists\}).m_{F2}(\Omega)+m_{F1}(\Omega).m_{F2}(\{\exists\})}{1-[m_{F1}(\{\exists\})m_{F2}(\{\nexists\})+m_{F1}(\{\nexists\})m_{F2}(\{\exists\})]} \\
&m_{F1\oplus F2}(\{\nexists\}) = \\
&\quad \frac{m_{F1}(\{\nexists\}).m_{F2}(\{\nexists\})+m_{F1}(\{\nexists\}).m_{F2}(\Omega)+m_{F1}(\Omega).m_{F2}(\{\nexists\})}{1-[m_{F1}(\{\exists\})m_{F2}(\{\nexists\})+m_{F1}(\{\nexists\})m_{F2}(\{\exists\})]} \\
&m_{F1\oplus F2}(\Omega) = 1-[m_{F1\oplus F2}(\{\exists\})+m_{F1\oplus F2}(\{\nexists\})]
\end{aligned}
\tag{11}
$$

1) Analyse des résultats:

a) Service de détection: De la même manière que dans le cas simple nous essayons de satisfaire les quatre conditions citées précédemment ; pour une valeur seuil $threshold = 0.8$, la figure 6 montre les couples de facteurs d'affaiblissement (p^P_{trust}, p^C_{trust}) des capteurs qui respectent la première et la seconde condition.

Les figures 7 et 8 montrent les couples de facteurs d'affaiblissement (p^P_{trust}, p^C_{trust}) des capteurs qui respectent la troisième condition. Pour la condition 3 dans le cas avec redondance, on a deux figures au lieu d'une seule car on a un double jeu de capteurs (deux capteurs caméra C_1, C_2 et deux capteurs de pression P_1, P_2), et la situation n'est plus la même suivant qu'une caméra ou un capteur de pression défaille. En particulier la figure 7 montre le cas où un des capteurs caméra est en panne, et la figure 8 montre le cas contraire, où un des capteurs pression est en panne.

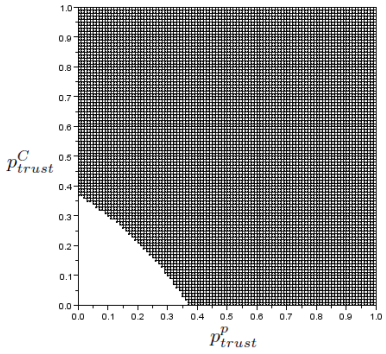

FIGURE 6. Couples p^i_{trust} satisfaisant les conditions 1 et 2

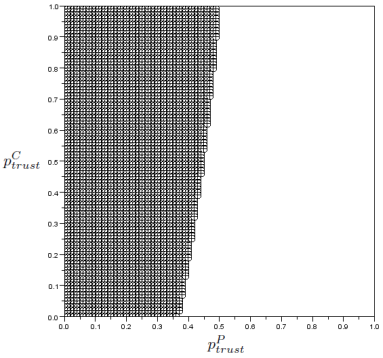

FIGURE 7. Couples p^i_{trust} satisfaisant les conditions 3 et 4 avec une caméra défaillante

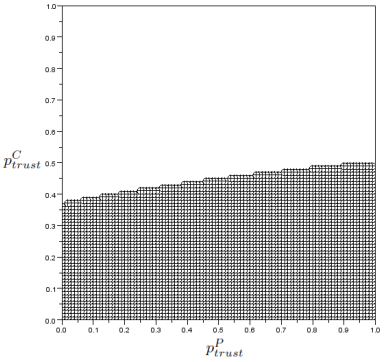

FIGURE 8. Couples p^i_{trust} satisfaisant les conditions 3 et 4 avec un capteur de pression défaillant

Dans les figures 7 et 8, la zone blanche correspond aux facteurs d'affaiblissement qui permettent de prendre une bonne décision. Or nous cherchons au contraire à ne pas décider dans ces cas afin de pouvoir détecter l'erreur.

Finalement les couples (p^P_{trust}, p^C_{trust}) respectant la quatrième condition sont les même que pour la troisième condition figures (7 et 8).

Par rapport au cas simple, on remarque que la surface de non-décision est plus petite. En effet, disposant maintenant de quatre capteurs, il y a plus de couples (p^P_{trust}, p^C_{trust}) qui permettent de prendre une décision.

Comme dans le cas simple, pour détecter une erreur dans

le système considéré les quatre conditions doivent êtres satisfaites. L'intersection des figures précédentes représentée en figure 9, donne les couples (p^P_{trust}, p^C_{trust}) permettant le service de détection.

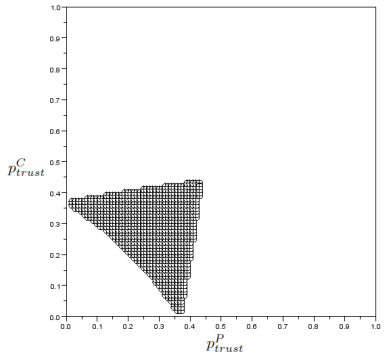

FIGURE 9. Service de détection pour un seuil 0.8

De même que pour le cas simple, on pourrait présenter les triplets (p^P_{trust}, p^C_{trust}, $threshold$) offrant le service de détection pour l'architecture complète.

b) Service de rétablissement par compensation: Ainsi que précisé en section II la compensation d'erreurs consiste à utiliser les redondances présentes dans le système pour déterminer le résultat correct en dépit de la présence d'une erreur.

Pour tolérer une erreur matérielle, l'architecture complète de la figure 1 doit tout d'abord détecter la présence d'une erreur, puis déterminer la sortie correcte à renvoyer. La détection est assurée de la même manière que précédemment : on choisit les paramètres satisfaisant le service de détection donnés en figure 9. Une fois l'erreur détectée, et sous l'hypothèse d'une erreur unique, il nous faut encore trouver laquelle des deux branches contient le capteur erroné. Si les deux branches respectent les critères de détection d'erreurs de la section VI-A, cela est facile : la branche contenant le capteur erroné n'aura pas réussi à décider, et l'autre branche aura abouti à la sortie correcte.

Le couple de facteurs d'affaiblissement permettant le service de rétablissement par compensation doit donc :

– permettre de détecter une erreur dans le cas complexe,
– et permettre de détecter une erreur dans le cas simple.

Il est ainsi l'intersection des figures 4 et 9 présentée en figure 10.

c) Service de masquage: L'architecture de la figure 1 peut tolérer une erreur matérielle (sortie d'un capteur erroné) sans la détecter. Pour ce faire il faut :

– que le système ait un comportement correct en l'absence d'erreur, ce qui revient à satisfaire les deux premières conditions représentées sur la figure 6,
– que le système aboutisse au résultat correct malgré la présence d'une erreur. Ceci correspond aux couples (p^P_{trust}, p^C_{trust}) des figures 11 et 12 facilement retrouvées à partir des figures 7 et 8 pour lesquels on décide de la présence (respectivement l'absence) d'une personne sur la chaise à bon escient.

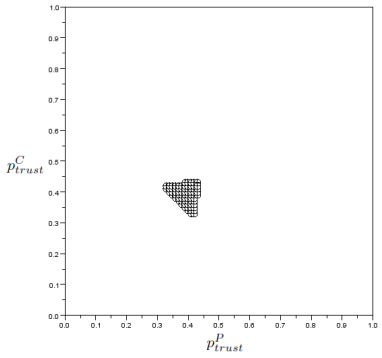

FIGURE 10. Service de compensation pour un seuil 0.8

Les couples (p^P_{trust}, p^C_{trust}) permettant le masquage de fautes sont donc ceux appartenant à l'intersection des figures 6, 11 et 12, représentée figure 13.

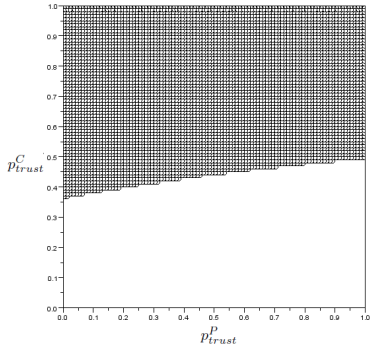

FIGURE 11. Le système aboutit au résultat correct malgré une erreur pression

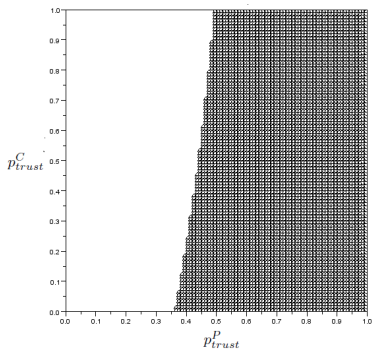

FIGURE 12. Le système aboutit au résultat correct malgré une erreur caméra

De même que précédemment, nous avons étudié les variations du paramètre *threshold*, que nous n'incluons pas ici pour des raisons de place.

d) Bilan: La figure 14 résume les différents services de tolérance aux fautes offerts par l'architecture complète de la figure 1. Ces services sont :

- **La détection d'erreur :** détecte une erreur matérielle.
- **Le rétablissement :** détecte et tolère une erreur matérielle.

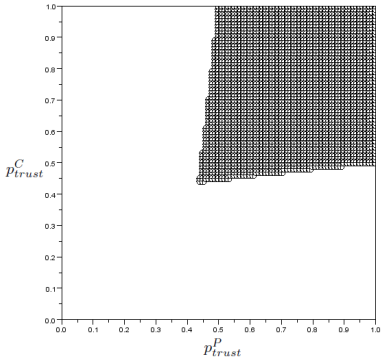

FIGURE 13. Service de masquage pour un seuil=0.8

- **Le masquage :** tolère, sans détection, une erreur matérielle.

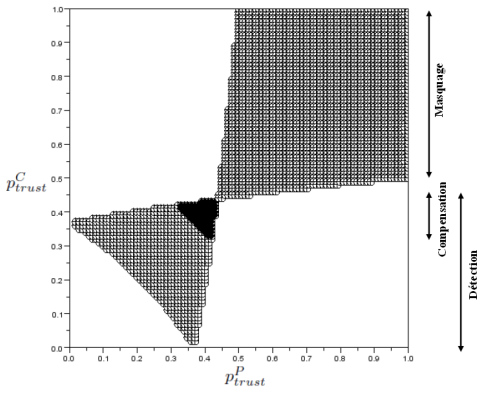

FIGURE 14. Services de tolérance aux fautes en fusion de données

Ces services peuvent êtres offerts dès lors que la fusion de données respecte les conditions de la figure14, identifiées lors d'une analyse formelle de son modèle. Dans le cas volontairement simple de notre étude théorique, cette analyse n'a pas montré de grandes difficultés. Cela pourrait ne pas être le cas dans des applications plus complexes où cette analyse pourrait même ne pas avoir de solutions. Seule une vraie diversification logicielle permettrait alors d'apporter des solutions en termes de tolérance aux fautes.

VII. CONCLUSION ET PERSPECTIVES

L'utilisation de la perception multi-capteurs commence à se répandre dans des applications critiques, ce qui soulève de plus en plus le problème de leur sûreté de fonctionnement. Nous avons présenté dans cet article un cas d'étude théorique simple permettant d'assurer des services de tolérance aux fautes en utilisant directement les mécanismes de fusion de données. A notre connaissance, cette étude est la première dans ce domaine. Nous avons montré qu'en analysant le modèle de la fusion de données et en choisissant avec attention certains paramètres, des services de détection, rétablissement et masquage d'erreurs physiques étaient possibles. Bien que notre analyse formelle n'ait pas posé de problèmes sur ce cas d'étude volontairement simple, elle pourrait s'avérer très difficile, voire impossible, sur des cas plus compliqués. Dans de futurs travaux, nous envisageons d'une part de porter ces résultats sur une application réelle tout en étudiant d'autres paramètres potentiels, comme le critère de combinaison (comparaison entre les differentes règles de combinaison). D'autre part, nous comptons étudier comment des mécanismes classiques de tolérance aux fautes pourraient assurer les mêmes services, et permettre de traiter en plus les erreurs logicielles, impossibles à tolérer sans diversification.

RÉFÉRENCES

[1] A. Avizienis, J.C. Laprie, B. Randell, and C. Landwehr. Basic concepts and taxonomy of dependable and secure computing. *Dependable and Secure Computing, IEEE Transactions on*, 1(1) :11–33, 2004.

[2] I. Bloch. Fusion d'informations numériques : panorama méthodologique. *Journées Nationales de la Recherche en Robotique*, pages 79–88, 2005.

[3] P. Caspi and R. Salem. Threshold and bounded-delay voting in critical control systems. In *Formal Techniques in Real-Time and Fault-Tolerant Systems*, pages 327–337. Springer, 2000.

[4] F. Delmotte. Detection of defective sources in the setting of possibility theory. *Fuzzy sets and systems*, 158(5) :555–571, 2007.

[5] F. Delmotte and G. Gacquer. Detection of defective sources with belief functions. In *Proceedings of the 12th Information Processing and Management of Uncertainty in Knowledge-Based Systems International Conference (IPMU'08), Torremolinos (Malaga), Spain, June*, pages 337–344, 2008.

[6] S. Démotier, W. Schon, and T. Denoeux. Risk assessment based on weak information using belief functions : a case study in water treatment. *Systems, Man, and Cybernetics, Part C : Applications and Reviews, IEEE Transactions on*, 36(3) :382–396, 2006.

[7] A. Dempster. Upper and lower probabilities induced by a multivalued mapping. *Classic Works of the Dempster-Shafer Theory of Belief Functions*, pages 57–72, 2008.

[8] D. Dubois, H. Prade, and H. Farreny. *Théorie des possibilités : applications à la représentation des connaissances en informatique*, volume 1. Masson, 1988.

[9] PJ Escamilla-Ambrosio and N. Mort. A hybrid kalman filter-fuzzy logic multisensor data fusion architecture with fault tolerant characteristics. In *Proceedings of the 2001 international conference on artificial intelligence*, pages 361–367, 2001.

[10] J.C. Laprie. Sûreté de fonctionnement des systèmes : concepts de base et terminologie : Sûreté de fonctionnement. *REE. Revue de l'électricité et de l'électronique*, (11) :95–105, 2004.

[11] V. Ricquebourg, M. Delafosse, L. Delahoche, B. Marhic, AM Jolly, and D. Menga. Combinaison de sources de données pour la détection de dysfonctionnement capteur. LFA, 2007.

[12] G. Shafer. *A mathematical theory of evidence*, volume 1. Princeton university press Princeton, 1976.

[13] P. Smets. Constructing the pignistic probability function in a context of uncertainty. In *Uncertainty in artificial intelligence*, volume 5, pages 29–39, 1990.

[14] S. Zug and J. Kaiser. An approach towards smart fault-tolerant sensors. In *Robotic and Sensors Environments, 2009. ROSE 2009. IEEE International Workshop on*, pages 35–40. IEEE, 2009.