Architecture de filtre de Kalman tolérante aux fautes pour la localisation des robots mobiles

Fault Tolerant Kalman Filter Architecture For Mobile Robots Localization

BADER Kaci. LUSSIER Benjamin et SCHÖN Walter
Université de Technologie de Compiègne
UMR CNRS 7253, Heudiasyc BP 20529

Résumé

La localisation précise est une fonctionnalité importante dans les robots autonomes et les véhicules intelligents. Elle utilise différents capteurs pour déterminer une position, ce qui est fondamental pour la navigation et le contrôle. Dans cet article, nous proposons une architecture de tolérance aux fautes adaptée à la fusion de données et les détails de son application pour la localisation d'un robot mobile. Nous utilisons deux types de capteurs pour percevoir l'état du robot et l'environnement: une unité de mesure inertielle (IMU) qui donne les accélérations et les vitesses angulaires du robot, et une caméra qui fournit des séquences d'images pour un algorithme d'odométrie visuelle. Un filtre de Kalman utilise ces entrées pour estimer la position du robot. La tolérance aux fautes est fournie dans cette application par une duplication / comparaison des algorithmes de diagnostic appropriés. La technique d'injection de fautes est utilisée pour évaluer les performances de notre architecture sur une étude de cas simulée.

Summary

Precise localization is an important functionality in autonomous robots and vehicles. It uses various sensors to determine a position, fundamental feature for navigation and control. In this paper we propose a fault tolerance architecture adapted to data fusion and detail its implementation for the localization of a mobile robot. We use two different types of sensors to perceive the robot state and environment: an Inertial Measurement Unit (IMU) gives the accelerations and angular velocities of the robot, while a camera provides sequences of images for a visual odometry algorithm. A Kalman filter uses these inputs to estimate the position of the robot. Fault tolerance is provided in this application by a duplication / comparison and appropriate diagnostic algorithms. The fault injection technique is used to evaluate the performance of our architecture on a simulated case study.

Introduction

La fusion de données multi-capteurs est fondamentale pour la perception des systèmes autonomes et fait donc l'objet de nombreux domaines et d'activités de recherche. Telle que décrite dans (Varshney et al, 1997) la fusion de données multi-capteurs se réfère à l'acquisition, le traitement, et la combinaison des informations recueillies par diverses sources de connaissances et de capteurs pour fournir une meilleure compréhension du phénomène étudié. Ces techniques profitent de la diversité des capteurs redondants et complémentaires afin de réduire les incertitudes et les imprécisions dans les données extraites. Cependant multiplier les capteurs et les algorithmes de fusion de données sous-jacents augmente par conséquent les risques de fautes matérielles et logicielles. Par ailleurs, la validation de ces mécanismes se heurte à deux problèmes majeurs : tout d'abord, en raison de leur programmation déclarative, le comportement des algorithmes de fusion est difficile à prévoir, ce qui les rend difficiles à valider par des approches formelles, comme la vérification par la preuve. Deuxièmement, l'environnement ouvert auquel les systèmes robotiques complexes sont confrontés génère un contexte d'exécution quasi-infini ce qui fait du test une tâche difficile et couteuse. Dans cet article, nous nous concentrons donc sur la tolérance aux fautes comme une alternative à la validation des mécanismes de fusion de données multi capteurs : comme il est difficile de supprimer toutes les fautes dans le système, nous cherchons à limiter leurs impacts sur ses fonctions. Dans notre approche, nous proposons une architecture basée sur la duplication / comparaison pour détecter et diagnostiquer les fautes dans un mécanisme de fusion de données qui utilise un filtre de Kalman sur deux types de capteurs pour estimer la position d'un robot mobile. Nous ciblons particulièrement les fautes matérielles sur les capteurs et les fautes logicielles dans le modèle et l'algorithme de filtre de Kalman.

Ce papier est organisé comme suit. Dans la section 2, nous présentons la tolérance aux fautes et la fusion de données, en particulier les filtres de Kalman, ainsi qu'un état de l'art. Dans la section 3, nous décrivons l'architecture système de robot simulé ainsi que le filtre de Kalman implémenté. La section 4 présente notre architecture de tolérance aux fautes et détaille les services de détection et de recouvrement offerts. Dans la section 5 nous décrivons notre étude de cas, et nous montrons les résultats simulés de notre architecture en utilisant l'injection de fautes. La section 6 conclut le papier, et donne quelques perspectives pour les travaux futurs.

Tolérance aux fautes et fusion de données

Dans cette section, nous donnons un aperçu des deux domaines abordés dans cet article : la tolérance aux fautes, et la fusion de données multi capteurs, puis nous présentons un état de l'art sur la tolérance aux fautes en fusion de données.

1 Tolérance aux fautes

La sureté de fonctionnement d'un système est sa capacité à fournir un service auquel on peut accorder une confiance justifiée (Avizienis et al, 2004), (`Laprie, 1992`). Cette notion englobe trois concepts différents : ses attributs (propriétés attendues du système), ses entraves (comportements inacceptables du système qui sont causes ou conséquences de la non sûreté de fonctionnement), et 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). La tolérance aux fautes est l'une de ces méthodes, qui vise à fournir des services corrects malgré la présence de fautes. Elle est généralement mise en œuvre grâce à la détection d'erreur et au rétablissement du système.

- *La détection d'erreurs* vise à détecter l'état erroné du système avant qu'il ne se propage jusqu'à provoquer une défaillance du système. Elle est communément réalisée par trois méthodes : duplication / comparaison, chien de garde temporel et contrôles de vraisemblance. Dans notre architecture, nous allons utiliser la *duplication / comparaison*, qui détecte les erreurs en comparant les résultats à partir d'au moins deux unités fonctionnellement identiques et indépendantes vis-à-vis des fautes à tolérer,

- *Le rétablissement du système* permet au système de retrouver un état fonctionnel lorsqu'une erreur a été détectée. Il est principalement réalisé par la gestion des erreurs grâce au recouvrement par poursuite, ou par compensation. Dans notre architecture, nous allons utiliser la *compensation d'erreur* où l'état erroné contient une redondance suffisante pour permettre sa transformation en un état correct.

2 Fusion de données et filtre de Kalman

La fusion de données (Bloch et al, 2001) consiste à combiner des informations qui proviennent de plusieurs sources et à exploiter ces informations dans diverses tâches telles que répondre à des questions, prendre des décisions, estimer des valeurs numériques, etc. Ces sources peuvent être des capteurs physiques observant la réalité et fournissant des informations différentes sur des événements possibles. D'autres définitions de la notion de fusion de données se trouvent dans (Bostrom et al, 2007). La fusion de données multi-capteurs combine plusieurs mesures de capteurs pour former une meilleure représentation de l'environnement observé, le *modèle du monde*. Elle 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).
Généralement trois grands cadres théoriques sont utilisés pour mettre en œuvre un mécanisme de fusion des données: la théorie des probabilités, la théorie des possibilités (Dubois et al, 1994), et la théorie des fonctions de croyance (Dempster, 1967), (Shafer, 1976). Dans ces travaux, nous nous concentrons sur le cadre probabiliste, et en particulier la méthode du filtre de Kalman.

Le filtre de Kalman a été inventé par R.E. Kalman et publié pour la première fois en 1960 (Kalman et al, 1960). Cette méthode a été largement appliquée dans de nombreuses applications robotiques (comme la navigation autonome, le suivi de cibles et la localisation). Les algorithmes de fusion à base de filtre de Kalman consistent à estimer l'état d'un système inconnu. Ce résultat est obtenu grâce à une série de calculs récurrents fournissant une meilleure estimation des variables d'état du système avec une correction proportionnelle à l'erreur entre une prédiction et les sorties des mesures de capteurs. Dans sa forme originale, le filtre de Kalman utilise un modèle du système pour la prédiction, et un ensemble de données de capteurs pour la correction [13]. Etant donné que la navigation est essentielle pour les robots mobiles, les filtres de Kalman ont pendant longtemps été utilisés dans ce type d'application (Choi et al, 2007) (Freeston et al, 2002), et (Congwei Hu et al, 2003). Comme le filtre de Kalman d'origine ne peut être appliqué qu'à des systèmes linéaires, le filtre de Kalman étendu (EKF) a été proposé pour les systèmes non linéaires. Cet EKF a été mis en œuvre avec succès pour l'estimation de position de robots dans [12], (Casanova et al, 2008), et (Son-Goo Kim et al, 2007).

3 La tolérance aux fautes en fusion de données par filtre de Kalman

La tolérance aux fautes apparaît dans de nombreuses études de filtre de Kalman. Les différentes approches proposées dans la littérature utilisent principalement la tolérance aux fautes par duplication / comparaison : elles utilisent un modèle analytique du système comme diversification des capteurs ciblés, puis le filtre de Kalman fusionne l'état estimé par ce modèle et les mesures données par les capteurs. Dans (Escamilla-Ambrosio et al, 2001) un filtre de Kalman est utilisé pour proposer une architecture hybride de fusion de données multi-capteurs intégrant le filtrage de Kalman et la technique de logique floue. Les principales caractéristiques de cette architecture sont sa tolérance aux fautes permanentes et transitoires des capteurs. Les fautes transitoires sont détectées en utilisant une technique de résidus, et les fautes permanentes sont détectées par une méthode de vote (Caspi et al, 2000). Dans (Zug et al, 2009) les auteurs proposent une architecture de tolérance aux fautes de capteurs. Cette approche utilise un système de capteurs abstrait (Marzullo, 1990) incorporé dans un capteur virtuel pour améliorer les caractéristiques du capteur en présence de bruit et de défaillances. En utilisant un modèle mathématique pour évaluer les données de capteurs redondants, cette approche permet d'obtenir une estimation fiable de position. (Allerton, 2008) présente les approches de fusion de données pour les systèmes de réseau inertiels. Dans ces systèmes, les auteurs utilisent des nœuds redondants qui sont structurés hiérarchiquement. La plupart des nœuds jouent le rôle d'esclaves tandis que l'un d'entre eux joue le rôle de maître, réalisant la fusion globale. Différentes matrices de transformation (rotation et translation) sont développées pour assurer l'alignement des mesures. Un filtre de Kalman est utilisé comme algorithme de fusion. Les auteurs utilisent la redondance des IMU pour assurer le masquage de fautes et avoir des sorties plus précises.

À notre avis une lacune importante de ces différentes techniques est qu'elles se concentrent uniquement sur les fautes matérielles, en s'appuyant sur les composants de filtre de Kalman pour détecter et tolérer les erreurs. En outre, ces filtres sont difficiles à concevoir et à valider formellement, car il n'y a pas de cadre de conception générique pour les modèles utilisés dans les filtres de Kalman. Leur conception dépend en effet de chaque application. De plus ces mécanismes de fusion incorporent souvent des valeurs déterminées empiriquement comme les gains, ou les covariances des mesures. En l'absence d'analyse formelle, la fiabilité des filtres de Kalman peut donc largement être mise en question. Les défaillances logicielles sont ainsi pour nous une grande préoccupation dans la fusion de données, et l'architecture proposée vise à tolérer les fautes matérielles dans les capteurs et les fautes logicielles dans les mécanismes de fusion.

Architecture Système

Dans cette section, nous présentons notre robot, nous décrivons les capteurs utilisés pour la perception de son état (IMU) ainsi que son environnement (caméra), et nous détaillons le filtre de Kalman mis en œuvre pour assurer sa localisation.

1 Robot mobile

Notre application est un robot mobile autonome, appelé WIFIBOT. La navigation dans un robot mobile requiert un certain nombre de fonctions, y compris la localisation, qui consiste à estimer la position du robot. Cette localisation est réalisée dans notre robot grâce à un algorithme de filtrage de Kalman combinant les données d'une unité de mesure inertielle (IMU) et d'un algorithme d'odométrie visuelle à partir d'une caméra comme illustré à la Figure 1. Cette méthode est appropriée en ce qui concerne la charge limitée de notre robot et son champ de fonctionnement requis (intérieur et extérieur)

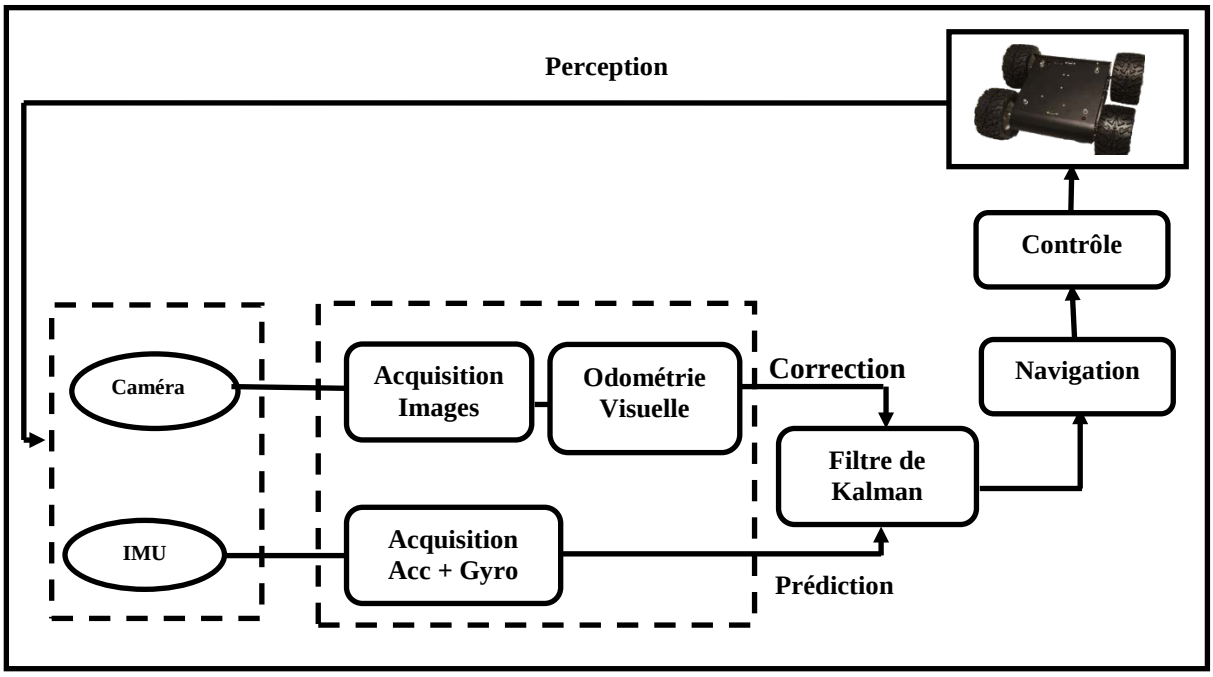

Figure 1. Architecture de Navigation

2 Unité de mesure inertielle

Dans notre architecture, nous utilisons des centrales inertielles IMU (Inertial measurement unit), qui fournissent les accélérations et les vitesses angulaires selon les trois axes. Dans notre cas d'application nous utilisons seulement l'accélération longitudinale et la vitesse angulaire autour de l'axe z.

3 Estimation de la position par la caméra et l'odométrie visuelle:

Les caméras sont des capteurs largement utilisés dans les systèmes autonomes, pour la perception des objets et leur identification, la localisation et la cartographie. Ce type de capteur fournit des données à un faible coût qui peut être utilisé pour de nombreuses applications différentes. Dans notre architecture, une caméra à bord du robot délivre une séquence d'images utilisées comme entrée pour un algorithme d'odométrie visuelle (Nistér et al, 2004). L'idée générale de l'odométrie visuelle est de détecter les similitudes entre deux images successives et déterminer les changements dans la position du robot à partir de ces points d'intérêt.

4 Fusion de données par filtre de Kalman:

Notre filtre de Kalman est un prédicteur-correcteur. Il utilise les sorties de l'IMU (après intégration) pour prédire l'état du système $Pos_{FK}^{p} = (x_{FK}^{p}, y_{FK}^{p}, \theta_{FK}^{p})$, il met à jour cet état prédit en utilisant les mesures de l'algorithme d'odométrie visuelle $Pos_C = (x_C, y_C)$. La sortie du filtre de Kalman est la position du robot corrigée $Pos_{FK}^{c} = (x_{FK}^{c}, y_{FK}^{c}, \theta_{FK}^{c})$ à chaque instant de son déplacement. Le processus de fusion est représenté sur la Figure 2. Les détails d'implémentation de cette configuration sont les suivants :

- **Modélisation :** La reconstruction de la trajectoire du robot se fait par l'intégration de la vitesse linéaire au fil du temps en fonction de la représentation cinématique bien connue d'un robot mobile à deux roues se déplaçant dans un plan (équation 1).
  Pour mettre en œuvre un filtre de Kalman, nous devons d'abord modéliser le problème en fonction des paramètres à estimer (équation 1) et les mesures de capteurs (équation 2) :

  - *Modèle du Système*: on utilise l'équation (1) comme modèle du système, prenant comme entrée $U_{IMU}(V, \omega)$, avec V la vitesse linéaire issue de l'intégration de l'accélération longitudinale a mesurée par l'accéléromètre, et ω la vitesse angulaire mesurée par le gyromètre selon l'axe z.

$$\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix} + \begin{bmatrix} \cos\theta_{t-1}\,\Delta t & 0 \\ \sin\theta_{t-1}\,\Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \cdot \begin{bmatrix} V_t \\ \omega_t \end{bmatrix} + w_t \qquad \{1\}$$

Avec x_t : position du robot sur l'axe des x, y_t : position du robot sur l'axe des y, θ_t : son orientation, Δt : Période d'échantillonnage, V_t : vitesse linéaire du robot, ω_t vitesse angulaire du robot.

$$Avec : V_t = V_{t-1} + a_t\,\Delta t \qquad \{2\}$$

- *Modèle d'observation* : Le modèle d'observation est basé sur les images de la caméra et l'algorithme d'odométrie visuelle. Il est mis en œuvre de la manière suivante (équation 3) :

$$Z_t = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} . \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} + v_t \qquad \{3\}$$

$avec$: w_t et v_t sont les vecteurs de bruits gaussiens de moyenne nulle et de covariance Q et R respectivement.

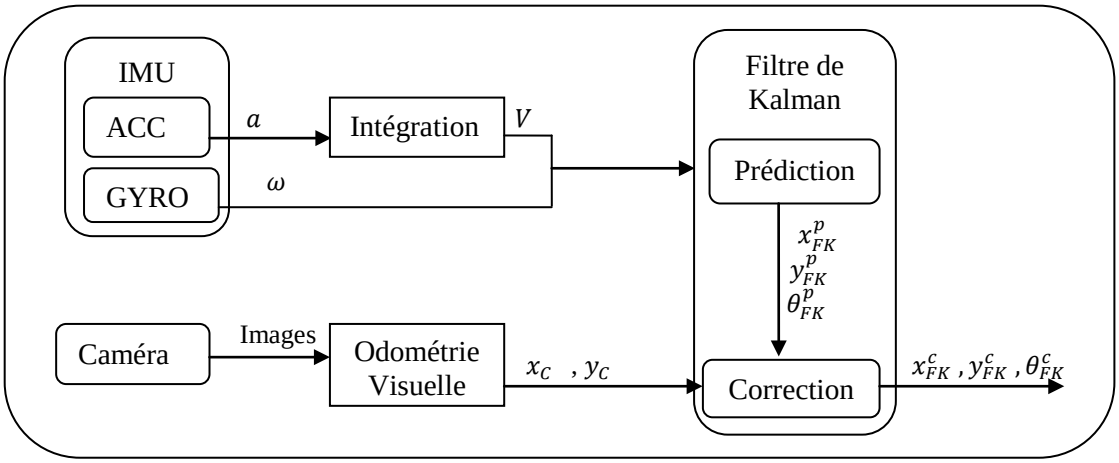

Figure 2. Filtre de Kalman pour la navigation inertielle assistée par l'odométrie visuelle

➢ **Application du filtre de Kalman :** Comme on le voit sur la Figure 2, notre filtre de Kalman consiste en deux étapes successives, la prédiction et la correction. La phase de prédiction utilise l'état estimé de l'instant précédent et les mesures de la centrale inertielle IMU pour produire une estimation de l'état courant. Dans l'étape de correction, les observations de l'instant courant après exécution de l'algorithme d'odométrie visuelle sont utilisées pour corriger l'état prédit dans le but d'obtenir une estimation plus précise.

1. **Étape de prédiction**: L'état corrigé $PosKF^c_{t-1}$ à l'instant précédent *t-1* et la mesure U^{IMU}_{t-1} perçue par la centrale inertielle IMU sont utilisés pour prédire l'état courant $PosKF^p_t$ à l'instant *t* et son incertitude P^p_t suivant l'équation (4).

$$\begin{cases} PosKF^p_t = PosKF^c_{t-1} + BU^{IMU}_{t-1} \\ P^p_t = A\,P^c_{t-1}\,A^T + Q \end{cases} \qquad \{4\}$$

Avec : Q la matrice de covariance du bruit de modèle du système.

2. **Etape de correction**: Une fois la mesure $Z_t = (x_c, y_c)$ de l'algorithme d'odométrie visuelle disponible, l'état prédit $PosKF^p_t$ peut être corrigé par l'introduction du résidu r_t pondéré par le gain du filtre de Kalman K_t, pour obtenir ainsi une estimation plus précise $PosKF^c_t$ et sa matrice de covariance P^c_t. Cette correction est effectuée suivant les équations (5).

$$\begin{cases} r_t = Z_t - H.PosKF^p_t \\ K_t = P^p_t.H^T.(H.P^p_t.H^T + R)^{-1} \\ PosKF^c_t = PosKF^p_t + K_t\,.r_t \\ P^c_t = (Id - K_tH)P^p_t \end{cases} \qquad \{5\}$$

Avec : R la matrice de covariance du bruit de mesure et H la matrice d'observation liant les mesures a l'état du système.

Noter que de notre point de vue, les paramètres du modèle A, B, Q et R (équations 4 et 5) sont particulièrement sensibles aux fautes de conception. En effet, Q et R sont généralement déterminés empiriquement après certaines expériences, et A et B font partie du modèle qui et peuvent être difficile à valider.

Architecture de tolérance aux fautes

L'architecture de tolérance aux fautes (Figure 3) proposée dans ce papier est basée sur la méthode classique de duplication / comparaison. Cette architecture implémente deux branches parallèles chacune exécutant un filtre de Kalman (utilisant une caméra et une centrale inertielle) décrits précédemment dans la section 3.4 pour l'estimation de la position d'un robot mobile. Une telle configuration peut tolérer une faute matérielle liée aux capteurs de perception et détecter une faute logicielle dans le filtre de Kalman, mais nécessite des moyens de comparer les résultats de chaque composant dupliqué (capteurs et filtres de Kalman) avec son redondant. Dans notre cas, les sorties des mécanismes de fusion sont comparées pour détecter une erreur, et les sorties des capteurs sont utilisées pour diagnostiquer l'erreur détectée et déterminer la sortie correcte du système de perception considéré. Dans cette section nous présentons les services de détection et de recouvrement d'erreurs de notre architecture, dans

l'hypothèse de la faute simple (matérielle ou logicielle). Ces services pourraient être étendus à des fautes multiples en augmentant le niveau de redondance de l'architecture.

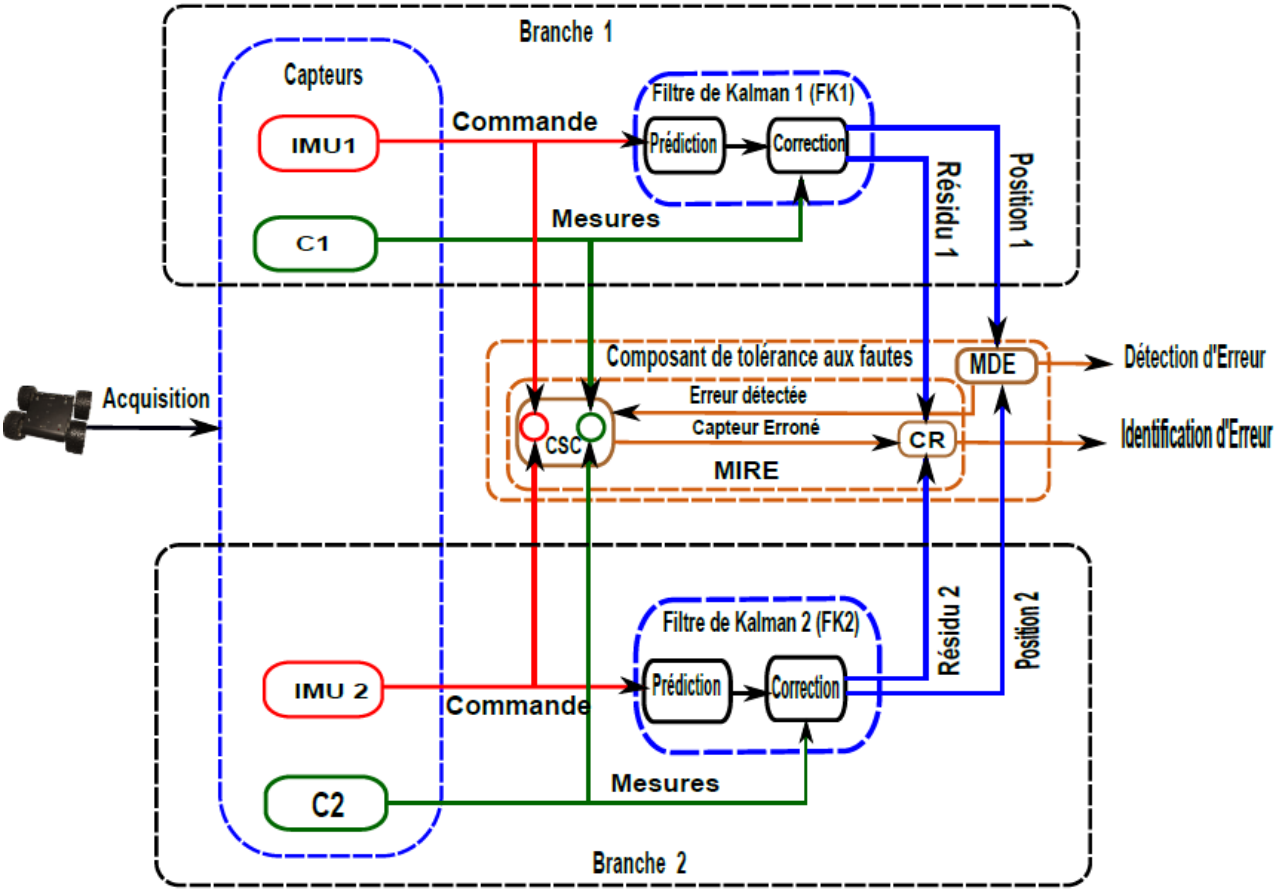

Figure3. Architecture de tolérance aux fautes matérielles et logicielles

1 Services de détection d'erreurs et de rétablissement du système

Notre architecture implémente deux branches parallèles, chacune exécutant un filtre de Kalman qui utilise deux capteurs (une centrale inertielle IMU et une caméra) pour estimer la position du robot. Les résultats de chacun des composants dupliqués sont comparés à ceux de son redondant diversifié : Par exemple sur la Figure 3, la sortie de FK1 est comparée à la sortie de FK2, la sortie du capteur IMU1 est comparée à la sortie de IMU2, et la sortie de la camera C1 est comparée à celle de la caméra C2. Les sorties des deux filtres de Kalman sont comparées pour détecter une erreur dans le système, et les sorties des capteurs ainsi que les résidus de filtres de Kalman sont comparés pour diagnostiquer l'erreur détectée et déterminer (en cas de faute matérielle) la sortie correcte du système de perception. Notons que la comparaison des sorties des caméras ne se fait pas directement sur l'image, ce qui serait complexe et coûteux en temps de calcul, mais sur la position qui en est extraite par odométrie visuelle. Notre architecture se compose ainsi de deux modules :

- ***Module de détection d'erreur (MDE)*** : il détecte les erreurs possibles dans le système en comparant les sorties des deux filtres de Kalman. Cette comparaison consiste à calculer une distance $dis_{(FK1\text{-}FK2)}$ entre les sorties des deux filtres de Kalman, et à la comparer à un seuil de détection spécifique $Seuil_{Det}$.

- ***Module d'identification et de recouvrement d'erreur (MIRE)*** : il diagnostique l'erreur détectée précédemment, et déduit, en cas de faute matérielle, la sortie correcte du système. Ce diagnostic se fait en deux étapes :
  - *Comparaison des sorties des capteurs* (CSC): elle identifie, en cas de faute matérielle, le type de capteur erroné en comparant les sorties des capteurs similaires.
    - Si la sortie d'un capteur s'écarte significativement de son dual, le système diagnostique une faute matérielle sur l'un de ces deux capteurs de même type. Le capteur défectueux sera alors identifié dans la comparaison des résidus.
    - Si les sorties des capteurs sont similaires, le système diagnostique une faute logicielle. En l'absence de plus de redondance, il n'est pas possible de pousser plus loin le diagnostic (identifier le filtre fautif) mais seulement de mettre le système erroné dans un état sûr. Cependant un autre filtre de Kalman FK3 à l'aide de deux sorties diversifiées parmi les IMU et les caméras pourrait être mis en œuvre pour diagnostiquer le filtre défectueux, et rétablir le système.

  - *Comparaison des résidus* (CR): elle identifie dans le cas d'une faute matérielle la branche erronée d'après la valeur des résidus des filtres de Kalman. En effet, ce résidu correspond au degré de consistance entre les données fusionnées, une valeur haute indiquant que les données des capteurs fusionnés sont en conflit. Nous nous appuyons ici sur le mécanisme de fusion de données, car nous avons déjà vérifié dans l'étape « *Comparaison des sorties des capteurs* » que l'écart constaté entre les deux branches est dû à des capteurs et non aux mécanismes de fusion

Connaissant maintenant le type de capteur défectueux et la branche erronée correspondante, le système a identifié le capteur erroné, et peut ainsi être rétabli à l'aide des valeurs de position de la branche sans erreur.

2 Analyse Qualitative

L'architecture proposée fournit les services de tolérance aux fautes suivantes :

- Détection et recouvrement d'une faute matérielle : elle détecte et recouvre une erreur matérielle dans les capteurs de perception (IMU1, IMU2, C1 ou C2).
- Détection d'une faute logicielle : elle détecte une erreur logicielle dans les algorithmes de filtre de Kalman (FK1, FK2). Cette faute ne peut être tolérée que par un troisième Filtre de Kalman diversifié et la méthode de vote majoritaire.

Dans ce papier, nous travaillons sur l'hypothèse de faute unique (pas plus d'une seule faute active en même temps dans le système), mais cette hypothèse pourrait être levée en utilisant plus de redondance, au niveau matériel et logiciel.

Simulation et évaluation de performances

Dans cette section, nous présentons une étude de cas par simulation pour valider les mécanismes de tolérance aux fautes proposés. Cette validation est basée sur l'injection de fautes : l'introduction délibérée de fautes dans le système pour observer son comportement et l'efficacité des mécanismes de tolérance aux fautes.

Nous avons fixé le temps d'échantillonnage à t = 0.5 seconde, et généré un bruit aléatoire pour chacun des capteurs simulés suivant une distribution normale, avec des valeurs N (0, 0.1) pour les IMU et N (0; 0.5) pour les cameras. Les matrices de covariances des bruits de modèle et de mesure seront donc respectivement Q = 0.01 et R = 0.25. Les différents seuils sont définis expérimentalement pour ne pas déclencher de fausses détections dans les cas nominaux.

Nous présentons d'abord le comportement nominal du système, puis les résultats de nos expériences avec deux fautes matérielles distinctes injectées.

1 Comportement nominal

Une simulation sans fautes injectées a été réalisée pour caractériser le comportement nominal. La Figure 4 représente la position du robot estimée par les deux filtres de Kalman (Pos^c_{FK1},Pos^c_{FK2}) et la sortie de notre architecture Pos_S: la moyenne de ces deux estimations. La Figure 5 représente la distance $dis_{(FK1\text{-}FK2)}$ entre les positions estimées par les deux filtres. Dans ce cas nominal, elle est inférieure au seuil de détection choisi ($Seuil_{Det}$ =4 m). Quatre autres seuils sont nécessaires :

- Un seuil pour comparer les positions issues des algorithmes d'odométrie visuelle : $Seuil_C$.=5
- Deux seuils pour comparer les sorties des deux IMU : $Seuil_\omega = 1.5.$, $Seuil_V = 1.5.$.
- Un seuil concernant les résidus pour assurer dans le cas d'erreur matérielle, que le conflit dans un filtre de Kalman est significativement plus élevé que dans l'autre: $Seuil_r$. = 4m.

Tous les seuils ont été choisis pour ne pas provoquer de fausses détections dans le comportement nominal.

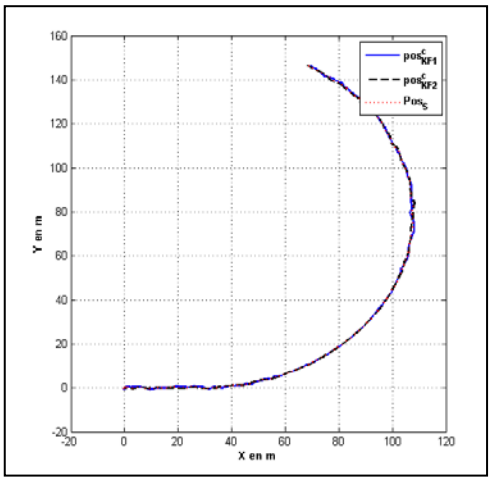

Figure 4. Comportement nominal : positions estimées

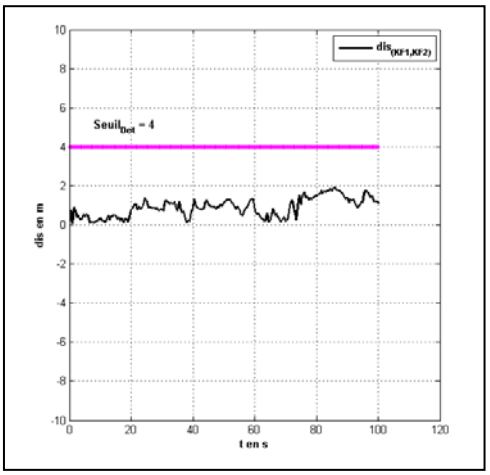

Figure 5. Comportement nominal : Distance entre la position des deux filtres

2 Comportement en présence de fautes

Nous présentons ici les résultats d'injections de deux fautes matérielles sur la camera C1 et la centrale inertielle IMU2.

- ***Trame bloquée sur la camera C1 :*** La première faute injectée que nous considérons est une faute « *trame bloquée* » dans la sortie de la camera C1 à l'instant *t = 49.5 seconde*, ayant pour conséquence que l'algorithme d'odométrie visuelle associé est bloqué en une position constante.

  Sur la Figure 6, nous observons que la distance $dis_{(FK1\text{-}FK2)}$ entre les positions estimées par les deux filtres de Kalman dépasse le seuil de détection $Seuil_{Det}$, et permet ainsi au système de détecter la présence d'une erreur à l'instant t=52,5 secondes. Nous voyons également sur la Figure 7 que les deux IMU ont des sorties similaires, tandis que la différence entre les positions issues des deux caméras C1 et C2 dépasse le seuil $Seuil_C$. La Figure 8 montre que la valeur absolue du résidu de FK1 | r1 | est significativement plus grande que la valeur absolue du résidu de FK2 | r2 |, indiquant que la branche FK1 contient une erreur. Connaissant maintenant le couple (C1,

C2) contenant le capteur défectueux et la branche erronée, le système peut identifier C1 comme capteur erroné, et prendre la sortie correcte Pos^{ε}_{FK2}à t =52,5 secondes comme indiqué dans la Figure 9, 3 secondes après l'activation de la faute.

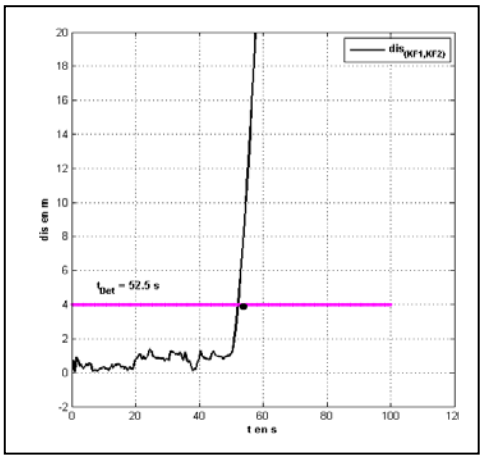

Figure 6. Trame bloquée dans C1: Distance entre les positions des deux filtres

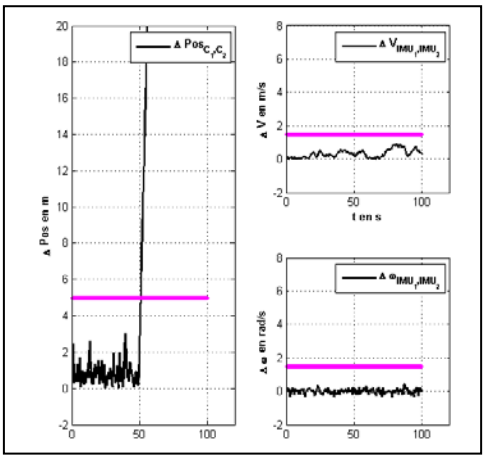

Figure 7. Trame bloquée dans C1: Comparaison des capteurs

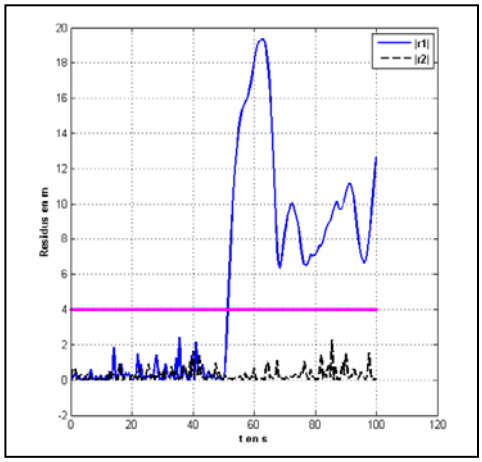

Figure 8. Trame bloquée dans C1: Comparaison des résidus

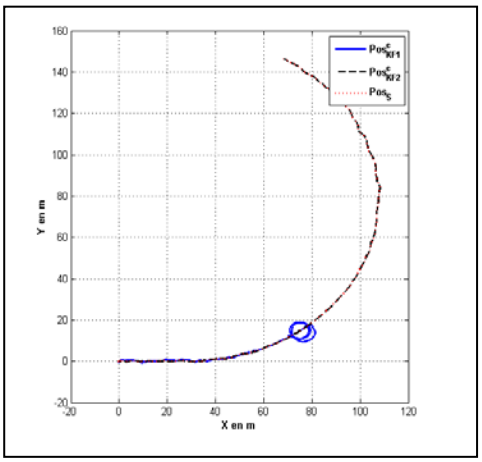

Figure 9. Trame bloquée dans C1: positions estimées

- ***Biais permanent dans IMU 2 :*** La seconde faute injectée est une faute de *biais permanent* dans l'accélération longitudinale mesurée par la centrale inertielle IMU2. C'est une faute fréquente dans ce type de capteurs. A l'instant t=39.5 seconde, nous ajoutons un biais de $0.1m/s^2$.
  Les Figures 10,11, 12, et 13 montrent que la faute est correctement détectée, diagnostiquée et tolérée à t = 57 seconde, à savoir 17.7 seconde après l'activation de la faute. Cette fois, ce sont les sorties des deux IMU qui sont dissimilaires et le résidu de FK2 |r2| qui est supérieur à la valeur absolue du résidu de FK1 |r1|, indiquant que la centrale inertielle IMU2 est erronée. Le laps de temps pour la détection est du au fait que, bien que la faute ait été injectée à t = 39.5 seconde, la position du filtre de Kalman FK2 prend un certain temps pour dériver et diverger de manière significative de celle estimée par le filtre de Kalman FK1 (environ 17.7 seconde)

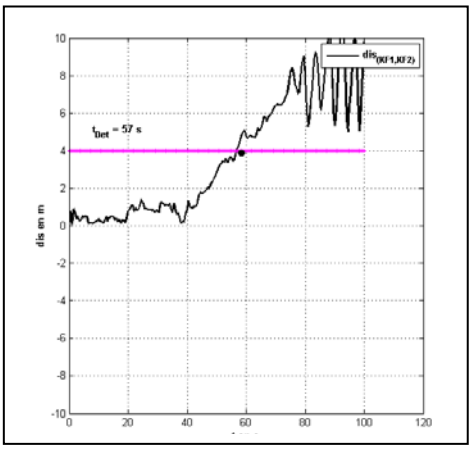

Figure 10. Biais permanent dans IMU2: Distance les positions des deux filtres

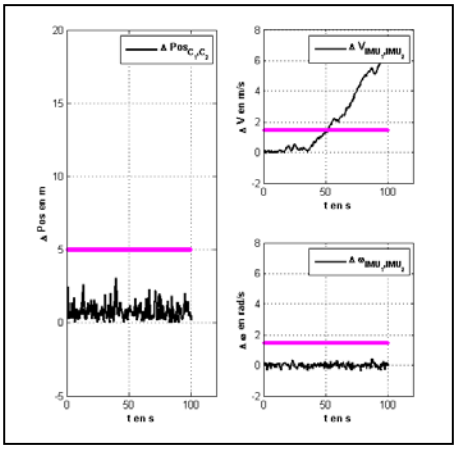

Figure 11. Biais permanent dans IMU2: Comparaison des capteurs

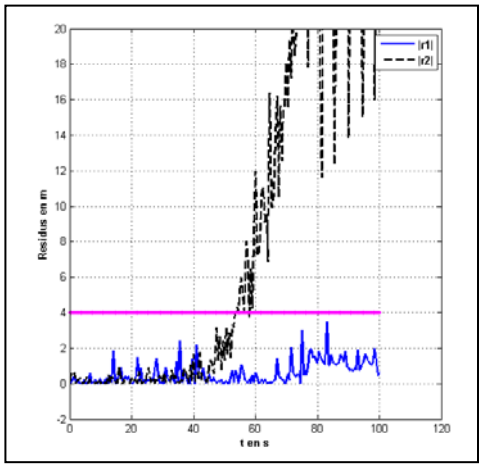

Figure 12. Biais permanent dans IMU2: Comparaison des résidus

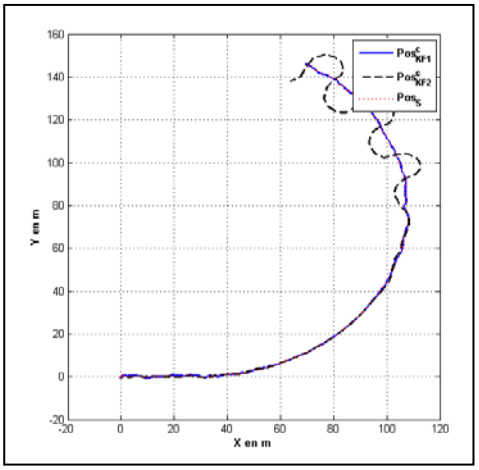

Figure 13. Biais permanent dans IMU2:positions estimées

La détection d'erreur n'est donc pas instantanée, du fait que la sortie du bloc du filtre de Kalman erroné peut prendre temps significatif pour s'écarter de celui qui est correct. Cela ne soulève pas de problème en ce qui concerne la sortie du système, car pendant ce temps la différence entre les deux sorties (correcte et erronée) des deux blocs (FK1, FK2) est inférieure au seuil de détection $Seuil_{Det}$, une valeur admissible. Cependant, hors de l'hypothèse de faute unique que nous avons considérée dans ce travail, l'activation d'une autre faute pendant cette période pourrait rendre le système incapable de détecter et de diagnostiquer les erreurs.

Conclusion

Nous avons présenté dans cet article une architecture de filtre de Kalman tolérante aux fautes basée sur la duplication / comparaison, et un exemple d'application pour la localisation d'un robot mobile. Nous avons également illustré son algorithme sur un exemple utilisant la simulation et l'injection de deux fautes matérielles. L'algorithme donné détecte aussi les fautes logicielles grâce à la diversification du filtre de Kalman (modèle et covariances de bruit), et pourrait les tolérer avec une redondance supplémentaire. Cependant cette architecture nécessite de comparer les sorties des capteurs dupliqués.
Pour nos travaux futurs nous envisageons :

- d'implémenter l'architecture proposée sur un robot réel,
- d'injecter des fautes logicielles pour valider leur détection,
- d'automatiser l'injection de fautes pour considérer un grand nombre de fautes pour continuer la validation de cette architecture,
- d'éliminer l'hypothèse de faute unique en ajoutant plus de redondance, un nouveau bloc redondant est nécessaire pour chaque faute supplémentaire qu'on veut tolérer,
- d'analyser formellement le modèle du filtre de Kalman pour comparer sa tolérance aux fautes intrinsèque à celle de notre architecture,
- d'implémenter un troisième filtre de Kalman diversifié pour tolérer les fautes logicielles.

Références

Avizienis et al, 2004, Basic concepts and taxonomy of dependable and secure computing, IEEE Transactions on Dependable Secure Computing, 1(1):11–33.
Allerton et al, 2008, Distributed data fusion algorithms for inertial network systems. Radar, Sonar & Navigation, IET, 2(1):51- 62.
Bloch et al, 2001, Fusion: General concepts and characteristics, International Journal of Intelligent Systems, 16(10):1107–1134.
Bostrom et al, 2007, On the definition of information fusion as a field of research, Technical Report: HS-IKI-TR-07-006, School of humanities and Informatics, University of Skvde.
Choi et al, 2007, The position estimation of mobile robot under dynamic environment. Proceedings of the 33rd Annual Conference of the IEEE Industrial Electronics Society, IECON, pages 134–138.
Congwei Hu et al, 2003, Adaptive kalman filtering for vehicle navigation, Journal of Global Positioning Systems, 2(1):42–47.
Casanova et al, 2008, Robot position tracking using kalman filter, Proceedings of the World Congress on Engineering, London UK, volume II.
Caspi et al, 2000, Threshold and bounded-delay voting in critical control systems, In Formal Techniques in Real-Time and Fault-Tolerant Systems, pages 327–337.Springer.
Dubois et al, 1994, Possibility theory and data fusion in poorly informed environments, Control Engineering Practice, 2(5):811–823.
Dempster, 1967, Upper and lower probabilities induced by a multivalued mapping, The annals of mathematical statistics, 38(2):325–339.
.Escamilla-Ambrosio et al, 2001, A hybridkalman filter-fuzzy logic multisensor data fusion architecture with fault tolerant characteristics, Proceedings of the international conference on artificial intelligence, pages 361–367.
Freeston et al, 2002, Applications of the kalman filter algorithm to robot localisation and world modeling, Electrical Engineering Final Year Project, University of Newcastle, Australia.
Kalman et al, 1960, A new approach to linear filtering and prediction problems, Journal of basic Engineering, 82(1):35–45.
Laprie, 1992, Dependability: Basic Concepts and Terminology, Springer-Verlag New York, Inc., Secaucus, NJ, USA.
Marzullo, 1990, Tolerating failures of continuous-valued sensors, ACM Transactions Computer Systems (TOCS), 8(4):284–304.
Nistér et al, 2004, Visual odometry, In Computer Vision and Pattern Recognition, Proceedings of the IEEE Computer Society Conference, volume 1, pages I–652.
Shafer, 1976, A mathematical theory of evidence, volume 1, Princeton university press Princeton.
Son-Goo Kim et al, 2007, Kalman filtering for relative spacecraft attitude and position estimation. Journal of Guidance, Control, and Dynamics, 30(1):133–143
Varshney et al, 1997, Multisensor data fusion, Electronics & Communication Engineering Journal, 9(6):245–253.
Zug et al, 2009, An approach towards smart fault-tolerant sensors, Proceedings of the International Workshop on Robotic and Sensors Environments. ROSE, pages 35–40.