

# *Model checking* paramétrique statistique du plan de vol de drone civil<sup>\*</sup>

Ran Bao<sup>1,2</sup>, Christian Attiogbe<sup>2</sup>, Paulin Fournier<sup>2</sup>, and Didier Lime<sup>3</sup>

<sup>1</sup> PIXIEL GROUP, Nantes, France <https://www.pixiel-group.com/>

<sup>2</sup> Université de Nantes - LS2N UMR CNRS 6004, Nantes, France

<sup>3</sup> Centrale Nantes - LS2N UMR CNRS 6004, Nantes, France

## Résumé

Les drones sont maintenant très répandus dans la société et sont souvent utilisés dans des situations dangereuses pour le public environnant. Il est alors nécessaire d'étudier leur fiabilité, en particulier dans le contexte de vols au-dessus d'un public. Dans cet article, nous étudions la modélisation et l'analyse de drones dans le contexte de leur plan de vol. Pour cela, nous construisons un modèle probabiliste paramétrique du drone et l'utilisons ainsi que son plan de vol pour modéliser la trajectoire du drone. Ce modèle prend en compte des paramètres comme la défaillance éventuelle du filtre ou du capteur (comme le GPS), ainsi que la force et la direction du vent. Du fait de la nature et de la complexité des modèles successifs obtenus, leur vérification avec les outils PRISM ou PARAM est impossible. Nous développons donc une nouvelle méthode d'approximation, appelée *Parametric Statistical Model Checking*, afin de calculer les probabilités de défaillance du drone. Cette méthode a été implémentée dans un prototype, que nous avons utilisé pour résoudre des difficultés complexes dans une étude de cas réelle.

## 1 Introduction

Les drones sont de plus en plus présents dans la société à travers les loisirs ou les activités industrielles. Ils peuvent être dangereux pour leur environnement, par exemple en cas de défaillance d'un drone qui vole au dessus d'une foule de personnes. Malheureusement jusqu'à aujourd'hui, il n'existe aucune réglementation au niveau international, mais il y a des recommandations ou règlements qui sont utilisés dans certaines régions ou certains pays ; par exemple les vols de drones ne sont pas autorisés au dessus du public, afin d'éviter les accidents en cas de dysfonctionnement du drone. Dans ce contexte, nous travaillons avec la société PIXIEL pour construire un système de drone fiable. PIXIEL est une entreprise experte en construction, sécurité et spectacle de drones civils. L'entreprise étant très attachée à la sécurité des acteurs et spectateurs, elle a pour priorité de s'assurer que ses systèmes de drone sont sécurisés et ne présentent pas de danger pour les humains pendant les spectacles. Concernant la pratique actuelle, les spectacles avec drones ne sont donnés que par beau temps et l'espace en dessous de la zone de vol des drones est interdit aux acteurs et au public. Cependant, il n'y a aucune preuve ou garantie que les drones suivent toujours les plans de vol prévus.

La gestion des spectacles exige une attention particulière pour le calcul de la trajectoire du drone ainsi que pour la précision des mesures concernant sa position effective dans l'espace et ses mouvements. Cependant, une étude rigoureuse est nécessaire pour garantir la fiabilité du système de contrôle des drones, par exemple en réduisant les risques de défaillance par un réglage approprié des paramètres de vol qui ont un impact sur le calcul de la trajectoire du drone. En conséquence, des questions soulevées sont : *comment prouver que la probabilité de*

---

<sup>\*</sup>Ce travail a été Soutenu par PIXIEL et l'Association Nationale Recherche Technologie (ANRT)

*défaillance du drone est faible ? quels paramètres doivent être pris en compte pour assurer la sécurité des personnes pendant les spectacles intégrant des drones ?*

Les pannes physiques sont bien gérées avec les drones *Hexarotors*<sup>1</sup> alors qu'une panne logicielle peut être beaucoup plus problématique et complexe à étudier. Dans ce cas, le comportement du drone peut devenir imprévisible. L'un des problèmes critiques dans ce contexte est l'éventuelle inexactitude de l'estimation de la position dans les systèmes de drones ; ce qui peut provenir soit des mesures imprécises des capteurs, soit d'une interprétation erronée des données provenant de ces capteurs. En dehors des défaillances des composants physiques du drone, il y a aussi un aspect beaucoup plus critique à prendre en compte : les conditions météorologiques. Par conséquent, une approche générale visant à améliorer la sécurité des drones consiste à étudier l'impact des imprécisions des mesures des positions du drone sur sa trajectoire par rapport à un plan de vol fixé, tout en tenant compte des conditions météorologiques.

De nombreux travaux sont consacrés au domaine des drones ; par exemple le fonctionnement d'un drone [20], la reconnaissance du mouvement [10], l'atterrissage automatique sur cible [17], la surveillance et le suivi [11], la détection des pannes et les dysfonctionnements [13] ou le développement d'un modèle de diagnostic élémentaire [9] pour la résolution de problèmes de composants systèmes. Nos travaux se rapprochent plus de cette seconde catégorie. Cependant, à notre connaissance, il n'existe aucun travail sur l'étude paramétrique de l'impact de l'imprécision des composants sur la trajectoire de drone. Dans [5, 24] les auteurs étudient à travers les problèmes d'*estimations garanties (secure estimation)* comment estimer les états réels d'un système de drone lorsque les mesures des capteurs sont corrompues, par exemple par des malveillants. Bien que les techniques qu'ils utilisent sont complètement différentes des nôtres, nous partageons le même objectif qui est d'éviter les états incorrects, afin que le drone ne se retrouve pas dans une mauvaise zone pour la sécurité.

Le but de nos travaux est donc de fournir les moyens d'étudier la fiabilité des drones multirotors dans le contexte où un plan de vol est connu à l'avance. Dans ce but, nous étudions en profondeur le système de drone, nous le formalisons et l'analysons à l'aide de méthodes probabilistes paramétriques. Parmi les composants d'un système de drones, nous nous intéressons particulièrement au *contrôleur de vol* (FCS - *Flight control system*), qui est chargé de calculer des estimations de la position du drone pendant son vol, afin d'adapter sa trajectoire au plan de vol prédéfini. Nous construisons donc un modèle formel du contrôleur de vol à travers plusieurs modèles probabilistes paramétrés (successifs) prenant en compte les éventuelles imprécisions de l'estimation de la position. La force du vent pouvant varier considérablement d'un point du plan de vol à un autre, nous utilisons également dans notre modèle des paramètres pour encoder la force du vent et ainsi permettre de s'adapter à des conditions météorologiques particulières.

Les contributions de cet article sont les suivantes : i) une méthode pour construire un modèle paramétré de systèmes de drone ; les paramètres peuvent ensuite être réglés jusqu'à obtenir des valeurs qui garantissent des seuils de sécurité souhaités ; ii) une technique de *model checking* statistique paramétrique ; cela nous permet d'analyser formellement les modèles paramétrés construits pour les drones. En effet, en raison de la complexité des modèles construits, les outils comme PRISM [15, 16] et PARAM [12] sont limités pour les analyses ; iii) une illustration de l'utilisation de notre méthode sur une étude de cas industrielle non triviale.

L'article est organisé de la manière suivante. Dans la section 2, nous donnons les bases essentielles pour comprendre le fonctionnement et le comportement des drones, puis nous construisons un modèle formel qui prend en compte leurs comportements. La section 3 est une introduction aux chaînes de Markov paramétrées et au *model checking* statistique. Les implémentations de nos modèles et des expérimentations sont présentées dans la section 4 ; enfin la section 5 présente

---

1. ils disposent de six moteurs

les conclusions et les perspectives de nos travaux.

## 2 Méthode de modélisation formelle d'un drone

Dans cette section, nous présentons notre méthode pour construire un modèle formel de drone civil multi-rotors. Rappelons que nous sommes intéressés par l'étude de la sécurité des drones, c'est-à-dire l'étude de la probabilité qu'un drone se trouve dans des situations dangereuses pour son environnement, notamment humain. Ces situations sont de deux types : soit le drone peut s'arrêter de voler et tomber, soit il peut entrer dans une zone "interdite" où il mettrait le public en danger. Comme indiqué précédemment, des drones professionnels peuvent gérer le risque de chute à l'aide de la redondance des composants matériels. De plus, tant qu'un drone reste dans une zone "sûre", même en cas de chute, il ne mettra pas le public en danger. L'objectif de notre modèle est donc d'évaluer la probabilité qu'un drone entre dans une zone "interdite".

Nous commençons par expliquer comment les zones sont calculées par rapport au plan de vol donné. Nous montrons ensuite comment le logiciel de contrôle d'un drone peut être décomposé en composants et nous nous focaliserons sur les plus importants d'entre eux. Finalement, nous détaillons comment les modèles formels (mathématiques) des composants importants sont construits et présentons ensuite le modèle global.

Dans le contexte du logiciel, la certification des systèmes aéroportés et des équipements embarqués (norme DO-178C) a défini cinq niveaux de zones de sécurité, le plus sûr étant la zone 1 et le plus dangereux, la zone 5. Ces zones sont caractérisées par leur distance par rapport au plan de vol prévu, comme illustré dans la figure 1.

### 2.1 Les zones de sécurité

La taille de chaque zone de sécurité n'est pas définitivement fixée; elle peut être définie pour un besoin spécifique ou pour une application donnée. En pratique, les zones de sécurité sont spécifiquement définies pour un environnement de vol et pour un plan de vol donné. Autour d'un drone les zones où peuvent se trouver du public sont catégorisées. Le principe général de cette définition est que personne ne doit être dans les zones 1 à 3, alors qu'il peut y avoir quelques personnes dans la zone 4 et que la plus grande partie des personnes se trouvent dans la zone 5. En conséquence, la probabilité que le drone mette en danger des humains est directement proportionnelle à la probabilité que le drone entre dans les zones 4 ou 5. Dans la suite de l'article, notre objectif sera donc de calculer cette probabilité, et ensuite étudier comment la minimiser sur la base d'un modèle.

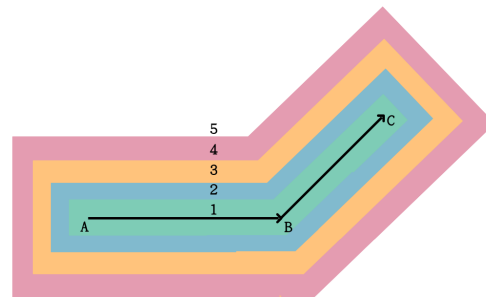


FIGURE 1 – Zones de sécurité

### 2.2 Les composants du drone

Nous décomposons maintenant un drone en composants matériels et composants logiciels et introduisons le plus important parmi eux : le composant logiciel contrôleur de vol (FC). Le FC est responsable de la collecte des données venant des différents capteurs. Ces données permettent de calculer la *position* et l'*attitude*<sup>2</sup> précises du drone et d'ajuster l'*attitude* de manière à suivre le plan de vol indiqué du mieux possible. La position du drone (différente de

2. *attitude* est définie par (*roll*, *pitch*, *yaw*) ; elle est différente de l'altitude

son *attitude*) est caractérisée par les coordonnées  $(x,y,z)$  où  $z$  l'altitude du drone reste fixe dans les trajectoires considérées<sup>3</sup>.

### 2.2.1 La contrôleur de vol

Comme présenté ci-dessus, le *contrôleur de vol* est le composant central de tout système de drone, car il est chargé de collecter les données des capteurs et de les traduire en orientation pour le drone. La figure 2 donne un aperçu du contrôleur de vol d'un drone. Notons que le contrôleur de vol peut être lié à des composants chargés de communiquer avec un poste de commande à distance.

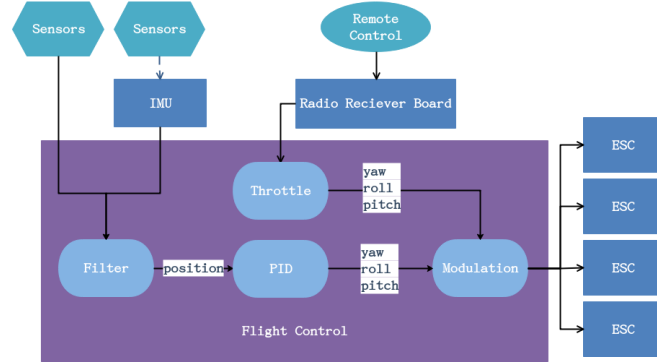


FIGURE 2 – Vue d'ensemble de contrôle de vol

Bien que ces composants soient nécessaires pour permettre à un pilote de prendre le relais en cas de défaillance du mode automatique du drone, nous considérerons dans la suite que les drones étudiés ici sont toujours en mode automatique.

Comme on peut le voir sur la figure 2, le comportement intuitif du contrôleur de vol est le suivant. Le composant *filtre* utilise des mesures des capteurs afin de calculer la position et l'orientation actuelles du drone. Comme les données peuvent être bruitées et imprécises, le filtre utilise des algorithmes complexes afin de nettoyer les bruits dans les mesures et calculer une position réaliste. Notons que dans certains cas, le filtre peut lui-même introduire des imprécisions dans la position calculée, et cela peut être problématique. Une fois que les estimations de la position et de l'orientation actuelles sont calculées, le *Proportional Integral Derivative (PID)* utilise ces informations pour calculer la trajectoire locale que le drone doit suivre afin d'être le plus proche possible du plan de vol prévu. Cette trajectoire locale se transforme alors en nouvelle orientation du drone. Enfin, le composant *modulation* transforme cette orientation en un signal envoyé au *Electronic Speed Controller (ESC)* qui contrôle la vitesse de chaque moteur.

Rappelons que nous sommes intéressés par le calcul de la probabilité pour qu'un drone entre dans une zone interdite pendant qu'il suit son plan de vol. Par construction, tant que les mesures de position et d'orientation sont parfaites, il n'y a aucune raison pour que le drone dévie de sa trajectoire prévue et donc la probabilité qu'il entre dans une zone interdite est nulle. Puisque les données capteurs peuvent être erronées, et aboutir à une déviation du plan de vol prévu, pouvant conduire à l'accès à une zone interdite, il est très important pour nous d'étudier le fonctionnement des filtres et de prendre en compte l'imprécision éventuelle de la position et de l'orientation dans la construction de notre modèle formel.

### 2.2.2 Le filtre

Le rôle du filtre est d'utiliser des mesures de capteurs afin de calculer la position et l'orientation du drone avec la meilleure précision possible. Cependant, une meilleure précision est obtenue au prix de la complexité : pour gagner en précision, les filtres utilisent des algorithmes complexes qui prennent beaucoup de temps ; par conséquent, plus les filtres sont précis plus ils sont lents ; ce qui implique que les positions sont calculées moins souvent, résultant finalement à des imprécisions.

3. drone multi-rotors

Il y a une grande variété de filtres dans l'industrie des drones ; on trouve par exemple, le filtre de Kalman étendu (*Extended Kalman Filter (EKF)*) [22], le *Explicit Complement Filter* [8], le *Gradient Descent* [19], le *Conjugate Gradient*, et un filtre plus précis mais lent : *Unscented Kalman Filter (UKF)* [6], etc. La plupart du temps les chercheurs utilisent le filtre EKF comme référence pour comparer les autres types de filtre, et expliquer les différences de précision et de rapidité. Tous les filtres améliorent leur précision pendant le vol à travers des essais d'adaptation, en particulier en enregistrant les bruits environnant et en les corrigeant. Cependant ces adaptations ne sont valides que pour un vol et sont perdues après le vol.

Puisque la précision de la position et de l'orientation estimée est d'une grande importance pour calculer la probabilité d'entrer dans une zone interdite, et puisque le choix d'un filtre a un impact direct sur ces mesures, nous avons décidé de traiter cette précision comme un paramètre de notre modèle. Nous détaillons cela dans la section 2.4.

### 2.3 Modèle formel du drone dans son environnement

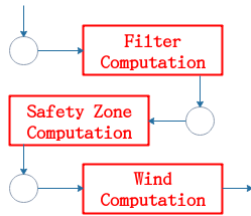


FIGURE 3 – Un logigramme des étapes de formalisation

Nous utilisons un logigramme pour présenter notre approche globale pour formaliser le fonctionnement d'un drone (voir Figure 3). Après une étape où le calcul effectué par le filtre reflète la précision de la position et de l'orientation, nous considérons les calculs des probabilités pour atteindre les zones de sécurité dans l'étape suivante ; en conséquence, l'idée est d'adapter la valeur d'orientation suivante en fonction du plan de vol initial afin d'assurer plus de sécurité. La dernière étape permet d'intégrer les perturbations dues au vent et calculer la position suivante du drone.

Dans la suite, nous montrons comment la position suivante du drone est calculée en fonction de la position actuelle estimée, et comment les erreurs dans l'estimation peuvent conduire le drone à pénétrer dans des zones interdites.

#### 2.3.1 Calcul de la position suivante

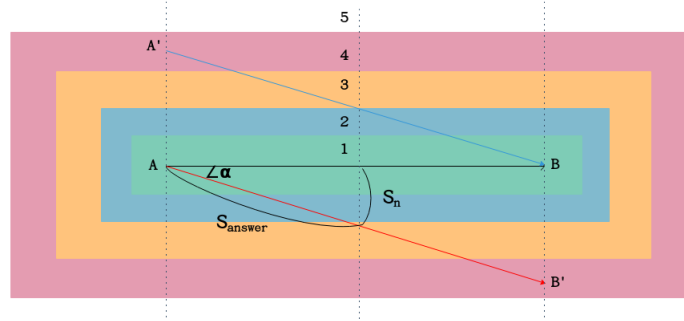


FIGURE 4 – Difficultés sur la localisation du drone et les positions erronées

Nous expliquons maintenant comment la prochaine position du drone est calculée en fonction de la position actuelle. En particulier, nous montrons que l'imprécision dans l'estimation peut conduire le drone à pénétrer dans une zone interdite.

Dans un souci de simplicité, nous supposons que le drone se déplace seulement dans le plan et que des imprécisions apparaissent uniquement sur un des axes ( $x$  ou  $y$ ). La figure 4 illustre la situation. Supposons que le plan de vol envisagé consiste à aller d'un point  $A$  à un point  $B$ .

Considérons aussi que la position actuelle du drone est exactement en  $A$  mais que la position estimée (prenant en compte les imprécisions des capteurs et du filtre) est en  $A'$ . Par conséquent, le PID essayera de corriger la déviation actuelle en changeant l'angle du drone de façon à le ramener vers  $B$ . Cependant, puisque le drone est effectivement en  $A$ , la correction va plutôt le conduire vers la position  $B'$  donc dans une zone interdite. Heureusement, l'estimation de la position se fait plusieurs fois entre  $A$  et  $B$  en fonction de la fréquence  $f$  du filtre. Donc, une nouvelle position sera estimée avant d'atteindre  $B'$ , avec l'espoir d'une meilleure précision, qui permettra au PID de recorriger la trajectoire. Nous devons aussi tenir compte du fait que la vitesse du drone est aussi calculée par rapport au plan de vol ; ce qui permet d'obtenir le temps et la distance restants avant le point de contrôle suivant. Nous montrons maintenant comment nous pouvons calculer la zone de sécurité qui sera atteinte par le drone avant que sa position ne soit de nouveau estimée. Dans la Figure 4 cette zone est représentée par la distance  $S_n$ .

Soit  $S_{answer}$  la distance que parcourt le drone avant l'estimation d'une nouvelle position. Soit  $V$  la vélocité du drone, qui est calculée par le PID afin d'atteindre  $B$  à temps, i.e. précisément en  $T$  unités de temps. Nous avons  $V = A'B/T$ , et  $S_{answer} = V/f = (A'B)/(T * f)$ . Finalement  $AA'/A'B = S_n/S_{answer}$ , et donc

$$S_n = \frac{AA'}{T * f}.$$

Notons que la déviation résultante est directement proportionnelle à  $AA'/f$ , d'où la nécessité de tenir compte du compromis entre précision et rapidité du filtre afin d'optimiser la probabilité de ne jamais rentrer dans une zone dangereuse. La prise en compte du vent suit un calcul similaire que celui qui vient d'être présenté ; cela nous a permis d'intégrer les paramètres du vent dans notre modèle.

## 2.4 Le modèle global final du drone

Le modèle global du système de contrôle de vol du drone est présenté dans la figure 5. Le but de ce modèle est de représenter les calculs qui ont lieu dans le FCS afin d'adapter la trajectoire du drone au plan de vol prévu, en fonction des imprécisions des estimations des mesures de position et d'orientation aussi bien que des perturbations par le vent. Dans ce modèle, la position exacte du drone est encodée avec les trois coordonnées dans l'espace ; ces coordonnées sont ensuite comparées avec le plan de vol pour décider à quelle zone de sécurité elles correspondent. Dès que le drone a atteint une zone interdite (les zones 4 ou 5), le calcul s'arrête. Le modèle utilise plusieurs paramètres probabilistes. Les paramètres FilterProba1, FilterProba2, FilterProba3, FilterProba4 et FilterProba5 représentent la précision de la position et de l'orientation estimées par le filtre et les capteurs. Le choix probabiliste résultant, représenté dans la boîte étiquetée **Filter Computation** donne alors la distance entre la position exacte et celle estimée du drone. Ce choix est suivi par un calcul dans la boîte étiquetée **Safety Zone Computation** qui calcule les coordonnées exactes de la prochaine position du drone, et permet de décider à quelle zone de sécurité appartient cette position. Lorsque le vent n'est pas pris en compte, le résultat de ce calcul est suffisant pour décider si le modèle doit poursuivre son déroulement. Lorsque le vent est pris en compte, une autre étape suit, elle est indiquée par la boîte étiquetée **Wind Computation** ; d'autres paramètres probabilistes y sont utilisés pour décider de la force du vent (nous supposons une direction constante) et une nouvelle position tenant compte de ces perturbations est calculée. Finalement, la zone de sécurité à laquelle appartient cette dernière position est calculée et si elle est une zone sûre le modèle continue avec le calcul d'une autre position.

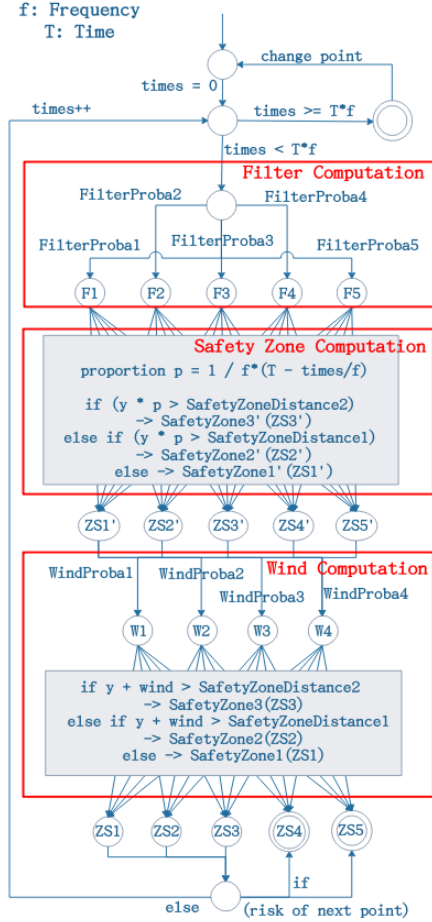


FIGURE 5 – Fonctionnement global du FCS

La fréquence du filtre ainsi que les positions et distances des points de contrôle dans le plan de vol sont donnés comme entrées du modèle. La position du point de contrôle dans le plan de vol permet de calculer la vitesse du drone, alors que la fréquence du filtre permet de fixer le nombre d'estimations de positions qu'il y aura dans un plan de vol donné (i.e. le nombre maximum d'itérations que le modèle doit effectuer).

### 3 Model Checking statistique paramétrique

Comme introduit précédemment, nous avons développé un modèle probabiliste paramétrique pour représenter le comportement de notre drone en fonction d'un plan de vol donné. Nous présentons maintenant la théorie nécessaire pour calculer formellement les probabilités pour qu'un drone ayant un plan de vol, entre dans une zone interdite. Nous commençons par rappeler les bases d'une technique classique de vérification qui est le *model checking* statistique (SMC), puis nous introduisons le formalisme de modélisation que nous avons utilisé : les chaînes de Markov paramétrées (*parametric Markov Chains-pMCs*), et finalement nous montrons comment SMC peut être adapté pour ce formalisme.

#### 3.1 Model Checking statistique

Rappelons qu'une chaîne de Markov (MC) est un modèle purement probabiliste  $\mathcal{M} = (S, s_0, P)$ , où  $S$  est un ensemble d'états,  $s_0 \in S$  est l'état initial, et  $P : S \times S \rightarrow [0, 1]$  est la fonction de transition probabiliste qui, étant donné un couple d'états  $(s_1, s_2)$ , donne la probabilité d'aller de  $s_1$  à  $s_2$ . Dans cette section, nous allons utiliser la terminologie classique des Chaînes de Markov [2].

Le *Model Checking* statistique standard [23] est une technique d'approximation qui permet de calculer une estimation de la probabilité pour qu'un système purement probabiliste satisfasse une propriété donnée<sup>4</sup>. En particulier, la technique de Monte Carlo utilise des échantillons des exécutions de longueur  $l$ ,  $\Gamma(l)$ , d'une MC donnée  $\mathcal{M}$  pour estimer la probabilité que  $\mathcal{M}$  satisfasse une propriété linéaire bornée donnée. Elle peut aussi être utilisée pour approximer la valeur espérée pour une certaine fonction de récompense  $r$  sur les exécutions  $\Gamma(l)$  de  $\mathcal{M}$ . Pour donner une intuition, nous rappelons brièvement comment fonctionne la méthode standard d'analyse de Monte Carlo dans le contexte du *model checking* statistique de chaînes de Markov. Dans ce contexte, nous avons un ensemble de  $n$  échantillons des exécutions de la chaîne de Markov. Ces exécutions sont générées de façon aléatoire en utilisant la distribution de probabilité définie à

4. Des techniques spécifiques des SMC permettent aussi d'estimer la satisfaction de propriétés qualitatives [18].



travers la chaîne de Markov. Chacune de ces exécutions est évaluée, fournissant une valeur en fonction de la fonction de récompense  $r$ .

Selon la loi des grands nombres [21], la valeur moyenne des échantillons fournit un bon estimateur pour la valeur attendue de la fonction de récompense  $r$  sur les exécutions de la chaîne donnée. De plus, le théorème central limite (ou théorème de la limite centrale) fournit un intervalle de confiance qui dépend seulement du nombre d'échantillons (pourvu que ce nombre soit suffisamment grand).

### 3.2 Les chaînes de Markov paramétrées (pMC)

Les chaînes de Markov ne sont pas appropriées à l'analyse de plans de vol de drones car les modèles développés dans ce contexte sont sujets à des données incertaines que nous traitons à l'aide de paramètres tels que la précision des estimations de position et d'orientation et aussi la force du vent. Les modèles résultants ne sont donc pas purement probabilistes puisqu'ils sont paramétrés. En conséquence nous avons besoin de modèles plus expressifs qui prennent en compte les paramètres de probabilité, tels que les chaînes de Markov paramétrées (*Parametric Markov Chains*) (voir par exemple [1]).

Une pMC est un  $n$ -uplet  $\mathcal{M} = (S, s_0, P, \mathbb{X})$  où  $S$  est un ensemble fini d'états,  $s_0 \in S$  est l'état initial,  $\mathbb{X}$  est un ensemble fini de paramètres, et  $P : S \times S \rightarrow \text{Poly}(\mathbb{X})$  est une fonction de transition probabiliste et paramétrique, exprimée comme un polynôme sur  $\mathbb{X}$ . Une valuation de paramètres est une fonction  $v : \mathbb{X} \rightarrow [0, 1]$  qui affecte des valeurs aux paramètres. Une valuation  $v$  est valide par rapport à une pMC  $\mathcal{M}$  lorsque, en y remplaçant les paramètres par leur valeurs, on obtient une chaîne de Markov MC (i.e. la somme des probabilités en sortie de tout état est égale à 1). Lorsque  $v$  est une valuation de paramètres valide par rapport à  $\mathcal{M}$ , on note  $\mathcal{M}^v$  la chaîne de Markov résultant.

Étant donné une pMC  $\mathcal{M}$ , une exécution  $\rho$  de  $\mathcal{M}$  est une séquence d'états  $s_0 s_1 \dots$  telle que pour tout  $i \geq 0$ ,  $P(s_i, s_{i+1}) \neq 0$  (i.e. la probabilité est soit une constante réelle strictement positive, soit une fonction des paramètres). De la même façon que pour les MCs, nous notons  $\Gamma_{\mathcal{M}}(l)$  l'ensemble de toutes les exécutions de longueur  $l$  et  $\Gamma_{\mathcal{M}}$  l'ensemble de toutes les exécutions finies. Notons que pour toute valuation valide  $v$ ,  $\Gamma_{\mathcal{M}^v}(l) \subseteq \Gamma_{\mathcal{M}}(l)$  puisque la valuation de certaines transitions peut être nulle (transition avec la probabilité 0).

### 3.3 SMC paramétrique

Le *model checking* statistique (SMC) standard ne peut pas être utilisé tel quel pour les pMC du fait de la nature des paramètres. En effet nous ne pouvons pas fournir des échantillons selon les transitions probabilistes paramétrées. Mais heureusement, la théorie sous-jacente au SMC peut être étendue pour prendre en compte des paramètres. La méthode que nous proposons dans la suite est similaire à la technique nommée *importance sampling*—échantillonnage préférentiel— [21]. Le but de cette technique est d'échantillonner un système stochastique en utilisant une distribution de probabilité choisie (différente de celle donnée dans le système) et de compenser les résultats en utilisant un coefficient de vraisemblance (*likelihood ratio*) afin d'estimer une mesure en fonction de la distribution initiale. Dans le contexte du SMC, l'échantillonnage préférentiel a surtout été utilisé pour estimer la probabilité d'événements rares [3] ou pour réduire le nombre d'échantillons nécessaires pour obtenir un certain niveau de garantie [14]. Cette technique a aussi été utilisée dans le contexte de chaîne de Markov à temps continu paramétré pour estimer la valeur d'une fonction d'objectif donnée, sur tout l'espace d'états du paramètre tout en utilisant un nombre réduit d'échantillons [4]. Cependant, à notre



connaissance, la technique d'échantillonnage préférentiel n'avait jamais été utilisée pour obtenir des fonctions symboliques des paramètres, comme nous le proposons ici.

L'intuition de la méthode de *model checking* paramétrique statistique que nous proposons est de fixer les probabilités de transition à une fonction arbitraire  $f$  que nous appelons *normalization function*, et d'utiliser ces probabilités de transition pour produire des échantillons de la chaîne de Markov probabiliste  $\mathcal{M}$ .

Cependant, au lieu d'évaluer les exécutions obtenues en utilisant directement la fonction de récompense  $r$ , nous définissons une nouvelle fonction de récompense paramétrique  $r'$  qui tient compte des probabilités de transition paramétriques. Nous montrons alors que pour toute valuation de paramètre  $v$ , l'évaluation de la valeur moyenne de  $r'$  sur un ensemble d'échantillons est un bon estimateur de la valeur de récompense  $r$  attendue sur  $\mathcal{M}^v$ . Le théorème central limite [21] permet aussi de fournir des intervalles de confiance paramétriques, mais nous ne détaillons pas cela ici (voir par exemple [7] pour plus de détails à ce sujet).

Le choix de la fonction de normalisation est crucial. En particulier, les résultats présentés ci-dessous exigent que la structure de graphe de la chaîne de Markov obtenue avec cette fonction soit identique à celle de la chaîne de Markov obtenue en utilisant la valuation de paramètres choisie. Le lecteur trouvera plus de détails dans [7]. Dans la suite, nous considérons uniquement des valuations de paramètres qui affecte des probabilités non nulles aux transitions paramétrées. Puisque nous utilisons une fonction de normalisation uniforme, les graphes des MCs obtenus sont identiques ; c'est la garantie que les résultats présentés ci après sont bien ceux attendus.

Soit  $Pa : \Gamma_{\mathcal{M}} \rightarrow Poly(\mathbb{X})$  une fonction de récompense paramétrique, pour toute valuation valide  $v$  et toute exécution  $\rho \in \Gamma_{\mathcal{M}^v}$  on a  $\mathbb{P}_{\mathcal{M}^v}(\rho) = Pa(\rho)(v)$ .

Pour toute fonction de normalisation valide  $f$  et toute exécution  $\rho \in \Gamma_{\mathcal{M}}$ , considérons une fonction de récompense paramétrique  $r'$  telle que  $r'(\rho) = \frac{Pa(\rho)}{\mathbb{P}_{\mathcal{M}^f}(\rho)}r(\rho)$ .

Soit  $\rho \in \Gamma_{\mathcal{M}^f}(l)$  un échantillon aléatoire de  $\mathcal{M}^f$  et soit  $Y$  la variable aléatoire définie par  $Y = r'(\rho)$ , les calculs suivants démontrent que, pour toute valuation valide  $v$  telle que  $\mathcal{M}^f$  et  $\mathcal{M}^v$  ont la même structure, on a  $\mathbb{E}(Y)(v) = \mathbb{E}_{\mathcal{M}^v}^l(r)$ .

$$\begin{aligned} \mathbb{E}(Y)(v) &= \left( \sum_{\rho \in \Gamma_{\mathcal{M}^f}(l)} \mathbb{P}_{\mathcal{M}^f}(\rho) y(\rho) \right)(v) = \left( \sum_{\rho \in \Gamma_{\mathcal{M}^f}(l)} \mathbb{P}_{\mathcal{M}^f}(\rho) \frac{Pa(\rho)}{\mathbb{P}_{\mathcal{M}^f}(\rho)} r(\rho) \right)(v) \\ &= \sum_{\rho \in \Gamma_{\mathcal{M}^f}(l)} Pa(\rho)(v) r(\rho) &= \sum_{\rho \in \Gamma_{\mathcal{M}^f}(l)} \mathbb{P}_{\mathcal{M}^f}(\rho) r(\rho) \\ &= \sum_{\rho \in \Gamma_{\mathcal{M}^v}(l)} \mathbb{P}_{\mathcal{M}^f}(\rho) r(\rho) &= \mathbb{E}_{\mathcal{M}^v}^l(r) \end{aligned} \quad (1)$$

Notre adaptation de la technique de Monte Carlo pour les pMC consiste ainsi à estimer la valeur attendue de  $Y$  pour obtenir un bon estimateur correspondant aux attentes de la fonction  $r$ . ■

Soit  $\rho_1, \dots, \rho_n$  un ensemble de  $n$  exécutions de longueur  $l$  de  $\mathcal{M}^f$ . Soit  $Y_i$  la variable aléatoire à valeurs dans  $Poly(\mathbb{X})$  telle que  $Y_i = r'(\rho_i)$ . Notons que les  $Y_i$  sont des copies indépendantes de la variable aléatoire  $Y$ , elles sont par conséquent distribuées de façon identique.

Soit  $\gamma$  la fonction paramétrique donnant leur valeur moyenne. A travers ces résultats, pour toute valuation valide  $v$  telle que  $\mathcal{M}^v$  et  $\mathcal{M}^f$  ont la même structure,  $\mathbb{E}_{\mathcal{M}^v}^l(r) = \mathbb{E}(Y)(v) = \mathbb{E}(\sum_{i=1}^n Y_i/n)(v) = \gamma(v)$ . Notre approximation paramétrique de la valeur attendue est par conséquent :

$$\hat{\gamma} = \sum_{i=1}^n Y_i/n.$$

Dans la suite nous utiliserons le *model checking* statistique paramétrique pour vérifier le modèle formel construit pour le drone.

## 4 Implémentations, expérimentations et résultats

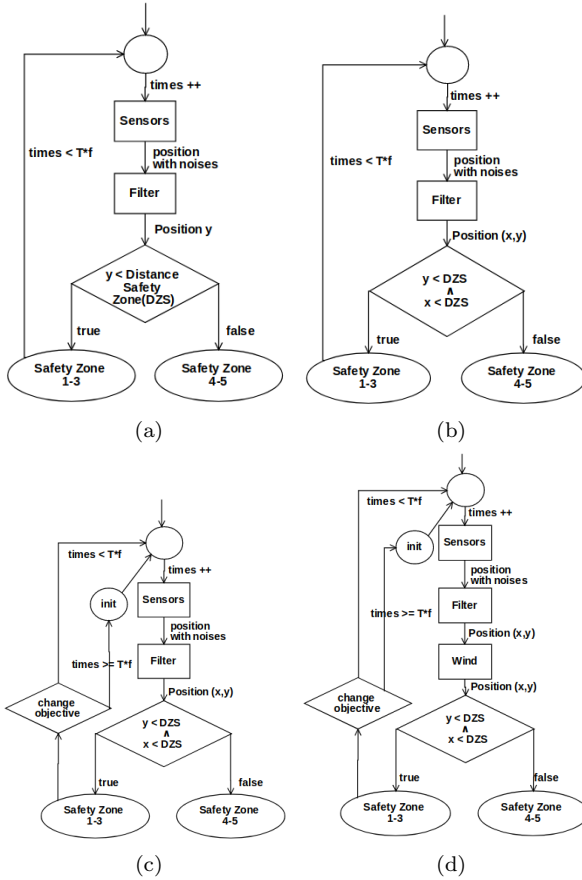


FIGURE 6 – Développement incrémental du modèle SMC

Le filtre nettoie les bruits qui perturbent les données provenant des capteurs. Dans cette première version, nous n'avons considéré que des déviations le long de l'axe des ordonnées ( $y$ ). À chaque étape de calcul, la position brute (imprécise) donnée par le filtre est estimée en utilisant la précision du filtre et des capteurs (comme une unique variable à valeur réelle), et est comparée à la position prévue par le plan de vol. La zone de sécurité est inférée à partir de la distance entre la position calculée par estimation et la position prévue dans le plan de vol. Si le drone entre dans la zone 4 ou dans la zone 5, le calcul s'arrête. Dans ce premier modèle, la précision du filtre est probabiliste mais non paramétrée, i.e. les valeurs de probabilité sont directement encodées dans le modèle. Ces valeurs résultent d'une série d'expérimentations préliminaires effectuées à l'aide d'un contrôleur de vol (disposant d'un filtre EKF) fixé sur le tapis roulant d'une chaîne de production ayant un parcours (avec des points) prédéfini et en boucle, ces valeurs ont été adaptées ensuite pour refléter des conditions de vol réalistes.

Puisque notre but est d'étudier le même problème avec différentes probabilités de précision, nous avons changé les valeurs des probabilités par des paramètres et avons ressoumis ce nouveau modèle à l'étude avec PRISM. Mais, du fait des variables à valeurs réelles utilisées dans le modèle et des nombreux calculs intermédiaires, PRISM n'était pas capable d'analyser ce modèle et

Nous avons introduit notre modèle formel complet dans la section 2 sous la forme d'un automate; nous expliquons maintenant comment nous avons, de façon incrémentale, implémenté et amélioré le modèle en nous servant de différents formalismes et outils de *model checking*. À chaque étape de l'implémentation du modèle, nous montrons les insuffisances qui ont conduit à l'étape suivante. Les différentes étapes de l'implémentation sont illustrées dans dans la figure 6.

En premier, une version partielle sans paramètres du modèle formel de la section 2.3 a été implémentée en utilisant l'outil PRISM [16]. Cette première version illustrée dans la figure 6a, correspond à un plan de vol très simple allant directement d'un point  $A$  à un point  $B$  en  $T$  unités de temps. Dans ce contexte, on estime à  $T * f$  le nombre de positions de contrôle intermédiaires, avec  $f$  la fréquence du filtre. Les dimensions considérées pour les 5 zones de sécurité sont respectivement 20m, 40m, 60m, 80m et 100m.

Comme indiqué dans la section 2, le

explose après 2 heures de calculs infructueux. Au vu de ces limitations avec l’outil PRISM, nous avons entrepris l’implémentation de notre modèle avec PARAM [12], un *model checker* pour les chaînes de Markov à temps discret paramétrique. PARAM est efficace et permet de calculer la probabilité de satisfaire les propriétés données en termes de polynômes ou de fonctions rationnelles des paramètres. Mais, de la même manière que PRISM, PARAM aussi échoue dans l’analyse de la version actuelle de notre modèle.

A ce stade, puisque PRISM et PARAM ont échoué pour l’analyse de notre plus simple modèle du fait de sa complexité relative, nous avons décidé d’utiliser une autre approche basée sur la *model checking* statistique paramétré (PSMC). Pour cela, nous avons développé un outil prototype<sup>5</sup>. Dans ce contexte, notre modèle a été implanté comme un programme python utilisant des variables à valeurs réelles aussi bien pour les positions du drone que pour les probabilités. Il s’en suit que le PSMC est particulièrement efficace, et a pu analyser notre modèle (en effectuant plus de 20k simulations) en moins d’une minute. Nous avons alors décidé de poursuivre nos expérimentations avec cet outil, et avec des versions raffinées de notre modèle.

Dans la seconde version du modèle, illustrée par la figure 6b, nous avons considéré aussi des déviations sur l’axe des abscisses ( $x$ ). Cela ne constitue pas un problème dans le cadre d’un plan de vol direct, mais cela peut devenir important dès lors que le plan de vol contient des courbures (comme illustré dans la figure 1). En effet dans ce contexte, les déviations le long de l’axe des  $x$  (par exemple si le drone prend du retard) pourraient aboutir au fait que le PID décide de couper la trajectoire et aller vers le point  $C$  avant même d’atteindre  $B$ , provoquant ainsi une trajectoire passant par les zones interdites. Là encore, notre outil a pu analyser ce modèle en un temps très court.

En ce qui concerne la troisième version du modèle comme cela est illustré dans la figure 6c, nous avons ajouté dans le plan de vol, comme illustré par la figure 1, un troisième point cible qui n’est pas aligné avec les deux autres. Dans cette troisième version, l’imprécision des estimations de position le long des  $x$ , peut aussi faire prendre du retard au drone et lui faire décider de couper le plan de vol comme expliqué ci dessus.

Finalement la dernière version du modèle illustrée par la figure 6d, prend en compte les perturbations dues au vent. Nous supposons ici que la direction du vent est constante mais que sa force est un paramètre. Cela nous permettra d’étudier le bon compromis entre la capacité du filtre et la fréquence en fonction des conditions météorologiques. Cette dernière version est la plus complexe que nous avons étudiée, et elle a pris plus de temps d’analyse que pour les versions précédentes. Avec notre outil prototype, l’analyse de 10K simulations a duré 190s alors que pour ce même nombre de simulations, elle n’avait duré que 28s dans le cas où le paramètre du vent n’était pas traité<sup>6</sup>.

Les résultats de notre prototype sont sous forme de polynômes dont les termes sont les paramètres du modèle d’entrée. Etant donné le nombre de paramètres, la taille du modèle et la longueur des simulations considérées, ces polynômes sont relativement énormes et par conséquent difficiles à présenter ici. L’exemple ci-après est une partie du polynôme en sortie représentant la probabilité pour qu’un drone entre dans les zones 4 ou 5 en utilisant notre dernier modèle :

$$\begin{aligned} &0.43 * ProbaFilter_3 * ProbaWind_1 + 0.16 * ProbaFilter_3 * ProbaWind_2 \\ &+ 0.17 * ProbaFilter_3 * ProbaWind_3 + 0.28 * ProbaFilter_3 * ProbaWind_4 \\ &+ 0.85 * ProbaFilter_4 * ProbaWind_1 + \dots \end{aligned}$$

5. disponible ici : <https://github.com/paulinfournier/MCpMC>, développé par P.Fournier, un des co-auteurs

6. Pour des raisons de confidentialité liées à l’entreprise partenaire, nous ne diffusons pas les modèles complets expérimentés dans l’outil, mais les versions précédentes avec PRISM et PARAM se trouvent sur <https://github.com/br4444/modelPrism/tree/master>

TABLE 1 – Résultats des expérimentations

	Model	10k		20k		50k	
		V1	V2	V1	V2	V1	V2
<b>Running time</b>	<b>6c</b>	28s		51-54s		142-143s	
Scenario 1	6c	4.99%	5.09%	4.74%	5.10%	4.91%	4.98%
Conf. interv.		±0.85%	±0.82%	±0.55%	±0.56%	±0.36%	±0.37%
Scenario 2	6c	10.38%	10.04%	9.82%	10.05%	9.95%	9.81%
Conf. interv.		±1.15%	±1.12%	±0.79%	±0.80%	±0.51%	±0.51%
<b>Running time</b>	<b>6d(np)</b>	28s		53-54s		149-155s	
Scenario 1	6d(np)	5.44%	5.31%	5.61%	5.21%	5.59%	5.47%
Conf. interv.		±0.98%	±0.86%	±0.69%	±0.64%	±0.42%	±0.43%
Scenario 2	6d(np)	10.8%	10.9%	10.8%	10.8%	10.9%	10.7%
Conf. interv.		±1.35%	±1.32%	±0.91%	±0.92%	±0.57%	±0.57%
<b>Running time</b>	<b>6d(p)</b>	185-190s		311-314s		612-621s	
Scenario 1	6d(p)	4.95%	5.97%	5.28%	6.62%	4.16%	5.61%
Conf. interv.		±5.22%	±5.71%	±4.71%	±6.25%	±1.86%	±4.38%
Scenario 2	6d(p)	9.55%	9.87%	10.3%	11.3%	9.57%	10.7%
Conf. interv.		±8.40%	±7.86%	±7.04%	±7.89%	±5.29%	±3.99%

Au lieu de présenter les polynômes résultats, nous présentons uniquement l'évaluation de ces polynômes avec des valeurs réalistes des paramètres. Nous avons défini deux scénarios (Scenario 1, Scenario 2) avec un ensemble de paramètres pour chacun. Pour les deux scénarios ProbaF0 (resp ProbaF1, ProbaF2, ProbaF3, ProbaF4) modélise la probabilité que la position estimée soit de 0 à 2m (resp. 2 – 4m, 4 – 6m, 6 – 8m, 8 – 10m) de la position effective. Dans le premier scénario, nous avons fixé ces valeurs à 0.15/0.3/0.4/0.1/0.05 et à 0.1/0.25/0.35/0.2/0.1 pour le second. Selon les expérimentations menées chez PIXIEL, le premier scénario est plus réaliste que le second. De façon similaire, les paramètres du vent correspondent à la probabilité d'avoir une force de vent de 0 – 20km/h, 20 – 30km/h, 30 – 50km/h et 50 – 70km/h respectivement ; et ils ont été définis à 0.55/0.43/0.01/0.01 (ce qui correspond à un temps typique de la région de Nantes) pour les expérimentations. Dans les deux scénario, la zone 4 (resp. zone 5) est située à 8m (resp. 50m) du plan de vol.

Nous avons synthétisé dans le tableau 1, les résultats de l'exécution des simulations pour les deux scénario ; la simulation avec PSMC est effectuée avec 10k, 20k et 50k échantillons. Chaque fois, un polynôme est calculé puis évalué en utilisant les paramètres. Pour illustrer la stabilité de nos résultats malgré leur nature statistique, chaque scénario est joué deux fois (indiqué par les étiquettes V1, V2 dans le tableau). Les valeurs reportées dans le tableau 1, représentent la probabilité pour que le drone atteigne les zones 4 ou 5 pendant son vol. Les valeurs de l'intervalle de confiance (Conf. interv.) sont calculées avec une méthode décrite dans [7].

Nous avons effectué des expérimentations avec les modèles formels présentés dans la figure 6c (sans le paramètre du vent) et dans la figure 6d (avec le vent) sur un plan de vol semblable à celui de la figure 1, pour une durée totale de vol de 5s et une fréquence de filtre de 1Hz. Nous avons considéré deux versions du modèle donné dans la figure 6d : la version étiquetée (np) où la force du vent est intégrée directement dans le modèle comme une probabilité constante, ce qui aboutit à un polynôme où les variables représentent uniquement la précision des positions estimées ; la version (p) où la force du vent est intégrée dans le modèle sous forme de paramètres

variables, permettant ainsi d'évaluer et d'optimiser le polynôme obtenu avec n'importe quelle force de vent. Notons que les résultats de la première version sont plus précis car le polynôme est produit de façon plus efficace et il a moins de variables. Selon que nous soyons intéressés par des informations spécifiques ou génériques concernant les conditions météorologiques, nous pouvons utiliser la première ou la seconde version. Notons aussi que, les probabilités d'entrer dans les zones interdites sont relativement élevées ; ce n'est pas surprenant puisque la zone 4 est à 8m de la trajectoire envisagée et la précision de l'estimation de la position peut aller jusqu'à 10m. Ces valeurs ont été choisies aussi grandes de façon délibérée pour notre étude, mais elles peuvent être choisies plus proche de la réalité lorsqu'on veut analyser le modèle réel.

## 5 Conclusion et travaux futurs

Nous avons présenté un modèle formel pour l'étude de la sûreté de drone en mode de vol automatique et avec un plan de vol prédéfini. Ce modèle est construit à l'aide de chaînes de Markov paramétrées ; il prend en compte la précision des estimations de la position et de l'orientation en utilisant les capteurs, les filtres et aussi les perturbations liées au vent. Nous avons aussi proposé une nouvelle technique de vérification pour les modèles probabilistes paramétrés : ils s'agit du *model checking* statistique paramétré (*parametric Statistical Model Checking*). Cette nouvelle technique a été implantée dans un outil prototype. Alors que les outils les plus au point dans la communauté, comme PRISM et PARAM ont le *time out* pendant la vérification de la version simpliste de notre modèle, nous avons réussi avec notre prototype à vérifier en moins de 12 minutes la plus complexe des versions du modèle.

En perspective, nous projetons de continuer à améliorer notre modèle pour intégrer la fréquence du filtre comme paramètre du modèle. La prise en compte de tous ces paramètres nous permettra d'avoir la probabilité paramétrique pour entrer dans des zones dangereuses en fonction de la fréquence du filtre et de la précision des probabilités.

L'étude et l'optimisation de ces probabilités paramétriques permettront à l'entreprise PIXIEL d'affiner le compromis entre la fréquence du filtre et la précision afin de choisir judicieusement en fonction des vols à effectuer, les bons composants de drone à utiliser.

**Remerciements** : Nous remercions vivement M. Benoît Delahaye pour ses suggestions et ses précieux conseils techniques tout au long de ce travail.

## Références

- [1] Alur, R., Henzinger, T.A., Vardi, M.Y. : Parametric real-time reasoning. In : Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing. pp. 592–601 (1993)
- [2] Baier, C., Katoen, J. : Principles of model checking. MIT Press (2008)
- [3] Barbot, B., Haddad, S., Picaronny, C. : Coupling and importance sampling for statistical model checking. In : International Conference on Tools and Algorithms for the Construction and Analysis of Systems. pp. 331–346. Springer (2012)
- [4] Bortolussi, L., Milios, D., Sanguinetti, G. : Smoothed model checking for uncertain continuous-time markov chains. *Information and Computation* **247**, 235–253 (2016)
- [5] Chang, Y.H., Hu, Q., Tomlin, C.J. : Secure estimation based Kalman Filter for cyber-physical systems against sensor attacks. *Automatica* **95**, 399–412 (2018)
- [6] de Marina, H.G., Pereda, F.J., Giron-Sierra, J.M., Espinosa, F. : UAV Attitude Estimation Using Unscented Kalman Filter and TRIAD. *IEEE Transactions on Industrial Electronics* **59**(11), 4465–4474 (Nov 2012)
- [7] Delahaye, B., Fournier, P., Lime, D. : Statistical model checking for parameterized models (Feb 2019), working paper

- [8] Euston, M., Coote, P., Mahony, R., Kim, J., Hamel, T. : A complementary filter for attitude estimation of a fixed-wing UAV. In : 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 340–345 (Sep 2008)
- [9] Freddi, A., Longhi, S., Monteriù, A. : A model-based fault diagnosis system for unmanned aerial vehicles. IFAC Proceedings Volumes **42**(8), 71 – 76 (2009), 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes
- [10] Gasior, P., Bondyra, A., Gardecki, S. : Development of Vertical Movement Controller for Multirotor UAVs. In : Szwedczyk, R., Zielinski, C., Kaliczynska, M. (eds.) Automation 2017 - Innovations in Automation, Robotics and Measurement Techniques, Proceedings of AUTOMATION 2017, March 15-17, 2017, Warsaw, Poland. Advances in Intelligent Systems and Computing, vol. 550, pp. 339–348. Springer (2017)
- [11] Gonzalez, L.F., Montes, G.A., Puig, E., Johnson, S., Mengersen, K.L., Gaston, K.J. : Unmanned aerial vehicles (uavs) and artificial intelligence revolutionizing wildlife monitoring and conservation. *Sensors* **16**(1), 97 (2016)
- [12] Hahn, E.M., Hermanns, H., Wachter, B., Zhang, L. : PARAM : A Model Checker for Parametric Markov Models. In : Touili, T., Cook, B., Jackson, P. (eds.) Computer Aided Verification. pp. 660–664. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
- [13] Heredia, G., Caballero, F., Maza, I., Merino, L., Viguria, A., Ollero, A. : Multi-Unmanned Aerial Vehicle (UAV) Cooperative Fault Detection Employing Differential Global Positioning (DGPS), Inertial and Vision Sensors. *Sensors* **9**(9), 7566–7579 (2009)
- [14] Jegourel, C., Legay, A., Sedwards, S. : Cross-entropy optimisation of importance sampling parameters for statistical model checking. In : International Conference on Computer Aided Verification. pp. 327–342. Springer (2012)
- [15] Kwiatkowska, M., Norman, G., Parker, D. : PRISM : Probabilistic Model Checking for Performance and Reliability Analysis. *ACM SIGMETRICS Performance Evaluation Review* **36**(4), 40–45 (2009)
- [16] Kwiatkowska, M., Norman, G., Parker, D. : PRISM 4.0 : Verification of probabilistic real-time systems. In : Gopalakrishnan, G., Qadeer, S. (eds.) Proc. 23rd International Conference on Computer Aided Verification (CAV’11). LNCS, vol. 6806, pp. 585–591. Springer (2011)
- [17] Kyriassis, S., Antonopoulos, A., Chanielakis, T., Stefanakis, E., Linardos, C., Tripolitsiotis, A., Partsinevelos, P. : Towards Autonomous Modular UAV Missions : The Detection, Geo-Location and Landing Paradigm. *Sensors* **16**(11), 1844 (2016)
- [18] Legay, A., Delahaye, B., Bensalem, S. : Statistical model checking : An overview. In : Proc. Runtime Verification - First International Conference, RV 2010, St. Julians, Malta, November 1-4, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6418, pp. 122–135. Springer (2010)
- [19] Madgwick, S.O.H. : An efficient orientation filter for inertial and inertial / magnetic sensor arrays. In : [www.semanticscholar.org/](http://www.semanticscholar.org/) (Technical report, 2010)
- [20] Máthé, K., Busoniu, L. : Vision and Control for UAVs : A Survey of General Methods and of Inexpensive Platforms for Infrastructure Inspection. *Sensors* **15**(7), 14887–14916 (2015)
- [21] Rubinstein, R.Y., Kroese, D.P. : Simulation and the Monte Carlo method, vol. 10. John Wiley & Sons (2016)
- [22] Sabatelli, S., Galgani, M., Fanucci, L., Rocchi, A. : A Double-Stage Kalman Filter for Orientation Tracking With an Integrated Processor in 9-D IMU. *IEEE Transactions on Instrumentation and Measurement* **62**(3), 590–598 (March 2013)
- [23] Sen, K., Viswanathan, M., Agha, G. : On statistical model checking of stochastic systems. In : International Conference on Computer Aided Verification. pp. 266–280. Springer (2005)
- [24] Zhou, Z., Ding, J., Huang, H., Takei, R., Tomlin, C. : Efficient path planning algorithms in reach-avoid problems. *Automatica* **89**, 28–36 (2018)