



Rapport de stage M2

Exploration de structures météorologiques cohérentes par une flotte de drones

Christophe Reymann

sous la direction de Simon Lacroix¹

Résumé

Ce rapport présente les travaux de recherche effectués entre le 2 février et le 31 juillet 2015, dans le cadre d'un stage effectué dans l'équipe RIS (Robotique et Interactions) du laboratoire LAAS-CNRS (Laboratoire d'Analyse et d'Architecture des Systèmes), sous la direction de Simon Lacroix. Dans le contexte du projet SkyScanner pour l'utilisation de flottes de micro-drones pour l'exploration de structures météorologiques, mon travail s'est porté sur le pilotage d'une flotte de drones. Ce rapport présente succinctement le projet et les partenaires impliqués, et formule la problématique posée. Après un état de l'art, les solutions envisagées de planification des déplacements de la flotte, ainsi que de modélisation de l'environnement aérologique sont décrites. Le rapport fait ensuite état de l'implémentation actuelle de celles-ci et présente des résultats préliminaires avant d'ouvrir les perspectives pour la suite du projet.

7 septembre 2015

1. directeur de recherche, LAAS-CNRS

Table des matières

1	Contexte	3
1.1	Le projet Skyscanner	3
1.2	Enjeux scientifiques	4
1.3	Objet et contributions du stage	5
2	Planification des déplacements de la flotte de drones	8
2.1	Formulation du problème	8
2.2	Etat de l'art orienté problème	9
2.3	L'approche Skyscanner	16
2.4	Planification de trajectoires	17
2.4.1	Modèle d'avion	17
2.4.2	Planification par optimisation stochastique	18
3	Modélisation de l'environnement	19
3.1	État de l'art	20
3.2	Régression par processus gaussiens	25
3.3	Intégrer une connaissance à priori : moyennes et noyaux	27
3.4	Liens avec la planification	28
3.4.1	Fonction d'utilité	28
3.4.2	Intégration étroite entre modèle et plan	32
4	Résultats expérimentaux	35
4.1	Validation sur simulations réalistes	35
4.1.1	Validation statistique du modèle de régression par processus Gaussiens	36
4.1.2	Qualité des observations et variabilité des hyper-paramètres	38
4.2	Intégration du modèle atmosphérique dans la planification de trajectoires	39
4.2.1	Fonction d'utilité	40
4.2.2	Résultats préliminaires	41
4.3	Simulation système réaliste : intégration avec Paparazzi	50
5	Conclusions et perspectives	51
5.1	Discussion des résultats préliminaires	51
5.1.1	Modèle d'environnement	51
5.1.2	Planification	52
5.2	Perspectives	53
	Bibliographie	54

A	Modèle d'avion	56
B	RSS workshop : poster abstract	59

Chapitre 1

Contexte

Profitant de la miniaturisation des technologies et des capacités croissantes des batteries, les drones (appelés aussi UAVs, de l'anglais *Unmanned Aerial Vehicles*), en particulier la classe des micro-drones (poids inférieur à 2kg), ont vu leur prix chuter et leur autonomie augmenter, élargissant ainsi leur domaine d'application. Ces plates-formes posent de nombreux défis techniques, tant au niveau de leur conception que de leur contrôle. En particulier, l'un des défis restant est de doter ces micro-drones de capacités de perception et de raisonnement suffisantes pour leur permettre d'effectuer leur mission sans intervention humaine. Les travaux présentés dans ce rapport ont été réalisés dans le contexte du projet SkyScanner, centré autour d'une application météorologique de micro-drones à voilure fixe.

1.1 Le projet Skyscanner

Si leur défilement hypnotique pousse l'âme à la rêverie, c'est avec une intensité tout autre que les météorologues scrutent les nuages. Car la compréhension des phénomènes météorologiques qui ont lieu en leur sein reste imparfaite, et est cruciale pour la mise en œuvre de modèles météorologiques et climatiques performants.

L'étude des nuages repose en partie sur des mesures effectuées à distance au moyen de radars ou LiDARs (capteurs actifs mesurant la réflexion par l'atmosphère d'une source laser), mais celles-ci demeurent insuffisantes. Certaines grandeurs restent inobservables par ce biais (par exemple les courants ascendants), et il est impossible de rendre compte de la dynamique fine des phénomènes microphysiques en jeu dans la formation des nuages. Il devient alors nécessaire d'effectuer des mesures in situ. Embarquer des instruments à bord d'avions a permis d'effectuer quelques progrès, mais ce genre de déploiement présente des inconvénients prohibitifs. Tout d'abord, la vitesse des avions ne leur permet pas de se concentrer sur un seul nuage. Il est ainsi presque impossible de prendre deux mesures du même nuage, et il faut donc effectuer un passage sur un champ de nuages et calculer des statistiques sur l'ensemble du champ. Ensuite, le coût non négligeable d'une telle campagne de mesures fait que leur usage reste relativement limité.

L'arrivée des UAVs à voilure fixe a changé cette dynamique : de faible coût,

ils peuvent être utilisés en plus grand nombre, et plus souvent. De petite taille, ils peuvent être déployés rapidement pratiquement n'importe où. Enfin, ils sont beaucoup plus lents et plus dynamiques, et peuvent donc suivre un phénomène météorologique particulier. Les météorologues ont en effet été parmi les premiers à adopter ces nouvelles technologies, et l'utilisation d'UAVs a déjà commencé à porter ses fruits [RRR⁺07][ERS⁺11].

Cependant, face à l'immensité d'un nuage qui peut faire plusieurs kilomètres cubes et à la rapidité de leur évolution, les ressources d'un seul UAVs restent insuffisantes. Une évolution naturelle est donc de considérer des flottes de drones, qui de manière coordonnée pourront réellement commencer à effectuer les premières véritables cartographies de nuages.

C'est le cœur du projet SkyScanner, qui se propose d'étudier et d'expérimenter la faisabilité du déploiement d'une flotte de drones à voilure fixe pour l'échantillonnage adaptatif de nuages de type cumulus pendant une période d'environ une heure. Profitant du savoir faire local en matière d'aviation, ce projet regroupe cinq partenaires toulousains : Le CNRM-GAME (Centre National de Recherches Météorologiques - Groupe d'étude de L'Atmosphère Météorologique), l'ENAC (École Nationale de l'Aviation Civile), l'ISAE (Institut Supérieur de l'Aéronautique et de l'Espace), l'Onera (Office National d'Etudes et de Recherches Aérospatiales) et enfin le LAAS-CNRS (Laboratoire d'Analyse et d'Architecture des Systèmes).

Ce projet de recherche fondamentale est supporté par la fondation STAE (Sciences et Technologies pour l'Aéronautique et l'Espace). Il a commencé officiellement en juin 2014 pour s'achever fin 2016. Au-delà des enjeux scientifiques, l'objectif de ce projet est de favoriser la cohésion d'une communauté locale de chercheurs travaillant transversalement sur la conception et le pilotage de drones, de l'optimisation aérodynamique aux algorithmes de pilotage multi-UAVs.

1.2 Enjeux scientifiques

Cinq enjeux scientifiques principaux ont été identifiés pour ce projet par les différentes équipes :

1. Une meilleure compréhension de la micro-physique des nuages
2. Une meilleure compréhension des phénomènes aérodynamiques à l'échelle de mini-drones
3. L'optimisation de la conception de mini-drones endurants
4. Contrôle de vol optimisé, récolte d'énergie
5. Contrôle adaptatif de flotte, dynamiquement guidée par les données récoltées

Le point 1. peut être considéré comme l'objectif de mission des flottes de drones dans ce projet. C'est l'objectif de l'équipe du CRNM en tant qu'utilisateurs des drones. En plus de l'instrumentation, le CNRM fournit également des simulations météorologiques de champs de cumulus, permettant à la fois de caractériser l'environnement dans lequel voleront les drones et de pouvoir effectuer des simulations réalistes. Leur expertise dans la connaissance physique des phénomènes aérologiques est également indispensable à la compréhension des objectifs et des enjeux du projet.

Les points 2. et 3. sont du ressort de l'équipe de l'ISAE, qui se focalise sur la conception d'un vecteur (le drone) à l'aérodynamique performante et adapté au type de mission.

Le point 4. est proposé par l'Onera, qui se propose de développer un modèle et une architecture de contrôle optimisée permettant une dépense énergétique plus faible, voire l'extraction de l'énergie éolienne locale.

Enfin le point 5. est co-développé par le LAAS appuyé par le savoir-faire de l'ENAC. L'enjeu ici est la planification des déplacements des UAVs dans un champ de courants dynamiques, dans le but de maximiser l'utilité de la mission, c'est à dire la qualité de l'information récoltée, tout en minimisant la dépense énergétique pour maximiser le temps de mission.

L'équipe 'drone' de l'ENAC fournit un appui opérationnel : l'autopilote et simulateur de vol Paparazzi¹, développé par l'équipe, sera la base commune utilisée pour l'implémentation des solutions, tant en simulation que pour des vols d'essais. Ceux-ci seront effectués par l'équipe avec des micro-drones instrumentés par leurs soins, pour tester en conditions réelles l'instrumentation, le contrôle et la planification. Ils développent également des modèles de drones, en particulier un modèle énergétique prenant en compte la contribution du moteur, modèles nécessaires tant pour le contrôle que pour la planification. Enfin, l'équipe travaille sur l'amélioration des capteurs et des méthodes d'estimation permettant la mesure du vecteur de vent tridimensionnel, à coût raisonnable et embarquable sur le mini-drone.

1.3 Objet et contributions du stage

Ce stage de six mois a été effectué à partir de février 2015 au LAAS, dans l'équipe RIS (Robotique et Interactions), et dans le contexte du projet SkyScanner, sous la direction de Simon Lacroix, et en collaboration avec Alessandro Renzaglia (post-doctorant).

Étant arrivé dans l'équipe en tout début de projet, qui a effectivement débuté avec l'arrivée d'Alessandro Renzaglia fin 2014, il nous a d'abord fallu effectuer une recherche bibliographique pour explorer l'état de l'art du domaine et pour comprendre les enjeux spécifiques de la navigation dans des champs de courants. Nous avons rapidement pris le parti de considérer deux problématiques majeures : la planification multi-agents et la modélisation de l'environnement sur laquelle va s'appuyer la planification. Les problématiques plus opérationnelles, comme par exemple les problèmes de communication, ont dans un premier temps été mises de côté sans pour autant être oubliées, car elles sont sources de contraintes fortes lors de vols réels.

Au vu de la complexité des scénarios, nous avons identifié le besoin d'une planification hiérarchique : localement une planification de trajectoires pour tirer parti des courants (par exemple ascendances et descendances) et mieux cartographier les zones intéressantes, et à plus grande échelle une planification de tâches pour identifier les zones à cartographier et affecter les drones à des tâches spécifiques.

La qualité de solutions engendrées dépendant en grande partie de la qualité du modèle de l'environnement dans lequel on planifie, il nous a paru crucial de traiter le problème de la modélisation conjointement, et de tirer parti d'un

1. <http://paparazziuav.org/>

couplage fort entre modèle et planification. Ainsi, nous avons mis en évidence le besoin de deux modèles, correspondant aux besoins de planification : un modèle fin et local, capable de reconstituer une carte spatio-temporelle fine des zones explorées, et un modèle de nuage plus conceptuel (plus géométrique) et à une échelle plus grossière, permettant d'identifier les paramètres à mesurer et les zones d'intérêt météorologique.

Forts de ce constat, nous avons décidé d'approcher le problème de bas en haut, en commençant par la planification de trajectoires et la modélisation locale des phénomènes météorologiques. En accord avec les intérêts personnels de chacun, et pour traiter les deux problèmes en parallèle, nous nous sommes réparti les tâches, Alessandro se concentrant sur l'aspect planification, et moi sur l'aspect modèle et sur l'intégration des deux.

En parallèle et dans le cadre de l'intégration de nos travaux dans le projet, nous avons commencé, avec l'appui de l'équipe de l'ISAE, à réfléchir à l'implémentation de nos solutions dans l'autopilote et simulateur de vol Paparazzi (développé à l'ENAC) et j'ai fait les premiers pas pour intégrer une simulation météorologique au simulateur de Paparazzi.

Ces travaux préliminaires ont été présentés lors du workshop "Principles of Multi-Robot Systems" lors de la conférence RSS (Robotics Science and Systems) 2015, le résumé du poster qui a été soumis est inclus en annexe B.

Pour résumer, mes contributions au projet lors de ce stage ont été :

- Participation aux réunions mensuelles de coordination du projet entre les différents partenaires
- Participation à l'analyse du problème et à la définition de l'approche globale
- État de l'art orienté problème
- Choix d'un modèle statistique de l'environnement et état de l'art de la régression par processus gaussien, réflexion sur les enjeux de cette technique
- Développement d'un simulateur simple en Python pour effectuer les premières expérimentations en attendant l'intégration avec Paparazzi
- Élaboration d'un premier modèle de la dynamique du micro-drone pour les besoins de la planification
- Implémentation des algorithmes de construction du modèle de l'environnement à partir de données éparses
- Intégration des simulations fournies par le CNRM aux simulations de vol, et analyse préliminaire de la capacité du modèle sur ces simulations
- Intégration du modèle d'environnement à la planification de trajectoires et réflexions sur la manière de bénéficier d'un couplage fort avec le modèle (au-delà de la simple utilisation des prédictions)
- Définition de fonctions d'utilité pour la récolte d'informations à partir du modèle d'environnement

Le chapitre 2 introduit la planification des déplacements de la flotte de drones en commençant par un état de l'art orienté problème, puis introduit succinctement les solutions employées jusqu'à présent dans le projet. Le chapitre 3 traite du modèle d'environnement. Après un bref état de l'art des techniques employées dans les références traitant de problèmes similaires, la régression par processus gaussien est introduite et son utilisation pour construire un modèle

aérogique local est présentée. Suit une discussion sur les manières d'utiliser ce modèle pour la planification. Le chapitre 4 décrit les premiers résultats expérimentaux issus de ces travaux. Enfin ce rapport se termine dans le chapitre 5 par une discussion des travaux réalisés et une ouverture sur les futures perspectives de travail.

Chapitre 2

Planification des déplacements de la flotte de drones

Depuis quelques années le déplacement autonome d'un micro-drone est un problème résolu. L'autopilote Papparazzi est ainsi capable de contrôler un drone du décollage à l'atterrissage de manière totalement autonome, sur une trajectoire préprogrammée. Cependant, nous cherchons à suivre un phénomène dynamique et partiellement observable : il faut donc intégrer un processus de raisonnement à bord, permettant une replanification en temps réel des mouvements des UAVs.

2.1 Formulation du problème

Commençons par poser le problème. Soient n robots, constituant une flotte possiblement hétérogène de robots (i.e. $n \geq 3$). Le but de la mission est de suivre et cartographier un phénomène météorologique de type cumulus, tout en prenant en compte des contraintes énergétiques dont dépend la durée de vol. Il s'agit donc d'une mission de récolte adaptative de données. On se donne un certain nombre m de variables à estimer le mieux possible au cours du temps. La mission consiste donc à estimer le vecteur de variables

$$\mathbf{U}(t) := (U_1(t), \dots, U_m(t)) \quad (2.1)$$

et les erreurs (incertitudes) associées :

$$\mathbf{I}(t) := (I_1(t), \dots, I_m(t)) \quad (2.2)$$

Ces variables peuvent l'être au sens mathématique : altitude de la base du cumulus, hauteur du nuage, vitesse moyenne de l'ascendance centrale à différentes altitudes, ou peuvent plus généralement représenter un processus plus complexe. On peut par exemple vouloir une modélisation (carte) précise des variables atmosphériques d'une zone restreinte du nuage afin de l'étudier plus en détail. Celles-ci sont les variables d'intérêt météorologique définies par l'opérateur.

On s'intéresse parallèlement à maximiser le temps de mission, c'est à dire à minimiser la perte d'énergie au cours du temps. Soit

$$\mathbf{E}(t) := (E_1(t), \dots, E_n(t)) \quad (2.3)$$

le vecteur représentant l'énergie des UAVs, soit Δ_t l'horizon temporel de planification et $X_i(t) = (x_i, y_i, z_i)(t)$ la position du drone i au temps t . Le problème se pose comme un problème d'optimisation multicritère :

$$\begin{aligned} \min \mathbf{I}(t_{start} + \Delta_t) \\ \max \mathbf{E}(t_{start} + \Delta_t) \end{aligned}$$

sous contraintes :

$$\begin{aligned} z_i(t) > z_{min}, & \quad \forall i \in 1, \dots, n \text{ (altitude minimale)} \\ X_i(t) \neq X_j(t), & \quad \forall (i, j), i \neq j \text{ (pas de collisions)} \\ E_i(t) > E_{min}, & \quad \forall i \in 1, \dots, n \text{ (retour possible à la base)} \end{aligned} \quad (2.4)$$

Le problème ainsi exprimé n'est pas *bien posé*. En effet, il reste à définir la sémantique de "min" et "max", c'est à dire à définir une fonction d'utilité $U : \mathcal{R}^{n+m} \rightarrow R$ qui synthétise l'utilité jointe de toutes les variables à optimiser. Celle-ci doit prendre en compte les préférences de l'opérateur ainsi que la corrélation éventuelle entre les variables.

2.2 Etat de l'art orienté problème

La planification est un des piliers de l'intelligence artificielle et de la robotique, et le but n'est pas ici de dresser un état de l'art de la planification dans son ensemble, mais de se focaliser sur le domaine restreint de la planification pour la navigation autonome de drones en environnement non contraint (c'est à dire sans obstacles), et possiblement en présence de courants. D'intérêt particulier sont les problèmes de gain d'énergie ("soaring") et de récolte d'informations ("information gathering"). On ne cherche pas dans notre cas à inventer de nouveaux outils de planification, mais plutôt à résoudre un problème concret, et c'est donc un état de l'art *orienté problème* qui est présenté ici. Enfin, il est à noter que celui-ci concerne uniquement la planification de trajectoires ("path planning"), l'aspect planification de tâches n'étant pas encore abordé dans la littérature relative au problème considéré.

Le tableau 2.1 compare les articles représentatifs choisis selon neuf critères groupés en trois catégories :

- Le type de mission choisie : collecte d'informations et/ou minimisation de la dépense énergétique, avec éventuellement possibilité de gain énergétique (soaring).
- La modélisation de l'environnement : bi- ou tri-dimensionnel, présence de vents, environnement réel ou simulé

- Le type de planification choisie : algorithmes exacts ou approchés (optimisation combinatoire), avec ou sans prise en compte de l'incertitude, mono- ou multi-agents

Critère / Article	ASGS13	Sou11	KKP14	NLFS13	LS11	CLS15	EA14	RSV13	WS12
Collecte inf.	Non	Non	Non	(Oui)	Oui	Oui	Non	Oui	Oui
Max. Énergie	Oui	Oui	Oui	Non	Oui	Oui	Oui	Non	Non
Soaring	Non	Non	Non	Oui	Oui	Oui	Oui	Non	Non
2D/3D	2D	2D	2D/3D	2.5D	3D	3D	3D	3D	2D
Vent	Oui	Oui	Oui	Non	Oui	Oui	Oui	Non	Non
Sim/Réel	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Réel	Sim
Exact/Approx.	E	E	A	A	A	E/A	—	A	—
Incertain	Oui	Non	Non	Non	Non	Non	—	Non	—
Multi-Agents	Non	Non	Non	Non	Non	Non	Non	Non	Oui

TABLE 2.1 – Tableau comparatif des publications de références choisies selon neuf critères mettant en relation le problème choisi, les modèles appliqués et les moyens mis en œuvre.

Cette catégorisation permet de mettre en évidence les grandes tendances des approches de ce type de problème, et les points encore peu abordés dans la littérature. La première distinction peut se faire selon le type de mission envisagée, la plupart des références envisagent soit la découverte d'un environnement inconnu, soit l'optimisation des trajectoires parcourues. Dans la plupart des cas, la trajectoire est optimisée dans un champ vectoriel connu influençant celle-ci. Le but est alors de minimiser la dépense énergétique.

Dans [ASGS13], les auteurs optimisent la trajectoire en minimisant la durée de trajet entre deux points à vitesse relative à l'air constante. Pour cela, ils utilisent un processus de décision markovien (MDP), l'environnement étant représenté par une grille bidimensionnelle. De manière intéressante, l'incertitude sur la valeur du vecteur vent est prise en compte et modélisée par des transitions non-déterministes : une distribution de probabilité gaussienne est appliquée sur la direction du vecteur vitesse du drone, ce qui permet de calculer la probabilité d'arrivée sur chaque case voisine (fig. 2.1).

M. Soullignac [Sou11] présente une variante de l'algorithme *wavefront expansion* (qui est en essence une recherche en profondeur d'abord), qu'il appelle *sliding wavefront expansion*. Les transitions sont ici envisagées entre deux positions sur les côtés de chaque case de la grille, et non plus directement entre deux cases (fig. 2.2). L'environnement n'est plus discret, mais grâce à une fonction de coût bien choisie, l'auteur propose une méthode permettant d'optimiser itérativement les positions sur les côtés de cases ("sliders" dans l'article) en résolvant un problème d'optimisation convexe. Si l'algorithme est cinq fois plus lent que la version discrète, il permet de résoudre les problèmes de faisabilité des chemins calculés, qui dans le cas classique, proviennent de l'approximation faite par la discrétisation (et ne satisfont donc pas toujours les contraintes cinématiques). Ces approches exhaustives souffrent cependant de problèmes de passage à l'échelle. En effet, la présence de courants ne permet pas de définir fa-

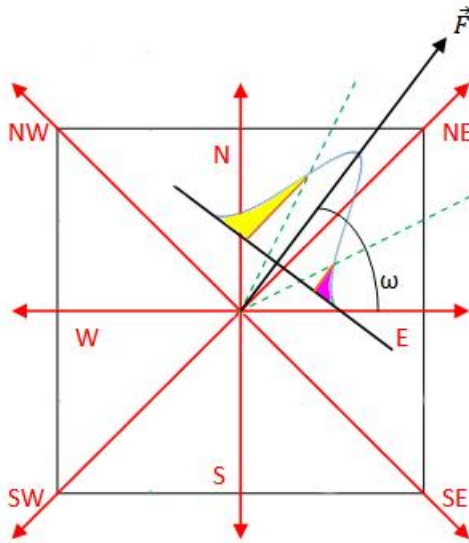


FIGURE 2.1 – Distribution de probabilité de transition dans les états suivants. La flèche noire représente le vecteur vitesse totale, l'aire jaune représente la probabilité d'atteindre l'état Nord, l'aire non coloriée celle d'atteindre l'état Nord-Est et celle en mauve celle d'atteindre l'état Est (extrait de [ASGS13]).

cilement une heuristique minorante et le nombre de chemins croît d'autant plus rapidement qu'ils ne sont pas facilement factorisables de par les contraintes de mouvements spécifiques aux UAVs (en particulier de changement de direction).

L'alternative est donc d'utiliser des techniques approchées, par exemple d'optimisation stochastique qui sont principalement utilisées dès que l'on considère des environnements plus complexes (tridimensionnels). L'approche la plus répandue actuellement est l'utilisation de RRT (Rapidly Exploring Random Trees) comme utilisée dans [KKP14]. Les auteurs introduisent un critère de comparaison de trajectoires basé sur l'effort produit pour se déplacer à contre-courant et l'appliquent sur différentes versions de cet algorithme (fig. 2.3). Ils obtiennent ainsi des chemins de bonne qualité énergétique. Est également abordé le problème de lissage des solutions obtenues : en effet, celles-ci sont basées sur un modèle trop simplifié du monde et du drone. Elles doivent donc, pour devenir réalisables, être lissées à posteriori pour tenir compte des contraintes de déplacement spécifiques du véhicule.

En se plaçant dans un scénario de collecte d'informations couplée à une optimisation énergétique, la complexité du problème augmente encore d'un cran. Le problème posé dans [NLFS13] est celui de l'utilisation de thermiques pour gagner de l'énergie et augmenter la durée de missions de détection de cible au sol. Il est abordé en alternant entre phases d'exploration, et phases d'exploitation des ascendances. Le problème est alors formulé comme un graphe dont les nœuds sont les phases d'exploitation et les arêtes sont les chemins entre deux thermiques. Pour gagner de l'énergie, l'UAV tourne en cercle dans les ther-

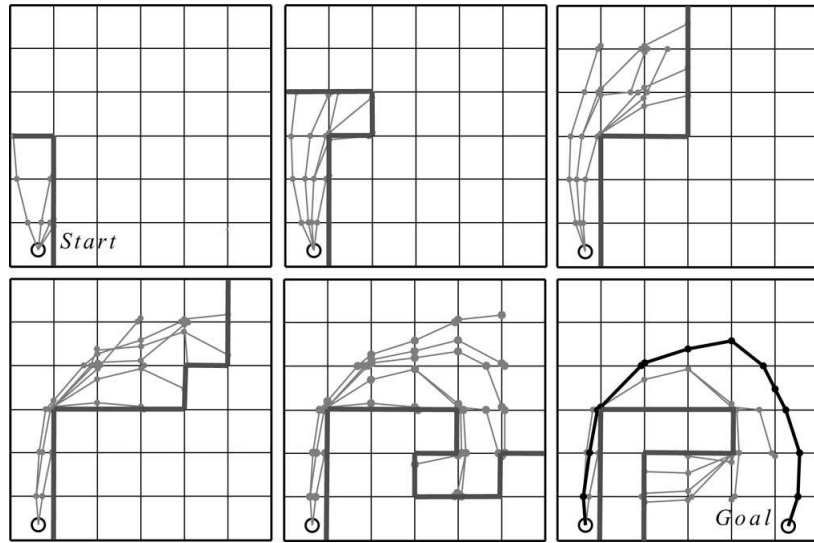


FIGURE 2.2 – Sliding wavefront expansion. Les chemins potentiels sont dessinés en gris clair, le chemin final en gris foncé (extrait de [Sou11]).

miques. Entre deux thermiques, le chemin est optimisé par un algorithme de type RRT qui sélectionne le chemin le plus informatif, c'est à dire celui où le drone pense avoir le plus de probabilité de détecter la cible. Le chemin dans le graphe est optimisé par un algorithme de type profondeur en fixant une profondeur maximale, puis une transition est effectuée sur le meilleur chemin calculé et le planificateur révisé le plan en relançant la recherche en profondeur, ce que les auteurs appellent "Lookahead Tree Search" (LTS). On peut noter deux points fortement limitants : premièrement, l'absence de vent dans les directions du plan, et ensuite, le fait que les thermiques soient considérées connues à priori, et fixes.

Dans un champ de courant inconnu à priori, la navigation se heurte à la difficulté supplémentaire de devoir explorer au fur à mesure celui-ci et donc de gérer une connaissance imparfaite lors de la planification. Les auteurs de [LS11] et [CLS15] abordent ce problème de deux manières différentes.

Dans le premier cas, une fonction d'utilité empirique est élaborée pour pouvoir obtenir un compromis entre exploration et exploitation des champs de courants. La planification se fait de manière hiérarchique. L'étape de plus haut niveau sélectionne une zone d'intérêt, révisée à intervalle de temps de l'ordre de quelques dizaines de secondes, tandis que celle de plus bas niveau planifie des trajectoires en utilisant la fonction d'utilité susmentionnée et une planification de type LTS (fig. 2.4). L'incertitude sur l'estimation du vent est prise en compte implicitement dans la fonction d'utilité, pour explorer les zones où l'incertitude est forte quand le drone a suffisamment d'énergie. On peut reprocher à cette approche le côté empirique et difficilement généralisable de la fonction d'utilité. Il est en effet difficile de justifier la pertinence des différents choix, tant théoriquement que dans la pratique en analysant les trajectoires obtenues.

Dans le deuxième article, la planification repose sur l'apprentissage par ren-

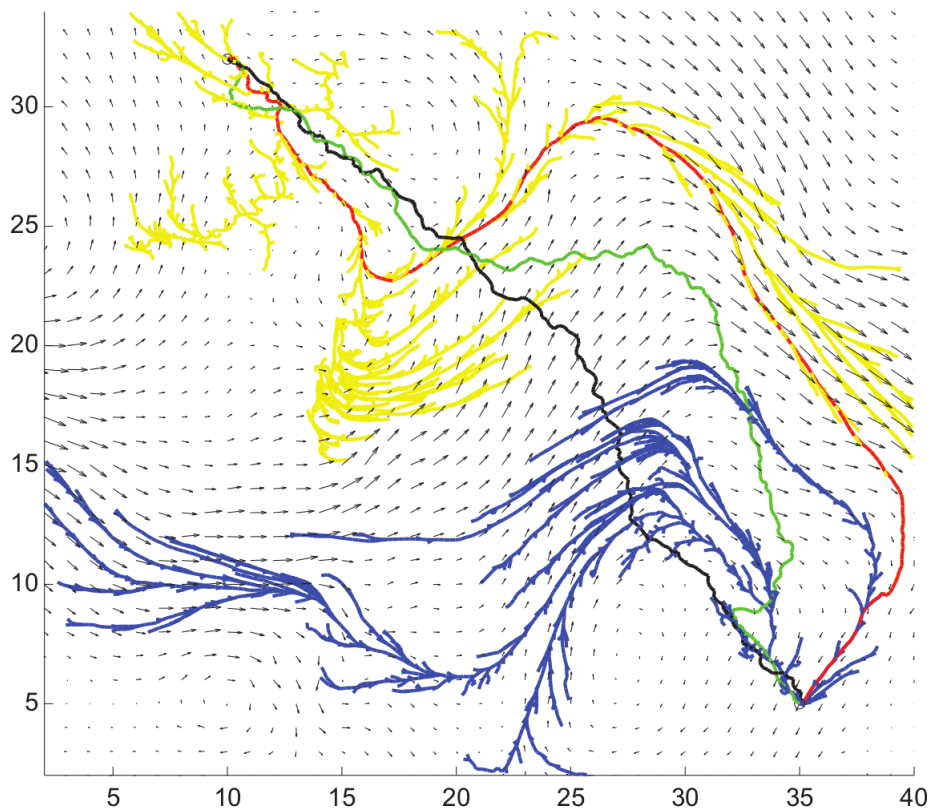


FIGURE 2.3 – Arbres VF-RRT bi-directionnels dans un vent bidimensionnel. Jaune : Un arbre typique émanant du noeud de départ. Bleu : Un arbre typique émanant du noeud d’arrivée. Chemins produits par les algorithmes RRT (noir), T-RRT (vert) et VF-RRT (rouge) (extrait de [KKP14]).

forcement de politiques d’un MDP (Q-learning), SARSA(λ). Les auteurs élaborent là aussi une fonction empirique de gain basée sur un compromis entre exploration de la zone et exploitation des courants ascendants pour gagner de l’altitude. Ils testent ensuite celle-ci sur plusieurs variantes de calcul de MDP (gloutonnes ou non) et d’approximation de la fonction d’utilité. Ils démontrent alors la capacité de leur algorithme à apprendre, après un certain nombre d’itérations, à exploiter le champ thermique et ainsi prendre de l’altitude (fig. 2.5). Cependant, deux points sont à considérer. Le premier est que les résultats les plus probants le sont sur les variantes les plus gourmandes en temps de calcul. Les variantes plus légères montrent des performances très dégradées. Le second est que le champ de courant ne varie pas entre les instances d’apprentissage, il est donc pertinent de se demander si la politique apprise sera suffisamment générale en conditions réalistes plus variées.

C’est en effet un point commun à la plupart des articles de l’état de l’art : les résultats obtenus le sont en simulation sur des scénarios fictifs. Les champs de courants sont artificiels (par exemple générés par des combinaisons de sinusoides). De même, le modèle de thermique est dans tous les cas généré par une

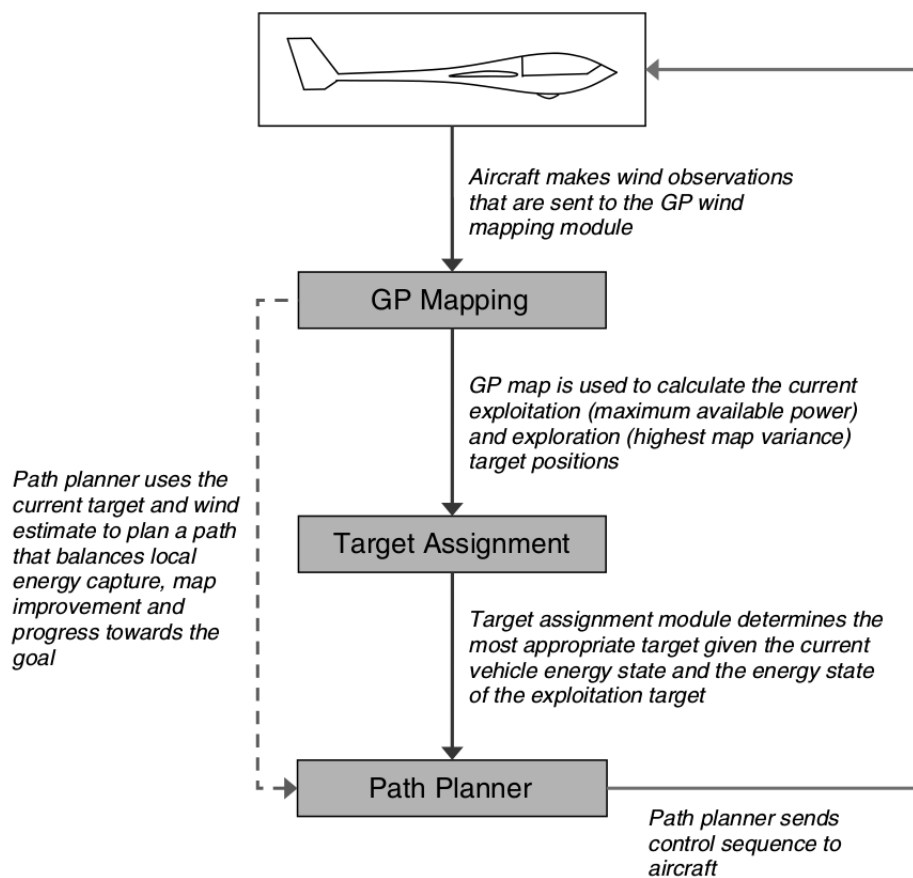


FIGURE 2.4 – Vue d’ensemble de l’architecture système de planification permettant exploration et exploitation simultanée de l’environnement pour un UAV de type planeur (extrait de [LS11]).

simple fonction gaussienne. Sur ce point, deux articles se distinguent.

Dans le cadre du projet VORTEX2, l’article [EA14] explore la possibilité d’envoyer des UAVs explorer les caractéristiques d’orages supercellulaires. Exploitant les gradients intenses de vents caractéristiques de ce type de phénomènes météorologiques, les auteurs mettent en évidence la possibilité de gagner de l’énergie par vol de gradient (*dynamic soaring*). Si l’environnement est là, considéré comme connu et la planification rudimentaire, le champ de courant est issu d’une simulation météorologique d’orage et peut donc être considéré comme suffisamment réaliste pour valider des résultats en simulation.

Quant au travail présenté par les auteurs de [RVS13], il est remarquable par plusieurs aspects. Pour commencer, le scénario présenté est extrêmement proche de celui du projet SkyScanner : « suivre et cartographier des structures [météorologiques] cohérentes », comme par exemple des nuages ou des panaches de fumées. Ensuite, les résultats présentés sont issus de vols réels, ce qui est extrêmement rare et témoigne d’un réel investissement en terme d’intégration. Un UAV est utilisé pour cartographier une colonne de fumée. Les trajectoires sont générées par perturbations et raffinements successifs d’un plan initial par une

dans ce type de problèmes, l'utilisation de solutions exactes semble vouée à l'échec, et l'état de l'art va dans ce sens : toutes les références planifiant en trois dimensions utilisent des techniques approchées. Dans ce contexte plusieurs pistes d'amélioration et de validation des solutions envisagées jusqu'à présent peuvent être dégagées à partir du tableau comparatif 2.1 :

- Prise en compte de l'incertitude dans la planification
- Proposition de solutions multi-agents
- Validation des algorithmes dans une plate-forme de simulation réaliste des phénomènes et processus réels
- Validation des systèmes en effectuant des vols réels

Plus particulièrement dictés par la spécificité du projet SkyScanner, deux autres points nous semblent d'une importance fondamentale : la prise en compte de la dynamique des phénomènes météorologiques en jeu, et la mise en place de fonctions de décision multicritère (énergie, information) plus solides.

En toile de fond, il nous apparaît crucial de considérer les modèles de l'environnement des UAVs. Les résultats de raisonnements sur le monde, par exemple la planification de trajectoires, sont intimement liés aux modèles du monde et aux différentes couches d'abstraction qu'ils impliquent. Un modèle trop simple accélérera la planification mais dégradera la qualité des solutions, au point de les rendre irréalisables [Sou11]. Au contraire, un modèle trop complexe souffrira de cette complexité qui se répercute dans les algorithmes utilisés, au risque que ceux-ci ne puissent plus tourner en temps réel ou dans une version trop dégradée pour obtenir des solutions satisfaisantes. Considérer les liens forts qui lient algorithmes et modèles nous semble une piste intéressante pour rendre les systèmes plus robustes, et moins dépendants de réglages manuels. C'est ce que nous développerons dans les chapitres suivants.

2.3 L'approche Skyscanner

Au vu de la dimensionnalité de la fonction à optimiser ainsi que de la taille d'un nuage, il semble inadéquat de vouloir planifier directement au plus bas niveau la trajectoire des agents. On imagine donc de manière standard une solution hiérarchique.

Au niveau le plus haut, la planification manipule des *recettes*, qui mettent en jeu un certain nombre d'agents pour effectuer des tâches simples. Par exemple, pour mesurer la hauteur du nuage, il faut au moins deux UAVs se synchronisant, l'un descendant à la base du nuage, l'autre cherchant le sommet de celui-ci. Après discussion avec les météorologues, il ressort qu'on peut distinguer deux types de tâches, correspondant à deux niveaux d'intérêt : d'une part, la mesure de grandeurs globales du nuage (taille, forme) et de l'autre, la mesure locale de grandeurs microphysiques spécifiques dans une zone donnée (par exemple, taille des gouttelettes au sommet du nuage). Ce niveau de planification permet de gérer de manière globale le mouvement des drones pour réaliser et combiner au mieux ces tâches.

Pour la suite, on supposera qu'il existe un tel niveau de planification, spécifiant les objectifs à court terme des UAVs. On s'intéresse alors au plus bas niveau de planification, qui manipule les trajectoires d'un ou plusieurs agents en vue d'optimiser la prise d'information et d'énergie locale. Ce type de planification

est aussi communément appelé *guidage* en aviation.

2.4 Planification de trajectoires

L'optimisation locale des trajectoires des UAVs doit reposer sur différents modèles :

- modèle de l'environnement
- modèle cinématique et énergétique de l'UAV

ainsi que sur une *fonction d'utilité multicritère*, qui permet de sélectionner la meilleure trajectoire à partir des informations disponibles dans les différents modèles.

2.4.1 Modèle d'avion

Le modèle d'avion choisi est un modèle cinématique simple, à la fois suffisant pour les besoins de la planification mais suffisamment simple pour pouvoir tester rapidement un grand nombre de trajectoires. De plus, en première approche, nous avons choisi de considérer un scénario de type planeur, c'est à dire sans influence d'un moteur. Les UAVs utilisés dans le projet étant dotés d'une hélice, l'incorporation d'un modèle moteur devra être considérée par la suite : un tel modèle est actuellement en cours d'élaboration avec l'ENAC.

On note \vec{V} le vecteur vitesse air de l'avion, c'est à dire la vitesse de l'UAV relative au vent local. La vitesse air dépend essentiellement de l'angle d'attaque de l'avion. Or, dans le cas de micro-drones dans un scénario d'optimisation énergétique et de récolte d'information, une trajectoire trop dynamique est pénalisante à la fois d'un point de vue énergétique et d'un point de vue de la prise d'information (dégradation de précision des mesures). En première approche, on peut donc considérer $\|\vec{V}\| = V_a$ constante. Le vecteur vitesse air de l'UAV est entièrement déterminé par le cap, c'est à dire la direction de l'avion sur le plan (x, y) , noté ψ et l'angle de descente noté γ .

La dynamique dans le référentiel terrestre d'un avion à un moment t est donc décrite par sa position $p = (x, y, z)^\top$ dans le référentiel terrestre, son cap ψ , sa variation associée $\dot{\psi}$. Pour un vent tridimensionnel $\vec{c} = (c_x, c_y, c_z)^\top$ au point $p(t)$, on approxime le mouvement du drone par les équations :

$$p(t + dt) = p(t) + (\vec{V} + \vec{c})dt \quad (2.5)$$

$$\psi(t + dt) = \psi(t) + \dot{\psi}(t)dt \quad (2.6)$$

Le vecteur vitesse est déterminé grâce aux équations suivantes, dont les calculs

sont détaillés en annexe A :

$$\vec{V} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \quad (2.7)$$

$$v_x = V_a \cos \gamma \cos \psi \quad (2.8)$$

$$v_y = V_a \cos \gamma \sin \psi \quad (2.9)$$

$$v_z = V_a \sin \gamma \quad (2.10)$$

$$\gamma = \arctan \left(\frac{1}{\frac{L}{D} \cos \theta} \right) \quad (2.11)$$

$$\theta = \arctan \left(\frac{\dot{\psi} V_a}{g} \right) \quad (2.12)$$

Il suffit alors de spécifier V_a et $\dot{\psi}$ pour commander le micro-drone. Leurs valeurs sont contraintes par les capacités de celui-ci : il est nécessaire de fixer des limites $\dot{\psi} \in [\dot{\psi}_{min}, \dot{\psi}_{max}]$ telles que celui-ci sera capable de conserver V_a constante à la valeur voulue sans se mettre en danger de décrochage. Les caractéristiques aérodynamiques du drone sont simplifiées au maximum et résumées par le coefficient $\frac{L}{D}$ qui représente la *finesse* de l'avion, c'est à dire le rapport entre la distance horizontale parcourue et la hauteur perdue par unité de temps. Ce coefficient dépendant uniquement de la vitesse air, et celle-ci étant considérée constante dans notre cas, ce coefficient est également constant.

2.4.2 Planification par optimisation stochastique

A partir du modèle d'avion et de la connaissance du champ de vent local, il est ainsi possible de reconstruire une trajectoire physique approchée à partir d'un vecteur de commandes envoyées à l'autopilote. L'approche retenue, semblable à celle de [LS11] est la suivante : soit dt un pas de temps fixé correspondant à la fréquence de modification des commandes, et Δ_t un horizon de planification. Une trajectoire peut-être représentée par un vecteur $\alpha \in \mathbb{R}^n$ avec $n = \lceil \frac{\Delta_t}{dt} \rceil$.

L'espace des commandes est exploré directement par échantillonnage : un ensemble de trajectoires α sont générées aléatoirement, puis sont transformées en coordonnées cartésiennes à partir des équations 2.6. Une fonction d'utilité est alors utilisée comme boîte noire pour évaluer et trier les différentes trajectoires générées. La plus prometteuse est alors optimisée par l'algorithme SPSA (Simultaneous Perturbation Stochastic Approximation)[Spa05] pour obtenir un maximum local.

Le chapitre suivant s'intéresse à la génération de la carte de vent locale, ainsi qu'à la manière d'utiliser le modèle qui en est fait pour déterminer l'utilité informative d'une trajectoire proposée par l'algorithme d'optimisation stochastique.

Chapitre 3

Modélisation de l'environnement

Dans une application robotique des algorithmes issus de l'Intelligence Artificielle, le(s) modèle(s) du monde qui entoure(nt) le ou les robots est(sont) d'une importance capitale. La modélisation sert de base de travail pour les algorithmes, et leurs performances sont directement liées à la qualité des modèles. Pour qu'un robot puisse correctement décider et agir, il faut qu'il soit capable de percevoir son environnement avec une précision suffisante et susceptible d'entretenir différents niveaux d'abstraction des modèles correspondants suivant les besoins des processus de décision.

Pour les besoins de la navigation des UAVs dans les nuages, il faut être capable :

- de localiser le nuage et d'identifier les zones d'intérêt pour la mission (et donc d'avoir une idée de la forme/structure générale du nuage)
- d'entretenir une carte des vents suffisamment précise pour pouvoir planifier des trajectoires intéressantes tant au point de vue énergétique qu'au niveau du gain d'information. Cette carte des vents peut éventuellement faire partie de l'objectif de la mission, à côté d'autres grandeurs comme la température, la pression ou l'humidité. L'UAV doit donc être capable de prédire la valeur de certaines grandeurs physiques en tout point du nuage et être en état d'évaluer la justesse de sa précision, c'est à dire l'incertitude associée à celle-ci.

Ces deux besoins correspondent à deux niveaux d'abstraction de modèles différents. Le premier modèle, appelé "modèle paramétrique" dans la suite de ce document, correspond à un modèle conceptuel géométrique *moyen* de nuage, instancié à partir d'un nombre réduit de paramètres estimés au cours de la mission. Un tel modèle n'existe pas encore dans la littérature, de l'avis des météorologues avec lesquels nous avons mené les discussions. Un des enjeux du projet est donc l'élaboration d'un tel modèle, intéressant tant pour pouvoir remplir les objectifs de la mission qu'en tant que tel comme modèle de cumulus pour la météorologie. En cours d'élaboration à partir de données de simulations numériques par l'équipe du CNRM, nous ne présenterons pas un tel modèle

dans ce document.

Mais pour pouvoir planifier des trajectoires, il faut un modèle plus fin estimant les grandeurs physiques nécessaires à la prédiction de la trajectoire (essentiellement le vent), mais aussi celles permettant d'évaluer l'utilité de celle-ci vis à vis de la mission (vent associé à des variables telles que température, pression, humidité et voire même taille de gouttelettes en suspension). De par la vitesse relativement faible du drone (à peu près 10 m/s) et son poids réduit (inférieur à 2 kg), celui-ci est particulièrement sensible aux vents et turbulences présents dans les nuages. Par ailleurs, la bonne manœuvrabilité de ce type de mini-drones à voilure fixe permet d'envisager une réponse rapide aux fluctuations environnementales, et en particulier de profiter des courants ascendants se présentant souvent sous forme de "bulles" ascensionnelles.

Prédire ces grandeurs en tous points à l'échelle d'un nuage pose deux problèmes principaux : la taille de l'environnement considéré, dans lequel sont prises des mesures ponctuelles très éparses, et le caractère dynamique des phénomènes en jeu. Il faut de plus ne pas négliger le bruit relativement important pouvant perturber les mesures, en particulier au niveau du vent. C'est pourquoi, un modèle de régression spatio-temporel capable de travailler avec des mesures éparses et bruitées, avec un temps de calcul raisonnable, et permettant d'estimer l'erreur commise sur les prédictions est donc indispensable. Il existe très peu d'outils efficaces permettant d'exploiter ce type de données.

3.1 État de l'art

Une possibilité de modélisation est d'utiliser des Champs aléatoires de Markov (MRF - Markov Random Field), très populaires pour l'analyse et la reconstruction d'images, solution envisagée dans [WS12]. L'environnement considéré est plan, et un MRF disposé selon une grille estime la valeur d'un processus (ici la température ou la concentration en chlorophylle sur la surface océanique). Les agents utilisent alors les valeurs des marginales des variables du processus aux points proches de leur position courante pour estimer, via une régression linéaire, leur gradient local. Celui-ci est alors utilisé pour guider une loi de contrôle qui leur fait suivre une courbe de niveau prédéfinie. Les croyances à posteriori sur les variables du processus, conditionnées par les observations effectuées par un agent, sont calculées localement puis échangées entre agents par passage de message le long d'un hyperarbre prédéfini (fig. 3.1). Cela permet de distribuer les calculs d'inférence, relativement lourds sur ce type de modèles graphiques. C'est le principal inconvénient de ce genre de modèle : les méthodes d'inférence utilisées sont souvent assez coûteuses et la convergence n'est pas garantie.

Il est donc nécessaire de faire des hypothèses supplémentaires sur les distributions de probabilités en jeu pour simplifier les calculs. Une hypothèse fréquemment effectuée est de supposer que les variables aléatoires suivent une loi de probabilité jointe gaussienne multivariée. On parle alors de processus gaussiens (GP - Gaussian Process). Sous cette hypothèse, le processus est entièrement défini par sa moyenne et sa variance. La moyenne est souvent considérée nulle et la variance est définie par la matrice de covariance entre les variables. Si on peut directement appliquer cette hypothèse sur un modèle graphique, on peut

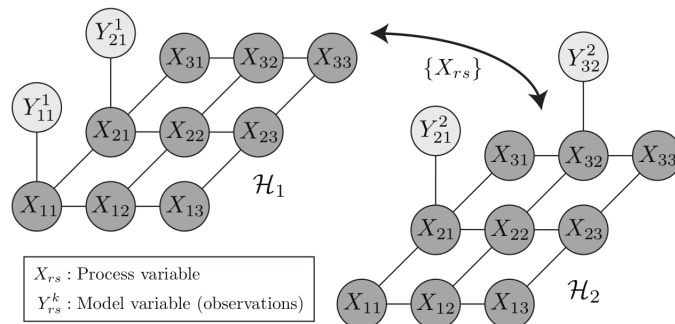


FIGURE 3.1 – Champ aléatoire de Markov multi-agents 3×3 , avec deux sous-graphes se partageant les variables de processus $\{X_{rs} : r, s = 1, \dots, 3\}$. Chaque agent maintient un ensemble de variables de modèle $\{Y_{rs}^k : r, s = 1, \dots, 3, k = 1, 2\}$ sur lesquelles les observations sont réalisées (toutes ne sont pas montrées pour plus de clarté). La structure d’hyperarbre (permettant le passage de message pour l’inférence) est $\mathcal{H}_1 - \mathcal{H}_2$ avec pour interface $\{X_{rs}\}$ (extrait de [WS12])

s’affranchir de la rigidité des MRF (et de l’hypothèse de Markov associée¹) et raisonner directement sur l’expression mathématique de ce type de processus. Cette technique appelée régression par processus gaussiens ou krigeage a été d’abord utilisée en géostatistiques avant de connaître un certain essor dans d’autres domaines d’application, en particulier l’apprentissage [RW06] (voir section 3.2 pour une brève introduction à la régression par processus gaussiens).

Ce modèle présente de nombreux avantages : il demande un nombre réduit de paramètres, permet d’estimer l’erreur commise à chaque prédiction et est relativement flexible. En effet, pouvant être vu comme une généralisation des machines à noyaux (type SVM), la covariance entre deux points est estimée par une fonction de noyau dont la forme analytique permet de fixer un a priori sur celle du processus (dérivabilité, stationnarité, etc...) et dont le paramétrage permet une plus grande souplesse (mais nécessite l’utilisation de techniques d’apprentissage). Enfin, il permet la prise en compte et l’estimation du bruit directement pendant la régression (fig. 3.2). Ces avantages se paient cependant par un coût algorithmique non négligeable.

En effet, si pour n exemples, la prédiction d’un nouveau point s’effectue en $\mathcal{O}(n^2)$ opérations, dans le cas général, l’apprentissage du modèle nécessite $\mathcal{O}(n^3)$ opérations. De plus, l’optimisation d’hyper-paramètres (on parle de la régression par processus gaussiens comme d’un modèle *semi-paramétrique*), nécessite de faire varier le modèle jusqu’à convergence et peut donc nécessiter un temps de calcul conséquent.

Un certain nombre de références récentes commencent à utiliser ces modèles dans différents contextes robotiques. Ainsi [HPOG12] et [DHP⁺13] utilisent des GP pour modéliser courants et variables biologiques dans l’océan. Ces modèles

1. L’hypothèse de Markov associée aux MRF dit que, sachant les voisins directs d’une variable dans le modèle graphique, celle-ci est indépendante de toutes les autres. Dans le domaine temporel, cela revient à dire que l’état futur du monde à la prochaine étape ne dépend que de l’état courant.

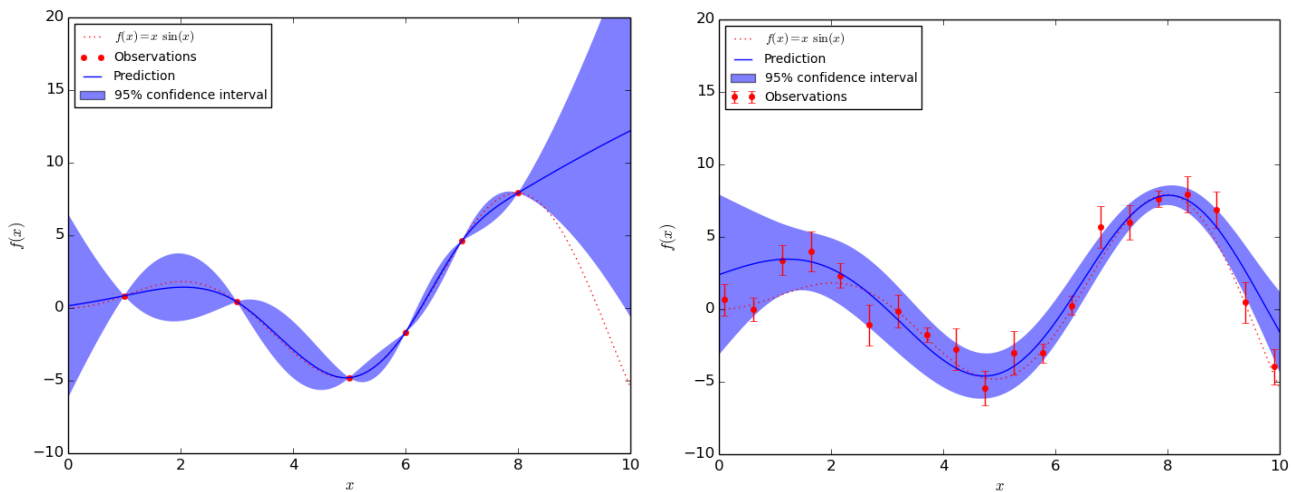


FIGURE 3.2 – Exemple de régression par processus gaussiens. En pointillé rouge, la fonction source ; en bleu, la fonction moyenne prédite à partir des observations (points rouges) . La zone bleue correspond à l’intervalle de confiance à 95% de la prédiction. A gauche, sans présence de bruit, à droite, avec un bruit gaussien (exemple introductif de la librairie scikit-learn).

permettent de guider la navigation de véhicules autonomes sous-marins dans des missions de récoltes de données. Ces applications sont très proches en essence de notre problématique, si ce n’est que les échelles de temps d’évolution dans le milieu aquatique sont beaucoup plus grandes, et permettent donc de s’affranchir de problèmes de rapidité d’exécution des algorithmes.

Plus proche de notre application, [LS11] et [ASGS13] s’intéressent à ces modèles pour cartographier un champ de vents tridimensionnel. Dans [LS11] les auteurs proposent de modéliser le vent par trois processus gaussiens indépendants dans chaque dimension. Ils introduisent également une formulation spatio-temporelle permettant de prendre en compte une estimation de la dérive moyenne du vent. Si cette formulation est intéressante, le vent est produit artificiellement par quelques formules mathématiques, et la dérive du vent introduite expressément. L’incertitude sur les prédictions est utilisée par une fonction d’utilité ad hoc qui implémente une politique d’exploration qui surestime le gain énergétique d’une trajectoire quand le drone a suffisamment d’énergie. Les auteurs ne quantifient pas autrement la qualité du modèle que par le fait que l’erreur quadratique moyenne diminue avant d’approcher un palier, et que le drone se sert de celui-ci pour gagner de l’énergie durant la navigation. Le coût algorithmique est limité par l’utilisation dans le modèle d’un nombre fixe maximal d’exemples (les plus vieux, moins pertinents étant oubliés au fur et à mesure grâce à une heuristique ad hoc), et par l’apprentissage hors-ligne des paramètres des noyaux. Dans [ASGS13], le processus gaussien estime aussi un champ de vent tridimensionnel, mais cette fois-ci pour piloter un processus de décision markovien appris par renforcement. Les auteurs introduisent pour cela une me-

1. <http://scikit-learn.org/>

sure de la pertinence d'une trajectoire par le gain d'information qu'elle suscite par diminution de l'incertitude : ils intègrent analytiquement la prédiction de l'erreur (variance) sur le domaine spatial d'intérêt considéré. Intéressante, leur technique se limite néanmoins aux noyaux intégrables (à moins d'utilisation de techniques d'échantillonnage), et le coût de l'intégration n'est pas explicité. Est cependant mentionné un autre moyen d'estimer le changement de l'incertitude par calcul de la divergence de Kullback-Leibler². Enfin, les auteurs évaluent deux implémentations de GP : une gardant tous les exemples, l'autre incluant une technique parcimonieuse (*sparsification*) gardant un nombre fixe d'exemples issus des travaux de [CO02]. Un des principaux résultats est la perte de qualité assez conséquente des solutions en utilisant cette dernière solution.

Ces deux exemples - à notre connaissance les seuls dans un contexte drone - introduisent les différentes problématiques posées par l'utilisation de processus gaussiens pour modéliser les variables d'intérêt :

- Comment choisir un noyau (et quelles informations doit-il encoder), quand et comment apprendre les hyper-paramètres ?
- En particulier, comment prendre en compte la dimension temporelle par rapport aux dimensions spatiales ?
- Peut-on raisonnablement faire l'hypothèse que les différentes variables modélisées sont indépendantes ?
- Comment utiliser au mieux les prédictions de l'erreur commise pour prendre des décisions sous incertitudes ?
- Comment réduire le coût algorithmique sans sacrifier pour autant les performances en prédiction ?

Sur ce dernier point, au-delà de la croissante puissance de calcul, un certain nombre de travaux récents permettent d'envisager l'utilisation de processus gaussiens en temps réel pour la planification embarquée. Ces travaux, inspirés soit du besoin de traiter des données de plus en plus nombreuses (*big data*) soit de plus en plus rapidement (robotique), sont axés autour de deux idées.

La première, déjà mentionnée, est l'utilisation parcimonieuse des données. L'idée introduite par [CO02] et dont de nombreuses variantes existent, consiste à fixer à priori un nombre maximal d'exemples et soit supprimer ceux qui ne sont pas assez informatifs (redondants), soit à peupler à priori l'espace d'exemples fictifs puis à raffiner l'estimation à priori de leur valeur en s'appuyant sur les nouveaux exemples arrivants avant de les oublier. Cette technique supposant une perte d'information, elle peut dans le cas général compromettre la qualité de la prédiction si ce nombre d'exemples à priori est mal choisi. Dans le cas général, (processus *localisé* et dynamique) si on ne veut pas perdre d'information, il est impossible de s'affranchir de la limite fixée par le théorème de Shannon, c'est à dire un échantillonnage à deux fois la fréquence maximale du processus à estimer.

La deuxième reprend une idée classique en algèbre linéaire : travailler avec des matrices creuses. Le coût algorithmique provient en effet essentiellement de manipulations de la matrice de covariance entre les exemples. Si celle-ci est suffisamment creuse, il est possible d'accélérer les calculs en ne considérant dans

². La divergence de Kullback-Leibler est une mesure non symétrique de la différence entre deux distributions de probabilités continues

ceux-ci que les éléments non nuls. Il faut d'abord s'assurer que la matrice de covariance est suffisamment creuse : il faut donc trouver des fonctions de noyaux intrinsèquement éparses dites "à support compact". Diverses solutions existent, la difficulté étant que la fonction satisfasse les critères d'une fonction de noyau (voir section 3.2), tout en étant représentative des corrélations spatiales qu'elle encode. Dans [MR09] les fonctions sinus et cosinus tronquées et centrées en zéro sont envisagées, dans [VV10] c'est une famille de fonctions polynomiales par parties qui est utilisée. Dans les deux cas, la qualité des solutions ne semble pas être impactée sur les données considérées, la rapidité de décroissance (dictée par le paramètre de *longueur caractéristique* propre à la plupart des noyaux) semble être plus importante que la formulation exacte de la fonction de covariance. Enfin, [RYH11] s'appuie sur cette idée pour proposer une implémentation de processus gaussiens *en ligne*, c'est à dire permettant à moindre coût de mettre à jour le modèle par de nouveaux exemples et présentant des performances compatibles avec une utilisation temps-réel. Cet algorithme est ensuite utilisé sur une application concrète (estimation de la pose de la tête d'un homme à partir du flux d'images) et comparé avec les autres implémentations de l'algorithme. Non seulement la qualité de l'estimation est au moins comparable à l'implémentation standard - et bien meilleure que celle fixant un nombre maximal d'exemples - mais l'algorithme est capable de maintenir la matrice suffisamment creuse et ainsi être beaucoup plus rapide tant en prédiction qu'en apprentissage du modèle.

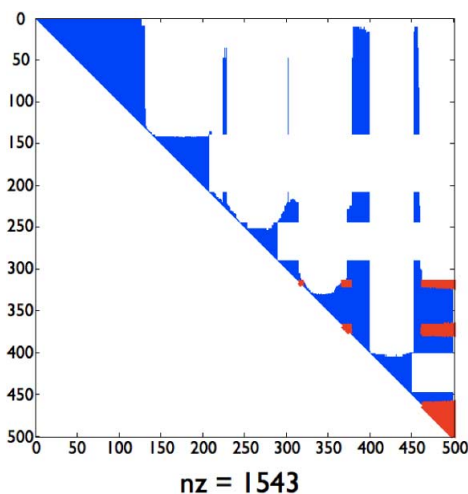


FIGURE 3.3 – Mettre à jour un facteur de Cholesky creux après avoir ajouté une ligne et une colonne coûte $\mathcal{O}(n)$ en temps en utilisant des rotations de Givens pour une matrice $n \times n$, mais en pratique souvent beaucoup moins. Dans le pattern creux du facteur de Cholesky (entrées non nulles marquées en bleu ou rouge), les entrées dont la valeur a changé au cours de la mise à jour sont marquées en bleu, celles inchangées en rouge. Dans cette matrice 500×500 seulement environ 1500 entrées sont modifiées (extrait de [RYH11]).

3.2 Régression par processus gaussiens

Au vu de leur importance centrale dans ce qui suit, et pour éclairer l'état de l'art à un lecteur non spécialiste, il semble nécessaire d'introduire ici quelques notions de base des processus gaussiens et de leur utilisation pour la régression. Il existe plusieurs manières équivalentes pour introduire ce concept, celle qui a été choisie ici est tirée de l'ouvrage de référence de C. E. Rasmussen & C. K. I. Williams : *Gaussian Processes for Machine Learning*, 2006 [RW06].

La notion de processus stochastique peut se voir comme une généralisation de distributions de probabilités au domaine fonctionnel. Là où la première définit une distribution sur *une variable aléatoire de dimension finie*, la notion de processus étend donc cette définition à *une infinité de variables aléatoires*. Quand on se restreint aux processus qui sont *gaussiens*, il s'avère qu'il est alors possible d'apprendre et d'inférer sur ces processus à un coût relativement raisonnable.

Plus formellement, *un processus gaussien est une collection de variables aléatoires, pour laquelle toute partie finie a une probabilité jointe gaussienne*. [RW06].

Le processus est alors entièrement défini par sa fonction moyenne $m(\mathbf{x})$ et sa fonction de covariance ou *noyau* $k(\mathbf{x}, \mathbf{x}')$, on note :

$$\begin{aligned} m(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})], \\ k(\mathbf{x}, \mathbf{x}') &= \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))], \end{aligned} \quad (3.1)$$

le processus gaussien s'écrit alors :

$$f(\mathbf{x}) = GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \quad (3.2)$$

Ici, les variables aléatoires sont les valeurs $f(\mathbf{x}) = y$ aux points \mathbf{x} . Pour la suite et pour plus de simplicité, on pose $m(x) = 0$, ce qui est le plus souvent le cas dans les exemples applicatifs, mais pas forcément nécessaire.

Dans le cadre de la régression, on observe un ensemble $(\mathbf{X}, \mathbf{y}) = \{(\mathbf{x}_i, y_i) | i = 1..n\}$ d'exemples pour lesquels on a mesuré une réalisation particulière du processus. On cherche alors la distribution de fonctions à posteriori, c'est à dire celle en accord avec les observations.

En termes probabilistes, cela revient à conditionner la probabilité gaussienne jointe par les observations. Puis, à partir de cette distribution, on cherche à prédire l'espérance et la variance d'un point quelconque \mathbf{x}_* (typiquement non présent dans l'ensemble d'exemples. Là encore, on conditionne la distribution de probabilité par le point \mathbf{x}_* . C'est une opération relativement simple, et on obtient finalement :

$$\begin{aligned} \bar{y}_* &= \mathbb{E}[y_* | \mathbf{x}_*, \mathbf{X}, \mathbf{y}] = K(\mathbf{x}_*, \mathbf{X})K(\mathbf{X}, \mathbf{X})^{-1}\mathbf{y}, \\ \mathbb{V}[y_*] &= \mathbb{E}[(y_* - \bar{y}_*)^2] = k(\mathbf{x}_*, \mathbf{x}_*) - K(\mathbf{x}_*, \mathbf{X})^\top K(\mathbf{X}, \mathbf{X})^{-1}K(\mathbf{x}_*, \mathbf{X}) \end{aligned} \quad (3.3)$$

Où $K(\mathbf{X}, \mathbf{X})$ représente la matrice $n \times n$ issue de l'application de la fonction k à toutes les paires d'éléments de X , aussi appelée matrice de covariance et alors

dénotée Σ . $K(\mathbf{x}_*, \mathbf{X})$ est de la même manière le vecteur $1 \times n$ de covariance entre x_* et les éléments du vecteur d'exemples \mathbf{X} .

Cette formule permet directement de se faire une idée du coût algorithmique de l'inférence. En supposant pré-calculée Σ^{-1} , calculer la moyenne ou la variance pour un point à partir des équations 3.3 se fait en $\mathcal{O}(n^2)$, coût dominé par le produit matriciel entre matrices de taille $1 \times n$ et $n \times n$ en $\mathcal{O}(1 \times n \times n) = \mathcal{O}(n^2)$. Quant au pré-calcul de la matrice de précision Σ^{-1} , il se fait en $\mathcal{O}(n^3)$ dans le cas général. En effet, si des algorithmes plus rapides existent, la décomposition de Cholesky est le plus souvent préférée pour sa meilleure stabilité numérique, et parce qu'il est possible d'effectuer certains calculs utiles directement à partir de la forme factorisée.

Cette formulation suppose cependant des exemples non bruités. Prendre en compte un bruit additif gaussien se fait de manière directe : il suffit d'ajouter la variance du bruit à la diagonale de la matrice de covariance Σ . On pose alors $\Sigma' = \Sigma + \sigma_n I$ pour un bruit homoscédastique³, ou $\Sigma' = \Sigma + \text{diag}(\sigma_{\mathbf{n}})$ si il est hétéroscédastique⁴.

Reste donc à spécifier la fonction de covariance, c'est à dire le noyau, k . Il est possible de choisir n'importe quel type de fonction, pourvu qu'elle respecte la condition d'être *positive, semi-définie et symétrique* (ce qui est par ailleurs nécessaire pour la décomposition de Cholesky et garantir l'inversibilité de Σ). D'un point de vue conceptuel, la covariance peut être vue comme encodant une notion de similarité entre deux points du processus. Instinctivement, deux points proches dans l'espace ou dans le temps auront des valeurs probablement proches dans la plupart des processus naturels pourvu que le processus soit suffisamment lisse (i.e. dérivable). La forme analytique de la covariance permet donc de mettre un à priori sur le type de processus modélisé : stationnarité, isotropie, dérivabilité. La covariance la plus couramment utilisée est la fonction exponentielle carrée :

$$k_{SE}(x, x') = \sigma_f e^{-\frac{|x-x'|^2}{2l^2}} \quad (3.4)$$

Ou sa variante :

$$k_{SE}(\mathbf{x}, \mathbf{x}') = \sigma_f e^{-\frac{1}{2}|\mathbf{x}-\mathbf{x}'| M |\mathbf{x}-\mathbf{x}'|} \quad (3.5)$$

Avec $M = \text{diag}(1/l^2)$, \mathbf{l} étant un vecteur de mêmes dimensions que \mathbf{x} . Ce noyau est stationnaire⁵ et isotrope⁶, dans sa version 3.4. La version 3.5 est par contre anisotrope et permet de spécifier l'importance de chaque dimension via le vecteur l , agissant comme des *longueurs caractéristiques*. Il est de plus infiniment dérivable. Cette dernière caractéristique n'est pas forcément très réaliste dans un grand nombre de processus réels, et d'autres types de noyaux, par exemple ceux de la classe dite de *Matern* permettent de spécifier la dérivabilité par un hyper-paramètre. En dimension d , le noyau défini par l'équation 3.5 comporte $d + 1$ paramètres. On ne se donne donc non pas un noyau, mais une famille de noyaux, dans laquelle il va falloir choisir celui dont la valeur des paramètres

3. bruit de variance uniforme

4. bruit de variance non-uniforme

5. invariant en translation

6. invariant en rotation

est la plus appropriée. Le modèle peut de plus comporter un certain nombre de paramètres libres supplémentaires, par exemple ceux associés au bruit (si le niveau de bruit est à priori inconnu).

Il existe de nombreuses méthodes pour sélectionner le bon modèle (c'est à dire fixer les hyper-paramètres des noyaux et du bruit), et il s'avère qu'elles sont toutes à peu près équivalentes quant au coût computationnel. Dans [RW06] deux techniques standard sont explicitées.

La première et la plus largement utilisée est l'*inférence bayésienne* : à partir d'un vecteur d'hyper-paramètres donné θ , on peut facilement calculer la probabilité marginale des exemples $p(\mathbf{y}|\mathbf{X}, \theta)$. En pratique, on calcule le log de la probabilité pour des raisons numériques :

$$\log p(\mathbf{y}|\mathbf{X}, \theta) = -\frac{1}{2}\mathbf{y}^\top \Sigma^{-1}\mathbf{y} - \frac{1}{2}\log |\Sigma| - \frac{n}{2}\log 2\pi \quad (3.6)$$

On cherche alors θ maximisant cette probabilité marginale. En dérivant l'équation 3.6, il est possible d'effectuer une optimisation par descente de gradient. Cependant, rien ne garantit la présence d'un optimum global, et il faut donc en pratique échantillonner différentes valeurs initiales de θ pour obtenir un meilleur optimum possible. Il est intéressant de noter que cette expression permet directement d'obtenir un équilibre entre sous- et sur-apprentissage : le premier terme valorise un modèle qui explique au mieux les données, tandis que le second pénalise un modèle trop complexe.

Optimiser la probabilité marginale des exemples permet de trouver le modèle qui explique au mieux les exemples, ce qui est différent d'obtenir celui qui aura les meilleures performances prédictives. Une possibilité est d'utiliser des techniques de *validation croisée* : calculer la probabilité marginale prédictive du modèle sur des exemples test, celui-ci étant entraîné sans prendre en compte ceux-ci.

Dans tous les cas c'est la complexité algorithmique de l'optimisation des hyper-paramètres θ qui domine le coût global de l'utilisation de modèles de régression par processus gaussiens, et ce par plusieurs ordres de magnitude. Cela pose problème en présence d'un grand nombre de données, ou si le modèle est construit en temps réel au fur et à mesure de l'arrivée d'exemples.

3.3 Intégrer une connaissance à priori : moyennes et noyaux

Comme explicité précédemment, il existe deux possibilités pour essayer de contraindre le modèle de régression par des connaissances à priori du processus :

- *Biais* le prédicteur, c'est à dire considérer une moyenne $m(\mathbf{x})$ non nulle
- Spécifier une famille de noyaux la mieux adaptée aux caractéristiques du processus

Pour le premier point, une possibilité serait dans notre cas de nous servir du modèle conceptuel développé à partir de simulations par les météorologues, qui est par essence une compilation moyenne statistique. On imagine qu'un tel

modèle pourrait spécifier un nuage moyen à partir de quelques caractéristiques (taille, diamètre, entraînement à la base du nuage, vents à différentes altitudes). Ces paramètres seraient estimés pendant la mission par les mesures prises par les drones, et donc par le modèle de régression. En retour, le modèle paramétrique fournirait une estimation des variables atmosphériques là où les mesures sont trop peu denses, voire inexistantes. A l’heure de l’écriture de ce rapport, le modèle conceptuel est toujours en cours d’élaboration, et n’est donc pas utilisé par la suite.

En ce qui concerne le choix du noyau, il semble de par l’état de l’art ([MR09], [VV10]) que certaines propriétés sont plus importantes que d’autres. En particulier sur des données réelles, bruitées, la dérivabilité du noyau et sa forme exacte importe relativement peu. Au contraire, les hypothèses de stationnarité et d’isotropie sont cruciales car elles contraignent l’information incluse dans les covariances. Si l’anisotropie est facile à prendre en compte, créer des noyaux non-stationnaires est autrement plus difficile. C’est pourtant le cas des phénomènes météorologiques : les nuages suivent un cycle diurne et leur dynamique dépend donc de l’heure de la journée. De plus, les variations atmosphériques ne sont pas uniformes dans le nuage : à la base et à l’intérieur peu de turbulences, avec des turbulences marquées sur le haut et sur les côtés. De la même manière, une bulle d’air chaud est forcément ascendante : elle se déplace donc selon une direction privilégiée dans l’espace.

Arriver à une formulation locale permettant de prendre en compte ces caractéristiques serait donc intéressant pour la qualité du modèle, en particulier pour arriver à de meilleures prédictions dans le futur. En effet, la dimension temporelle a cela de particulier que les mesures peuplent une seule direction (le passé), alors que les prédictions concernent uniquement l’autre direction (le futur), où aucun exemple n’est disponible. Intégrer un *a priori* sur le sens de variation spatiale des phénomènes atmosphériques par rapport à la dimension temporelle semble le seul moyen de gérer leur dynamisme intrinsèque.

3.4 Liens avec la planification

Après avoir choisi un modèle adéquat, encore faut-il intégrer celui-ci à la planification. Il est possible de se servir uniquement de la prédiction faite en chaque point pour prédire l’influence du vent sur l’UAV, mais il est possible de faire beaucoup mieux. Premièrement, il faut pouvoir se servir à bon escient de l’incertitude de chaque prédiction *via* la variance. Deuxièmement, on peut remarquer que les hyper-paramètres contiennent des informations riches apprises par le modèle sur le processus. Il est ainsi envisageable d’utiliser ces paramètres, et de manière plus générale, la fonction de covariance comme base pour les hyper-paramètres de la planification : horizon temporel pertinent, intervalle de planification cohérent ou encore fréquence d’échantillonnage.

3.4.1 Fonction d’utilité

Pour décider de la trajectoire la plus pertinente, il faut pouvoir calculer l’utilité attendue d’emprunter celle-ci. Dans le contexte de SkyScanner, l’opti-

misation est multicritère : gain d'informations de différentes natures et gain (ou perte) d'énergie. Par exemple, on peut chercher à cartographier les vents dans une zone particulière du nuage, et en même temps utiliser cette carte pour tirer parti des courants ascendants et éviter les courants descendants. Pour ce qui est du gain d'information, on peut estimer l'utilité d'une trajectoire par rapport à un processus $y = f(\mathbf{x})$ modélisé par un processus de régression gaussien à partir de l'expression de la variance de ce dernier. En effet, comme on peut le constater dans l'équation 3.3, l'expression de la variance prédite $\mathbb{V}[y_\star]$ en un point test \mathbf{x}_\star ne dépend pas de la valeur du processus aux points \mathbf{X} servant d'exemples pour le modèle. Dans notre exemple, les différents y correspondent alors à la force du vent dans chaque direction au point de coordonnées x .

Première possibilité, utilisée dans [CLS15] : simuler l'échantillonnage de points sur la trajectoire et mettre à jour la matrice de covariance avec les nouveaux exemples. L'incertitude du modèle est alors définie comme l'intégrale de la variance sur le domaine d'intérêt. Pour un processus de dimensions d (dimensions spatiales et éventuellement temporelle) dont on s'intéresse à l'hypercube $\{x|\forall i, x_i \in [x_{m,i}, x_{M,i}]\}$, l'incertitude totale sur le domaine s'exprime

$$V_t = \int_{x_{m,1}}^{x_{M,1}} \dots \int_{x_{m,d}}^{x_{M,d}} \mathbb{V}[f(x_{\star,1} \dots x_{\star,d})] dx_{\star,1} \dots dx_{\star,d} \quad (3.7)$$

L'hypercube sur lequel est calculée l'intégrale correspond à la zone d'intérêt spatio-temporelle à cartographier. On en déduit alors le gain d'information espéré après avoir au temps $t + \Delta t$ exécuté la trajectoire et mis à jour le modèle :

$$I_{gain} = V_t - V_{t+\Delta t} \quad (3.8)$$

En utilisant le noyau exponentiel au carré (équation 3.5), une forme analytique de cette intégrale existe et peut être calculée en $\mathcal{O}((n+m)^2)$, où n est le nombre d'exemples au temps t et m les exemples échantillonnés le long de la trajectoire considérée. En l'absence de formule analytique, l'intégrale peut être approximée par échantillonnage.

Une autre solution couramment employée pour comparer la divergence de deux distributions de probabilité est d'employer la divergence de Kullback-Leibler. L'entropie relative de deux distributions gaussiennes $\mathcal{N}(\mu_0, \Sigma_0)$ et $\mathcal{N}(\mu_1, \Sigma_1)$ s'obtient par la formule :

$$KL(\mathcal{N}_0||\mathcal{N}_1) = \frac{1}{2} \log |\Sigma_1 \Sigma_0^{-1}| + \frac{1}{2} tr \Sigma_1^{-1} ((\mu_0 - \mu_1)(\mu_0 - \mu_1)^\top + \Sigma_0 - \Sigma_1) \quad (3.9)$$

On observe qu'il est nécessaire que les deux matrices de covariance soient de même dimension, ce qui n'est pas forcément notre cas : la matrice du modèle a priori Σ_0 est de dimension $n \times n$, après intégration des nouveaux exemples, elle est de manière générale de taille $(n+m) \times (n+m)$. De plus, la dimension de ces matrices peut être relativement importante ce qui impacte lourdement le calcul du déterminant.

On propose donc tout d'abord, non pas comme dans le premier cas de considérer la distribution *jointe* des exemples \mathbf{X} et des nouveaux échantillons \mathbf{X}_{new} , mais de considérer la distribution *conditionnelle* de l'un par rapport à l'autre.

Ainsi à partir des notations de la section précédente :

$$\Sigma_{\mathbf{X}_{new}|\mathbf{X}} = K(\mathbf{X}_{new}, \mathbf{X}_{new}) - K(\mathbf{X}_{new}, \mathbf{X})\Sigma_{\mathbf{X}}^{-1}K(\mathbf{X}_{new}, \mathbf{X})^{\top} \quad (3.10)$$

Le tout s'effectue en $\mathcal{O}(nm^2 + mn^2)$ opérations. On résout ainsi les deux problèmes précédents : les deux matrices sont de même taille, fixe et relativement petite.

On peut alors, à la place de la KL-divergence, calculer directement l'entropie différentielle de Shannon, qui mesure la quantité d'incertitude dans la distribution (ou encore la quantité d'information qu'elle contient) :

$$H[\mathcal{N}(\mu, \Sigma)] = \frac{1}{2} \log |\Sigma| + \frac{m}{2} (\log 2\pi e) \quad (3.11)$$

Pour une loi normale en dimension m , le coût du calcul de H provient essentiellement du calcul du déterminant de Σ , en $\mathcal{O}(m^3)$. Calculer l'entropie de la distribution $\mathbf{X}_{new}|\mathbf{X}$ s'effectue donc en $\mathcal{O}(m^3 + nm^2 + mn^2) \approx \mathcal{O}(n^2)$, car $m \ll n$ et est constant.

La trajectoire qui est la plus informative est alors celle qui maximise $H(\mathbf{X}_{new}|\mathbf{X})$. L'alternative en utilisant la KL-divergence est de marginaliser les nouveaux exemples (calculer $\mathbf{X}|\mathbf{X}_{new}$) pour déterminer l'influence de ces nouvelles informations sur l'ancien modèle. On travaille alors sur des matrices de taille $n \times n$, ce qui est beaucoup plus coûteux. Bien que le calcul de l'entropie présente certains désavantages (par exemple la valeur obtenue peut être négative), ce n'est pas très important dans notre cas : on cherche à comparer les trajectoires les unes par rapport aux autres et non à calculer une quantité absolue d'information.

En définitive, juger de l'utilité informative d'une trajectoire peut être vu comme un problème de planification optimale (bayésienne) d'expériences⁷. En planification optimale, on cherche à déterminer les prochains points de mesures à réaliser en fonction du modèle de régression statistique courant. Le but est de déterminer ceux qui apporteront le plus d'information au modèle, ce qui, dans le cas de l'estimation par moindres carrés, revient à minimiser la variance à posteriori.

Dans ce type de modèle, cela revient donc à minimiser la matrice de covariance Σ . L'inverse de cette matrice Σ^{-1} est aussi appelée matrice d'information de Fisher : minimiser la variance c'est donc maximiser la matrice d'information Σ^{-1} . Cette notion devient évidente lorsqu'on observe de plus près l'équation 3.3. Comme Σ est positive-définie, Σ^{-1} l'est aussi et donc par définition $\forall x \in \mathbb{R}^n, x^{\top}\Sigma^{-1}x > 0$. Donc pour un point de mesure particulier x_{\star} , la variance peut s'écrire $\mathbb{V}[y_{\star}] = k(x_{\star}, x_{\star}) - f(x_{\star})$ avec $f, k > 0$. Maximiser Σ^{-1} permet donc de minimiser la variance. Cependant, maximiser Σ^{-1} est plus un concept qu'une opération bien définie. On peut définir de nombreux critères d'optimisation à partir de celui-ci :

- *E-optimalité* : maximiser la valeur propre minimale Σ^{-1} (et donc minimiser la variance maximale)
- *D-optimalité* : maximiser le déterminant de Σ^{-1} , ce qui est équivalent à maximiser l'entropie différentielle de Shannon

7. (bayesian) optimal design

— *T-optimality* : maximiser la trace de Σ^{-1}

Il est aussi possible de s'intéresser directement à la variance à posteriori : c'est le critère d'*I-optimality*, où l'on minimise l'intégrale de la variance dans le domaine d'intérêt.

Il convient de remarquer que dans le cadre de l'optimisation en temps réel de la trajectoire, le temps est limité : plus on pourra évaluer de candidates, meilleure sera la qualité de la trajectoire finale. Améliorer la rapidité de calcul du modèle atmosphérique, comme suggéré précédemment, bénéficiera donc aussi à la planification.

Nous avons implémenté trois critères de décision : *D-* et *T-optimality* à partir de l'équation 3.10 et *I-optimality* (pour le noyau exponentiel carré à partir de la formulation analytique présentée dans [CLS15]). L'implémentation a été réalisée en Python en utilisant les libraires de calcul numérique *numpy*⁸ et *scipy*⁹, et dans certains cas directement la librairie LAPACK¹⁰. La mise à jour de l'inverse de la matrice de covariance a été implémentée en utilisant le lemme d'inversion de matrice pour plus d'efficacité (mise à jour en $\mathcal{O}(n^2)$). Aucune autre optimisation particulière n'a été implémentée.

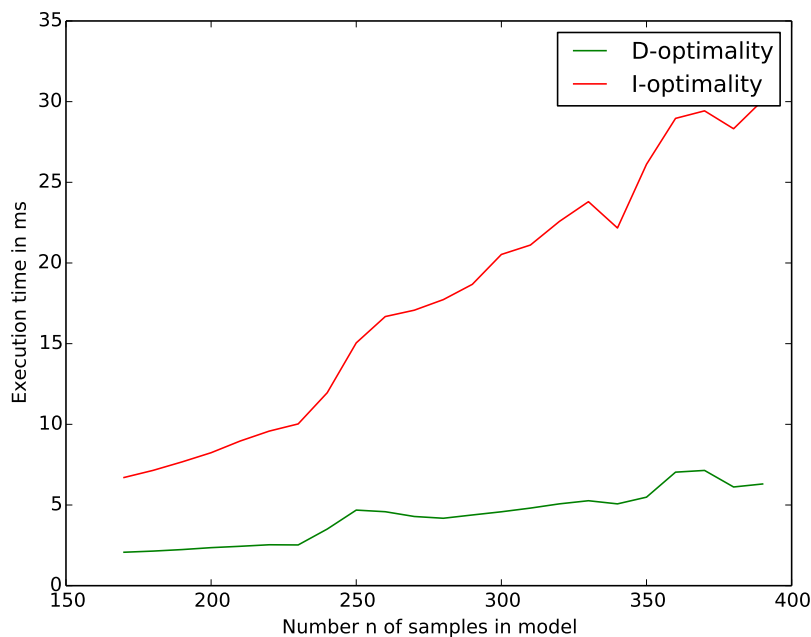


FIGURE 3.4 – Temps moyen d'exécution en ms du calcul des critères d'optimalité D et I pour $m=10$ nouveaux points de mesure et une taille de modèle variable.

8. <http://www.numpy.org/>

9. <http://www.scipy.org/>

10. <http://www.netlib.org/lapack/>

La figure 3.4 compile le temps d'exécution moyen pour les critères D et I (T non montré car le temps d'exécution est quasi-identique à celui du critère D). On observe que le choix du critère a un impact non négligeable sur le temps d'optimisation, malgré des complexités équivalentes. Même si une meilleure optimisation du code pourrait réduire cette différence, il est peu probable qu'elle soit complètement effacée : un gain de performance bien plus important peut être attendu de l'utilisation de modèles "creux".

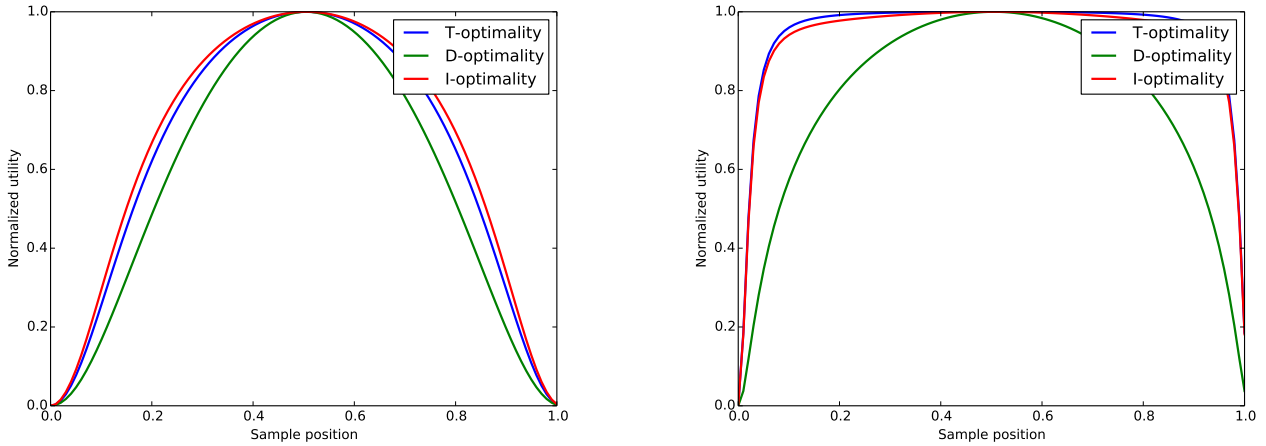


FIGURE 3.5 – Utilité normalisée d'un nouvel exemple dans un univers 1D, selon les critères T, D et I. Le modèle est préalablement peuplé d'exemples espacés entre eux d'une unité. Le noyau k utilisé est le noyau exponentiel au carré, de longueur caractéristique $l = 0.5$ ($\sigma_f = 1$). Sont présentés deux niveaux de bruits différents, à gauche $\sigma_n = 0.1$, à droite $\sigma_n = 0.001$.

Il nous a également semblé intéressant de pouvoir visualiser la différence de valuation des solutions par les différents critères. Dans la figure 3.5 on se place dans un univers 1D, où le modèle est préalablement peuplé d'exemples espacés entre eux d'une unité. On montre la valeur du critère d'optimalité sur l'ensemble des positions d'un seul nouvel exemple situé dans l'intervalle $[0, 1]$, en utilisant pour le noyau exponentiel carré de longueur caractéristique 0.5 et deux niveaux de bruit différents. Les valeurs ont été normalisées par souci de comparabilité. Le comportement global des trois critères est équivalent en terme de classement des alternatives : il n'est pas surprenant de constater que la position la plus utile est à mi-chemin entre les exemples du modèle, et que la moins utile est de reprendre un échantillon au même point. Plus étonnant, les comportements de T et I sont très proches dans cette situation, quel que soit le niveau de bruit. La D-optimalité se distingue par contre par une sensibilité au bruit beaucoup moins importante. En pratique, nous avons par la suite utilisé arbitrairement le critère D (plutôt que le critère T).

3.4.2 Intégration étroite entre modèle et plan

La planification de trajectoires définie au chapitre précédent suppose un certain nombre de paramètres : l'horizon temporel de la planification Δt , le pas de

temps dt entre deux positions atomiques pour la planification, et la résolution spatiale de la grille dx sur laquelle la planification est effectuée. Enfin, on peut aussi considérer en paramètre la fréquence d'échantillonnage (c'est à dire le pas de temps entre deux prises de mesures), qui n'influe pas directement sur la planification mais pose des contraintes pratiques, en particulier de communication.

Il est tout d'abord évident que dt et dx sont liés : il convient de choisir ces grandeurs de telle sorte que pendant le laps de temps dt la variation de position de l'UAV soit comprise en moyenne entre un et deux dx . Cela découle de la variation du vent entre deux positions successives : si l'UAV ne change pas de cap et à commandes constantes, seul le vent peut être responsable d'une déviation de trajectoire de l'UAV. Il faut donc qu'entre deux positions de la trajectoire, c'est à dire entre deux commandes, le vent *varie suffisamment faiblement pour pouvoir être considéré constant*. C'est implicitement ce qui est fait en fixant pour la planification le pas dx où le vent est considéré constant. En fixant un seuil de tolérance sur la variation du vent, on peut donc chercher à fixer les valeurs maximales de dt et dx permettant de respecter cette propriété dans la plupart des cas.

Le choix de la valeur de Δ_t est également important pour la planification : plus Δ_t est faible, moins cher la simulation coûtera et plus la myopie de l'UAV sera prononcée. Il faut donc trouver un bon compromis entre temps de calcul et planification à plus long terme. Cependant, la qualité de la trajectoire dépend de celle du modèle : si la prédiction locale est très incertaine, augmenter Δ_t ne bénéficiera pas forcément à la planification car la trajectoire réalisée risque de dévier fortement de celle prédite. Faire dépendre Δ_t de la qualité du modèle permettrait donc d'éviter des calculs inutiles et de rendre l'agent plus réactif lorsque la qualité de la prédiction se dégrade.

De la même manière, pour fixer la fréquence d'échantillonnage, on peut spécifier l'incertitude maximale sur le chemin de l'UAV de la prédiction à posteriori du vent. Pour simplifier, on considère l'UAV sur un morceau de trajectoire rectiligne et infinie. En fixant la fréquence d'échantillonnage, on peut alors estimer, à partir de l'expression de la variance 3.3, la variance maximale sur son chemin : c'est la variance du point situé à mi-chemin entre deux mesures. On se place sur un chemin de longueur infinie, il donc faut prendre soin d'inclure un nombre suffisant d'exemples dans le noyau pour que la valeur approchée soit cohérente, c'est à dire inclure tous les points contribuant perceptiblement à la prédiction. On peut alors pour un noyau et un niveau de bruit connu, tracer la courbe d'évolution de la variance maximale. Celle-ci est strictement monotone, il est donc trivial de calculer pour un seuil de tolérance particulier la fréquence d'échantillonnage minimale requise, comme le montre la figure 3.6.

Le bénéfice de laisser le modèle fixer les hyper-paramètres de la planification est double. D'une part, on élimine la nécessité d'experts pour les déterminer, ce qui bénéficie à la robustesse de la solution. D'autre part, cela permet de commencer à prendre en compte l'incertitude de manière indirecte. En effet, la planification utilisée par ailleurs ne se soucie à aucun moment de l'effet sur la trajectoire planifiée de l'incertitude des prédictions fournies. Si, pour l'instant, nos réflexions sur ce sujet sont préliminaires, nous sommes convaincus du potentiel bénéfique à continuer à étayer celles-ci.

Les deux chapitres précédents ont, après avoir établi un bref résumé de l'état

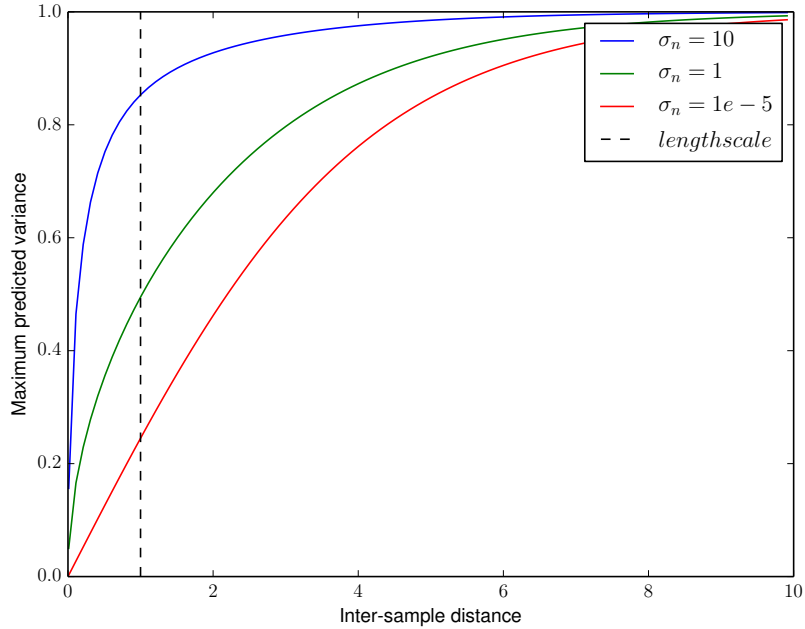


FIGURE 3.6 – Variance prédite maximum le long d’une trajectoire rectiligne dans la direction de plus petite longueur caractéristique, en fonction de la distance inter-échantillons (et donc indirectement de la fréquence d’échantillonnage), pour le noyau exponentiel au carré avec $l = 1$ et $\sigma_f = 1$. La courbe est tracée pour trois niveaux de variance σ_n du bruit.

de l’art, exposé les outils de planification et modèles que nous avons développés pour tenter de résoudre le problème posé. Le chapitre suivant décrit les résultats expérimentaux préliminaires issus de la confrontation de nos outils avec des simulations numériques météorologiques.

Chapitre 4

Résultats expérimentaux

Ce chapitre décrit nos efforts d'implémentation et de tests préliminaires des outils présentés en détail dans les chapitres précédents, en commençant par la présentation des simulations météorologiques dont nous avons bénéficié. Il continue par les résultats de l'évaluation des modèles statistiques sur ces simulations. Puis, après avoir introduit un scénario de mission et une fonction d'utilité simple, nous détaillerons les premiers résultats de planification prenant en compte la prédiction des courants de vents par le modèle de régression par processus gaussiens. Enfin, le chapitre se termine par une ouverture sur les travaux pour intégrer les solutions développées au simulateur de vol réaliste servant de plate-forme d'intégration pour les travaux des différentes équipes du projet SkyScanner.

4.1 Validation sur simulations réalistes

Dans le cadre du projet, l'équipe du CNRM-Game a produit une simulation de microphysique des nuages dans un scénario de production de cumulus de beau temps, grâce au simulateur Meso-NH développé par le CNRM. Une des contributions du projet sera de valider l'ensemble des composants développés dans une simulation réaliste, tant au point de vue de la dynamique du drone qu'au point de vue atmosphérique. Le but, d'ici la fin du projet, est de faire voler les drones en simulation dans un environnement simulé.

La simulation produite concerne un cube de 4×4 km de côté et 3 km de hauteur. La résolution horizontale est de 10 m, pour une résolution verticale variable entre 5 et 100 m suivant la hauteur. Pour les besoins de la simulation, les bords opposés horizontaux sont considérés voisins, c'est à dire que sortir par le bord nord de la simulation revient à y entrer par le bord sud, et inversement. Un certain nombre de variables d'intérêt microphysique sont simulées, les principales étant :

- Le vent tridimensionnel Cx, Cy, Cz
- La température, l'humidité et la pression
- La quantité de gouttelettes d'eau en suspension

Le pas de temps des sorties de simulation est 1s, c'est à dire qu'on dispose pour chaque point de l'espace d'une valeur pour chaque variable toutes les secondes. Chaque sortie de simulation pesant environ 700 Mo, une minute de simulation

occupe 42 Go, et une heure environ 2.5 To. Pour simuler la formation de cumulus, il est nécessaire de simuler une journée entière soit environ 16 h, chaque heure de simulation prenant environ trois jours de calcul sur le cluster de Météo France. Au vu de la taille des sorties de simulation, seule une sortie toutes les heures a été effectuée, puis une heure d'intérêt a été sélectionnée puis re-simulée avec une sortie toutes les minutes, puis toutes les secondes. A l'heure de la rédaction de ce rapport, nous avons commencé à analyser 30 minutes de résultats à la minute, et 15 minutes de résultats à la seconde sont d'ores et déjà disponibles. La figure 4.1 montre une visualisation de simulation Meso-NH.

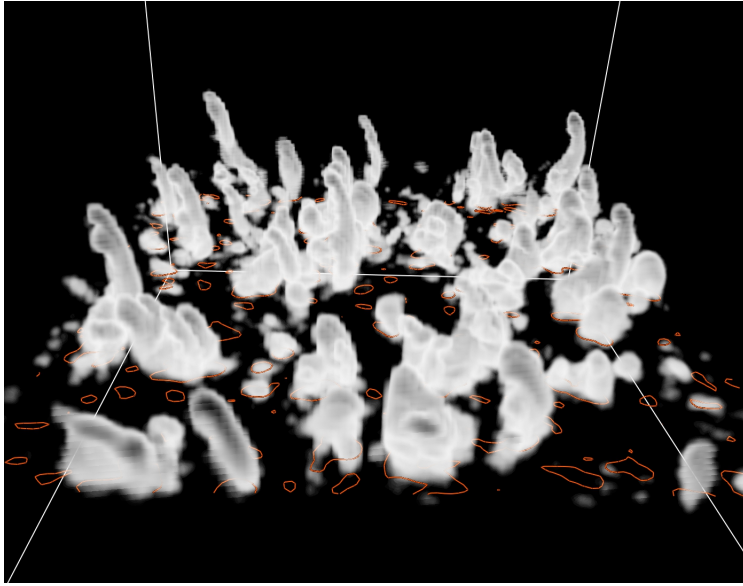


FIGURE 4.1 – Visualisation d'une simulation Meso-NH (axe vertical étiré). Les nuages sont identifiés par la présence d'eau liquide (concentration en nuances de gris), et les courbes isométriques de vent vertical $0.5 \text{ m}\cdot\text{s}^{-1}$ sur le plan de hauteur $z = 1 \text{ km}$ ont été tracées en orange.

4.1.1 Validation statistique du modèle de régression par processus Gaussiens

En premier lieu, ces résultats de simulations sont intéressants pour pouvoir tester et valider les modèles statistiques développés. N'ayant disposé des résultats à la seconde qu'à la fin du stage, nous n'avons pu tester pour l'instant la validité du modèle que sur les trois dimensions spatiales. La dimension temporelle, qui par ailleurs présente le plus de difficulté, n'a donc pas encore pu être abordée.

Pour tester le modèle atmosphérique, nous avons donc sélectionné quelques instantanés du cube simulé, puis un sous-cube d'intérêt dans les instantanés. Un nombre n_{seed} d'exemples sont choisis à partir des plus proches voisins d'un exemple (graine) aléatoire. Ces exemples sont utilisés pour optimiser les hyperparamètres du modèle. Ensuite, un sous-échantillonnage du sous-cube d'intérêt est réalisé avec n_{train} exemples pour entraîner le modèle. Il est alors évalué sur

l'ensemble du sous-cube. En pratique, nous avons fixé $n_{seed} = 300$ dans nos tests pour des raisons de temps d'exécution, alors que n_{train} est de l'ordre de quelques milliers d'exemples.

On définit plusieurs métriques :

$$RMSE(Y_\star) = \sqrt{\frac{1}{|Y|} \sum_{Y_\star} (\bar{Y}_\star - Y_\star)^2} \quad (4.1)$$

$$MVAR(Y_\star) = \mathbb{E}[\mathbb{V}[Y_\star]] \quad (4.2)$$

respectivement l'erreur quadratique moyenne de prédiction et la variance prédictive moyenne. Il peut également être utile de vérifier si le modèle prédit correctement son erreur. On définit la quantité supplémentaire :

$$\sigma_s(y_\star) = \frac{(\bar{y}_\star - y_\star)}{\sqrt{\mathbb{V}[y_\star]}} \quad (4.3)$$

le rapport entre l'erreur et la déviation standard prédite de l'erreur. C'est donc le nombre de σ signé de l'événement consistant à mesurer y_\star sous le modèle de la loi normale $\mathcal{N}(\bar{y}_\star, \mathbb{V}[y_\star])$. Par exemple si $\sigma_s(y_\star) = 3$, l'événement y_\star est à 3σ (une chance sur trois cent soixante-dix). La distribution de σ_s sur Y_\star permet de vérifier la qualité de l'estimation de $\mathbb{V}[y_\star]$ par rapport à l'erreur.

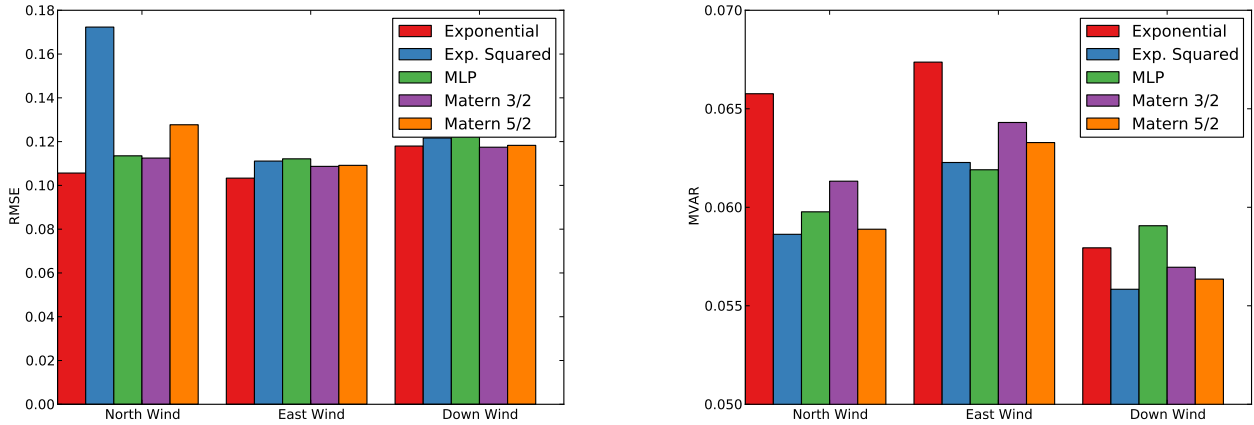


FIGURE 4.2 – Valeur de RMSE et MVAR pour six noyaux différents et pour les trois composantes du vent. Pour chaque noyau, dix itérations d'optimisation des hyper- paramètres avec des exemples différents ont été effectuées et les résultats de celles ayant produit la plus basse RMSE ont été retenus.

Les résultats présentés dans les figures 4.2 à 4.4 sont typiques des résultats obtenus. Dans l'ensemble, la valeur absolue de l'erreur prédictive dépend de la densité des exemples d'entraînement. Cependant, les résultats sont relativement homogènes suivant les différents noyaux et les différentes composantes du vent, tant pour l'erreur (RMSE) que pour la variance prédite. Tous les noyaux testés sont de type "base radiale", et on retrouve donc ici l'idée que la formulation du noyau compte moins que les informations qu'il encode (ici la valeur des

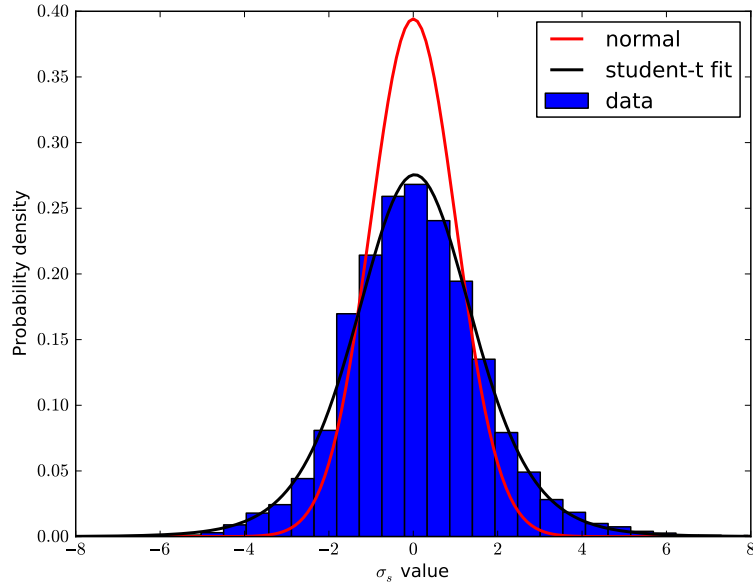


FIGURE 4.3 – Distribution de σ_s pour le noyau exponentiel carré et la composante "bas" du vent. Dix itérations d'optimisation des hyper-paramètres avec des exemples différents ont été effectuées et les résultats de celles ayant produit la plus basse RMSE ont été retenus. En rouge : densité de probabilité $\mathcal{N}(0, 1)$. En noir : densité de probabilité de la loi t de Student de moyenne 0, de variance 1.4 et avec $df = 7.6$.

hyper- paramètres). La distribution de σ_s (figure 4.3) montre que l'estimation de l'erreur est relativement bonne, sans biais mais légèrement sous-estimée. Cela se traduit par une déviation standard légèrement supérieure à 1. La distribution ne suit pas une loi normale mais plutôt de Student, car chaque prédiction n'est pas issue de la même distribution de probabilité (de plus les distributions ne sont pas indépendantes). Cela se traduit par des événements (erreurs de prédiction) largement improbables ($> 6\sigma$, soit $<$ une chance sur cinq cents millions) à priori selon les prédictions.

4.1.2 Qualité des observations et variabilité des hyper-paramètres

Si les résultats précédents sont encourageants quant à la capacité de reconstruction des GP pour les phénomènes atmosphériques que l'on cherche à étudier, il n'en reste pas moins que les UAVs ne peuvent échantillonner aléatoirement l'environnement : leur trajectoire est contrainte par leur dynamique, et donc la qualité du modèle va dépendre des observations faites. Il est donc nécessaire de vérifier la capacité de reconstruction par les observations contraintes du drone. De plus, il n'est pas possible de connaître à priori les hyper-paramètres du modèle : chaque nuage évolue de manière différente, et le processus n'est pas

stationnaire. Les hyper-paramètres diffèrent donc potentiellement d'une mission à l'autre, et pendant chaque mission.

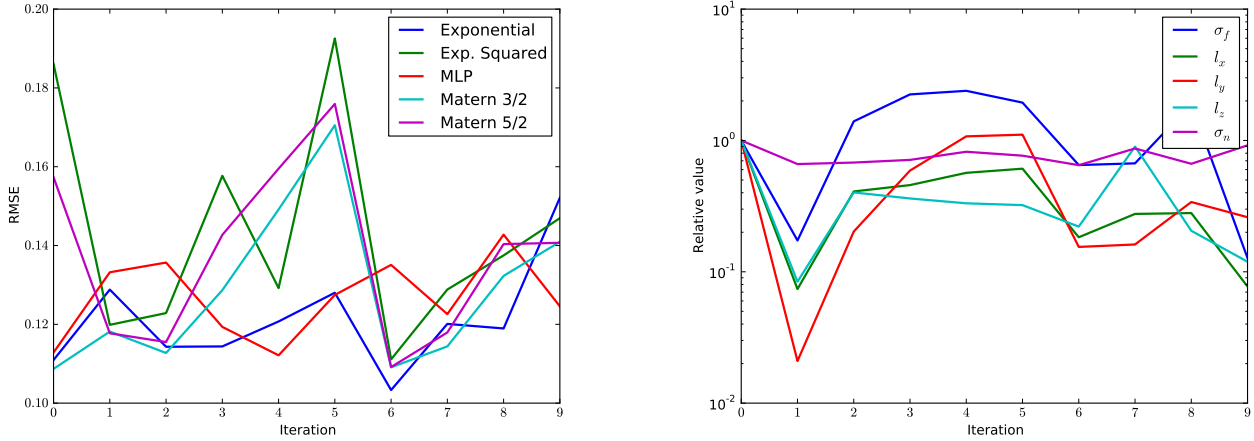


FIGURE 4.4 – Résultats avec le noyau exponentiel pour dix optimisations des hyper-paramètres avec des exemples différents. A gauche : RMSE pour chaque itération. A droite : valeur des hyper-paramètres à chaque itération, relativement à la valeur de la première (axe y logarithmique).

Pour illustrer l'importance de la variation des hyper-paramètres, la figure 4.4 montre la variation des hyper-paramètres et de l'erreur (RMSE) sur dix itérations d'optimisation, chaque itération utilisant un nouvel ensemble d'exemples. Non seulement on observe une importante variation de la valeur des hyper-paramètres qui peuvent changer de plusieurs ordres de grandeurs avec des exemples différents, mais de plus, on observe entre les noyaux une certaine corrélation : certains ensembles d'exemples sont défavorables à certains ensembles de noyaux, comme on peut clairement le voir entre les itérations quatre et six. Cette variabilité des hyper-paramètres conduit à des modèles largement différents d'une itération à l'autre, et est principalement liée à la variation de l'ensemble d'exemples. Ainsi implémenter des solutions permettant d'incorporer plus d'exemples dans le modèle pourra éventuellement permettre une meilleure stabilité de l'optimisation en minimisant la différence entre les jeux d'exemples à chaque itération.

4.2 Intégration du modèle atmosphérique dans la planification de trajectoires

Afin d'effectuer les premières expérimentations d'intégration du modèle atmosphérique dans la planification, nous avons développé un simulateur très simple en Python, en reprenant le code développé pour tester les processus de régression gaussiens et pour lire les résultats de simulation Meso-NH. Un pas de simulation se déroule ainsi : une carte locale des vents, telle que prédite par

le modèle atmosphérique, est fournie au planificateur avec une fonction d'utilité $U : R^{n \times m} \rightarrow R^n$, où n est le nombre de commandes par trajectoire et m le nombre de drones. Chaque trajectoire générée par le planificateur est évaluée par la fonction U et la meilleure est sélectionnée. Celle-ci est alors exécutée en boucle ouverte, en utilisant le même modèle que la planification mais en se basant sur la vraie valeur du vent, lue dans la simulation Meso-NH. Un échantillonnage est effectué sur la trajectoire : à pas de temps réguliers, une mesure est prise en lisant la valeur du vent et en appliquant un bruit gaussien, puis la mesure est transmise au modèle. Bien que nous ayons discuté en détail de l'évaluation de l'information prise sur une trajectoire, il reste à bâtir une fonction d'utilité multicritère prenant en considération l'énergie gagnée et perdue sur la trajectoire.

4.2.1 Fonction d'utilité

Pour ce faire, nous avons décidé dans un premier temps d'utiliser une fonction d'utilité très simple. Nous avons défini un type de mission mettant en jeu trois critères à optimiser :

- U_E : rester le plus proche possible d'une altitude d'intérêt
- U_{IG} : utilité des informations acquises pour la carte des vents
- U_B : rester dans la zone d'intérêt

La valeur de ces trois critères est normalisée, c'est à dire $0 \leq U_E, U_{IG}, U_B \leq 1$.

Pour l'utilité énergétique U_E , on définit une altitude optimale z_{opt} . Soit dz_i la variation d'altitude entre le début et la fin de la trajectoire et z_i l'altitude initiale pour l'UAV i . On définit également la quantité dz_{max} qui est la variation d'altitude maximum attendue au cours d'une trajectoire. L'utilité énergétique est alors définie par les équations :

$$U_E(dz_i) = \left(1 - \frac{|\text{sign}(z_{opt} - z_i)dz_{max} - dz|}{dz_{max} + |dz|}\right) \left(1 - \frac{1}{1 + \frac{(z_{opt} - z_i)^2}{l_E}}\right) \quad (4.4)$$

$$U_E(dz) = \frac{1}{m} \sum_{i \in [1..m]} U_E(dz_i) \quad (4.5)$$

$$(4.6)$$

La première partie valorise une variation d'altitude dans le sens de l'altitude cible, et la seconde partie atténue la récompense plus on s'approche de celle-ci.

Pour l'utilité de récolte d'informations U_{IG} , nous avons sélectionné l'un des critères décrit en section 3.4.1. Pour normaliser la valeur v produite par le critère d'optimisation, on utilise la valeur optimale théorique v_{opt} d'une trajectoire :

- pour les critères basés sur la matrice de covariance, on utilise la matrice diagonale $\Sigma_{opt} = (k(x_*, x_*) + \sigma_n^2)I$ comme limite supérieure .
- pour le critère d'intégration, on utilise la même matrice de covariance et un modèle précédent vide (sans exemples).

En effet, la variance d'une mesure indique la quantité d'information qu'elle contient, tandis que les valeurs de covariance (en dehors de la diagonale) indiquent les redondances des mesures dues aux corrélations entre elles. Dans un ensemble théorique de mesures maximisant l'information, celles-ci sont donc non corrélées, et donc la matrice de covariance est diagonale. En supposant la

fonction de covariance stationnaire, la variance de chaque exemple $k(x_*, x_*)$ est alors constante et la valeur d'utilité de la matrice diagonale Σ_{opt} peut-être utilisée comme limite supérieure des fonctions d'utilité se basant sur le modèle de régression.

Soit v la valeur du critère d'optimalité (par exemple D-optimalité) pour l'ensemble des points de mesures sur la trajectoire des UAVs, et v_{opt} la valeur optimale du critère pour le même nombre de mesures, on pose alors :

$$U_{IG}(v) = \frac{1}{1 + \frac{(v_{opt}-v)^2}{l_{IG}}} \quad (4.7)$$

Pour forcer les drones à rester dans la zone d'intérêt, on compte le nombre n_{out} de points des trajectoires à l'extérieur de la zone d'intérêt. On définit alors l'utilité U_B :

$$U_B(n_{out}) = \frac{1}{1 + \frac{n_{out}^2}{l_B}} \quad (4.8)$$

L'utilité finale est alors l'utilité moyenne :

$$U = \frac{1}{3}(U_E + U_{IG} + U_B) \quad (4.9)$$

Quelle que soit la fonction U utilisée, ses propriétés vont fortement influencer la qualité des solutions produites par l'optimisation stochastique. En particulier, celle-ci incluant une étape de suivi de gradient, il est particulièrement important d'utiliser une fonction continue et dérivable en tout point pour que cette étape soit efficace. C'est ce qui justifie l'utilisation d'une combinaison linéaire des différents critères. Les facteurs d'échelles l_E , l_{IG} et l_B ont été définis par essais successifs.

4.2.2 Résultats préliminaires

Pour simplifier la simulation et la visualisation, les résultats préliminaires présentés ici dans les figures 4.5 à 4.11 sont effectués à vent constant verticalement (le vent ne dépend pas de l'altitude). Le champ de vent choisi provient d'une coupe à altitude 1km d'une simulation Meso-NH. Au centre, se situe l'ascendance générée par un cumulus. Deux scénarios sont simulés. Le premier contient un seul agent. Pour chaque itération de planification (10 secondes temps réel), 500 trajectoires sont évaluées. Le second scénario contient trois agents, pour chaque itération 1500 trajectoires jointes sont évaluées. Enfin, le modèle d'environnement est limité à 250 points de mesures pour des raisons pratiques de temps de calcul.

Pour un UAV, la fonction d'utilité choisie permet bien de profiter du courant ascendant : il se circonscrit d'abord à la zone d'ascendance, jusqu'à atteindre l'altitude de consigne puis élargit son périmètre de mesure (fig. 4.5, 4.6 et 4.8). La figure 4.7 permet de se rendre compte de l'état de la carte des ascendances à la fin de la simulation.

Pour trois UAV les résultats sont moins bons, comme on peut le voir sur la figure 4.11 : aucun UAV n'a atteint l'altitude de consigne après 20 itérations,

alors que celle-ci est atteinte après seulement 15 itérations dans le cas mono-agent. Sur la figure 4.9 on peut se rendre compte du caractère plus chaotique des trajectoires : alors qu'aucun drone n'a atteint l'altitude de consigne, ceux-ci ont passé plus de temps dans les zones moins favorables énergétiquement (en bleu). On peut l'expliquer en grande partie par la difficulté du planificateur à explorer l'espace des configurations qui croît exponentiellement avec le nombre de drones.

Un autre effet est celui de la saturation rapide du modèle (fig. 4.10) : celui-ci étant limité en nombre de points de mesures, il oublie des zones préalablement cartographiées. Selon la dynamique du processus, il se peut cependant que le problème soit moins important : avec une dynamique forte, un point trop ancien ne contiendra plus aucune information concernant le temps présent.

Enfin, la définition même de la fonction d'utilité énergétique multi-UAV, optimisant la moyenne de l'utilité énergétique pour chaque UAV, est à prendre en compte et peut influencer négativement les résultats. Une meilleure fonction d'utilité permettrait peut-être dans cette situation d'obtenir de meilleurs résultats sans optimisation de l'algorithme de planification.

Ces premiers résultats sont encourageants, ils devront cependant être validés en prenant en compte les simulations aérologiques non simplifiées dans le temps et dans l'espace.

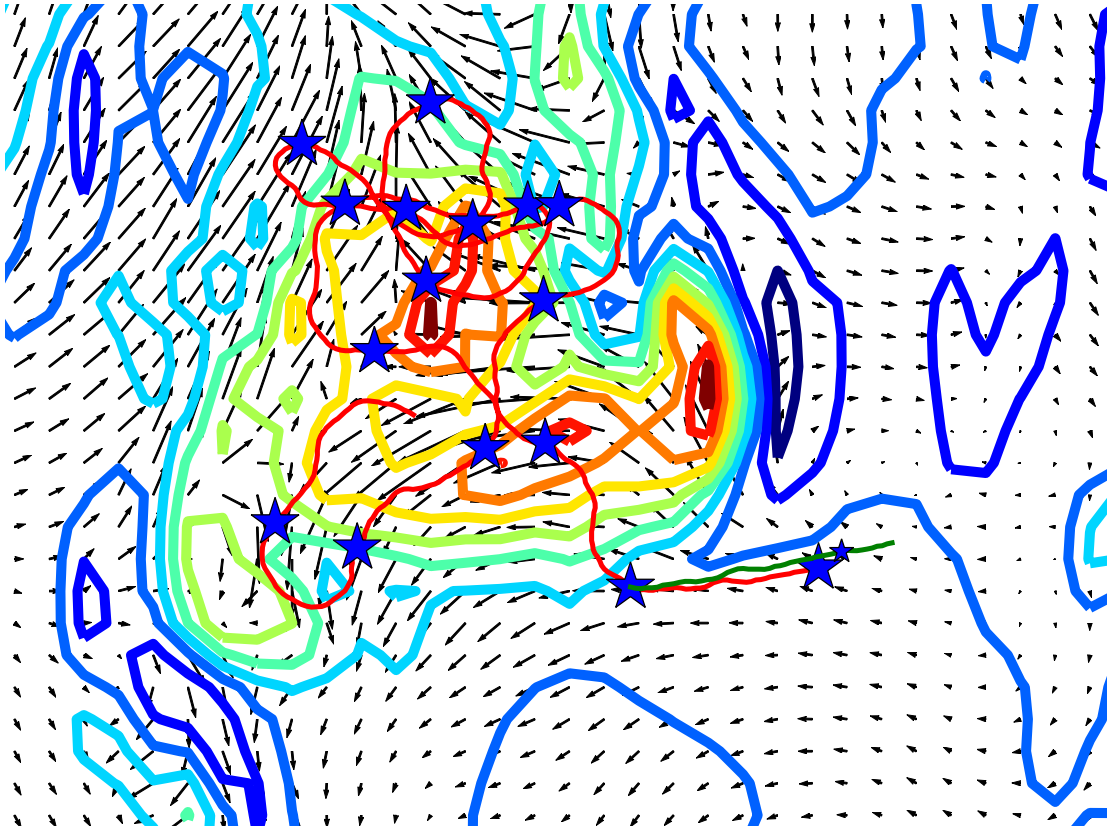


FIGURE 4.5 – Résultats de simulation pour un agent après 15 itérations. En rouge, la trajectoire exécutée; les étoiles bleues représentent les itérations de planification. Les contours colorés permettent de visualiser les courants ascendants (vers le rouge) et descendants (vers le bleu). Les flèches noires indiquent la force et la direction du vent latéral.

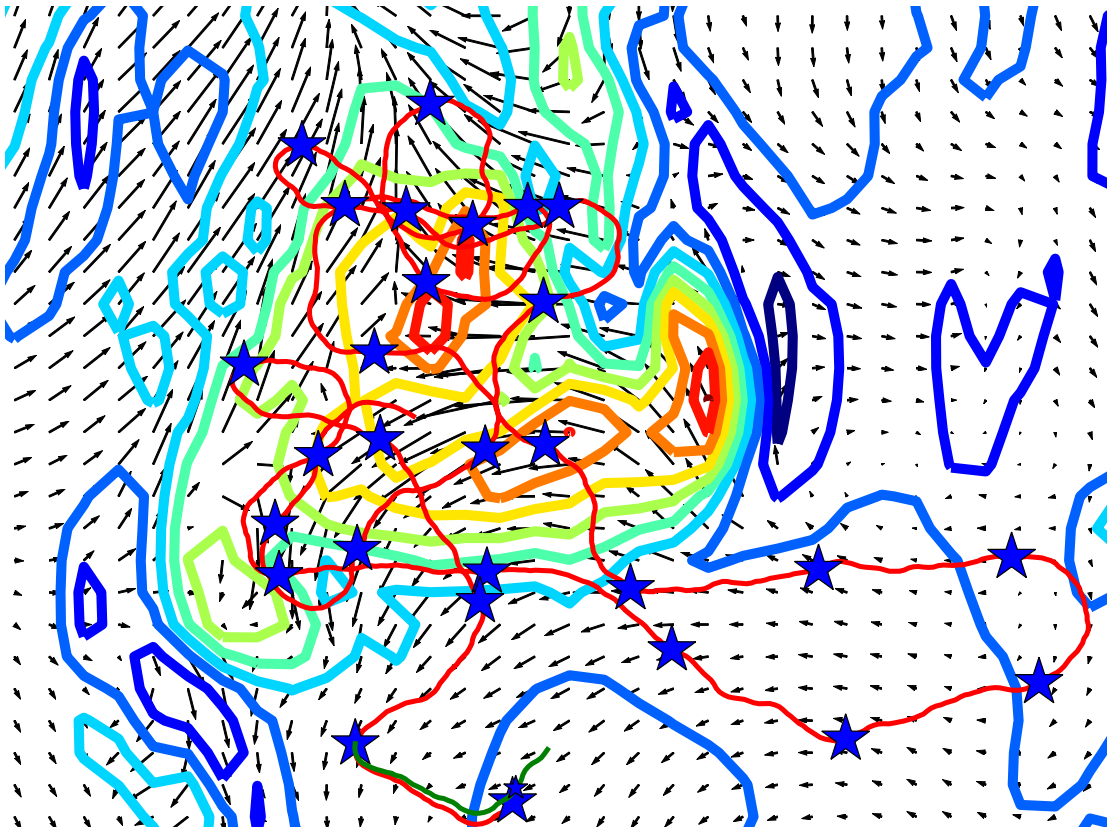


FIGURE 4.6 – Résultats de simulation pour un agent après 27 itérations. En rouge, la trajectoire exécutée; les étoiles bleues représentent les itérations de planification. Les contours colorés permettent de visualiser les courants ascendants (vers le rouge) et descendants (vers le bleu). Les flèches noires indiquent la force et la direction du vent latéral.

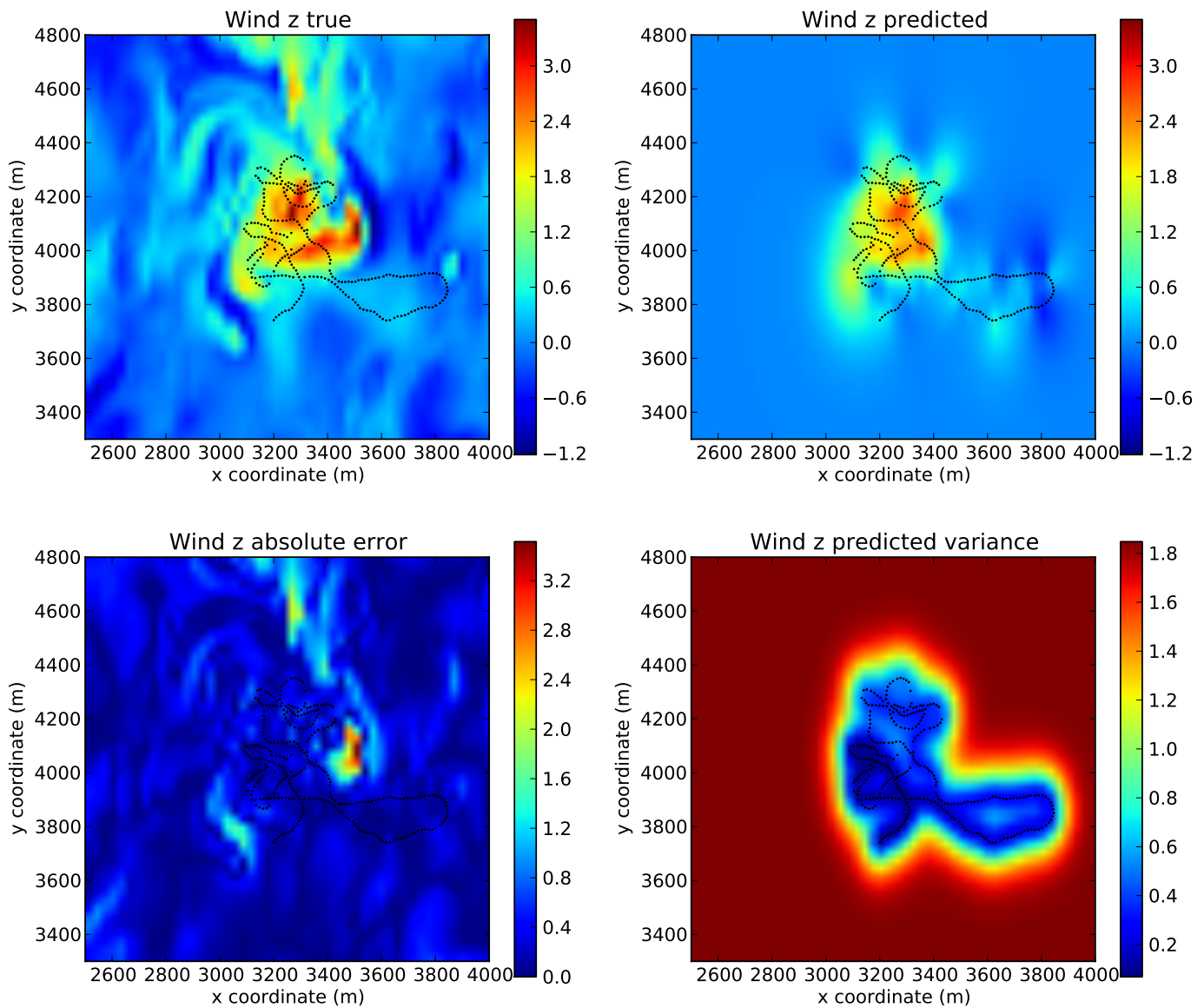


FIGURE 4.7 – Résultats de simulation pour un agent. Carte du vent vertical et modèle après 27 itérations. Chaque point noir correspond à une mesure prise en compte dans le modèle. En haut, à gauche : vérité terrain. En haut, à droite : prédiction du modèle. En bas, à gauche : erreur absolue de la prédiction. En bas, à droite : variance prédite du modèle. Vent en m/s.

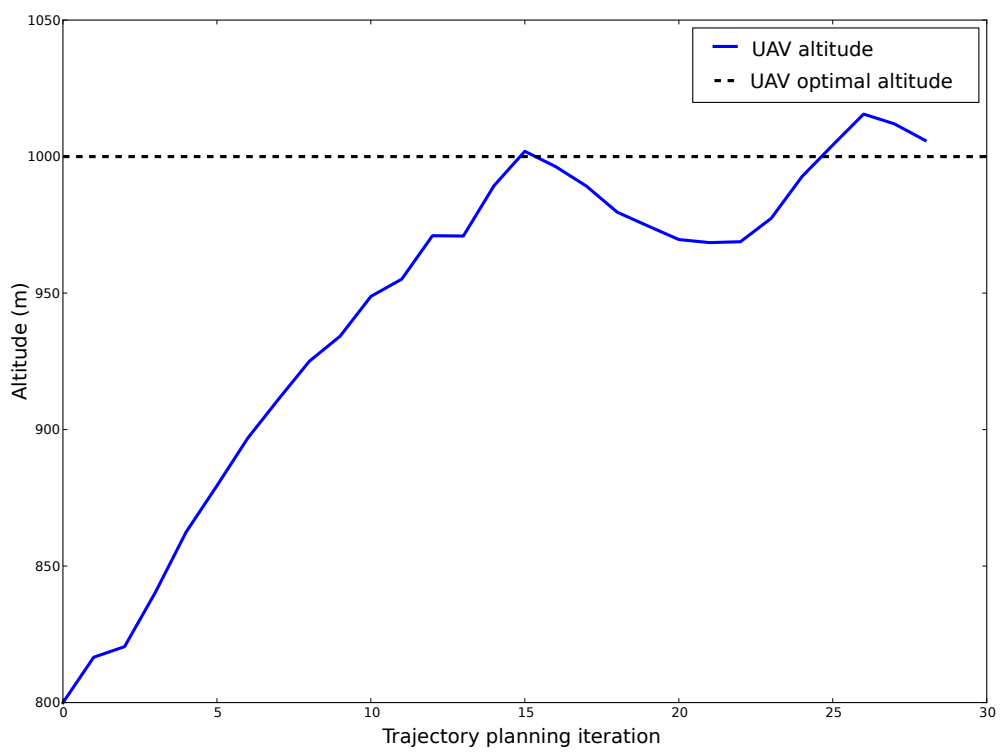


FIGURE 4.8 – Résultats de simulation pour un agent. Altitude de l’agent à la fin de chaque itération. Les pointillés noirs indiquent l’altitude cible de l’UAV. Une itération de planification correspond à 10 secondes temps réel.

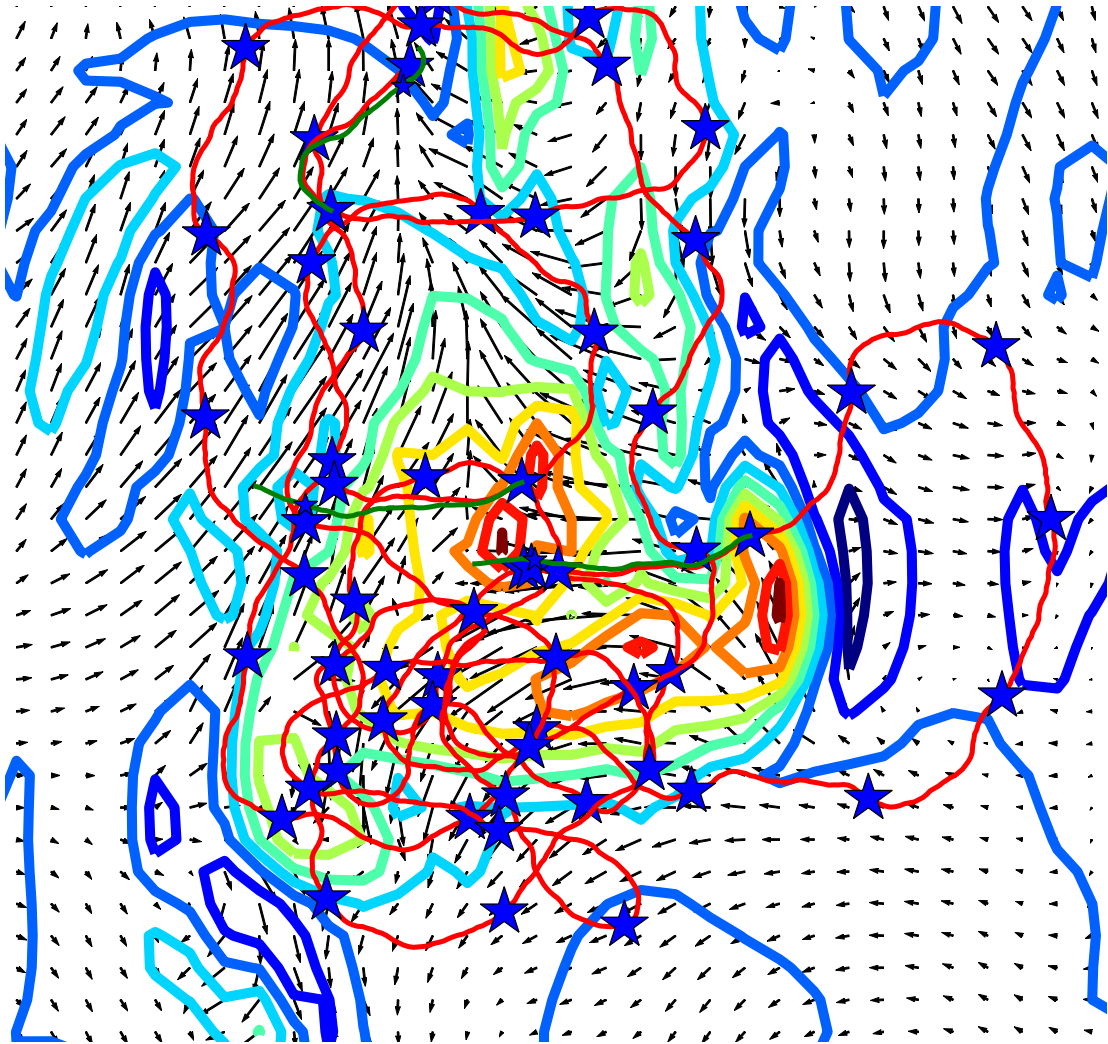


FIGURE 4.9 – Résultats de simulation pour trois agents après 20 itérations. En rouge, la trajectoire exécutée; les étoiles bleues représentent les itérations de planification. Les contours colorés permettent de visualiser les courants ascendants (vers le rouge) et descendants (vers le bleu). Les flèches noires indiquent la force et la direction du vent latéral.

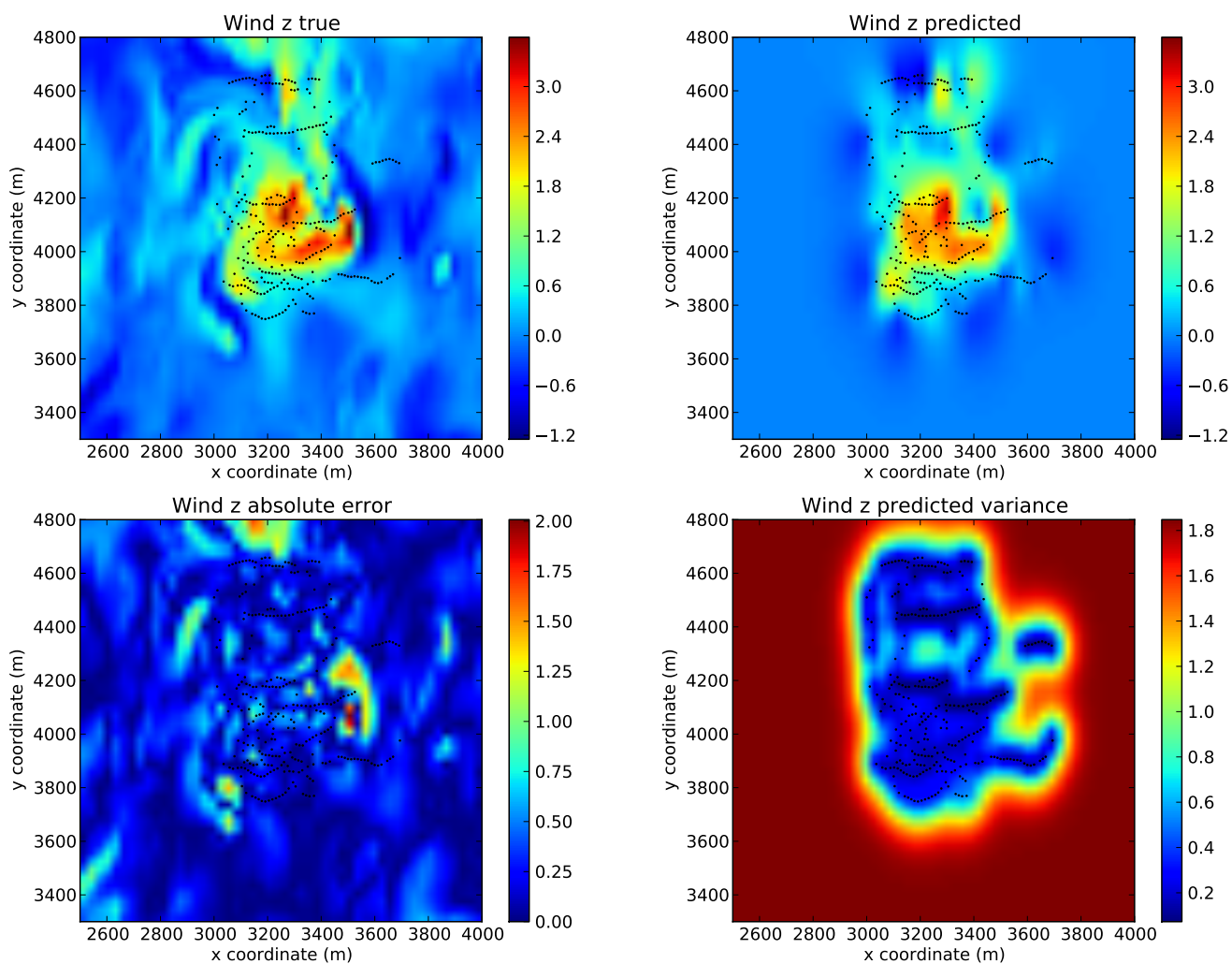


FIGURE 4.10 – Résultats de simulation pour trois agents. Carte du vent vertical et modèle après 27 itérations. Chaque point noir correspond à une mesure prise en compte dans le modèle. En haut, à gauche : vérité terrain. En haut, à droite : prédiction du modèle. En bas, à gauche : erreur absolue de la prédiction. En bas, à droite : variance prédite du modèle. Vent en m/s.

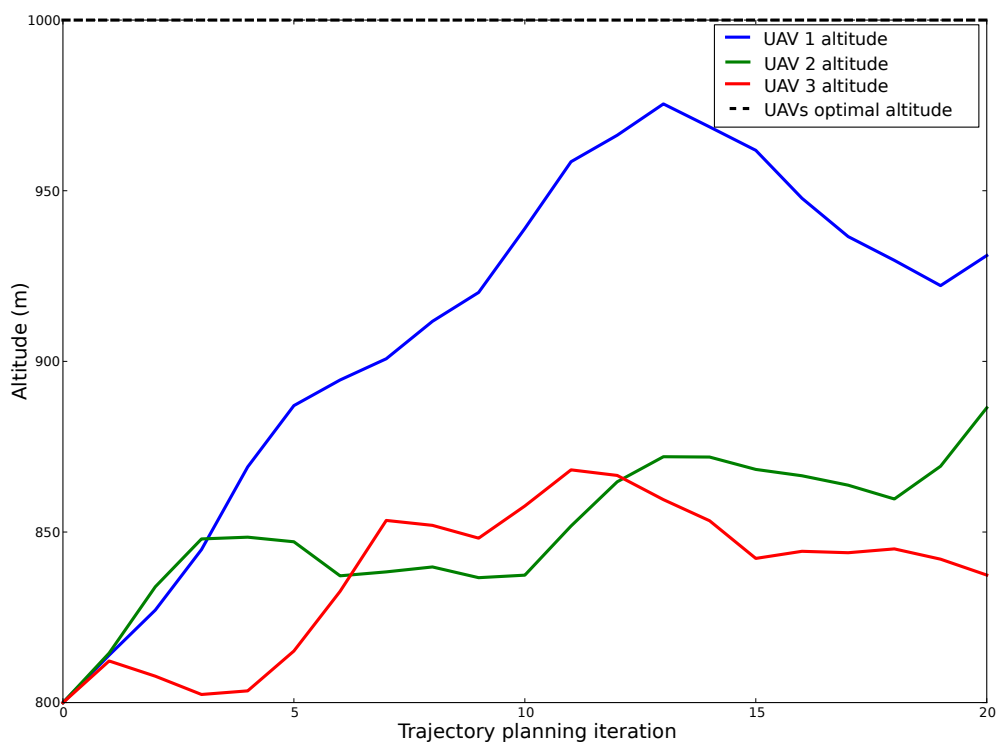


FIGURE 4.11 – Résultats de simulation pour trois agents. Altitude de l’agent à la fin de chaque itération. Les pointillés noirs indiquent l’altitude cible de l’UAV. Une itération de planification correspond à 10 secondes temps réel.

4.3 Simulation système réaliste : intégration avec Paparazzi

Si les simulations actuelles intègrent un environnement atmosphérique réaliste simulé, le modèle d'avion est trop simple pour valider la faisabilité des trajectoires planifiées. Il est donc nécessaire d'intégrer un simulateur de vol réaliste à la simulation. Paparazzi UAV est un projet open-source de développement de matériels et logiciels embarqués, en particulier un autopilote, pour le vol autonome de micro-drones. Papparazi comprend aussi un simulateur, qui depuis peu inclut un modèle de dynamique de vol complet basé sur JSBSim¹. Développée en partie à l'ENAC, cette plate-forme a été choisie pour effectuer l'intégration des différentes parties du projet. Pour nos besoins, trois éléments sont indispensables :

- Le modèle complet d'un micro-drone, incluant un modèle énergétique du moteur
- Une interface de contrôle du drone prenant en entrée les sorties de la planification
- L'intégration des simulations Meso-NH dans Paparazzi comme modèle atmosphérique

Pour le premier point, un premier modèle est en cours d'élaboration par l'équipe de l'ENAC. En parallèle, nous discutons avec eux pour améliorer le modèle utilisé pour la planification, et en particulier prendre en compte l'effet du moteur qui offre une dimension supplémentaire de planification.

Pour le second point, il n'existe pas encore de solution intégrée à Papparazi : il n'y a pas de possibilité de spécifier des trajectoires complexes. Pour l'instant, il est seulement possible d'effectuer quelques figures géométriques simples (lignes droites, cercles . . .) ou une navigation par points de passage suffisamment espacés. Il sera donc nécessaire pour intégrer la planification de trajectoire d'étendre au sein de Paparazzi les capacités de contrôle de l'UAV.

Pour le dernier point, nous avons implémenté la solution utilisée pour la planification décrite dans les sections précédentes au simulateur de vol NPS, en lui adjoignant la capacité de prendre en entrée des conditions atmosphériques envoyées sur un port réseau. Les conditions atmosphériques peuvent pour l'instant soit provenir de notre modèle atmosphérique Meso-NH, soit du modèle d'environnement intégré au simulateur de vol FlightGear².

1. <http://jsbsim.sourceforge.net/>

2. <http://www.flightgear.org/>

Chapitre 5

Conclusions et perspectives

Dans les chapitres précédents, nous avons posé le problème et établi l'état de l'art, avant de proposer une approche pour sa résolution. De manière classique en robotique, nous sommes en présence d'un problème portant sur un *système* dont les composants sont interdépendants : les interactions entre les différents modèles et algorithmes sont capitales, et il est donc nécessaire de valider chaque composante individuellement et à l'intérieur du système. À l'issue de ce stage, c'est donc une ébauche de ce système qui a été implémentée, comportant :

- Un modèle d'avion
- Un modèle d'environnement et les algorithmes nécessaires à sa mise à jour au cours de la mission
- Un bloc de raisonnement et de planification sur les modèles
- Un simulateur météorologique
- Un simulateur de vol

Par la suite, chacun de ces composants devra être amélioré, tout en continuant à nous appuyer sur les résultats de simulations pour diriger nos efforts.

5.1 Discussion des résultats préliminaires

Quoique préliminaires, les résultats de nos premières expériences nous ont permis de dégager un certain nombre de points difficiles qu'il faudra par la suite résoudre. On peut, pour chaque thématique (construction du modèle de l'environnement et planification) distinguer deux catégories : défis techniques et problèmes fondamentaux. Les premiers relèvent de la nécessité de trouver des solutions efficaces, soit pour traiter un grand nombre de données, soit pour que les algorithmes puissent s'exécuter en temps réel. Les seconds correspondent à des problèmes fondamentalement complexes et pour lesquels il n'existe pour l'instant pas de solution appropriée qui se dégage de la littérature. Enfin, un certain nombre de questions dégagées lors de l'analyse du problème et de l'état de l'art restent, faute de temps, en suspens.

5.1.1 Modèle d'environnement

Le modèle de régression s'est révélé efficace pour prédire les vents locaux et permettre la planification de trajectoires intéressantes. Cependant nous avons

dû employer une méthode approchée et fortement limiter le nombre d'exemples dans notre modèle, ce qui limite fortement la taille de la zone explorée. Nous avons pu fournir à l'algorithme de planification une grille d'une dizaine de milliers de points à partir d'un modèle de quelques centaines, et ce, en temps réel. La mise à jour du modèle avec quelques dizaines d'exemples à chaque intervalle de planification ne pose pas non plus de problèmes à cette échelle. L'optimisation des hyper-paramètres cependant reste trop lente pour être viable en temps réel, même avec si peu d'exemples. Pour espérer modéliser avec suffisamment de précision de plus grandes zones, il faudra donc implémenter les solutions techniques permettant un calcul plus rapide des matrices de covariance envisagées au chapitre 3.

Un autre problème mis en évidence est celui de la variabilité des hyper-paramètres lors de cycles répétés d'optimisation : ils sont de qualité variable, l'espace des solutions semblant contenir beaucoup d'extremums locaux. Ainsi, il arrive parfois que l'optimisation ne trouve pas de bonne solution, et celle retenue produit alors des résultats aberrants. Pour limiter ce type de problème, une solution pourrait être de mieux contraindre ces paramètres a priori, soit à partir de connaissances du processus à modéliser, soit par analyse statistique de la distribution de ces hyper-paramètres sur différents résultats de simulations. De plus, le manque de stabilité semble en grande partie provenir de l'ensemble d'exemples utilisé pour l'optimisation de ces paramètres : en rendant cette optimisation plus rapide, il est alors possible d'utiliser un plus grand jeu d'exemples et donc de mieux stabiliser les hyper-paramètres d'une itération à l'autre.

Pour finir, il reste le problème de la validité de l'approche face à la dynamique du nuage. Nous n'avons malheureusement pas eu le temps lors de ce stage d'aborder cette question, qui risque cependant d'être centrale par la suite pour la qualité du modèle. L'analyse d'un modèle spatio-temporel nécessite une résolution spatiale fine, de l'ordre de la seconde. De tels résultats préliminaires de simulations Meso-NH nous ont été récemment transmis, mais la quantité de données (plusieurs To) s'est révélée prohibitive pour l'utilisation des routines d'analyses mises en œuvre jusqu'à présent. L'analyse de ces données nécessitera donc un travail préalable de préparation de celles-ci, ainsi que la mise en place d'un cache local pour éviter les délais importants de communication avec le disque.

5.1.2 Planification

Pour la planification de trajectoires, deux problèmes principaux se posent. Le plus difficile est celui de la définition d'une fonction d'utilité multicritère. Celles employées dans les expériences présentées dans ce rapport sont très simples, et souffrent du problème habituel d'être peu générales, c'est à dire définies par un expert pour satisfaire les contraintes d'un problème particulier. Le défi qui se pose est de définir une stratégie agnostique au problème, et, où les préférences d'un expert sont communiquées via des quantités commensurables et non des paramètres abstraits. L'étude plus poussée des fonctions d'utilité et de décisions au sein du cadre théorique de la décision multicritère et multi-agents semble ici indiquée.

Cependant, la faiblesse la plus évidente ressortant de nos expériences est la

myopie des algorithmes d'optimisation employés. En effet, la taille des problèmes d'optimisation à résoudre ne permet pas d'optimiser une trajectoire sur plus d'une dizaine de secondes : au-delà aucune information n'est prise en compte, ce qui résulte en un mauvais comportement à long terme de l'agent. Or, si l'introduction du niveau de planification de tâches permettrait d'atténuer ce genre de comportements, atténuer la myopie de la planification de trajectoires résulterait en de bien meilleures performances.

Enfin, on observe une baisse très importante de la qualité des résultats lorsqu'on planifie de manière jointe pour plusieurs agents à la fois. La taille de l'espace d'optimisation joint croît en effet exponentiellement avec le nombre d'agents, et donc la qualité des solutions souffre très rapidement de la planification jointe. Or, une partie des buts considérés est strictement locale à l'agent : ainsi chaque agent optimise son énergie indépendamment des autres. Une piste possible serait donc de considérer une planification hybride : chaque agent entretenant des solutions optimales localement, et l'étage de planification multi-agents utilisant ces solutions individuelles en première approximation.

5.2 Perspectives

Les travaux présentés dans ce rapport constituent un instantané au bout de six mois du projet SkyScanner de l'état de nos recherches. Ces travaux vont se prolonger, en collaboration avec les équipes des autres laboratoires jusqu'à fin 2016. Ma participation continuera tout au long de ma thèse, au cours de laquelle je poursuivrai les travaux commencés au cours de ce stage.

A court terme, nos efforts devront porter sur la réalisation de simulations plus réalistes. Pour cela, il faudra tout d'abord optimiser les outils existants pour pouvoir profiter des résultats de simulations à la seconde Meso-NH et ainsi valider la pertinence de notre méthode en environnement dynamique. D'autre part, il faudra interfacer notre planificateur à Paparazzi, pour pouvoir profiter d'une simulation de vol plus réaliste. En parallèle à ces travaux techniques, il sera nécessaire d'améliorer le modèle d'environnement pour pouvoir gérer ces conditions de simulation encore plus complexes. Premièrement, en tirant parti des techniques suggérées dans ce rapport pour améliorer la rapidité et la stabilité du modèle de régression. Deuxièmement, un modèle plus global du nuage devra être développé et servira d'à priori là où aucune observation n'a été faite. Celui-ci sera instancié et précisé au fur et à mesure de la mission par les observations faites in situ, les deux modèles se renforçant ainsi mutuellement. Enfin, on pourra tirer parti des relations entre ces modèles et la planification, en affinant les idées esquissées en fin de chapitre 3.

A moyen terme, il sera nécessaire de s'appuyer sur un planificateur plus élaboré, en particulier en développant une couche de planification de plus haut niveau. Une réflexion sur les fonctions de décisions multi-critères et multi-agents à mettre en œuvre ne devra pas être négligée. Enfin, il faudra tenir compte des contraintes spécifiques de puissance de calcul et de communication propres aux micro-drones pour pouvoir déployer les solutions développées sur les UAVs et valider nos algorithmes en effectuant des vols.

Bibliographie

- [ASGS13] W.H. Al-Sabban, L.F. Gonzalez, and R.N. Smith. Wind-energy based path planning for unmanned aerial vehicles using markov decision processes. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 784–789, May 2013.
- [CLS15] Jen Jen Chung, Nicholas RJ Lawrance, and Salah Sukkarieh. Learning to soar : Resource-constrained exploration in reinforcement learning. *The International Journal of Robotics Research*, 34(2) :158–172, 2015.
- [CO02] Lehel Csató and Manfred Opper. Sparse on-line gaussian processes. *Neural computation*, 14(3) :641–668, 2002.
- [DHP⁺13] J. Das, J. Harvey, F. Py, H. Vathsangam, R. Graham, K. Rajan, and G.S. Sukhatme. Hierarchical probabilistic regression for auv-based adaptive sampling of marine phenomena. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 5571–5578, May 2013.
- [EA14] J. Elston and B. Argrow. Energy efficient uas flight planning for characterizing features of supercell thunderstorms. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 6555–6560, May 2014.
- [ERS⁺11] Jack S Elston, Jason Roadman, Maciej Stachura, Brian Argrow, Adam Houston, and Eric Frew. The tempest unmanned aircraft system for in situ observations of tornadic supercells : design and vortex2 flight results. *Journal of Field Robotics*, 28(4) :461–483, 2011.
- [HPOG12] G. Hollinger, A. Pereira, V. Ortenzi, and Sukhatme G. Towards improved prediction of ocean processes using statistical machine learning. In *Proceedings of the Science and Systems Workshop on Robotics for Environmental Monitoring (RSS)*, 2012.
- [KKP14] Inyoung Ko, Beobkyoon Kim, and Frank Chongwoo Park. Randomized path planning on vector fields. *The International Journal of Robotics Research*, 33 :1664–1682, 2014.
- [LS11] Nicholas RJ Lawrance and Salah Sukkarieh. Autonomous exploration of a wind field with a gliding aircraft. *Journal of Guidance, Control, and Dynamics*, 34(3) :719–733, 2011.
- [MR09] Arman Melkumyan and Fabio Ramos. A sparse covariance function for exact gaussian process inference in large datasets. In *Proceedings*

of the 21st international joint conference on Artificial intelligence, pages 1936–1942. Morgan Kaufmann Publishers Inc., 2009.

- [NLFS13] J. Nguyen, N. Lawrance, R. Fitch, and S. Sukkariéh. Energy-constrained motion planning for information gathering with autonomous aerial soaring. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 3825–3831, May 2013.
- [RRR⁺07] Veerabhadran Ramanathan, Muvva V Ramana, Gregory Roberts, Dohyeong Kim, Craig Corrigan, Chul Chung, and David Winker. Warming trends in asia amplified by brown cloud solar absorption. *Nature*, 448(7153) :575–578, 2007.
- [RSS15] Principles of multi robot systems, workshop at rss2015 conference. <http://mrs-rss2015.sciencesconf.org/>, July 2015.
- [RVS13] S. Ravela, T Vigil, and I Sleder. Tracking and mapping coherent structures. In *International Conference on Computational Science, ICCS 2013*, June 2013.
- [RW06] Carl Edward Rasmussen and Christopher KI Williams. Gaussian processes for machine learning. *the MIT Press*, 2006.
- [RYH11] Ananth Ranganathan, Ming-Hsuan Yang, and Jeffrey Ho. Online sparse gaussian process regression and its applications. *Image Processing, IEEE Transactions on*, 20(2) :391–404, 2011.
- [Sou11] M. Soullignac. Feasible and optimal path planning in strong current fields. *Robotics, IEEE Transactions on*, 27(1) :89–98, Feb 2011.
- [Spa05] James C Spall. *Introduction to stochastic search and optimization : estimation, simulation, and control*, volume 65. John Wiley & Sons, 2005.
- [SSD12] H. Soh, Yanyu Su, and Y. Demiris. Online spatio-temporal gaussian process experts with application to tactile classification. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4489–4496, Oct 2012.
- [VV10] Jarno Vanhatalo and Aki Vehtari. Speeding up the binary gaussian process classification. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*, number arXiv preprint arXiv :1203.3524, 2010.
- [WS12] Ryan K. Williams and G. Sukhatme. Probabilistic spatial mapping and curve tracking in distributed multi-agent systems. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1125–1130, May 2012.

Annexe A

Modèle d'avion

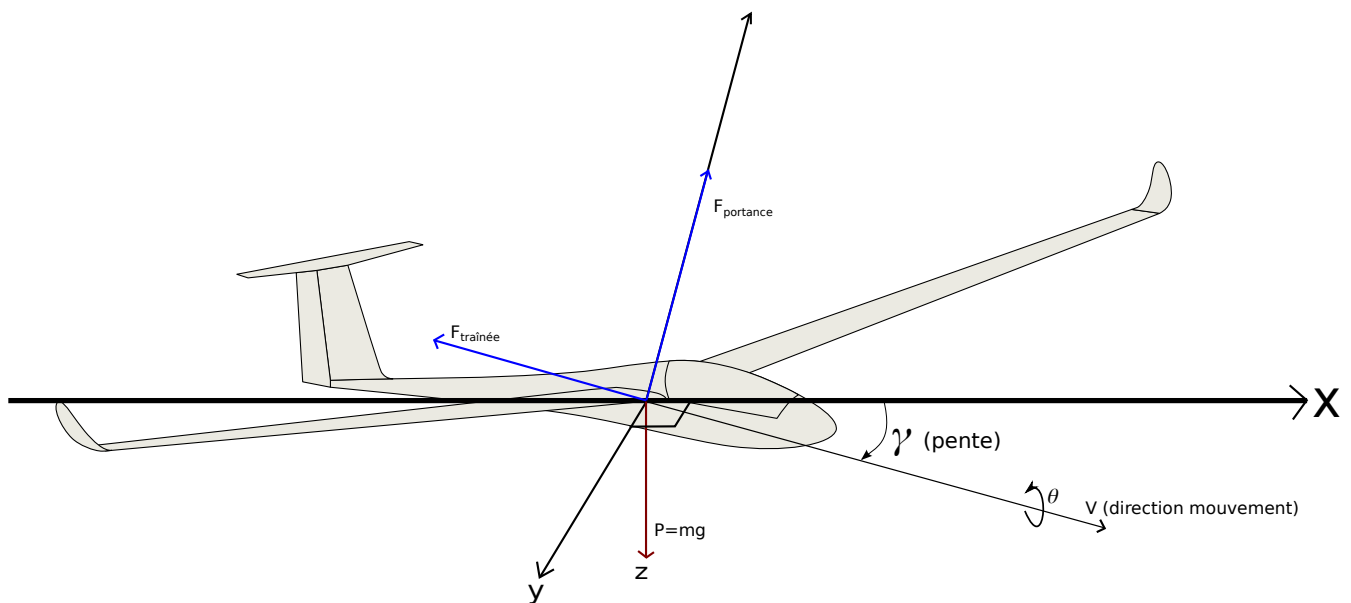


FIGURE A.1 – Modèle de situation d'un UAV et forces appliquées.

Axe Ox : tangent au virage.

Axe Oy : vers le centre du virage.

Axe Oz : vers le bas.

V : direction du vecteur vitesse.

$F_{portance}$: portance aérodynamique, perpendiculaire au vecteur vitesse (mouvement), vers le haut.

$F_{traînée}$: traînée aérodynamique, dans la direction contraire à celle du mouvement.

P : poids, vers le bas.

γ : pente du mouvement.

θ : roulis.

ψ (non représenté) : cap de l'UAV dans le référentiel terrestre. Dans la direction de l'axe Ox .

Pour simplifier le modèle, on ne prend pas en compte les périodes d'accélération autres que celles nécessaires à la réalisation du virage. Soit F la somme totale des forces dans le repère (Ox, Oy, Oz) on a donc

$$0 = F_x = F_z \quad (\text{A.1})$$

De plus la force résultante centripète nécessaire à maintenir le virage s'exprime sur l'axe Oy , par définition :

$$F_y = m \frac{V_{xy}^2}{r} \quad (\text{A.2})$$

$$r = \frac{V_{xy}}{\dot{\psi}} \quad (\text{A.3})$$

$$F_y = m \frac{V_{xy}^2 \dot{\psi}}{V_{xy}} = m V_{xy} \dot{\psi} \quad (\text{A.4})$$

avec V_{xy} la vitesse horizontale de l'avion, $\dot{\psi}$ la variation de cap, m la masse de l'avion et r le rayon de braquage.

Or par analyse des forces en jeu :

$$0 = F_x = F_{trainée} - P \sin \gamma = -\frac{1}{2} \rho S V^2 C_x - mg \sin \gamma \quad (\text{A.5})$$

$$0 = F_z = F_{portance} \cos \theta - P \cos \gamma = \frac{1}{2} \rho S V^2 C_z \cos \theta - mg \cos \gamma \quad (\text{A.6})$$

avec P le poids de l'UAV, ρ la masse volumique de l'air, S la surface de portance, C_x le coefficient de trainée et C_z le coefficient de portance.

On déduit donc de A.5 et A.6 :

$$\frac{1}{\tan \gamma} = \frac{C_z}{C_x} \cos \theta \quad (\text{A.7})$$

$$\frac{1}{\tan \gamma} = \frac{L}{D} \cos \theta \quad (\text{A.8})$$

$$\gamma = \arctan \frac{1}{\frac{L}{D} \cos \theta} \quad (\text{A.9})$$

avec $\frac{L}{D}$ la finesse de l'avion.

Par application des forces sur l'axe Oy on en déduit

$$F_y = F_{portance} \sin \theta = \frac{1}{2} \rho S V^2 C_z \sin \theta \quad (\text{A.10})$$

Puis en utilisant la définition précédente de F_y A.4 :

$$\sin \theta = 2 \frac{m \dot{\psi}}{\rho S C_z} \frac{V_{xy}}{V^2} \quad (\text{A.11})$$

Rappelons que

$$V_{xy} = V_0 \cos \gamma \quad (\text{A.12})$$

$$V_z = V_0 \sin \gamma \quad (\text{A.13})$$

et donc par définition de V_{xy}

$$\sin \theta = 2 \frac{m\dot{\psi}}{\rho S C_z} \frac{\cos \gamma}{V} \quad (\text{A.14})$$

Puis par définition de F_z (équation A.6) :

$$\cos \gamma = \frac{1}{2} \frac{\rho S V^2 C_z \cos \theta}{mg} \quad (\text{A.15})$$

On remplace alors A.15 dans A.14

$$\sin \theta = 2 \frac{m\dot{\psi}}{\rho S C_z} \frac{\rho S C_z V^2 \cos \theta}{2mg} \frac{1}{V} \quad (\text{A.16})$$

$$= \frac{\dot{\psi} V}{g} \cos \theta \quad (\text{A.17})$$

$$\tan \theta = \frac{\dot{\psi} V}{g} \quad (\text{A.18})$$

Et donc

$$\theta = \arctan \left(\frac{\dot{\psi} V}{g} \right) \quad (\text{A.19})$$

En fin de compte :

$$\vec{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \quad (\text{A.20})$$

$$v_x = V \cos \gamma \cos \psi \quad (\text{A.21})$$

$$v_y = V \cos \gamma \sin \psi \quad (\text{A.22})$$

$$v_z = V \sin \gamma \quad (\text{A.23})$$

$$\gamma = \arctan \left(\frac{1}{\frac{L}{D} \cos \theta} \right) \quad (\text{A.24})$$

$$\theta = \arctan \left(\frac{\dot{\psi} V}{g} \right) \quad (\text{A.25})$$

Annexe B

RSS workshop : poster
abstract

Monitoring Atmospheric Phenomena within Low-Altitude Clouds with a Fleet of Fixed-Wing UAVs

Alessandro Renzaglia^{1,2}, Christophe Reymann^{1,3} and Simon Lacroix^{1,2}

I. INTRODUCTION

Atmospheric scientists have been early users of UAVs, from which significant scientific results have rapidly been obtained (*e.g.* [1]). UAVs indeed bring forth several advantages over manned flight to probe atmospheric phenomena: low cost, ease of deployment, possibility to evolve in high turbulences [2], etc. Yet atmospheric phenomena span the three spatial dimensions and evolve over time, and their analysis requires much more data than a single UAV can gather. Fleets of UAVs are the natural solution to gather more and more relevant information, and they can especially collect synchronised observations of a series of distant areas.

This article depicts on-going work on the development of a fleet control approach to probe low-altitude cumulus clouds. From an atmospheric science point of view, there remain numerous uncertainties and even unknowns in the cloud micro-physics models that could be alleviated with the acquisition of a variety of data *within and around* the cloud. Wind currents, pressure, temperature, humidity, liquid water content, radiance, aerosols are data of interest that must be collected with a spatial and temporal resolution of respectively about 10 m and 1 Hz over the cloud lifespan. Deploying a fleet of UAVs for this purpose raises a series of challenges: exploring the cloud is a poorly informed and highly constrained adaptive sampling problem, in which the UAVs motions must be defined so as to maximize the amount of gathered information and the mission duration.

A wholesome global approach has been defined, which casts the overall problem in a hierarchy of two modeling and decision stages. A macroscopic parametrized model of the cloud is exploited at the higher level to set information gathering goals, possibly with an atmospheric scientist in the loop, to each of which a subset of the UAV team is allocated, considering *e.g.* their current position in the cloud, their on-board energy level, and their sensing capacities (because of payload constraints, the UAVs may not all embark the same sensor suite). These goals typically consist of cloud regions to explore, and are handled by the lower level, which optimizes the selected UAVs trajectories using an on-line updated dense model of the variables of interest.

The article focuses on this latter level. It sketches the modeling and the trajectory generation processes that actively drive a handful of UAVs within a given area, aiming at max-

imising the information gain while minimizing the energy consumption.

II. ENVIRONMENT MODEL

To plan energy-efficient and informative trajectories, a model that represents both the wind currents and the atmospheric variables to measure is required. The accuracy of these information is of course of utmost importance, as it is the dimension that steers the information gathering and that conditions the expectation of the path costs estimates.

The considered context raises two main issues: the size of the three-dimensional space, in which UAVs collect very sparse measurements, and the dynamics of the considered atmospheric phenomena. The only way to be able to predict short term atmospheric conditions from the sparse measurements is to make use of the strong spatio-temporal correlations of the atmospheric processes. Considering the little available knowledge about these, there are no many efficient tools to tackle this problem.

Recent work have shown that Gaussian Processes (GP) can successfully be used to perform spatio-temporal regression in robotics problems. GP is a very general non-parametric framework, where the underlying process is modeled by “a collection of random variables, any finite number of which have a joint Gaussian distribution” [3]. Under this assumption, the process is defined only by its mean and covariance functions. The mean function is often assumed to be zero, but it can be used to set a prior. The covariance function represents similarity between points: given a set of n samples (x, y) and assuming zero mean, the GP $y = f(x)$ is fully defined by the Gram matrix $K_{n,n} = [k(x_i, x_j)]$ of the covariances between the sample points. The choice of the kernel function k conditions the process distribution as it sets a prior on the process properties such as isotropy, stationarity or smoothness. The particularity of the GP model is to provide full predictive distributions over all possible f , whose mean and variance at each point can be interpreted as the process predicted value and associated error.

Until recently, the usage of GP models for online problems has been prevented by prohibitive inference cost in $\mathcal{O}(n^3)$, due to the Cholesky decomposition of the K matrix which must be updated each time new samples are added. Recent algorithmic advances for streaming data and greater computing power spawned several contributions solving online problems in the robotics community [4].

Working with meteorologists, we aim to prove that this method is adequate for mapping atmospheric conditions by assessing its efficiency on realistic data from meso-scale

¹CNRS, LAAS, 7 av du colonel Roche, F-31400 Toulouse, France; {firstname.lastname} at laas.fr

²Univ de Toulouse, LAAS, F-31400 Toulouse, France

³Univ de Toulouse, UPS, LAAS, F-31400 Toulouse, France

simulations. This work focuses on choosing and comparing appropriate kernel functions to take advantage of domain specific priors, and show that realistic atmospheric variables can successfully be estimated online by a fleet of UAVs (figure 1).

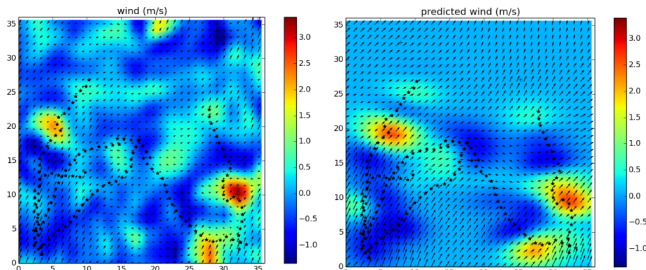


Fig. 1. Illustration of the estimation of the horizontal wind direction and speed provided the application of GP regression using data perceived along two UAVs trajectories (denoted by black stars). Left: ground truth, output of a realistic atmospheric simulation; right: predicted wind.

III. TRAJECTORIES GENERATION

The GP regression scheme reconstructs a local air flow and atmospheric variables map using measurements, from which optimal trajectories that maximize the information gathered can be generated. The motion model of the UAVs assumes a fixed constant speed v_0 with respect to the air. The resultant ground velocity is $v = v_0 + c$, where c is the air speed. The motion for each UAV is then completely characterized by the values of its three directional angles. We do not assume any constraint on the current speed, which can be greater than v_0 , and hence define unreachable areas. For the sake of simplicity we consider here the two-dimensional version of the problem, in which only one control angle α determines the UAV motion. Trajectories are generated over a short time horizon ΔT , defined by the frequency at which the GP hyper-parameters are updated. Within this time horizon, we consider m sections of duration dt in which the UAV directions are constant: the trajectory for the robot j is described by the sequence of angles $\alpha_i^{(j)}$, $i \in \{1, \dots, m\}$.

We can now formulate the trajectory generation problem as a constrained optimization problem:

$$\operatorname{argmax}_{\alpha^{(1)}, \dots, \alpha^{(N_r)}} \sum_{t=t_0}^{t_0+\Delta T} U(\mathbf{x}_t^1(\alpha^{(1)}), \dots, \mathbf{x}_t^{N_r}(\alpha^{(N_r)})) \quad (1)$$

$$s.t. \quad |\alpha_i^{(j)}| \leq \alpha_{max} \quad \forall i, j \quad (2)$$

where $\mathbf{x}_t^{(j)}$ is the position of the UAV j at time t , $U(\cdot)$ is the utility map and N_r is the number of UAVs involved in the mission. To tackle this optimization, we propose a two-step approach: a first phase based on a blind random search in order to have a good trajectories initialization, followed by a gradient ascent algorithm to optimize them. To perform the gradient ascent we adopted a constrained version of the Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm [5], which ensures a faster convergence to a local maximum with respect to classic

gradient approximation algorithms. The blind random search is achieved creating a set of feasible trajectories obtained by a constrained random sampling of directions angles α_i , and exploiting the approximated field generated by the GP regression. The trajectories are then evaluated using the local utility map U , as in eq. (1), and the best set of N_r trajectories is the initial configuration for the gradient ascent phase. The presence of the first sampling step is due to the strong dependence of the gradient-based solution on the initial configuration. In this way, even though we only have local convergence guarantees, the probability of getting stuck in local maxima far for the global optimal trajectories is reduced. Figure 2 shows some trajectories obtained in a fictitious two-dimensional current field, that steers the UAVs towards high utility regions while trying to follow the wind direction.

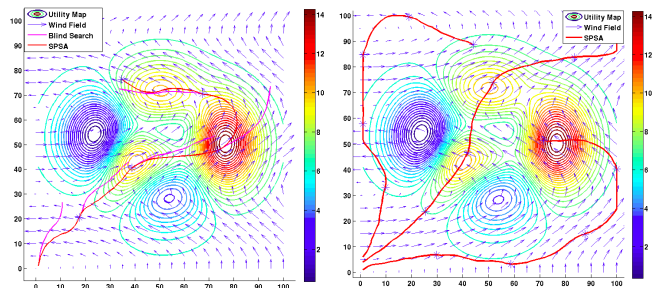


Fig. 2. Left: one UAV is moving in a 2D environment where a scalar utility map and a wind field are defined. The trajectories initialized by a blind search at every time-horizon ΔT are shown in magenta, and the final trajectories provided by the SPSA algorithm are in red. Right: 3 UAVs are steered in the same environment to maximize the gathered information (only the final trajectories are shown).

IV. FUTURE WORK

By integrating tightly the environment modeling and the path planning processes, we hope to achieve a mutually beneficial improvement. Indeed, by taking into account not only the predicted values and errors, but also the shape of the kernel function to drive the path planning algorithm, one should allow the on-line definition of the spatio-temporal scales, adapted to the atmospheric conditions at hand.

ACKNOWLEDGMENTS

This work is made in the context of the SkyScanner project, supported by the STAE foundation.

REFERENCES

- [1] V. Ramanathan, M. V. Ramana, G. Roberts, D. Kim, C. Corrigan, C. Chung, and D. Winker, "Warming trends in asia amplified by brown cloud solar absorption," *Nature*, vol. 448, no. 7153, pp. 575–578, 2007.
- [2] J. S. Elston, J. Roadman, M. Stachura, B. Argrow, A. Houston, and E. Frew, "The tempest unmanned aircraft system for in situ observations of tornadic supercells: Design and vortex2 flight results," *Journal of Field Robotics*, vol. 28, p. 461–483, 2011.
- [3] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. the MIT Press, 2006.
- [4] N. R. Lawrence and S. Sukkariéh, "Path planning for autonomous soaring flight in dynamic wind fields," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011.
- [5] J. C. Spall, *Introduction to stochastic search and optimization: estimation, simulation, and control*. John Wiley & Sons, 2005, vol. 65.