



**HAL**  
open science

## Simulateur de flotte de drones

Osamah Saif, Zhicheng Hou, Isabelle Fantoni, Guillaume Sanahuja

► **To cite this version:**

Osamah Saif, Zhicheng Hou, Isabelle Fantoni, Guillaume Sanahuja. Simulateur de flotte de drones. 4èmes Journées Démonstrateurs, Jun 2013, Angers, France. hal-00927086

**HAL Id: hal-00927086**

**<https://hal.science/hal-00927086v1>**

Submitted on 10 Jan 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Simulateur de flotte de drones

Osamah Saif<sup>1</sup>, Zhicheng Hou<sup>1</sup>, Isabelle Fantoni<sup>1</sup>, Guillaume Sanahuja<sup>1</sup>

<sup>1</sup>Heudiasyc, UMR 7253, Université de Technologie de Compiègne, Centre de Recherches de Royallieu, BP 20529 60205 COMPIEGNE cedex - France

*osamah.saif / zhicheng.hou / isabelle.fantoni / gsanahuj@hds.utc.fr*

## Resumé

Dans cet article, nous présentons un simulateur multi-drones. La stratégie de contrôle utilisée dans ce simulateur est une approche décentralisée. L'objectif est d'utiliser une loi de commande qui est indépendante du nombre d'agents composant la flottille de drones. Une commande linéaire PD est la base de développement de cette stratégie de contrôle. Différents scénarios sont mis en œuvre dans ce simulateur.

**Mots-clés:** *UAV, commande décentralisée, simulateur, agrégation, navigation.*

## 1 Introduction

Depuis quelques années, différentes applications de drones aériens (Unmanned Aerial Vehicles) voient le jour dans les domaines militaires, mais aussi civils. L'utilisation de plusieurs véhicules en flotte permet de couvrir de vastes zones de recherche, de surveillance, d'inspection etc.. Récemment, l'attention de certains chercheurs scientifiques s'est tournée vers l'imitation du comportement des oiseaux dans une flotte. Ils tentent d'effectuer un vol en formation autonome.

Avant que les scientifiques s'intéressent au vol en lui-même, le comportement collectif des animaux a attiré l'attention des biologistes. Ils ont étudié des volées d'oiseaux, des essaims d'insectes, des troupeaux de quadrupède et des bancs de poissons. Ils ont essayé de comprendre le secret du mouvement collectif, harmonique et sans collision des individus. En outre, ils ont cherché à découvrir les motivations qui conduisent les animaux à se rassembler en groupes comme des volées d'oiseaux. Pour en savoir plus sur ces sujets, on peut consulter [1], [2].

Il existe différentes stratégies de contrôle de la formation de vol. Nous citons ici trois d'entre eux : *Meneur-suiveur (approche hiérarchique)*, *Meneur virtuel* et *Commande basée sur le comportement (approche décentralisée)*. Dans l'approche de meneur-suiveur, les individus dans la formation suivent un agent (ou un avion) qui est désigné comme le chef de file. Une trajectoire de la mission de vol en formation est suivie par le chef, et les disciples suivent le chef. Cette stratégie est simple et largement mise en œuvre dans la formation des agents multiples [3], [4]. Cependant, il révèle quelques inconvénients. Par exemple, toute la formation dépend d'un agent et si un problème apparaît pour le guide, toute la formation sera affectée.

La méthode de meneur virtuel consiste à remplacer un chef de la formation, de l'approche de meneur-suiveur, par un meneur virtuel. Tous les agents de la formation reçoivent la trajectoire de la mission. Cette trajectoire est celle du meneur virtuel lui-même. Un des principaux inconvénients de cette stratégie est que le risque de collision entre les agents augmente, car il n'y a pas de retour (feedback) à la formation. Par ailleurs, la trajectoire prédéfinie de la mission diminue l'autonomie de la formation [4].

Dans la stratégie de contrôle comportemental, chaque agent suit certaines règles pour parvenir à la formation. L'objectif du contrôle de la formation est, par conséquent, décomposé en règles

plus petites. En fait, cette méthodologie est inspirée du mouvement collectif des animaux. Un des premiers travaux techniques sur cette stratégie est le modèle distribué de comportement proposé par *Reynolds* dans [5]. Bien que *Reynolds* était spécialisé en infographie, son travail a inspiré les chercheurs en automatique et en robotique qui ont appliqué les règles de *Reynolds* dans le cadre théorique et expérimental. *Reynolds* a inspiré ses règles grâce aux biologistes pour étudier le mouvement collectif des animaux. Il a estimé que chaque individu dans un troupeau doit suivre ces règles afin d'effectuer le comportement de groupe. Ces règles sont les suivantes :

- 1) Evitement de collisions : éviter les collisions avec les voisins proches ;
- 2) Mise en correspondance de la vitesse : tentative de faire correspondre la vitesse avec celles des voisins proches ;
- 3) Centrage de la flotte : tentative de rester près de voisins proches.

Ces règles ont été citées par plusieurs travaux en automatique et en robotique [10], [8]. Ils ont utilisé ces règles pour développer des stratégies de commande et d'algorithmes pour le rassemblement de systèmes multi-agents dynamiques.

Dans cet article, une stratégie de commande décentralisée pour atteindre une flotte de plusieurs UAVs est utilisée. Cette stratégie a été introduite dans [6] et dans [7]. Dans ces deux précédents travaux, le modèle des quadrirotors, utilisé pour la validation des algorithmes de commande, est un modèle linéaire. En revanche, dans cet article, les mêmes algorithmes de commande sont implémentés dans un simulateur qui utilise un modèle non linéaire de quadrirotors. Ce simulateur est destiné pour simuler une flotte de quadrirotors dans un environnement semblable au temps réel.

Les grandes lignes de cet article sont : les préliminaires sur la modélisation de plusieurs drones par la théorie des graphes, sur le cadre d'application, sur le modèle utilisé pour chaque drone et sur la stratégie de commande d'une flotte de drones sont, d'une part abordés dans la section 2. Ensuite, le simulateur multi-drones est présenté dans la section 3. Par ailleurs, différents scénarios de formation et de navigation multi-UAVs sont présentés dans la section 4. Enfin, nous concluons par des remarques et des indications sur des travaux futurs dans la section 5.

## 2 Préliminaires

Dans cette section, nous présentons premièrement une méthodologie de modélisation d'une flotte de drones aériens en utilisant la théorie des graphes. Cette méthodologie a été utilisée précédemment dans [11] et [10]. Cela permet de représenter graphiquement et analytiquement l'interaction de plusieurs drones. Pour en savoir plus sur la théorie des graphes, on peut se référer à [9]. Ensuite, la structure de la flotte est formulée dans l'espace euclidien et le modèle dynamique des véhicules aériens utilisé dans ce travail est présenté. Finalement, une brève introduction à la stratégie de commande multi-drones est présentée.

### 2.1 Modélisation de multiples UAVs par la théorie des graphes

Un système composé de multiples UAVs peut être vu comme un graphe  $G = (V, E)$ . Les UAVs sont représentés par un ensemble de sommets (points)  $V = \{v_i\}$ ,  $i \in I = \{1, \dots, N\}$  où  $N$  est le nombre d'UAVs. Les connections entre les UAVs sont représentées par un ensemble de liens  $E = \{e_{i,j}\}$  et  $e_{i,j}$  est le lien entre  $(v_i, v_j)$  avec  $i, j \in I$ . Une matrice de proximité  $A$  est une matrice  $N \times N$  dont les éléments  $a_{i,j} = 1$  si  $e_{i,j} \in E$  et  $a_{i,j} = 0$  sinon. Dans un espace euclidien, les sommets représentent les positions des UAVs.

Un graphe est appelé indirect si  $E$  satisfait l'équivalence suivante :  $\forall i, j \in I, i \neq j, e_{i,j} \in E \Leftrightarrow e_{j,i} \in E$ . Par conséquent, notre graphe est indirect puisque chaque UAV mesure les états des autres

UAVs. Les voisins d'un UAV  $v_i$  peuvent être modélisés par un ensemble  $\eta_i = \{v_j \in V : a_{i,j} \neq 0\}$ . La Figure 1 montre quatre UAVs représentés comme un graphe indirect.

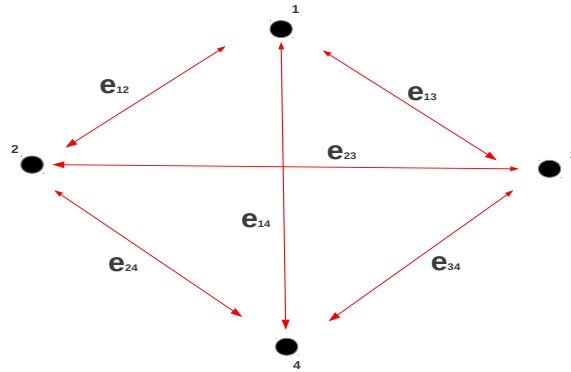


FIGURE 1 – Quatre UAVs dans un graphe

## 2.2 Structure euclidienne

Le cadre d'application est un espace euclidien. Par conséquent, il devrait y avoir une correspondance entre l'espace du graphe et l'espace euclidien. Les sommets correspondent aux positions des UAVs. Ainsi, un sommet  $v_i$  possède sa position correspondante  $\mathbf{p}_i \in \mathbb{R}^f$  où  $f$  est la dimension de l'espace.

Nous redéfinissons les voisins d'un UAV  $v_i$  dans l'espace euclidien par  $\eta_i = \{v_j \in V : \|\mathbf{p}_j - \mathbf{p}_i\| < c\}$ , où  $\|\cdot\|$  est la norme euclidienne, et  $c$  est la portée du voisinage.

Un des objectifs de la commande en flottille de multiples UAVs est d'assurer l'anti-collision des individus de la flotte. La contrainte de la commande peut être écrite comme suit :

$$\|\mathbf{p}_j - \mathbf{p}_i\| = d, \quad \forall v_j \in \eta_i, \forall v_i \in V \quad (1)$$

où  $d$  est une distance de sécurité ou le rayon de la région d'anti-collision, avec  $d < c$ .

## 2.3 Modèle et commande d'un drone (UAV)

Ce démonstrateur est un drone de type X4, ou quadrirotor (voir figure 2). Ce drone à voilure tournante a l'avantage de décoller et atterrir verticalement, mais aussi de pouvoir effectuer du vol stationnaire. Le quadrirotor est sans doute l'une des configurations de drone la plus utilisée. En effet, la forme de la structure lui permet de ne pas avoir d'autre actionneur que ses quatre moteurs, ce qui la rend mécaniquement très simple et facile à construire. Les moments de roulis et tangage sont obtenus en jouant sur les vitesses de rotation de deux moteurs opposés ; de plus, deux des quatre rotors ont une hélice à pas inverse ce qui permet d'annuler le couple de lacet.

Les différents couples et forces dus aux moteurs pris en compte dans cette étude sont représentés sur la figure 3. Ainsi, chacun des quatre moteurs  $M_i$  produit une force  $f_i$  et un couple  $\tau_i$  sur l'axe  $z_b$ . La poussée totale est donc  $u = f_1 + f_2 + f_3 + f_4$ . Le couple  $\tau_{x_b}$  autour de l'axe  $x_b$  est obtenu par la différence de forces  $f_2 - f_4$  et le couple  $\tau_{y_b}$  autour de l'axe  $y_b$  par la différence de forces  $f_1 - f_3$ . Enfin le couple  $\tau_{z_b}$  autour de l'axe  $z_b$  est obtenu par la somme des couples produits par les moteurs  $\tau_1 + \tau_3 - \tau_2 - \tau_4$  ;  $M_1$  et  $M_3$  tournant en effet dans le sens positif alors que  $M_2$  et  $M_4$  tournent en sens inverse.

Le modèle dynamique du quadrirotor est obtenu en supposant que les pales des rotors sont indéformables et que la force de trainée ainsi que l'effet de sol sont négligeables. Malgré cela, le modèle obtenu (voir [12]) est non linéaire et contient de nombreux termes de couplage entre

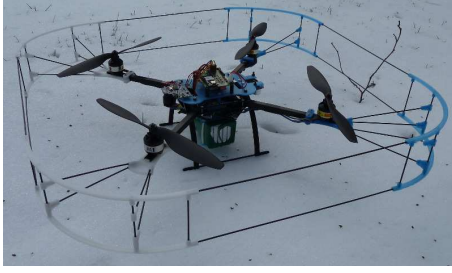


FIGURE 2 – Photo du démonstrateur.

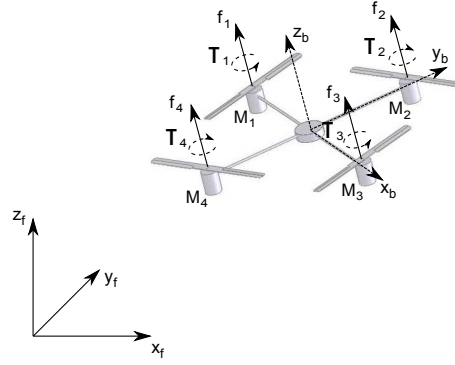


FIGURE 3 – Schéma du quadrirotor.

les différents états. Cependant, le quadrirotor est souvent étudié dans le cas du vol stationnaire ou quasi stationnaire (déplacements latéraux très lents par faibles changements d'orientations). Dans ce cas, le modèle peut être linéarisé autour de la position d'équilibre et revient à :

$$m\ddot{x} = u(\theta + \psi\phi) \quad (2a)$$

$$m\ddot{y} = u(\theta\psi - \phi) \quad (2b)$$

$$m\ddot{z} = u - mg \quad (2c)$$

$$I_{b_{xx}}\ddot{\phi} = \tau_{x_b} \quad (2d)$$

$$I_{b_{yy}}\ddot{\theta} = \tau_{y_b} \quad (2e)$$

$$I_{b_{zz}}\ddot{\psi} = \tau_{z_b} \quad (2f)$$

où  $\xi = [x, y, z]^T$  représente la position du centre de gravité du quadrirotor par rapport au repère fixe,  $\eta = [\phi, \theta, \psi]^T$  sont ses angles d'Euler,  $m$  est sa masse et  $I_{b_{xx}}, I_{b_{yy}}, I_{b_{zz}}$  les différents moments d'inertie.

La loi de commande retenue est basée sur des fonctions de saturations, dans laquelle chaque état est borné séparément. La stabilité d'une telle loi a été prouvée dans les travaux de [13]. Appliquée au quadrirotor, nous obtenons :

$$u = \frac{-\sigma_{pz}(k_{pz}(z - z_d)) - \sigma_{dz}(k_{dz}\dot{z}) + mg}{\cos\theta \cos\phi} \quad (3a)$$

$$\tau_{x_b} = \frac{I_{b_{xx}}}{g} \left[ \sigma_{py}(k_{py}(y - y_d)) + \sigma_{dy}(k_{dy}\dot{y}) - \sigma_{p\phi}(k_{p\phi}\phi) - \sigma_{d\phi}(k_{d\phi}\dot{\phi}) \right] \quad (3b)$$

$$\tau_{y_b} = -\frac{I_{b_{yy}}}{g} \left[ \sigma_{px}(k_{px}(x - x_d)) + \sigma_{dx}(k_{dx}\dot{x}) + \sigma_{p\theta}(k_{p\theta}\theta) + \sigma_{d\theta}(k_{d\theta}\dot{\theta}) \right] \quad (3c)$$

$$\tau_{z_b} = -I_{b_{zz}} \left( \sigma_{p\psi}(k_{p\psi}\psi) + \sigma_{d\psi}(k_{d\psi}\dot{\psi}) \right) \quad (3d)$$

où  $k_{pz}, k_{dz}, k_{py}, k_{dy}, k_{p\phi}, k_{d\phi}, k_{px}, k_{dx}, k_{p\theta}, k_{d\theta}, k_{p\psi}$  et  $k_{d\psi}$  sont des constantes positives. Les variables  $x_d, y_d$  et  $z_d$  représentent la position désirée. Les fonctions  $\sigma_{pz}, \sigma_{dz}, \sigma_{py}, \sigma_{dy}, \sigma_{p\phi}, \sigma_{d\phi}, \sigma_{px}, \sigma_{dx}, \sigma_{p\theta}, \sigma_{d\theta}, \sigma_{p\psi}$  et  $\sigma_{d\psi}$  sont des fonctions de saturations du type :

$$\sigma_{b_i}(s) = b_i, \text{ si } :s > b_i \quad (4a)$$

$$\sigma_{b_i}(s) = s, \text{ si } : -b_i \leq s \leq b_i \quad (4b)$$

$$\sigma_{b_i}(s) = -b_i, \text{ si } :s < -b_i \quad (4c)$$

## 2.4 Commande multi-drones

Le problème de multiples agents (ou véhicules) est que le coût de calcul et la complexité de la conception de la commande augmentent avec le nombre d'agents et la connectivité entre les agents [11]. Dans cette section, nous introduisons brièvement une stratégie de commande multi-agents qui prouve l'indépendance de la conception de commande sur le nombre d'agents et de la connectivité. De plus, cette stratégie de commande proposée est mise en œuvre de manière totalement décentralisée. La nature linéaire de la commande utilisée est l'une des principales caractéristiques simples qui nous a permis d'élaborer cette stratégie. Dans cet article, nous supposons que chaque agent peut mesurer les états de tous les agents dans son champ de vision.

La partie la plus importante de notre travail est la génération de référence. En effet, nous affirons que le rassemblement en flotte est un problème de génération de référence plutôt qu'un problème de conception de commande. Comme nous sommes dans le contexte entièrement décentralisé, nous définissons un *agent focal* (central) qui représente chaque agent seul dans un problème de formation d'une flotte. Ce qui est valable pour l'agent central est également applicable pour chacun des autres agents.

L'idée principale est de préparer une entrée de référence pour l'agent focal. Cette référence représente tous les états mesurés des agents à portée de vue de l'agent focal. La loi de commande pour l'agent focal est conçue de telle sorte qu'elle stabilise la dynamique de l'agent focal et peut suivre une référence.

Nous commençons par définir le profil de la référence pour une interaction entre deux agents.  $v_i$  est l'agent focal et  $v_j$  est l'agent détecté dans le champ de vision de l'agent focal. Par conséquent, le vecteur de référence  $\mathbf{r}_j$  généré en raison de l'occurrence de l'agent  $v_j$  dans le champ de vision de l'agent  $v_i$  est défini comme :

$$\mathbf{r}_j = \begin{pmatrix} \mathbf{p}_j - d \mathbf{n}_{ij} \\ \dot{\mathbf{p}}_j \end{pmatrix} \quad (5)$$

$\mathbf{n}_{ij}$  est le vecteur unitaire directionnel entre les agents  $i$  et  $j$  défini par :

$$\mathbf{n}_{ij} = \frac{\mathbf{p}_j - \mathbf{p}_i}{\|\mathbf{p}_j - \mathbf{p}_i\|} \quad (6)$$

Considérons maintenant  $N$  agents qui tendent à former une flotte. La loi de commande suivante d'un agent focal  $i \in V$

$$\mathbf{u}_i = -Kx_i + C_a \quad (7)$$

assure et stabilise l'agrégation et la formation de la flotte de  $N$  agents, où  $K$  est la matrice de gain de commande par retour d'état.  $C_a$  est la fonction de génération de référence du système en boucle fermée de l'agent focal  $i$  qui est écrite comme suit :

$$C_a = \frac{1}{N} \sum_{j=1}^N \mathbf{r}_j \quad (8)$$

Concernant la navigation d'une flotte de drones vers une destination désirée, la stratégie de commande précédente peut être modifiée pour qu'elle assure un tel comportement. En fait, si nous

modélisons la destination désirée comme un agent statique, *i.e.*, une position fixe, une vitesse nulle et une distance de sécurité égale à zéro, nous pouvons écrire la référence de navigation comme suit :

$$\mathbf{r}_{nav} = \begin{pmatrix} \mathbf{p}_{nav} \\ \mathbf{0} \end{pmatrix} \quad (9)$$

Par conséquent, la fonction de génération de référence peut être écrite comme ci-dessous :

$$\mathbf{C}_a = \frac{1}{N+1} \left( \sum_{j=1}^N \mathbf{r}_j + \mathbf{r}_{nav} \right) \quad (10)$$

Une démonstration plus détaillée de cette stratégie de commande peut être trouvée dans [6] et [7].

**Remarque 1** *Cette stratégie est une loi de commande simple qui ne nécessite pas un coût de calcul important.*

**Remarque 2** *La distance de sécurité  $d$  est dépendante du nombre d'agents. Les écarts entre les individus vont converger vers des distances autour de  $d$ . Plus le nombre d'agents est grand, plus la distance de sécurité doit être élevée.*

### 3 Simulateur multi-drones

Dans le cadre du projet Robotex, le laboratoire Heudiasyc va s'équiper d'une flotte de drones afin d'effectuer des travaux de recherche sur le vol en formation. Cependant les vols en formation peuvent s'avérer risqués, c'est pourquoi le laboratoire développe un simulateur de flotte de drones. Son but est de pouvoir exécuter sur un ordinateur un code identique à celui utilisé dans les vrais drones afin d'effectuer toutes les étapes de mises au point en toute sécurité. Pour cela, l'ordinateur fonctionne sous Linux avec son extension temps réel *Xenomai*, tout comme le drone. Des capteurs et actionneurs virtuels sont reliés à un modèle du drone permettant de calculer les différents états à chaque instant. De plus, les drones évoluent dans un environnement 3D virtuel (voir figure 4), grâce au moteur 3D *Irrlicht*, ce qui permet de placer des caméras sur les drones afin d'effectuer du traitement d'image. Celles-ci sont alors traitées tout comme sur le drone, car l'ensemble des fonctions du *DSP* (intégré au calculateur du drone) peuvent être émulées sur le PC. Le moteur 3D *Irrlicht* permet par ailleurs de gérer les collisions entre les drones et l'environnement.

Le programme du drone (qu'il soit simulé ou non), est relié à une station sol à partir de laquelle tous les paramètres (lois de commandes, filtres, etc) sont modifiables à la volée. La station sol permet également de visualiser tous les graphes désirés pour représenter l'état du drone.

Cet environnement de développement permet donc de simuler l'ensemble des capteurs des drones réels ; de tester et débogger les programmes avant les vols réels. Cette approche nous permet un gain de temps important et évite de nombreux accidents.

## 4 Scénarios de simulation

### 4.1 Agrégation des drones

Dans cette partie de l'article, nous présentons une simulation du rassemblement de plusieurs UAVs. La simulation est effectuée en utilisant le simulateur décrit dans la section précédente. Le nombre de drones est  $N = 3$ . La simulation est effectuée dans un espace tridimensionnel. Les positions initiales des drones sont données par la matrice  $P_{init}$  suivante :

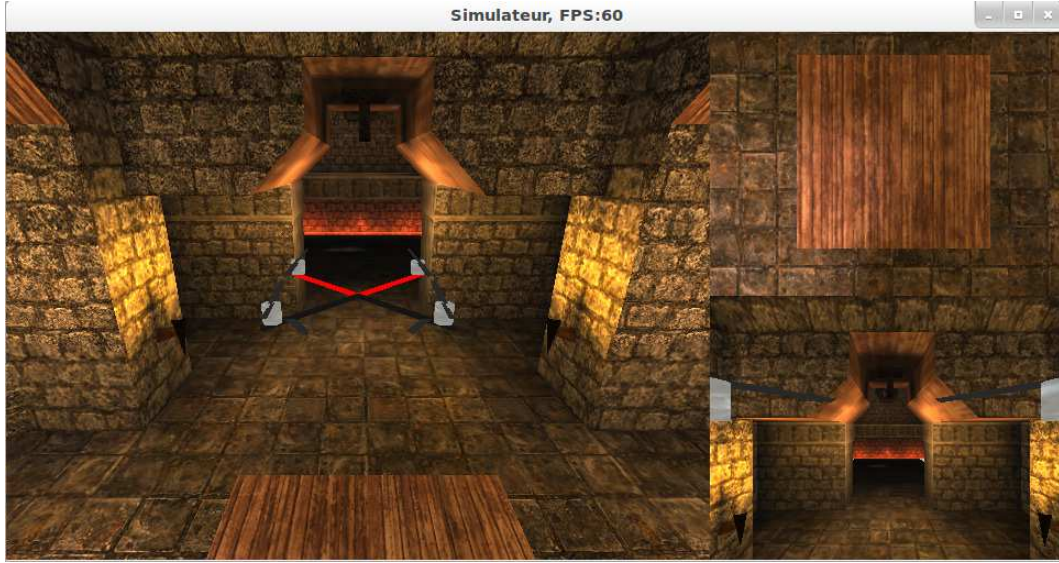


FIGURE 4 – Capture d’écran du simulateur. En haut à droite, vue de la caméra pointant vers le sol. En bas à droite, vue de la caméra pointant vers l’avant.

$$P_{init} = \begin{pmatrix} -3.5 & -1 & -4 \\ -4 & -3 & -1.5 \\ 1 & 1 & 1 \end{pmatrix} \quad (11)$$

où les lignes et les colonnes de cette matrice représentent, respectivement, les positions  $xyz$  et le nombre des drones. Les vitesses initiales des drones sont considérées comme nulles.

Les paramètres de rassemblement sont définis comme suit :  $d = 2$ ,  $c = 7$ . Par conséquent, nous supposons que le champ de vision de chaque UAV couvre l’ensemble de l’espace de simulation. En outre, nous supposons que chaque UAV peut mesurer les états de tous les autres UAVs dans son champ de vision.

La figure 5 montre quatre clichés de simulation exécutés dans le temps. Cette simulation est réalisée en utilisant la méthode de contrôle dans (7), appliquée sur chaque drone indépendamment. Au début, les drones ont des positions dispersées dans l’espace de simulation. Chaque drone effectue une action de répulsion quand il y a des agents dans sa région libre de collision. Cette action garantit des distances de sécurité entre les UAVs. En outre, il active le comportement d’agrégation quand un drone ou plus est dans sa région de voisinage ou dans son champ de vision. Par conséquent, les drones commencent à converger entre eux pour former une flotte. La convergence est réalisée sans risque de collision. Après un temps suffisant, la flotte d’UAVs est formée et stabilisée comme le montre la figure 5d après un léger dépassement visible sur la figure 5c. De plus, nous pouvons voir clairement les distances de sécurité entre les individus. Ces distances de sécurité assurent une formation d’UAVs sans collision.

## 4.2 Navigation des drones

Dans cette partie, nous présentons une simulation de navigation de plusieurs UAVs. Le nombre de drones est  $N = 3$ . Les positions initiales des drones sont données par la matrice  $P_{init}$  suivante :



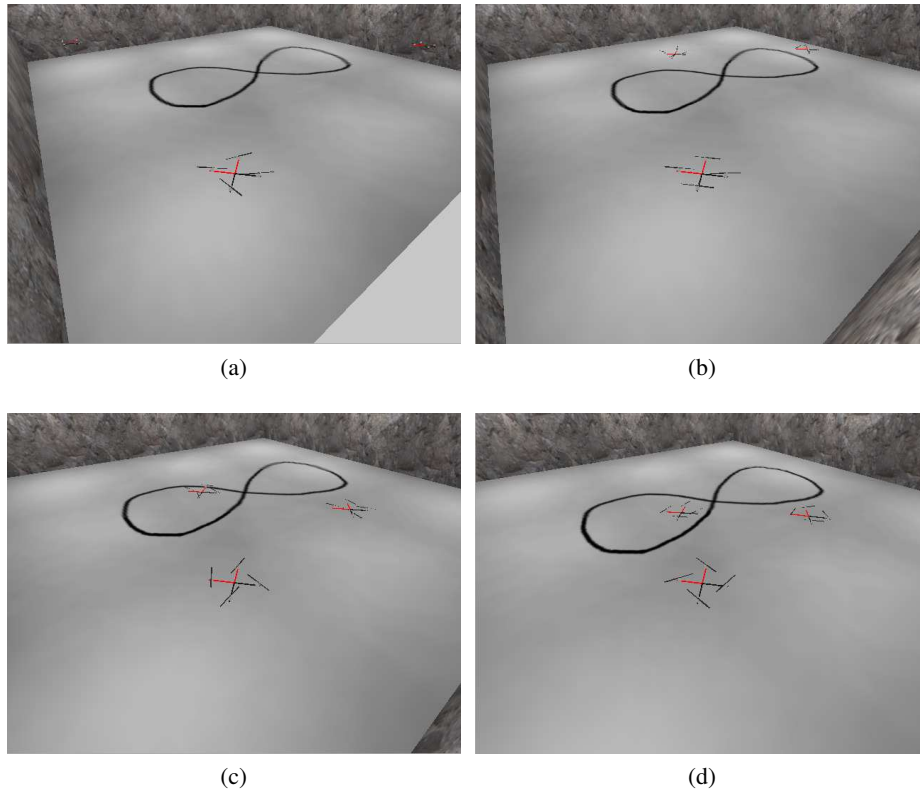


FIGURE 5 – Rassemblement en flotte.

$$P_{init} = \begin{pmatrix} -3.5 & -1 & -4 \\ -4 & -3 & -1.5 \\ 1 & 1 & 1 \end{pmatrix} \quad (12)$$

Les vitesses initiales des drones sont considérées comme nulles. Le vecteur de navigation ou la position finale désirée est  $P_{nav} = [ 3.5 \ 3.5 \ 1 ]^T$ .

Les paramètres de simulation sont les mêmes que celles de rassemblement.

La figure 6 montre quatre clichés de simulation exécutés dans le temps. Cette simulation est réalisée en utilisant la fonction de génération de référence dans (10) appliquée sur chaque drone indépendamment. Au début, les drones ont des positions dispersées dans l'espace de simulation. Chaque drone exécute le comportement d'agrégation pendant la navigation vers la destination souhaitée. Les distances de sécurité sont maintenues tout au long du chemin de navigation. Enfin, la flotte est uniformément formée et stabilisée vers la position de destination.

## 5 Conclusion

Dans cet article, nous avons présenté un simulateur multi-drones. Ce simulateur émule l'exécution des algorithmes de contrôle en temps réel. Nous avons aussi introduit les méthodes de commande d'une flotte de drones. Ces méthodes sont basées sur des algorithmes de commande décentralisés. Différents scénarios d'agrégation et de navigation en flatterie, sont simulés à travers le simulateur multi-drones.

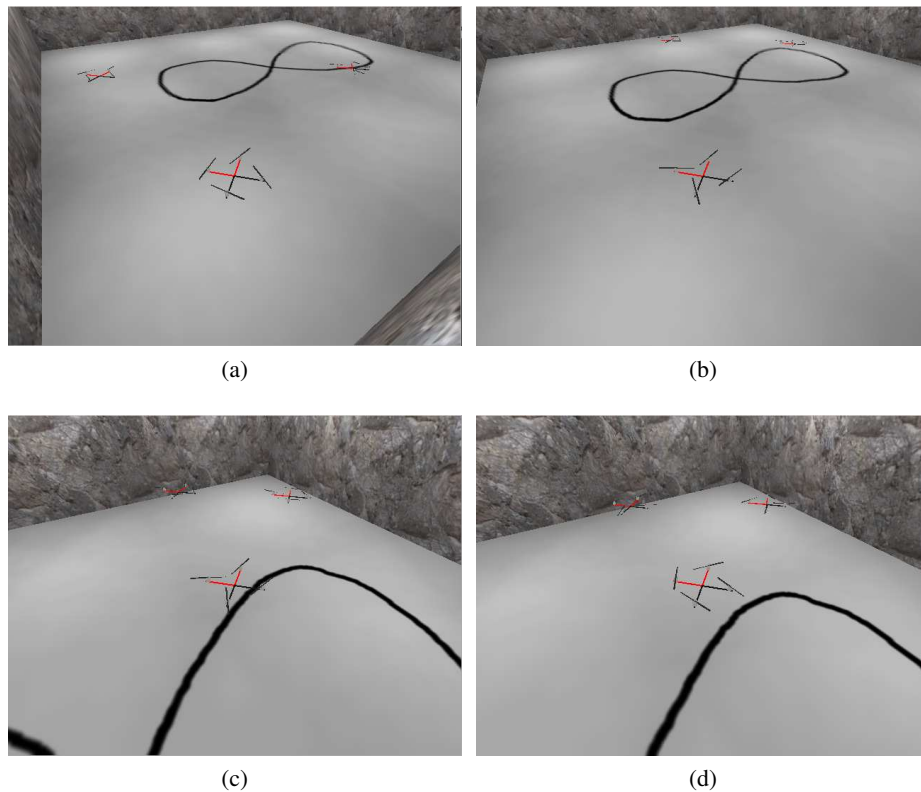


FIGURE 6 – Navigation de flotte.

## 6 Remerciement

Ce travail a été réalisé et financé dans le cadre du Labex MS2T. Il a été soutenu par le gouvernement français, à travers les programmes «Investissements d'avenir» gérés par l'Agence Nationale de la Recherche (Référence ANR-11-IDEX-0004-02).

G. Sanahuja est cofinancé par l'Union européenne. L'Europe s'engage en Picardie avec le Fonds européen de développement régional.

## Références

- [1] Partridge. The structure and function of fish schools. *Scientific American*, 246(06) :114+, 1982.
- [2] Iain D. Couzin. Collective cognition in animal groups. *Trends in Cognitive Sciences*, 13(1) :36 – 43, 2009.
- [3] J.-A. Guerrero and R. Lozano. *UAV Flight formation control*. John Wiley-ISTE, 2012.
- [4] M. Chiamonti, F. Giulietti, and G. Mengali. Formation control laws for autonomous flight vehicles. In *Control and Automation, 2006. MED '06. 14th Mediterranean Conference on*, pages 1 –5, June 2006.
- [5] Craig W. Reynolds. Flocks, herds, and schools : A distributed behavioral model. In *Computer Graphics*, pages 25–34, 1987.
- [6] O. Saif and I. Fantoni. Commande LQR d'une flotte de multiples véhicules aériens. *JD / JNMACS*, Strasbourg, France, September 2013.

- [7] O. Saif and I. Fantoni. Multi-agent flocking by LQR control. *IROS*, Tokyo, Japan, November 2013. Submitted paper.
- [8] Gianluca Antonelli, Filippo Arrichiello, and Stefano Chiaverini. Flocking for multi-robot systems via the null-space-based behavioral control. *Swarm Intelligence*, 4(1) :37–56, 2010.
- [9] Reinhard Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer-Verlag, Heidelberg, third edition, 2005.
- [10] R. Olfati-Saber. Flocking for multi-agent dynamic systems : algorithms and theory. *IEEE Transactions on Automatic Control* , 51(3) :401 – 420, March 2006.
- [11] Reza Olfati-Saber and Richard M. Murray. Distributed cooperative control of multiple vehicle formations using structural potential functions. In *IFAC World Congress*, Barcelona, Spain, July 2002.
- [12] P. Castillo, R. Lozano and A. Dzul, *Modelling and Control of Mini-Flying Machines*, Springer-Verlag in Advances in Industrial Control, juillet 2005.
- [13] G. Sanahuja, P. Castillo and A. Sanchez, *Stabilization of  $n$  integrators in cascade with bounded input with experimental application to a VTOL laboratory system*, International Journal of Robust and Nonlinear Control, 2009, DOI : 10.1002/rnc.1494.