

# Planification de Mission pour un Système de Lancement Aéroporté Autonome

Svetlana DICHEVA, Yasmina BESTAOU<sup>1</sup>

**Résumé :** Dans le cadre d'une coopération avec le CNES (projet PERSEUS), nous nous intéressons à la planification de mission d'un système de lancement aéroporté autonome. L'algorithme utilisé pour cette planification est l'algorithme A\* (A-étoile) en présence d'obstacles et de conditions atmosphériques défavorables. L'algorithme génère une séquence de points de passage, en fonction de l'objectif désiré et des contraintes environnementales. Notre résultat principal est la généralisation de cet algorithme d'un environnement à deux dimensions à un environnement à trois dimensions avec contraintes. La génération des points de passage du chemin optimal déjà trouvé dans les différentes zones de mission sera présentée par l'intermédiaire des résultats de simulation.

**Mots-clés -** Planification de mission, Guidage, Algorithme A\*

## I. INTRODUCTION

Nos études concernent les lanceurs de satellites réutilisables. Dans ce domaine, les recherches aux Etats Unis ont démarré depuis plus d'une décade. Les véhicules de démonstration Pegasus, X-33 et X-34 ont été développés par la NASA. Ces véhicules sont destinés à tester de nouveaux concepts et technologies à étage simple ou étages multiples où chaque étage doit retourner sur Terre en vue d'une réutilisation. En France, la faisabilité d'un concept innovant de lancement aéroporté par véhicule polyvalent est actuellement étudiée au CNES, études auxquelles nous sommes associées dans le cadre du projet PERSEUS (Projet Etudiant de Recherche Spatiale Européen Universitaire et Scientifique). Nous nous intéressons à un système de Lancement Assisté par Aéroporteur Autonome Réutilisable (L3AR). L'objectif de cet article est de présenter la stratégie de mission dynamique pour la gestion de certains aléas en vol atmosphérique des véhicules aérospatiaux autonomes.

Ce nouveau concept étudié par le projet PERSEUS est le suivant : le lanceur n'est pas allumé au sol, à vitesse nulle, mais il est d'abord emporté pendant la première partie de la mission par un porteur indépendant, jusqu'à une certaine altitude à laquelle le lanceur est largué puis allumé. Le porteur constitue en fait le premier étage du système de lancement et communique au lanceur une certaine énergie. A masse totale égale, un lanceur aéroporté permettrait un gain en performance par rapport à un lanceur partant du sol à vitesse nulle. De plus, le porteur serait réutilisable, ce qui permettrait une réduction des coûts de lancement. La séparation du lanceur de satellites de son porteur est supposé s'effectuer dès l'atteinte des conditions ad-hoc (altitude et Mach) et au plus tard après le parcours d'une distance de 50 km par rapport au lieu de décollage.

La mission de séparation consiste à amener le lanceur à une altitude de 16000m, à une vitesse de Mach 0.8, porteur-lanceur stabilisé et porteur à pente nulle afin d'injecter le lanceur sur sa trajectoire optimale. Le largage doit s'effectuer de façon à assurer des trajectoires divergentes entre les deux véhicules, le porteur devant s'éloigner le plus rapidement possible du lanceur. A l'instant de la séparation, le véhicule porteur est un siège de phénomènes transitoires très importants du fait de la perte de masse qui peut atteindre le tiers de la masse au décollage. C'est une difficulté majeure dans le lancement aéroporté.

Les phases de descente et d'attente sont supposées s'effectuer à proximité de l'aérodrome d'atterrissage mais correspondent néanmoins à une distance de parcours respectable. La phase d'attente se caractérise par un délai de ¼ d'heure à une altitude de 3000m. La descente est parcourue à une vitesse définie et limitée en Mach [1]. Cette dernière phase sera réalisée de façon à augmenter la distance parcourue au retour. La mission L3AR se caractérise ainsi suivant les segments identifiés sur le schéma suivant

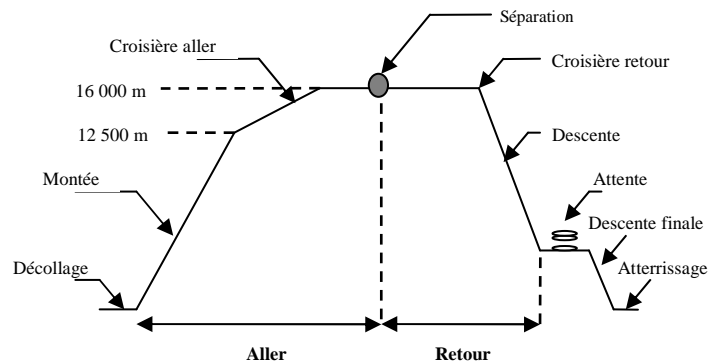


Fig. 1 Phases de Mission L3AR

Le système de lancement réutilisable qui représente une structure complexe bi-étages composée d'un porteur et d'un lanceur possède plusieurs niveaux d'interaction avec les systèmes de pilotage automatique. L'objectif de la planification de mission est de chercher le plan de vol le plus approprié pour que le système de lancement puisse atteindre les objectifs de sa mission (la séparation des deux étages) et éviter les obstacles. Plusieurs contraintes doivent être satisfaites : le temps nécessaire pour accomplir la mission, la consommation de carburant, l'évitement de certains aléas (zones à conditions atmosphérique défavorables), et les normes de la circulation aérienne.

L'intelligence embarquée permet à un véhicule non-habité d'atteindre les objectifs de sa mission en assurant sa survie et en prenant en compte les aléas qui surviennent en cours de la mission. Cette autonomie décisionnelle est nécessaire dès que le véhicule évolue dans un environnement mal connu et

<sup>1</sup>Laboratoire IBISC, Université d'Evry Val d'Essonne, 38, rue du Pelvoux 91020 EVRY, France [dicheva.bestouai@iup.univ-evry.fr](mailto:dicheva.bestouai@iup.univ-evry.fr)

dynamique. Nous nous sommes inspirées des travaux de planification en robotique pour calculer des plans de vol dynamiques. Dans la littérature, la planification de chemin est étudiée le plus souvent à l'aide des algorithmes Dijkstra et A\*(A-étoile) [2], Voronoï [3] ainsi que ceux basés sur la programmation orientée objet [4]. L'algorithme de Dijkstra évalue le coût du mouvement d'un nœud vers n'importe quel nœud suivant. Il détermine le coût du chemin optimal le plus court comme étant le coût de connexion le plus petit entre deux nœuds consécutifs qui se trouvent sur l'itinéraire. Une autre recherche appelée « Best-First-Search » estime la distance entre la position courante du système de lancement et le point final. Pour répondre à toutes ces exigences, l'algorithme A\* utilise les avantages de l'algorithme de Dijkstra et du « Best-First-Search ». L'algorithme A\* ne trouve pas uniquement le chemin le plus court pour chaque configuration de la mission, mais choisit également la meilleure solution initiale pour initialiser l'algorithme. Cette solution est basée sur le choix du nœud initial qui doit être sur la droite joignant la source au point final.

Dans cet article, une configuration s'appelle une phase de mission ou une partie d'une phase où le chemin optimal pourrait avoir des points de passage avec des paramètres similaires. Nous développons une approche de planification de mission qui permet la génération de points de passage pour les différentes zones de mission (basé sur l'algorithme A\*). La complexité de la mission s'exprime dans la génération des points de passage, leur priorité et leurs paramètres qui influencent tous les niveaux du système de guidage, de navigation et de contrôle. L'objectif est de trouver une séquence de configurations permettant au système L3AR de réaliser les phases de la mission en présence des obstacles connus, les critères et les contraintes devant être définis.

La séquence de point de passage est définie dans un espace 3D, ce qui exige une amélioration de l'algorithme classique A\* prévu pour un système évoluant dans un espace plan. Les résultats sont développés à l'aide du logiciel Matlab® avec un modèle atmosphérique dynamique.

Cette communication est organisée comme suit : le paragraphe II présente le système L3AR. La stratégie de mission dynamique qui traite les problèmes de gestion des aléas en vol est présentée dans le paragraphe III. L'algorithme et son application dans l'architecture du système est présenté dans le paragraphe IV. Enfin, le paragraphe V propose quelques conclusions et perspectives de recherche.

### III. SYSTEME L3AR

Les équations de translation d'un véhicule aérien peuvent être dérivées des lois de Newton-Euler. Si sa vitesse est inférieure à Mach 5, la Terre peut être considérée comme étant un repère de référence inertiel [5].

Les équations cinématiques du porteur sont données par :

$$\begin{aligned}\dot{x} &= V \cos \chi \cos \gamma + W_x \\ \dot{y} &= V \sin \chi \cos \gamma + W_y \\ \dot{z} &= V \sin \gamma + W_z\end{aligned}\quad (1)$$

Où  $W = (W_x \ W_y \ W_z)^T$  représente la vitesse du vent exprimée dans le repère inertiel.

Les équations dynamiques sont données par les relations suivantes :

$$\begin{aligned}\dot{V} &= -\frac{C_D(M, \alpha) A_{ref} \rho V^2}{2m} - g \sin \gamma + \frac{T \cos \alpha}{m} \\ &\quad - \dot{W}_x \cos \gamma \cos \chi - \dot{W}_y \cos \gamma \sin \chi - \dot{W}_z \sin \gamma \\ \dot{\gamma} &= \frac{C_L(M, \alpha) A_{ref} \rho V \cos \sigma}{2m} - \frac{g \cos \gamma}{V} + \frac{T \sin \alpha \cos \sigma}{mV} \\ &\quad + \frac{\dot{W}_x \sin \gamma \cos \chi}{V} - \frac{\dot{W}_y \sin \gamma \sin \chi}{V} - \frac{\dot{W}_z \cos \gamma}{V} \\ \dot{\chi} &= \frac{C_L(M, \alpha) A_{ref} \rho V \sin \sigma}{2m} + \frac{T \sin \alpha \sin \sigma}{mV \cos \gamma} + \frac{\dot{W}_x \sin \chi}{V \cos \gamma} - \frac{\dot{W}_y \cos \chi}{V \cos \gamma}\end{aligned}\quad (2)$$

Où x, y et z sont les coordonnées cartésiennes de la position du véhicule,  $\gamma$  représente l'angle de pente,  $\chi$  le cap,  $\sigma$  l'angle de vectorisation, V la vitesse,  $\rho$  la densité massique, m représente la masse du porteur,  $A_{ref}$  sa surface

caractéristique,  $C_L, C_D$  sont respectivement les coefficients de portance et de trainée dépendant du nombre de Mach M et de l'angle d'incidence  $\alpha$  (voir fig. 2).

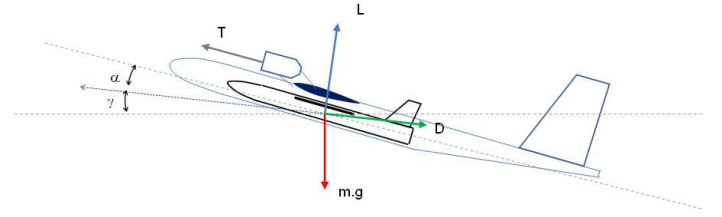


Fig. 2 Système L3AR

Une stratégie de mission dynamique est appliquée sur le système L3AR. Cette stratégie décrit les actions nécessaires pour qu'il atteigne les objectifs de sa mission en évoluant dans une région en présence d'obstacles.

### III. STRATEGIE DE MISSION DYNAMIQUE : GESTION DES ALEAS EN VOL

Une mission est effectuée en réalisant des actions [6] : des actions de mouvements, des actions sur l'environnement, des informations... Une mission a un début et une fin entre lesquelles des actions sont réalisées. L'objet de cette mission est le largage du lanceur de satellites et les objectifs sont toutes les configurations que doit respecter le plan de vol. Le but de la stratégie de mission est de sélectionner les objectifs et de trouver la manière de les réaliser en tenant compte de l'environnement. Parmi les solutions possibles, le planificateur doit choisir celle qui optimise un critère respectant certaines contraintes non-linéaires.

Le problème pourrait se baser sur la décomposition en deux niveaux : le haut niveau qui correspond à la réalisation des objectifs de la mission (atteindre les points de passage nécessaires pour la réalisation de largage), et le bas niveau qui décrit cette réalisation en fonction des contraintes. Les incertitudes existent toujours dans le système. Pour les traiter correctement une réactualisation périodique des paramètres utilisés pour atteindre les objectifs est envisagé. Cet article s'intéresse au niveau le plus haut.

En intelligence artificielle, un algorithme statique est utilisé hors ligne afin de produire un ou plusieurs plans réalisables. Un algorithme dynamique est ensuite utilisé en ligne pour construire progressivement la solution adaptée aux aléas qui surviennent. Les termes statique et dynamique caractérisent l'environnement dans lequel le plan est exécuté. La planification classique suppose un environnement statique, ce qui signifie qu'il n'y a aucune incertitude. Un algorithme prédictif est alors utilisé hors ligne pour produire un plan unique qui peut être alors exécuté en ligne sans remises en cause. Cet algorithme pourrait être A\* qui cherche le chemin optimal le plus court pour atteindre les points significatifs pour la réalisation de la mission. L'avantage est qu'il est souvent utilisé dans les environnements où différents obstacles existent.

Dans le cas d'un environnement dynamique, plusieurs techniques sont envisageables [7]-[8] :

- conserver une approche prédictive hors ligne, complétée par un ou plusieurs algorithmes réactifs exécutés en ligne lorsqu'un aléa rend le plan initial incohérent, remettant celui-ci en cause et obligeant le plus souvent à re-planifier
- prendre en compte les incertitudes dès la phase de construction hors ligne : on parle alors d'approches proactives ; dans le cas idéal le plan ainsi produit s'adaptera automatiquement en ligne aux aléas qui se présenteront.
- Planifier toujours de manière prédictive mais cette fois-ci en ligne, à court terme, selon un processus d'horizon fuyant ; dans ce cas, l'exécution va progressivement lever certaines incertitudes et permettre de poursuivre la planification des étapes suivantes.

Dans [9], cinq types d'environnement sont analysés :

1. Obstacles statiques à propriétés déjà connues.
2. Obstacles, conflits et menaces non-adverses à paramètres déjà connus.
3. Obstacles, conflits et menaces non-adverses à paramètres inconnus.
4. idem que le cas 3) avec des contraintes détectées à proximité de l'avion.
5. idem que le cas 4) avec menaces dangereuses.

La mission que nous étudions correspond au deuxième type d'environnement. Nous utilisons l'algorithme A\* en ligne de manière périodique avec une période supérieure à celle des microcontrôleurs utilisés par le pilotage automatique, en réactualisant grâce aux mesures effectuées en temps réel.

Le plan de vol s'intéresse aux variables suivantes:

1. Les paramètres du point vers lequel le véhicule se dirige,
2. Le temps nécessaire pour atteindre le point de passage,
3. Le niveau des ressources : carburant,
4. L'état de réalisation pour chaque objectif,
5. L'état des paramètres pour le largage,
  - Altitude constante ( $H=0$ )
  - Pente nulle ( $\gamma=0$ )
  - Vitesse constante, Accélération nulle
6. L'état du système après largage,
7. Le modèle dynamique du véhicule ne permet que des valeurs limitées de sa vitesse ( $V_{min}$  et  $V_{max}$ ) et de temps

nécessaire pour réaliser des actions pour chaque configuration.

#### IV. ALGORITHME DE PLANIFICATION A\* EN 3D

L'algorithme A\* génère la séquence des points de passage en minimisant la distance totale parcourue. Le plan est composé de plusieurs points de passage en 3D qui définissent le passage par les différentes zones de mission. A l'initialisation, l'algorithme est généré hors-ligne, ensuite le plan de vol est actualisé d'une manière périodique. A\* cherche les solutions optimales dans un champ d'action qui est séparé par des cases, pouvant être appelés cellules.

##### 4.1. Conception de l'algorithme A\*

L'algorithme A\* est basé sur une méthode heuristique classique qui utilise la position initiale et finale pour trouver le chemin optimal. Il effectue une recherche dans le champ d'action et génère les points de passage pour le chemin optimal. Dans l'algorithme classique, chaque pas au long du chemin optimal est trouvé à l'aide de la formule:

$$F(n) = G(n) + H(n) \quad (3)$$

Où  $F(n)$  évalue le coût total du chemin (entre la position initiale et finale),  $F(n)$  se compose de deux variables :  $G(n)$  et  $H(n)$ .  $G(n)$  est la distance actuelle entre la position initiale et la position courante du système et  $H(n)$  est le coût estimé entre la position courante et la position finale. Le nœud avec le coût total le plus petit est choisi comme nœud suivant, qui est le point de passage suivant pour le chemin optimal. Le fonctionnement de l'algorithme exige la formation de deux listes d'exécution pour réaliser la recherche dans la région d'action :

- La première s'appelle "Liste Ouverte", elle contient toutes les cellules qui sont en cours d'évaluation et l'information pour le nœud précédent. A la fin l'algorithme A\* calcule la somme des données pour ces nœuds parentaux et dessine le résultat final. Pour cette raison A\* est bidirectionnel parce qu'une fois il examine le chemin vers l'objectif et qu'une autre fois il revient vers le point initial pour définir le chemin optimal.
- La deuxième s'appelle "Liste Fermée", elle contient des cellules qui sont déjà évaluées, mais aussi l'information pour le nœud initial et les obstacles.

La différence entre l'application en 2D et en 3D est qu'en 3D le nombre de cellules qui seront examinées augmente avec chaque pas de déplacement. Dans le plan 2D il y a seulement des mouvements horizontaux, alors qu'en 3D il y a également des mouvements verticaux. De cette manière les mouvements possibles vers les nœuds successifs augmentent de huit à vingt-six. Il y a donc plus de solutions possibles. Les cellules qui sont rejetées et ne sont pas des solutions sont classées dans la liste Fermée. Nous nous sommes inspirées de la référence [10], où les auteurs s'intéressent au problème de routage des mètres.

L'approche de A\* en 3D commence avec l'initialisation de la région d'action dans un format que le logiciel Matlab® peut reconnaître. Les zones « interdites », qui sont les obstacles comme par exemple les régions avec des conditions

atmosphériques défavorables, sont initialisées. L'algorithme de programmation est présenté ci-dessous:

**Initialisation:**

- Initialiser les paramètres des objectifs
- Initialiser les coordonnées des obstacles
- Mettre** tous les obstacles dans la liste Fermée
- Initialiser la position initiale du système L3AR ( $n_0$ )
- Mettre**  $n_1$  (le premier nœud à analyser) dans la liste Ouverte

**Traitement:**

- Tant que** (la liste Ouverte n'est pas vide et le point final n'est pas atteint) **faire**
- Pour** chaque  $n$  (du groupe des nœuds successifs) **faire** Calculer le coût total  $F(n)$  pour chaque nœud
- Pour** chaque déplacement possible **faire**
- Calculer  $H(n)$  (la distance estimée entre la position actuelle et la position finale)
- Construire le chemin entre  $n_1$  et  $n_f$  - le point final (Générer les points de passage satisfaisant les contraintes de distance)
- Mettre**  $n$  dans la liste Fermée
- Mettre** le point suivant dans la liste Ouverte

**End**

4.2. Introduction de contraintes dans l'algorithme A\*

On introduit les points de passage qui représentent une partie du processus de navigation [11]. Cet algorithme donne la possibilité de trouver le chemin réalisable en utilisant seulement un registre avec la collection des points de passage plutôt que d'essayer de chercher dans tout l'environnement.

Les références de position, vitesse et orientation du véhicule (guidage et navigation) sont envoyées au système de pilotage qui agit directement sur les actionneurs (ailerons, gouvernes, volets) et les actionneurs sur la dynamique du véhicule. Ces données sont comparées avec les données réelles qui viennent des capteurs. De cette manière on ferme la boucle de guidage classique présentée en fig.3. L'algorithme A\* en 3D, réactualisé à un intervalle  $\lambda$ , présente une connexion originale entre la navigation et le guidage. L'algorithme envoie les nouvelles références nécessaires devant être suivies par le système L3AR.

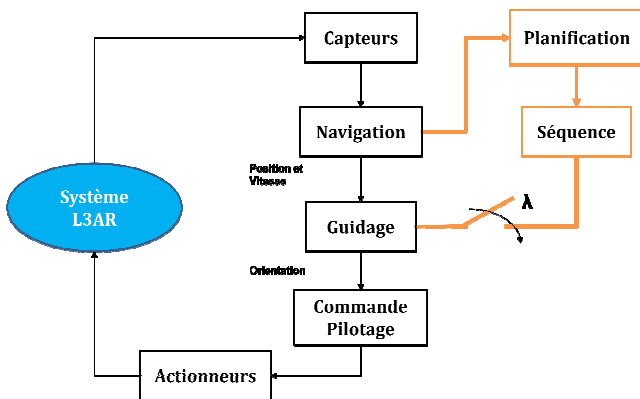


Fig.3 Système L3AR

Chaque fois que le système de lancement atteint un objectif, ce point de passage est supprimé de la liste avec les tâches courantes et continue de se diriger vers le point suivant de la liste. L'algorithme de planification de séquence est généré dans le fichier avec l'initialisation de tous les paramètres et conditions initiales du modèle. Les paramètres de chaque point de passage peuvent varier au cours de la mission. L'environnement est supposé dynamique car les paramètres (pression, température, relief) peuvent changer mais aussi les conditions atmosphériques et il faut éviter l'abordage d'autres véhicules aériens. Si en réaction à ces changements, un nouveau point est déterminé et est introduit dans la liste d'exécution des tâches cela imposera un nouveau calcul des paramètres des points de passage suivants.

Ces points de passage permettent de passer en toute sécurité d'une configuration à une autre lorsqu'il se produit d'importants changements dans la configuration dus aux conditions externes. Ces séquences peuvent être interrompues à tout moment si une configuration finale plus intéressante est spécifiée par le système. Dans ce cas, les trajectoires de reconfiguration doivent aussi être calculées.

Il s'agit maintenant de transformer cette séquence de points de passage en 3D en un chemin que pourra suivre le porteur. L'approche suivie comporte deux étapes. Tout d'abord, nous nous intéressons au modèle cinématique en proposant des chemins à courbure et torsion continues et bornées. Ensuite en utilisant le modèle dynamique, nous introduirons les bornes sur la vitesse, l'accélération et les dérivées angulaires. Nous ne nous penchons pas sur le problème de pilotage qui consiste à transformer des consignes de références en ordres pour les gouvernes de direction, les volets et les ailerons.

4.3. Résultats de simulation

Les résultats de simulation sont présentés pour illustrer l'application de A\* en 3D. Les dimensions de l'espace géométrique de recherche sont la longueur L, la largeur D et la hauteur H sont respectivement égaux à 10 cellules.

La position initiale du système de lancement est définie par  $P_{L3AR} = \{x_{L3AR}, y_{L3AR}, z_{L3AR}\}$ . Les coordonnées des objectifs sont  $P_{i1} = \{x_{i1}, y_{i1}, z_{i1}\}$ ,  $i=1...n$  ( $n$ -le nombre d'objectifs). Les coordonnées des obstacles sont  $P_{oj} = \{x_{oj}, y_{oj}, z_{oj}\}$ ,  $j=1...m$  ( $m$ -le nombre d'obstacles).

Dans le maillage utilisé en 3D les paramètres de la position initiales sont introduits  $P_{L3AR} = \{2, 2, 2\}$  ainsi que la position de l'objectif final  $P_{i1} = \{9, 9, 9\}$ ;  $n=1$  et il y a seulement un objectif. Les obstacles sont initialisés.  $m=4$  pour les quatre obstacles  $P_{o1} = \{8, 8, 8\}$ ,  $P_{o2} = \{6, 6, 5\}$ ,  $P_{o3} = \{5, 6, 6\}$ ,  $P_{o4} = \{6, 5, 6\}$ . Le résultat du chemin optimal trouvé en présence des quatre obstacles (petits par rapport aux dimensions du système L3AR) définis par leurs coordonnées est présenté sur la figure suivante :

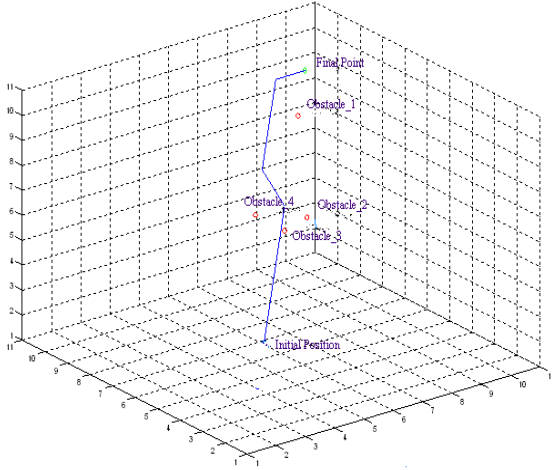


Fig.4 Chemin optimal le plus court en présence de quatre obstacles stationnaires à éviter

Sur la fig.5 l'environnement est spécifié avec les mêmes obstacles sauf qu'un nouvel obstacle est introduit dont les coordonnées initiales sont  $P_{o1}=\{5, 6, 7\}$  et avec une longueur, une largeur et une hauteur dimensionnées dans le plan 3D.

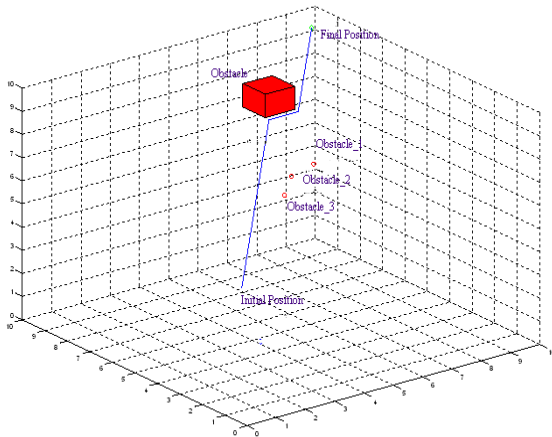


Fig.5 Chemin optimal le plus court en présence de quatre obstacles stationnaires, dont un est un cube à éviter

Dans la dernière simulation deux objectifs sont initialisés en même temps. Leur priorité est connue d'avance. L'algorithme A\* trouve le chemin optimal le plus court jusqu'au premier objectif, de cette manière le point final de la première configuration devient le point initial pour la deuxième configuration, pour atteindre le deuxième objectif. En plus des obstacles définis précédemment, un obstacle à forme sphérique, de dimension non négligeable par rapport aux dimensions du système L3AR, est ajouté. Celui-ci est défini par son centre  $P_{o1}=\{6, 6, 5\}$  et son rayon  $r=0.5$ .

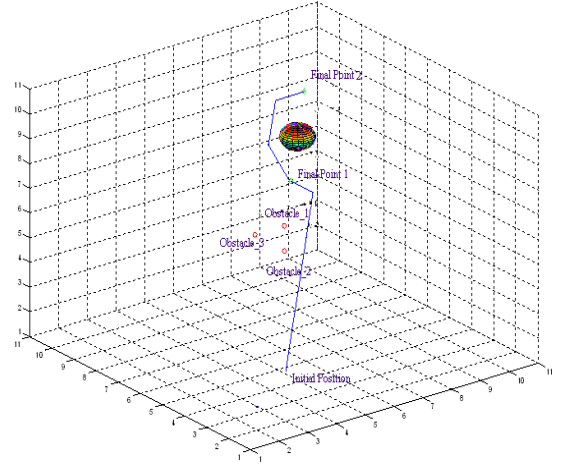


Fig.6 Chemin optimal le plus court en présence de deux objectifs à atteindre et différents obstacles stationnaires à éviter

#### 4.4. Transformation d'un chemin en lignes brisées en un chemin à courbure continue

La construction de la trajectoire se base essentiellement sur les courbes de Dubins [12]. La trajectoire se compose alors d'une succession de segments de droite et d'arcs de cercles (fig. 7).

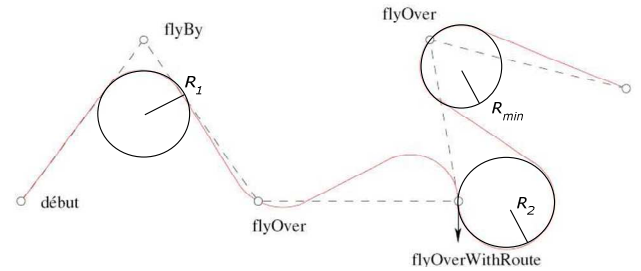


Fig.7 Trajectoire avec les trois différents types de point de passage

Les courbes de Dubins proposent une solution simple et efficace pour transformer le chemin brut déjà trouvé par A\* en une trajectoire réalisable par le système L3AR. Les manœuvres sont facilitées en utilisant trois types de points de passage spécifiques : 'flyBy', 'flyOver' et 'FlyOverWithRoute' qui définissent le rayon optimal pour que le système puisse effectuer le passage par les nœuds.

#### IV. CONCLUSION

Nous avons proposé une méthode pour la planification de mission d'un système de lancement L3AR basé sur la génération des points de passage avec l'algorithme A\* en 3D en satisfaisant les critères nécessaires pour une mission réalisée dans une région limitée comportant des obstacles connus. Les points de passage sont générés avant le début de la mission (hors ligne) avec une réactualisation de paramètres en cours de déroulement. Les résultats de simulation sont présentés en calculant le chemin optimal le plus court en présence de différents obstacles et plusieurs objectifs. Une solution de transformation du chemin brut dans une trajectoire réalisable par le système est également proposée.

Nous avons ainsi utilisé la géométrie puis la cinématique. Une extension évidente de ce travail est d'utiliser également la dynamique du système L3AR, en considérant les contraintes sur la vitesse, la poussée et les dérivées à l'ordre deux de toutes les variables considérées.

#### REFERENCES

- [1] N. Bérend, M. Bourgaie, S. defoort, J. Hermetz, P. Schmollgruber 'Lancement aéroporté par véhicule polyvalent' ONERA Rapport technique, Sept. 2008
- [2] W. Oomkens, M. Mulder, M. M. Van Paassen, M. H. J. Amelink, 'UAVs as Aviators: Environment Skills Capability For UAVs', IEEE Int. Conf. on Systems, Man & Cybernetics, 2008, pp. 2426-2431
- [3] D. Eppstein. "Finding the k shortest paths," 35<sup>th</sup> Annual Symp. On Computer Science, 1994, pp. 154-165
- [4] S. Gualandi, B. Tranchero. "Concurrent constraint programming-based path planning for uninhabited air vehicles," Proc. of SPIE, Sept. 2004
- [5] L. Mangiacasale "Flight Mechanics of a  $\mu$ -Airplane with a MATLAB Simulink Helper", Edizioni Libreria CLUP, Milan, 1998.
- [6] E. Chanthery 'Planification de mission pour un véhicule autonome' Thèse de doctorat, LAAS, Université de Toulouse, 2006
- [7] Y. Bestaoui, S. Dicheva 'Lois de Commande d'un système L3A – Analyse de l'existant' Rapport de contrat de collaboration n° 2008/82431
- [8] Y. Bestaoui, S. Dicheva '3D flight plan for an autonomous aircraft' 48<sup>th</sup> AIAA Aerospace Sciences Meeting, Jan. 2010, Orlando, FL, paper AIAA 2010-415
- [9] H. Yang, Y. J. Zhao 'Trajectory Planning for Autonomous Aerospace Vehicles amid Known Obstacles and Conflicts' Journal of Guidance, Control, and Dynamics, Vol. 27, No. 6, 2004, pp. 997-1008
- [10] L. Niu, G. Zhuo 'An Improved Real 3D A\* Algorithm for Difficult Path Finding Situation', The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVII. Part B4. Beijing 2008
- [11] D. Mulvaney, Y. Wang, I. Sillitoe 'Waypoint-based mobile robot navigation', 6<sup>th</sup> World Congress on Intelligent Control and Automation, June 21-23, Dalian, China, pp. 9063-9067
- [12] Y. Bestaoui, H. Dahmani, K. Belharet 'Geometry of translational trajectories for an autonomous Aerospace vehicle with wind effect' AIAA Aerospace Sciences Meeting, Orlando, FL, Jan. 2009, paper AIAA-2009-417