

TABLE DES MATIÈRES

	Page
INTRODUCTION	1
CHAPITRE 1 REVUE DE BIBLIOGRAPHIE.....	5
1.1 Génération et suivi de trajectoire en 4 dimensions.	5
1.2 L'évitement.....	7
1.3 La méta-heuristique	8
CHAPITRE 2 LES ALGORITHMES PRINCIPAUX	10
2.1 Le système d'évitement des NFZ.	10
2.1.1 Le principe de travail.	10
2.1.2 L'évitement de zone.....	10
2.1.3 L'optimisation de l'évitement de zone	13
2.2 Détermination du chemin le plus court.....	15
2.2.1 La méthode de la descente	15
2.2.2 La méthode de recherche par tabou	16
2.2.3 Principe retenu	17
2.2.4 Les limites de la méthode	21
CHAPITRE 3 ÉTUDE EN DEUX DIMENSIONS.....	22
3.1 Hypothèses.....	22
3.2 Premier cas.....	23
3.2.1 Mise en place	23
3.2.2 Fonctionnement de l'algorithme de résolution	23
3.3 Second cas.....	29
3.3.1 Mise en Place	29
3.3.2 Résolution	29
CHAPITRE 4 ÉTUDE EN TROIS DIMENSIONS	33
4.1 Hypothèses.....	33
4.2 Préambules.....	34
4.2.1 Préambule relatif à l'évitement des NFZ	34
4.2.2 Préambule relatif au choix du plus court chemin.....	34
4.3 Premier cas.....	35
4.3.1 Mise en place	35
4.3.2 Fonctionnement de l'algorithme de résolution	35
4.4 Second cas.....	39
4.4.1 Mise en place	39
4.4.2 Résolution	39

CHAPITRE 5	ÉTUDE EN QUATRE DIMENSIONS.....	42
5.1	Hypothèses.....	42
5.2	Prise en compte de la vitesse.....	43
5.2.1	Préambule.....	43
5.2.2	Méthode de fonctionnement.....	43
5.2.3	Méthode de calcul.....	44
5.3	Premier cas.....	47
5.3.1	Mise en place.....	47
5.3.2	Méthode de résolution.....	48
5.4	Second cas.....	51
5.4.1	Mise en place.....	51
5.4.2	Résolution.....	51
CHAPITRE 6	LES AIDES AU PILOTAGE ET LES FONCTIONS SUPPLÉMENTAIRES.....	53
6.1	Préambule.....	53
6.2	Les fonctions supplémentaires.....	54
6.2.1	Le calcul de la distance parcourue.....	54
6.2.2	Le calcul du temps de vol en tous points.....	54
6.2.3	Estimation de l'heure d'arrivée.....	54
6.2.4	La consommation du carburant.....	56
6.2.5	Estimation de la variation de masse.....	56
6.2.6	Calcul de la poussée nécessaire.....	57
6.3	Validation des fonctions créées.....	58
CHAPITRE 7	VALIDATION DES TRAVAUX.....	62
7.1	Méthode de validation.....	62
7.2	Résultats et validation.....	63
CONCLUSION.....		67
RECOMMANDATIONS.....		71
LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES.....		72

LISTE DES TABLEAUX

	Page
Tableau 2.1	Temps de réponse de la méthode en fonction du nombre de points.....21
Tableau 3.1	Sélection de la trajectoire pour le cas 2 en 2D31
Tableau 4.1	Sélection de la trajectoire pour le cas 1 dans le plan horizontal.....36
Tableau 4.2	Sélection de la trajectoire pour le cas 1 dans le plan vertical.....37
Tableau 4.3	Sélection de la trajectoire pour le cas 2 dans le plan horizontal.....39
Tableau 4.4	Sélection de la trajectoire pour le cas 2 dans le plan vertical.....40
Tableau 5.1	La phase d'accélération dans le premier cas.....49
Tableau 5.2	La phase à vitesse constante dans le premier cas49
Tableau 5.3	La phase de décélération dans le premier cas.....50
Tableau 5.4	Tableau de la vitesse en fonction de la position.....51
Tableau 6.1	Extrait de la matrice des données de vols recueillies automatiquement60
Tableau 7.1	Résultats obtenus pour la variation de poids, de la poussée et de la consommation de carburant.....65

LISTE DES FIGURES

	Page
Figure 2.1	Trajectoire d'évitement de la NFZ passant par 2 points A et B.....12
Figure 2.2	Représentation d'un waypoint dans une NFZ.....13
Figure 2.3	Trajectoire optimisée d'évitement de NFZ entre les points A et B.....15
Figure 2.4	Schéma de l'algorithme de la nouvelle méthode pour le calcul de la trajectoire optimisée.18
Figure 2.5	Le plus court chemin trouve par la combinaison de la méthode tabou et de la descente.19
Figure 3.1	Première étape de l'algorithme pour le cas 1: Évitement des NFZ en 2D.....25
Figure 3.2	Deuxième étape de l'algorithme pour le cas 1: Création de la chaîne des waypoints en 2D.....26
Figure 3.3	Troisième étape de l'algorithme pour le cas 1: Détection du chemin le plus court et viable en 2D.27
Figure 3.4	La trajectoire finale retenue dans le premier cas analysé.....28
Figure 3.5	Premières étapes de l'algorithme pour le cas 2 : évitements des NFZ en 3D.....29
Figure 3.6	Deuxième étape de l'algorithme pour le cas 2: Création de la chaîne des waypoints en 3D.....30
Figure 3.7	Troisième étape de l'algorithme pour le cas 2: Détection du chemin le plus court en 3D.31
Figure 3.8	La trajectoire finale retenue dans le deuxième cas analysé.....32
Figure 4.1	La trajectoire retenue dans le cas 1 suite à l'évitement des NFZ.....37
Figure 4.2	La trajectoire retenue dans le cas 1 suite à l'évitement des NFZ dans le plan de la latitude versus la longitude (gauche) et de l'altitude versus la longitude (droite).38
Figure 4.3	La trajectoire retenue dans le cas 2 suite à l'évitement des NFZ.....40

Figure 4.4	La trajectoire retenue dans le cas 2 suite à l'évitement des NFZ dans le plan de la latitude versus la longitude (gauche) et de l'altitude versus la longitude (droite).....	41
Figure 5.1	Schéma du calcul de la vitesse lors d'un vol accéléré.	46
Figure 5.2	Schéma du calcul de la vitesse lors d'un vol décéléré.	47
Figure 5.3	Trajectoire finale calculée dans le premier cas, en 4D.....	48
Figure 5.4	Trajectoire finale après l'évitement de zone dans le 2nd cas.....	51
Figure 6.1	Explication du visuel du système d'alarme RTA ERROR.....	55
Figure 6.2	Forces appliquées à l'avion en descente.	57
Figure 6.3	Forces appliquées à l'avion en montée.	57
Figure 7.1	Trajectoire finale dans le plan de l'altitude versus la longitude du vol.....	63
Figure 7.2	Le vol de validation dans le plan de la latitude versus la longitude.....	63

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

2D	Deux Dimensions
3D	Trois Dimensions
4D	Quatre Dimensions
ATC	Air Traffic Control
CI	Cost Index
ESTACA	Ecole Supérieure des Techniques Aéronautiques et de la Construction Automobile
ETA	Estimated Time Arrival
ETS	Ecole de Technologie Supérieure
FAA	Federal Aviation Administration
FL	Flight Level, niveau de vol
FMS	Flight Management System
FREER	Free Route Experimental Encounter Resolution
GERAD	Groupe d'Études et de Recherche en Analyse des Décisions
LARCASE	Laboratoire de Recherche en commande active, avionique et aéroserveoélasticité.
IATA	Association Internationale du Transport Aérien
NFZ	No-Fly Zones
RTA	Required Time Arrival

LISTE DES SYMBOLES ET UNITÉS DE MESURE

$x_{\text{centre_NFZ}}$	Abscisse du centre d'une NFZ
$z_{\text{centre_NFZ}}$	Altitude du centre d'une NFZ
C_x	Coefficient de trainee
h	Heure
J	Joule
K	Kelvin
Km	Kilomètre
Mol	Mole
r	Rayon d'une NFZ
R	Constante des Gaz Parfaits
s	Seconde
V_x	Vitesse au point X
$y_{\text{centre_NFZ}}$	Ordonnée du centre d'une NFZ

INTRODUCTION

Au cours du temps, le trafic aérien ne fait qu'augmenter et se densifier. De ce fait, le contrôle aérien est une tâche de plus en plus critique qui demande précision et rapidité d'action. En décembre 2009, malgré la crise économique qui a touché le monde entier, l'Association Internationale du Transport Aérien (IATA) annonçait que le transport de passagers et de fret augmentait encore par rapport à la même période en 2008. La progression du transport de passagers était de 2.1% et celle du transport de fret était de 9.5% [1].

A l'heure actuelle, une modification de trajectoires pour éviter un obstacle au bon déroulement du vol est un processus très long. Les obstacles sont les éléments à éviter comme les nuages à contourner, les zones interdites de survol ou les conflits de trafic aérien déjà trop dense. Ces obstacles, dont la liste n'est pas exhaustive, sont nommés No-Fly Zones (NFZ). L'obstacle doit être détecté puis le contrôleur doit proposer une solution au pilote : une route d'évitement. Ce dernier doit alors accepter le changement et le signifier au contrôleur au sol qui doit effectuer lui-même les actions correctives nécessaires en cas de conflit avec d'autres appareils. Le pilote, quant à lui, doit modifier manuellement son plan de vol en paramétrant son FMS.

La circulaire d'information aéronautique du 14 février 2008 éditée par NAVCANADA donne un exemple de ces procédures [2].

Dans ce mémoire, nous proposons d'embarquer au sein même de l'avion la fonction de détection d'obstacle et de proposition d'une nouvelle route afin d'éviter les éventuels obstacles. Le but principal est de diminuer la charge de travail des contrôleurs aériens et des pilotes, mais aussi de faciliter la gestion du trafic aérien.

Cela s'inscrit dans une vision novatrice du FMS qui serait amélioré par l'ajout de fonctions. En effet, les prochaines générations de FMS et les prochaines modifications des procédures de l'Air Traffic Control (ATC) se devront d'être novatrices car elles prendront en compte la

densification croissante du trafic aérien. La diminution de la charge de travail des contrôleurs aériens et des pilotes deviendra un enjeu principal. Bien entendu, la sécurité des vols, les enjeux de réduction des coûts, de consommation de carburant et de durée des vols restent au centre de cette nouvelle approche des systèmes embarqués.

Le système proposé prend en compte ces enjeux et, comme il est destiné aux avions de transport de passagers, restreint également les manœuvres possibles de l'appareil en interdisant toute rotation d'angle supérieure à trente degrés ([3]). En effet, si l'avion tolère des limites plus grandes, il est nécessaire de rappeler que le confort des passagers est aussi une des missions des transporteurs aériens et le dépassement de ces limites pourrait provoquer leur inconfort.

Pour résoudre les problèmes mentionnés plus haut, la méthode méta-heuristique a été choisie. Il s'agit d'une méthode qui permet de trouver une solution optimale à un problème posé. En d'autres termes, c'est un processus d'optimisation. Le but est d'analyser l'ensemble de l'espace des solutions afin d'isoler les solutions qui semblent d'être les meilleures en fonction des paramètres observés. Ces paramètres seront développés dans les sections concernées. Il n'y a par contre aucune garantie que la solution trouvée soit la meilleure existante, elle est simplement l'une des meilleures.

Couplés à cette méthode méta-heuristique, des algorithmes d'interpolation sont utilisés afin de détecter les chemins les plus courts. Ce travail a été réalisé en deux dimensions (latitude et longitude), puis en trois dimensions (ajout de l'altitude) et enfin en quatre dimensions (ajout du temps par l'intermédiaire de la vitesse dans certains points de passage). Cette dernière dimension des 4D donne l'accès à la mesure de la vitesse de l'avion en chaque point de la trajectoire et offre la possibilité de travailler sur la prédiction des trajectoires de vol.

Ensuite, plusieurs méthodes ont été développées, comme le calcul du temps de vol effectué et restant, de la poussée nécessaire ou encore de l'estimation du carburant consommé. Nous les nommerons les sorties supplémentaires.

Le principe de fonctionnement de l'algorithme est le suivant :

- 1) entrée des points de cheminement (*waypoints*) du plan de vol initial;
- 2) entrée des NFZ;
- 3) détection du plus court chemin viable;
- 4) entrée de la vitesse aux points de cheminement (*waypoints*) initiaux et du profil d'accélération de l'avion;
- 5) calcul de la vitesse en tous points de la trajectoire;
- 6) calcul des sorties supplémentaires (qui seront développées dans les chapitres concernés);
- 7) enregistrement des résultats obtenus sous forme des données de vol dans un fichier Excel.

L'existence de la « boîte noire » est simulée pendant la dernière étape, ce qui permet d'analyser l'ensemble des données de vol pour la validation du programme.

Dans la suite de cet exposé, les points 1) à 3) seront détaillés pour l'application de la nouvelle méthode en deux et trois dimensions. Ensuite, les points 1) à 5) seront détaillés pour le travail en quatre dimensions. Enfin, les points 6) et 7) seront évoqués pour l'espace en quatre dimensions. Les points 1) à 7) sont donc traités en quatre dimensions. Le système d'évitement des NFZ et celui de sélection du plus court chemin seront présentés dans le chapitre 2, qui sera un chapitre d'explications préalables.

Dans les chapitres 1 à 4 les unités utilisées seront arbitraires, c'est-à-dire que les distances seront exprimées en « unités de distance » où une unité est un écart élémentaire entre deux valeurs consécutives. Les graphiques seront gradués pour chaque axe, en gardant 5 unités entre les graduations.

Dans le chapitre 4, nous mentionnons les unités de temps, de vitesse et d'accélération. Il s'agit encore d'unités arbitraires. Une unité de vitesse reste cependant exprimée en « unité de distance parcourue par unité de temps » et une unité d'accélération est exprimée en « variation de l'unité de vitesse pendant une unité de temps ».

Ce processus avait pour but de centrer la recherche sur les algorithmes pour, ensuite, les valider en utilisant des valeurs réelles. La validation proposée dans le chapitre 7 montre que les résultats obtenus correspondent à la réalité, ce qui permettra de conclure quant à l'utilité de cet algorithme pour l'obtention de données expérimentales de vol.

Il est à noter que lors de son utilisation réelle, ce programme sera relancé régulièrement afin de transmettre aux pilotes une information sûre, optimale et adaptée à la situation présente.

CHAPITRE 1

RERCHERCHES BIBLIOGRAPHIQUES

Dans ce chapitre, les principaux travaux sur le sujet choisi seront exposés ainsi que l'évolution de la recherche dans les différents domaines qui y sont liés.

Le premier sujet étudié sera la *génération et le suivi de trajectoire en 4 dimensions* (ajout de la dimension de temps aux trois dimensions d'espace, ce qui est nécessaire aux calculs de la « prédiction de trajectoire »)

Le sujet suivant est *l'évitement*. Les méthodes les plus fréquemment utilisées seront répertoriées afin d'analyser leurs utilisations à la recherche sur ce sujet.

Enfin, la méthode *méta-heuristique* sera détaillée afin de mieux expliquer sa contribution au domaine actuel des connaissances.

1.1 Génération et suivi de trajectoire en 4 dimensions

La génération et le suivi de trajectoire en quatre dimensions sont des sujets tout à fait récents. Ils sont surtout étudiés dans le cadre du *free-flight*. En 1996, les Messieurs Vu N. Duong et Eric G. Hoffman [4] ont décrit cette notion. Le *free-flight* consiste à embarquer dans l'avion les systèmes d'ATC. Cette méthode rendrait l'avion plus autonome et permettrait de réduire la charge de travail des contrôleurs aériens. Les projets de Free Route Experimental Encounter Resolution FREER-1 -1 et FREER-2 ([3]) ont prouvé que le *free-flight* était possible sur des zones restreintes en espace et en nombre d'avions. Des projets plus récents menés par EUROCONTROL attestent la continuation de la recherche dans cette direction [5].

Ce même rapport [5] affirme que les charges de travail des contrôleurs aériens et des pilotes ne seront pas augmentées quand la transition de l'information en 3D à l'information en 4D aura lieu (page 23). Il explique même que cette transition devrait être bien accueillie par les professionnels (page 31).

Le travail de génération et de suivi de trajectoire en 4D est considéré comme adapté par les administrateurs de la sécurité aérienne. La recherche s'était emparée de ces sujets avant mêmes les conclusions de l'EUROCONTROL et de la FAA.

Dès 1992, la société Honeywell, par l'intermédiaire de Sam Lidén, travaillait sur ce sujet [6] en concluant que les coûts d'exploitation seraient grandement baissés par l'utilisation de trajectoire en 4D, en comparaison avec leur utilisation en 3D.

En 1999, Yann Le Falbek proposait une génération de trajectoires à l'aide des techniques de réseaux de neurones [7] alors qu'en 2001, Nobuhiro Yokoyama proposait une optimisation de la génération de trajectoires en utilisant les algorithmes génétiques [8].

En 2005, Vormer, Frizo Jan-Peter [9] étudiait déjà les aspects théoriques et opérationnels des trajectoires en 4D lors des optimisations des trajectoires d'arrivée aux aéroports. Il avait aussi étudié la manière dont ces aspects en 4D contribuaient à la baisse de la charge de travail des pilotes et des contrôleurs.

En 2007, la compagnie Honeywell, cette fois-ci par l'intermédiaire de Rasa Weber et Eva Crück, soulignait les avantages des trajectoires en 4D pour l'amélioration de l'ATC [10]. L'année suivante, EUROCONTROL produisait le rapport que nous avons déjà mentionné [5].

Cet historique, qui montre une évolution non exhaustive du travail dans le domaine des trajectoires en 4D, prouve que ce projet a sa place dans le monde actuel de la recherche et que la génération de trajectoires en 4D est tout à fait à l'ordre du jour.

1.2 L'évitement

Afin que l'algorithme d'évitement développé soit efficace et novateur, nous avons mené une revue de littérature à ce sujet. Plusieurs méthodes ont été proposées dans le passé.

Dès 1994, Nicolas Alech [11] proposait une méthode d'évitement utilisant les algorithmes génétiques en 2 et 3 dimensions. En 1997, Jean-François Bosc [12] décrivait et comparait plusieurs méthodes d'évitement de conflits aériens que l'on peut adapter aux évitements de zones aériennes. Il avait développé la théorie selon 3 méthodes : celle des forces glissantes décrite initialement par Karim Zegal en 1993 [13], la méthode répulsive basée sur une analogie avec l'électromagnétisme, et une méthode hybride, combinant les deux précédentes. En 1998, Frédéric Médioni [14] expliquait que les systèmes d'évitement pouvaient très bien être embarqués dans les avions.

Cependant, en 2004, la méthode méta-heuristique a été utilisée dans le découpage de l'espace aérien. Cette technique a été décrite par Charles-Edmond Bichot [15] et consistait à fractionner l'espace aérien pour dé-densifier les zones de conflits intenses. En 2005, Olivier Revelin [16] utilisait la méta-heuristique dans le but de résoudre les conflits aériens. Comme la méthode permet de changer une trajectoire pour éviter un conflit, nous avons conclu qu'elle peut permettre un changement de trajectoire pour éviter une NFZ.

C'est pour cette raison que le choix d'une méthode méta-heuristique est apparu comme une bonne idée. Ce type de méthodes est récent mais leur utilité est déjà reconnue. La possibilité de proposer des avancées utiles est aussi grande qu'intéressante.

Dans le but de conclure ce petit historique de la notion d'évitement de zones, rappelons que, dans le cadre du free-flight, les zones à éviter sont de deux types [4]: les conflict-zones (zones de conflits aériens) et les No-Go zones (zones interdites de survol de façon temporaire ou permanente). Nous réunirons ces deux types de zones sous le nom de « No-Fly Zones » : NFZ.

1.3 La méta-heuristique

Plutôt que de lister quelques dizaines de méthodes d'optimisation existantes pour aboutir à la méta-heuristique, c'est cette méthode qui sera directement présentée.

Il s'agit d'une famille d'algorithmes d'optimisation. Tous ces algorithmes ont pour but la recherche d'un optimum local, défini par l'utilisateur, dans l'espace des solutions étudiées. Par exemple, il est possible d'utiliser la méthode méta-heuristique pour trouver la solution exprimée par la valeur minimale de l'ordonnée d'une courbe donnée. En effet, l'ensemble des solutions est un ensemble de points sur la courbe, et la solution recherchée est le point qui a la plus petite ordonnée.

Cependant, si la méthode méta-heuristique peut s'expliquer à l'aide de concepts très simples, elle peut aussi résoudre des systèmes très complexes ([14] et [15]).

Il existe plusieurs types de méthodes méta-heuristiques. Certaines de ces méthodes considèrent et traitent une population de solutions (algorithmes génétiques [17], par exemple). D'autres de ces méthodes, auxquelles notre travail fera appel, sont des méthodes itératives qui traitent les solutions une à une. Ces méthodes explorent l'ensemble des solutions dans le but d'y trouver la solution la plus optimale possible, selon des critères entrés par l'utilisateur. Il existe plusieurs de ces méthodes : la méthode de la descente, la méthode de recherche par tabou (taboo method), la méthode du recuit simulé et autres.

Les méthodes que nous utiliserons sont la méthode de la descente et la méthode de recherche par tabou. La méthode de recherche par tabou, introduite par Fred Glover en 1986 (un livre est publié à ce sujet en 1997 par le même auteur [18]), apporte l'innovation d'interdire certaines solutions déjà testées, ce qui permet un grand gain d'efficacité qui sera utile lors du travail que nous allons développer. Les avantages de ces deux méthodes (simplicité et efficacité) seront exploités au maximum durant notre travail.

Le Groupe d'Études et de Recherche en Analyse des Décisions (GERAD), basé à l'École Polytechnique Montréal, a publié un recueil très intéressant sur les méthodes existantes en méta-heuristiques [17]. Plusieurs ouvrages et auteurs de référence y sont cités et on y trouve aussi la description des méthodes les plus utilisées.

CHAPITRE 2

LES ALGORITHMES PRINCIPAUX

2.1 Le système d'évitement des NFZ

2.1.1 Le principe de travail

L'évitement des NFZ est un point clé du programme que nous avons développé. Dans l'utilisation réelle de notre programme, au sein d'un FMS, les NFZ seront des entrées automatiques dépendantes du radar météorologique ou du radar de trafic aérien.

En période de développement, de simulation et de validation, les NFZ sont entrées manuellement par l'opérateur. Le principe actuel est d'entourer la NFZ par un cercle avec x_centre_NFZ et y_centre_NFZ qui sont les coordonnées du centre de la NFZ et r son rayon. Cette méthode n'est pas optimale, mais c'est l'algorithme d'évitement qui était testé et non pas la forme à éviter. Un algorithme de détection de forme pourrait nous donner une NFZ optimale et une adaptation de notre processus d'évitement suffirait à finaliser cette partie du travail sur l'évitement des NFZ.

2.1.2 L'évitement de zone

Lorsque le centre de la NFZ a été saisi ainsi que le périmètre du cercle qui la définit, la première version du processus d'évitement travaillait comme suit. Tout d'abord, l'algorithme vérifie qu'aucun *waypoint* initial n'est présent dans les NFZ. Si tel est le cas, une modification du plan de vol initial est requise et les actions à prendre doivent être définies et authentifiées par les professionnels. Le programme retourne donc l'information et demande un traitement manuel de la situation.

Si ce cas ne se présente pas, une NFZ est présente entre deux points (par exemple, les points A et B) et le programme crée automatiquement quatre points d'évitement autour d'elle. Leurs coordonnées sont les suivantes :

1 : $(x_{centre_NFZ}, y_{centre_NFZ} + 2r)$; 2 : $(x_{centre_NFZ}, y_{centre_NFZ} - 2r)$;
 3 : $(x_{centre_NFZ} - 2r, y_{centre_NFZ})$; 4 : $(x_{centre_NFZ} + 2r, y_{centre_NFZ})$.

Les points sont placés à une distance de $2r$ du centre de la NFZ de manière à se trouver à une distance minimale r de la NFZ, ce qui permet l'évitement dans tous les cas testés. Il s'agit d'un choix fait durant le développement pour tester le programme. Par la suite, il sera montré comment cette distance est réduite pour optimiser le système d'évitement.

Quand les points d'évitement sont disposés autour de la NFZ, le programme teste automatiquement toutes les routes possibles entre A et B dans le but de déterminer la plus efficace. Ces routes sont composées du point A , du point B et d'au moins un des quatre points d'évitement créés. C'est le passage par au moins un de ces points qui permet l'évitement de la zone. En effet, l'ajout d'un ou de plusieurs points d'évitement dans la trajectoire initiale permet de dérouter l'avion de manière à éviter la NFZ. Comme le reste du plan de vol n'est pas modifié, l'avion retrouve ensuite sa trajectoire initiale. Les quatre points d'évitement ne sont jamais utilisés ensemble car il est plus efficace d'en utiliser un seul en contournant la NFZ par son autre côté.

Dès qu'une de ces routes est validée, c'est à dire dès que la trajectoire proposée ne coupe plus la NFZ, la route est mise en mémoire. Suite aux tests de toutes les routes, les routes validées sont toutes mesurées et la plus courte est alors conservée. Les points d'évitement sont alors rajoutés dans ce qui est nommé la chaîne de *waypoints*. Par exemple si la route conservée utilise les points d'évitement 2 et 3, la chaîne de *waypoints* pour lier A à B (qui est originalement $[A, B]$) devient $[A, 2, 3, B]$.

La figure 2.1 montre les résultats obtenus avec la méthode de méta-heuristique. Sur la figure 2.1, nous pouvons voir la trajectoire directe liant les 2 points A et B en pointillé. La NFZ est représentée sous la forme d'un cercle hachuré et le résultat de la méthode de méta-heuristique sous la forme des trois droites, en traits pleins liant A et B par l'intermédiaire de deux waypoints d'évitement, qui forment ainsi la trajectoire d'évitement qui contourne le NFZ. Les deux cercles en haut et à gauche de la NFZ sont des points d'évitement générés automatiquement, mais qui ne sont pas utilisés par l'algorithme. En effet, les utiliser aurait rallongé la trajectoire d'évitement.

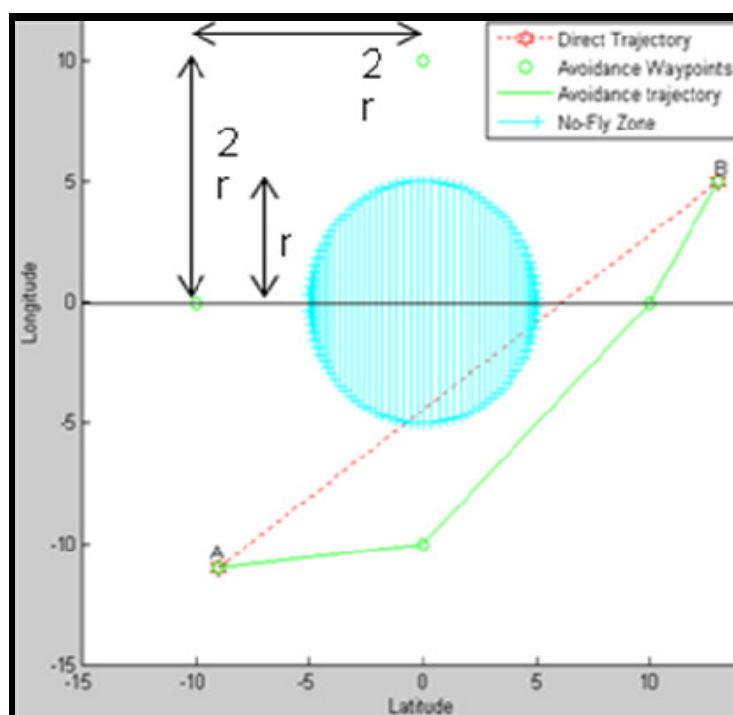


Figure 2.1 Trajectoire d'évitement de la NFZ passant par 2 points A et B.

Si un *waypoint* se trouve dans une NFZ, voici ce que retourne le programme :

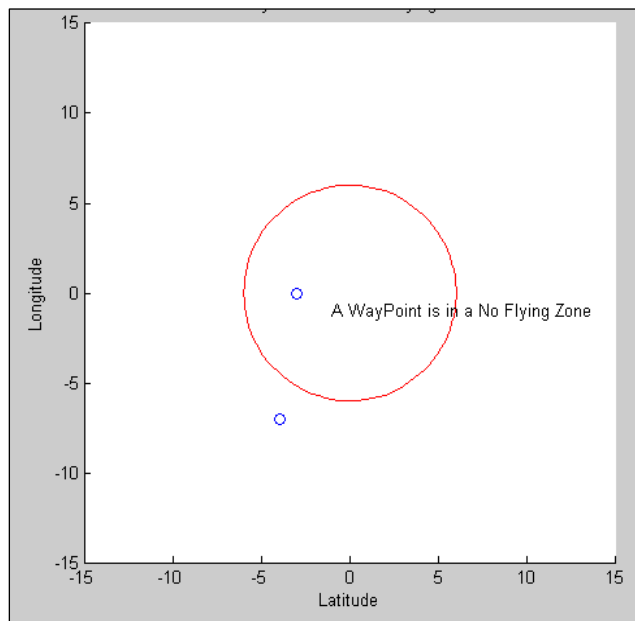


Figure 2.2 Représentation d'un *waypoint* dans une NFZ.

2.1.3 L'optimisation de l'évitement de zone

La méthode a été testée et validée dans les trois situations possibles :

- 1) un *waypoint* est dans la NFZ. Dans ce cas, un traitement manuel est requis;
- 2) la trajectoire traverse la NFZ. Dans ce cas l'évitement automatique de la zone a lieu;
- 3) la trajectoire ne traverse pas la NFZ. Dans ce cas, aucune action correctrice n'est faite.

L'étape suivante est alors l'optimisation de la trajectoire. Le processus étant correct, il a fallu ensuite trouver un moyen de contourner la zone le plus efficacement possible, donc en parcourant une distance minimale. Pour ce faire, la méthode d'évitement est la même, mais le placement des points d'évitement est différent.

Soit d une longueur telle que $d = r + n * pas$ où n est le nombre d'itérations nécessaires pour éviter la NFZ et le pas est une constante fixée lors du développement du programme. Dans le cas de ce développement, le pas vaut 0.1 unité de longueur. À la première itération ($n = 1$), $d = r + 0.1$. À la deuxième itération ($n = 2$), $d = r + 0.2$, et à la dixième itération ($n = 10$), $d = r + 1$.

Les points d'évitement sont alors placés comme suit :

$$(x_centre_NFZ, y_centre_NFZ + d); (x_centre_NFZ, y_centre_NFZ - d); \\ (x_centre_NFZ - d, y_centre_NFZ); (x_centre_NFZ + d, y_centre_NFZ).$$

À chaque itération, toutes les routes d'évitement sont testées. Si l'une d'elle est validée, le programme termine l'itération en cours et sélectionne la plus courte route valide. Sinon, n augmente d'une unité et le programme teste à nouveau les différentes routes possibles.

Ainsi, le chemin d'évitement trouvé passera au plus près de la NFZ en réduisant donc la distance à parcourir.

Sur la figure 2-3, le rayon de la NFZ (le grand cercle hachuré) est de 3 unités de longueur et le point d'évitement (*avoidance waypoint*) est à 5.3 unités de longueur du centre de la NFZ. Les points d'évitement sont donc placés plus proche de la NFZ que dans le cas non optimisé. Cela raccourcit les trajectoires d'évitement. Sur la figure 2-3, on voit que la route directe, en pointillé, pénètre la NFZ. L'algorithme d'évitement se déclenche. La trajectoire en trait plein est la trajectoire d'évitement sélectionnée par l'algorithme.

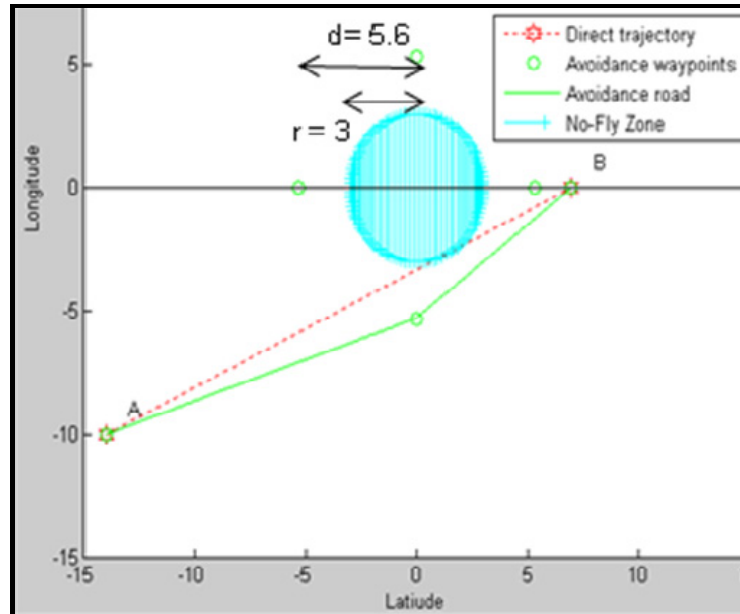


Figure 2.3 Trajectoire optimisée d'évitement de NFZ entre les points A et B.

2.2 Détermination du chemin le plus court

Suite à l'étape d'évitement des NFZ, le programme dispose d'une chaîne de *waypoints*. Dans le but de créer une trajectoire viable pour l'avion, des algorithmes d'interpolation entre les *waypoints* sont utilisés. Ensuite, une méthode de méta-heuristique que nous avons créée sera utilisée pour déterminer la trajectoire la plus courte.

Cette méthode combine les avantages de deux méthodes déjà existantes que nous allons présenter : la méthode de la descente et la méthode de recherche par tabous.

2.2.1 La méthode de la descente

La première est la méthode de la descente [17]. Il s'agit d'une méthode de recherche de l'optimum local dans laquelle on considère S étant l'ensemble des chemins d'évitement possibles et $f(s)$ la longueur du chemin d'évitement s .

Voici le principe du programme :

- 1) Choisir aléatoirement une solution s , un chemin d'évitement possible, dans S
- 2) Déterminer aléatoirement une solution s' , un chemin d'évitement possible, dans S

Il y a ensuite deux possibilités :

- 3.1) Si $f(s') < f(s)$, poser $s = s'$
- 3.2) Sinon retourner à l'étape 2)

Cependant, cette méthode a un défaut très contraignant : elle s'arrête au premier extremum trouvé. En d'autres termes, elle n'explore qu'une petite partie de l'espace des solutions et la recherche s'arrête à la première qui est acceptable selon les critères entrés par l'utilisateur.

Comme ce choix ne nous paraît pas satisfaisant, nous avons exploré d'autres méthodes afin de remédier à ce défaut.

2.2.2 La méthode de recherche par tabous

La seconde méthode est la méthode de recherche par tabous. C'est une toute autre méthode : elle ne se base pas sur la méthode de la descente.

La méthode de recherche par tabous se déroule en plusieurs étapes.

Tout d'abord, il faut fixer un critère d'arrêt d'optimisation, par exemple le nombre d'itérations. Une solution de référence est trouvée, puis fixée et d'autres solutions sont testées. Dès qu'une solution testée est meilleure que la solution de référence, elle la remplace et est déposée dans une liste de tabous, ce qui signifie que les d'autres solutions testées et non retenues sont exclues des prochaines recherches d'optimisation.

Ainsi, un nombre minimum d'itérations sont nécessaires pour atteindre une solution optimale. Si le critère d'arrêt est le temps, le nombre d'itérations ou le nombre d'itérations sans amélioration, la recherche devient plus efficace. L'algorithme s'arrête quand le critère d'arrêt d'optimisation est atteint.

2.2.3 Principe retenu

Pour commencer, rappelons que les solutions étudiées sont des trajectoires liant les *waypoints*. Le critère pour trouver la meilleure solution est « la trajectoire la plus courte est la meilleure ». Plus tard dans le travail, à partir du chapitre 3, nous ajouterons un critère de viabilité de la trajectoire. Ce critère sera expliqué dans les parties concernées.

La solution retenue par cet algorithme reprend le principe de la méthode de la descente à laquelle certaines propriétés de la recherche par tabous ont été ajoutées. En effet, le programme est muni d'un critère d'arrêt d'optimisation qui considère *10 itérations successives sans effectuer d'amélioration*. Ce critère, un compromis entre la rapidité des résultats et l'efficacité de la solution retenue, est trouvé par expérience durant les essais du programme. Ce critère est donc spécifique à ce travail.

Par ailleurs, la liste des solutions considérées comme tabous est à taille variable: quand une solution est ajoutée à la liste des tabous, elle ne chasse pas les autres solutions de la liste mais elle s'y ajoute. L'algorithme créé est donc hybride et novateur.

Nous avons retenu le principe suivant, pour lequel les solutions sont des interpolations polynomiales réalisées à l'aide des points présents dans la chaîne de *waypoints*.

Ces nouveaux points d'évitement, situés entre les *waypoints*, sont obtenus sur l'ensemble de la trajectoire suite à l'évitement des NFZ. L'une de ces fonctions est choisie aléatoirement puis sa longueur totale est calculée : elle est la fonction de référence. Ensuite, les autres fonctions testées sont également choisies aléatoirement et mesurées de la même manière. Si l'une d'elle est plus courte que la fonction de référence, elle devient la fonction de référence.

Voici le pseudo code de la solution développée pour ce programme :

Soit - S l'ensemble des chemins d'évitement possibles

- $f(s)$ la longueur du chemin d'évitement s

- T la liste des tabous et n un compteur

1) Poser $n = 0$ et $T = []$ (cela signifie que la liste est vide). Choisir une solution s dans S

2) Déterminer une solution s' dans S

3) Si $f(s') < f(s)$, poser $s = s'$ et $n = 0$. Poser $T = [T, s]$: s devient une solution tabou à ne pas tester à nouveau

4) Si $f(s') \geq f(s)$, poser $n = n+1$

Il y a ensuite deux possibilités :

5.1) Si $n < 10$, retour à l'étape 2)

5.2) Sinon la solution de référence est la solution finale retenue

Le schéma explicatif suivant présente la méthode de manière graphique :

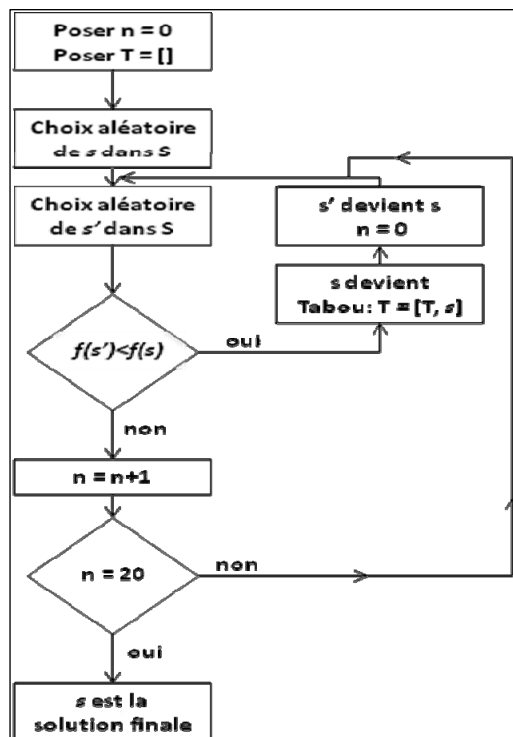


Figure 2.4 Schéma de l'algorithme de la nouvelle méthode pour le calcul de la trajectoire optimisée.

Sur la figure 2-5, les résultats obtenus par cette nouvelle méthode sont visibles.

Tel que précisé dans la légende de la figure 2-5, les *waypoints* sont représentés par des cercles. La trajectoire en trait plein est la plus courte existante car elle relie *waypoints* consécutifs par des droites. Toutes les courbes en pointillés représentent des trajectoires qui ont été testées. La trajectoire en pointillés épais est également une courbe testée. Cependant, elle est la plus courte. Elle a été affichée d'une manière différente pour montrer qu'il s'agit du chemin le plus court qui est finalement retenu.

Le résultat était connu d'avance : le plus court chemin est la ligne droite. Cependant, l'algorithme a été capable de trouver la bonne réponse ce qui permet de le valider.

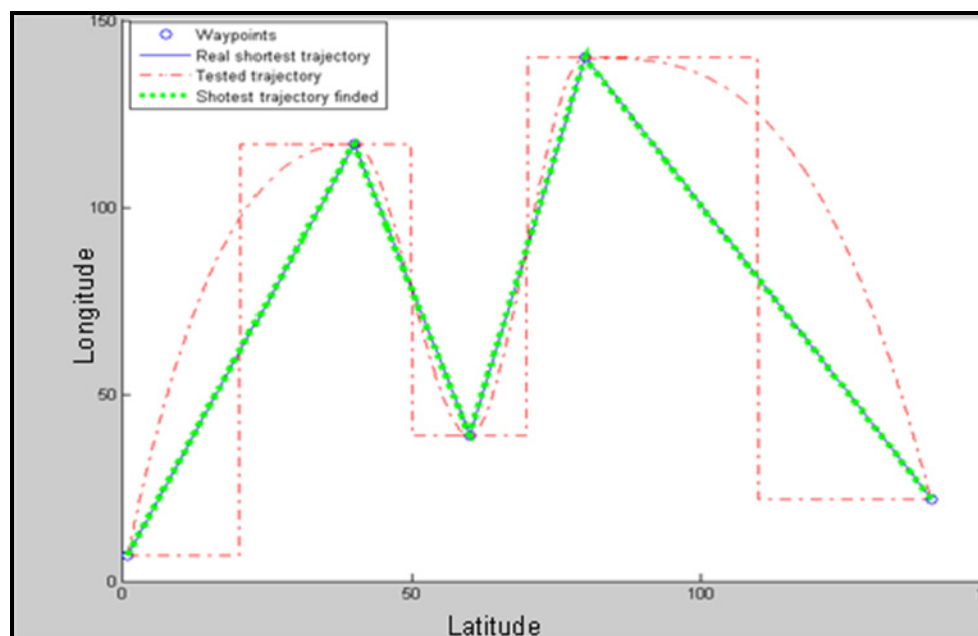


Figure 2.5 Le plus court chemin trouve par la combinaison de la méthode tabou et de la descente.

Ici, les choix aléatoires ont seulement impliqués les tests de trois types d'interpolations différents. D'autres types d'interpolations sont disponibles (cinq à l'heure actuelle). De plus, la procédure d'ajout de nouveaux types d'interpolations a été programmée pour être simple. La facilité de maintenance et d'optimisation de notre programme nous semblait importante.

Les interpolations disponibles actuellement sont celles proposées par Matlab: pChip, Cubic, Linear, Nearest, Spline. Pour ajouter une méthode d'interpolation, il suffit de rajouter une fonction en Matlab décrivant une nouvelle interpolation dans le dossier de travail, et ensuite de l'ajouter dans la routine de sélection aléatoire des méthodes d'interpolation pour l'ajouter parmi les choix possibles.

Décrire les méthodes d'interpolation n'est pas un point clé de ce travail car elles sont utilisées comme des outils, mais nous allons quand même les présenter brièvement.

Les explications proviennent du logiciel Matlab [18].

La méthode Nearest considère un point dont l'abscisse est au milieu des abscisses de deux points donnés consécutifs (dans notre cas, deux *waypoints* consécutifs) est sélectionné. L'algorithme calcule ensuite la moyenne entre les ordonnées des deux points choisis et l'arrondit à la valeur entière la plus proche. Cette valeur devient l'ordonnée du point sélectionné. L'algorithme considère ce nouveau point créé (qui possède désormais une abscisse et une ordonnée) comme un point donné et recommence le même travail. Elle s'arrête quand le nombre de points demandés par l'utilisateur est atteint.

La méthode Linear considère un point dont l'abscisse est pris au milieu des abscisses de deux points donnés consécutifs (dans notre cas, deux *waypoints* consécutifs) est sélectionné. L'algorithme calcule ensuite la moyenne entre les ordonnées de ces deux points. Elle considère ce nouveau point créé (qui possède désormais une abscisse et une ordonnée) comme un point donné et recommence le même travail. Elle s'arrête quand le nombre de points demandés par l'utilisateur est atteint.

La méthode Cubic utilise, à partir des points donnés (dans notre cas, les *waypoints*), une fonction Cubic qui va remplacer la courbe discrète.

La méthode Pchip, pour *Piecewise Cubic Hermite Interpolating Polynomial*, utilise la fonction polynomiale par morceau de Hermite qui va remplacer la courbe discrète.

La méthode Spline fonctionne comme la méthode Pchip, mais en utilisant plusieurs types de polynômes.

2.2.4 Les limites de la méthode

Afin de chercher les limites en temps de calcul de la méthode que nous avons créée, nous avons réalisé plusieurs tests. Le but est de trouver à partir de quel moment le calcul devenait trop long. Le tableau suivant montre les résultats obtenus pour des différents nombres des *waypoints*. Les points disposés l'ont été de manière aléatoire.

Tableau 2.1 Temps de réponse de la méthode en fonction du nombre de points

Nombre de points	Temps pour réponse
2	Instantané
3	Instantané
5	< 0.5 s
10	< 0.5 s
15	< 0.5 s
20	< 1 s
25	< 1s
50	< 1.5 s

Nous pouvons constater que l'algorithme de calcul est rapide. Il faudrait considérer les calculs pour les lignes avec 20 points et 25 points. En effet, lors d'un trajet transatlantique (qui sera l'exemple de notre trajet de validation), les compagnies aériennes utilisent environ 10 *waypoints*. Si on considère que le système évite automatiquement un maximum de 4 zones, ceci peut générer 4 à 12 points d'évitement. Nous pouvons constater que l'algorithme offre une réponse rapide dans ces situations, moins de 1 seconde. Quand le nombre de points approche 50, la situation devient plus longue à régler. Cependant, cette situation correspondrait à plus de 12 évitements lors d'un vol, ce qui est énorme. La méthode s'avère rapide, elle sera donc utilisée dans toute la suite du travail.

CHAPITRE 3

ÉTUDE DES TRAJECTOIRES DE VOL EN DEUX DIMENSIONS

Dans ce chapitre, deux cas seront traités. Le premier cas sera détaillé dans son fonctionnement, le second sera donné à titre de démonstration. Avant de montrer ces deux cas, les hypothèses de travail de cette section seront établies. Il faut souligner que l'algorithme décrit au chapitre 2 est appliqué dans cette section pour étudier l'optimisation des trajectoires de vol en 2D.

3.1 Hypothèses

Plusieurs hypothèses seront considérées. Certaines seront maintenues dans les chapitres suivants, d'autres sont spécifiques à cette partie du travail. Les hypothèses de ce chapitre sont les suivantes :

- Tous les *waypoints* sont de type Fly-Over.

Comme les *waypoints* sont Fly-Over, l'avion doit nécessairement passer par ces points.

- Il n'y a pas des *waypoints* dans une NFZ.

Dans le cas contraire, un traitement manuel est exigé, comme cela a déjà été mentionné dans le chapitre 2.

- Les NFZ sont considérées comme fixes.

Certaines NFZ peuvent être mobiles, une perturbation météorologique en est un bon exemple. Cependant, comme le but de ce travail est de raisonner en prédiction, il est possible de savoir à quel endroit l'obstacle sera présent sur la route à un instant donné. C'est pourquoi, dans le travail que nous présentons, les NFZ sont fixes.

- L'altitude de l'avion est constante.

Comme le travail est en deux dimensions, l'altitude de l'appareil est considérée comme constante dans tout ce chapitre.

- La vitesse de l'avion est constante.

A cette étape du travail, le temps n'entre pas encore en compte. Pour cette raison, la vitesse de l'avion n'est pas modélisée et elle est considérée comme constante. De plus, ce travail a lieu en vol de croisière.

- Les routes d'évitement ne contiennent pas d'autres NFZ

Il est proposé qu'il n'existe pas d'autres NFZ autour de la trajectoire de l'appareil que celles considérées dans la mise en place du problème.

3.2 Premier cas

3.2.1 Mise en place

Pour résoudre ce problème, les *waypoints* considérés sont disposés dans un espace en deux dimensions dont les limites sont $X_{min} = -15$ unités, $X_{max} = 15$ unités, $Y_{min} = -15$ unités, $Y_{max} = 15$ unités. Les *waypoints* du plan de vol ont les coordonnées (X, Y) suivantes:

A(-14, -14), **B**(-5, -5), **C**(0, 2), **D**(5, 5) et **E**(14, 0).

Deux NFZ avec les coordonnées (X, Y, r) sont présentes sur la trajectoire de l'appareil: **NFZ**₁ = (-10, -10, 1) et **NFZ**₂ = (2, 10, 1).

3.2.2 Fonctionnement de l'algorithme de résolution

La première étape dans la résolution du problème est d'entrer les données dans le programme, ce qui se fait directement par une l'interface en Matlab.

L'interface du programme a été faite pour être la plus simple et la plus réaliste possible. Dans la réalité, le pilote ne fait qu'entrer les *waypoints* du vol dans l'ordinateur de bord et les NFZ seront des entrées automatiques transmises par d'autres systèmes embarqués. Dans le programme que nous avons créé, cette facilité d'utilisation est recréée ; le logiciel nous demande d'entrer les coordonnées des *waypoints* un par un. Une fois ceci fait, le logiciel

récapitule les points entrés afin de vérifier les entrées programmées. Les vrais FMS permettent la même vérification.

Par la suite, l'entrée des NFZ se fait de la même manière mais, encore une fois, cette étape ne sera pas exécutée dans l'utilisation réelle du programme que nous développons.

Une fois que les données sont entrées dans le programme, l'algorithme fonctionne en trois parties. Dans la première partie, il détecte si la trajectoire pénètre des NFZ et, si oui, à quels endroits. Chaque portion de la trajectoire est étudiée dans l'ordre suivant: la portion entre les deux premiers *waypoints*, puis la portion entre le deuxième et le troisième *waypoint* et ainsi de suite.

Dans le cas présent, il y a une violation de NFZ entre les points **A** et **B** et une autre entre les points **D** et **E**. Quand une violation est constatée sur une portion de trajectoire, l'algorithme alerte l'utilisateur par le message: **No_flying_zone = 1**. Dans le cas contraire, le message annonce **No_flying_zone = 0**.

L'algorithme traite indépendamment les différentes violations afin de proposer la chaîne de *waypoints* pour éviter les NFZ. Il est à noter que si deux NFZ sont sur la même portion de trajectoire (donc entre deux *waypoints* consécutifs), les deux NFZ sont traitées indépendamment, mais la chaîne de *waypoints* d'évitement est choisie de manière à éviter toutes les NFZ de cette portion.

Dans la première étape la proposition d'évitement faite par l'algorithme donne :

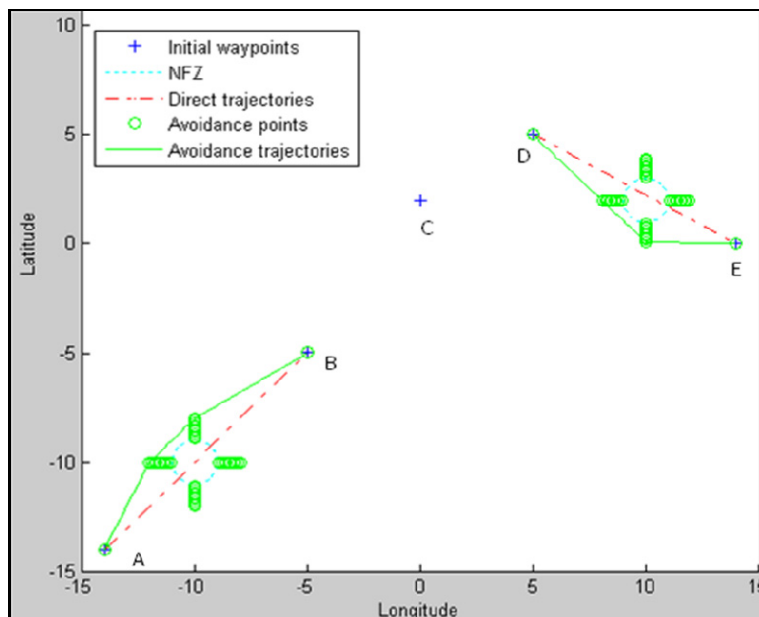


Figure 3.1 Première étape de l'algorithme pour le cas 1: Évitement des NFZ en 2D.

La théorie décrite dans le chapitre 2 a été appliquée pour l'évitement des NFZ et celles-ci ont été traitées indépendamment l'une de l'autre. Une chaîne de *waypoints* qui matérialise toute la trajectoire d'évitement est créée.

Les cercles en pointillés sont les NFZ et les trajectoires en tirets sont les routes directes. Pour plus de clarté, seul le chemin d'évitement valide a été présenté pour chaque NFZ : il s'agit des courbes en traits pleins sur le schéma. Les points d'évitement à chaque itération (cercles autour des NFZ) ont été conservés sur la figure pour montrer la précision de la recherche d'une trajectoire d'évitement la plus courte possible. Ces points résultent de l'utilisation de l'algorithme décrit dans le chapitre 2.

La seconde étape de l'algorithme est la transmission de toute la chaîne en simplifiant l'information le plus possible pour en faciliter le traitement. Les étoiles sont les points de cheminement de la chaîne et ceux entourés d'un cercle correspondent aux points de

cheminement initiaux (de gauche à droite : **A**, **B**, **C**, **D** et **E**). La suite du programme n'utilise donc pour entrée que la chaîne résultante de *waypoints* en étoiles, que nous pouvons visualiser sur la figure suivante 3-2.

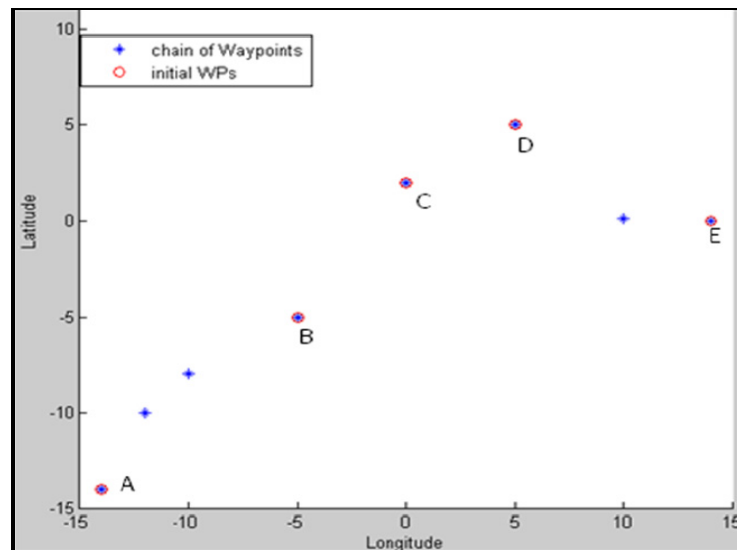


Figure 3.2 Deuxième étape de l'algorithme pour le cas 1: Création de la chaîne des waypoints en 2D.

L'étape suivante, qui est la troisième et dernière phase de la résolution, consiste à relier les *waypoints* en créant la plus courte trajectoire possible. Pour cela, le processus décrit dans le chapitre 2.2 est utilisé et se déroule en plusieurs parties.

Tout d'abord, l'espace des abscisses sera échantillonné de manière à multiplier par dix (10) le nombre de points disponibles. Cette valeur de dix (10) est choisie dans le but d'obtenir un bon compromis entre la précision des résultats et la rapidité de calcul. Il sera dit plus tard que cette précision est évidemment différente quand le système est testé en grandeur réelle. Dès que l'espace des abscisses est échantillonné de la bonne manière, les interpolations sont effectuées en fonction des ordonnées disponibles ce qui est réalisé pour chacune des interpolations testées par le programme de la manière décrite dans le chapitre 2.2.

La figure suivante montre les tests réalisés. Nous qualifions une trajectoire de « **viable** » si elle ne présente pas de virage dont le taux est supérieur à trente degrés [3]. Dans cet exemple, sur les différentes routes testées, une seule route n'est pas retenue car elle est non viable, c'est celle qui lie les points par des droites. Deux autres trajectoires sont testées, mais l'une d'elle est plus courte que l'autre. Ainsi la trajectoire viable qui est en pointillé est testée et rejetée alors que la courbe en trait plein est testée et conservée. En effet, elle est la plus courte des trajectoires viables. Le nombre de tests, ici deux, dépend complètement des choix aléatoires de techniques d'interpolation, comme nous l'avons expliqué dans le chapitre 2.2. La trajectoire retenue est donc un compromis entre la viabilité d'une route et sa longueur.

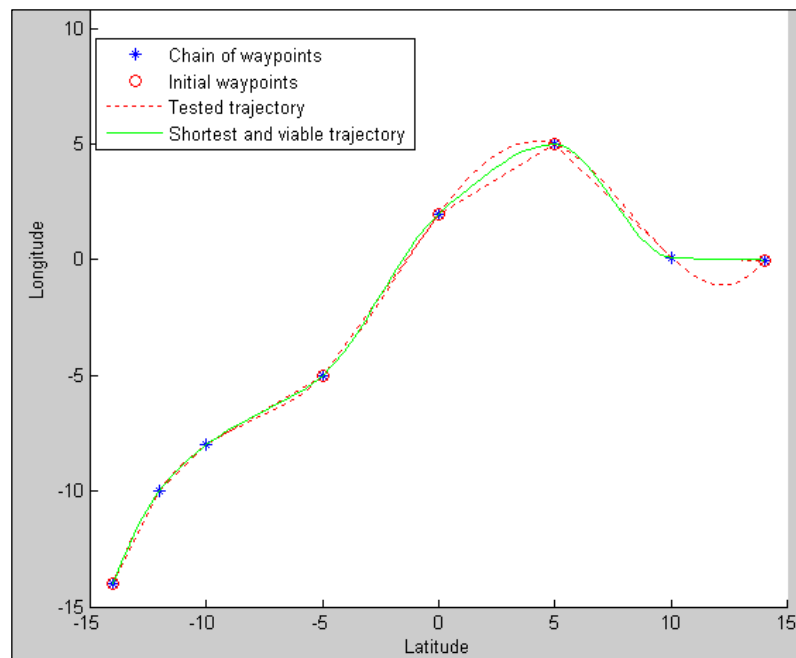


Figure 3.3 Troisième étape de l'algorithme pour le cas 1: Détection du chemin le plus court et viable en 2D.

Dans le tableau suivant, les caractéristiques des trajectoires testées sont données. Ces caractéristiques sont la longueur, le fait de savoir si la route est la plus courte ou pas et le fait de savoir si la route est viable. Le résultat sera la sélection ou non de la trajectoire. La trajectoire sélectionnée par l'algorithme sera *la trajectoire viable la plus courte* parmi les trajectoires testées. Ici, il s'agit de la trajectoire liée à l'interpolation cubique : elle est viable et c'est la plus courte des trajectoires viables. Plus précisément, les routes viables sont

calculées avec les interpolations de type *Nearest* et *Cubic*. La plus courte de ces deux routes est celle calculée avec l'interpolation de type *Cubic*.

Tableau 3.1 : Sélection de la trajectoire pour le cas 1 en 2D

Nom de l'interpolation	Longueur (en unité)	Route la plus courte	Viabilité	Sélection
Nearest	49.96	Non	Oui	Non
Spline	38.36	Oui	Non	Non
Cubic	38.45	Non	Oui	Oui

Concernant l'information donnée au pilote, elle sera réduite à sa plus simple expression et seule la trajectoire définitive sera donnée. Le visuel final serait semblable à ce que montre la figure 3-4 La légende et les axes du graphique ne sont donnés qu'à titre indicatif. Le pilote ne verra, sur son FMS, que la trajectoire finale ainsi que les *waypoints* initiaux avec les conventions de couleurs réglementaires pour les FMS. Ainsi, les professionnels ne seront pas saturés d'informations. Bien sûr, l'accès au reste des informations peut être disponible sur demande.

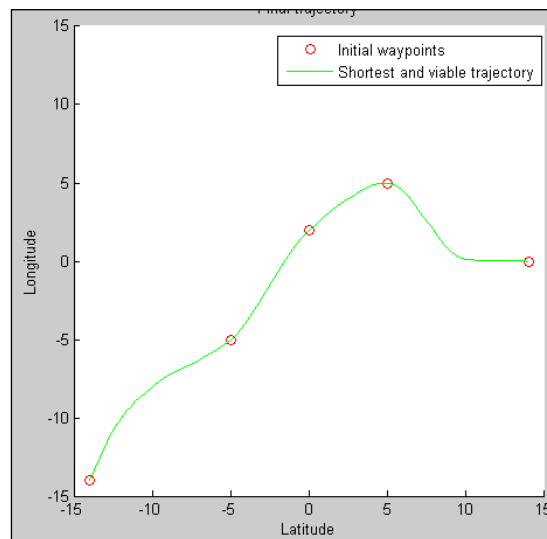


Figure 3.4 La trajectoire finale retenue dans le premier cas analysé.

3.3 Second cas

3.3.1 Mise en place

Pour résoudre ce problème, les points considérés sont disposés dans un espace en deux dimensions dont les limites sont $X_{\min} = -15$ unités, $X_{\max} = 15$ unités, $Y_{\min} = -15$ unités et $Y_{\max} = 15$ unités. Les *waypoints* initiaux ont les coordonnées (X, Y) suivantes : **A**(-14,14), **B**(-5,-7), **C**(0,-2), **D**(10,3) et **E**(14,5). Deux NFZ sont présentes sur la trajectoire de l'appareil et ont les coordonnées (X, Y, r): **NFZ₁** (-10, 10, 1) et **NFZ₂**(5, 0, 0.5).

3.3.2 Résolution

La première étape de détection et d'évitement des NFZ donne le graphique 3-5. Les NFZ sont les cercles en pointillés, les routes directes sont en tirets et les routes d'évitement en traits pleins.

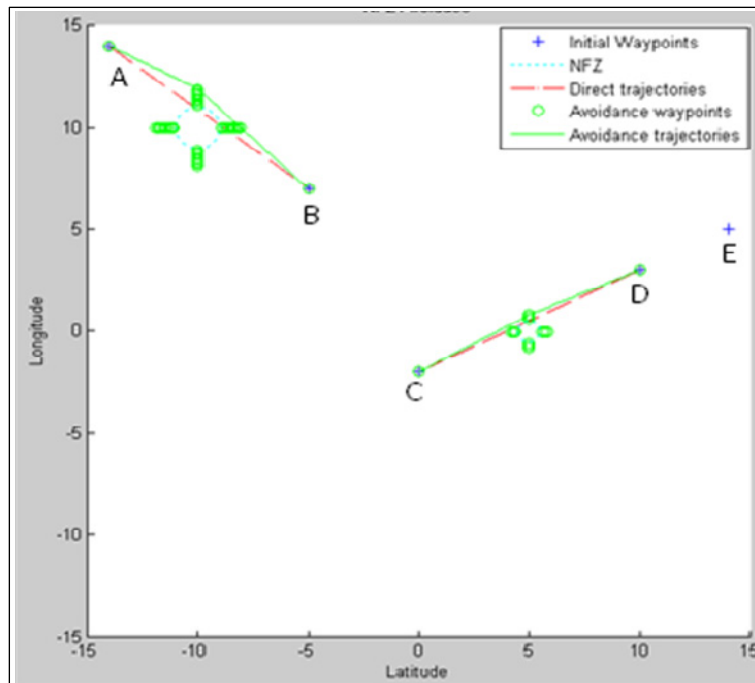


Figure 3.5 Première étape de l'algorithme pour le cas 2 : évitements des NFZ en 3D.

Les cercles en traits pleins sont les *waypoints* d'évitement générés automatiquement. Les trajectoires d'évitement passent par un ou deux des points d'évitement comme nous le disions dans le chapitre 2.

La chaîne de *waypoints* créée est la suivante :

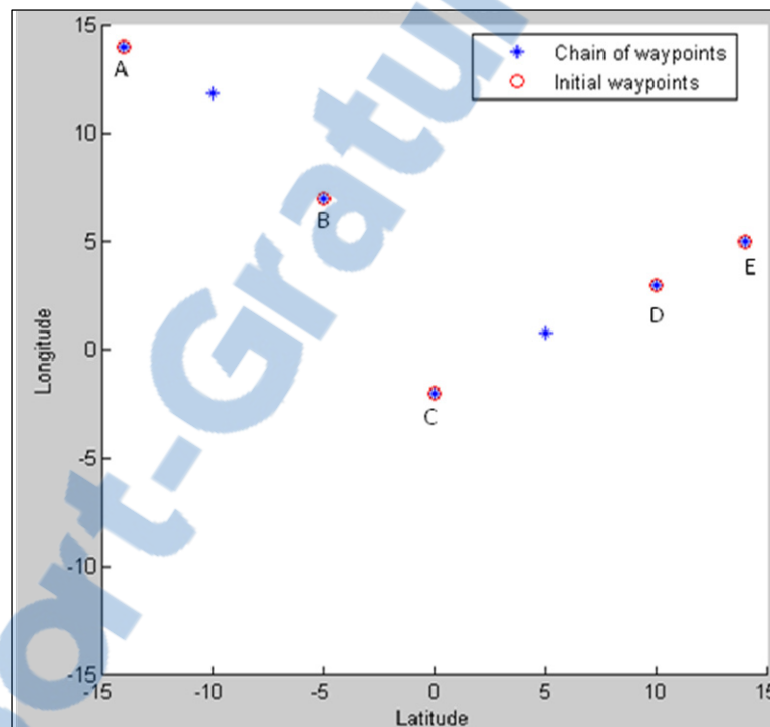
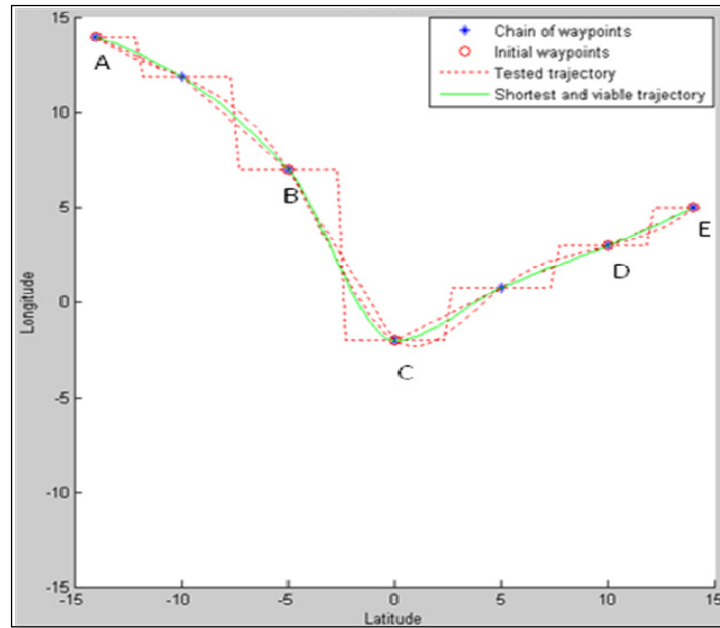


Figure 3.6 Deuxième étape de l'algorithme pour le cas 2: Création de la chaîne des waypoints en 3D.

Le choix de la trajectoire optimisée est donné sur la figure 3-7. Encore une fois, les routes en pointillés sont toutes les routes testées par la méthode que nous avons créée et la route en trait plein est celle qui est retenue. Elle a été retenue car elle est la plus courte des trajectoires viables. Ce choix est réalisé, premièrement en tenant compte des viabilités des routes et deuxièmement de leurs longueurs (voir le tableau suivant 3.1).



**Figure 3.7 Troisième étape de l'algorithme pour le cas 2:
Détection du chemin le plus court en 3D.**

Le tableau de comparaison des trajectoires obtenues par les différentes interpolations testées est le suivant, dans lequel nous pouvons voir la viabilité et la longueur des routes testées.

Tableau 3.1 : Sélection de la trajectoire pour le cas 2 en 2D

Type de l'interpolation	Longueur (en unité)	Route la plus courte	Viabilité	Sélection
Nearest	48.91	Non	Non	Non
Linear	37.67	Oui	Non	Non
Cubic	38.01	Non	Oui	Oui
Pchip	38.01	Non	Oui	Non
Spline	38.57	Non	Oui	Non

Dans ce cas, les trajectoires calculées par les interpolations de type Cubic et pChip ont les mêmes longueurs. L'une des deux doit alors être retenue au détriment de l'autre. Dans notre travail, si une fonction testée a la même longueur que la fonction de référence, elle est considérée comme moins efficace. Ceci est une convention choisie lors du développement. C'est la première des deux fonctions testées qui est retenue. Dans le cas ici présent, c'est la fonction d'interpolation de type Cubic. Il est rappelé que les choix des fonctions testées est aléatoire.

Le rendu au pilote sera le suivant, encore une fois sans la légende et avec les conventions de couleur réglementaires pour les FMS.

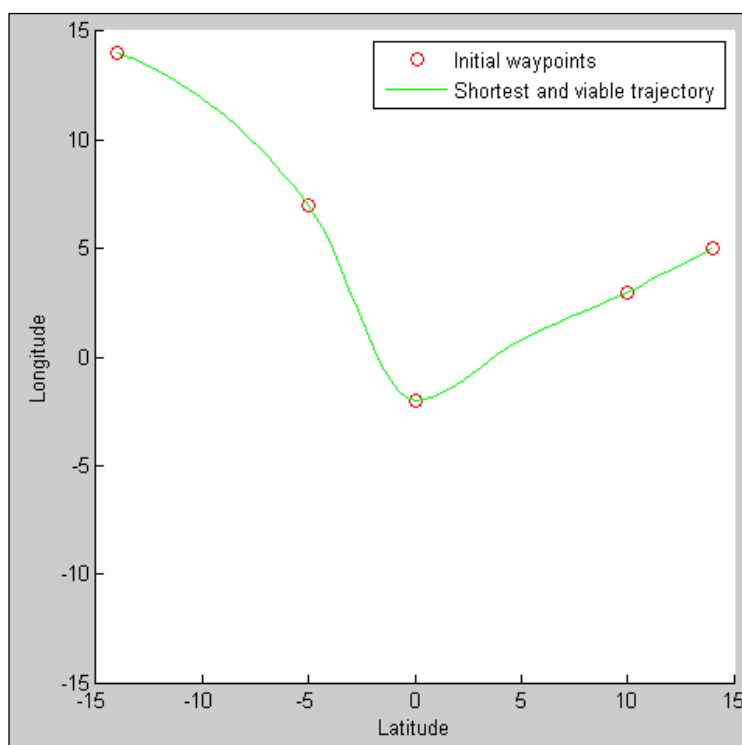


Figure 3.8 La trajectoire finale retenue dans le deuxième cas analysé.

CHAPITRE 4

ÉTUDE DES TRAJECTOIRES DE VOL EN TROIS DIMENSIONS

Dans ce chapitre, deux cas seront traités. Le premier cas sera détaillé dans son fonctionnement, le second cas sera donné à titre de démonstration. Avant de montrer ces deux cas, les hypothèses de travail de cette partie seront établies.

4.1 Hypothèses

Nous allons considérer plusieurs hypothèses. Certaines seront maintenues dans les chapitres suivants et d'autres sont propres à cette partie du travail. Les hypothèses considérées sont les suivantes. Les hypothèses déjà expliquées dans les chapitres précédents ne sont pas redéveloppées ici :

- Tous les *waypoints* sont de type Fly-Over.
- Aucun *waypoint* ne se trouve dans une NFZ.
- Les NFZ sont considérées comme fixes.
- La vitesse de l'avion est constante.
- Les routes d'évitement sont libres de toutes autres NFZ.
- Les changements d'altitude sont faibles.

Ce travail traite du vol de croisière. Dans cette configuration de vol, les changements d'altitude sont faibles. Par exemple, le passage du niveau de vol 340 (FL340 : 34,000 pieds d'altitude) au niveau de vol 350 (FL350 : 35,000 pieds d'altitude) ne demande qu'une variation de 1000 pieds, soit 300 mètres.

4.2 Préambules

4.2.1 Préambule relatif à l'évitement des NFZ

Afin d'appréhender ce chapitre correctement, il est important de dire que les évitements de NFZ doivent se faire en palier. Cette recommandation, faite par les constructeurs et appliquée par les compagnies aériennes, s'explique simplement. Pour commencer, l'avion est plus maniable en palier. En effet, éviter une zone en montée reviendrait à utiliser le mouvement de lacet de l'avion plutôt que son virage en roulis ce qui serait moins efficace. Par ailleurs, l'utilisation du mouvement de lacet en évitement de zone rendrait le vol non symétrique, ce qui engendrerait une augmentation du coût d'exploitation. Enfin, les efforts sur la structure sont moindres lors d'un mouvement de roulis que lors d'un mouvement de lacet. C'est pour ces raisons que, dans ce chapitre et dans les suivants, l'appareil repassera en palier avant d'éviter les zones de vol et reprendra ensuite son changement d'altitude.

4.2.2 Préambule relatif au choix du plus court chemin

Ici encore, le choix du plus court chemin utilise l'algorithme de méta-heuristique décrit dans le chapitre 2.2. Ce programme est en fait utilisé deux fois. En effet, pour diminuer le temps de compilation et faciliter la maintenance du programme, il a été décidé de travailler deux fois en deux dimensions (latitude versus longitude et latitude versus altitude), tel qu'expliqué dans les paragraphes suivants.

Lors de sa première utilisation, le programme réagit exactement comme dans le chapitre précédent. La latitude est discrétisée, puis une interpolation permet d'obtenir l'espace des longitudes. Ainsi, à chaque échantillon de latitude correspond un échantillon de longitude.

Lors de sa seconde utilisation, le programme utilise la discrétisation des latitudes dans le but, toujours par interpolation, d'obtenir l'espace des altitudes. Ainsi, à chaque échantillon de latitude correspond un échantillon d'altitude.

Les deux interpolations sont optimisées indépendamment l'une de l'autre. Ensuite, la trajectoire est dessinée en prenant en compte la succession de triplets de données. Ces triplets sont composés d'une latitude, d'une longitude et d'une altitude qui définissent la position du point dans l'espace en 3D.

4.3 Premier cas

4.3.1 Mise en place

Pour résoudre ce problème, les points considérés sont disposés sur un espace en trois dimensions dont les limites sont $X_{\min} = -15$ unités, $X_{\max} = 15$ unités, $Y_{\min} = -15$ unités, $Y_{\max} = 15$ unités, $Z_{\min} = -15$ unités et $Z_{\max} = 15$ unités.

Les *waypoints* du plan de vol ont les coordonnées (X, Y, Z) suivantes : **A**(-14,-14, -5), **B**(-10, -10, -2), **C**(-5, -5, 0), **D**(5, 3, 0), **E**(8, 6, 0), **F**(12, 6, 1) et **G**(14, 6, 1). Une NFZ est présente sur la trajectoire de l'appareil et a les coordonnées (X, Y, Z, r) = (0, 0, 0, 1).

4.3.2 Fonctionnement de l'algorithme de résolution

La première étape dans la résolution du problème est d'entrer les données au programme ce qui se fait directement par l'interface Matlab. Les données sont les coordonnées tridimensionnelles des *waypoints*, du *centre* des NFZ et le *rayon* des NFZ. Encore une fois, dans sa version d'exploitation, le logiciel ne demandera pas d'entrer manuellement les NFZ. Les informations du radar météorologique et du radar de trafic fourniront automatiquement les coordonnées des NFZ au logiciel.

Une fois que les données sont entrées, l'algorithme travaille à nouveau en trois parties. La première partie consiste à détecter si la trajectoire pénètre des NFZ et, si oui, à quels

endroits. Dans ce but, chaque portion trajectoire (trajectoire entre deux *waypoints* consécutifs) est étudiée. Dans notre cas, il y a une violation de NFZ entre les points C et D (cette violation est constatable sur la figure 4-1). L'algorithme nous alerte par le message : `No_flying_zone = 1`, comme dans le chapitre précédent.

Ensuite, les *waypoints d'évitement* sont calculés afin de déterminer la trajectoire d'évitement. La méthode est toujours la même que celle décrite dans le chapitre 2. En effet, l'avion évite les NFZ en palier. C'est pourquoi la NFZ est encore considérée comme un cercle et que l'évitement s'effectue en deux dimensions. A la fin de cette phase, le programme a donc à sa disposition une chaîne de *waypoints* exprimée en trois dimensions d'espace.

La dernière phase de la résolution est le choix de la trajectoire la plus courte. Comme ceci a été expliqué, la détermination du trajet le plus court est effectuée en deux étapes. D'abord, les longitudes sont déterminées en fonction des latitudes. Ensuite les altitudes sont déterminées en fonction des latitudes.

Voici les tableaux de choix d'interpolation. Il est à noter que la viabilité de la trajectoire a la même signification que précédemment : « une trajectoire est qualifiée de viable si elle ne présente pas de virage dont le taux est supérieur à trente degrés » [3].

Dans le plan de la latitude versus la longitude :

Tableau 4.1 Sélection de la trajectoire pour le cas 1 dans le plan horizontal

Type de l'interpolation	Longueur (en unité)	Route la plus courte	Viabilité	Sélection
Nearest	42.93	Non	Non	Non
Linear	35.87	Oui	Non	Non
Cubic	35.98	Non	Non	Non
Spline	36.05	Non	Oui	Oui

Dans le plan de la latitude versus l'altitude:

Tableau 4.2 Sélection de la trajectoire pour le cas 1 dans le plan vertical

Type de l'interpolation	Longueur (en unité)	Route la plus courte	Viabilité	Sélection
Nearest	38.86	Non	Non	Non
Linear	29.50	Oui	Non	Non
Cubic	29.60	Non	Oui	Oui
Spline	29.58	Non	Oui	Non

Dans le plan de la latitude versus la longitude, c'est l'interpolation *Spline* qui est retenue car c'est la plus courte des trajectoires viables. Dans le plan de la latitude versus l'altitude, c'est aussi l'interpolation *Spline* qui est retenue car c'est la plus courte des trajectoires viables. On peut remarquer que les résultats obtenus par la méthode *Pchip* ne sont pas montrés dans les tableaux. La raison est que le critère d'arrêt de l'algorithme est atteint avant de tester cette méthode, ce qui est dû aux choix aléatoires des méthodes d'interpolation. Dans les graphiques suivants, les cercles en pointillés seront les NFZ, la route en tirets sera la route directe et la route en pointillés sera la trajectoire d'évitement.

Une fois que cette étape est effectuée, les triplets de coordonnées sont reliés :

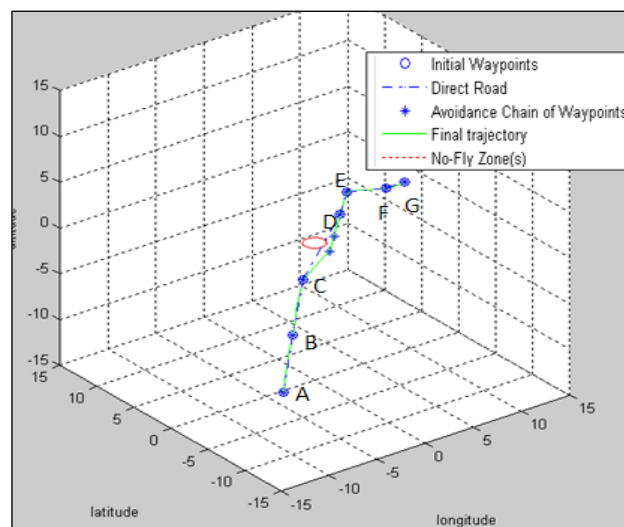


Figure 4.1 La trajectoire retenue dans le cas 1 suite à l'évitement des NFZ.

Ci-dessous sont fournis les résultats obtenus pour les trajectoires dans les plans de la latitude versus la longitude et de l'altitude versus la longitude.

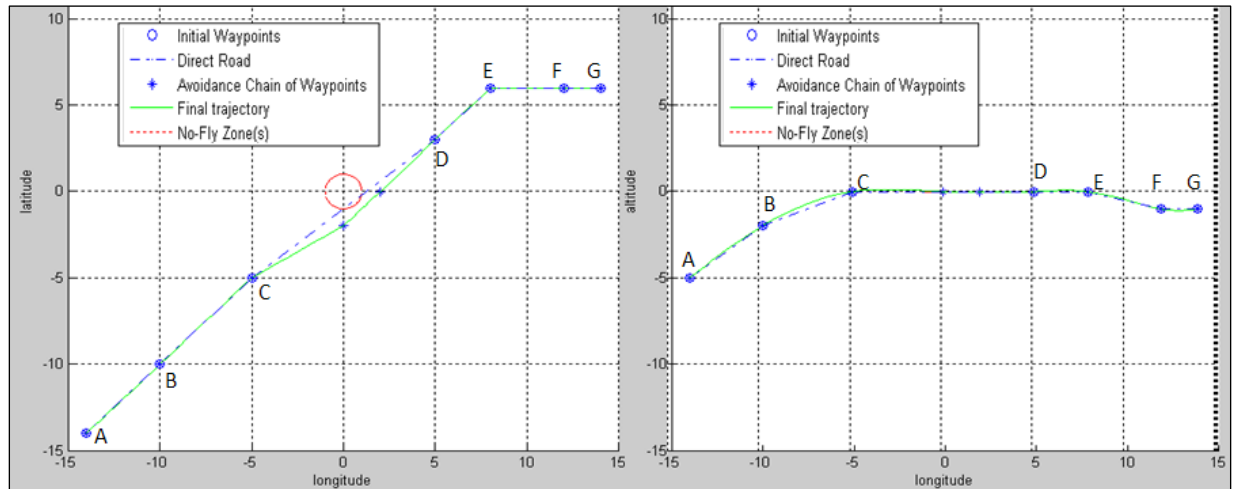


Figure 4.2 La trajectoire retenue dans le cas 1 suite à l'évitement des NFZ dans le plan de la latitude versus la longitude (gauche) et de l'altitude versus la longitude (droite).

Nous pouvons constater, sur la partie de droite de la figure 4-2, que l'altitude lors de l'évitement de la NFZ est constante. Concernant l'information à donner aux pilotes, des études d'ergonomie sont nécessaires car aucun FMS actuel n'offre d'informations graphiques en trois dimensions. Cette information est toujours donnée sous la forme du plan de vol latéral (latitude versus longitude) et les FMS les plus récents (dont celui de l'A380) donnent simplement un indice d'altitude.

Encore une fois, nous utilisons des unités arbitraires pour créer les algorithmes. Ceux-ci seront validés en utilisant des paramètres réels dans le chapitre 7.

4.4 Second cas

4.4.1 Mise en place

Pour ce problème, les points considérés se trouvent dans un espace en 3D dont les limites sont $X_{\min} = -15$ unités, $X_{\max} = 15$ unités, $Y_{\min} = -15$ unités, $Y_{\max} = 15$ unités, $Z_{\min} = -15$ unités et $Z_{\max} = 15$ unités. Les *waypoints* du plan de vol ont les coordonnées (X, Y, Z) suivantes : **A**(-15, -15, -2), **B**(-14, -14, -2), **C**(-6, -6, 0), **D**(0, -2, 0), **E**(5, 4, 0), **F**(8, 6, 1), **G**(10, 6, 1), **H**(12, 6, 0.5), **I**(13, 6, 0.25), **J**(14, 6, 0) et **K**(15, 6, 0). Deux NFZ sont présentes sur la trajectoire de l'appareil : **NFZ₁** (-10, -10, -1, 1) et **NFZ₂** (2, 0.5, 0, 1).

4.4.2 Résolution

Suite à l'évitement des deux NFZ et à la mise en place de la chaîne de *waypoints*, voici les tableaux de choix d'interpolation dans le plan de la latitude versus la longitude.

Tableau 4.3 Sélection de la trajectoire pour le cas 2 dans le plan horizontal

Nom de l'interpolation	Longueur (en unité)	Route la plus courte	Viabilité	Sélection
Nearest	47.08	Non	Non	Non
Linear	39.53	Oui	Non	Non
Cubic	39.75	Non	Oui	Oui
Spline	40.53	Non	Oui	Non

Dans le plan latitude-altitude, le tableau de sélection des trajectoires est le suivant :

Tableau 4.4 Sélection de la trajectoire pour le cas 2 dans le plan vertical

Nom de l'interpolation	Longueur (en unité)	Route la plus courte	Viabilité	Sélection
Nearest	32.30	Non	Non	Non
Linear	30.70	Oui	Non	Non
Cubic	30.79	Non	Oui	Oui
Spline	30.82	Non	Oui	Non

On peut remarquer que les résultats obtenus par la méthode *Pchip* ne sont pas montrés dans les tableaux. La raison est que le critère d'arrêt de l'algorithme est atteint avant de tester cette méthode, ce qui est dû aux choix aléatoires des méthodes d'interpolation.

Dans le plan de la latitude versus la longitude, la méthode choisie est la méthode *Cubic* ainsi que dans le plan de la latitude versus l'altitude. Elle est dans les deux cas la plus courte des trajectoires viables. La résolution de ce cas est représentée sur la figure suivante :

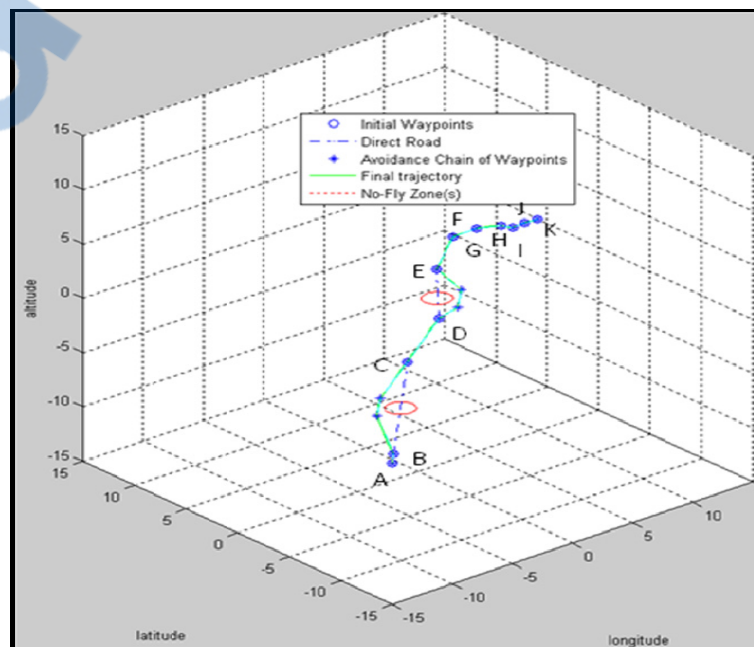


Figure 4.3 La trajectoire retenue dans le cas 2 suite à l'évitement des NFZ.

Les trajectoires dans les deux plans de vol (latéral et vertical) sont les suivantes :

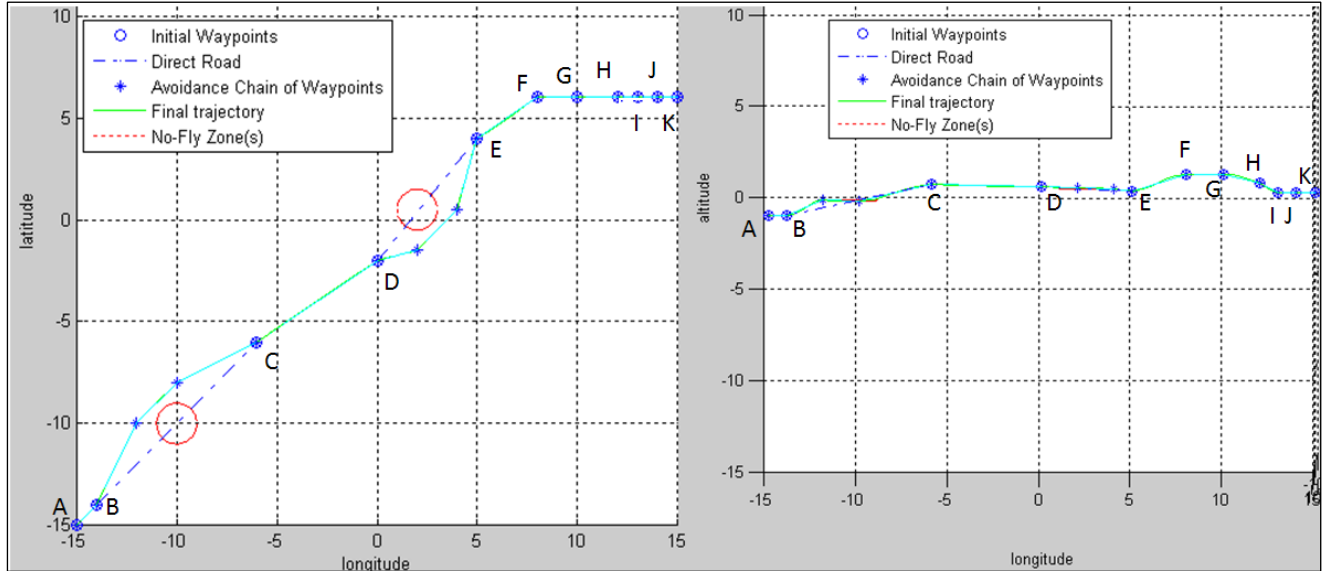


Figure 4.4 La trajectoire retenue dans le cas 2 suite à l'évitement des NFZ dans le plan de la latitude versus la longitude (gauche) et de l'altitude versus la longitude (droite).

Il est possible d'observer, sur le plan vertical (partie de droite de la figure 4-4), que l'évitement de zone entre les points B et C se fait bien en palier, c'est-à-dire à altitude constante. Après avoir passé le point B, l'avion commencera par prendre de l'altitude, puis il passera en palier pour éviter la NFZ et reprendra son ascension jusqu'à atteindre le point C.

CHAPITRE 5

ÉTUDES DES TRAJECTOIRES DE VOL EN QUATRE DIMENSIONS

Encore une fois, deux cas seront traités. Le premier cas sera détaillé dans son fonctionnement, le second cas sera donné à titre de démonstration. Avant de montrer ces deux cas, les hypothèses de travail de cette partie seront établies.

5.1 Hypothèses

Dans ce chapitre, plusieurs hypothèses seront considérées. Certaines hypothèses seront maintenues dans les chapitres suivants et d'autres sont propres à cette partie du travail.

Voici l'ensemble des hypothèses de ce chapitre :

- Tous les *waypoints* sont de type Fly-Over.
- Aucun *waypoint* ne se trouve dans une NFZ
- Les NFZ sont considérées comme fixes.
- Les routes d'évitement sont libres de toutes autres NFZ.
- Les changements d'altitude sont faibles.
- Les changements de vitesse sont faibles.

Ce travail traite du vol de croisière. Dans cette configuration de vol, les changements de vitesse sont faibles, de l'ordre de 10%.

- L'accélération de l'avion est considérée comme constante.

Dans ce travail, le principe de résolution est testé ainsi que le réalisme de la simulation. Si les résultats sont validés avec une accélération constante, un travail futur pourrait être de remplacer cette donnée par les courbes d'accélération réelle (donc une accélération variable) de l'avion pour augmenter le réalisme de la simulation.

5.2 Prise en compte de la vitesse.

5.2.1 Préambule

Dans l'hypothèse où les variations de vitesse sont faibles, elles sont considérées comme sans influence sur les trajectoires de l'avion, notamment sur les rayons de giration. C'est pourquoi le programme les demande à l'opérateur après la détermination de la trajectoire la plus courte. Les entrées pour le calcul de la vitesse en chaque point de la trajectoire sont les triplets de positions de chaque point de la trajectoire et les vitesses aux *waypoints* initiaux.

5.2.2 Méthode de fonctionnement

Dans un premier temps, l'algorithme va agir exactement comme il le fait en trois dimensions. Ensuite, le programme va demander la vitesse de passage requise au dessus des waypoints. De nos jours, c'est le pilote qui demande à l'ATC une autorisation de changement de vitesse ou c'est l'ATC qui le propose au pilote. Cependant, en accord avec la vision proposée dans ce travail, l'autonomie de l'appareil doit être la plus grande possible.

De plus, un évitement de NFZ va retarder l'appareil dans son plan de vol, il faut donc que celui-ci soit capable de rattraper ce retard afin de ne pas biaiser le Required Time Arrival (RTA). Dans le but de remédier aux problèmes de temps tout en conservant la problématique de diminution de la charge de travail de l'ATC, il est nécessaire que le FMS propose automatiquement une adaptation de sa vitesse. Celle-ci reste soumise à l'acceptation ou au rejet par les pilotes.

Dans cette partie de travail, la propriété testée est la suivante : « L'avion peut adapter sa propre vitesse afin de survoler le *waypoint* à la vitesse souhaitée ».

La priorité est donnée à la vitesse de l'appareil pour rejoindre son point d'arrivée au plus proche du RTA. Ainsi, lors d'une accélération entre deux *waypoints*, l'avion devra atteindre

la vitesse maximale le plus rapidement possible. Au contraire, si une décélération est nécessaire, l'avion gardera sa vitesse maximale le plus longtemps possible afin de décélérer le plus tard possible. Ce cas se présente, par exemple, si nous estimons qu'il est plus rentable de ralentir pour laisser passer une NFZ plutôt que de la contourner en accélérant.

Laisser passer une NFZ peut signifier attendre qu'un conflit se résolve ou encore attendre qu'une NFZ de nature météorologique se déplace afin de ne plus se trouver sur la trajectoire de l'avion.

Enfin, un avion est capable d'accélérer et de décélérer en montée et en descente, *a fortiori* pour les petites variations que nous considérons. Pour cette raison, aucune distinction n'est faite entre la montée, le palier et la descente en ce qui concerne les changements de vitesse.

5.2.3 Méthode de calcul

La méthode de calcul de la vitesse en tout point de la trajectoire est la suivante. Nous appelons « point de la trajectoire » ou « point » chaque triplet de coordonnées (altitude, longitude, latitude) qui constitue une portion de la trajectoire créée.

Au début de cette étape, la trajectoire de l'appareil est discrétisée de manière très précise avec un pas très faible. Ensuite, le programme nécessite les entrées suivantes :

- Le profil d'accélération de l'appareil.

Pour tester cette fonction, l'accélération et la décélération sont constantes et ont les mêmes valeurs absolues.

- La vitesse de passage au dessus de chacun des *waypoints* du plan de vol.

Nous allons étudier la manière de calcul de la vitesse en chaque point de la trajectoire dans toutes les situations possibles pour une trajectoire entre un point A dont la vitesse de survol est V_a et un point B dont la vitesse de survol est V_b . Le cas $V_a = V_b$ ne sera pas traité : il suffit de dire que la vitesse reste la même en tous points.

a) Cas de l'accélération : $V_a < V_b$

Dans ce cas, la vitesse en chaque point de la trajectoire est calculée. L'échantillonnage étant assez précis, nous faisons l'hypothèse que chaque fraction de trajectoire est assimilable à une droite. Le calcul de la distance D entre deux points est donné par la formule suivante :

$$D = \sqrt{\left[(x_i(n) - x_i(n-1))^2 + (y_i(n) - y_i(n-1))^2 + (z_i(n) - z_i(n-1))^2 \right]} \quad (1)$$

La vitesse de départ (V_a), la distance entre deux points consécutifs et l'accélération en étant toutes connues, on peut calculer la vitesse au point suivant.

Ainsi, par récurrence, l'obtention de la vitesse en chaque point est très rapide. Dès que la vitesse cible (V_b) est atteinte (admettons que cela se produise au point C), la vitesse en chaque point de la trajectoire reste constante, jusqu'au survol du *waypoint* B, donc entre C et B. Si la vitesse cible n'est pas atteignable, l'algorithme peut recalculer les vitesses aux prochains *waypoints* dans le but d'obtenir la durée de vol nécessaire pour respecter le RTA.

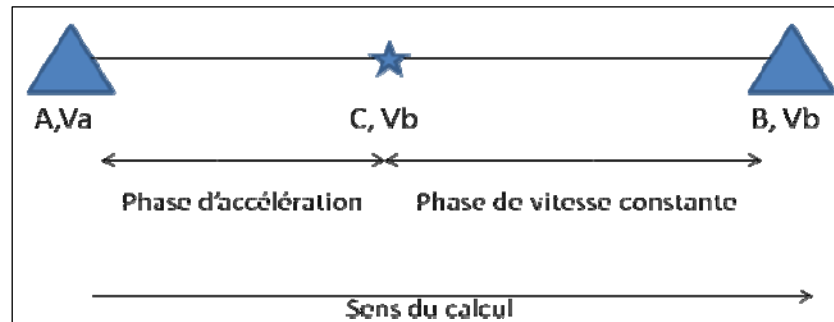


Figure 7.1 Schéma du calcul de la vitesse lors d'un vol accéléré.

Comme tout ce travail est un travail de prédiction, une solution viable et définitive est offerte aux pilotes sans que ceux-ci soient soumis au stress de la prise de décision imminente.

b) Cas de la décélération : $V_a > V_b$

A nouveau, le programme calcule la vitesse en chaque point de la trajectoire. L'hypothèse selon laquelle chaque fraction de trajectoire est assimilable à une droite est toujours valable et les calculs des distances restent les mêmes. Il est souhaité que l'avion décélère le plus tard possible. La méthode de travail est ici présentée.

La vitesse d'arrivée au *waypoint* cible est connue (il s'agit de V_b), ainsi que le taux de décélération et la distance entre deux points consécutifs de la trajectoire discrétisée. Dans ce cas, l'algorithme part du *waypoint* cible (le point B) et calcule la vitesse en chaque point dans le sens inverse à ce que sera la trajectoire de l'avion. En d'autres termes, le programme se déroule à l'envers : il part du point cible, ici le point B, et la vitesse de chaque point précédent est calculée.

Dorénavant, c'est quand la vitesse du *waypoint* de départ est obtenue (V_a) en un point (soit C ce point) que la vitesse reste constante pour tous les points précédant C dans la trajectoire.

Ainsi, entre A et C, la vitesse sera égale à V_a et entre C et B la vitesse sera en constante diminution jusqu'à atteindre V_b .

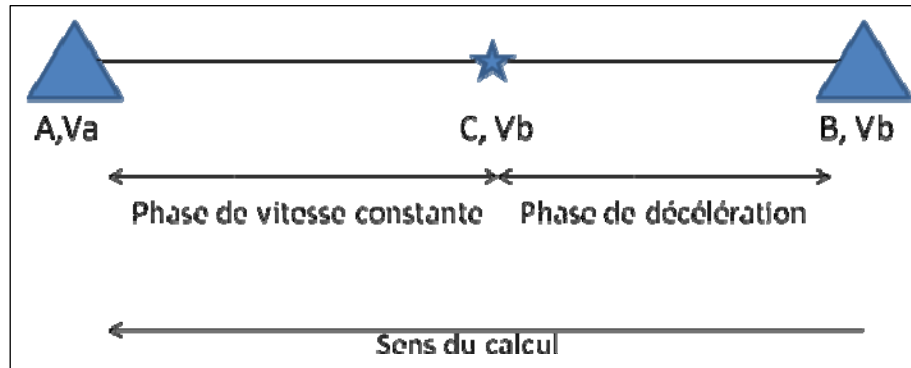


Figure 5.2 Schéma du calcul de la vitesse lors d'un vol décéléré.

Si la vitesse cible V_b n'est pas atteignable, les vitesses aux prochains *waypoints* sont recalculées dans le but d'obtenir la durée de vol nécessaire pour respecter le RTA. Comme tout ce travail est un travail de prédiction, une solution viable et définitive est offerte aux pilotes sans que ceux-ci soient soumis aux contraintes de la prise de décision imminente.

5.3 Premier cas

5.3.1 Mise en place

Pour ce problème, les points considérés sont disposés dans un espace en 3D dont les limites sont $X_{\min} = -15$ unités, $X_{\max} = 15$ unités, $Y_{\min} = -15$ unités, $Y_{\max} = 15$ unités, $Z_{\min} = -15$ unités et $Z_{\max} = 15$ unités. La vitesse est donnée en unités de vitesse, comme nous l'expliquons dans l'introduction. Les *waypoints* ont les coordonnées $(X, Y, Z, \text{Vitesse de passage})$ suivantes : **A**(-14,-14, 0, 14), **B**(-5, 0, 0, 15), **C**(6, 9, 0, 15) et **D**(14, 14, 0, 14).

Le taux de variation de vitesse est de 2 unités d'accélération. Ceci est une donnée que nous imposons au programme. Il n'y aura pas de NFZ sur la trajectoire étudiée. L'algorithme de détection de NFZ est déclenché, mais comme il n'y a pas de violation de NFZ, l'algorithme d'évitement, lui, n'est pas déclenché. Dans le second cas de ce chapitre, dans la section 5.4, l'algorithme d'évitement sera déclenché.

5.3.2 Méthode de résolution

Le programme commence par demander les données nécessaires aux calculs des vitesses. L'opérateur entre alors le profil d'accélération de l'avion et les vitesses obligatoires de passage aux *waypoints*. Dans l'utilisation réelle du FMS, le profil d'accélération est une donnée fournie par un autre système embarqué, tout comme les vitesses de passage des *waypoints*. Encore un fois, nous travaillons ici avec des valeurs arbitraires et dans le chapitre 7 le travail est validé pour des données réelles.

Voici la trajectoire choisie par notre algorithme. Encore une fois, la route directe est en tiret et la route viable la plus courte est en trait plein :

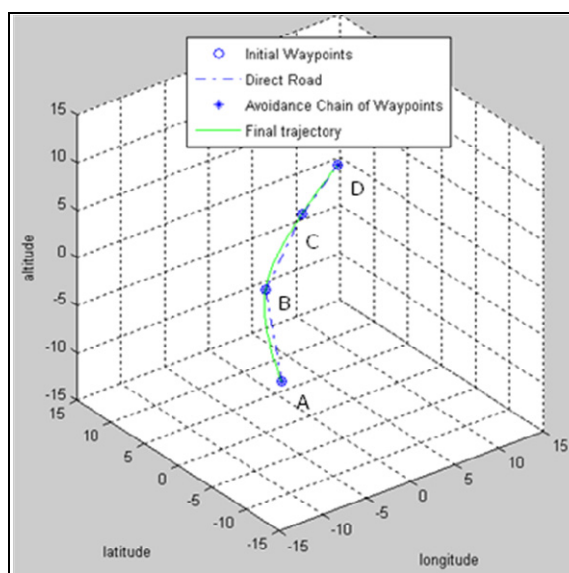


Figure 5.3 Trajectoire finale calculée dans le premier cas, en 4D.

Voici les résultats obtenus pour la phase d'accélération entre les points A et B. La vitesse cible (V_b) est atteinte le plus tôt possible et reste à cette valeur jusqu'à ce que la prédiction atteigne le point cible (le *waypoint* B).

Tableau 5.1 La phase d'accélération dans le premier cas

X	Y	Z	Vitesse	
-14	-14	0	14	← Coordonnées du point A
-13.625	-13.2953	0	14.11321	
-13.25	-12.5972	0	14.22463	
-12.875	-11.9061	0	14.33424	
-12.5	-11.2227	0	14.44201	
-12.125	-10.5474	0	14.54792	
-11.75	-9.88092	0	14.65194	
-11.375	-9.22367	0	14.75407	
-11	-8.57625	0	14.85428	
-10.625	-7.93919	0	14.95255	
-10.25	-7.31305	0	15	← À ce point, la vitesse devient égale à 15 unités de vitesse
-9.875	-6.69837	0	15	
...				
-5.375	-0.41581	0	15	
-5	0	0	15	← Coordonnées du point B

Pour la phase de vol à vitesse constante entre B et C, voici les résultats :

Tableau 5.2 La phase à vitesse constante dans le premier cas

X	Y	Z	Vitesse	
-5	0	0	15	← Coordonnées du point B
-4.54167	0.488802	0	15	
...				
-4.08333	0.962425	0	15	
5.541667	8.678045	0	15	
6	9	0	15	← Coordonnées du point C

Dans la phase de décélération, entre C et D, la constatation est celle attendue : la vitesse est égale à 15 unités de vitesse jusqu'au point de décélération. À partir de ce point, la vitesse diminue jusqu'à la vitesse cible de 14 unités de vitesse.

Tableau 5.3 La phase de décélération dans le premier cas

X	Y	Z	Vitesse
6	9	0	15
6.333333	9.23351	0	15
...			
7.666667	10.14624	0	15
8	10.36904	0	14.9846
8.333333	10.58968	0	14.93295
8.666667	10.80815	0	14.88097
9	11.02442	0	14.82865
9.333333	11.2385	0	14.77598
9.666667	11.45038	0	14.72297
10	11.66004	0	14.6696
10.33333	11.86748	0	14.61588
...			
12	12.87102	0	14.34174
12.33333	13.06493	0	14.28578
12.66667	13.25656	0	14.22942
13	13.44588	0	14.17267
13.33333	13.63291	0	14.11552
13.66667	13.81762	0	14.05797
14	14	0	14

← Coordonnées du point C

← La décélération débute à ce point

← Coordonnées du point D

Bien entendu, ce qui n'est pour le moment que des « unités de mesure » deviendra de vraies unités dans la partie consacrée à la taille réelle, le chapitre 7 de ce mémoire.

5.4 Second cas

5.4.1 Mise en place

Pour ce problème, les points considérés sont disposés dans un espace en 3D dont les limites sont $X_{\min} = -15$ unités, $X_{\max} = 15$ unités, $Y_{\min} = -15$ unités, $Y_{\max} = 15$ unités, $Z_{\min} = -15$ unités et $Z_{\max} = 15$ unités. La vitesse est donnée en unités de vitesse, comme nous l'expliquons dans l'introduction de ce mémoire. Les *waypoints* ont les coordonnées (X, Y, Z, Vitesse de passage) suivantes : **A**(-14, -14, -2, 14), **B**(-5, -6, -2, 15), **C**(0, 0, 0, 15), **D**(10, 12, 1, 17) et **E**(14, 14, 1, 16.5). Le taux de variation de vitesse est donné en unités d'accélération. Il y aura une NFZ sur la trajectoire étudiée avec les coordonnées (X, Y, Z, r), que nous définissons comme **NFZ₁** (-9, -10, -2, 1).

5.4.2 Résolution

Voici la trajectoire dessinée par l'algorithme :

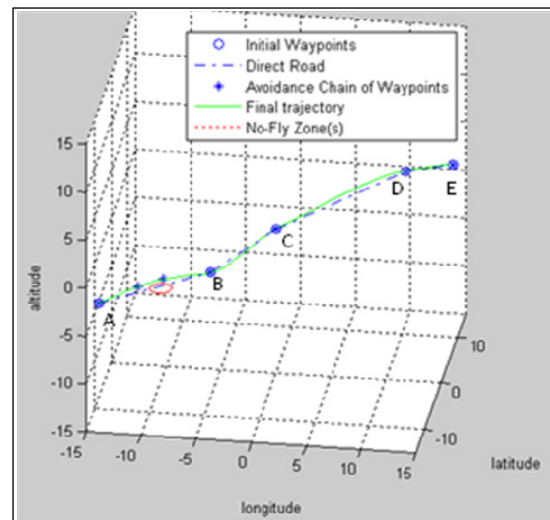


Figure 5.4 Trajectoire finale après l'évitement de zone dans le 2nd cas.

Voici le résumé du tableau des vitesses :

Tableau 5.4 Tableau de la vitesse en fonction de la position

X	Y	Z	Vitesse
-14	-14	-2	14
-13.7692	-13.6497	-2	14.08863
... (accélération)			
-11	-10	-2	14.99944
-10.8462	-9.82548	-2	15
... (vitesse constante à 15 unités de vitesses)			
-5.30769	-6.20084	-2	15
-5	-6	-2	15
... (vitesse constante à 15 unités de vitesse)			
0	0	0	15
0.769231	0.952208	0.129202	15.25365
... (accélération)			
6.153846	8.161432	0.826062	16.93793
6.923077	9.094219	0.886599	17
... (vitesse constante à 17 unités de vitesse)			
10	12	1	17
... (vitesse constante à 17 unités de vitesse)			
11.53846	12.93916	1	17
11.84615	13.10405	1	16.94405
... (décélération)			
13.69231	13.90223	1	16.56626
14	14	1	16.5

Encore une fois, il est possible de constater que les profils d'accélération et de décélération suivent la logique développée au début de ce chapitre.

CHAPITRE 6

LES AIDES AU PILOTAGE ET LES FONCTIONS SUPPLÉMENTAIRES

6.1 Préambule

A ce stade du travail, le système est fonctionnel en quatre dimensions. C'est-à-dire que l'algorithme développé est capable de détecter les NFZ sur une trajectoire initiale, de proposer une route d'évitement qui est la plus courte possible et de trouver la vitesse en chaque point de cette trajectoire.

Pour rendre le système plus fonctionnel et permettre de l'appliquer sur des cas réels, il a été décidé de lui apporter des fonctions supplémentaires. Ainsi, il sera possible d'utiliser ce travail comme la base de création d'un logiciel de génération ou de validation de données de vol. Les fonctions ajoutées calculent les caractéristiques suivantes :

- La distance parcourue en tous points,
- Calcul du temps de vol réalisé en tous points,
- L'heure estimée d'arrivée (Estimated Time Arrival, ETA),
- La consommation de carburant,
- La variation de la masse,
- La poussée nécessaire en tous points.

Dans ce chapitre, chacune de ces fonctions sera expliquée et son intérêt sera justifié.

6.2 Les fonctions supplémentaires

6.2.1 Le calcul de la distance parcourue

Ce calcul est très important car le résultat est une entrée pour le calcul de plusieurs autres fonctions. La méthode de calcul de la longueur est décrite dans le chapitre précédent : la trajectoire finale est discrétisée de manière très dense. Ainsi, la distance entre deux points est très faible et la trajectoire entre deux points est considérée comme une droite. Cette hypothèse se vérifie d'autant mieux que la discrétisation est dense, c'est pourquoi l'échantillonnage a été augmenté à plus de 1,000 points entre deux *waypoints* consécutifs. Dans l'utilisation en taille réelle qui est à réaliser dans le chapitre 7 (qui sera un vol trans-Atlantique), la distance entre deux *waypoints* est toujours inférieure (<) à 1,000 km, ce qui signifie qu'entre deux points consécutifs de la trajectoire échantillonnée, la distance est toujours inférieure à 1 km due à l'échantillonnage de 1,000 points.

6.2.2 Le calcul du temps de vol en tous points

La vitesse et la distance entre chaque point étant connue, nous sommes en mesure de calculer le temps de vol effectué en tous points. Ceci est nécessaire pour les fonctions suivantes, en particulier pour l'estimation de l'heure d'arrivée.

6.2.3 Estimation de l'heure d'arrivée

Comme nous l'avons précisé, le respect du RTA est très important dans l'exploitation aéronautique. Les FMS actuels incorporent déjà cette fonction et elle reste nécessaire à la bonne exploitation des appareils.

Dans le rapport EEC Report n408 intitulé *4D Trajectory Management Pilot Simulation*, écrit en juin 2008 [5], EUROCONTROL décrit le fonctionnement des FMS utilisant l'heure estimée d'arrivée (page 6). Nous y apprenons qu'une erreur est déclenchée quand l'Estimated Time Arrival (ETA) sort de la fenêtre prédéfinie par le FMS de

[-2 min; +2 min] par rapport au RTA. La figure suivante, extraite de ce même rapport (page 6), illustre ces propos.



Figure 6.1 : Explication du visuel du système d'alarme RTA ERROR.

Tiré de EEC Report Num 408 (2008, p.6)

Sur la figure 6-1, la fenêtre de gauche montre un vol dont le ETA reste dans la fenêtre de [-2 min; +2 min] par rapport au RTA, plus précisément le RTA est à 00 :43 :00 et que l'ETA est à 00 :43 :15. Aucune alarme ni aucun avertissement n'est déclenché. Au contraire, la fenêtre de droite montre un vol donc l'ETA sort de la fenêtre de tolérance. Le RTA est toujours de 00 :43 :00 mais l'ETA est cette fois de 00 : 40 : 15. Un avertissement RTA ERROR apparaît alors. Le système indique également le temps de retard estimé par rapport au RTA.

Dans notre programme, le calcul de l'ETA a donc été mis en place (avec pour entrée l'heure de départ et le temps total de vol prévu) ainsi qu'une fonction de RTA_error qui donne l'erreur entre le RTA et l'ETA et qui précise s'il s'agit de l'avance ou de retard dans le temps de vol.

6.2.4 La consommation du carburant

Dans le but d'optimiser le vol en terme de coût mais également de pollution, il est important de calculer la consommation de carburant. La décision a été prise de travailler avec un avion de type Boeing 747-200 répondant aux données du constructeur. Le site internet de Boeing [21] communique les données suivantes :

- Capacité maximale d'emport de carburant: 199,158 litres;
- Rayon d'action maximale: 12,700km;
- Vitesse à 35,000ft (11 000m) : Nombre de Mach égal à 0.84 soit environ 933 km/h.

Les estimations se basent sur les formules suivantes:

$$\text{Consommation (litres/km)} = \text{Emport maximal de carburant} / \text{Rayon d'action maximal}$$

$$\text{Consommation (litres/secondes)} = \text{Consommation (litres/km)} * \text{Vitesse (km/h)} / 3600$$

Le calcul donne une consommation kilométrique de 15.6 litres/km ce qui équivaut une consommation d'environ 4 litres/s. Grâce à la connaissance du temps de vol, nous pouvons calculer la consommation totale du carburant.

6.2.5 Estimation de la variation de masse

Lorsque la consommation de carburant est estimée, la variation de la masse de l'appareil est aussi estimée. La masse volumique du kérosène à 35, 000 ft est située entre 0.755kg/litres et 0.840kg/litres La référence prise pour ce travail est la moyenne de ces deux valeurs des masses volumiques qui est égale à 0.797kg/litres A chaque point, nous évaluons la variation de la masse de l'appareil en effectuant l'hypothèse que cette variation n'est due qu'à la consommation du carburant.

6.2.6 Calcul de la poussée nécessaire

Le calcul de la poussée nécessaire apporte des informations utiles pour le contrôle, même si elles ne seront pas renvoyées au pilote. Le pilote se concentre sur la vitesse et non pas sur sa conversion en poussée. Les paramètres les plus importants dans le vol de l'avion sont les suivants: la traînée (*drag* en N), la poussée (*thrust* en N), le coefficient de traînée (C_x non-dimensionnel), la surface alaire de l'avion (S en m^2), la vitesse (V en km/h), la masse volumique de l'air à l'altitude de croisière $\rho(h)$ et γ l'angle d'attaque de l'avion. Les trois cas à traiter sont: vol en palier, en descente (Figure 6-2) et en montée (Figure 6-3).

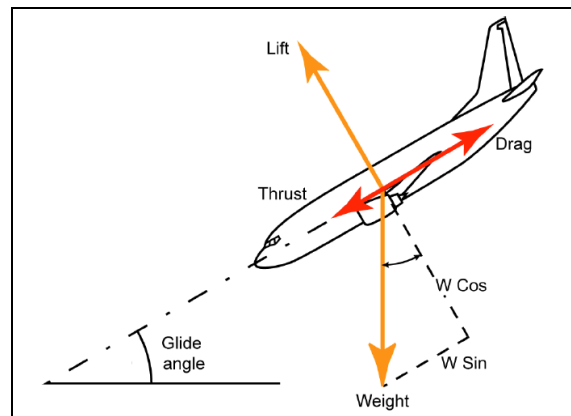


Figure 6.2 Forces appliquées à l'avion en descente.

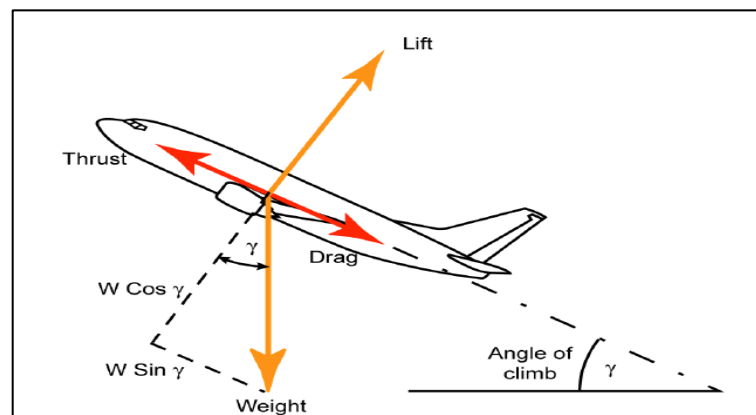


Figure 6.3 Forces appliquées à l'avion en montée.

6.3 Validation des fonctions créées

Pour cette partie, les points de cheminement ont été pris au hasard mais représentent des distances réelles. À partir de maintenant, les distances sont exprimées en kilomètres (km), les durées en heures (h), les vitesses en kilomètres par heure (km.h^{-1}), les accélérations en kilomètres / heures / secondes ($\text{km.h}^{-1}.\text{s}^{-1}$), les masses en kilogrammes (kg) ou en tonnes (T) et les consommations en litres.

Le profil de vitesse a été choisi avec attention. En effet, il y aura une accélération, une décélération et une phase de vol à vitesse constante. Ces paramètres ont été choisis dans le but de tester l'adaptation automatique de la vitesse utilisant des données en taille réelle. Par ailleurs, le terme de hauteur ne sera plus utilisé et il sera remplacé par la notion d'altitude.

Nous allons commencer par valider toutes les fonctions supplémentaires, en utilisant les données fournies par la compagnie Boeing sur son site internet [19].

Pour valider les fonctions précédentes, un test en « taille réelle » a été mené. La latitude varie de 0 à 6,000 km, la longitude de 400 à 1,000 km et l'altitude est constante à 11 km. Voici le tableau de position des *waypoints* : **A**(0, 400, 11), **B**(780, 530, 11), **C**(1332, 660, 11), **D**(1930, 800, 11), **E**(2630, 900, 11), **F**(3523, 1000, 11), **G**(4000, 970, 11), **H**(4750, 950, 11), **I**(5600, 940, 11) et **J**(6000, 934, 11).

Dans la page suivante, les données entrées par l'opérateur sont le profil d'accélération, les vitesses requises aux *waypoints*, la consommation de l'appareil, l'heure de départ et le RTA. Dans la réalité, certaines de ces données seront fournies par d'autres systèmes embarqués. Les autres valeurs qui apparaissent sont calculées automatiquement par le programme.

The length of the trajectory is 6061.7852 km

Enter the plane acceleration profil (km/h/s): 35

Enter the Waypoint Required Speed: 880

Enter the Waypoint Required Speed: 880

Enter the Waypoint Required Speed: 890

Enter the Waypoint Required Speed: 890

Enter the Waypoint Required Speed: 890

Enter the Waypoint Required Speed: 890

Enter the Waypoint Required Speed: 890

Enter the Waypoint Required Speed: 890

Enter the Waypoint Required Speed: 880

Enter the Waypoint Required Speed: 880

What is the airplane fuel consumption (l/s)? 4

The total flight time is 6h 49min 45s

Enter the departure hour: 12

Enter the departure minute: 04

The Estimated Time Arrival is 18h 53min 45s

Enter the Required Time Arrival hour: 18

Enter the Required Time Arrival minute: 55

RTA Error is 0h 1min 15s with advance

Do you want to save the flight data (Yes: press 1, No: press any other key)? 1

Data are available in fly_data.xls

Voici un extrait de la matrice fly_data.xls qui enregistre les données de vol de l'appareil pour chacune des fonctions calculées automatiquement. L'altitude est constante à 11 km.

Tableau 6.1 Extrait de la matrice des données de vols recueillies automatiquement

Dist. (km)	Lat. (km)	Long. (km)	Vitesse (km/h)	Temps de vol (h)	Temps de vol restant (h)	Fuel cons. (litres)	Variation du poids (kg)
0	0	400	880	0	6.8294129	0	0
0.79	0.78	400.10	880	0.0009	6.82895789	12.88	10.26
1.57	1.56	400.20	880	0.0018	6.82850283	25.76	20.53
2.36	2.34	400.30	880	0.0027	6.82804777	38.63	30.79
3.15	3.12	400.39	880	0.0036	6.82759271	51.51	41.06
790.12	779.22	529.85	880	0.8979	6.37527127	12929.26	10304.62
790.92	780	530	880	0.8988	6.37481623	12942.28	10315
791.48	780.55	530.11	880.02	0.8994	6.37384926	12951.50	10322.34
792.04	781.11	530.22	880.05	0.9001	6.37288234	12960.71	10329.69
792.61	781.66	530.33	880.07	0.9007	6.37191545	12969.93	10337.03
1042.91	1025.89	585.05	889.97	1.1835	5.94775388	17042.53	13582.89
1043.47	1026.44	585.18	889.99	1.1841	5.94679782	17051.73	13590.23
1044.04	1026.99	585.32	890	1.1848	5.94584176	17060.93	13597.56
1044.61	1027.54	585.45	890	1.19	5.94488570	17070.13	13604.89
1752.60	1716.3	749.26	890	1.98	4.88	28525.15	22734.55
4061.39	4000	970	890	4.5751	2.22770030	65880.99	52507.15
5061.85	5000.15	946.56	890	5.6993	1.34977205	82068.14	65408.31
5408.16	5346.45	942.94	890	6.0883	1.08951234	87671.45	69874.14
5409.02	5347.30	942.93	889.99	6.0893	1.08887249	87685.21	69885.12
5660.03	5598.30	940.02	880.07	6.3729	0.90004958	91769.50	73140.3
5660.88	5599.15	940.01	880.03	6.3739	0.89940960	91783.43	73151.39
5661.74	5600	940	880	6.3748	0.89876962	91797.35	73162.49

Les valeurs sont approximées pour nos interprétations. Les chiffres significatifs sont gardés pour éviter les erreurs d'approximations successives.

Nous pouvons maintenant établir la cohérence entre les données et la réalité. La vitesse moyenne est 887,63 km/h, ce qui donne un vol d'une durée d'environ 6,83 heures soit de 6 heures et 50 minutes, ce qui est cohérent avec ce que le programme propose.

La consommation de carburant est de 78,38 tonnes, ce qui est cohérent pour un vol de type transatlantique. En effet, 98,000 litres de carburant sont utilisés pour un vol de 6,000 km. Le rayon d'action maximal du Boeing 747-200 est de 12,700 km et son emport maximal de carburant est de 199,158 litres. Encore une fois, ceci tend à montrer la cohérence des résultats: la moitié de l'emport maximal de carburant est nécessaire pour parcourir la moitié du rayon d'action maximal.

Pour valider le calcul de la poussée, nous utilisons les entrées suivantes :

La masse est de 370 T (qui sera entrée en Newton), le coefficient de trainée (C_x) utilisé est de 0.075, la surface alaire est de 511m^2 et la masse volumique de l'air qui est obtenue par la relation des gaz parfaits valable dans la troposphère: $\rho = p/RT$ où R est la constante des gaz parfaits et qui vaut $287.1\text{ J}\cdot\text{mol}^{-1}\cdot\text{K}^{-1}$, la pression est exprimée en Pascal à l'altitude choisie et la température est donnée en Kelvin à l'altitude choisie. Pour une altitude de 11,000 m, les valeurs sont 22632 Pa pour la pression et 216.65 K pour la température. La formule précédente permet d'obtenir la masse volumique= 0.3639 kg/m^3 .

Les valeurs obtenues sont comprises entre 416,670 N et 426,200 N pour un vol en palier à une vitesse minimale de 880 km/h et maximale de 890 km/h. Le Boeing 747-200 étant motorisé par quatre réacteurs développant entre 230 kN et 240 kN, ces valeurs sont tout à fait cohérentes. Il sera montré dans la suite de ce travail que la dépendance entre la poussée et la masse de l'appareil est prise en compte dans l'algorithme de calcul.

CHAPITRE 7

VALIDATION DES TRAVAUX

7.1 Méthode de validation

Afin de valider nos hypothèses de travail et de juger de la cohérence des données obtenues par les différentes fonctions de l'algorithme, un trajet entre Paris et Montréal sera simulé. Ce trajet contiendra deux NFZ problématiques (c'est-à-dire qu'elles seront sur la trajectoire, donc à éviter), et deux NFZ non-problématiques (c'est-à-dire qu'elles ne seront pas sur la trajectoire, donc ne devront entraîner aucune modification de route). Ces zones sont fictives mais peuvent être considérées comme des phénomènes météorologiques ou encore des conflits aériens à éviter

Une première NFZ non-problématique sera placée au dessous de la trajectoire de vol, telle que si elle était à la même altitude que l'avion, elle devrait être évitée, et l'autre NFZ sera à la même altitude que l'avion, mais pas sur la trajectoire, ce qui contribuera à montrer la robustesse de l'évitement de zones : l'avion ne doit pas éviter une zone « non-gênante ».

Par ailleurs, pour valider que la trajectoire entre deux échantillons successifs est considérée comme une droite, la distance moyenne entre deux points sera donnée en pourcentage par rapport à la longueur totale de la trajectoire. Il en sera de même pour valider l'hypothèse selon laquelle les variations d'altitude sont faibles.

Enfin, le temps de vol sera comparé aux temps de vol réels trouvés sur les sites Internet d'Air-France [20] et d'Air-Canada [21]. Les données réelles en termes de carburant consommé et de poussée seront comparées aux données fournies par Boeing [19].

Les coordonnées des *waypoints* (Abscisse, Ordonnée, Altitude, Vitesse de passage) mesurées en km sont les suivants : **A**(0, 375, 11, 900), **B**(428, 478.5, 11, 900), **C**(1500, 600, 11, 930), **D**(2100, 600, 11.3, 933), **E**(3214, 600, 11.3, 933), **F**(4200, 500, 11.3, 933), **G**(5142, 357,

11.3, 910), **H**(5535, 214, 11, 900), **I**(5678, 142.8, 10.5, 900) et **J**(5857, 0, 10, 900). Les coordonnées des quatre NFZ (Abscisse du centre, Ordonnée du centre, Altitude du centre, rayon), données en km sont les suivantes : **NFZ₁**(964, 535, 11, 20), **NFZ₂**(2700, 600, 11.3, 10), **NFZ₃**(4800, 400, 5, 20) et **NFZ₄**(200, 25, 11, 10).

7.2 Résultats et validation

Voici le tracé de la trajectoire obtenue dans le plan de l'altitude versus la longitude :

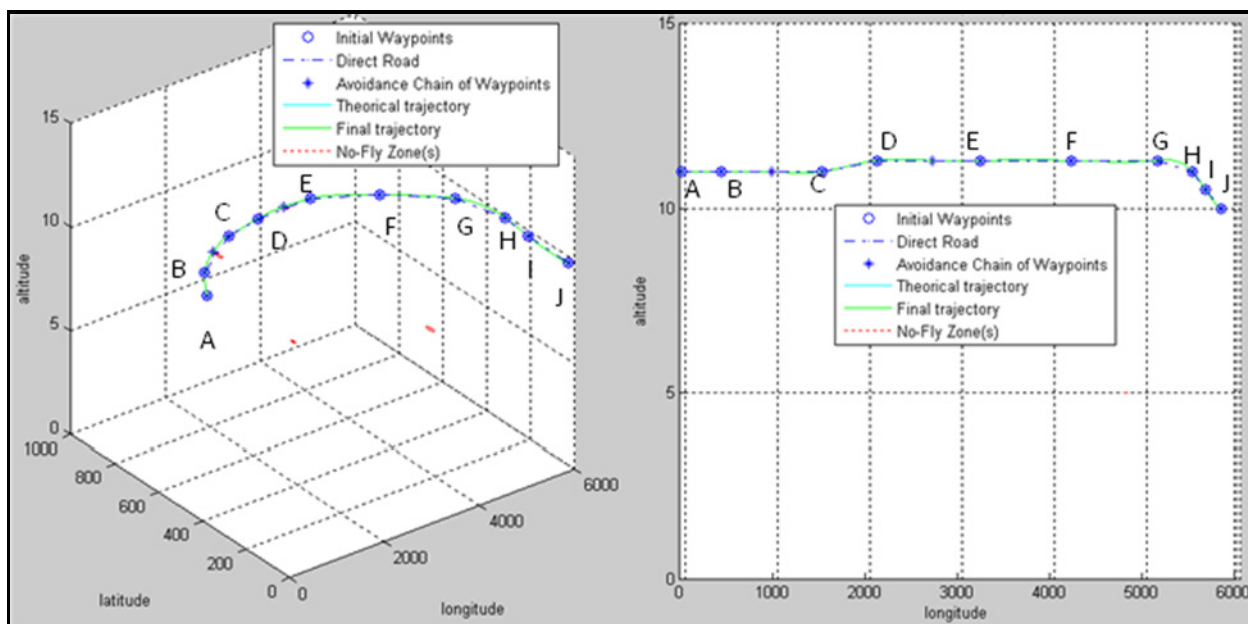


Figure 7.1 Trajectoire finale dans le plan de l'altitude versus la longitude du vol.

Dans le plan de la latitude versus la longitude, nous obtenons la figure suivante :

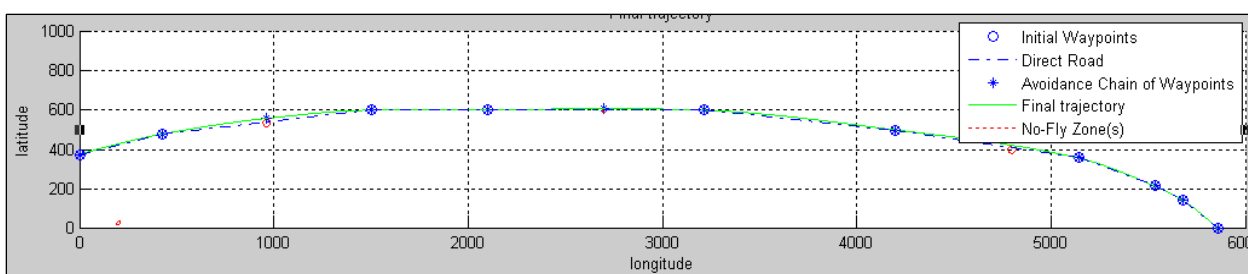


Figure 7.2 Le vol de validation dans le plan de la latitude versus la longitude.

Encore une fois, dans les schémas ci-dessus, les étoiles cerclées représentent les points de cheminement du plan de vol initial. Les cercles en pointillé sont les NFZ et les étoiles sont les points de la chaîne de *waypoints* comprenant les *waypoints* initiaux et les *waypoints* d'évitement. La trajectoire en pointillés est la trajectoire théorique. Elle relie tous les points par des droites et passe à travers les NFZ. La trajectoire en traits pleins, elle, est la trajectoire d'évitement la plus courte trouvée par l'algorithme.

La durée de vol est de 6 heures 29 minutes et 39 secondes. Les compagnies aériennes Air France et Air Canada vendent le vol avec une durée d'environ 6h45 (AF344 décolle à 13h30 et se pose à 15h15 avec 6 heures de décalage horaire et AC871 décolle à 13h40 pour se poser à 15h20) bloc-bloc, c'est-à-dire entre le moment où le stationnement de départ est quitté jusqu'au moment où le stationnement d'arrivée est atteint, ce qui comprend le roulage aux aéroports, la montée initiale et la descente. Notre avion est modélisé à la vitesse de croisière entre Paris et Montréal, cependant, le temps simulé reste assez proche du temps réel pour valider la durée du vol de l'avion.

Sur la figure 7-2 montrant le plan de la latitude versus la longitude, on voit que la trajectoire est tout à fait viable, et cela reste de même sur le figure 7-1 montrant le plan de l'altitude versus la longitude. Il faut faire attention aux échelles pour cette dernière figure.

La matrice *fly_data*, qui enregistre les données de vol de la simulation en chaque point de la trajectoire, contient 10,990 points pour modéliser une trajectoire de 5,992.3 km. Cela donne une précision inférieure (<) à un point tous les 550 m. Il y a un point tout les 0.01% de trajectoire. Les angles de variation étant très faibles, l'hypothèse selon laquelle la trajectoire entre deux points peut être considérée comme une droite est valide. Il en est de même pour les variations d'altitude inférieures (<) à 1 km à 0.01% de la trajectoire.

L'indexation de la variation du poids au calculateur de poussée est bien effective comme on peut le constater sur les dernières ligne de *fly_data*. À une vitesse constante, la poussée nécessaire diminue avec le temps.

Tableau 7.1 Résultats obtenus pour la variation de poids, de la poussée et de la consommation de carburant

Distance parcourue (km)		Vitesse (km/h)		Consommation de carburant (litres)	Variation de poids(kg)	Poussée (N)
5,976.09		900		93258.13	74326.73	439755.7
5,976.34	...	900	...	93262.09	74329.89	439742.5
5,976.59		900		93266.05	74333.04	439729.4
5,976.83		900		93270.01	74336.19	439716.3
5,977.08		900		93273.96	74339.35	439703.1
5,977.33		900		93277.92	74342.51	439689.9

La consommation de carburant reste également cohérente puisqu'elle est quelque peu inférieure à la capacité d'emport maximal pour une distance parcourue inférieure à la moitié du rayon d'action maximal.

Le calculateur de l'ETA est aussi correct ainsi que celui du RTA_error tel montré sur les deux exemples suivants :

Exemple 1

The total flight time is 6h 29min 39s
 Enter the departure hour : 10
 Enter the departure minute : 00
 The Estimated Time Arrival is 16h 29min 39s
 Enter the Required Time Arrival hour : 17
 Enter the Required Time Arrival minute : 22
 RTA Error is 0h 52min 44s with advance

Exemple 2

The total flight time is 6h 29min 39s
Enter the departure hour : 10
Enter the departure minute : 00
The Estimated Time Arrival is 16h 29min 39s
Enter the Required Time Arrival hour : 15
Enter the Required Time Arrival minute : 00
RTA Error is 1h 29min 16s with delay

Il est visible que le calcul de l'ETA prend bien en compte l'heure de départ et le temps de vol. Il est également démontré que l'estimation du RTA dans les deux directions (ETA < RTA ou ETA > RTA) est validée.

CONCLUSION

En conclusion de ce mémoire, nous pouvons dire que la démarche scientifique a été efficace. Il est certes plus long de programmer d'abord des cas simples puis de les complexifier par la suite que de se confronter tout de suite à un problème difficile, mais le résultat montre que la stratégie est efficace.

La méthode de travail

Pour résumer cette stratégie, les points clés sont les suivants :

- codage des algorithmes principaux;
- travail en 2D avec des unités arbitraires;
- travail en 3D avec des unités arbitraires;
- travail en 4D avec des unités arbitraires;
- travail en 4D avec des valeurs réelles.

Chaque étape s'est conclue avec une étape de validation dont quelques exemples ont été retranscrits dans cette étude. Ainsi, le suivi de progression était très simple et quand une erreur apparaissait, des bases justes et validées étaient disponibles pour la rechercher et la corriger.

Au final, le résultat de ce travail est un logiciel de suivi et d'optimisation automatique de trajectoire avec l'évitement autonome des NFZ. Ce logiciel propose aussi le calcul de plusieurs données de vols importantes (comme la vitesse, l'ETA ou encore la consommation de carburant) qui peuvent être utilisées pour le suivi des vols en temps réel, le calcul de l'indice du coût (*Cost Index*) ou pour servir d'entrées à d'autres systèmes embarqués de l'avion.

Les apports de la recherche

L'apport principal est le programme créé. Ce logiciel a été validé pour le Boeing 747-200 mais il peut être utilisé pour n'importe quel avion dont les paramètres nécessaires sont connus.

Il permet donc au laboratoire LARCASE d'avoir accès à une plateforme de tests supplémentaires. Par ailleurs, le code Matlab est convertible automatiquement sur d'autres plateformes ce qui rend ce logiciel encore plus intéressant. En effet, il pourra être converti en C ou en FLSim afin d'être implémenté sur les simulateurs de vol du laboratoire LARCASE.

Un autre apport est la méthode de méta-heuristique créée pour résoudre le problème d'optimisation. Même si elle est une combinaison de deux méthodes connues et maîtrisées, elle offre l'avantage de présenter une alternative aux autres méthodes existantes en combinant les avantages de ces deux méthodes.

Par ailleurs, c'est la première fois que le laboratoire LARCASE utilise la méthode méta-heuristique à recherche de solution locale qui n'est pas basée sur des populations de solutions comme l'algorithme génétique. Ce travail lui offre une expérience dans le domaine et met une méthode fonctionnelle à disposition des projets à venir.

Ce travail s'inscrit aussi dans l'effort de développement durable que poursuit le secteur aéronautique depuis plusieurs années. La réduction des distances parcourues entraîne une baisse des émissions de gaz à effet de serre et l'évitement automatique des zones permettra de diminuer l'impact des nuisances sonores.

Enfin, ce travail apporte sa pierre à l'édifice de la recherche sur le free-flight et offre de nouveaux axes de réflexions quant à cette notion. Nous avons montré que la grande majorité des cas d'évitement pouvaient être traités de manière automatique.

Tous ces travaux représentent ma contribution dans la présente recherche.

Cependant, pour que le projet soit implémentable sur les avions, il faut disposer d'un système de communication entre les appareils qui soit très performant pour que tous les avions d'une zone soient informés des changements de trajectoires des autres avions de cette même zone (cela renvoie au travail sur la sectorisation de l'espace aérien comme ce qui est proposé par Charles-Edmond Bichot [15]). Il faudra aussi avoir un puissant algorithme de résolution de conflits en route qui est une autre partie du thème « Amélioration du FMS par l'ajout de fonction » développée en ce moment même au LARCASE par Romain Brisemeur.

Les travaux futurs

Bien entendu, ce devoir ouvre également la porte à bien d'autres travaux qui pourront l'améliorer. En voici les principaux.

La première chose va être d'améliorer le système de détection du plus court chemin. Comme nous le disions dans le chapitre consacré à cet algorithme, il est facile de densifier la recherche du plus court chemin. Plusieurs algorithmes qui ont été développés parallèlement à ce travail existent déjà au LARCASE. Un algorithme de type A* appliqué au système en trois dimensions (il faudra l'appliquer prochainement en 4D) est déjà prêt, ainsi qu'une méthode d'algorithme génétique en 4D. Le laboratoire pourra aussi utiliser son expertise en réseaux de neurones ou encore en logique floue.

Il pourrait également être utile d'améliorer le système d'évitement de zones. Il est encore possible de créer des trajectoires plus courtes durant un évitement, par exemple, en considérant tous les points d'évitement déjà proposés plutôt que ceux de la dernière itération.

Enfin, il pourrait être intéressant de coupler ce logiciel avec celui développé parallèlement au LARCASE, toujours par Romain Brisemeur et qui concerne l'évitement latéral des conflits

aériens en route. Étudier le cas de deux avions sur des trajectoires qui deviennent séquentes à cause d'un évitement de zone pourrait être intéressant.

En effet, si nous parvenons à ce que plusieurs avions suivent chacun leur trajectoire en évitant automatiquement les NFZ et en gérant de manière autonome leurs conflits en route, nous pourrions contribuer à une grande avancée dans la théorie du free-flight.

D'autres travaux qui demandent plus d'implication seront détaillés dans la section sur les recommandations qui suivra le chapitre des conclusions.

Après une année de recherche sur ce travail, où l'équipe du LARCASE travaillant sur le FMS et moi-même avons traversé toutes les étapes d'un projet en partant de l'identification des besoins de l'utilisateur, de l'idéation des projets jusqu'à leur réalisation et en passant par les présentations aux différents organismes, nous pouvons dire que l'expérience vécue est à la hauteur de nos attentes. Les résultats sont ceux que nous espérions et sont même plus encourageants que nous l'attendions. Quand les différents projets du LARCASE concernant le FMS seront rassemblés, il ne fait pas de doute que plusieurs innovations seront proposées aux industriels et nous avons la conviction que certaines seront retenues.

Le LARCASE a donc à sa disposition une base de connaissance et de recherche qui ne demande qu'à être exploitée et, dans la partie suivante, nous proposerons plusieurs idées de développements futurs qui pourraient continuer dans la même direction que les recherches menées tant par la FAA que par EUROCONTROL : les deux grands organismes de contrôle et de management du trafic aérien. Le LARCASE pourra ainsi se doter de projets qui seront à la pointe de la technologie et de l'avancement technique dans ce domaine de l'aéronautique

RECOMMANDATIONS

Le premier travail pourrait être la validation du logiciel sur d'autres avions et FMS afin de le rendre toujours plus utile. Il en va de même avec l'ajout de nouvelles sorties supplémentaires.

La mise en place d'un système de détection de forme à partir des images radar afin de créer des NFZ les plus petites possibles peut être considérée. L'automatisation de la détection des NFZ sera mise en place, ainsi que l'amélioration de la méthode d'évitement de NFZ en fonction de ces systèmes.

Comme nous le disions, ce travail implique le développement et la mise en place d'un système de communication performant entre les appareils.

Enfin, une bonne piste pour l'amélioration de ce système sera de boucler l'algorithme afin d'asservir le tout selon un indice de coût (*Cost Index*) demandé. En effet, nous avons fait le choix d'optimiser selon la plus courte distance, mais maintenant que le calcul du RTA, de la consommation de carburant et du temps de vol est implémenté, il est possible d'optimiser la philosophie même du logiciel.

Grâce à la connaissance du temps de vol et à la consommation de carburant, il devient possible de calculer le prix d'exploitation (en trouvant aisément des données comme le salaire de l'équipage et le prix de revient d'un vol). La connaissance du RTA permet de quantifier le temps de retard et il est possible donc de le joindre au coût horaire du retard. En connaissant ces valeurs, il devient possible de créer une variante du présent logiciel qui permettrait de trouver le chemin le plus efficace en fonction d'un *Cost Index* imposé par l'opérateur.

LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES

- [1] IATA. 2009. «Press release n°52: Continued Traffic Improvement - But Much Ground Still to be recovered». In Le site de l'IATA. En ligne. <<http://www.iata.org/pressroom/>>. Consulté le 3 Janvier 2010.
- [2] NAVCANADA. 2010. «Circulaires d'information aéronautique». En ligne. 140p. <<http://www.navcanada.ca>>. Consulté le 3 Janvier 2010.
- [3] Géraud Granger (2002). «Détection et résolution de conflits aériens : modélisations et analyse», Thèse de Doctorat, Toulouse, France, ENAC, 152p.
- [4] Vu N. Duong et Eric G. Hoffman. 1997. «Conflict resolution advisory service in autonomous aircraft operations», Digital Avionics Systems Conference, 16th DASC., AIAA/IEEE vol°2 , p 9.3-10, 9.3-17.
- [5] EUROCONTROL EXPERIMENTAL CENTER. 2008. 4D Trajectory management pilot simulation, EEC Report n° 408
- [6] Lidèn Sam. 1992. «Optimum 4-D guidance for long flights», Digital Avionics Systems Conference, IEEE/AIAA 11th. p262-267.
- [7] Le Falbek Yann. 1999. «Prévision de trajectoires d'avions par réseaux de neurones», Thèse de Doctorat, Toulouse, ENSEEIHT, 270p.
- [8] Yokoyama Nobuhiro. 2001. «Flight Trajectory Optimization Using Genetic Algorithm Combined with Gradient Method», ITEM, Volume 1, No 1, Paper 10.
- [9] Vormer, Frizo Jan-Peter. 2005. «Theoretical and operational aspects of optimal airport arrival trajectories», Thèse de doctorat. Technische Universiteit Delft, Pays-Bas, 253 p.
- [10] Weber, R. et Cruck, E. 2007. «Subliminal ATC Utilizing 4D Trajectory Negotiation» Integrated Communications, Navigation and Surveillance Conference. p. 1-9
- [11] Alech Nicolas. 1994. «Construction de trajectoires d'évitement optimales par algorithmes génétiques», Rapport de stage, Toulouse, ENSEEIHT, 59p.
- [12] Bosc Jean-François. 1997. «Techniques d'évitement réactif et simulation du trafic aérien», Thèse de Doctorat, Toulouse, Institut polytechnique de Toulouse, 134p.
- [13] Zegal Karim. 1993. «A reactive approach for distributive air traffic control». In International conference on artificial intelligence & Expert systems.

- [14] Médioni Sébastien. 1998. « Méthodes d'optimisation pour l'évitement aérien : systèmes centralisés, systèmes embarqués», Thèse de doctorat, Paris, École Polytechnique, 230p.
- [15] Bichot Charles-Edouard. 2004. «Optimisation du découpage de l'espace aérien par diverses méta-heuristiques», Thèse de D.E.A., Toulouse, ENAC, 89p.
- [16] Ravelin Olivier. 2005. «Utilisations de méta-heuristiques pour la résolution en vitesse de conflits aériens», Mémoire de Maitrise, Toulouse, ENAC, 41p.
- [17] GERAD. 2007. «Métaheuristiques». En ligne. 12p.
<<http://www.gerad.ca/~alainh/Metaheuristiques.pdf>>. Consulté le 5 Juillet 2009.
- [18] The Matworks Inc. , The Matlab User's Guide.
- [19] The Boeing Company, 2010. « Technical Characteristics -- Boeing 747-400». In The Boeing Company. En ligne.
<http://www.boeing.com/commercial/747family/pf/pf_400_prod.html>. Consulté le 20 Novembre 2009
- [20] Air France, 2010. «Réservations en ligne». In Air France. En Ligne.
<www.airfrance.fr>. Consulté le 20 Novembre 2009
- [21] Air Canada, 2010. «Réservations en ligne». In Air Canada. En Ligne.
<www.aircanada.com>. Consulté le 20 Novembre 2009