ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE

FLIGHT DECK CENTERED COST EFFICIENT 4D TRAJECTORY PLANNING

M.Sc. THESIS

Mevlüt UZUN

Department of Aeronautics and Astronautics

January 2016

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE

FLIGHT DECK CENTERED COST EFFICIENT 4D TRAJECTORY PLANNING

M.Sc. THESIS

Mevlüt UZUN
(511131127)

Department of Aeronautics and Astronautics

Thesis Advisor: Prof. Dr. Gökhan İnalhan

January 2016

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

KOKPİT OTOMASYONU TABANLI 4D ROTA PLANLAMASI

YÜKSEK LİSANS TEZİ

Mevlüt UZUN
(511131127)

Uçak ve Uzay Mühendisliği

Tez Danışmanı: Prof. Dr. Gökhan İnalhan

Ocak 2016

Mevlüt UZUN, a M.Sc. student of ITU Graduate School of ScienceEngineering and Technology 511131127 successfully defended the thesis entitled “FLIGHT DECK CENTERED COST EFFICIENT 4D TRAJECTORY PLANNING ”, which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Prof. Dr. Gökhan İnalhan** ....
Istanbul Technical University

Jury Members : **Prof. Dr. Gökhan İnalhan** ....
Istanbul Technical University

Assist. Prof. Dr. Emre Koyuncu ....
Istanbul Technical University

Assoc. Prof. Dr. İlker Bekmezci ....
Air Force Academy

Date of Submission : 27 November 2015
Date of Defense : 25 December 2015

To my family,

FOREWORD

First of all, I would like to thank my advisor Emre Koyuncu for being an excellent mentor throughout my last three years. His views on how to perform research and write papers extended my vision and had a great influence on me. I would also like to thank Prof. Dr. Gokhan Inalhan for his limitless supports and being an idol for me. I learnt many things from his experiences and I thank him for bringing me into amazing research topics. I feel lucky for being a part of Control and Avionics family. Without my wonderful bandmates, my last seven years in ITU would have been much more stressful and overwhelming. I would like to thank Cüneyt Çalışkan for sticking with me at my music career and being a wonderful wise friend. His place is impossible to replace. I would also like to thank Melih Gücer for being an excellent bandmate and friend. They are the only guys that i can blether with. I am so glad to have strong relationships with Mehmet Hasır and Aykut Efe, who are my friends since childhood, and Gözde Kadıoğlu who is the smartest and most dedicated girl I have ever met. Finally, I would like to thank my precious family for their love and support. I am so lucky to have wonderful and beautiful two sisters. I would like to thank my younger sister Ayşenur for being a mother, mentor and friend to me. She is one of the top class dental surgeon in Turkey. I would also like to thank my older sister Tuba for being such a patient and polite woman and moralizing me at the hard times. An of course, I would like to thank my parents for being the most amazing parents ever. They always encouraged me to push the limits and supported me every single moment. I am where I am because of them...

January 2016

Mevlüt UZUN
Researcher Assistant

TABLE OF CONTENTS

Page

FOREWORD .... **ix**
TABLE OF CONTENTS .... **xi**
ABBREVIATIONS .... **xiii**
LIST OF TABLES .... **xv**
LIST OF FIGURES .... **xvii**
SUMMARY .... **xix**
ÖZET .... **xxi**
1. INTRODUCTION **1**
  1.1 Description of the Work and Motivation 2
  1.2 Literature Review 5
2. FLIGHT DECK TESTBED **11**
  2.1 Integrated Testbed: Flight Deck Simulator and ATM Testbed 13
  2.2 Next Generation Synthetic Vision Screens 15
  2.3 Augmented Reality Based Head-up Display 19
3. PROBLEM DEFINITION **23**
4. AIRCRAFT PERFORMANCE MODEL **25**
    4.0.1 Aircraft Trajectory Cost Definition 29
5. COST EFFICIENT LOCAL TRAJECTORY GENERATION **33**
  5.1 Cruise 35
  5.2 Climb 38
  5.3 Descent 40
6. TACTICAL 4D TRAJECTORY PLANNING: CONFLICT DETECTION AND RESOLUTION **45**
  6.1 Sampling-based Conflict Resolution 49
    6.1.1 Importance Sampling with Cross-Entropy 53
7. INTEGRATED SIMULATIONS **61**
8. CONCLUSION **69**
REFERENCES **71**
APPENDICES **77**
CURRICULUM VITAE **79**

ABBREVIATIONS

4DOD	: 4D Operational Display
ADS-B	: Automatic Dependent Surveillance - Broadcast
AI	: Aircraft Intent
ANSP	: Air Navigation Service Providers
APM	: Aircraft Performance Model
AFM	: Aerodynamic Forces and Configurations Model
ALM	: Aircraft Limitations Model
ATM	: Air Traffic Management
BADA	: Base of Aircraft Data
CAS	: Calibrated Airspeed
CD	: Conflict Detection
CE	: Cross Entropy
CDR	: Conflict Detection and Resolution
CI	: Cost Index
CMB	: Climb Phase
CPDL	: Controller-Pilot Data Link
CRZ	: Cruise Phase
DES	: Descend Phase
ECCF	: Economic Cruise Cost Function
EM	: Expectation Minimization
FI	: Flight Intent
FL	: Flight Level
FMS	: Flight Manageent System
FMC	: Flight Management Computer
GVA	: Vertical Accuracy
GMM	: Gaussian Mixture Model
HUD	: Head-up Display
LIDL	: Low Idle
MILP	: Mixed Integer Linear Programming
MDP	: Markov Decision Process
NACp	: Horizontal Position Accuracy
NACv	: Horizontal Velocity Accuracy
OPM	: Operation of Configuration Parameters Model
PFM	: Propulsion Forces Model
RBT	: Reference Business Trajectory
RRT	: Rapidly Exploring Random Trees
SESAR	: Single European Sky ATM Research
SVD	: Synthetic Vision Display
TAS	: True Airspeed
TCAS	: Traffic Collision Avoidance System
TP	: Predicted Trajectory
UAV	: Unmanned Aerial Vehicle
vDST	: Visual Decision Support Tools

LIST OF TABLES

Page

Table 4.1 : Equations of Trajectory Computation and Aircraft Performance Model 30
Table 5.1 : Maneuver library S_m for Maneuver Automaton 33
Table 7.1 : Scenario 1: Flight template and maneuver mode sequence in solution trajectory 62
Table 7.2 : Scenario 2: Flight template and maneuver mode sequence in solution trajectory 64
Table 7.3 : Scenario 3: Flight template and maneuver mode sequence in solution trajectory 67
Table 7.4 : Average CDR computation times for different multiple-thread scenarios 67

LIST OF FIGURES

Page

Figure 1.1 : B737 – 800 Flight Deck test platform with experimental visual decision support tools for future ATM realm: Head up Display (HUD), Synthetic Vision Display (SVD) and 4D Operational Display 2
Figure 1.2 : The envisioned data exchange and trajectory occurrence procedures for the future airspace needs 3
Figure 2.1 : Airspace Model and ATM Testbed for in-operation experiments 12
Figure 2.2 : Architecture of the integrated next generation flight deck system with novel add-on modules 13
Figure 2.3 : Software Architecture of the integrated System 15
Figure 2.4 : Synthetic Vision Display (SVD) and 4D Operational Display (4DOD) screens in the flight deck 16
Figure 2.5 : Definitions of the symbology in 4D Operational Display (4DOD) 17
Figure 2.6 : Definitions of the symbology in Synthetic Vision Display (SVD) 18
Figure 2.7 : Transparent Screen overlay for HUD augmented reality implementations 20
Figure 2.8 : Definitions of the symbology in Head-Up-Display (HUD) 21
Figure 4.1 : Aircraft local trajectory generation through the Aircraft Performance Model and parametric definitions in BADA 4 28
Figure 5.1 : Flight Template Automaton with *Cruise*, *Climb* or *Descent* templates. 34
Figure 5.2 : Cruise flight template automaton 36
Figure 5.3 : Economy cruise cost function vs. Mach profile for low CI and high CI values 37
Figure 5.4 : Effects of wind directions in evaluations of optimum cruise Mach profile 37
Figure 5.5 : *Climb* flight template automaton 38
Figure 5.6 : Cost function vs. Mach profile for *Climb* flight template with low and high *CI* values 40
Figure 5.7 : CAS and Mach profiles for the example climb action – climbing from FL170 to FL220 41
Figure 5.8 : Wind direction effect on optimum climb Mach $M_{tgt,cmb}$ 41
Figure 5.9 : *Descent* flight template automaton 42
Figure 5.10: Cost function vs. CAS profile for *Descent* flight template with low and high *CI* values 43
Figure 5.11: Mach and CAS profiles for the descent template – descending from FL300 to FL260 43
Figure 5.12: Wind direction effect on optimum descent CAS 44

Figure 6.1 : Ground perspective: conflict monitoring with flight intent and reachable sets associated with different performance models. 46
Figure 6.2 : Airborne perspective: conflict monitoring with flight intent exchange and ADS-B. 47
Figure 6.3 : *RRT** algorithm solutions are shown after 100, 600 and 1200 vertices generation respectively. 50
Figure 6.4 : Pseudo-random sampling and asymptotic convergence in *RRT** with 40, 120 and 400 samples 53
Figure 6.5 : Importance sampling strategy of CE and asymptotic convergence in *RRT** with 40, 120 and 400 samples in *RRT** 54
Figure 6.6 : Trajectory cost convergence with the number of vertices in pseudo-random sampling and CE sampling 59
Figure 6.7 : Computational effort with the number of vertices in pseudo-random sampling and CE sampling 59
Figure 7.1 : Conflict resolution trajectory for the multiple-thread Scenario 1...... 61
Figure 7.2 : Trajectory cost convergence curve of CR algorithm for the Scenario 1 61
Figure 7.3 : CAS and Mach profiles of conflict resolution trajectory to the Scenario 1 62
Figure 7.4 : Altitude profile of conflict resolution trajectory to the Scenario 1 63
Figure 7.5 : Conflict resolution trajectory for the multiple-thread Scenario 2...... 63
Figure 7.6 : Trajectory cost convergence curve of CR algorithm for the Scenario 2 64
Figure 7.7 : CAS and Mach profiles of conflict resolution trajectory to the Scenario 2 65
Figure 7.8 : Altitude profile of conflict resolution trajectory to the Scenario 2 65
Figure 7.9 : Conflict resolution trajectory for the multiple-thread Scenario 3...... 65
Figure 7.10: Trajectory cost convergence curve of CR algorithm for the Scenario 3 66
Figure 7.11: Speed profiles of conflict resolution trajectory to the third scenario. 66
Figure 7.12: Altitude profile to the third scenario 67
Figure 7.13: A screenshot from *4D Operational Display* (4DOD) illustrating initial (red), updated (green) and intruder aircraft's (dark blue) trajectories 68

FLIGHT DECK CENTERED COST EFFICIENT 4D TRAJECTORY PLANNING

SUMMARY

Considering the transformation in roles of existing air traffic management technologies, future flight operations and flight deck systems will need additional avionics and operational procedures that involve adaptive algorithms and advanced decision support tools.

The first part of the thesis presents novel visual flight deck decision support tools and interfaces utilizing next generation synthetic vision and augmented reality based visualisation technologies in order to meet the requirements of the future flight operations defined in NextGen and SESAR 2020+ visions. These avionics are envisioned to aid pilots for conducting their new in-flight tasks such as; collaborative tactical planning with intent negotiation/sharing; fully understanding/analysing/interpreting solution with their alternatives and proposing modification on the solution subject to negotiation; and aware of required response, execute it or allow collision avoidance module to perform its automated response. Visual Decision Support Tools allow the flight crew to interact with new autonomous systems and provide with visual understanding on the evolving flight operation by fusing all tactical level data and visualising them. In this work, two groups of display structure have been proposed. A split head-down *Synthetic Vision* screen pair aims to support the pilots in managing both low level and high level tactical tasks with fully understanding the situation in 4D. Synthetic Vision Display (SVD) side provides the pilots synthetic vision and also incorporates required additional guidance and limited operational information. 4D Operational Display (4DOD) provides higher level operational information giving building enhanced understanding on the states of the operation and results of any modification on processing flight intent. Haptic interfaces allow the flight crew to change demonstrated detail levels in both 2D+time and 3D+time. The other display, which is *Head-Up-Display (HUD)*, provides pilot to efficiently operate flight operation by eliminating the need of looking to head-down screen; and aims to present all essential flight information in the pilot's forward field through augmented reality implementations. For hardware integration and experimental purposes, an integrated testbed including full replica B737-800 Flight Deck Testbed and ATM Testbed has been modified as enabling operational tests and validations of these new tools.

The main purpose of this study is to provide a theoretical framework for tactical 4D-trajectory planning and conflict resolution of an aircraft equipped with novel automation tools. The proposed 4D-trajectory-planning method uses recent algorithmic advances in both probabilistic and deterministic methods to fully benefit from both approaches. We have constructed an aircraft performance model based

on BADA 4 with high-level hybrid flight template automatons and low-level flight maneuver automatons. This multi-modal flight trajectory approach is utilized to generate cost-efficient local trajectory segments instead of solving complex trajectory-generation problems globally. The proposed sampling-based trajectory planning algorithm spatially explores the airspace and provides proper separation through local trajectory segments and guarantees asymptotic optimality under certain conditions. Moreover, we have integrated the cross-entropy method, which transforms the sampling problem into a stochastic optimization problem, rapidly converges on the minimum cost trajectory sequence by utilizing available flight plans, and reduces the amount of sampling. The integration of the proposed strategies lets us solve challenging, real-time in-tactical 4D-trajectory planning problems within the current and the envisioned future realm of air traffic management systems.

KOKPİT OTOMASYONU TABANLI 4D ROTA PLANLAMASI

ÖZET

Hava trafik yönetimi teknolojilerindeki mevcut sistemlerin dönüşümü göz önüne alındığında, gelecek uçuş operasyonlarının ve kokpit içi sistemlerin yeni aviyonik sistemlere ve operasyonel prosedürlere ihtiyaç duyacağını söylemek mümkündür. Özellikle adaptif algoritmalar ve gelişmiş karar destek sistemleri bu ihtiyaçların temelini oluşturmaktadır. Bu konseptlerin hayata geçirilmesi Hava Trafik Yönetimi kapsamında görevlerin ve sorumlulukların değişmesinde büyük rol oynayacaktır. En iyi karar yeri, en iyi karar zamanı ve en iyi karar veren bu bağlamda temel faktörlerdir. Örneğin; kontrolcüler hava trafiğini yönetmede yüksek derecede rol sahibi olacak ve bireysel rotalara müdahale sayısını azaltacaklardır. Pilotlar uçuş esnasında daha aktif olacak; çevreyi gözlemleme ve yönetme, seçenekleri analiz ete veya gerektiği durumda ayırma manevrası uygulama gibi önemli görevlerde daha çok görev alacaktır. Uçuş ekibinin rolündeki bu değişimler mevcut görevlerin yeniden tanımlanmasına gidilmesinin yanı sıra insan faktörü performansını da etkileyecektir. Geleceğin kokpit içi sistemlerinde uçuş ekibinin bu yeni görevleri başarıyla gerçekleştirmesini sağlayan yeni nesil cihazlar ve algoritmalar gerekecektir.

Bu tez kapsamında yapılan ilk çalışma, yeni nesil sentetik vizyon ve artırılmış gerçeklik tabanlı görselleştirme teknolojileri kullanılarak görsel kokpit içi karar destek araçları ve arayüzleri tasarımıdır. Dizayn edilen bu araçların NextGen ve SESAR 2020+ programlarında tanımlanmış gelecek uçuş operasyonlarının gereksinimlerini karşılaması amaçlanmaktadır. Bu aviyonik sistemler ile pilotların niyet paylaşımı/pazarlığı ile işbirlikçi taktiksel planlama, çözümleri alternatifleri ile birlikte tam olarak anlama/analiz etme/yorumlama ve yeni çözüm önerme gibi uçuş operasyonlarında desteklenmesi vizyonlanmıştır. Ek olarak, gerekli cevabın farkında olma, uygulama veya çarpışma önleyici sisteme otomasyon yetkisi verme gibi görevlerde de karar destek sağlanması hedeflenmiştir. Görsel karar destek sistemleri uçuş ekibinin yeni otonom sistemler ile etkileşimini ve tüm taktiksel veriyi görselleştirerek içinde bulunulan durumu veya gelişmekte olan uçuş operasyonunu anlaşılır olmasını mümkün kılmaktadır. Bu proje kapsamında iki farklı görsel yapı sunulmaktadır. Kokpitin Primary Flight Display bölgesinde yer alan sentetik vizyon ekran çifti pilotların 4D ortamda durum farkındalığı ile düşük ve yüksek seviyede taktiksel görevleri yönetmesini sağlamaktadır. SVD kısmı pilota yapay görsellik sağlamakla beraber gerekli güdüm, uçuş ve kısıtlı seviyede operasyonel bilgileri içermektedir. Tunnel-in-the-sky konsepti ile pilot, odaklanılan veya karar verilen rotayı tüneller aracılığıyla manuel olarak takip edebilir. Bununla beraber standart sentetik vizyon (sanal gerçeklik) ekranı fonksiyonlarını da kullanabilir. İrtifa ve hız bilgileri, radar frekans değerleri, harita ve yükselti bilgisi, hava koşulları gibi temel uçuş operasyonu verisi bu ekranda gösterilmektedir. 4D Operasyonel

Ekranı (4DOD) operasyon durumu ile ilgili farkındalığı artırmak ve uçuş niyeti üzerindeki modifikasyonları göstermek üzere yüksek seviyede operasyonel bilgileri sağlamaktadır. Pilot, hem kendi yörüngesini kontrol edebilmekte hem de trafikteki uçaklara ait rotaları izleyebilmektedir. Aynı zamanda ileriye dönük hızlandırılmış simulasyon fonksiyonu da bulunmaktadır. Kokpitin veri bağlantısı üzerinden yer ile rota ve uçuş planı paylaşımı sürecinin yönetilmesi bu ekran aracılığıyla olmaktadır. Haptik arayüzler ile uçuş ekibi gösterilen bilgileri ve görselleri 2D+zaman ve 3D+zaman boyutunda yönetebilmektedir. Sentetik vizyon ve 4DOD ekran çiftine paralel olarak pilotun görüş hizası üzerine inşa edilmiş Head-up Display (HUD)bulunmaktadır. HUD aracılığıyla pilot benzer şekilde temel uçuş durum bilgilerini aşağıya bakma gereği duymadan izleyebilmekte, tunnel-in-the-sky konsepti sayesinde hedef yörüngeyi tüneller arasından uçmaya çalışarak takip edebilmektedir. Bu görsel karar destek sistemleri ve algoritmalarının donanım olarak entegrasyonu, Boeing 737-800 uçuş simulatörü üzerinde gerçekleşmiştir. Sentetik vizyon ve 4DOD ekran çifti Primary Flight Display (PFD) monitörleri üzerinde çizdirilmiştir. Head-up Display (HUD), kaptan pilot ile ön cam arasına yerleştirilmiştir. Özel bir film kullanılarak görüntü arkadan mini-projeksiyon cihazı aracılığıyla yansıtılmıştır. Her bir görsel karar destek sistemi simulatörün ağına bağlanmış olup veri akışını kontrol eden ve yöneten algoritmalar düzenlenmiştir. Uçuş simulatörü, Hava Trafik Kontrolü test ortamı ile birleştirilerek geliştirilen yeni nesil aviyonik konseptlerinin uçuş operasyonları üzerindeki etkileri resmedilmiştir. Hava Trafik Kontrolü test ortamı trafik ve hava durumu tasarlayıcı, Hava Trafik Kontrol ekranları ve kontrolörün davranışının benzetim çalışmalarını yapan modellerden oluşmaktadır. Test ortamı aynı zamanda ALLFT+ tabanlı geçmiş uçuşlara ait gerçek veri kullanarak önceden belirlenmiş veya düzenlenebilen senaryoların oynatılmasını sağlamaktadır. Trafik ve hava durumu tasarlayıcı modül Demand Data Repository veritabanı üzerinden beslenen havaalanı ve hava sahası kapasite bilgilerini ve Aeronautical Information Publication'dan gelen operasyonel bilgileri içermektedir. Benzer şekilde, modifiye edilmiş senaryolar veya geçmiş hava durumu bilgileri METAR verisi üzerinden aktarılmaktadır. Test ortamı günümüz hava trafik kontrol ekranları, ses ile iletişim, otonom veya karar destekli kontrol operasyonlarını ifade eden modeller aracılığı ile hem günümüz operasyonlara hem de geleceğe yönelik çalışmalara ait senaryoları koşabilmektedir.

Projenin ikinci aşaması ve ana amacı ise taktiksel 4D yörünge planlaması ve otomasyon araçları ile donatılmış uçak için "conflict resolution", ya da potansiyel çarpışma önleyici ve bunu otonom olarak yapan sistemler için teorik çerçeve tasarlanmasıdır. Yoğun trafik ortamında veya yeni rota hesaplanması gibi durumlarda yerden bağımsız, uçak üzerinde ve otonom olarak hem gerçeklenebilir, hem de maliyeti düşük rotaların üretilmesi istenmektedir. Önerilen 4D yörünge planlama metodu hem olasılıksal hem de deterministik algoritmaların yeni özelliklerini içermekle beraber iki yöntemin de başarılı taraflarını birleştirmektedir. Uçak performans modeli ise yörünge tayini için gerekli bir bileşen olup BADA 4 üzerinden sağlanmaktadır. Uçağın kinodinamik modellemesinde standart yörünge uygulamalarında kullanılan 3-serbestlik dereceli veya diğer adıyla nokta kütle hareket modeli kullanılmıştır. Bu modelde uçağın hali hazırda kendi içerisinde kararlı ve kontrol edilebilir olduğu kabul edilip, takip ettiği yörünge ile ilgilenilmektedir. Uçağa etkiyen kuvvetlerin veya uçak performansının modellenmesi EUROCONTROL'ün bir ürünü olan Base of Aircraft Data (BADA) aracılığıyla yapılmıştır. Projede son sürüm olan BADA 4 kullanılmıştır. Bu versiyon, öncekilerden farklı olarak uçağa etkiyen kuvvetleri uçağın durumları ve atmosfer koşullarına bağlı olarak parametrik ifade etmektedir. Teknik altyapısını Boeing'in sağladığı bu veritabanı, gelişmiş modellemeleri sayesinde nominal değerlerin üzerine çıkarak parametre öngörmesi ve optimizasyon gibi işlemleri yapılabilir kılmaktadır. Oluşturulan bu performans modeli yüksek-seviye hibrid uçuş kalıpları otomatları ve alçak-seviye manevra otomatlarını kapsamaktadır. Bu modellemedeki amaç, uçak hareketini tırmanma, seyir ve alçalma şeklinde üç farklı kalıp altında toplamaktır. Her bir uçuş kalıbı kendine özel manevra sekansı içermektedir. BADA 4 matematiksel modelleri aracılığıyla her bir uçuş kalıbı için tanımlı manevra sekansını düşük maliyet ile gerçekleştiren parametreler öngörülmüştür. Esasında bu problem, bir uçağın başlangıç ve bitiş olarak verilen iki nokta arasında en düşük maliyetli rotayı takip etmesi problemidir. Uçak denklemlerinin ve kısıtlamaların lineer olarak ifade edilememesi, bu problemin tek bir seferde global olarak çözülmesini zorlaştırmaktadır. Ek olarak bu modülün uçak üzerinde çalışacağı düşünülecek olursa bu hesaplamaların çok kısa zaman aralıklarında gerçekleşmesi beklenmektedir. Çok-modlu yaklaşım sayesinde kompleks olan yörünge planlama problemini global olarak çözmek yerine lokal ve düşük maliyetli yörüngeler tayin edilmektedir. Bu noktadaki dezavantaj ise yaklaşımın verdiği çözümün optimum değerden uzaklaşmasıdır.

Daha üst seviyede ise hesaplanan düşük maliyetli lokal rota parçaları oluşturan ve uzayı tarayan RRT* algoritması kullanılmıştır. RRT*,örnekleme tabanlı bir hareket planlama algoritması olup hava sahasını keşfetmeye çalışarak lokal yörünge segmentleri üzerinden ayırma yapmaktadır. İlk adım olarak uzayda bir konum örnekleyip, ardından uçuş kalıpları ve gelişmiş performans modelini kullanarak uçağı bu noktaya düşük maliyet ile getirmeye çalışmaktadır. Örneklenen konuma, mesafe olarak ağaçta hali hazırda bulunan en yakın konumdan bağlanmaya çalışılır. Bu, arama uzayının hızlı ve ilerleyerek keşfedilmesinin temelidir. Lokal maliyetlerin yanında başlangıç konumundan itibaren harcanan maliyet de hesaba katıldığı için ağaç sürekli olarak toplam maliyeti düşük olan uçuş segmenti sekanslarını üreterek büyür. Önceden belirlenmiş örnekleme sayısına ulaşıldığında algoritma durur. Kullanılan algoritma aynı zamanda belirli koşullar altında asimptotik optimalliği sağlamaktadır. Asimptotik optimallik, örnekleme sayısı sonsuza yaklaştıkça problemin optimal çözüme yakınsama özelliğidir. RRT* aynı zamanda olasılıksal bütünlüğü sağlamaktadır: Örnekleme sayısı sonsuza yaklaştıkça çözüm bulma olasılığı 1'e yakınsamaktadır. Bunlara ek olarak, örnekleme için cross-entropy yöntemi kullanılmıştır. Bu yöntem ile örnekleme problemi stokastik optimizasyon problemine dönüştürülerek hızlı bir şekilde minimum maliyetli yörünge sekansı oluşturulmuştur. Akıllı örnekleme yapılırken halihazırdaki uçuş planları kullanılmış, dolayısıyla örnekleme sayısının düşük tutulabilmesi sağlanmıştır. Standart rastgele örnekler almak yerine daha akıllı örnekleme yapmak, optimum sonuca daha çabuk ulaşılmasını sağlamıştır. Ancak, her adımda oluşturulan küme içinden ağırlıklandırması yüksek olan elit set çekildiği için hesaplama yükü artmıştır.

Proje kapsamında hem Avrupa'nın hem Amerika'nın hava trafik yönetimi konusunda yaptığı kapsamlı araştırmalar incelenmiş ve buradaki trendler takip edilmiştir. Hava trafik yönetiminde kapasiteyi artırmak üzere yer kontrolcülerinin görevlerini daha çok genel akışı yönetmesi vizyonlanmış; pilotların ise daha çok aktif rol aldığı bir dünya çizilmiştir. Pilotlara karar vermelerinde destek olacak görsel sistemler tasarlanmış, yer ile uçağın aynı anda işbirlikçi bir biçimde uçuş operasyonunu yönettiği konseptler eklenmiştir. Bunların yanında çarpışmaları gözleyen ve gerektiği durumda otonom ayırma yapabilen sistemler için algoritma tasarlanmıştır. Geleceğin hava trafik koşullaru vizyonlanarak göz önünde bulundurulmuş, önerilen yöntemin hem bugünün hem de geleceğin hava trafik yönetim sistemine katkı sağlaması amaçlanmıştır.

1. INTRODUCTION

A central theme in both NextGen and SESAR visions is paradigm shift from a purely centralized tactical intervention model toward more efficient strategic planning and more proactive tactical operations [1, 2]. The implementation of these concepts will significantly change the roles and responsibilities in the Air Traffic Management (ATM) system. For instance, air traffic controllers will have a high-level tactical role in managing traffic flow and no longer intervene in individual trajectories. Thus, pilots supported with automation systems will become more active during the flight in order to monitor the environment, generate a separation maneuver, if needed, and check alternate plans. This transformation will not only redefine the existing roles of the flight crew but also create further responsibilities that inherently affect human performance requirements. Therefore, the future flight deck will require additional avionics, operational procedures with adaptive algorithms, and automation systems with advanced decision-support tools that can enable pilots to handle the entire tactical operation. To meet the requirements of future flight operations, we envision integrating novel automation modules into the current structure of flight deck systems andbuild a B737-800 flight deck testbed to integrate these concepts and test the developed algorithms (Figure 1.1).

Tactical 4D-trajectory planning is the process that provides persistent conflict check and proper resolution when required. The conflict detection (CD) process guarantees appropriate separation between aircraft during flight. CD algorithms compare the spatial distance between any two aircraft with the mandated separation minimum. In current operational practice, aircraft are kept 3–5 nmi apart horizontally or 1000 ft vertically to provide a sufficient safety margin. The conflict resolution (CR) process generates an appropriate action that suitably solves potential conflicts detected by the CD. By considering the time horizon, tactical conflict detection and resolution typically involves challenging issues such as estimating aircrafts' future positions, predicting potential conflict, and issuing the proper conflict alert. The main difficulty in predicting

Figure 1.1 : B737 – 800 Flight Deck test platform with experimental visual decision support tools for future ATM realm: Head up Display (HUD), Synthetic Vision Display (SVD) and 4D Operational Display

the aircraft future position comes from disturbances influencing the flight path such as wind or uncertain actions of other aircraft.

As the realm of ATM evolves, the way of managing flight operations and handling the tactical requirements will change. Fully-tactical operation capability will enable the airline to dynamically redefine preferred needs according to evolving conditions. Currently, neither ground-based nor on-board systems account for the own aircraft's intended flight plan (e.g. providing recovery to the original plan) or the preferences of the flight operator. In addition to safety, cost effectiveness in tactical planning will also be a sensitive issue in the future. For example, dynamic cost index management [3] might enable airlines to dynamically regulate how phases of the flight are directed (e.g. fly faster or save fuel). This approach allows operators to redirect the aircraft according to needs of passengers or their financial strategies. Moreover, [4] shows that small modifications to the cruise phase such as cruise altitude or speed can achieve significant cost reductions in cruise fuel burns.

1.1 Description of the Work and Motivation

In this work, our objective is to provide a theoretically sound and practically efficient framework for solving tactical 4D-trajectory problems. The proposed method involves a sophisticated aircraft performance model based on BADA 4 and recent algorithmic advances in stochastic approaches to motion planning. Such probabilistic algorithms embed stochastic behavior, which are inherent in air traffic. The proposed method also utilizes operational cost objectives in the calculation of cost-efficient trajectory segments through predefined flight-template automatons.

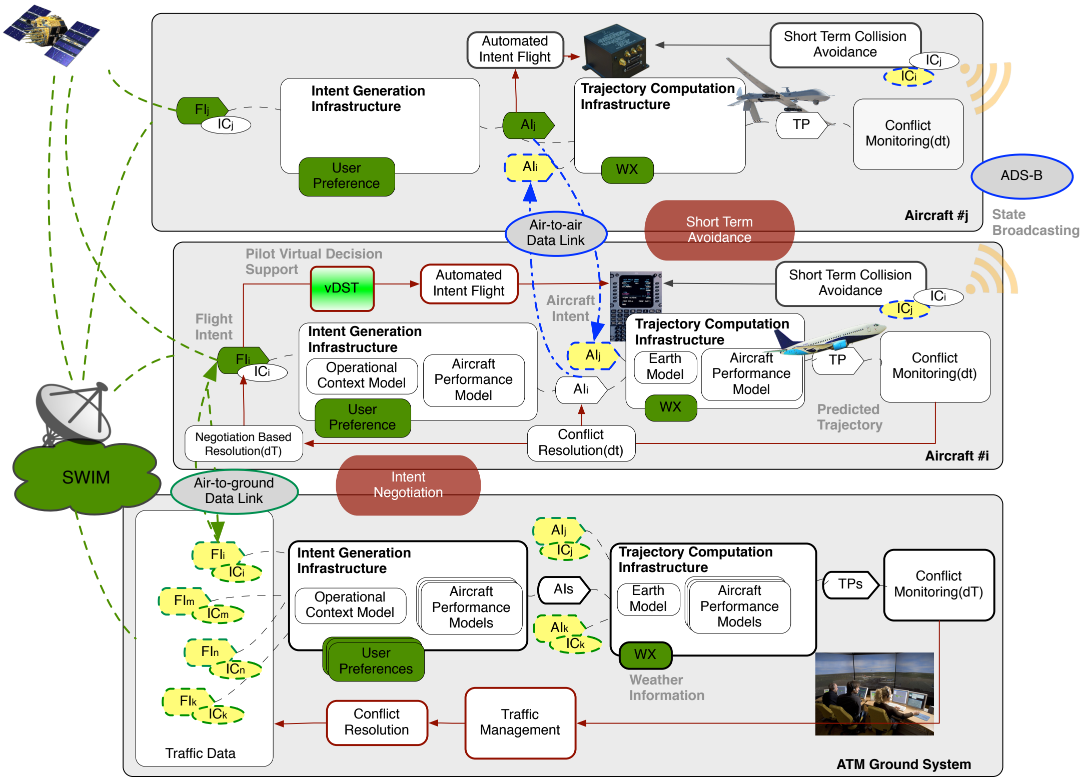


Figure 1.2 : The envisioned data exchange and trajectory occurrence procedures for the future airspace needs

These flight templates employ their own approximate trajectory optimization and involve lower-level maneuver mode automatons that effectively utilize advanced performance definitions in BADA 4. Specifically, the maneuver mode automatons provide low-level control input sequences, which are compatible with the current flight management systems (FMS). The formal description of a control sequence is a potential candidate to be a communication frame of the controller-pilot data link (CPDL) or to be transformed into any other data link standard.

The sampling-based trajectory planner algorithm presented here spatially explores the airspace and provides proper separation with local trajectory segments. The algorithm also guarantees asymptotic optimality under certain conditions. Moreover, we have integrated the cross-entropy method, which transforms sampling problems into stochastic optimization problems, which enables more efficient sampling. The initialization of the problem exploits the last-available flight plan that was compromised due to uncertain conditions such as wind speed change. The idea behind importance sampling with cross entropy is that the new plan is most likely to be spatially similar to the original flight plan. This practice is also inherent to ATM, where the strategic flight plan (i.e. Reference Business Trajectory) already reflects many objectives of the stakeholders (e.g. airlines, air traffic flow managers) subject to comprehensive optimization that is run in ground systems. In the hypothetical worst-case scenario, where the new flight plan is far from the previous optimum, the importance sampling iteratively converges on a low-discrepancy sampling, which is purely quasi-random sampling. Otherwise, and mostly, the cross-entropy sampling rapidly converges on a delta function, in the other words, the minimum cost trajectory. The integration of the proposed strategies let us resolve conflicts in challenging, real-time in-tactical situations.

The proposed algorithms needs advanced integration into the flight deck structure and additional human interaction channels, so we describe the whole picture in order to provide a clear understanding of the problem within the scope of ATM systems. In order to meet the requirements of future flight operations, we envision integrating novel automation modules into the current structure of flight deck systems and build a B737-800 flight deck testbed to integrate these concepts and test the new algorithms (Figure 1.1). This integrated structure uses two-level autonomy in a different kind of time horizons such as *Collaborative 4D Trajectory Planning* and *Short Term Collision Avoidance*, both of which involve distinctive tools, procedures, data handling, and algorithms. The *Visual Decision Support Systems*, integrated with these modules, allow the flight crew to monitor the processes and interact with them at a manageable level. Figure 1.2 demonstrates the entire structure and its add-on modules.

During nominal tactical operation, or *Collaborative 4D Trajectory Planning*, processes are mostly collaborative, where the pilot cooperates with the ground systems and uses in-flight decision support and automated tools, as seen in Figure 1.2. Such structure incorporates all tactical level information (e.g. weather, intent exchange, user preferences, and traffic information) obtained from both air-to-air data links and air-to-ground data exchange. Ground-based intent negotiation requests may arise in with changes in drastic weather, operational constraints, conflict detection, emergency situations, or when a detected aircraft does not follow the anticipated behavior. *Intent Generation Infrastructure* translates a flight intent (FI), which is typically a formal description of flight objectives attached to a strategic flight plan, into an aircraft intent (AI), which provides a detailed formal description of the navigational commands of the aircraft [5]. *Trajectory Computation Infrastructure* generates a unique predicted trajectory (TP) that relies on an aircraft performance model. This structure permits low-level aircraft intent (AI) sharing between aircraft through air-to-air data links. Low-level intent sharing also enables "machine-to-machine" talk where the pilot can communicate with unmanned systems, as we anticipate the integration the UAVs into national airspace. The Short Term Collision Avoidance module is not connected to the AI exchange and works independently. Thus, it provides redundancy in flight deck systems (e.g. TCAS).

In both nominal flight operations and active collaborative decision-making processes, it is essential to keep the pilot in the loop at a manageable level. Moreover, pilots must recover flight control from an automation failure. The new virtual *Decision Support Tools* (vDST) with a head-down synthetic vision display (SVD) and augmented reality-based head-up display (HUD) give the pilot a full understanding of the evolving situation. In addition to these common display concepts, another synthetic vision display concept, the 4D-Operational Display (4DOD) (Figure 1.1),has been developed to manage tactical 4D-trajectory-based operations. This virtual decision support tool provides pilots with a 4D projection (three spatial dimensions and time) of the trajectories (including predicted trajectories of the surrounding aircraft) and allows pilots modify the trajectory or requesting a re-plan. *Conflict Monitoring* and *Conflict Resolution* functions perform the CDR algorithms that will be presented in this study, which continuously ensure proper aircraft separation and generate required actions in case of a failure to separate.

1.2 Literature Review

A comprehensive literature review of conflict detection and resolution is given in [6]. Many real-time conflict-detection and resolution systems use some form of open-loop planning algorithms. Open-loop planners compute a "one-shot" trajectory projection and plan without considering how future information will alter future actions. These online algorithms first generate a sequence of actions to take from the current state, and the plan is updated whenever a new observation arrives. The algorithm in [7], NextCAS II, provided a model-based solution that computes alert thresholds that do not violate an intruder's protection zone. In [8],a model based on Mixed Integer Linear Programming (MILP), involving approximate model (point mass model) of aircraft dynamics with linear constraints, is applied to open-loop aircraft collision avoidance problems. [7] utilized the MILP method for solving conflicts arising among several aircraft, but considered only velocity-changing actions. [9] suggested a multi-layered open-loop "almost blind engagement" process where the planner tries to solve ownship's trajectory according to belief states of the intruder aircraft and updates the projected belief whenever a new measurement arrives. Unlike open-loop methods that generate static plans as new information becomes available updated, closed-loops methods [10, 11] generate an action sequence that minimizes action cost by accounting for future actions and updates alert likelihoods with new information. Online Markov Decision Process (MDP) algorithms [12, 13] address the shortcomings of offline methods by only planning for the current belief state instead of planning for all possible situations. A hybrid solution has been proposed [14] where the calculation for the expected utility of being in a particular belief state and required action are selected online; action utilities are computed offline. [15] provides a conflict-resolution algorithm for solving a parametric-optimization problem of the point-mass model and utilizes formal definitions for the predefined trajectory parameterization of the aircraft intent.

In real time applications, such as tactical conflict resolution, the principal concern is to find a feasible solution as fast as possible and to enhance the "quality" of solution in the remaining time. Sampling-based algorithms have received considerable attention in the trajectory-planning literature. As such, there has been increasing recent interest (e.g. [16–19]) to demonstrably improve the quality of a sampling algorithm's solution as computation time increases. Sequential sampling-based algorithms do not stop sampling once a feasible trajectory is found in order to find a better solution. Sampling-based methods in trajectory planning randomly sample a set of states from the state-space and check their connectivity without fully knowing the obstacles. This approach provides significant savings in computation time since collision checks are performed when required. The connectivity of these samples is strongly connected to feasibility and reachability notions in planning problems. Even though sampling-based methods do not provide completeness, they are probabilistically complete where the probability of finding a feasible solution, if one exists, approaches one as the number of samples increases. One such kinodynamic sampling-based planner is the

Rapidly-Exploring Random Tree (RRT) algorithm, first proposed in [20]. Recently, the RRT algorithm and its variants have been successfully demonstrated on different dynamic systems [21–25]. An important step toward efficient optimization using randomized planners was taken in [19], which proves that the RRT algorithm converges on a non-optimal solution with a probability of one. Furthermore, this research proposed a new algorithm, RRT^*, and showed that it is globally and asymptotically optimal while maintaining the same probabilistic completeness and computational efficiency of the original RRT. RRT^* is a superior algorithm in comparison to other sampling-based methods.

A common concern in randomized algorithms is their lack of repeatability, which makes it impossible to certify their success and performance. Two successive runs of these algorithms may not produce identical solutions even under identical initial conditions, while a deterministic algorithm always has the same result. It is not possible to give a clear proof that any randomized algorithm solves a motion planning problem very quickly. Besides, [17] clearly showed that deterministic sampling strategies outperform purely random sampling in solving many-dimensional problems. To address this issue, a meta-heuristic can monitor the growth of the number of samples and resets the search graph if its size exceeds a certain threshold. This is necessary because processing complexity increases dramatically (e.g. finding the nearest node) as the size of the search graph increases. The performance of sampling-based algorithms can be further improved by employing more efficient adaptive sampling. Several methods have been proposed [26–31] that utilizing information from previous sampling loops. [32] transformed the sampling problem into a stochastic-optimization problem, in which cross-entropy (CE) was used to estimate the parameter set of a distribution, which guides the algorithm to sample around the optimal trajectory. The CE method was originally introduced by [33] for estimating rare-event probabilities, and since then, the method and its adaptations have become useful tools for multi-extremal nonlinear optimization. Specifically, [32] has integrated the CE into the RRT^* to iteratively update the distribution in accordance with costs until the distribution closes around the optimum trajectory. In this project, we closely followed [32], and propose a more generalized form of CE sampling for conflict resolution.

Motion-planning problems for aerospace vehicles are complicated by the fact that planners based on optimal performance begin to fail, by means of computation, when one takes into account the constraints of the aircraft's dynamic equations. To reduce the complexity of this problem, motion-description languages and quantized control concepts were proposed in [34].Multi-modal maneuver-modeling framework basically consists of decomposing a trajectory into a set of maneuver modes and associating maneuver parameters. The complexity of maneuver planning is significantly reduced by reducing the dimension of the problem (modal sequences have lower dimensions than state-space descriptions) and relaxing parameter optimization by designing specific optimization procedures for each mode. The planner constructs a trajectory with proper modal sequence by switching instead of performing a highly complex multi-objective parametric optimization over the full flight envelope. This approach has been successfully applied in [35] for autonomous combat aerial vehicles, which involves complex modeling and control. In the approach presented in [35], parameterized sub-maneuvers are build up into complex sequences and make it possible to address almost any arbitrary maneuver and the entire flight envelope.

In parallel with this research, closed-loop hybrid control systems were developed for the same purpose using linear temporal logic [36]. For aerospace vehicles, a hybrid model for aircraft traffic management calculating the largest controlled invariant subset of each aircraft's protected zone has been developed [37], where relatively simplistic horizontal maneuver modes are used for an algorithmic demonstration of a hybrid approach. Similarly, [38] suggested maneuver automaton, which uses a number of feasible system trajectories to represent the building blocks of the aircraft's motion plan and a trajectory-based control system, which asymptotically regulates the actual trajectory to the trajectory generated by maneuver automaton. However, motion plans and controllable trajectories are restricted to the library of the maneuver automaton. Such libraries are built by interpolating between feasible trajectories [39]. [40] extended this for online planning of feasible trajectories in partially unknown environments by using receding horizon iterations.

Maneuver Modes and the Modal Inputs Configuration of a maneuvering aircraft can be explicitly described in terms of a single state trajectory. However, it is also possible to construct the maneuver by representing it as a sequence of predefined maneuver modes and associated parameters. In [41], a parameterized maneuver library is built where each maneuver mode is represented by a set of state constraints and state equations that evolve according to the modal inputs.

The rest of the thesis is structured as follows. Initially, the integrated flight deck testbed with novel virtual decision support tools is explained. Then we give a formal definition of the trajectory planning problem. *Aircraft Performance Model* section gives the details of the performance model based on BADA 4 and the cost definition. *Cost Efficient Local Trajectory Generation* section provides a method for generating local trajectory segments through the modal decomposition approach. The flight template automaton and its modal maneuver library are explained in this section. *Tactical 4D Trajectory Generation* section explains persistent conflict detection and resolution (CDR) based on a stochastic planning algorithm that uses *Local Trajectory Generation* and the importance-sampling method. The *Integrated Simulations* section gives some examples from the experimental results. *Conclusion* summarizes the method and presents potential future research.

2. FLIGHT DECK TESTBED

In this project, the author presents novel visual decision support tools and interfaces incorporating next generation synthetic vision and augmented reality based visualisation in order to support the flight crew. The presented head-down *Synthetic Vision* screen pair enables pilots to manage both advanced low level and high level tactical tasks with fully understanding the situation in 4D. Synthetic Vision Display (SVD) side provides the pilots synthetic vision and also incorporates required additional guidance and limited operational information. 4D Operational Display (4DOD) side aims to present higher level operational information allows understanding the states of the operation and results of any modification on processing flight intent. The interface allows pilots to change demonstrated detail levels in both 2D+time and 3D+time. The other display, which is *Head-Up-Display (HUD)*, provides pilot to efficiently operate flight operation by eliminating the need of continually transition from head-down to head-up; and aims to present all essential flight information in the pilot's forward field through augmented reality implementations. Even in low-visibility operations (e.g. due to fog, clouds, unlighted landing etc.), pilots can easily manage the flight by ensuring following the "visual tunnels" appear in the head up display. These visual decision support tools are envisioned to significantly increase situational awareness (SA) of the pilots during the flight operations.

Situation awareness (SA) refers to the operator's understanding of the relevant environment state and the operator's ability to anticipate future changes and developments in that environment. Specifically, there are three levels of situational awareness constructed by humans. These levels are perception, comprehension and projection [42]. Progression of these layers, the level of Automation and the extend of SA does not indeed exhibit a simple 1-1 relation. For example, inappropriate levels of the automation can impact SA with results such as automation complacency, automation mistrust, increased workload, and automation transparency. For example, high levels of automation can indeed create cases in which the pilot no longer actively processes information to maintain an awareness of the system state In other words; pilot falls out-of-the-loop due to over-trust in the system. Such fall-outs effectively diminish the pilot's ability to recover from automation failure . When the pilot perceives the automation to be unreliable and gives excessive attention to monitor the automation, SA can also be diminished with high workload and result in a phenomenon called attention tunneling [43]. In attention tunneling, all attention is drawn only to the primary task at hand. SA is also reduced while interacting with a decision support system which requires extensive evaluation of alternatives and choices [44]. The additional workload associated with extensive evaluation and selection naturally reduces the resources available for maintaining SA. A system is transparent when the underlying information behind the automation can be accessible . In a fully transparent system a pilot may be led to attend to too much and too low level system information, resulting in high workload and diminished SA [45].

By considering these factors, an expectation from a good decision support system is that it should provide transparency at a manageable workload level. In general, any form of automation support that unintentionally hides information seems to be in conflict with the responsibilities of the pilot (even if it might result in low workload and good performance). A cooperative process, in which the automation enables the pilot to be in-the-loop, is considered to be the optimal outcome of the design [45]. The conflict resolution experiments conducted in support this proposition. For example, in the SA test scenarios it is observed that the response times of the operators to immediate questions about past, present, and future events were faster if the operator is in interactive and manual conditions. This is in comparison to response times when

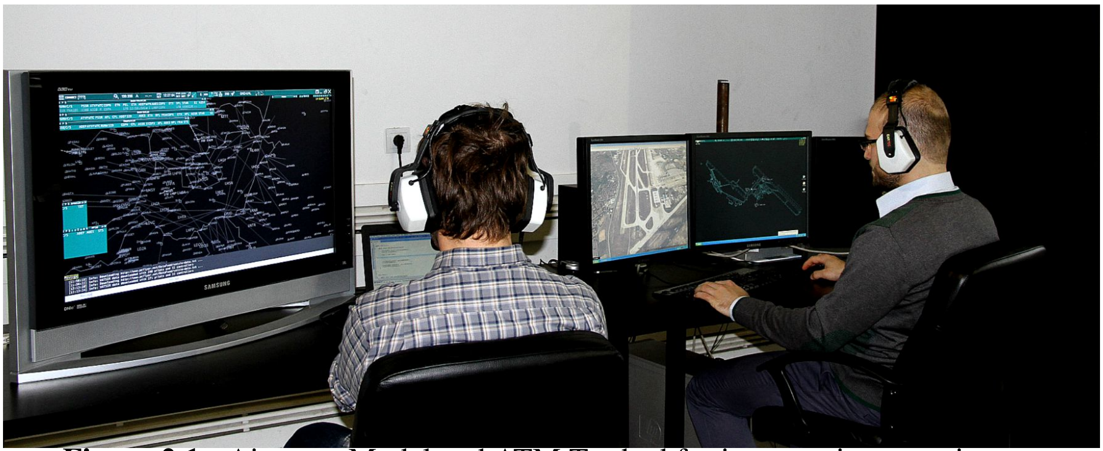


Figure 2.1 : Airspace Model and ATM Testbed for in-operation experiments

the operator is in fully automated condition (complacency). Relatively better SA in the interactive and manual conditions implies that conflict resolution systems may profit from keeping the pilot actively engaged in the task. However, evaluating conflict free flight plan with their alternatives, in both space and time, within various constraints, is a complex task especially in immediately emerging traffic situations (short term and mid term). The crew cannot be expected to perform such a complex task without some form of automated observation-evaluation-strategy generation support. Therefore, the pilot is located in-the-loop, but at a higher strategic level where he or she is constantly aided with safe flight plans and alternatives.

2.1 Integrated Testbed: Flight Deck Simulator and ATM Testbed

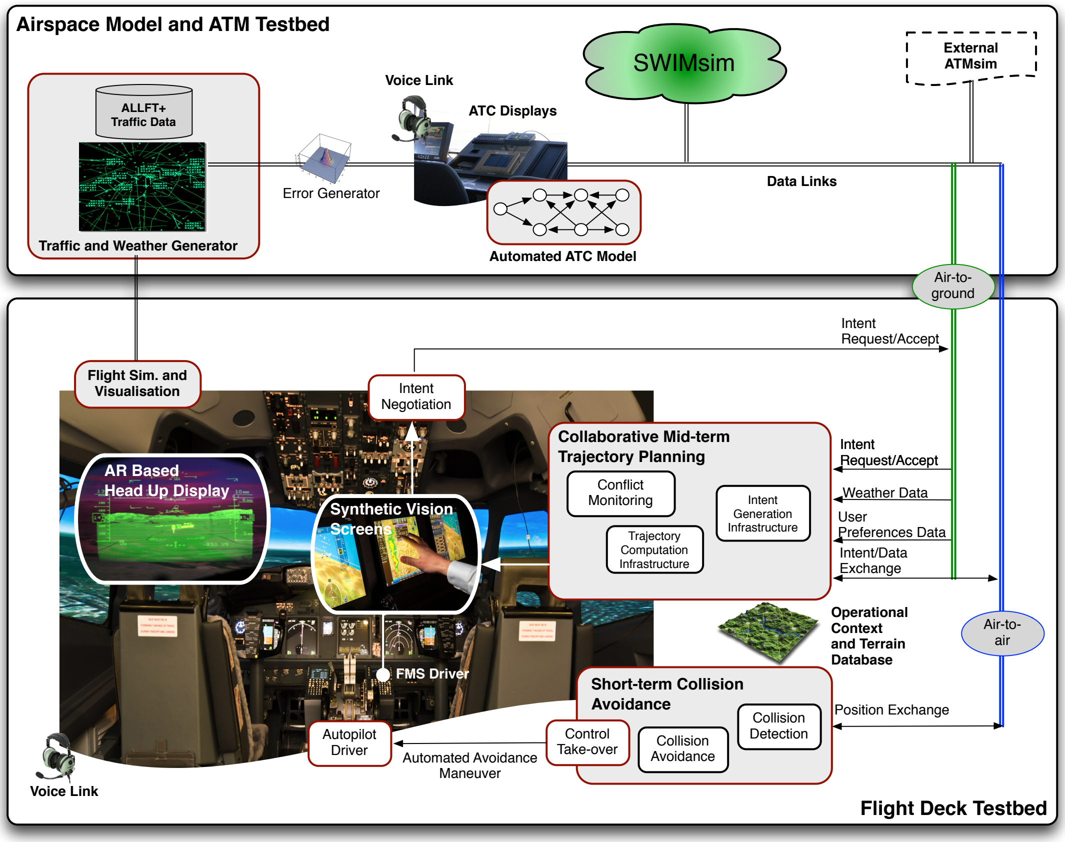


Figure 2.2 : Architecture of the integrated next generation flight deck system with novel add-on modules

The integrated system including a B737-800 flight deck testbed and an ATM testbed is envisioned to validate innovative add-on avionics and features come into the flight deck automation systems in order to meet the requirements of the future flight operations. The given flight deck structure uses two different autonomy levels and handles switching these autonomy level modes considering the required response time. These two process cycles at different autonomy levels are represented with *Collaborative Mid-term Trajectory Planning* and *Short Term Collision Avoidance* modules where both are involving different tools, procedures and algorithms. The visual *Decision Support Systems*, allow the flight crew to efficiently monitor the processes, and interact with them at a manageable level. Through these objectives, two groups of displays, head-down Synthetic Vision Displays (including separated Synthetic Vision Screen and 4D Operational Screen) and Head-up-Diplay (HUD) are integrated into the flight deck to support the pilots and significantly enhance their situational awareness of the pilot. Figure 2.2 demonstrates whole integrated structure and its add-on modules.

Airspace Model and ATM Testbed (seen in Figure 2.1) involves air traffic management related simulation tools such as: Traffic and Weather Generator, ATC displays and Automated ATC Models. The testbed allows to simulate ALLFT+ based historical traffic data set or any custom scenario in the same form. The *Traffic and Weather Generator* incorporates airport and airspace capacity information from the historical Demand Data Repository (DDR) data set and operational context information comes from the Aeronautical Information Publication (AIP) in order to create complete airspace picture. Similarly, customised scenarios or historical weather effects can be regenerated with the simplified version of the METAR data. The testbed allows to perform both traditional air traffic control operations via ATC displays and voice communication, and fully automated or aided traffic control operations through the hybrid *Automated ATC Models* which is ongoing research. The software structure of the entire with their physical links are also given for further understanding in Figure 2.3

In the nominal tactical flight operations, it is expected that the pilot cooperates with the ground systems through a data link, and uses decision support and automated tools. In this operation mode, the envisioned system decision support tools incorporates all tactical level information (i.e. weather data, intent data, user preferences data and traffic data) obtained from both on-board sensing (including air-to-air data link) and air-to-ground data exchange. The pilots can also manage *Intent Negotiation* process via visual *Decision Support Tools* initiated by either the flight deck or the ground system. The ground based intent negotiation request may emerge in some circumstances

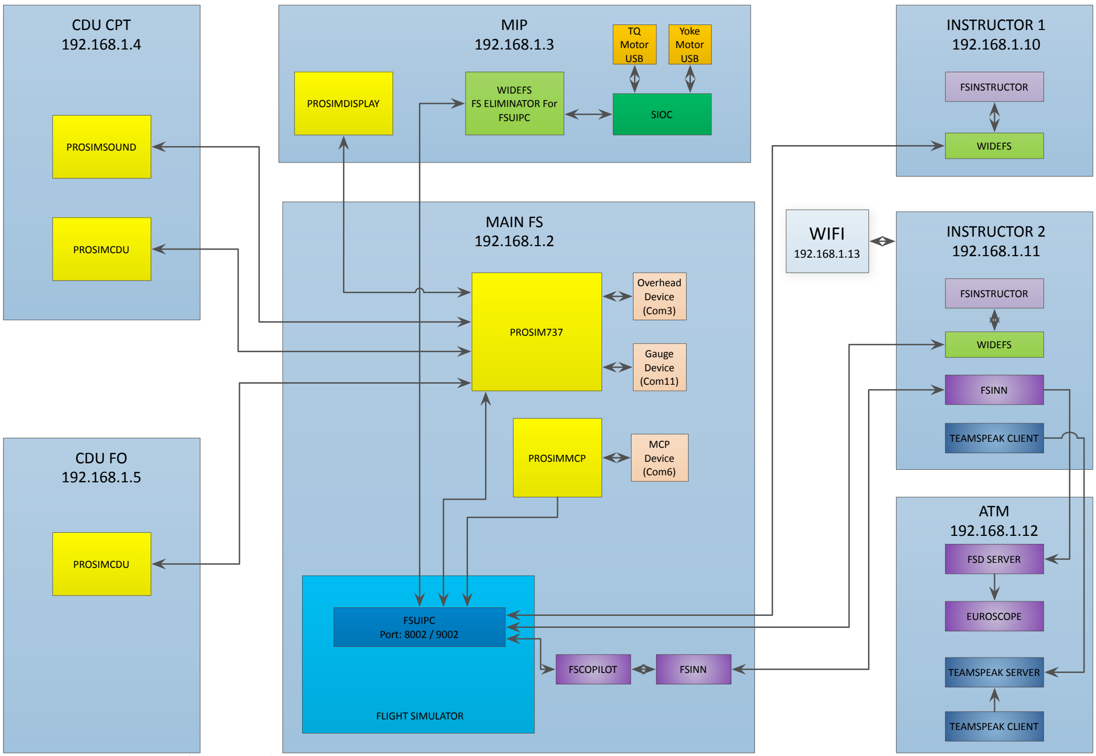


Figure 2.3 : Software Architecture of the integrated System

such as drastic weather change, change in operational constraints, conflict detection, emergency situations or detection of an aircraft does not conform to the anticipated behaviour. During the intent negotiation, pilot can monitor the requested trajectory, modify the solution, or request re-planning through the 4D Operational Screen. Similarly, the flight deck may also create an intent negotiation cycle and pilot can request an acceptance on the modified intent sequence (e.g. direct route to a fix or efficient flight path around hazardous weather). *Trajectory Computation Infrastructure* (TCI) and *Intent Generation Infrastructure* (IGI), automatically validates the feasibility of the given intent data, and *Conflict Monitoring* block checks potential conflicts between the predicted trajectories in the traffic.

2.2 Next Generation Synthetic Vision Screens

Presented synthetic vision display includes two separate screen (Figure 2.4); which one for synthetic vision flight and the other for operational management. These screens are envisioned to provide the pilot with full understanding on the evolving flight operation and effects of any intervention. Even in automated nominal flight operations, it is important to keep pilot in-the-loop at a suitable level where the flight crew should recover the flight control from an automation failure. Therefore, on the track of

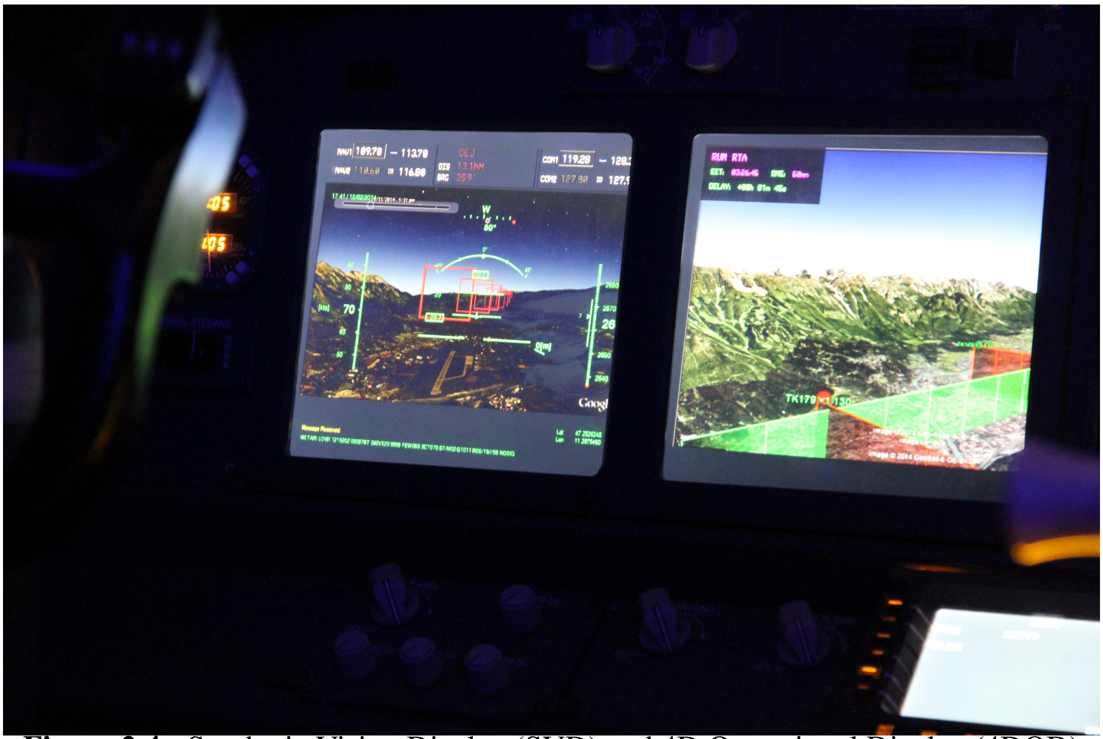


Figure 2.4 : Synthetic Vision Display (SVD) and 4D Operational Display (4DOD) screens in the flight deck

the negotiated trajectory, the flight crew is continuously supported with information about the current state and objectives of the operation (e.g. intent trajectory, RTA objectives, delays, ascending/descending slope and glide slope) and the environment (e.g. surrounding traffic, potential loss of separation, proximity to the terrain). During the intent negotiation process, one synthetic vision screen demonstrates processing flight intent to the pilot and enables required interaction to accept, modify or request re-planning – which are the functions of the collaborative decision making. Through the 4D Operational Display (4DOD), the flight crew can understand the states of the operation and results of any modification on processing flight intent. Whenever the negotiation has been concluded with a success, the negotiated intent can be executed autonomously via FMS (as seen in the Figure 2.2), or pilot can choose to follow the trajectory manually with guidance of the tunnel-in-the-sky visualisation on Synthetic Vision Display (SVD) and HUD.

The *4D Operational Operational Display* (4DOD) provides the pilots with high-level information about the whole flight operation and trajectory. Through the 4DOD, the pilot can monitor the flight trajectories (negotiated or processing) of the ownship and surrounding aircraft in four dimensions (including time); environmental effects such as weather, airspace boundaries, terrain obstacles; status of the flight involving required

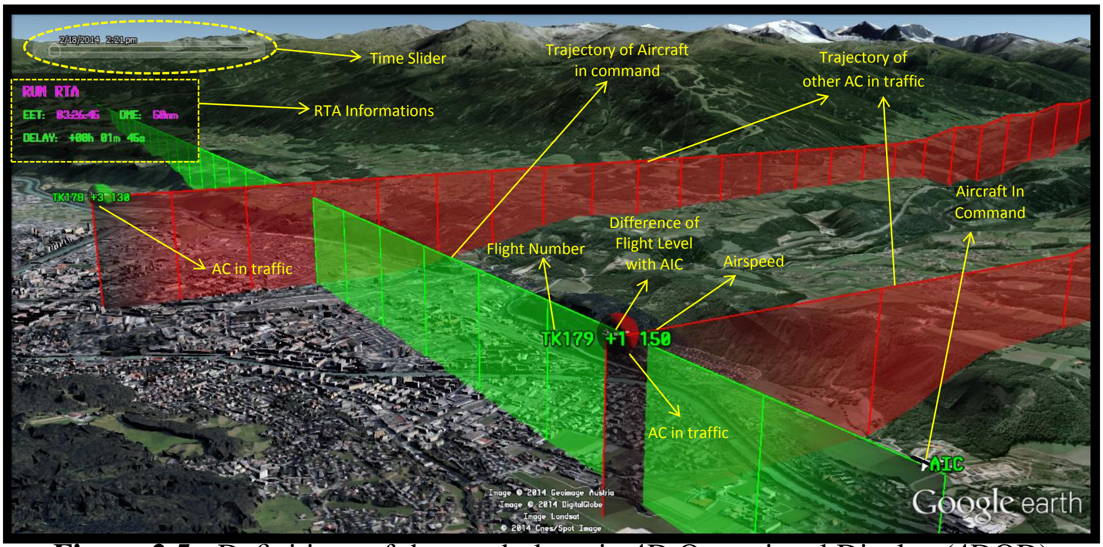
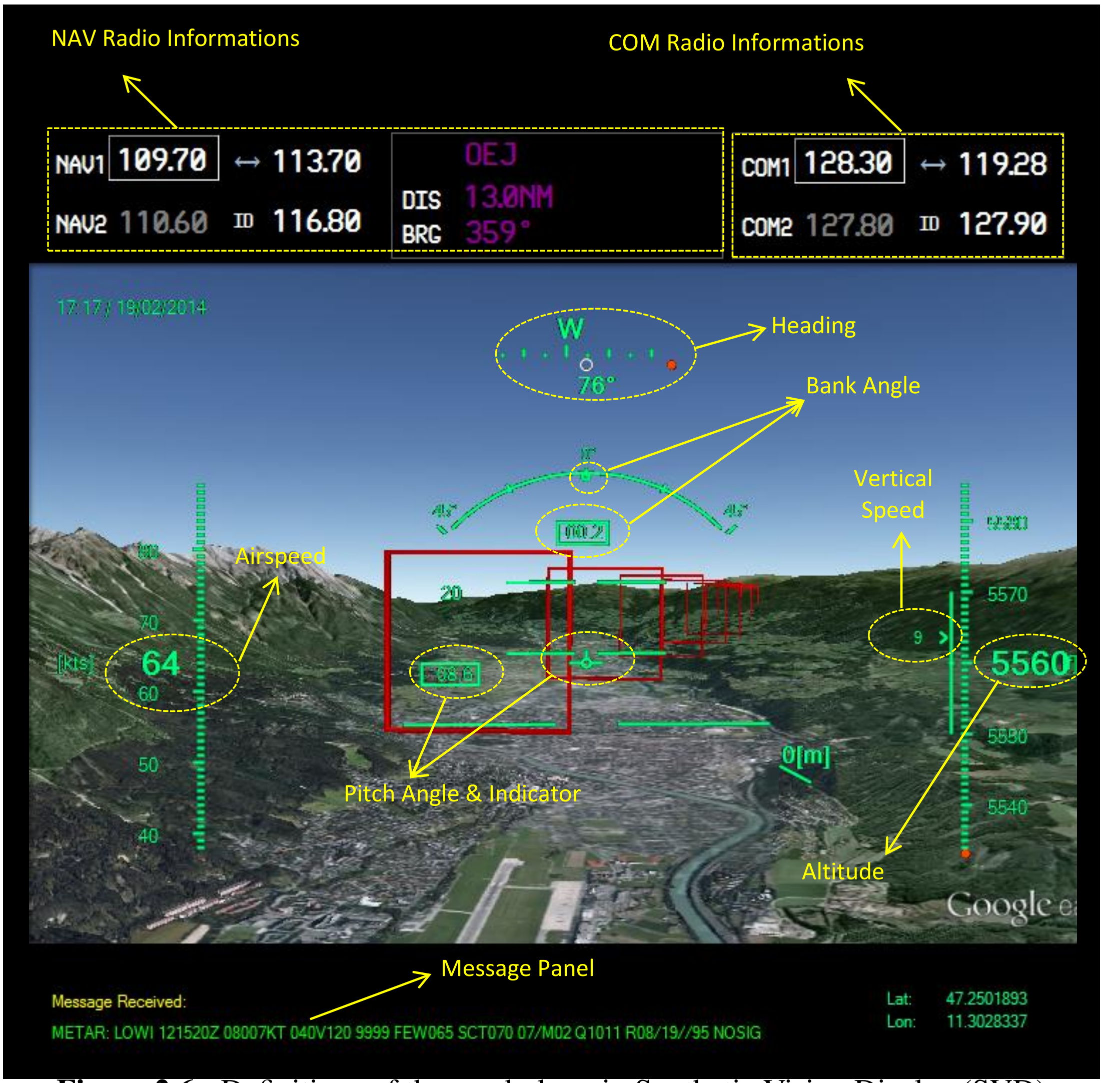


Figure 2.5 : Definitions of the symbology in 4D Operational Display (4DOD)

time of arrival objectives, delays, estimated capacity of the airspace; and safety related warnings such as conflict probability predictions. The display gives 3D visualisation ability to the pilot as supervisor, and he/she can easily change supervisor look-angle and look-position using haptic interfaces. For experimental purposes, two types of haptic interfaces have been included; an external trackpad and 3D navigator mouse; which both provides better 3D navigation on the operational map overlay. The flight crew can also monitor future projection of the trajectories using the time slider button on the screen, or initiating fast time simulation of the flight. This is where the third dimension (time) perception is provided to the user. Specifically, the flight crew a) can see the flight trajectories of the ownship and surrounding aircraft in 2D map overlay, in a traditional way; b) may choose to go into details using 3D navigation (e.g. around the potential conflict); c) are able to go forward on time to see the projected future; and d) even may chose to perform fast time simulation for entire or specific part of the flight. The Figure 2.5 gives definition for main symbols in 4DOD.

The 4DOD is envisioned to increase not only "transient situational awareness" but also enhance fully understanding the entire flight operation. In the context of the 4D trajectory based operation, it has to be handled Required Time Arrival (RTA) objectives and neutralised delays in the air in order to obtain both optimal flight regimes and efficient use of the airspace. The 4DOD also demonstrates these types of information to the flight crew. In the collaborative negotiation with the ground segments, the flight crew can evaluate these objectives and performance scores (both in time and fuel efficiency) of the processing trajectories and their alternatives result in custom modifications. Through this screen, the flight crew can accept the trajectory on which ATC requested negotiation; or can modify existing trajectory by adding or removing fixes and then puts it on the ATC for acceptance. This communication is handled via air-to-ground data link and formal intent languages.



NAV1 109.70 ↔ 113.70
NAV2 110.60 ID 116.80
OEJ
DIS 13.0NM
BRG 359°
COM1 128.30 ↔ 119.28
COM2 127.80 ID 127.90
17:17 / 19/02/2014

Message Received:
METAR: LOWI 121520Z 08007KT 040V120 9999 FEW065 SCT070 07/M02 Q1011 R08/19//95 NOSIG

Lat: 47.2501893
Lon: 11.3028337

Figure 2.6 : Definitions of the symbology in Synthetic Vision Display (SVD)

The Synthetic Vision Display (SVD) gives the pilots synthetic vision and also incorporates additional guidance and operational information. In addition to standard motion related information such as airspeed, vertical speed, altitude and inertial angles; the envisioned screen also demonstrates planned/negotiated trajectory through the "tunnel-in-the-sky" demonstration. The pilot can operate the entire flight without having to look up in case of the low visibility flight operations. Tunnel visualisation also gives a continuous perception across the whole trajectory from surface operation to landing with glide slope. In addition to synthetic terrain visualisation, It also enables to visualise the weather through the METAR data; and other soft obstacles such as closed airspace (segregated for other users), airspace constrained altitude levels and high loaded traffic volumes. The definitions for the symbology in the Synthetic Vision Display (SVD) have been given in Figure 2.6.

2.3 Augmented Reality Based Head-up Display

The proposed structure of the Head-Up-Display (HUD) seen in Figure 2.7 aims to present all essential flight information in the pilot's forward field of view eliminating the need of continually transition from head-down instruments to head-up. It is envisioned that HUD provides "informational summary" about the transient status of the flight including near-term objectives. In addition to presenting flight path marking, flight path acceleration, speed and altitude meters, glide-slope angle, and runway aim point demonstrations, similarly as in SVD, negotiated continuous trajectory demonstration is provided through "tunnel-in-the-sky".

The demonstration of "tunnel-in-the-sky" is obtained through a combination of all tactical level informations such as negotiated trajectory, airport location, glide-slope angle, take-off/landing runway with clearance, all come from Flight Management System (FMS). The negotiated trajectory information at all phases (including land operations, take-off en-route and landing) is transformed into virtual tunnel visualisation in order to aid the pilot. It is aimed that pilot can operate the entire flight by following the demonstrated virtual tunnel ensuring safety. In addition to path curvature and torsion mostly associated with ascent/descent and turn actions of the aircraft, continuously streaming lights at the corner of the tunnel frames provides the pilot effective flight direction perception. The brief descriptions of the nominal HUD symbology can bee seen in Figure 2.8.

The transparent head-up-display screen also enables to demonstrate text based pop-up message boxes to give high-level status information. Required Time Arrival (RTA), which is one of the important concepts of the 4D Trajectory management, is demonstrated with the related information such as; next destination fix name, remaining distance and negotiated RTA. In addition to this, an another coloured message box (i.e. green for positive and red for negative values) shows predicted delay time for the next fixes. It also enables to demonstrate pop-up messages for the

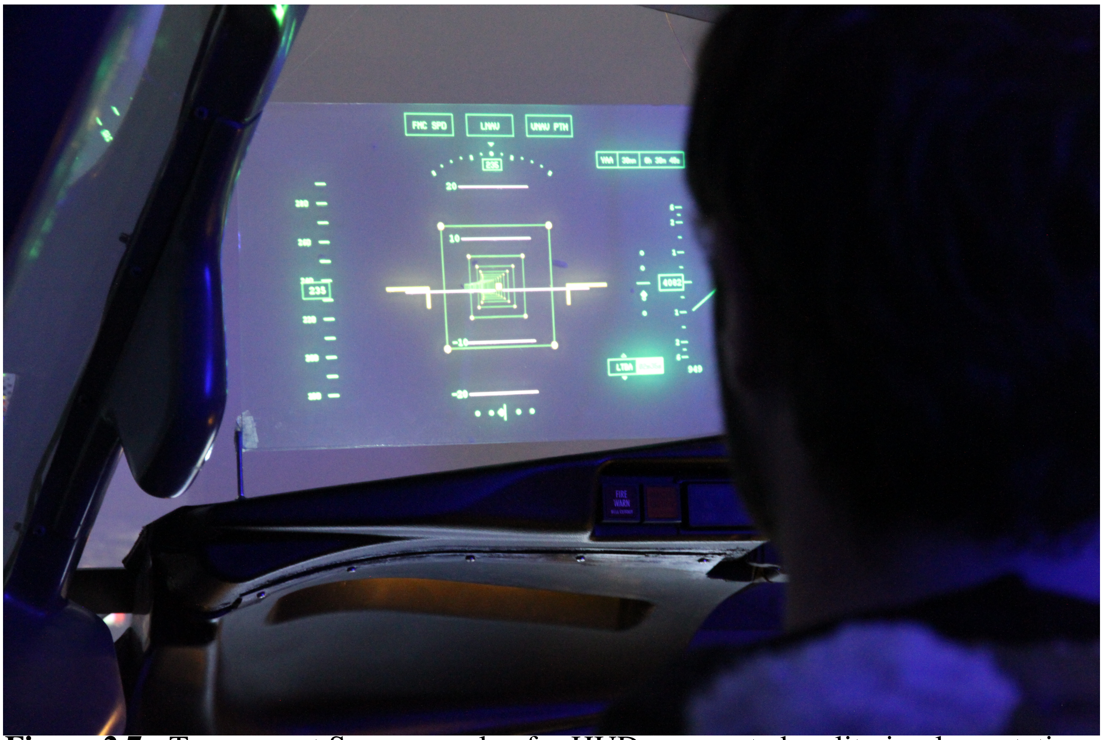


Figure 2.7 : Transparent Screen overlay for HUD augmented reality implementations

check lists according to related situations (e.g. engine start-up, emergency and required traffic and conflict avoidance messages etc.)

The integration of these new implementations and other possible applications on the current flight decks will possibly saturate the flight crew with a huge amount of information and interaction loads. In order to make possible the pilot to perform these redefined tasks, they should be supported with a new form of decision support tools utilizing recent interface and visualisation technologies. Therefore, conceptual visual decision support tools are also added to the envisioned flight deck structure. The integration of such tools aims to keep the pilot in-the-loop, but at a higher level where he or she is managing the operation, understanding newly evolving situations and analysing the potential solutions with their alternatives.

Two forms of visual decision support interfaces, Augmented Reality Based Head-Up Display and Synthetic Vision Display, are introduced in this module. In the nominal flight operation, it is expected that the pilot cooperates with automated tools by interacting them through the interfaces of the visual decision support tools. 2D and 3D information visualisation is created on both Augmented Reality Based Head-Up Display and Synthetic Vision Display by fusing all tactical level data obtained from both on-board sensing and air-to-ground data exchange. The flight

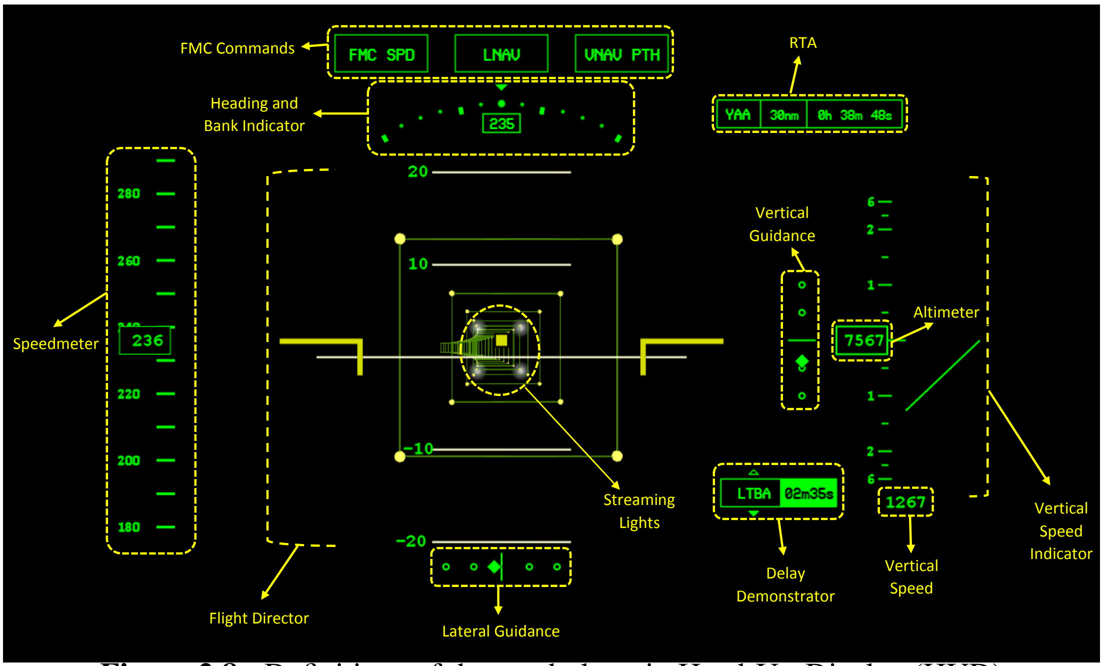


Figure 2.8 : Definitions of the symbology in Head-Up-Display (HUD)

crew is continuously supported with information flow about the current state of the operation (e.g. RTA objectives, delays and ascent/descent slope) and environment (e.g. surrounding traffic, potential loss of separation, proximity to the terrain). The 3D visualisation use gives the pilot sense that he or she is flying in the virtual tunnel (along the track of the trajectory) appearing in the sky. Other objects such as terrain, obstacles, aircrafts are visualised with graphical and textual objects. The Figure 1.1 shows this application while running in our experimental synthetic display. This tunnel-in-the-sky visualisation can be also used in head-worn displays (HWD) and goggles enabling to create spatially integrated and large field-of-view augmentation. We are currently using aligned projector layers to create this effect in our integrated test platform . This visual demonstration aims to prevent "tunnelling effect" results in degraded flight deck efficiency and reduced safety margins. In the analysis of the American Airlines Flight 965 accident at Cali, Columbia, it is reported that the flight crew spent a huge amount of time heads-down trying to program the FMS to perform an infrequent task, which is not well supported by the interface, and this inefficient interaction contributed to the occurrence of the accident with several other circumstances [46].

During the intent negotiation process, which is an interaction intensive process, the Synthetic Vision Display demonstrates processing flight intent to the pilot and enables required interaction to accept, modify or request replanning. Through the 2D visualisation of Synthetic Vision Display (see the Figure 1.1), the flight crew can readily understand the results of any modification (in both space and time) on processing flight intent. The touch screen interface allows the pilot to easily modify the requested intent by interpreting resulting trajectories. The potential conflicts, which are detected by onboard *Conflict Monitoring*, are also demonstrated in terms of space and time, so the flight crew can analyse the situation and evaluate the required solution.

3. PROBLEM DEFINITION

The trajectory generation problem can be given in two parts [19]. First part is *feasibility*, which refers to performable actions of the aircraft within its limits of maneuverability, performance, and control as well as environmental and operational constraints. The second part is *optimality*, which considers costs in terms of fuel and time.

Consider the following general form of the time-invariant dynamics of the aircraft;

$$\dot{x}(t) = f(x(t), u(t)), \quad x(0) = x_0, \tag{3.1}$$

where the $x(t) \in X \subseteq \mathbb{R}^n$, $u(t) \in U \subseteq \mathbb{R}^m$ such that $n, m \in \mathbb{N}$ and the state $x_0 \in X$ is called the initial state of the ownship. Similarly let $\chi_j(t) \in X \subseteq \mathbb{R}^n$, $\vartheta_j(t) \in U \subseteq \mathbb{R}^m$ represent the predicted trajectory set and the control input set for the reachable set of the surrounding aircraft. Let X_{obs} and X_{arr} represent the obstacle region (static obstacles) and arrival region respectively. Then we can define the conflict-free space depending based on time (due to dynamic conflict avoidance) as $X_{free}(t) : X \setminus X_{obs} \cup X_{sep}(t)$, where $X_{sep}(t)$ denotes the set of regions centered at ${}^*x_j(\tau)$ such that $\chi(\tau) = \bigcup {}^*x_j(\tau)$ for all $t \in [0, \tau]$. Here, ${}^*x_j(\tau)$ represents all states for all aircraft that can be reached from an initial state $x_j(0)$ at time $\tau > 0$. This region is defined by a set of aircraft separation cylinders. So, a dynamically-feasible trajectory in X_{free} (an abbreviation for $X_{free}(t)$) starts at x_{init} and ends in the arrival region X_{arr}.

Feasibility: The feasible trajectory generation problem can be defined as finding a feasible trajectory if one exists or otherwise report a failure [47]. For a bounded connected open set $X \subset \mathbb{R}^n$, and obstacle region $X_{obs} \subset X$, an initial state $x_{init} \in X_{free}$ and a arrival region $X_{arr} \subset X$, a feasible trajectory is $x : [0, \tau] \rightarrow X_{free}$ such that $x(0) = x_{init}$ and $x(\tau) \in X_{arr}$, if one exists.

Let $J : x_{X_{free}} \rightarrow R_{>0}$ be a cost function for all collision-free trajectories. The optimality problem of trajectory planning can be defined as generating a feasible trajectory with minimum cost [47].

Optimality: For a given bounded connected open set $X \subset \mathbb{R}^n$, and obstacle region $X_{obs} \subset X$, an initial state $x_{init} \in X_{free}$ and a arrival region $X_{arr} \subset X$, find a trajectory $x^* : [0, \tau] \rightarrow cl(X_{free})$ such that;

(i) $x^*(0) = x_{init}$ and $x^*(\tau) \in X_{arr}$,
(ii) $J(x^*) = min_{x \in \sum_{cl(X_{free})}} J(x)$,

where $cl(X_{free})$ denotes closure of X_{free}. Moreover, by considering the local trajectory generation, for two path segments $x_1, x_2 \in \Sigma_{X_{free}}$, let the concatenation of two paths be $x_1|x_2 \in \Sigma_{X_{free}}$, then the cost function should satisfy;

(i) $J(x_1|x_2) \geq J(x_1)$,
(ii) $J(x_1|x_2) = J(x_1) + J(x_2)$.

The overall 4D-trajectory generation problem is to find a proper trajectory sequence with states and effective time interval representations that are subject to dynamic modification through conflict resolution. That is:

$$\pi(t_0 : t_{end}) = \{(x_0, \tau_0), (x_1, \tau_1), \ldots, (x_{end}, \tau_{end})\}. \tag{3.2}$$

where $t_\varepsilon > 0$, and $t_{current} < t_0 - t_{min\_action}$. This ensures that the solution trajectory begins with at least the minimum required time to perform a safe action before the first collision. The each x_i is the minimum-cost trajectory segment that is generated by *Local Trajectory Generation.* The proper sequence $pi(t_0 : t_{end})$, which will be sent to the flight management system (FMS) to control the aircraft, is generated through the *Tactical 4D Trajectory Planning* to find an asymptotically optimal trajectory.

The following sections first explains the Aircraft Performance Model that is used, and then give the cost-efficient *Local Trajectory Generation* and finally *Tactical 4D Trajectory Planning* algorithm.

4. AIRCRAFT PERFORMANCE MODEL

The proposed Aircraft Performance Model (APM) heavily depends on the Base of Aircraft Data (BADA) 4 dataset [48]. The operational version, BADA 3, does not include all the relevant physical dependencies and provides little flexibility. Moreover, BADA 3 does not allow for dynamic calculation of the forces acting on the aircraft, which are the functions of the input variables used in APM. Specifications in BADA 3 only make it possible to evaluate maximum climb/take off and maximum cruise/descent thrust forces and their nominal values. These descriptions are not open for optimization. BADA 4, currently under development, aims to meet advanced functional and precision requirements of new ATM systems. For example, it provides a generalized thrust model as a function of the throttle parameter and Mach number. Such generalized expressions allow us to develop advanced cost-optimization procedures utilizing modal parametric definitions of aircraft performance. We have rigorously studied the BADA 4 dataset and integrated all the relevant functions into our three-degrees-of-freedom (3DOF) motion equations and cost-efficient trajectory calculations, which are essential for effective cost minimization. Figure 4.1 demonstrates our parameter handling in trajectory generation.

Aircraft Performance Model (APM) details an aircraft's performance parameters operational limits. Specifically, the BADA 4 dataset includes Aerodynamic Forces and Configurations Model (AFM) for drag and lift coefficient calculations, Propulsive Forces Model (PFM) for thrust and fuel coefficient calculations, and Aircraft Limitation Model (ALM) for identifying geometric, kinematic, dynamic, and environmental operation limitations. The Operation of Configuration Parameters Model (OPM) defines transition time for both high-lift devices and landing gear configurations [49]. The interactions between these different models can be seen in Figure 4.1.

The following 3DOF motion equations are considered sufficient to describe the aircraft dynamics in an air traffic management (ATM) context.

$$
\begin{aligned}
\dot{V_{TAS}} &= \frac{T-D-W\sin\gamma}{m}-\dot{w_1} && \textbf{(4.1)}\\
\dot{\chi_h} &= \frac{\frac{L\sin\mu}{m}+(\dot{w_3}\sin\mu-\dot{w_2}\cos\mu)}{v\cos\gamma} && \textbf{(4.2)}\\
\dot{m} &= -F && \textbf{(4.3)}\\
\dot{\lambda} &= \frac{V_{TAS}\cos\gamma\sin\chi+w_2}{(N_c+h)\cos\varphi} && \textbf{(4.4)}\\
\dot{\varphi} &= \frac{V_{TAS}\cos\gamma\cos\chi+w_1}{(M_c+h)} && \textbf{(4.5)}\\
\dot{h} &= V_{TAS}\sin\gamma, && \textbf{(4.6)}\\
L &= \frac{W\cos\gamma-m(\dot{w_3}\cos\mu+\dot{w_2}\sin\mu)}{\cos\mu}. && \textbf{(4.7)}
\end{aligned}
$$

where $[V_{TAS},\chi_h,m,\lambda,\varphi,h]\in X\subseteq\mathbb{R}^n$ denote the states of the aircraft and represent the true airspeed, true airspeed yaw, mass, latitude, longitude and altitude of the aircraft, respectively. $[\gamma,\delta_T,\mu]\in U\subseteq\mathbb{R}^m$ are the control variables that represent the flight path angle, the throttle parameter and the aerodynamic bank angle. $W[N]$ is the aircraft weight, $D[N]$ is the total drag, $T[N]$ is the total thrust, $L[N]$ is the total lift force and F is the fuel consumption rate in $[kg/s]$. M_c is the ellipsoid radius of curvature in the meridian plane and N_c is in the prime vertical according to the WGS84 earth model. The wind gradients are represented by $\dot{w_1}$, $\dot{w_2}$ and $\dot{w_3}$, which are defined in a proper axes system. $\Delta=[\delta_{HL},\delta_{LG},\delta_{SB}]$ is the configuration input set where δ_{HL} is the position of the high lift devices, δ_{LG} is the landing gear configuration and δ_{SB} is the speed break configuration. The *Earth Model* is described by the vector $E_m=[\delta,\theta,g,w]$, where δ is the local pressure ratio, θ is the local temperature ratio, $g[m/s^2]$ is the local acceleration of gravity and $w[m/s]$ is the local wind speed vector.

The equations for the drag force D and the weight W are given as:

$$D=\frac{1}{2}\kappa p_0\delta M^2SC_D,\quad W=mg, \qquad \textbf{(4.8)}$$

where κ is the adiabatic index, p_0 is the pressure at sea level and S is the wing area. M is the Mach number, $M=\frac{V_{TAS}}{a}$, where a is the speed of sound and v is the true airspeed. C_D is the drag coefficient, which is defined as a function of the Mach number, the lift coefficient and the configuration of the aircraft Δ:

$$C_D=C_D(M,C_L,\Delta), \qquad \textbf{(4.9)}$$

The lift coefficient C_L is calculated from the lift force in Eq. 4.7

$$C_L = \frac{2L}{\kappa p_0 \delta M^2 S}. \tag{4.10}$$

The total thrust T and fuel consumption rate F are given as:

$$T = W_{MTOW} \delta_T C_T \tag{4.11}$$

$$F = W_{MTOW} a_0 \delta_T \sqrt{\theta} C_F, \tag{4.12}$$

where $W_{MTOW} = m_{MTOW} g$ is the maximum take-off weight, a_0 is the speed of sound at sea level and δ_T is the throttle control parameter, which is an abstraction of the real flight control available for the pilot and the *flight management computer* (FMS) to manipulate the thrust. $C_T = C_T(M, \delta_T)$ is the thrust coefficient and $C_F = C_F(M, \delta_T)$ is the fuel coefficient. These coefficients are obtained from *Propulsive Forces Model* part of BADA 4.

In summary, given the states $X(t_k)$ and inputs $U(t_k), \Delta(t_k)$ at time t_k, the derivatives in Eqs. 4.1-4.6 need be calculated to obtain the state configuration $X(t_{k+1})$ by integration. To calculate the derivatives, first the lift force L is calculated using the Eq. 4.7. Once the lift L is calculated, the lift coefficient can be determined from the Eq. 4.10. Next, the drag coefficient is computed using the Eq. 4.9, depending on the configuration of high lift devices, speed brakes and landing gears. Then the thrust and fuel coefficients are calculated. Next, the drag D, the thrust T and the fuel consumption F are computed from Eqs. 4.8, 4.11 and 4.12 respectively. Once these forces are obtained, the state derivatives in Eqs. 4.1-4.6 can be integrated for trajectory propagation. At every integration step, the feasibility of certain parameters and states is checked by the *a/c limitation model*, which is defined in the ALM part of BADA 4.

The constraints and limitations on the state and control variables are checked at each step by the ALM. In BADA 4, the performance limitations are categorized into five distinct models; *geometric, kinematic, buffet, dynamic* and *environmental* models. These constraints are given as follows,

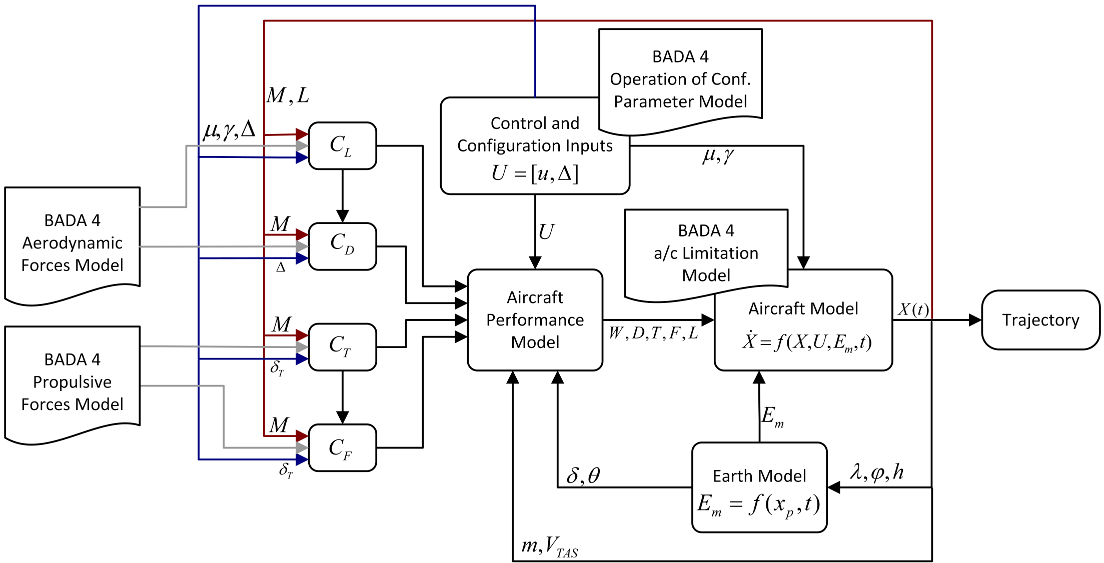


Figure 4.1 : Aircraft local trajectory generation through the Aircraft Performance Model and parametric definitions in BADA 4

$$max[0,h_{as_{min}}] \leq h_{Hp} \leq min[h_{Hp_{max}}(\delta_{HL}),h_{as_{max}}] \tag{4.13}$$

$$M \leq M_{M_0}(\delta_{LG}) \tag{4.14}$$

$$\delta_{T_{min}} \leq \delta_T \leq \delta_{T_{max}} \tag{4.15}$$

$$V_{CASstall}(\delta_{HL},\delta_{LG}) \leq V_{CAS} \leq V_{M_0}(\delta_{HL},\delta_{LG}) \tag{4.16}$$

$$V_{CAS} \leq 250 \text{ kts for } h \leq 10,000 \text{ ft} \tag{4.17}$$

$$0 \leq C_L \leq C_{Lmax}(M,\delta_{HL},\delta_{LG}) \tag{4.18}$$

$$m_{min} \leq m \leq m_{MTOW} \tag{4.19}$$

$$n_{min}(\delta_{HL}) \leq n \leq n_{max}(\delta_{HL}), \tag{4.20}$$

where, $h_{as_{min}}$ is the minimum altitude allowed in the airspace, h_{Hp} is the geopotential pressure altitude, $h_{Hp_{max}}(\delta_{HL})$ is the maximum geopotential pressure altitude when the high-lift devices are applied and $h_{as_{max}}$ is the maximum altitude allowed in the airspace. $M_{M_0}(\delta_{LG})$ denotes the maximum operating Mach number depending on the landing gear configuration. $\delta_{T_{min}}$ is the minimum throttle setting and $\delta_{T_{max}}$ is the maximum throttle setting. V_{CAS} is the calibrated airspeed, $V_{CASstall}(\delta_{HL},\delta_{LG})$ is stall calibrated airspeed depending on the high-lift device and landing gear configuration. $C_{Lmax}(M,\delta_{HL},\delta_{LG})$ represents the function for the maximum lift coefficient depending on the Mach number, high-lift device and landing gear configuration. m_{min} is the minimum operating mass while m_{MTOW} is the maximum take-off mass. $n_{min}(\delta_{HL})$ is the function for the minimum loading factor depending on the high-lift device configuration and $n_{max}(\delta_{HL})$ is the maximum loading factor function depending on the high-lift device configuration.

The certified operating ceiling altitude h and the maximum geopotential pressure altitude $h_{Hp_{max}}(\delta_{HL})$ are obtained from the *Geometric Limitations Model*. This height depends on whether the high-lift devices are deployed. If the high-lift devices are in effect, operational use of certain airspaces limits the maximum and minimum flight altitude as well. The maximum calibrated airspeed $V_{CAS_{max}}$ and Mach number M_{M_0} of the aircraft are obtained from the *Kinematic Limitations Model* for all combinations of the high-lift device δ_{HL} and the landing gear δ_{LG} deployments. The relationship between the true airspeed V_{TAS} and the calibrated airspeed V_{CAS} is given as follows:

$$V_{TAS} = \sqrt{\frac{2p}{\mu\rho}\left\{\left(1+\frac{p_0}{p}\left[\left(1+\frac{\mu\rho_0}{2p_0}V_{CAS}{}^2\right)^{\frac{1}{\mu}}-1\right]\right)^{\mu}-1\right\}}, \tag{4.21}$$

where p is the air pressure, ρ is the air density, μ is a function of the adiabatic index κ, p_0 is the air pressure at the sea level, ρ_0 is the air density at the sea level and the speeds are in $[m/s]$.

Another limitation for the calibrated airspeed V_{CAS} comes from the operational use of airspaces, i.e. V_{CAS} must be under 250 knots while the aircraft is below 10,000 ft altitude. The maximum lift coefficient $C_{L_{max}}$ as a function of aircraft aerodynamic configuration is given in *Buffet Limitations Model* for both clean and non-clean configuration modes. For the *clean configuration*, the maximum lift coefficient is a function of Mach number. For the *non-clean configuration*, corresponding maximum lift coefficients are defined for every combination of the high-lift device and landing gear position. The operational mass m limits are acquired from *Dynamic Limitations Model*. The Dynamic Limitations Model also provides the maximum and minimum load factors n depending on whether the high-lift devices are used or not. Table 4.1 summarizes the equations of trajectory computation and aircraft performance model.

4.0.1 Aircraft Trajectory Cost Definition

The cost-to-travel $J(x^*)$ for a given trajectory x^* is expressed as a combination of fuel cost J_f, time cost J_t and en-route overfly charges J_r, that is;

$$J = c_f\delta m + c_t\tau + \sum_{n} c_{r_i}\delta d_i, \tag{4.22}$$

Table 4.1 : Equations of Trajectory Computation and Aircraft Performance Model

Aircraft EoM	$\dot{x} = f(x, u, \Delta, E, t)$ $x_p = [\lambda, \varphi, h]$
Control and Config. Inputs	$U = [\gamma, \delta_T, \mu]$ $\Delta = [\delta_{HL}, \delta_{LB}, \delta_{SB}]$
Earth Model Eq.	$E_m = f(x_p, t)$
Aircraft Performance Model	$W = f(m, E)$ $D = f(C_D, M, E)$ $T = f(W, M, E, C_T)$ $F = f(W, M, E, C_F)$ $C_L = f(L, M, E, \Delta)$ $C_D = f(C_L, M, \Delta)$ $C_T = f(\delta_T, M)$ $C_F = f(\delta_T, M)$ $\delta_T = f(M, E)$
Aircraft Limitations Model	$g_{lim}(x, \dot{x}, u, \Delta, t) \leq 0$
Operation of Conf. Param. Model	$f : \Delta \rightarrow [0, \tau]$ $\tau_{trans} = f(\Delta_1, \Delta_2)$

where c_f is the per *lb* fuel cost in *cents*, δm is the consumed fuel in *kg*, c_t is the per *hour* time cost in *dollars* and τ is the flight time for a given trajectory segment. En-route overfly charges J_r, are the costs to airspace users by the air navigation service providers (ANSPs). The zone dependent charge notion is defined in [50], such as $c_{r_i} = pt_i$ where t_i is the airspace dependent unit rate per *kilometer[km]* and p is the weight factor, i.e. $p = \sqrt{m_{MTOW}/50}$. In Eq. 4.22, d_i denotes the great circle distance flown over the charging zones and expressed in *kilometers[km].* The entry and exit points to the zones are outlined as filed in the flight plan. Therefore, we accept that these points have been already fixed before the aircraft is airborne. Thus, the rate of this term does not appear in the tactical trajectory optimization procedure. For operational considerations, we also give a definition for the cost index CI, which is a parameter representing the ratio between the time cost and the fuel cost of a flight operation, as defined in [51], i.e. $CI = \frac{c_t\ [\$/hr]}{c_f\ [cents/lb]}$. The Flight Management Computer (FMC) fully utilizes this performance parameter to generate any operational behavior that influences the descent, ascent and cruise modes. The parameter interval varies for different aircrafts, for example, [52] indicates that CI is $0-500$ for the B737-800 and $0-999$ for the B777. For instance, if the pilot commands zero cost index through the FMC interface, the performance behavior yields maximum range airspeed, and the fuel consumption remains at a minimum by ignoring the time-related cost. If the cost index is maximized, flight time will be minimized with maximum climb/descent velocity and cruise Mach numbers, by neglecting the fuel cost. Hence, this index parameter strongly conforms to the *objective*s [3]. In summary, the following tactical cost-to-go for a given trajectory segment is

$$J = c_f(\delta m + CI\tau) + C_r, \tag{4.23}$$

where C_r denotes the fixed en-route overfly charges coming from the last filed flight plan.

5. COST EFFICIENT LOCAL TRAJECTORY GENERATION

The problem of finding an optimal trajectory $J(x^*) = min_{u \in U} J(x)$ in real-time applications is highly complex because the constraints are nonlinear. In this section, we present a multi-modal decomposition approach to trajectory optimization. The multi-modal approach uses approximate solutions via maneuver decomposition instead of solving the problem globally, so it significantly reduces computational complexity and enables real-time local trajectory generation. The required inputs of the trajectory generation algorithm are initial state x_{init}, a reference objective position $x_{wpt} = [\lambda_{wpt} \varphi_{wpt} h_{wpt}]$ and a reference cost index CI. The algorithm returns a dynamically feasible trajectory segment with the proper control input set and their effective time intervals. This local planning approach ignores obstacles and leaves collision checks to the trajectory planner that combines the trajectory segments.

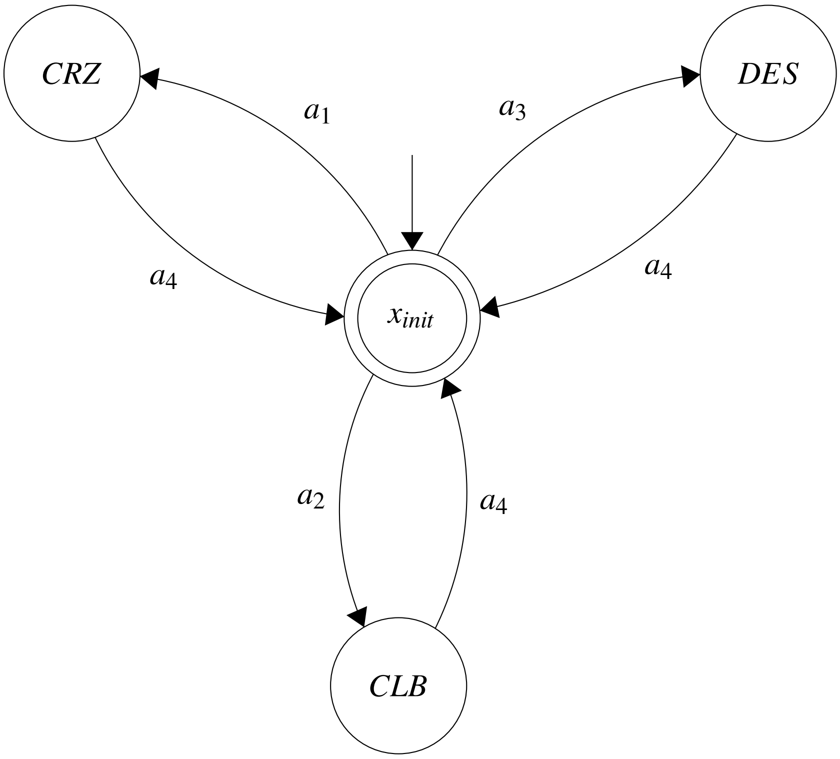
The rationale behind the modal decomposition is to first determine the required flight template and then generate an appropriate maneuver sequence with the corresponding parameter set using a finite maneuver library. The predefined flight templates are *Cruise (CRZ)*, *Climb (CMB)* and *Descend (DES)* and their switching logic is illustrated in Figure 5.1. These flight templates can be also directly mapped to the maneuver library S_m, which is described in Table 5.1.

Table 5.1 : Maneuver library S_m for Maneuver Automaton

ACC	Level-thrust acceleration
DEC	Level-thrust deceleration
CAS	Hold calibrated airspeed
Mach	Hold Mach number
LVL	Level Flight

By following this approach, a proper maneuver sequence can steer the aircraft from the initial state x_{init} to a reference goal point x_{wpt}, if it is applicable. Maneuver sequences with their transition states and time schedules are determined by performing a gradient-descent search of the parameter space of the predefined maneuvers. Once the switching conditions are computed, the complete trajectory along with the corresponding inputs is obtained. Thus, the 3DOF trajectory optimization problem is approximated and finds cost-efficient switching parameters between the predefined maneuver modes. This flight template selection and cost minimization formalism is parallel to the trajectory generation procedure in the flight management systems (FMS). In our structure, the modal maneuver sequence selection in the flight templates differs due to the volume of the region of interest, where our aim is to generate relatively small trajectory segments resolving potential conflicts while the FMS calculates cost-efficient trajectories for the entire flight.

The procedure for maneuver decomposition is given by a two-layer finite state automaton. The first step is to determine the flight template sequence through a higher-level automaton, which is shown in Figure 5.1. This automaton involves basic flight templates, which are cruise (CRZ), climb(CLB) and descent(DES). Specifically, the *Flight Template Automaton* compares the initial altitude h_{init} and the reference altitude h_{wpt} to switch states.



$$a_1 : h_{init} = h_{wpt}, h_{init} \in x_{init}$$
$$a_2 : h_{init} < h_{wpt}$$
$$a_3 : h_{init} > h_{wpt}$$
$$a_4 : x_p = x_{wpt}$$

Figure 5.1 : Flight Template Automaton with *Cruise*, *Climb* or *Descent* templates.

Note that the flight templates and the corresponding maneuver set also involve longitudinal dynamics. To generate 3D motion and keep the aircraft on a horizontal track, a lateral path controller generates the bank angle μ and is coupled with these maneuver modes. To integrate the motion equations, Euler discretization is utilized with a step size of $\Delta t = 0.1$ second. Cost-effective maneuver composition for *Cruise*, *Climb* and *Descent* flight templates are described in the following subsections.

5.1 Cruise

For the *Cruise* template; the objective is to find a proper cruise Mach number $M_{tgt,crz}$ that minimizes the *Economic Cruise Cost Function*. ECCF is a method typically preformed in FMC [49]. Note that the wind speed w is also taken into account in this minimization.

$$dJ = c_f F dt + c_t dt = \frac{c_f F + c_t}{(V_{TAS} + w)} dr \tag{5.1}$$

$$ECCF = \frac{dJ}{c_f dr} = \frac{c_f F + c_t}{c_f (V_{TAS} + w)} = \frac{CI + F}{V_{TAS} + w} \tag{5.2}$$

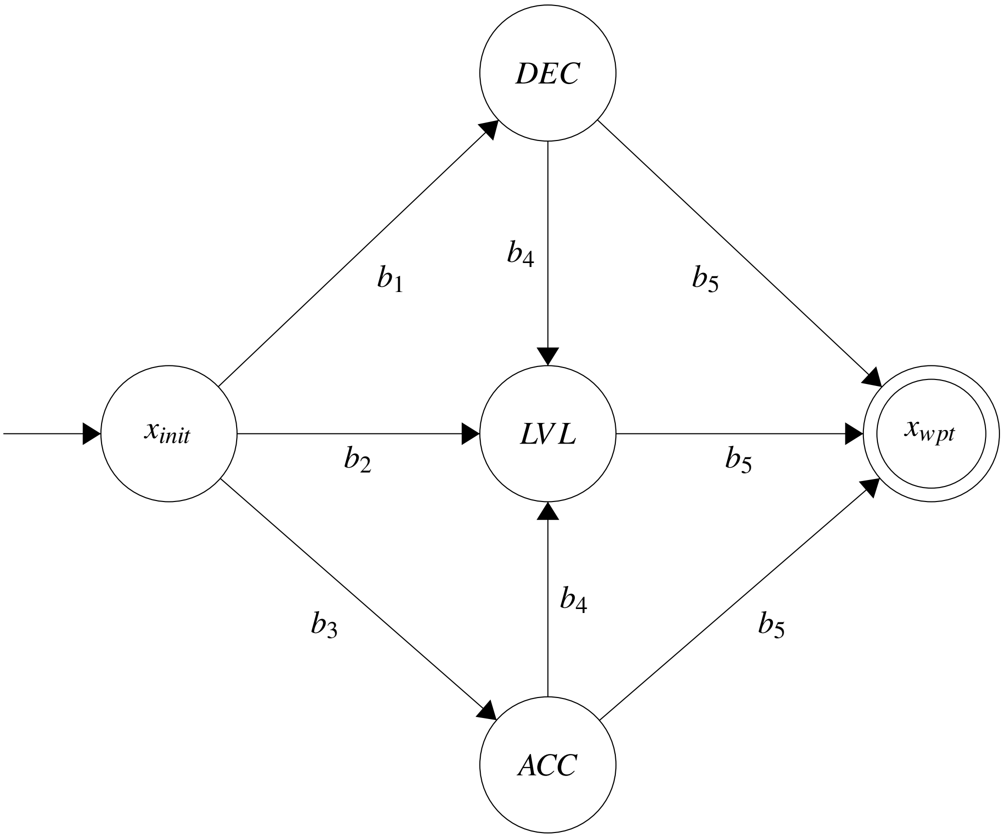
$$M_{cruise} = \{M_i, i \in \mathbb{N}, 0 < M_i \leq M_{mo}\} \tag{5.3}$$

$$ECCF(M_{tgt,crz}) = min_{M \in M_{cruise}} ECCF(M) \tag{5.4}$$

It is supposed that the variations in speed and altitude are relatively small and hence they are ignored. The simplifications for *Cruise* template can be given as follows:

$$L = W, \quad T = D \quad and \quad \gamma = 0 \tag{5.5}$$

This flight template involves a set of maneuvers S_{cruise} = { *Level-Thrust Acceleration, Level-Thrust Deceleration, Level Flight* }. Switching between these maneuvers is controlled by the automaton depicted in Figure 5.2. For the acceleration maneuver *ACC*, the throttle parameter δ_T is set to its maximum limit $\delta_{T max}$ to obtain the maximum thrust available. On the other hand, LIDL (low idle) rating where δ_T is set to its minimum $\delta_{T LIDL}$ to induce a lower thrust T than the drag force D generating



$$b_1 : M_{tgt,crz} < M_{init}, M_{init} \in x_{init}$$
$$b_2 : M_{tgt,crz} = M_{init}$$
$$b_3 : M_{tgt,crz} > M_{init}$$
$$b_4 : M(t + \tau_M) = M_{tgt,crz}, \tau_M \in [0, \tau]$$
$$b_5 : x_p = x_{wpt}$$

Figure 5.2 : Cruise flight template automaton

decelerated maneuver *DEC*. Eventually, the *Level Flight* mode follows each of these maneuvers where the throttle parameter δ_T is adjusted to a certain level, that is;

$$\delta_{T\,lvl} = \Big\{ \delta_T \in [\delta_{T\,min}, \delta_{T\,max}] | T(\delta_T, M) = D(M) \Big\} \tag{5.6}$$

Figure 5.3 shows the Mach number variation for different cost index *CI* values for Boeing 737-800 aircraft flying at an altitude of FL250. The blue curve indicates the smallest cost index (i.e. $CI = 0$), where fuel saving is preferential without considering time related cost, and results in a lower cruise Mach number profile. The effects of the different directions of the wind on the economy cruise Mach number can be seen in Figure 5.4, where the increment in the true airspeed V_{TAS} due to tail wind results in a decreased optimum cruise speed, and head wind shifts the optimal cruise Mach number $M_{tgt,crz}$ to higher values. Note that the numbers for the cost function axes (y-axis) are removed from the plots, as the the scale is completely depends on c_f/c_t value.

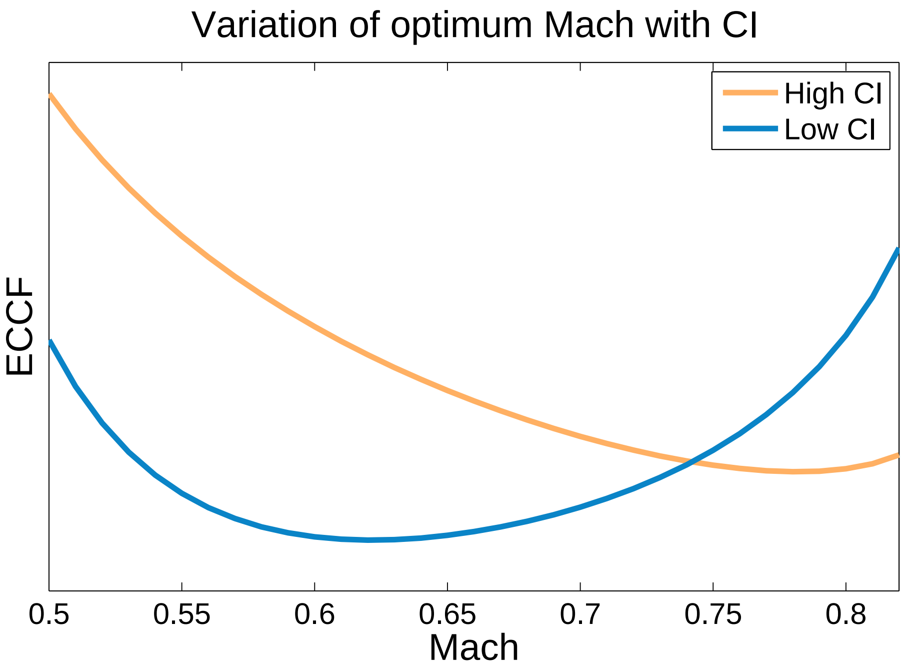


Figure 5.3 : Economy cruise cost function vs. Mach profile for low CI and high CI values

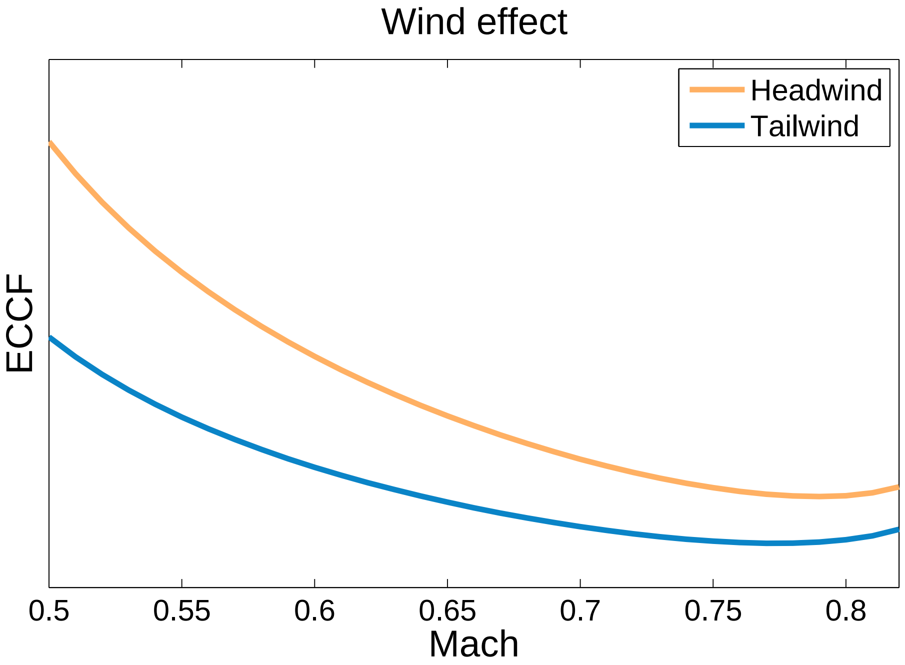


Figure 5.4 : Effects of wind directions in evaluations of optimum cruise Mach profile

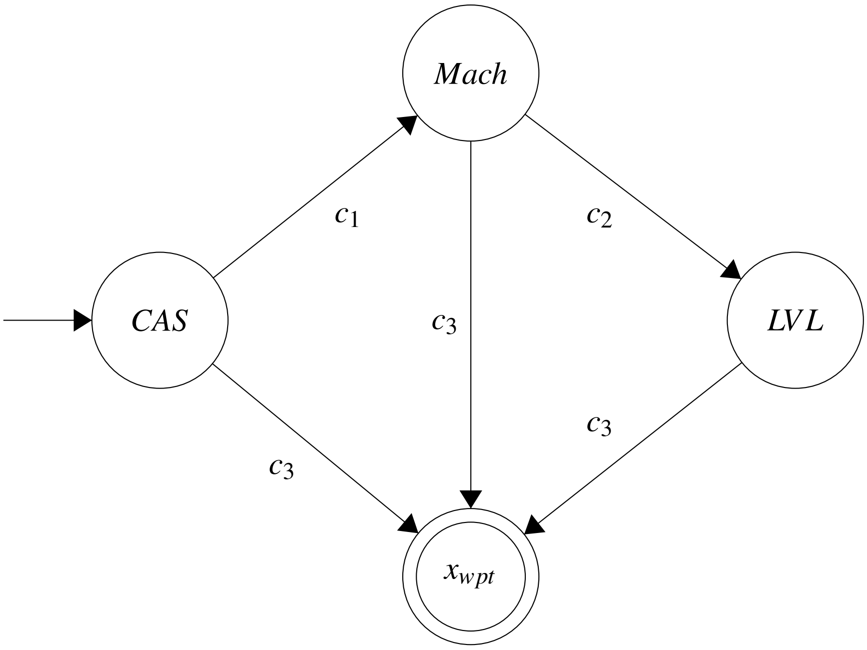
5.2 Climb

The Climb flight template provides a proper Mach number $M_{tgt,cmb}$ profile that minimizes the total cost.

$$M_{climb} = \{M_i, i \in \mathbb{N}, M_{init} < M_i \leq M_{mo}\} \quad \textbf{(5.7)}$$

$$J(M_{tgt,cmb}) = min_{M \in M_{climb}} J(M) \quad \textbf{(5.8)}$$

The Climb template utilizes a set of maneuvers such that $S_{cmb} = \{Hold\ CAS, Hold\ Mach, Level\ Flight\}, S_{cmb} \subset S_m$. Cost effective trajectory generation for *Climb* template involves speed profile scheduling with *CAS/Mach Climb* action. In this regime, the FMS first holds its CAS until a specified Mach number $M_{tgt,cmb}$ is reached. The maneuver is followed by holding the Mach number and adjusting CAS until reaching a reference altitude h_{wpt}. Finally, the aircraft follows the level-flight maneuver until arriving at the reference point x_{wpt}. The finite state automaton given in Figure 5.5 executes these maneuver switching procedures for *Climb* template.



$c_1 : M(t + \tau_M) = M_{tgt,cmb}, \tau_M \in [0, \tau]$
$c_2 : h(t + \tau_h) = h_{wpt}, \tau_h \in [0, \tau]$
$c_3 : x_p = x_{wpt}$

Figure 5.5 : *Climb* flight template automaton

Throttle parameter δ_T is set to maximum climb rating $\delta_{T\,mcmb}$ during climbing. In BADA 4, this rating is formulated as:

$$\delta_T = f(M, \delta) \tag{5.9}$$

where the δ_T is calculated at each step with respect to the Mach number M and atmospheric conditions. This formulation results in one intent equation and leaves the last one to obtain flight path angle γ to maintain the required speed schedule. The equation with a proper discretization depending on true airspeed V_{TAS} can be written as:

$$V_{TASk+1} - V_{TASk} = \Delta t \left(\frac{T_k - D_k}{m_k} - g sin\gamma_k - \dot{w}_1(k) \right) \tag{5.10}$$

During the calculation of path angle seen in Eq. 5.10, following assumptions are employed: a) the derivative of the wind speed $\dot{w}$ is very small relative to the aircraft speed profile, b) For $\Delta t = 0.1$ step size, variations in parameters related to atmospheric conditions such as temperature ratio θ and the pressure ratio δ are assumed to be very small (i.e. $E_m(k+1) \simeq E_m(k)$). Due to these assumptions, the required true airspeed V_{TAS} to perform *Hold CAS* maneuver can be estimated through CAS to TAS conversion through Eq. 4.21. Similarly, in the *Hold Mach* maneuver, equivalent true airspeed V_{TAS} can be stated as:

$$V_{TAS} = \sqrt{\kappa R T_h} M_{tgt,cmb}, \tag{5.11}$$

where T_h is the air temperature and R is the gas constant.

Thus, the flight path angle γ is the only unknown parameter in Eq. 5.10, and can be evaluated from the following expression;

$$\Gamma = \frac{1}{g} \left(\frac{T_k - D_k}{m_k} - \frac{V_{TASk+1} - V_{TASk}}{\Delta t} \right) \tag{5.12}$$

$$\gamma_k = sin^{-1}\Gamma \tag{5.13}$$

The flight path angle γ is given as;

$$\gamma = \begin{cases} \gamma_{max} & \text{if } \sin^{-1}\Gamma > \gamma_{max} \text{ or } \Gamma > 1 \\ \sin^{-1}\Gamma & \text{o.w} \end{cases} \tag{5.14}$$

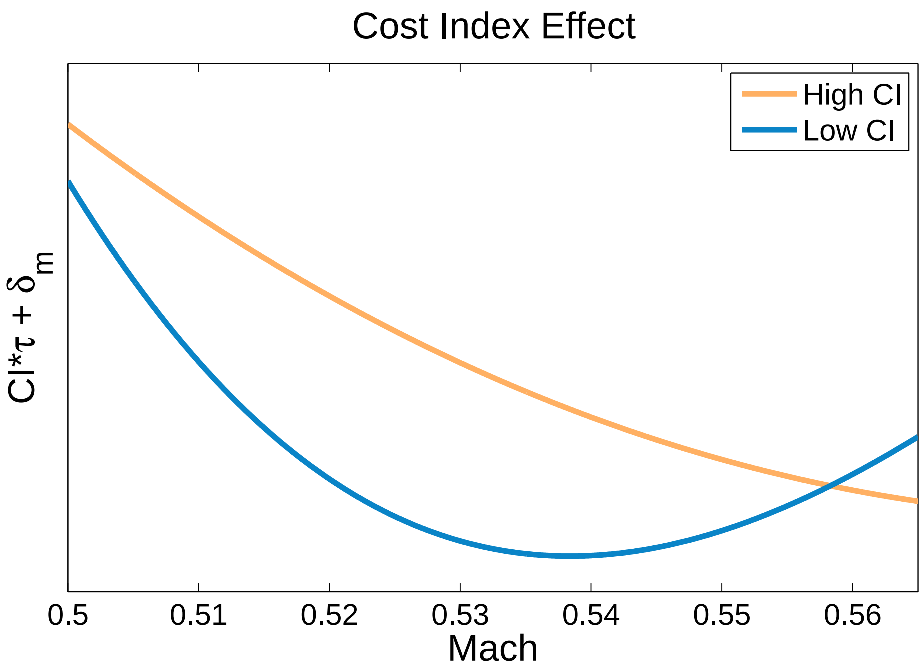


Figure 5.6 : Cost function vs. Mach profile for *Climb* flight template with low and high *CI* values

where $\gamma_{max} > 0$. To demonstrate the cost management for the *Climb*, an example test case where the aircraft climbs to FL210 from FL110 with an initial calibrated airspeed of 252 kts is simulated. Figure 5.6 shows the relation between the cost function and Mach number for the two different CI values. The blue curve denotes the cost curve for a low cost-index (0) and a high cost-index (100) is represented by the orange curve. As can be seen in the figure, the economic climb Mach number is seen at lower rates for lower cost-index values. Figure 5.7 illustrates the performed speed profile for CAS/M climb and level-flight for this scenario. The effect of wind direction is given in Figure 5.8. Figure 5.7 shows that the optimum Mach number $M_{tgt,cmb}$ has higher values with a headwind and lower values with a tailwind.

5.3 Descent

In the *Descent* template, the aircraft starts its descent by adjusting CAS and holding initial Mach number M_{init}. It is followed by decreasing Mach number by holding CAS. After descending to the target altitude h_{wpt}, the aircraft performs steady-level flight. The objective is to find a proper calibrated airspeed CAS profile that minimizes the total cost of the trajectory segment where CAS_{mo} is the maximum operating calibrated airspeed.

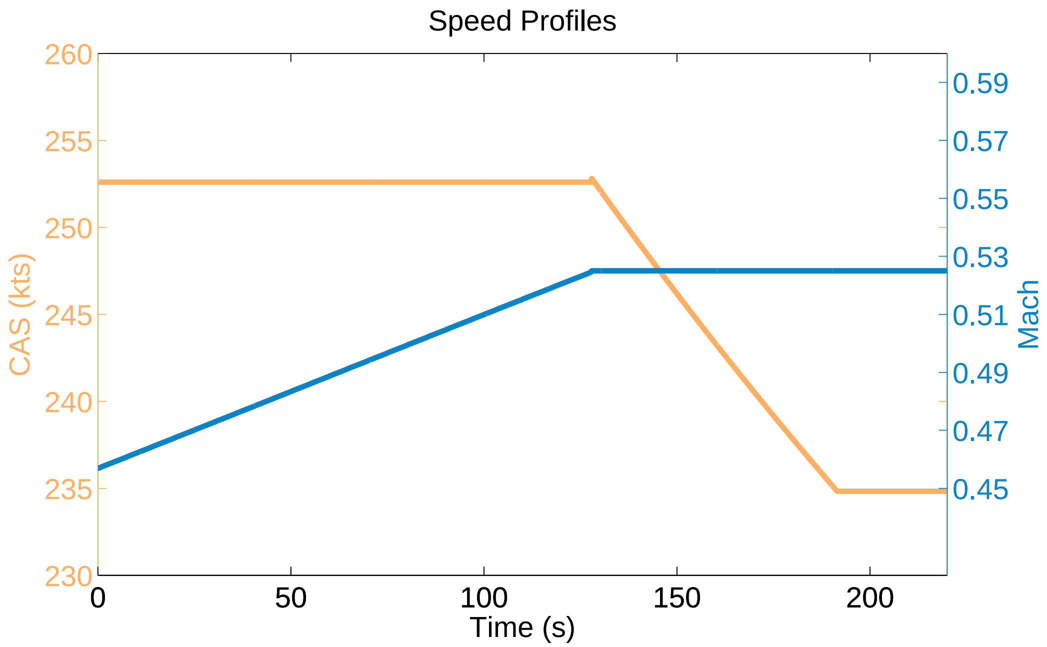


Figure 5.7 : CAS and Mach profiles for the example climb action – climbing from FL170 to FL220

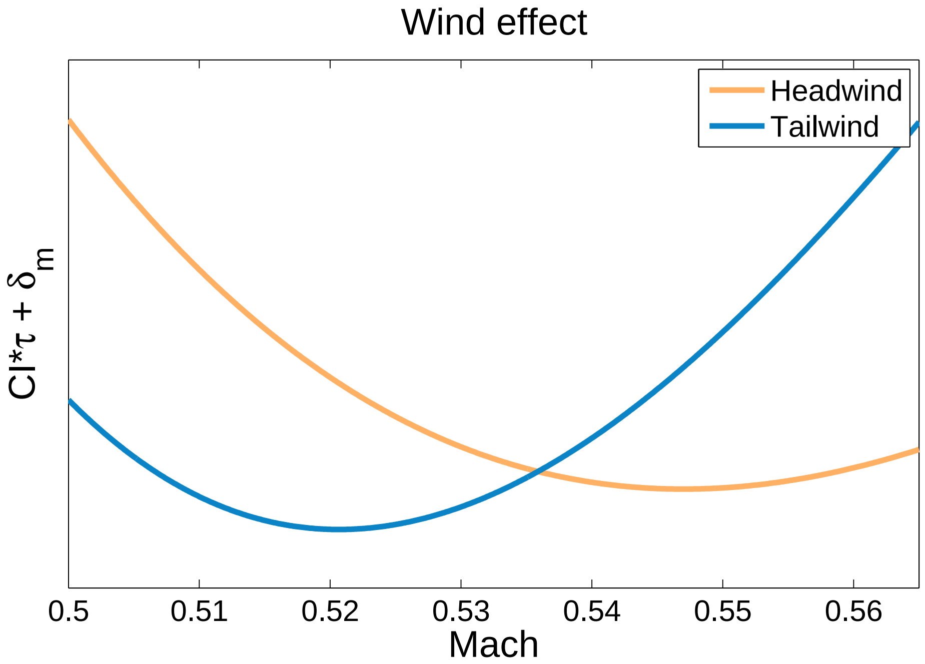
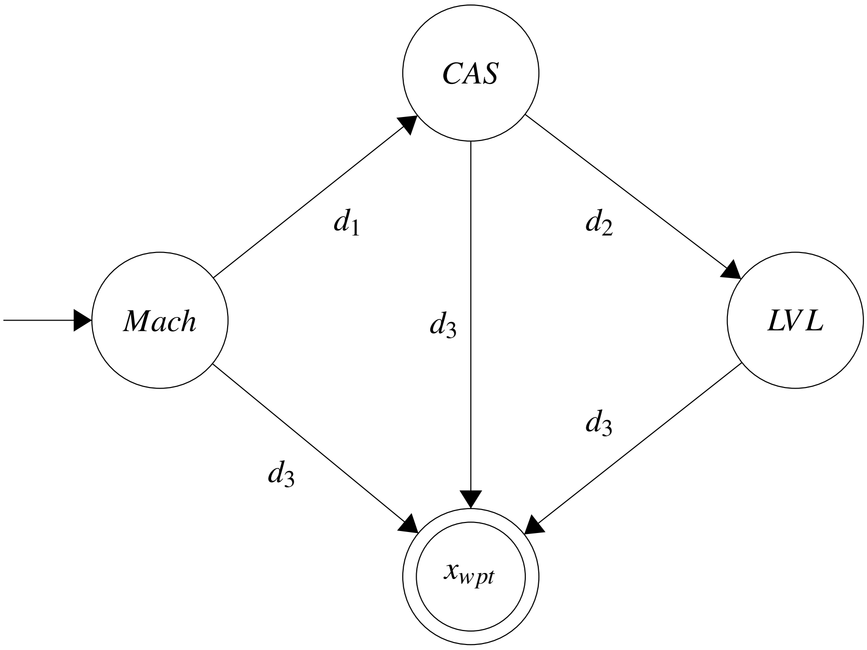


Figure 5.8 : Wind direction effect on optimum climb Mach $M_{tgt,cmb}$



$d_1 : CAS(t+\tau_{CAS}) = CAS_{tgt,des}, \tau_{CAS} \in [0,\tau]$
$d_2 : h(t+\tau_h) = h_{wpt}, \tau_h \in [0,\tau]$
$d_3 : x_p = x_{wpt}$
Figure 5.9 : *Descent* flight template automaton

$$CAS_{descent} = \{CAS_i, i \in \mathbb{N}, CAS_{stall} < CAS_i \le CAS_{mo}\}$$

$$J(CAS_{tgt,des}) = min_{CAS \in CAS_{descent}} J(CAS)$$

Similar to the *Climb* template, this template uses the following set of maneuvers;

$$S_{des} = \{Hold\ Mach, Hold\ CAS, Level\ Flight\}, S_{des} \subset S_m$$

Switching between these maneuvers is controlled by the *Descent* automaton, which is shown in Figure 5.9. Throttle parameter δ_T is set to $\delta_{TLIDL}(M)$ rating and calculated at each step with respect to the Mach number M and atmospheric conditions. This results in a lower thrust T than the drag force D and the aircraft starts to descent. The flight path angle γ is computed at each step using the same approach presented in the *Climb* section. To prevent a steep descent maneuver, flight path angle γ is restricted to a value of $\gamma_{min} < 0$.

$$\gamma = \begin{cases} \gamma_{min} & \text{if } \sin^{-1}\Gamma < \gamma_{min} \text{ or } \Gamma < -1 \\ \sin^{-1}\Gamma & \text{o.w} \end{cases} \tag{5.15}$$

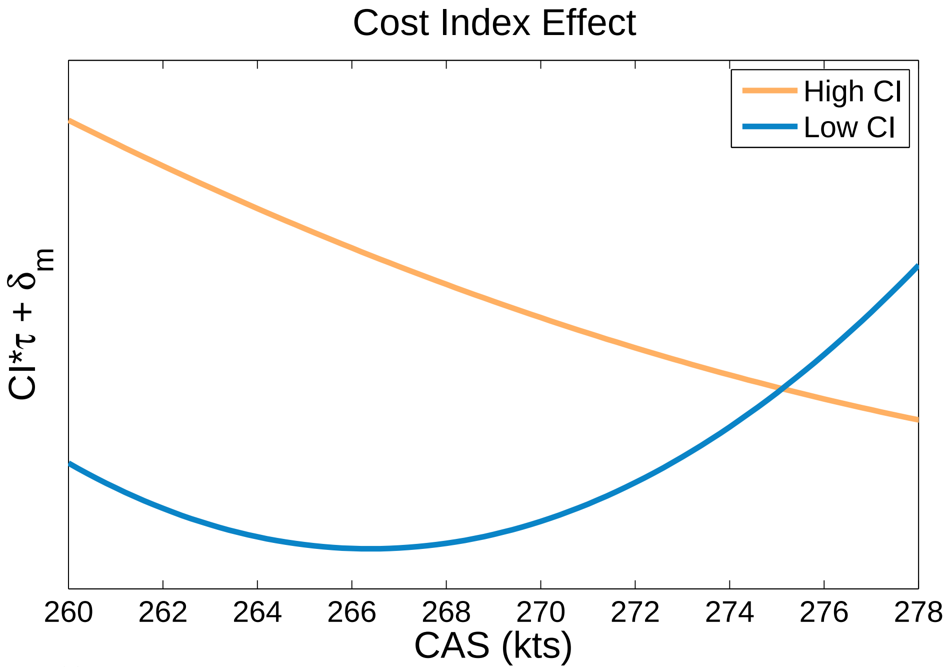


Figure 5.10 : Cost function vs. CAS profile for *Descent* flight template with low and high *CI* values

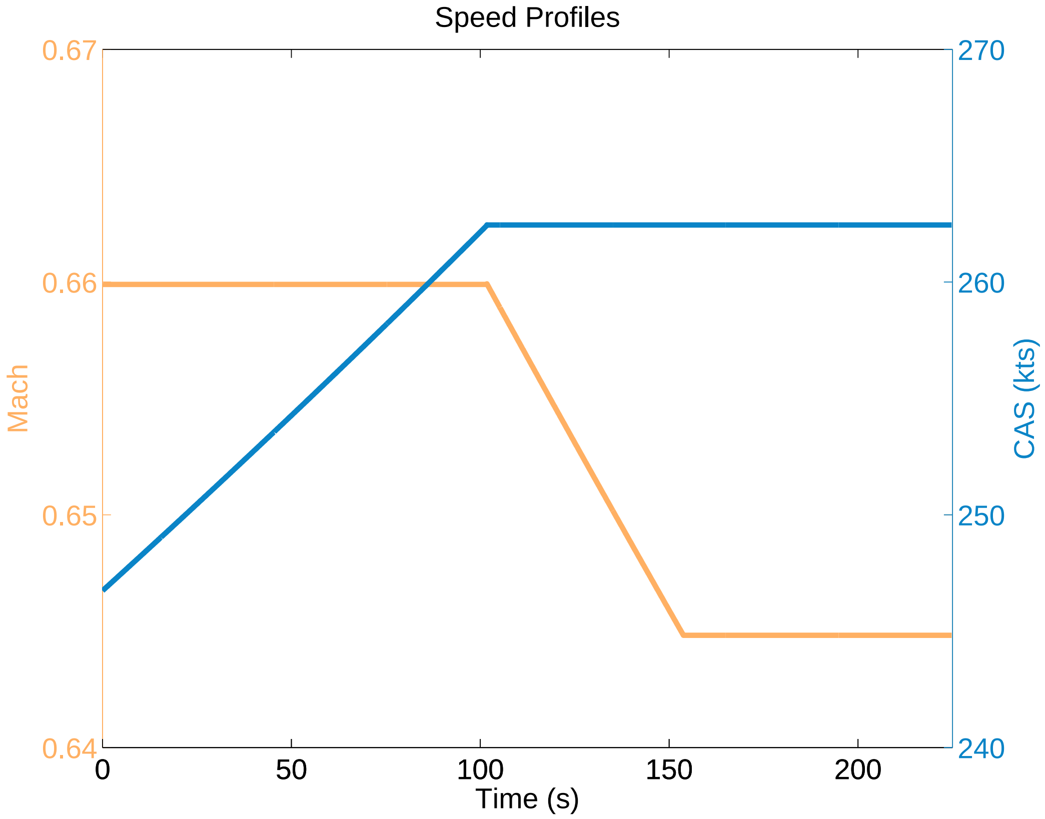


Figure 5.11 : Mach and CAS profiles for the descent template – descending from FL300 to FL260

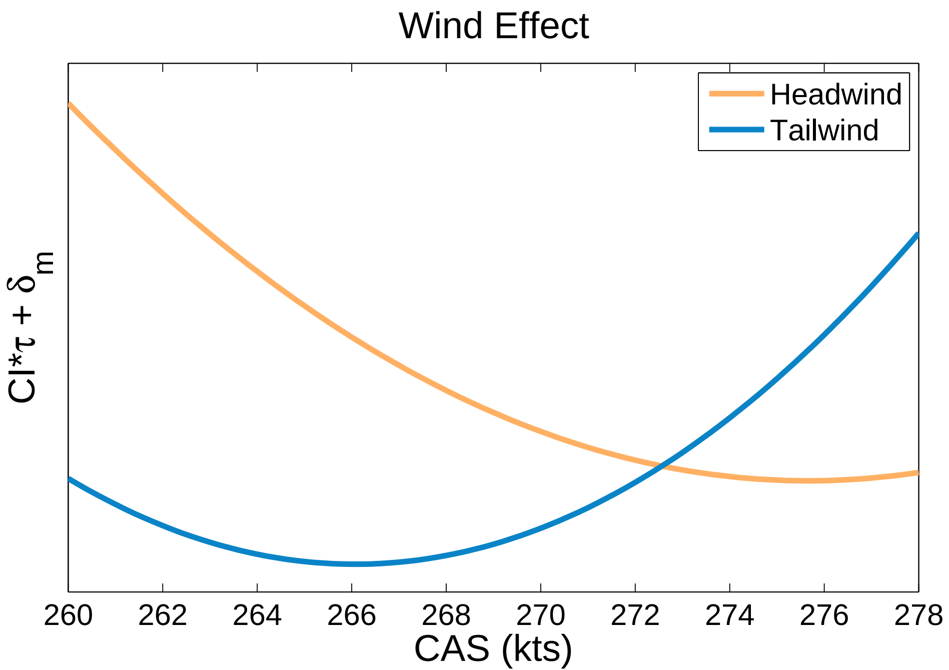


Figure 5.12 : Wind direction effect on optimum descent CAS

In order to demonstrate the cost management for the *Descent* flight template, we simulate a test case where the aircraft descends from FL300 to FL260 with an initial Mach number of 0.66. Figure 5.10 shows the relation between the cost function and CAS value for different *CI* values. The blue curve represents the low cost-index (0) while higher cost-index (100) is depicted with the orange curve. As can be seen in the figure, at the higher cost index, the minimum of the cost function J will be higher, which means the aircraft can descend to the level-off altitude and continue with a level flight while maintaining higher CAS values. Figure 5.11 illustrates the speed profile for M/CAS descend and level-flight for this scenario. The effect of wind direction is given in Figure 5.12 where the optimum calibrated airspeed $CAS_{tgt,des}$ will be higher with a headwind and will be lower in case of tailwind.

For each flight template, a lateral path controller is employed to generate the horizontal components for the maneuvers. In order to calculate the control input set $u_{p_{k+1}}$ depending on the desired bearing $\chi_{hk+1} \in x_{wpt}$, the inverse dynamics $u_{p_{k+1}} = f(x_{k+1}, x_k, \Delta t, g_{lim})$ is employed, where $u_p = [\gamma, \mu] \subset U$.

Following section gives 4D Trajectory Planning algorithm which combines local trajectory segments between the sampled states.

6. TACTICAL 4D TRAJECTORY PLANNING: CONFLICT DETECTION AND RESOLUTION

Tactical 4D-trajectory planning is a procedure that spatially explores the airspace for feasible trajectory segments and provides proper separation with persistent conflict check and resolution. Such motion-planning problems strongly depend on the reachability notion due to the dynamic constraints of the aircraft. We suppose that ground systems can predict future trajectories based on computationally complex calculations with approximate performance models and airborne systems are obliged to frequently share information. It is expected that next generation on-board FMSs will exchange parametric uncertainties for each state through available data links.

Let $\chi_j(t)$ in Figure 6.1 denote all possible states that can be reached from a state $x_j(\tau)$ for any $t > \tau$ time without considering obstacles. This estimation of χ_j is derived differently by the ground segment and the on-board FMS due to information availability.

Suppose that *Trajectory Generation Infrastructure* of the ground segment has access to dynamic representations of all types of aircraft, i.e., Aircraft Performance Models (APM), which is generally given in a $\dot{x}_j(t) = f(x_j(t), u(t))$ form, where $x_j(t_0) = x_{j,0}$. Hence, the set of states χ_j that can be reached from a given state $x_j(\tau)$ within the small time interval $t_\varepsilon > 0$ is denoted through the following expression:

$$\chi_j(\tau + t_\varepsilon) = \mathscr{R}_\varepsilon \underset{\tau+t_\varepsilon}{|} (x_j(\tau)) \; for\ any \quad \tau : [0, \infty] \; and\ t_\varepsilon > 0. \tag{6.1}$$

This definition is built on the reachability notion subject to dynamic constraints. A set of reachable states from z (ε-reachable set) is $\mathscr{R}_\varepsilon(z) = \{z' \in Z \mid \exists x \in X_{z,z'}$ such that $x(t) \in \mathscr{B}_\varepsilon(z)\ \forall t \in [0, \tau]\}$ satisfies $\dot{x}_j(t) = f(x_j(t), u(t))$. Let $\mathscr{B}_\varepsilon(z)$ be the closed ball centered at z, which is $\mathscr{B}_\varepsilon(z) = \{z' \in Z \mid ||z' - z|| \leq \varepsilon\}$ and $\varepsilon > 0$. If the given dynamic system is time invariant, then the reachable set grows monotonically, such that:

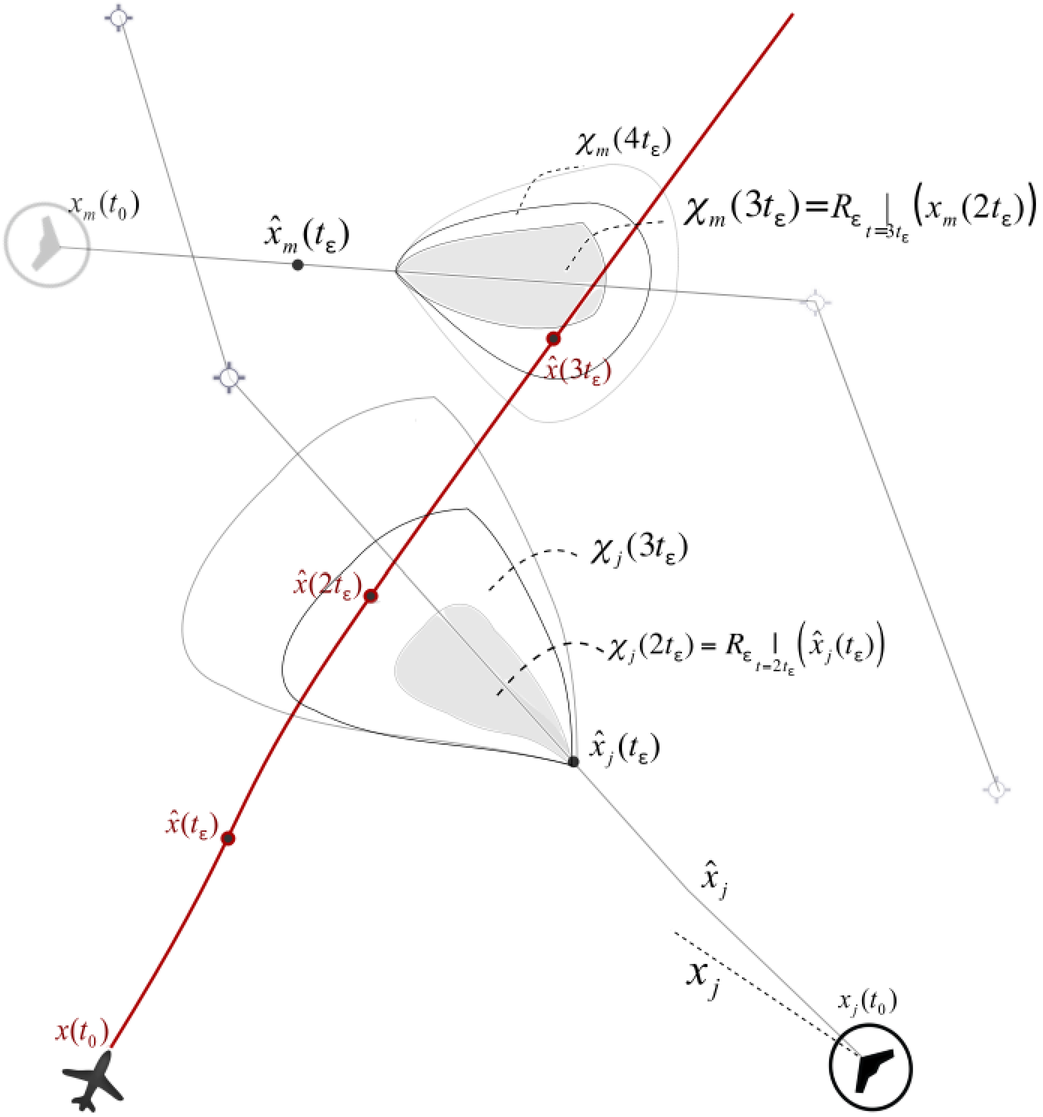

$\chi_m(4t_\varepsilon)$
$\chi_m(3t_\varepsilon)=R_\varepsilon\big|_{t=3t_\varepsilon}(x_m(2t_\varepsilon))$
$x_m(t_0)$
$\hat{x}_m(t_\varepsilon)$
$\hat{x}(3t_\varepsilon)$
$\chi_j(3t_\varepsilon)$
$\hat{x}(2t_\varepsilon)$
$\chi_j(2t_\varepsilon)=R_\varepsilon\big|_{t=2t_\varepsilon}(\hat{x}_j(t_\varepsilon))$
$\hat{x}_j(t_\varepsilon)$
$\hat{x}(t_\varepsilon)$
$\hat{x}_j$
x_j
$x_j(t_0)$
$x(t_0)$

Figure 6.1 : Ground perspective: conflict monitoring with flight intent and reachable sets associated with different performance models.

$$\mathscr{R}_\varepsilon\big|_{t_2}(x_j(t_1)) \subset \mathscr{R}_\varepsilon\big|_{t_3}(x_j(t_1))\ for\ t_1 \leq t_2 \leq t_3. \tag{6.2}$$

The best practice for the estimation of χ_j is to depend on another estimation which will bound the growth. Let $\hat{x}$ and $\hat{x_j}$ be the approximate (nominal) linear trajectories of the ownship and all other aircraft respectively, which is driven from current velocities (seen in Figure 6.1). It is easy to obtain set of states $\hat{x}(\tau)$ and $\hat{x}_j(\tau)$ for any $\tau : [0,\infty]$. Hence, *Conflict Detection* check is done over the path $\hat{x} : [0,T] \rightarrow X$ such that if a portion of the path at τ can be reached by any other aircraft with any admissible control input set $u_j(\tau - t_\varepsilon) \in U_j \subseteq \mathbb{R}^m$. Let $t_\varepsilon > 0$ be a small amount of time such that $u_j(\tau - t_\varepsilon)$ transforms the state $\hat{x}_j(\tau - t_\varepsilon)$ into the set of state $\chi_j(\tau)$ for any $\tau \in [t_\varepsilon, T]$. If such an admissible control input set exists, potential loss of separation may occur. Figure 6.1

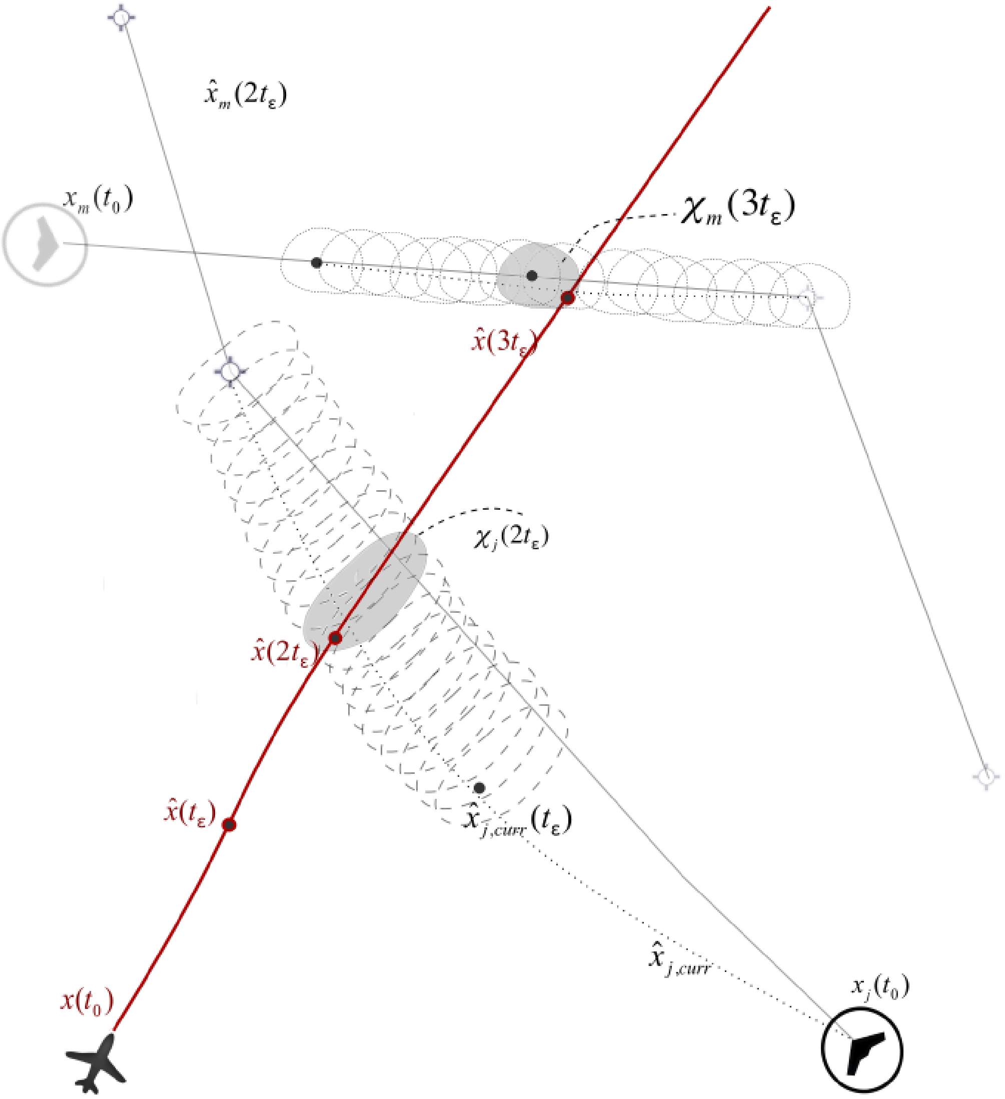


Figure 6.2 : Airborne perspective: conflict monitoring with flight intent exchange and ADS-B.

.

depicts conflict check operation from the ground for non- homogenous airspace that involves many types of aircraft.

Suppose that in highly-heterogeneous airspace, the APMs for the other aircraft are not available to the airborne *Trajectory Generation Infrastructure* of FMS. While the flight intent provides future projection of the traffic, ADS-B/In (direct communications from surrounding aircraft) provides frequent and inherently more precise information about current states of the surrounding aircraft. Suppose that ADS-B/In reception also provides uncertainty parameters for some states, e.g. horizontal position accuracy (NACp), horizontal velocity accuracy (NACv), and vertical accuracy (GVA). Their operational limits are given numerically in [53]. Hence unlike the reachability notion, a possible set of states has a known, unbounded probability distribution that can be represented by a Gaussian distribution with a mean and covariance. Let $\hat{x_j}$ in Figure 6.2 be the linear approximation of the trajectories, depending on the currently available set of states for all surrounding aircraft, and $P_{x,j}(t)$ be the states' time-dependent covariance. That is $\hat{x}_j = \hat{x}_{j,curr}\big|_t$ such that $t \in [0,T]$. Note that, the $\hat{x}_{j,curr}\big|_t$ depends on time, since it is recalculated as new ADS-B/In information becomes available. Thus the potential future set of states χ_j estimation for each aircraft is seen as an error estimation problem by the onboard FMS through following expression:

$$\chi_j(\tau) = \chi_{j,curr}(\tau) \quad for\,any \quad \tau \in [0,T] \tag{6.3}$$

where $\chi_{j,curr}$ are instantaneous Gaussian distribution of the predicted state over the approximate paths. That is;

$$\chi_{j,curr}(\tau) \sim N(\hat{x}_{j,curr}\big|_\tau(\tau), P_{x,j}(\tau)) \tag{6.4}$$

Similar to the previous path approximation, $\hat{x}_{j,curr}\big|_t$ is the linear approximate trajectories (seen in Figure 6.2) depending on current ADS-B information share. A set of states $\hat{x}_{j,curr}(\tau)$ is derived by considering the current velocities for any $\tau : [0,\infty]$. Hence, *Conflict Detection* check is done along the path $\hat{x} : [0,T] \to X$ to query whether a portion of the path at τ, i.e. $\hat{x}(\tau)$, can be reached by any other aircraft with their estimated set of state, i.e. $\chi_{j,curr}(\tau)$. If this overlap exists, potential loss of separation may occur. Figure 6.2 demonstrates typical airborne conflict check implementation with instantaneous ADS-B information availability.

Now we can give a common and generalized definition of loss of separation for both the ground segment and on-board *Trajectory Generation Infrastructure* based on own estimation of the possible future set of states. Thus the following condition indicates the potential loss of separation such that:

$$if \;\; \hat{x}(\alpha t_\varepsilon) \in \bigcup_j^N \chi_j(\beta t_\varepsilon) \cup X_{obs} \quad and \quad \alpha = \beta$$

a potential collision occurs around at time $t = \alpha t_\varepsilon$ and an avoidance action is required before $t = \alpha t_\varepsilon - \delta t_{min\_action}$. Note that $\alpha, \beta \in \mathbb{N} > 1$. The "required response time" term is defined as the minimum time for creating an appropriate response (including comprehending, evaluating, and reacting) to solve the occurring and evolving situation. The following subsection explains the procedure for generating trajectories for avoidance and recovery.

6.1 Sampling-based Conflict Resolution

Collision detection is a persistent process in both ground systems and airborne FMS. Intervention depends on the required action time after detecting a potential collision. The ground systems, with human operators, mostly monitor potential collisions and develop potential solutions by modifying flight plans with minimum deviation. If the required time for action is not enough, immediate action may be needed without the human operator's involvement. It is envisioned that, in such a case, the automated collision avoidance would fully control the aircraft or instruct the pilot.

In order to generate feasible trajectories for a given aircraft model, we chose to implement the *RRT^** algorithm proposed in [19] for its asymptotic optimality in addition to the many other advantages of sampling-based strategies. Asymptotic optimality means that the solution converges on an optimal solution as the number of samples approaches infinity (refer to Figure 6.4). The approach provides an open-loop plan that computes a trajectory projection without considering how future information will alter the future actions. The main structure of the algorithm is given in the following pseudo-code:

Algorithm 1: *RRT^* with Cross-Entropy sampling*

```
$V \leftarrow z_{init}$, $E \leftarrow \emptyset$, $i \leftarrow 0$, $\pi \leftarrow Quantize(FI)$
while $i < N$ do
    $G \leftarrow (V, E)$
    $x_{rand} \leftarrow CE\_Sample(\pi)$
    $(V, E) \leftarrow Extend(G, z_{rand})$
    $\pi \leftarrow Near(G, z_{goal}, |V|)$
    $i \leftarrow i + 1$
```

In this pseudo-code (Algorithm 1), *GE_Sample()* function returns a sample state by utilizing Cross-Entropy sampling which is detailed in the following subsection. A directed graph $G = (V, E)$ is composed of a vertex set V and an edge set E. A directed path on G is a proper sequence $(v_1, v_2, ..., v_n)$ of vertices such that $(v_i, v_{i+1}) \in E$ $\forall i \geq 1$. An elite trajectory set π includes edge sequences that can reach the goal. In

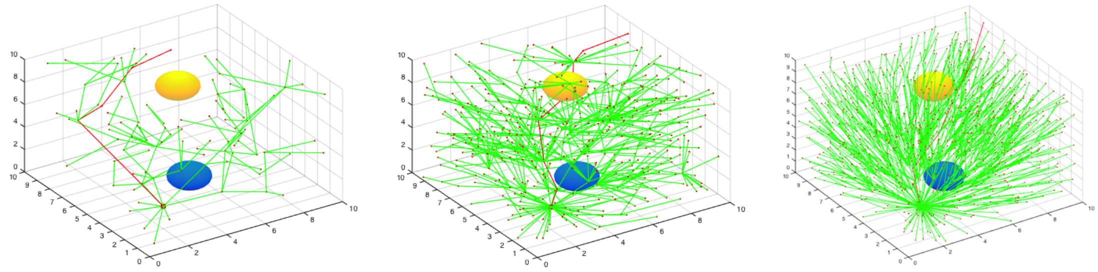


Figure 6.3 : RRT^* algorithm solutions are shown after 100, 600 and 1200 vertices generation respectively.

.

order to immediately build a non-empty set, the elite set n initially involves flight plan trajectory (FI) which is not conflict-free. The initial probability distribution that employs cross-entropy sampling employ is set over the quantized states of the reported flight plan. The *Quantize* function quantizes the flight plan (FI) into a set of states. The idea behind of this initialization is that the minimum cost trajectory is most likely to be near the original flight trajectory. Note that the success of the proposed algorithm does not strongly depend on this assumption where CE sampling disperses over the search space until finding a new elite trajectory by adding variance smoothing. We have slightly modified the RRT^* algorithm algorithm to integrate CE sampling.. *Near* function in Line 6 (Algorithm 1) collects the edge sequences almost reaching to the goal state z_{goal} without considering their costs.

For a given graph $G = (V,E)$ and a point $x \in X_{free}$, the function $Nearest : (G,x) \rightarrow v$ returns a vertex $v \in V$ that is the closest to the x state in terms of distance. We can define in a formal description such that $Nearest(G,x) = argmin_{v \in V} dist(x,v)$.

The *dist* function returns the optimal cost of trajectory between two states without considering obstacles. Hence, $dist(x_1,x_2) = \min_{\tau \in \mathbb{R}_{\geq 0}, u \in U} J(x)$ where $\dot{x}(t) = f(x(t),u(t))$ for all $t \in [0,\tau]$ and $x(0) = x_1$, $x(\tau) = x_2$.

We can also define *Near* function as more generalized form of the *Nearest* function. For a given graph $G = (V,E)$, a point $x \in X_{free}$ and a point $d \in \mathbb{N}$, the function *Near* : $(G,x) \rightarrow V'$ returns a vertex set such that $V' \subseteq V$ and for all vertices $x' \in V'$, satisfies $dist(x',x) \leq l(d)$. The distance threshold $l(d)$ is chosen base on a closed ball of volume $\gamma \log(d)/d$ (refer to [54]) where the γ is an appropriate constant.

Conflict_Free is a Boolean function and returns *true* if a generated trajectory segment $x(t)$ lies in $X_{free}(t)$ for all $t \in [0,\tau]$, otherwise returns *false*. Please recall that $X_{free}(t)$:

Algorithm 2: *Extend()*

```
1  $V' \leftarrow V, E' \leftarrow E$
2  $z_{nearest} \leftarrow Nearest(G, z)$
3  $(x_{new}, u_{new}, \tau_{new}) \leftarrow Generate(z_{nearest}, z)$
4  $z_{new} \leftarrow x_{new}(\tau_{new})$
5  if $Conflict\_Free(x_{new})$ then
6      $V' \leftarrow V' \cup \{z_{new}\}$
7      $z_{min} \leftarrow z_{nearest}$
8      $Z_{near} \leftarrow Near(G, z_{new}, |V|)$
9      forall the $z_{near} \in Z_{near}$ do
10         $(x_{near}, u_{near}, \tau_{near}) \leftarrow Generate(z_{near}, z_{new})$
11         if $Conflict\_Free(x_{near}) \quad and \quad x_{near}(\tau_{near}) = z_{new}$ then
12             $c' \leftarrow J(z_{near}) + J(x_{near})$
13             if $c' < J(z_{new})$ then
14                 $z_{min} \leftarrow z_{near}$
15     $E' \leftarrow E' \cup \{(z_{min}, z_{new})\}$
16     forall the $z_{near} \in Z_{near} \setminus \{z_{min}\}$ do
17         $(x_{near}, u_{near}, \tau_{near}) \leftarrow Generate(z_{new}, z_{near})$
18         if $Conflict\_Free(x_{near}) \quad and \quad J(z_{near}) >$
           $J(z_{new}) + J(x_{near}) \quad and \quad x_{near}(T_{near}) = z_{near}$ then
19             $z_{parent} \leftarrow Parent(z_{near})$
20             $E' \leftarrow E' \setminus \{(z_{parent}, z_{near})\}$
21             $E' \leftarrow E' \cup \{(z_{new}, z_{near})\}$
22 return $G' = (V', E')$
```

$X \setminus X_{obs} \cup X_{sep}(t)$, where $X_{sep}(t)$ denotes set of regions centered at $\chi_j(t)$ (representing trajectories of the surrounding aircraft) for all $t \in [0, \tau]$.

For two states $x_1, x_2 \in X$, $Generate(x_1, x_2)$ function returns a terminal time T, required inputs $\sum_{i=1}^{m} u_i(t) \in U$ and a trajectory segment $x(t) : [0, \tau] \to X$ connecting x_1 and x_2. Note that, for any $\varepsilon > 0$ and for any two states $x_1, x_2 \in X$, *Generate* function satisfies $||x_1 - Generate(x_1, x_2)(t)|| < \varepsilon$ property for all $t \in [0, \tau]$.

Here trajectory planning uses local cost-efficient trajectory segment generation that depends on the Aircraft Performance Model (APM); this is detailed in the previous section. The optimality guarantee of the algorithm strongly depends on holding the "additivity" property, i.e. cost function J satisfies $J(x_1|x_2) = J(x_1) + J(x_2)$.

Generally, the *RRT** algorithm first extends the nearest neighbour (initially x_{init} is the only vertex in the tree) toward the sample (Algorithm 2 – Lines 2-6). However, it generates a path segment to the x_{new} from the vertex within X_{near} set, incurring minimum cost (Algorithm 2 – Lines 7-15). Finally, it extends the new vertex x_{new} toward the vertices in X_{near}, which can be reached through x_{new} with a lower cost (Algorithm 2 – Lines 16-21). For instance, an example run of *RRT** is given in Figure 6.3 to demonstrate its asymptotic convergence as the number of sampling increases through the Halton sequence sampling.

The cost of the minimum trajectory in the *RRT** algorithm converges on a robustly optimal solution J^*, i.e. $\mathbb{P}(\{\lim_{i\to\infty} Y_i = J^*\}) = 1$, where Y_i is the cost of the best trajectory segment after *Extend()* procedure is run under following conditions:

Monotonicity: For two path segments $x_1, x_2 \in \Sigma_{X_{free}}$, let the concatenation of two paths be $x_1|x_2 \in \Sigma_{X_{free}}$, then the cost function satisfies $J(x_1|x_2) \geq J(x_1)$.

Additivity: For all $x_1, x_2 \in \Sigma_{X_{free}}$, the cost function also J satisfies $J(x_1|x_2) = J(x_1) + J(x_2)$.

Continuity: The cost function J satisfies Lipschitz continuity in the following sense: there exists a constant κ such that for any two paths $x_1 : [0,t_1] \to X_{free}$ and $x_2 : [0,t_2] \to X_{free}$, $|J(x_1) - J(x_2)| \leq \sup_{\tau\in[0,1]} ||x_1(\tau t_1) - x_2(\tau t_2)||$.

Local Controllability: Let $\mathscr{B}_\varepsilon(z)$ denote the closed ball centered at z, which is $\mathscr{B}_\varepsilon(z) = \{z' \in Z \quad | \quad ||z' - z|| \leq \varepsilon\}$. Define a set of reachable state from z (ε-reachable set) $\mathscr{R}_\varepsilon(z) = \{z' \in Z \quad | \quad \exists x \in X_{z,z'}$ such that $x(t) \in \mathscr{B}_\varepsilon(z) \quad \forall t \in [0,\tau]\}$

For any state $z \in X$, the set $\mathscr{R}_\varepsilon(z)$ of all states that can be reached from z with a path (ignoring obstacles) that lies entirely inside the ε-ball centered at z, where $\varepsilon \in \mathbb{R}_{\geq 0}$. This assumption guarantees that aircraft dynamics satisfy this weakened version of local controllability.

Conflict-free Trajectories: For an optimal and feasible trajectory $x^* : [0,\tau] \to X_{free}$ and a continuous function $q : \mathbb{R}_{>0}$ with $lim_{\varepsilon\to 0} q(\varepsilon) = x^*$ such that $\varepsilon \geq 0$

(i) x_ε is a ε-collision-free path where $x_\varepsilon(0) = z_{init}$ and $x_\varepsilon(\tau) \in X_{arr}$,

(ii) For $z_1 = x_\varepsilon(t_1)$ and $z_2 = x_\varepsilon(t_2)$ such that $t_1 < t_2$, then the ball of radius $\alpha||z_1 - z_2||^n$

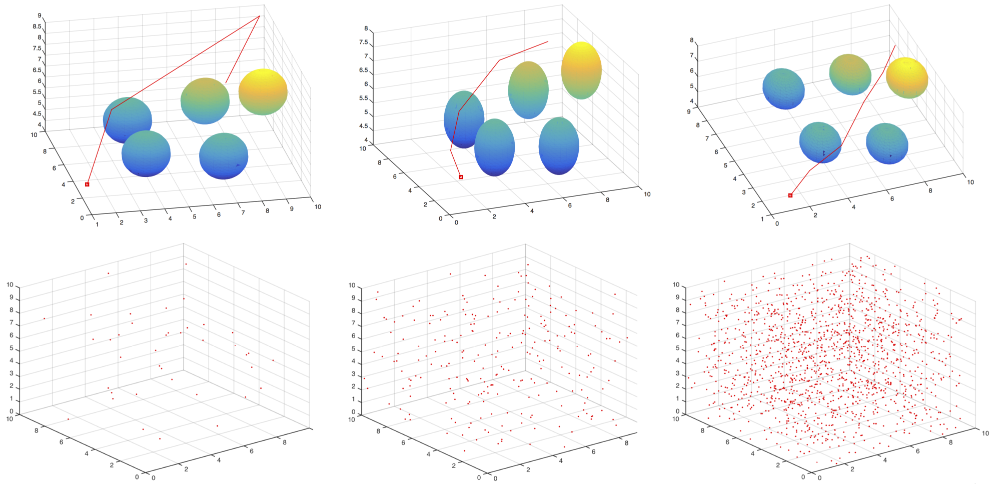


Figure 6.4 : Pseudo-random sampling and asymptotic convergence in RRT^* with 40, 120 and 400 samples

centered at z_2 is ε-reachable from z_1, where $\alpha \in \mathbb{R}_{>0}$ is a constant.

Hence, this assumption guarantees the existence of an optimal trajectory considering differential constraints and spacing between obstacles (including dynamic obstacles such as aircraft).

The result will be a trajectory with their states and effective time intervals which will be sent to FMS to control the aircraft. That is:

$$\pi(t_0 : t_{end}) = \{(x_0, \tau_0), (x_1, \tau_1), \dots, (x_{end}, \tau_{end})\}. \quad \textbf{(6.5)}$$

where $t_0 = \tau$, $\tau_{end} \leq t_{end} - t_\varepsilon$ and surely $t_{current} < t_0 - t_{min\_action}$. The t_{min_action} is the minimum required time to perform a safe action before the first detected collision. This decrement ensures that the solution path begins t_{min_action} in advance.

6.1.1 Importance Sampling with Cross-Entropy

Regarding the sampling strategy, *discrepancy* and *dispersion* are two common criteria to measure uniformity and "quality" of the sampling strategies. Let $Z = [0,1]^n \in \mathbb{R}^n$ be a n-dimensional unit space to generate sample set where $Z = z_0, \dots, z_{N-1}$ denotes a finite set of N n-dimensional points. Let $\mathfrak{B}$ is a nonempty Lebesgue-measurable subset in Z and λ_n denotes n-dimensional Lebesgue measure (or volume), then a general notion of *discrepancy* [55] of the P is given as;

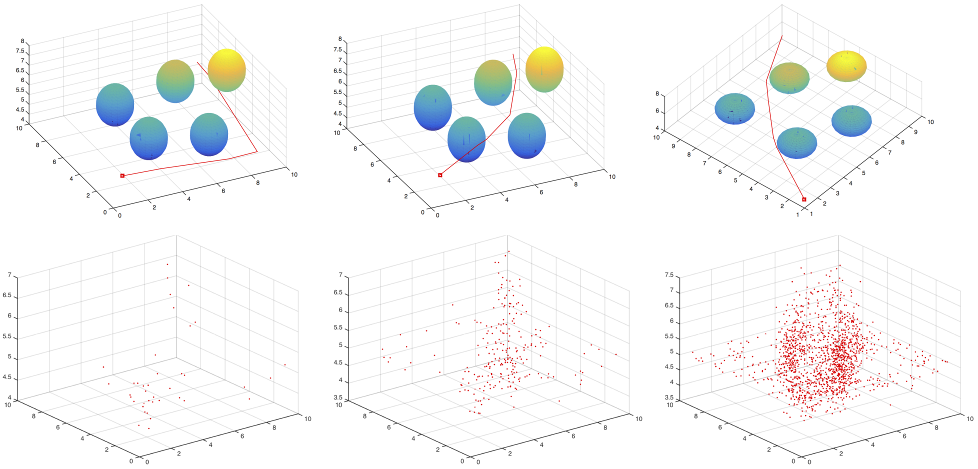


Figure 6.5 : Importance sampling strategy of CE and asymptotic convergence in RRT^* with 40, 120 and 400 samples in RRT^*

$$D_N(P;\mathfrak{B}) = \sup_{B \in \mathfrak{B}} \left| \frac{|P \cap R|}{N} - \lambda_n(B) \right|. \qquad \textbf{(6.6)}$$

The discrepancy notion can be defined as a quantitative measure of irregularity in the sampling distribution, in the other words, it measures the largest volume estimation error [17]. Note that $0 \leq D_N(P;\mathfrak{B}) \leq 1$ such that $\lim_{N \to \infty} D_N(P;\mathfrak{B}) = 0$.

For a given n-dimensional unit space $X = [0,1]^n \in \mathbb{R}^n$ and n-dimensional point set $Z = z_0, \dots, z_{N-1} \in Z$, the *dispersion* notion is defined as;

$$d_N(P;Z) = \sup_{z \in Z} \min_{0 \leq n < N} d(z, z_n), \qquad \textbf{(6.7)}$$

where d denotes any distance metric, e.g. Euclidean distance. Dispersion can easily defined as the radius of the largest ball that does not contain any point of P. For any finite set P of N points following relation between discrepancy and dispersion is given as (see proof in [55];

$$d_N(P;Z) \leq D_N(P;\mathfrak{B}). \qquad \textbf{(6.8)}$$

This relation shows that the low-discrepancy sampling is also a low-dispersion sampling, but the converse is not true (e.g. every dense sequence in Z should be uniformly distributed, which is not true). Dispersion has an obvious relationship with the optimization that bounds the error for motion planning problems. The lower bound for any point set P of N point with Sukharev sampling (point set sampling) in

n dimension is defined as $d_N(P) \leq \frac{1}{2[N^{1/n}]}$ in [55]. Note that this is the best possible point set dispersion, which depends on the dimension and fixed number of points. In motion planning problems, since the amount of sampling is not known in advance, we prefer to use the lowest possible discrepancy, hence the lowest dispersion in selecting a sampling strategy. An example of this is Halton sequence sampling (pseudo-random) within the RRT^* algorithm, which is given in Figure 6.4. The figure demonstrates the convergence on an optimal trajectory with pseudo-random sample distribution of RRT^* with 40, 120 and 400 number of vertices, respectively.

Sampling performance can be further improved by importance sampling, in which the sampling distribution over the state space incrementally concentrates on promising regions. The sampling problem becomes a stochastic optimization problem of finding a proper sample set that leads the algorithm to the minimum cost trajectory. For this purpose, we have integrated the CE method. CE is an adaptive algorithm, first introduced in [56], that estimates probabilities of rare events through variance minimization. CE uses an iterative procedure that first generates a set of samples from a specified distribution and then updates associated parameters. This procedure continues until the distribution of the sample set approaches a delta function. To integrate CE sampling into the planning algorithm, we closely follow [56].

Let $Z \in \mathfrak{Z} \subset \mathbb{R}^n$ be a n-dimensional space to generate sample, and $f(.;v)$ be a probability density defined on the $\mathfrak{Z}$. Consider following estimator;

$$\ell = \mathbb{E}[H(Z)] = \int_z H(z) f(z;v) dz \tag{6.9}$$

where H is a measurable function. The problem was originally to find a trajectory with minimum cost, such that $J(z) \leq \gamma$. Suppose that $\ell \in \mathbb{R}$ is very small real number. Hence, this formalism translates the problem into estimation of rare event probabilities, that is;

$$\ell = \mathbb{P}_v(J(z) \leq \gamma) = \mathbb{E}_v[I_{\{J(z)\leq\gamma\}}] \tag{6.10}$$

where the $I_{\{J(z)\leq\gamma\}}$ is 1 if $J(z) \leq \gamma$, 0 otherwise. The use of Monte-Carlo sampling with low-discrepancy (e.g. Halton sequence) may require a large sampling effort to properly estimate ℓ, since $\{J(z) \leq \gamma\}$ is a small subset of the entire space, that is, a rare event. The alternative approach is to use *importance-sampling*, which generates a sample from a probability density function g defined by $\mathfrak{Z}$. Then estimator $\hat{\ell}$ becomes;

$$\hat{\ell} = \frac{1}{N} \sum_{i=1}^{N} I_{\{J(z) \leq \gamma\}} \frac{f(Z_i; \nu)}{g(Z_i)} \tag{6.11}$$

Let g^* be the optimal density for g, that is;

$$g^*(z) = \frac{I_{\{J(z) \leq \gamma\}} f(z; \nu)}{\ell} \tag{6.12}$$

By applying this density in Eq. 6.11, we get;

$$\ell = I_{\{J(z) \leq \gamma\}} \frac{f(Z_i; \nu)}{g^*(Z_i)}, \quad \forall i \tag{6.13}$$

As seen in Eq. 6.13, g^* depends on ℓ, which is also unknown. By choosing g in probability density $f(.;\nu)$, the problem becomes determining the optimal parameter ν, such that distance between g^* and $f(.;\nu)$ is minimal. The *Kullback-Leibler* (KL) distance between the two densities g and h, which is the CE between g and h, is defined as;

$$\begin{aligned} &\mathfrak{D}(g,h) = \\ &\quad \mathbb{E}_g[ln \frac{g(Z)}{h(Z)}] = \int_z g(z) \ln g(z) dz - \int_z g(z) \ln h(z) dz \end{aligned} \tag{6.14}$$

Minimizing $\mathfrak{D}(g^*, f(.;\nu))$ with respect to ν is equivalent to solve the following problem;

$$\underset{\nu}{\operatorname{argmax}} \int g^*(z) \ln f(z;\nu) dz \tag{6.15}$$

Eventually the optimal importance density parameter ν^* can be evaluated as;

$$\underset{\nu}{\operatorname{argmax}} \, \mathfrak{D}(\nu) = \underset{\nu}{\operatorname{argmax}} \, \mathbb{E}[I_{\{J(z) \leq \gamma\}} \ln f(Z, \nu)] \tag{6.16}$$

A numerical estimation of ν^* can be obtained by solving following stochastic problem;

$$\hat{v}^* = \underset{v}{\operatorname{argmax}} \frac{1}{N} \sum_{i=1}^{N} I_{\{J(z)\leq\gamma\}} \ln f(Z_i, v) \tag{6.17}$$

where $Z_1, Z_2, \ldots, Z_n$ are i.i.d samples from $f(z; v)$.

CE optimization transforms the problem into searching for the minimum cost, i.e. $\gamma^* = \min_{z\in\mathfrak{Z}} J(z)$ associated with trajectories. If γ is very close to γ^*, then $f(z; v^*)$ accumulates its probability mass around to x^* such that it approaches to a delta distribution. In the selection of probabilistic densities, a *Gaussian Mixture Model* (GMM) [57] is preferred due to its ability to form smooth approximations of arbitrarily-shaped sample distributions. For model parameter estimation, the Expectation-minimization (EM) algorithm, which maximizes the likelihood of the GMM, is utilized. The selection of GMM component number k is an arbitrary parameter such that the selection of k depends on the complexity of the environment. In [32], it is empirically shown that more than four components do not improve the solution. Future studies may identify the proper k value.

Algorithm 3: *CE_Sampling*()

```
1  ${\gamma_i}_{i=1}^{|\pi|} \leftarrow GetCost(\pi)$
2  $ind \leftarrow Sort({\gamma_i}_{i=1}^{N})$
3  $\pi_{elite} \leftarrow \pi[ind(\rho : end)]$
4  $Z \leftarrow BackTrack(\pi_{elite})$
5  $f(z; v^*) \leftarrow GMMFit(Z)$
6  $v^* \leftarrow Smoothing(v^*)$
7  repeat
8      $z_{rand} \sim$ Sample $f(z; v^*)$
9  until $Conflict\_Free(z)$
10 return $z_{rand}$
```

In this pseudo code (Algorithm 3), the *CE_Sampling* generates a sample state z_{rand}. *GetCost* function returns the cost of values γ_i for each vertex $z \in \pi \subset V$. Remember that V denotes vertex set in directed graph $G = (V, E)$. *Sort()* function sorts these cost values and its elite set is stored in π_{elite} (Algorithm 3 – Line 3). The elite set involves that the vertices with cost values lower than γ_ρ. In order to build importance sampling regions, *BackTrack* function backtracks parents of the elite set of vertices connecting to the initial state z_{init}. *GMMFit* function uses the EM algorithm to estimate the parameters in Gaussian mixture model for the sample set Z. The *Smoothing* function updates parameters of Gaussian mixture model v_{t-1} to v_t , such that:

$$\hat{v}_t = \alpha \tilde{v}_t + (1-\alpha)\hat{v}_{t-1} \quad \textbf{(6.18)}$$

where $\tilde{v}_t$ is the parameter found in Eq. 6.16. $\alpha \in \mathbb{R}$ is the arbitrary smoothing parameter with $0.7 < \alpha \leq 1$, where $\alpha = 1$ indicates no-smoothing. The sample routine picks a conflict-free sample from the constructed distribution (Algorithm 3 – Line 7-10). Figure 6.5 shows convergence rate of the RRT^* algorithm, this time with CE-based importance sampling. Note that, sampling concentrates around the parametric set of the optimal path as the sample increases.

The convergence rate and the computational effort of the CE sampling in comparison to pseudo-random sampling (i.e. Halton Sequence) are analyzed, and statistical results are plotted. Figures 6.6 and Figure 6.7 show an example with a 10X10X10 box environment, that is, a three-dimensional information space. The results are from experiments repeated 100 times. The standard deviation in convergence to an optimal path as the number of sample increases is displayed with the error bars. As seen in the plot, sampling with CE benefits from importance sampling as it requires a smaller number of vertices to converge on the optimal solution. The average computational effort to generate a certain number of vertices is also analyzed. The effort rate of importance sampling is much less than that of pseudo-random sampling. The CE sampling requires additional computational effort while the number of samples increases, as it sorts the samples and shrinks them into an elite set at each iteration. However, the CE sampling reaches the minimum cost value with less computational effort as the exponentially growing trend begins after reaching the minimum cost value. The black indicators in Figure 6.7 show where CE and pseudo-random sampling methods reach the minimum cost value.

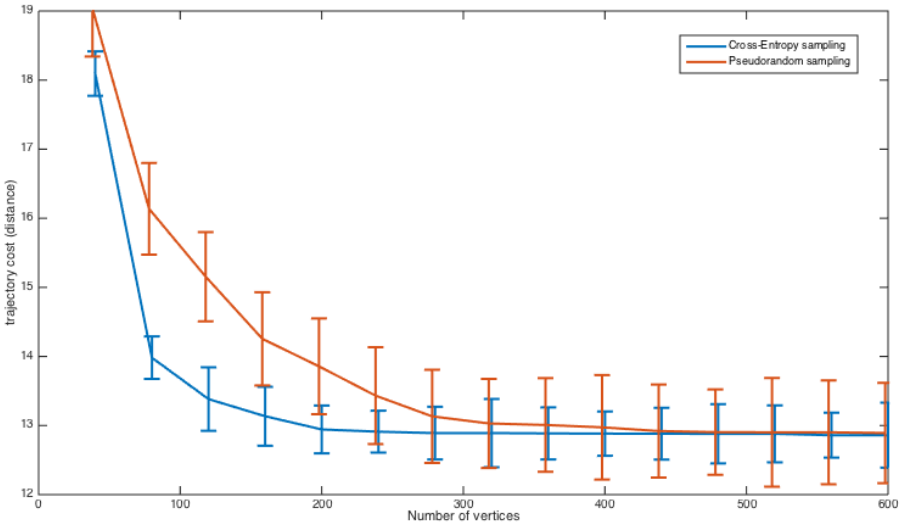


Figure 6.6 : Trajectory cost convergence with the number of vertices in pseudo-random sampling and CE sampling

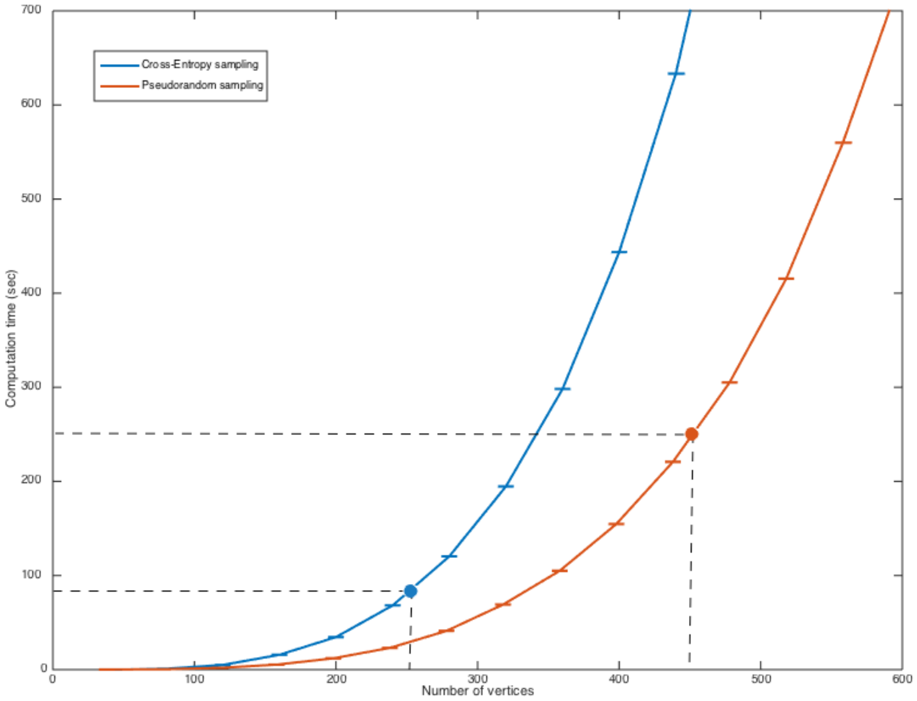


Figure 6.7 : Computational effort with the number of vertices in pseudo-random sampling and CE sampling

7. INTEGRATED SIMULATIONS

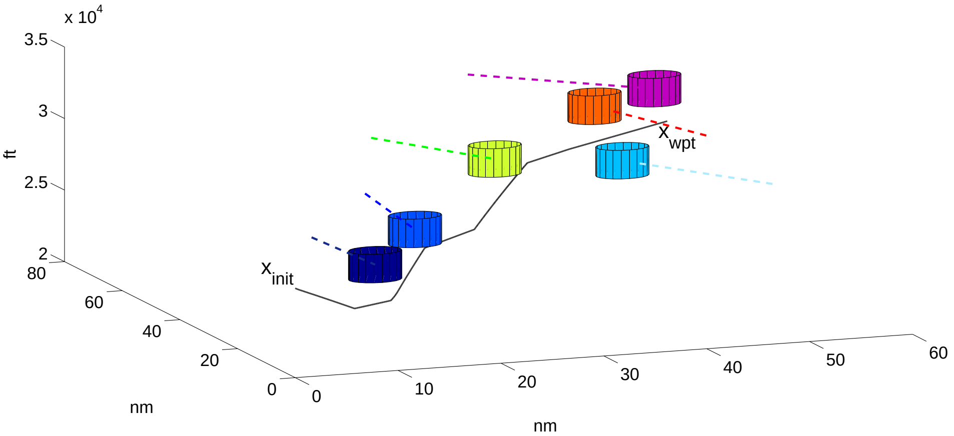


Figure 7.1 : Conflict resolution trajectory for the multiple-thread Scenario 1

This section presents the simulation results of some example scenarios. In these scenarios, we study standard separation problems with multiple aircraft. Trajectory projections up to ten minutes for the surrounding aircraft are determined through the reported states, which are updated at ~ 1 Hz simulation computer. For the sake of simplicity, we have used a point-mass model for the moving intruder aircraft with fixed speed and fixed heading profiles. In these scenarios, a Gaussian distribution for wind is applied with an average wind speed of $7[m/s]$.

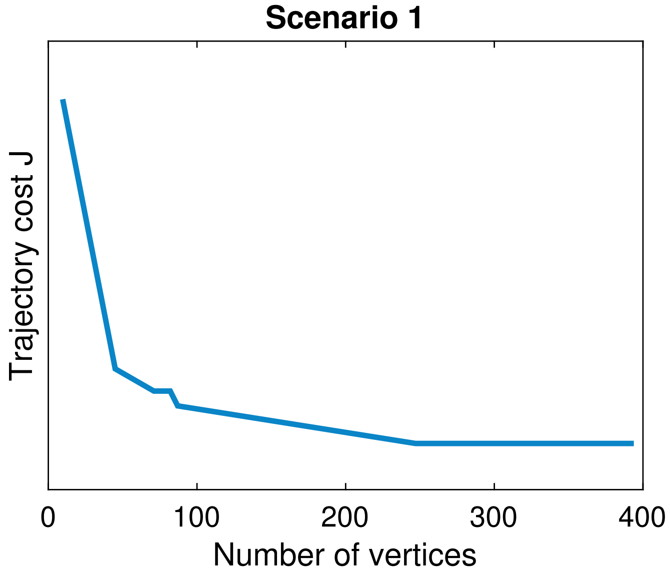


Figure 7.2 : Trajectory cost convergence curve of CR algorithm for the Scenario 1

For the separation minimums, we have applied standard horizontal radar separation in en-route airspace, which is 5 nmi and 2000 vertical ft. These boundaries are shown as a cylinder around intruder aircraft. Note that these cylinders show positions of the intruder aircraft at the last second of the simulations. On the other hand, we have ignored the semicircular (or hemispheric) rule, which applies east–west track split at certain flight levels, since we instead aim to emulate future airspace, which will most likely allow self-separation operation and enable self-regulated flight level to increase capacity.

Table 7.1 : Scenario 1: Flight template and maneuver mode sequence in solution trajectory

Flight Template	Maneuver Sequence	Time Interval[sec]
Descent	Mach Descent	[0, 88.7]
	Level Flight	[88.7, 142.4]
Climb	CAS Climb	[142.4, 212.6]
	Level Flight	[212.6, 323.3]
Climb	CAS Climb	[323.3, 428.4]
	Level Flight	[428.4, 498.7]
Cruise	Level Thrust Acc.	[498.7, 531.8]
	Level Flight	[531.8, 649.2]

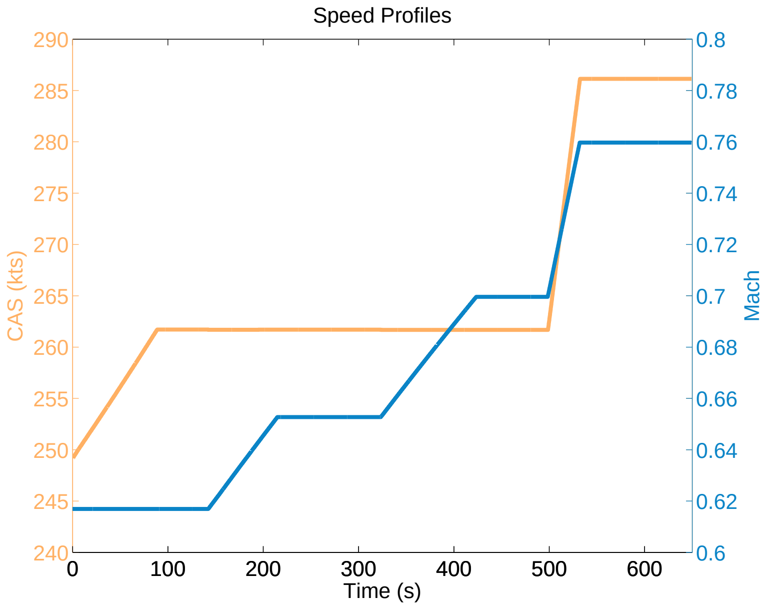


Figure 7.3 : CAS and Mach profiles of conflict resolution trajectory to the Scenario 1

In the first scenario, shown in Figure 7.1, potential conflicts with multiple aircrafts are considered. The conflict resolution (CR) algorithm generates a trajectory tracking descent, climb, and cruise templates as seen in Table 7.1 with their effective time

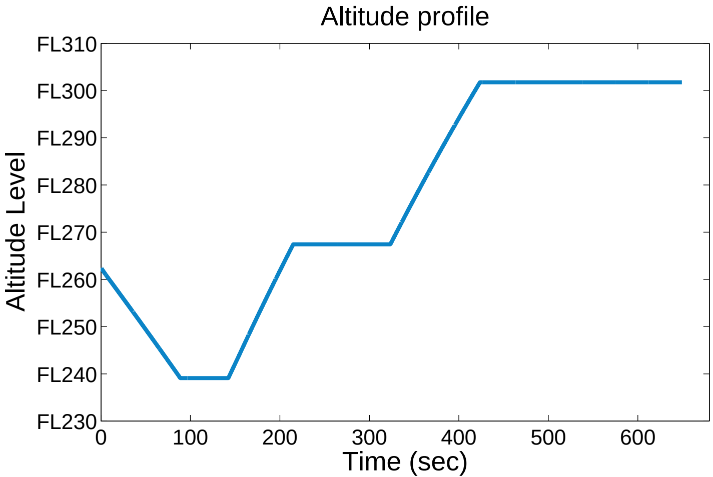


Figure 7.4 : Altitude profile of conflict resolution trajectory to the Scenario 1

intervals. Figure 7.2 shows the cost convergence curve of the cross-entropy based CR algorithm while the number of samples is increasing. Note that, we have chosen to remove the values for the trajectory cost as it originally depends on the real market values of fuel per gallon and time-related costs. Specifically, through the maneuver tree search, the CR algorithm gives a CAS/M Descent first, then climbs to FL300 and accelerates to economic cruise Mach $M_{tgt,crz}$. The maneuver sequence also involves a lateral component. Note that, the cost index is $CI = 50$ for this scenario and the entire maneuver sequence takes approximately 630 seconds. The speed (CAS and Mach) and altitude profiles with their effective time intervals are given in Figures 7.3–7.4.

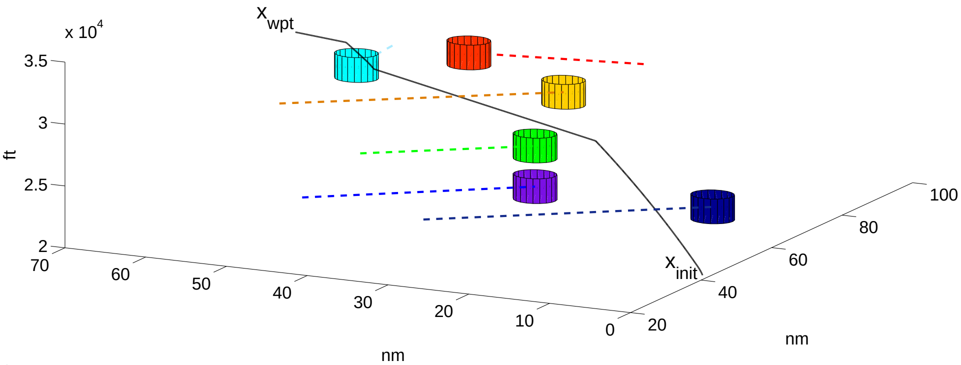


Figure 7.5 : Conflict resolution trajectory for the multiple-thread Scenario 2

In the second scenario, shown in Figure 7.5, similar to Scenario 1, the CR algorithm generates a cascade climb maneuver to resolve potential conflicts. Note that the cost index is 50 for this scenario, and the entire maneuver sequence takes approximately 680 seconds. Cost convergence curve of the CR algorithm while the number of

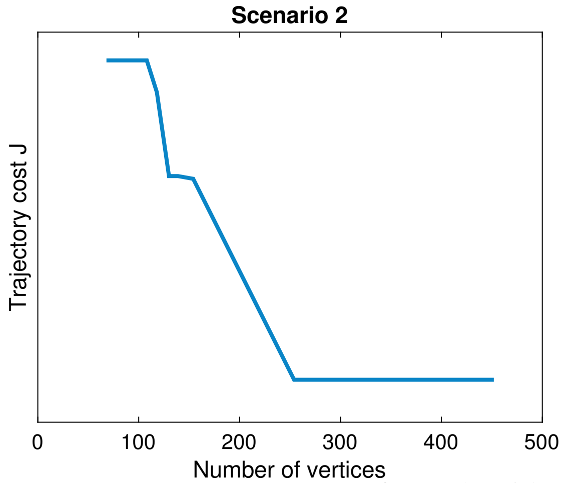


Figure 7.6 : Trajectory cost convergence curve of CR algorithm for the Scenario 2

sampling increasing is demonstrated in Figure 7.6. Table 7.2 gives the generated flight templates and maneuvers over the solution trajectory. At the first stage of the climb, the aircraft increases its Mach number M until it reaches the target altitude h_{wpt} while maintaining the calibrated airspeed V_{CAS}. This means that the optimum climb Mach number $M_{tgt,cmb}$ is the speed at which the aircraft reaches the level-off altitude and the *Hold Mach* maneuver is performed instantly. At the second stage of the climb, the optimum climb Mach number $M_{tgt,cmb}$ is equal to the aircraft's initial Mach number M_{init} at this point. As can be seen in Figure 7.7, the *Hold CAS* maneuver is executed immediately, and the aircraft climbs to the level-off altitude while maintaining its Mach number M. The altitude profile for the second scenario is illustrated in Figure 7.8.

Table 7.2 : Scenario 2: Flight template and maneuver mode sequence in solution trajectory

Flight Template	Maneuver Sequence	Time Interval[sec]
Climb	CAS Climb	[0, 195.6]
	Mach Climb	[195.6, 196.1]
	Level Flight	[196.1, 575.1]
Climb	CAS Climb	[575.1, 611.5]
	Level Flight	[611.5, 678.0]

The third scenario, shown in Figure 7.9, simulates another dense en-route airspace. The cost index is 50 for this scenario, and the entire maneuver sequence takes approximately 570 seconds. The trajectory planning algorithm generates a single *Climb* action with its maneuver sequences as shown in Table 7.3. Cost convergence curve of the CR algorithm while the number of sampling increasing is demonstrated in Figure 7.10. The aircraft starts to climb from FL240 to FL350 with an initial calibrated

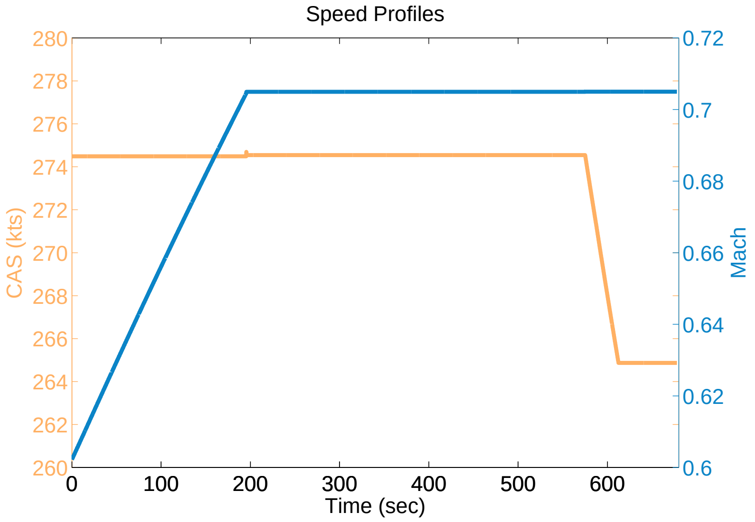


Figure 7.7 : CAS and Mach profiles of conflict resolution trajectory to the Scenario 2

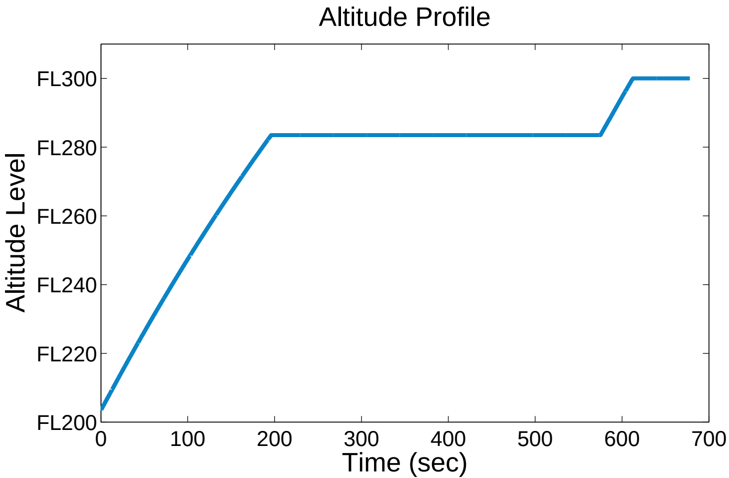


Figure 7.8 : Altitude profile of conflict resolution trajectory to the Scenario 2

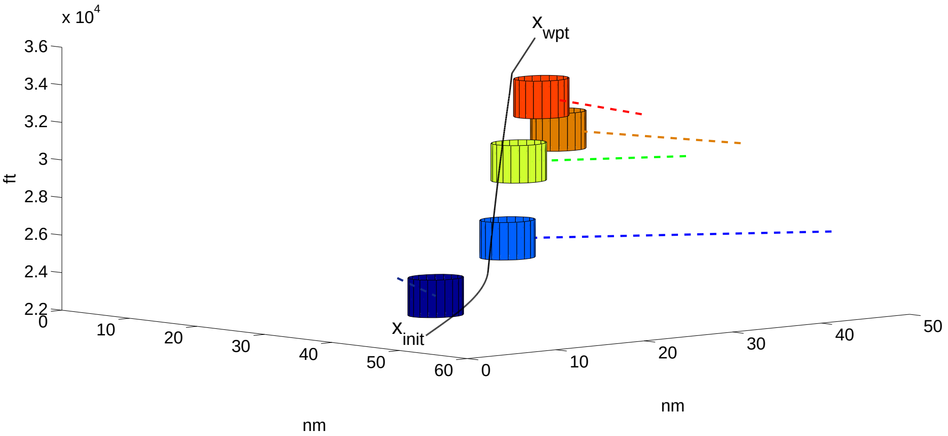


Figure 7.9 : Conflict resolution trajectory for the multiple-thread Scenario 3

airspeed of 263 kts. After approximately 5 minutes, the automaton switches to the *Hold Mach* maneuver and the aircraft starts to climb while maintaining the optimum climb Mach number $M_{tgt,crz} = 0.75$ until it reaches to the target altitude h_{wpt}. The generated speed and altitude profile of the aircraft is given in Figures 7.11–7.12.

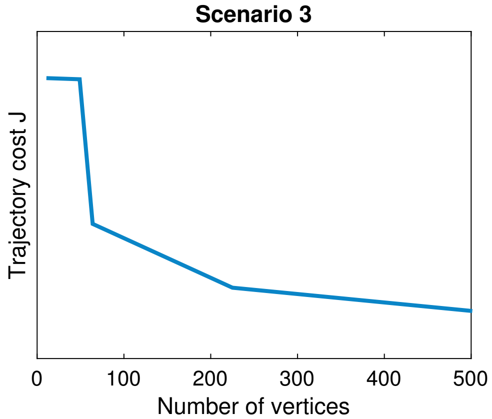


Figure 7.10 : Trajectory cost convergence curve of CR algorithm for the Scenario 3

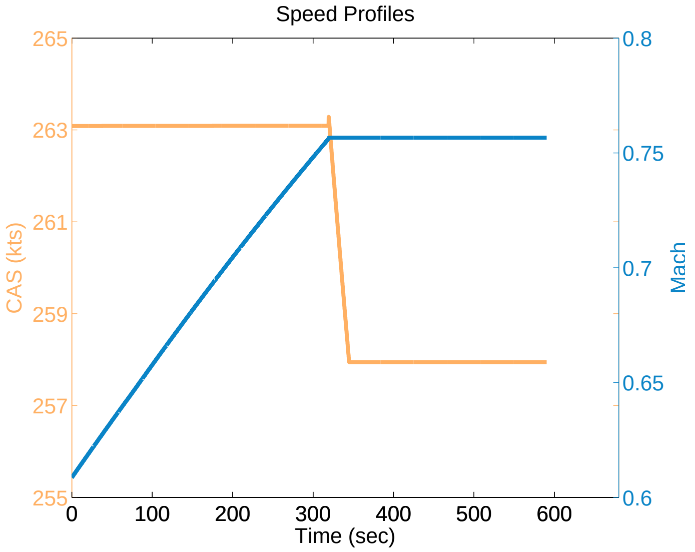


Figure 7.11 : Speed profiles of conflict resolution trajectory to the third scenario

In addition to these scenarios, to demonstrate the real-time applicability of the method, we have run recurrent simulations with the same sampling number (e.g. 200 vertices) and obtained computation times for the different scenarios with different numbers of aircraft. Average computation times were obtained for 100 runs performed with a 2.67 GHz processor and 8 GB RAM. The results are given in Table 7.4. Note that computation efforts show that the proposed method is favorable for real-time applications even for a highly complex multi-aircraft separation.

For operational demonstration, we have integrated proposed CDR algorithm into the developed conceptual decision support tools that are briefly explained in Introduction section. These trajectories might be generated by the ground systems and transferred through the data links, or by the onboard system of the aircraft. A screenshot from 4D Operational Display (4DOD), which is deployed in B737-800 Flightdeck testbed (seen in 1.1) and providing visual understanding for the flight trajectory, is given in

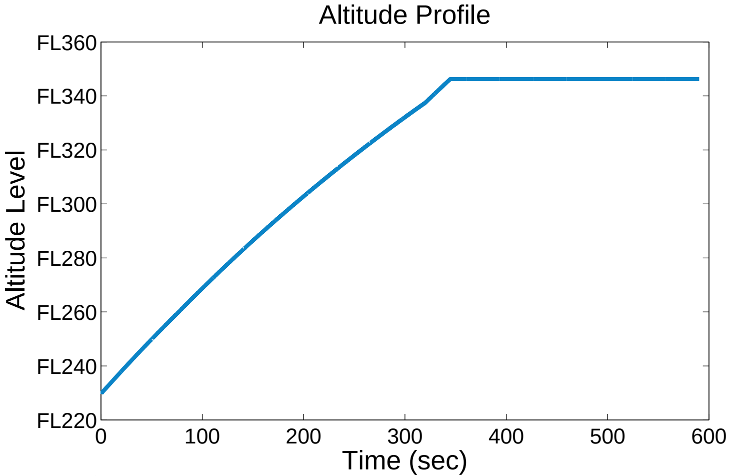


Figure 7.12 : Altitude profile to the third scenario

Table 7.3 : Scenario 3: Flight template and maneuver mode sequence in solution trajectory

Flight Template	Maneuver Sequence	Time Interval[sec]
Climb	CAS Climb	[0, 319.6]
	Mach Climb	[319.6, 344.8]
	Level Flight	[344.8, 506.4]

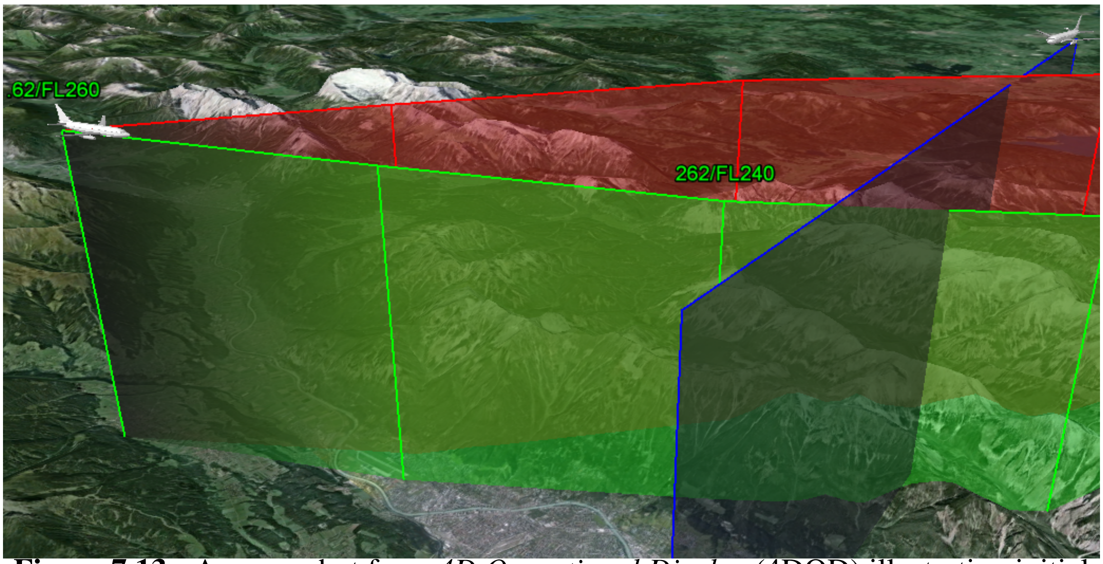


Figure 7.13 : A screenshot from *4D Operational Display* (4DOD) illustrating initial (red), updated (green) and intruder aircraft's (dark blue) trajectories

Table 7.4 : Average CDR computation times for different multiple-thread scenarios

Num. of a/c	1	2	3	5	10	20
time (sec)	3.57	4.07	6.93	8.59	16.18	24.39

Figure 7.13. This screenshot that shows initial descent (in the solution trajectory) is taken while running the simulation for Scenario 1.

8. CONCLUSION

In this project, I have presented a theoretical framework for in-tactical 4D-trajectory generation and shown that an aircraft equipped with automated tools can achieve airborne self-separation. The proposed framework can also be easily extended to strategic planning problems. The proposed structure involves a demonstrably optimal sampling-based algorithm (RRT^*) with an embedded cross-entropy strategy and multi-mode local trajectory optimization. The cost-efficient modal trajectory segment generation utilizes advanced aircraft performance model and relies on BADA 4.

As local planners, we have developed a flight template automaton with cruise, climb and descent templates. Each template employs its own approximate trajectory optimization method. These templates utilized the lower-level maneuver mode automatons, which involve predefined maneuver mode sequences. The parametric model of these maneuver modes uses advanced aircraft performance description based on BADA 4. This local trajectory-generation algorithm provides a control input sequence for a feasible trajectory segment, which is structured to be compatible with the formal language of current flight management systems (FMS). Thus, we have envisioned extending the proposed method to negotiation-based separation assurance through a data link.

As the main contribution of the paper is to adding proper importance-sampling utilizing flight plans and directly mapping the stoctachic planning method into the completely realistic problem, we have compared the proposed method with the standard RRT* algorithm using pseudo-random sampling, which is the closer method (most popular as well) to our fashion. The planning algorithm RRT^* utilizes the local trajectory segment generation and instinctively embeds the stochastic nature of events, which are inherent in air traffic realm, e.g. unpredictable weather conditions and uncertain aircraft movements. It guarantees asymptotic optimality under certain conditions. Its probabilistic search routine rapidly explores the airspace through the local trajectory segment generation, which employs flight template and maneuver mode automatons.

We have replaced the standard random (or quasi-random) sampling generation routine with the CE method in order to benefit from its importance-sampling strategy based on stochastic optimization. We have run our empirical analyses so that CE rapidly converges on the optimal sampling set with a small number of vertices. We have demonstrated that these algorithms can be implemented in realistic multi-threat simulations and also provided average computational efforts for these multi-threat scenarios.

In future research, we are planning to extend the CDR algorithm to an entirely collaborative approach so that aircraft exchange their flight intents to provide enhanced situational awareness. Moreover, we will further improve the maneuver automatons to cover the full flight envelope of the aircraft and consider rare behaviors.

REFERENCES

[1] **NextGen JPDO** (2013). Concept of Operations, **Technical Report**, Joint Planning and Development Office I Version 3.2.

[2] **SESAR JU** (2013). The Concept of Operations at a glance, **Technical Report**, SESAR JU Deliverable 3 (DLM-0612-001-02-00).

[3] **Cook, A.**, **Tanner, G.**, **Williams, V. and Meise, G.** (2009). Dynamic cost indexing &ndash; Managing airline delay costs, *Journal of Air Transport Management*, *15*(1), 26–35.

[4] **Jensen, L. and Hansman, R.J.** (2014). Fuel Efficiency Benefits and Implementation Consideration for Cruise Altitude and Speed Optimization in the National Airspace System, **Technical Report**, MIT International Center for Air Transportation.

[5] **López-Leonés, J.**, **Vilaplana, M.A.**, **Gallo, E.**, **Navarro, F.A. and Querejeta, C.** (2007). The aircraft intent description language: a key enabler for air-ground synchronization in trajectory-based operations, *Digital Avionics Systems Conference, 2007. DASC '07. IEEE/AIAA 26th*, pp.1. D. 4–1–1. D. 4–12.

[6] **Kuchar, J.** (2000). A review of conflict detection and resolution modeling methods, *IEEE Transactions on Intelligent Transportation Systems*.

[7] **Christodoulou, M.A. and Kodaxakis, S.G.** (2006). Automatic commercial aircraft-collision avoidance in free flight: the three-dimensional problem, *Intelligent Transportation Systems, IEEE Transactions on*, *7*(2), 242–249.

[8] **Richards, A. and How, J.P.** (2002). Aircraft trajectory planning with collision avoidance using mixed integer linear programming, *American Control Conference, 2002. Proceedings of the 2002*, pp.1936–1941 vol.3.

[9] **Koyuncu, E. and Inalhan, G.** (2013). Exploiting Delayed and Imperfect Information for Generating Approximate UAV Target Interception Strategy, *Journal of Intelligent and Robotic Systems*, *69*(1-4), 313–329.

[10] **Kochenderfer, M.J. and Chryssanthacopoulos, J.P.** (2011). Robust Airborne Collision Avoidance Through Dynamic Programming, *Massachusetts Institute of Technology, Lincoln Laboratory, Project Report ATC-371*.

[11] **Kochenderfer, M.J. and Chryssanthacopoulos, J.P.** (2011). Partially-controlled Markov decision processes for collision avoidance systems, *International Conference on Agents and Artificial Intelligence*, Rome, Italy.

[12] **Winder, L.F.** (2004). Hazard Avoidance Alerting with Markov Decision Processes, *Ph.D. thesis*, Massachusetts Institute of Technology.

[13] **Wolf, T. and Kochenderfer, M.** (2011). Aircraft Collision Avoidance Using Monte Carlo Real-Time Belief Space Search, *Journal of Intelligent and Robotics Systems*, *64*, 277–298.

[14] **Chryssanthacopoulos, J.P.** (2011). Accounting for state uncertainty in collision avoidance, *Journal of Guidance*.

[15] **Valenzuela, A. and Rivas, D.** (2011). Conflict Resolution in Converging Air Traffic Using Trajectory Patterns, *Journal of Guidance, Control, and Dynamics*, *34*(4), 1172–1189, `http://dx.doi.org/10.2514/1.50751`.

[16] **Kuffner, J.J. and LaValle, S.M.** (2000). RRT-Connect: An Efficient Approach to Single-Query Path Planning, *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pp.995–1001.

[17] **LaValle, S.M.** (2004). On the Relationship between Classical Grid Search and Probabilistic Roadmaps, *The International Journal of Robotics Research*, *23*(7-8), 673–692.

[18] **Lindeman, S. and LaValle, S.M.**, (2005). Current Issues in Sampling-Based Motion Planning, Springer Berlin Heidelberg, p. 10.

[19] **Frazzoli, E. and Karaman, S.** (2010). *Incremental sampling-based algorithms for optimal motion planning*, Int. Journal of Robotics Research.

[20] **LaValle, S. and Kuffner, J.** (1999). Randomized kinodynamic planning, *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, *1*, 473–479 vol.1.

[21] **Karaman, S. and Frazzoli, E.** (2010). Optimal kinodynamic motion planning using incremental sampling-based methods, *Decision and Control (CDC), 2010 49th IEEE Conference on*, IEEE, pp.7681–7687.

[22] **Luders, B.D.**, **Karaman, S. and How, J.P.** (2013). Robust sampling-based motion planning with asymptotic optimality guarantees, *AIAA Guidance*.

[23] **Lindemann, S.R. and LaValle, S.M.** (2004). Steps toward derandomizing RRTs, *Proceedings of the Fourth International Workshop on Robot Motion and Control, RoMoCo'04*.

[24] **Karaman, S.**, **Walter, M.R.**, **Perez, A.**, **Frazzoli, E. and Teller, S.** (2011). Anytime Motion Planning using the RRT*, *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, IEEE, pp.1478–1483.

[25] **Şucan, I. and Kavraki, L.E.** (2000). A Sampling-Based Tree Planner for Systems With Complex Dynamics, *IEEE Transactions on Robotics*, *28*(1), 116–131.

[26] **Kurniawati, H. and Hsu, D.** (2004). Workspace importance sampling for probabilistic roadmap planning, *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 2, IEEE, pp.1618–1623.

[27] **Burns, B. and Brock, O.** (2005). Toward Optimal Configuration Space Sampling., *Robotics: Science and Systems*, Citeseer, pp.105–112.

[28] **Hsu, D.**, **Sánchez-Ante, G. and Sun, Z.** (2005). Hybrid PRM sampling with a cost-sensitive adaptive strategy, *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, IEEE, pp.3874–3880.

[29] **Kalisiak, M. and van de Panne, M.** (2007). Faster motion planning using learned local viability models, *Robotics and Automation, 2007 IEEE International Conference on*, IEEE, pp.2700–2705.

[30] **Li, Y. and Bekris, K.E.** (2010). Balancing state-space coverage in planning with dynamics, *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, IEEE, pp.3246–3253.

[31] **Denny, J. and Amato, N.M.** (2011). Toggle PRM: Simultaneous Mapping of C-free and C-obstacle-A Study in 2D, *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, IEEE, pp.2632–2639.

[32] **Kobilarov, M.** (2012). Cross-entropy motion planning, *The International Journal of Robotics Research*, *31*(7), 855–871.

[33] **Rubinstein, R.Y. and Kroese, D.P.** (2004). *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*, Springer Science & Business Media.

[34] **Brockett, R.W.** (1990). Languages for motion description and map making, *Proc. of Symposia in Applied Mathematics*, *14*, 181–293.

[35] **Ure, N.K. and Inalhan, G.** (2012). Autonomous Control of Unmanned Combat Air Vehicles: Design of a Multimodal Control and Flight Planning Framework for Agile Maneuvering, *Control Systems, IEEE*, *32*(5), 74–95.

[36] **Fainekos, G.**, **Gazit, H.K. and Pappas, G.J.** (2005). Hybrid controllers for path planning : a temporal logic approach, *in IEEE Conference on Decision and Control*.

[37] **Tomlin, C.**, **Pappas, G.J. and Sastry, S.** (1998). Conflict Resolution for Air Traffic Management: A Study in Multi-Agent Hybrid Systems, *IEEE Transactions on Automatic Control*, *43*.

[38] **Frazzoli, E.**, **Dahleh, M.A. and Feron, E.** (2005). Maneuver-Based Motion Planning for Nonlinear Systems With Symmetries, *IEEE Transactions on Robotics*, *21*(6), 1077–1091.

[39] **Dever, C**, **Mettler, B**, **Feron, E**, **Popovic, J and McConley, M** (2006). Nonlinear Trajectory Generation for Autonomous Vehicles via Parameterized Maneuver Classes, *Journal of Guidance Control and Dynamics*, *29*, 289–302.

[40] **Koyuncu, E.**, **Ure, N.K. and Inalhan, G.** (2010). Integration of Path/Maneuver Planning in Complex Environments for Agile Maneuvering UCAVs, *JOURNAL OF INTELLIGENT & ROBOTIC SYSTEMS*, *57*(1-4), 143–170.

[41] **Koo, T.J.**, **Pappas, G.J. and Sastry, S.** (2001). Multi-Modal Control of Systems with Constraints, *40th IEEE Conference Decision and Control*, Orlando, Florida, USA, pp.2075–2081.

[42] **Endsley, M.R.** (1995). Measurement Of Situation Awareness In Dynamic Systems, *Human Factors*, *37*(1), 65–84.

[43] **Parasuraman, R. and Wickens, C.D.** (2008). Humans: Still Vital After All These Years of Automation., *Human Factors*, *50*(3), 511–520.

[44] (2003).

[45] **van Marwijk, B.J.A.**, **Borst, C.**, **Mulder, M.**, **Mulder, M. and van Paassen, M.M.** (2011). Supporting 4D Trajectory Revisions on the Flight Deck: Design of a Human–Machine Interface, *The International Journal of Aviation Psychology*, *21*(1), 35–61.

[46] **Ladkin, P.** (2004). AA965 Cali Accident Report, Near Buga, Colombia, Dec 20, 1995, *Peter Ladkin Universität Bielefeld Technical Report*, Napa, California.

[47] **LaValle, S.M.** (2006). Planning Algorithms, *Book*, 1023.

[48] **Nuic, A.**, **Poinsot, C.**, **Iagaru, M.**, **Gallo, E.**, **Navarro, F.A. and Querejeta, C.** (2005). Advanced Aircraft Performance Modeling for ATM: Enhancements to the Bada Model, *Digital Avionics Systems Conference, 2005. DASC 2005. The 24th*, IEEE, pp.2–2.B.4–1.

[49] **EUROCONTROL** (2014). User Manual for the Base of Aircraft Data (BADA) Family 4, *EEC TechnicalScientific Report No. -*.

[50] **EUROCONTROL** (2013). Principles For Establishing The Cost-Base For En Route Charges And The Calculation Of The Unit Rates, *European Organisation for the Safety of Air Navigations*, 1–54.

[51] **Altus, S.** (2009). Effective Flight plans can Help Airlines Economize, *Boeing Aero*.

[52] **Roberson, B. and Pilot, S.S.** (2007). Fuel Conservation Strategies: cost index explained, *Boeing Aero Quarterly*.

[53] **SC-186, RTCA** (2004). Minimum Operational Performance Standards for Universal Access Transceiver (UAT) Automatic Dependent Surveillance Broadcast (ADS-B), **Technical Report**, RTCA.

[54] **Karaman, S. and Frazzoli, E.**, (2011). Incremental Sampling-Based Algorithms for a Class of Pursuit-Evasion Games, Algorithmic Foundations of Robotics IX, Springer Berlin Heidelberg, Berlin, Heidelberg, pp.71–87.

[55] **Niederreiter, H.** (1992). *Random Number Generation and Quasi-Monte Carlo Methods*, SIAM.

[56] **Rubinstein, R.Y. and Kroese, D.P.** (2004). *The Cross Entropy Method: A Unified Approach To Combinatorial Optimization, Monte-carlo Simulation (Information Science and Statistics)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[57] **Reynolds, D.**, (2009). Gaussian mixture models, Encyclopedia of Biometrics, Springer, pp.659–663.

APPENDICES

CURRICULUM VITAE

Name Surname: Mevlüt Uzun

Place and Date of Birth: 08.03.1991

Adress: Istanbul Teknik Üniversitesi, Uçak ve Uzay Bilimleri Fakültesi, 312-A, Maslak - İstanbul

E-Mail: uzunm@itu.edu.tr

B.Sc.: Aeronautical Engineering, 2013

B.Sc.: Control and Automation Engineering, 2015

M.Sc.: Aeronautical and Astronautical Engineering

PUBLICATIONS/PRESENTATIONS ON THE THESIS

- **Uzun M.**, Guner G., Koyuncu E. and Inalhan G., Integrated Flight Deck Testbed with Next Generation Visual Decision Support Tools, *6th International Conference on Research in Air Transportation*, May 2014
- **Uzun M.**, Guner G., Koyuncu E. and Inalhan G., Flight Deck Centered 4D Trajectory Planning and Collision Avoidance with Flight Envelope Sampling, *International Conference on Application and Theory of Automation in Command and Control Systems*, September 2015
- Koyuncu E., **Uzun M.** and Inalhan G., Cross-Entropy Based 4D Trajectory Generation for Airborne Conflict Resolution, Journal of Aerospace Engineering, (*Accepted*)
- Panel Participant, "Flight Deck Centered Tactical Trajectory Planning", *BADA User Conference*, Boeing East Headquarters, Washington DC, September 2015