ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

AUTOFLY-AID: FLIGHT DECK AUTOMATION SUPPORT WITH DYNAMIC 4D TRAJECTORY MANAGEMENT FOR RESPONSIVE AND ADAPTIVE AIRBORNE COLLISION AVOIDANCE

Ph.D. THESIS

Emre KOYUNCU

Department of Aeronautics and Astronautics Engineering

Aeronautics and Astronautics Program

MAY 2015

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

AUTOFLY-AID: FLIGHT DECK AUTOMATION SUPPORT WITH DYNAMIC 4D TRAJECTORY MANAGEMENT FOR RESPONSIVE AND ADAPTIVE AIRBORNE COLLISION AVOIDANCE

Ph.D. THESIS

Emre KOYUNCU
(511102102)

Department of Aeronautics and Astronautics Engineering

Aeronautics and Astronautics Program

Thesis Advisor: Doç. Dr. Gökhan İNALHAN

MAY 2015

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

AUTOFLY-AID: HAVADA ÇARPIŞMADAN KAÇINMA İÇİN ESNEK VE UYARLAMALI 4 BOYUTLU DİNAMİK ROTA YÖNETİMI İLE UÇUŞ KARAR DESTEK SİSTEMİ

DOKTORA TEZİ

Emre KOYUNCU
(511102102)

Uçak ve Uzay Mühendisliği Anabilim Dalı

Uçak ve Uzay Mühendisliği Programı

Tez Danışmanı: Doç. Dr. Gökhan İNALHAN

MAYIS 2015

Emre KOYUNCU, a Ph.D. student of ITU Graduate School of Science Engineering and Technology 511102102 successfully defended the thesis entitled **"AUTOFLY-AID: FLIGHT DECK AUTOMATION SUPPORT WITH DYNAMIC 4D TRAJECTORY MANAGEMENT FOR RESPONSIVE AND ADAPTIVE AIRBORNE COLLISION AVOIDANCE"**, which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Doç. Dr. Gökhan İNALHAN** Istanbul Technical University	.....
Jury Members :	**Prof. Dr. İbrahim ÖZKOL** Istanbul Technical University	.....
	Prof. Dr. Cengiz HACIZADE Istanbul Technical University	.....
	Prof. Dr. Aydan CAVCAR Anadolu University	.....
	Yrd. Doç. Dr. Sertaç KARAMAN Massachusetts Institute of Technology	.....

Date of Submission : **6 April 2015**
Date of Defense : **11 May 2015**

To my family who bought me a Commodore 64
instead of buying a car for themselves...

FOREWORD

First and I would like to express my gratitude to my thesis advisor and "master" Prof. Gokhan Inalhan. It is impossible not to be truly impressed by his vision, creativity, and enthusiasm. He has made many sacrifices so that he could go far beyond his obligations, in order to make us better researcher and ensure our success. He has been an exemplary researcher to me, and he has been very patiently and diligently advising and guiding me over all those years (indeed a lot). My personality has greatly improved by his outstanding advisory and leadership.

This research was sponsored in part by SESAR WP-E HALA! Research Network. I would like to thank Dr. Eduardo Garcia for his advises, tremendous support and guidance during this thesis. Thanks to offer me the opportunity to initiate fruitful collaboration with Boeing Research and Technology Europe. Thanks to all the ATM guys in Boeing R&TE, Dr. Miguel Vilaplana, Dr. Javier Lopez, Luis Alto and (to be Dr.) Enrique Casado for the great moments we have spent together in the office and outside the office. Luis, there are not enough words to express my sincere appreciation for your never-ending support while compiling, building and debugging my codes again and again... Javi, you will easily notice the pieces of the thesis from our fruitful discussions. I would like to extend my gratitude to all the Boeing R&TE for all the lessons I have learnt and the ones I will learn in the future from you.

I am thankful also to Prof. Sertac Karaman for his advices, guidance, help and struggle with this thesis, and more important, his friendship. Thanks to offer me the opportunity to visit MIT and provide the great ideas that makes "complete" this research. I know that our collaboration and "gourmet adventures" will continue...

There are not enough words to express my sincere gratitude to Prof. Ibrahim Ozkol for his infinite support and all the lessons I have learnt from him. He has been a role model for me in dedication and being focused even under a heavy burden.

I would like to apologise to all my friends, for all the time, I had to invest in working (sometimes useless) instead of sharing it with them. Thanks for your patient, understanding and the most important thing, your friendship during this long and hard path. Your patient will be far rewarded, be aware of it!

Finally, I cannot thank my family enough for hanging in there with me throughout the years that I have spent far away from them. Without their never-ending powerful spiritual support, it would have been impossible to end this story. Every time, hearing supportive voice of my father, concerned voice of my mother and lovely voice of my sister has made me strong much more, and I can never repay them for the love and care that they have shown me.

As a last words, I want to dedicate this story to my “master”, namely supervisor:

In a forest a fox bumps into a little rabbit, and says, "Hi, junior, what are you up to?"
"I’m writing a dissertation on how rabbits eat foxes", said the rabbit.
"Come now, friend rabbit, you know that’s impossible!".
"Well, follow me and I’ll show you."

They both go into the rabbit's dwelling, and after a while the rabbit emerges with a satisfied expression on his face.

Comes along a wolf. "Hello, what are we doing these days?".
"I'm writing the second chapter of my thesis, on how rabbits devour wolves".
"Are you crazy? Where is your academic honesty?".
"Come with me and I'll show you". As before, the rabbit comes out with a satisfied look on his face and a diploma in his paw.

Finally, the camera pans into the rabbit's cave and, as everybody should have guessed by now, we see a mean-looking, huge lion sitting next to some bloody and furry remnants of the wolf and the fox. The moral: It's not the contents of your thesis that are important – it's your PhD advisor that really counts."

Mayıs 2015 Emre KOYUNCU

TABLE OF CONTENTS

Page

ABBREVIATIONS

4DT	**:** Four Dimensional Trajectory (3D Path+Time)
ACAS	**:** Airborne Collision Avoidance System
ADS-B	**:** Automatic Dependent Surveillance Broadcast
AIDL	**:** Aircraft Intent Description Language
ANSP	**:** Air Navigation Service Provider
AR	**:** Augmented Reality
BADA	**:** Base of Aircraft Data, EUROCONTROL
BRTE	**:** Boeing Technology and Research Europe
CNS	**:** Communication, Navigation and Surveillance
FIDL	**:** Flight Intent Description Language
FMS	**:** Flight Management System
FNPT	**:** Flight and Navigation Procedures Trainer
HALA!	**:** Higher Level Automation in ATM Research Network
HUD	**:** Head-up-Display
IFR	**:** Instrument Flight Routes
ITU-CAL	**:** Istanbul Technical University, Controls and Avionics Laboratories
NextGen	**:** Next Generation Air Transportation System
NMAC	**:** Near and Mid Air Collision
RTA	**:** Required time of Arrival
SBAS	**:** Space Based Augmentation System
SESAR	**:** Single European Sky ATM Research
SID	**:** Standard Instrument Departure
STAR	**:** Standard Terminal Arrival Routes
SWIM	**:** System Wide Information Management
TBO	**:** Trajectory Based Operation
TCAS	**:** Traffic Collision Avoidance System

LIST OF TABLES

Page

LIST OF FIGURES

Page

AUTOFLY-AID: FLIGHT DECK AUTOMATION SUPPORT WITH DYNAMIC 4D TRAJECTORY MANAGEMENT FOR RESPONSIVE AND ADAPTIVE AIRBORNE COLLISION AVOIDANCE

SUMMARY

This thesis, namely, AUTOFLY-Aid Project, aims to develop and demonstrate novel flight deck automation support algorithms and tools for potential conflict avoidance and performance-optimal flight using "dynamic 4D trajectory management". The developed automation support system is envisioned not only to improve the primary shortcomings of existing on-board traffic collision avoidance systems (e.g. TCAS), but also to develop new conceptual add-on avionics and procedures enabling intent data exchange, decision support systems with augmented reality and flight control hand-over implementation in dynamically evolving scenarios. The main concepts which has been developed in AUTOFLY-Aid project are a) design and development of the mathematical models of the full composite airspace picture from the flight deck perspective, as seen/measured/informed by the aircraft flying in the sky of the SESAR and NextGen 2020+ vision and beyond, b) design and development of a dynamic 4D trajectory planning algorithm can generate at real-time flyable (i.e. dynamically and performance-wise feasible) alternative trajectories for both short-term and mid-term scale across the evolving stochastic composite airspace picture and c) development and testing of the automation support system on a Boeing 737-800 Flight Simulator with conceptual procedures, automated flight control implementations, and reality augmented based decision support demonstrations providing the flight crew with quantified and visual understanding of evolving situation.

Evaluation from a purely centralized tactical intervention model towards a more strategic planning and progressive introduction of more autonomous and decentralized tactical operation with more proactive systems are key concepts in both NextGen and SESAR future ATM paradigm shift vision. Implementing of these new-generation ATM concepts will significantly change the human role in the ATM system by considering "best decision place", "best decision time" and "the best decision player". Through these objectives, AUTOFLY-Aid envisions to take some of the work off the controller by delegating some responsibility to flight decks in an efficient manner. The developed automation system offers persistent in-flight hazard and flight efficiency monitoring and tactical flight trajectory planning as a function of look-ahead time and dynamically changing environmental/operational conditions (and with uncertainty reduction in a feedback loop) obtained via both in-flight sense and ground-air data link. The automation system switches autonomy level according to the required response time in order to find "the best decision player" through asking "where are men better at, where are machines better at". In mid-term safety assurance mode, it is expected that pilot uses a visual decision support tools (e.g. tunnel-in-the-sky visualization) with fully situational awareness for safe and performance optimal flight. These visual advisories are generated by fusing all tactical level information feed from both on-board sensing and ground-air data/information exchange. If the

reaction time permits, the system allows pilots to freely switch between the generated alternative plans, modify the solution or request re-planning. In any case of the immediate potential threat is detected (i.e. immediate response is required or late response is detected), the autonomous system may take over the flight control to solve safety-critical situation happening "almost surely" (e.g. midair collision, terrain collision etc.). This hybrid approach allows dynamic role assignment by switching between defined autonomy level modes in terms of the "required response time".

AUTOFLY-AID: HAVADA ÇARPIŞMADAN KAÇINMA İÇİN ESNEK VE UYARLAMALI 4 BOYUTLU DİNAMİK ROTA YÖNETİMI İLE UÇUŞ KARAR DESTEK SİSTEMİ

ÖZET

Günümüz standartlarında pilot ve kule arasındaki iletişim sesli olarak radyo aracılığı ile sağlanmakta, ancak bu operasyon şekli, artan hava trafiğini kontrol etmekte, olağanüstü durumları verimli bir şekilde kontrol etme konusunda yetersiz kalmaktadır. Bu yüzden geliştirilmekte yeni bilgi paylaşımı sistemleri ile (SWIM) uçağa yer kontrol merkezinden gelen operasyon verileri, hava durumu, trafik verileri, yeni uçuş planı gibi bilgilerin data linkler üzerinden Günümüz standartlarında pilot ve kule arasındaki iletişim sesli olarak radyo aracılığı ile sağlanmakta, ancak bu operasyon şekli, artan hava trafiğini kontrol etmekte, olağanüstü durumları verimli bir şekilde kontrol etme konusunda yetersiz kalmaktadır. Bu yüzden geliştirilmekte yeni bilgi paylaşımı sistemleri ile (SWIM) uçağa yer kontrol merkezinden gelen operasyon verileri, hava durumu, trafik verileri, yeni uçuş planı gibi bilgilerin data linkler üzerinden aktarılması planlanmaktadır. Bu data linklerin kullanılması ile kokpit içerisinde pilotun sorumlulukları ve iş görevleri tanımları değişmekte; daha önce hava trafik kontrolü tarafından gerçekleştirilen taktik seviyede ayrışma yönetimi fonksiyonu uçuş ekibinin sorumluluğuna bırakılmakta, trafik kontrolörü daha çok stratejik yükümlülükleri olan, güvenlik açısından daha yüksek seviyede gözlemci seviyesine çıkmaktadır. Ancak kokpit içerisine aktarılan ve artan datanın yönetilmesi ile birlikte, bu bilgileri kullanarak verimli olan ancak karmaşıklaşan uçuş operasyonlarını, pilotların gelişmiş otomasyon ve karar destek sistemleri olmaksızın yönetebilmesi de zorlaşacaktır. Bunun yanında rutin uçuş modlarının otomatik hale gelmesi, pilotu durumsal farkındalığını ilgisizlik ve süreç dışına itilmesi nedeniyle azaltmamalı, herhangi bir olağan dışı durumda, durumu anında kontrol altına alabilecek seviyede pilot halen süreçlerin içerisinde kalmalıdır.

AUTOFLY-Aid olarak adlandırılan bu tez çalışması, dinamik 4-Boyutlu rota yönetimi ile çarpışmadan kaçınma ve verimli uçuş rotaları planlamaya yarayan yeni nesil uçuş karar destek algoritma ve cihazlarının geliştirilmesi ve kavramsal tasarımının gerçekleştirilmesini amaçlamıştır. Geliştirilen karar destek sistemleri halihazırda var olan kokpit içi çarpışmadan kaçınma sistemlerinin (bknz. TCAS) eksikliklerini gidermeyi vizyonlamanın ötesinde, uçuşta veri değişimi, sanal gerçeklik tabanlı pilot karar destek, hızla değişen durumlar için otonom uçuş kontrolü sağlama gibi fonksiyonlara olanak sağlayan ek kavramsal aviyonikler ve prosedürler geliştirilmesi de amaçlanmıştır.

AUTOFLY-Aid'in ana konseptleri; a) SESAR ve NextGen modernizasyonlarının 2020+ vizyonları ve ötesindeki hava sahasının kokpit içerisinden algısının matematiksel olarak modellenmesi, b) anlık ve orta-mesafede kompozit bir hava sahasında hızla değişen durumlara karşı alternatifleriyle beraber uçulabilir rotalar ve manevralar üreten 4-boyutlu rota planlama algoritmalarının geliştirilmesi, c) değişen durumlarda

pilota görsel anlama ve durumsal farkındalık kazandıracak sanal gerçeklik karar destek sistemleri, otonom uçuş kontrolü sağlama ve bunun gibi yenilikçi prosedürler içeren bu uçuş otomasyonu sistemlerinin Boeing 737-800 Uçuş Simülatörü içerisine entegrasyonu ve testlerinin yapılmasıdır.

Tamamen merkezi olarak taktiksel seviyede uçuşa müdahale modelinden, daha etkin stratejik seviyede planlama yapma ve daha fazla otomasyon destekli ve daha aktif arayüzler içeren merkezcil olmaktan uzak taktik operasyonlar hem SESAR hem de NextGen gelecek hava trafiği paradigma değişimlerinde ana mesele olarak durmaktadır. Bu yeni nesil Hava Trafik Yönetimi (ATM) konseptleri "en iyi karar noktası", "en iyi karar zamanı" ve "en iyi karar vericiyi" değerlendirilmesiyle insanın ATM sistemi içerisindeki rolünü ciddi şekilde değiştirecektir.

Bu amaçlar doğrultusunda, AUTOFLY-Aid yerdeki hava trafik kontrolörünün bir takım iş yükü ve sorumluluklarını etkin bir şekilde kokpit içerisine taşımayı amaçlamıştır. Geliştirilmiş olan otomasyon sistemi sürekli olarak dinamik çevresel ve operasyonel değişkenleri izleyerek ya da yer sistemlerinden veri linkleri aracılığıyla toplayarak uçuş güvenliğini ve verimliliğini gözlemler ve dinamik uçuş rotası planlaması yapar. Bu sistem gerekli otomasyon seviyesini gerekli aksiyon sürelerini değerlendirerek "en iyi karar vericiyi", "nerede insan iyi, nerede makina iyi" sorgusu yaparak belirler. Olağan durum çalışma modunda, pilot görsel karar destek sistemlerini kullanarak (örneğin sanal tünel içerisinde uçuş) en üst seviye durumsal farkındalık ile güvenli ve verimli uçuşunu gerçekleştirebilmektedir. Bu görsel karar destek sunumları kokpitin kendi duyargaları ve yer-hava arası veri paylaşımları ile edindiği bilgilerin bileşkesinden elde edilmektedir. Eğer gerekli reaksiyon süresi izin verir ise, pilot bu göstergeler üzerinden alternatif rota planları üretebilir, sonuçları değerlendirebilir, tekrar planlama talep edebilir.

Bu tez kapsamında nominal çalışma modunda pilotun karar destek ihtiyaçlarını karşılayacak iki farklı kokpit içi konseptsel arayüz tasarımı yapılmıştır. Sentetik Vizyon ekranı (SVD) ile pilot standart sentetik Arayüzlerinin sunduğu durumsal parametrelerin yanı sıra "tünel içinde uçma" hissi ile sürekli olarak uzaydaki 3 boyutlu konumu ve zamanda ilerleyişi açısından desteklenmektedir. Bunun yanında uçuşa uygun bir çok kritik bilgi de doğrudan bu ekranlar üzerinde aktarılmıştır. 4 Boyutlu Operasyonel Ekran (4DOD) ile pilot uzun vadede uçuş operasyonu içerisindeki bütün bilgilere 3 boyutlu bir arayüz ile ulaşabilmektedir. Bu ekran sayesinde pilot kendi planlanan yörüngesini ve çevre trafikteki uçakların planlanan yörüngelerini 3 boyutlu bir ekran üzerinden görebilmekte, haptik bir araç ile kolayca trafik içerisinde gezinebilmekte, olası ktirik durumları gözlemleyebilmekte, hızlı-oynatma simülasyon modu ile gelecek zamandaki hava trafiğinin projeksiyonunu da izleyebilmektedir. Yer sistemleri ve hava trafik kontrolörü ile olan bütün etkileşimlerde anlaşma üzerinde olunan yörüngeyi de bu ekranlarda görebilmekte yine benzer şekilde ileri zaman projeksiyonları yapabilmektedir. Bu sayede pilotun anlaşma yörüngesi üzerindeki farkındalığı çok yüksek olmaktadır. Anlaşması yapılan yörünge doğrudan Uçuş Yönetim Sistemine (FMS) gönderilebilmektedir. 4DOD çevre uçkaklardan gelen konum ve amaç bildirimleri ile sürekli kendi arayüzünü güncellemektedir. Bu güncelleme, zaman projeksiyonları ve çarpışma denetimi her bir uçağın kendi performans modelleri üzerinden yapılmakta ve tez kapsamında geliştirilen stokastik algoritmalar ile yapılmaktadır.

Anlık bir tehdit algısı oluştuğunda (anlık reaksiyon gerekli olduğu ya da geç kalınan reaksiyon tespiti olduğunda) otomasyon sistemi potansiyel kritik problemi (havada çarpışma, yere çarpma vb.) çözmek amacıyla uçuş kontrolünü ele geçirebilmektedir. Bu hibrid yaklaşım gerekli aksiyon zamanları değerlendirmesi yaparak bu şekilde bir otonomi seviyesi geçişlerini kontrol edebilmektedir. Burada tamamen yerden bağımsız bir karar mekanizması çalışmakta, sadece çevre uçaklardan alınan anlık konum bilgisi ile (ADS-B In üzerinden) anlık çarpışma denetimi yapılmaktadır. Bunun için Oyun Teorisi yaklaşımı ile yine tez kapsamında geliştirilen stokastik algoritmalar kullanılmıştır. Bunun yanı sıra, çarpışmadan kaçınma, iniş/kalkış ve yer operasyonlarında fonksiyonel halde kullanılmak üzere anlık operasyon yönetimi için bir pilot baş-üstü arayüzü (Head-up-display) geliştirilmiş, burada sanal gerçeklik faktörlerinden yararlanılmıştır. Bu sayede pilot bütün operasyonlarını gerçek dış görüntü ile eşleşen (transparan bir arayüz ile) sanal bir tünel içi uçuş hissiyatı ile gerçekleştirebilmektedir. Bu sayede pilot anlık kritik uçuş parametrelerini, 3 boyutlu uzaydaki yerini ve zamanla projeksiyonunu neredeyse tamamen bu ekran aracılığı ile yapabilmesi amaçlanmıştır. Çarpışmadan kaçınma ya da nominal uçuş manevraları (seyir, iniş/kalkış, yer hareketleri) bu ekran üzerine doğrudan aktarılmakta ve anlık durum değişkenleri izlenmektedir.

Tez kapsamında üretilen nominal 4-boyutlu taktiksel yörünge yönetimi, anlık çarpışmadan kaçınma manevra planlaması, konsept sentetik vizyon, operasyon ve pilot baş üstü arayüzleri İTÜ Havacılık ve Uzay Teknolojileri araştırma merkezinde bulunan Boeing 737-800 Uçuş simülatörü üzerine entegre edilmiş ve operasyon testleri yapılmıştır. Bu sayede üretilen yöntemlerin uygulanabilirliği ve teknoloji gösterimi de gerçekleştirilmiştir. aktarılması planlanmaktadır. Bu data linklerin kullanılması ile kokpit içerisinde pilotun sorumlulukları ve iş görevleri tanımları değişmekte; daha önce hava trafik kontrolü tarafından gerçekleştirilen taktik seviyede ayrışma yönetimi fonksiyonu uçuş ekibinin sorumluluğuna bırakılmakta, trafik kontrolörü daha çok stratejik yükümlülükleri olan, güvenlik açısından daha yüksek seviyede gözlemci seviyesine çıkmaktadır.

Ancak kokpit içerisine aktarılan ve artan datanın yönetilmesi ile birlikte, bu bilgileri kullanarak verimli olan ancak karmaşıklaşan uçuş operasyonlarını, pilotların gelişmiş otomasyon ve karar destek sistemleri olmaksızın yönetebilmesi de zorlaşacaktır. Bunun yanında rutin uçuş modlarının otomatik hale gelmesi, pilotu durumsal farkındalığını ilgisizlik ve süreç dışına itilmesi nedeniyle azaltmamalı, herhangi bir olağan dışı durumda, durumu anında kontrol altına alabilecek seviyede pilot halen süreçlerin içerisinde kalmalıdır.

AUTOFLY-Aid olarak adlandırılan bu tez çalışması, dinamik 4-Boyutlu rota yönetimi ile çarpışmadan kaçınma ve verimli uçuş rotaları planlamaya yarayan yeni nesil uçuş karar destek algoritma ve cihazlarının geliştirilmesi ve kavramsal tasarımının gerçekleştirilmesini amaçlamıştır. Geliştirilen karar destek sistemleri halihazırda var olan kokpit içi çarpışmadan kaçınma sistemlerinin (bknz. TCAS) eksikliklerini gidermeyi vizyonlamanın ötesinde, uçuşta veri değişimi, sanal gerçeklik tabanlı pilot karar destek, hızla değişen durumlar için otonom uçuş kontrolü sağlama gibi fonksiyonlara olanak sağlayan ek kavramsal aviyonikler ve prosedürler geliştirilmesi de amaçlanmıştır. AUTOFLY-Aid'in ana konseptleri; a) SESAR ve NextGen modernizasyonlarının 2020+ vizyonları ve ötesindeki hava sahasının kokpit içerisinden algısının matematiksel olarak modellenmesi, b) anlık ve orta-mesafede

kompozit bir hava sahasında hızla değişen durumlara karşı alternatifleriyle beraber uçulabilir rotalar ve manevralar üreten 4-boyutlu rota planlama algoritmalarının geliştirilmesi, c) değişen durumlarda pilota görsel anlama ve durumsal farkındalık kazandıracak sanal gerçeklik karar destek sistemleri, otonom uçuş kontrolü sağlama ve bunun gibi yenilikçi prosedürler içeren bu uçuş otomasyonu sistemlerinin Boeing 737-800 Uçuş Simülatörü içerisine entegrasyonu ve testlerinin yapılmasıdır.

Tamamen merkezi olarak taktiksel seviyede uçuşa müdahale modelinden, daha etkin stratejik seviyede planlama yapma ve daha fazla otomasyon destekli ve daha aktif arayüzler içeren merkezcil olmaktan uzak taktik operasyonlar hem SESAR hem de NextGen gelecek hava trafiği paradigma değişimlerinde ana mesele olarak durmaktadır. Bu yeni nesil Hava Trafik Yönetimi (ATM) konseptleri "en iyi karar noktası", "en iyi karar zamanı" ve "en iyi karar vericiyi" değerlendirilmesiyle insanın ATM sistemi içerisindeki rolünü ciddi şekilde değiştirecektir. Bu amaçlar doğrultusunda, AUTOFLY-Aid yerdeki hava trafik kontrolörünün bir takım iş yükü ve sorumluluklarını etkin bir şekilde kokpit içerisine taşımayı amaçlamıştır. Geliştirilmiş olan otomasyon sistemi sürekli olarak dinamik çevresel ve operasyonel değişkenleri izleyerek ya da yer sistemlerinden veri linkleri aracılığıyla toplayarak uçuş güvenliğini ve verimliliğini gözlemler ve dinamik uçuş rotası planlaması yapar. Bu sistem gerekli otomasyon seviyesini gerekli aksiyon sürelerini değerlendirerek "en iyi karar vericiyi", "nerede insan iyi, nerede makina iyi" sorgusu yaparak belirler. Olağan durum çalışma modunda, pilot görsel karar destek sistemlerini kullanarak (örneğin sanal tünel içerisinde uçuş) en üst seviye durumsal farkındalık ile güvenli ve verimli uçuşunu gerçekleştirebilmektedir. Bu görsel karar destek sunumları kokpitin kendi duyargaları ve yer-hava arası veri paylaşımları ile edindiği bilgilerin bileşkesinden elde edilmektedir. Eğer gerekli reaksiyon süresi izin verir ise, pilot bu göstergeler üzerinden alternatif rota planları üretebilir, sonuçları değerlendirebilir, tekrar planlama talep edebilir. Bu tez kapsamında nominal çalışma modunda pilotun karar destek ihtiyaçlarını karşılayacak iki farklı kokpit içi konseptsel arayüz tasarımı yapılmıştır. Sentetik Vizyon ekranı (SVD) ile pilot standart sentetik Arayüzlerinin sunduğu durumsal parametrelerin yanı sıra "tünel içinde uçma" hissi ile sürekli olarak uzaydaki 3 boyutlu konumu ve zamanda ilerleyişi açısından desteklenmektedir. Bunun yanında uçuşa uygun bir çok kritik bilgi de doğrudan bu ekranlar üzerinde aktarılmıştır. 4 Boyutlu Operasyonel Ekran (4DOD) ile pilot uzun vadede uçuş operasyonu içerisindeki bütün bilgilere 3 boyutlu bir arayüz ile ulaşabilmektedir. Bu ekran sayesinde pilot kendi planlanan yörüngesini ve çevre trafikteki uçakların planlanan yörüngelerini 3 boyutlu bir ekran üzerinden görebilmekte, haptik bir araç ile kolayca trafik içerisinde gezinebilmekte, olası ktirik durumları gözlemleyebilmekte, hızlı-oynatma simülasyon modu ile gelecek zamandaki hava trafiğinin projeksiyonunu da izleyebilmektedir. Yer sistemleri ve hava trafik kontrolörü ile olan bütün etkileşimlerde anlaşma üzerinde olunan yörüngeyi de bu ekranlarda görebilmekte yine benzer şekilde ileri zaman projeksiyonları yapabilmektedir. Bu sayede pilotun anlaşma yörüngesi üzerindeki farkındalığı çok yüksek olmaktadır. Anlaşması yapılan yörünge doğrudan Uçuş Yönetim Sistemine (FMS) gönderilebilmektedir. 4DOD çevre uçkaklardan gelen konum ve amaç bildirimleri ile sürekli kendi arayüzünü güncellemektedir. Bu güncelleme, zaman projeksiyonları ve çarpışma denetimi her bir uçağın kendi performans modelleri üzerinden yapılmakta ve tez kapsamında geliştirilen stokastik algoritmalar ile yapılmaktadır.

Anlık bir tehdit algısı oluştuğunda (anlık reaksiyon gerekli olduğu ya da geç kalınan reaksiyon tespiti olduğunda) otomasyon sistemi potansiyel kritik problemi (havada çarpışma, yere çarpma vb.) çözmek amacıyla uçuş kontrolünü ele geçirebilmektedir. Bu hibrid yaklaşım gerekli aksiyon zamanları değerlendirmesi yaparak bu şekilde bir otonomi seviyesi geçişlerini kontrol edebilmektedir. Burada tamamen yerden bağımsız bir karar mekanizması çalışmakta, sadece çevre uçaklardan alınan anlık konum bilgisi ile (ADS-B In üzerinden) anlık çarpışma denetimi yapılmaktadır. Bunun için Oyun Teorisi yaklaşımı ile yine tez kapsamında geliştirilen stokastik algoritmalar kullanılmıştır. Bunun yanı sıra, çarpışmadan kaçınma, iniş/kalkış ve yer operasyonlarında fonksiyonel halde kullanılmak üzere anlık operasyon yönetimi için bir pilot baş-üstü arayüzü (Head-up-display) geliştirilmiş, burada sanal gerçeklik faktörlerinden yararlanılmıştır. Bu sayede pilot bütün operasyonlarını gerçek dış görüntü ile eşleşen (transparan bir arayüz ile) sanal bir tünel içi uçuş hissiyatı ile gerçekleştirebilmektedir. Bu sayede pilot anlık kritik uçuş parametrelerini, 3 boyutlu uzaydaki yerini ve zamanla projeksiyonunu neredeyse tamamen bu ekran aracılığı ile yapabilmesi amaçlanmıştır. Çarpışmadan kaçınma ya da nominal uçuş manevraları (seyir, iniş/kalkış, yer hareketleri) bu ekran üzerine doğrudan aktarılmakta ve anlık durum değişkenleri izlenmektedir.

Tez kapsamında üretilen nominal 4-boyutlu taktiksel yörünge yönetimi, anlık çarpışmadan kaçınma manevra planlaması, konsept sentetik vizyon, operasyon ve pilot baş üstü arayüzleri İTÜ Havacılık ve Uzay Teknolojileri araştırma merkezinde bulunan Boeing 737-800 Uçuş simülatörü üzerine entegre edilmiş ve operasyon testleri yapılmıştır. Bu sayede üretilen yöntemlerin uygulanabilirliği ve teknoloji gösterimi de gerçekleştirilmiştir.

1. INTRODUCTION

The paradigm shift from a purely centralized tactical intervention model towards a more efficient strategic planning and more proactive tactical operations is a key concept in both NextGen and SESAR visions as reported in [7, 8]. Implementation of these concepts will significantly change the roles and responsibilities in Air Traffic Management (ATM) system for deciding of "best place", "best time" and "best decision maker". For example, the air traffic controllers will have a high-level tactical role to manage the traffic flow, and no longer intervene with the 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 it is needed, and check the alternative plans. This transformation will not only redefine the existing roles of the flight crew but also create additional responsibilities that inherently affect the human performance requirements. Therefore, the future flight deck will require additional avionics, operational procedures with adaptive algorithms, automation systems with advanced decision support tools enabling the pilots to handle the entire tactical operation.

The conflict detection (CD) process guarantees the appropriate separation between the aircraft during their flights. The CD algorithms compare the spatial distance between any pair of aircraft with the mandated separation minima. In the current operational practice, aircraft are kept 3 to 5 nmi apart laterally or 1000 ft vertically to provide sufficient safety margin. The conflict resolution (CR) process generates an appropriate action that satisfactorily solves the potential conflicts detected by the CD. Considering the time horizon, tactical conflict detection and resolution typically involves some challenging issues, such as predicting the aircraft future position, predicting the conflict and issuing the proper conflict alert. The difficulty in predicting the aircraft future position mainly comes from disturbances influencing the flight path such as wind and uncertain intended action of the others.

Figure 1.1: Boeing 737-800 full replica flight deck testbed in ITU Aerospace Research Center.

In both SESAR and NextGen visions, multi-layer structure will continue to play a significant role in ensuring the safety and security of the flight operations. With the concepts of this new ATM realm, the flight decks will also have to be equipped with multi-layer safety automation, where at least one system must work independent from the ground systems [8]. This structure reduces dependency and isolates the system of common mode failures, such as single data error that would invalidate the entire system. By taking these facts into account, non-intent-based tactical collision avoidance tools, i.e. Airborne Collision Avoidance System (ACAS), which does not require any knowledge of the flight intent of the aircraft, will still become crucial when the separation assurance process fails. However, ACAS itself, has no direct impact on the controller's function in providing the separation and balancing the airspace capacity. It issues an alert to prevent potential collision after the proper separation has been already lost. Moreover, ACAS does not submit substantial solution to multiple aircraft intrusion problems, where it processes the problem one-by-one. Therefore, its effective capability significantly degrades in high-density airspaces such as terminal areas.

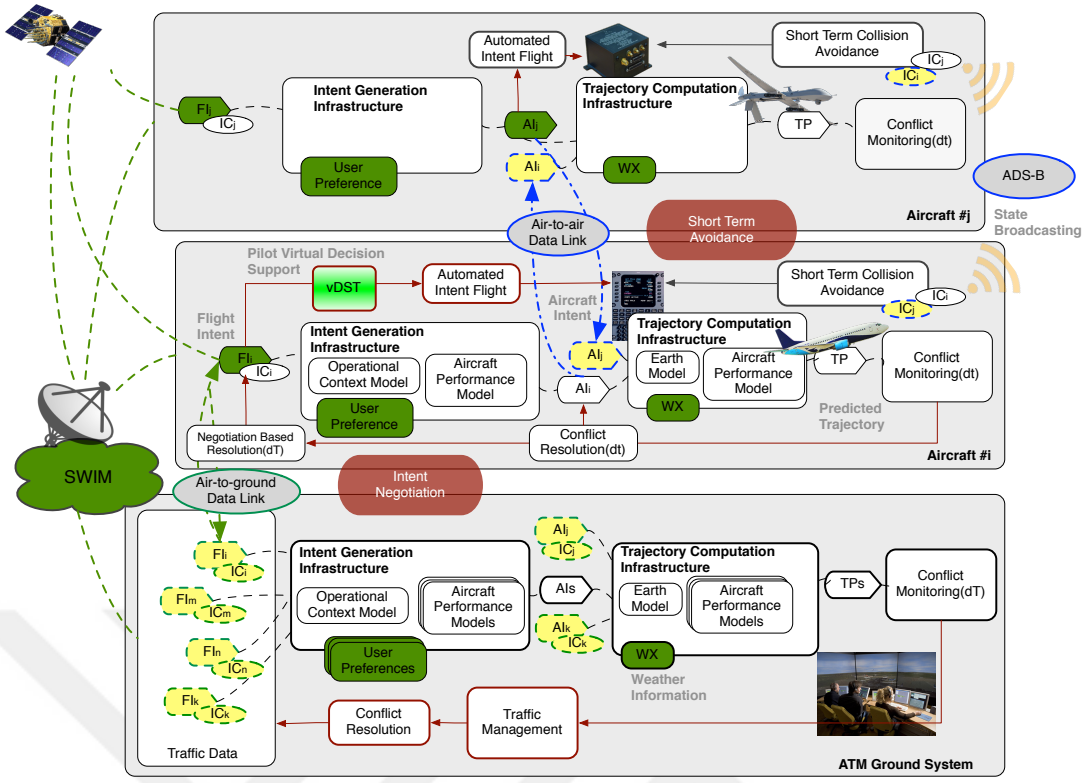

Figure 1.2: The envisioned data exchange and trajectory occurrence procedures for the future airspace needs.

On the other side, with the evolving ATM realm, the way of the managing the flight operations and tactical needs will also change. Fully tactical planning capability will enable airline operators to dynamically redefine the preferred needs according to the evolving conditions. Both ground-based and on-board systems in the current implementation of separation assurance do not 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 effective in-tactical planning will be a delicate issue in the future for the economic viability of the air transportation. For example, dynamic cost parameter (i.e. cost index) managing proposed in [9], which determines how the phases of the flight will be directed (e.g. fly faster or save fuel), enables the airlines to recover delays according to needs of their passengers or their financial strategy. It is shown in [10] that small modifications in the cruise phase operating condition, such as cruise altitude and speed reduction, can achieve significant cost reduction such that they showed 3.71% reduction in cruise fuel-burn hypothetically.

In order to meet the requirements of the future flight operations, we have envisioned to integrate novel automation modules into the current structure of the flight deck

systems. This integrated structure uses two-level autonomy in different kind of time horizons, i.e. Collaborative Mid-term Trajectory Planning and Short Term Collision Avoidance, where both are involving distinctive tools, procedures, data handling and algorithms. The 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 entire envisioned integrated structure and its add-on modules. In the mid-term horizon, processes are mostly operated in a collaborative manner, where the pilot cooperates with the ground systems and uses decision support and automated tools. This module incorporates all tactical level information (i.e. weather data, intent data, user preferences data and further traffic information) obtained from both onboard sensing (including air-to-air data link) and air-to-ground data exchange. The ground-based intent negotiation request may emerge in some circumstances such as a drastic weather change, change in operational constraints, conflict detection, emergency situations or detection of an aircraft does not conform to the anticipated behavior. The Short Term Collision Avoidance module (seen in the Figure 1.2) is an isolated system from the intent data exchange and works independently. Thus, it provides redundancy into the flight deck system (e.g. TCAS). This module only uses position information about the aircraft in the surrounding traffic obtained via air-to-air link. The Conflict Detection block persistently monitors occurrence probabilities of the potential collisions with other aircraft and terrain obstacles within a limited region. Whenever the threat(s) is/are detected (i.e. immediate response is required or late response is detected), it is envisioned that the autonomous system takes over the flight control to solve the issue with the required avoidance maneuvers, which are generated by Short Term Collision Avoidance block. Figure 1.3 summarizes the contributions of this thesis.

In both nominal flight operations and active collaborative decision-making process, it is important to keep the pilot in-the-loop at a manageable level. Moreover, pilots should also recover the flight control from an automation failure. The novel virtual Decision Support Tools (vDST), involving head-down synthetic vision display (SVD) and augmented reality-based head-up display (HUD), gives the pilot full understanding on the evolving flight operation. In addition to these common concepts, another synthetic vision display concept, namely 4D Operational Display (4DOD) (shown in

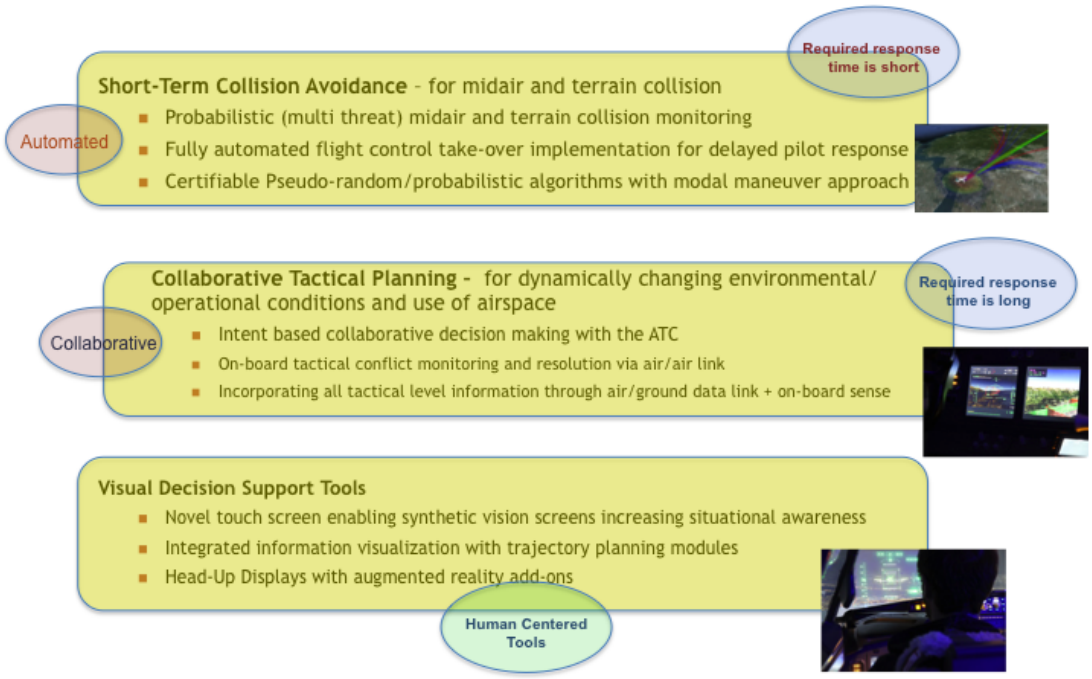

Figure 1.3: Contributions of the thesis in tactical trajectory generation.

3.1), has been developed in ITU CAL for the research purposes. This virtual decision support tool provides 3D virtual projection of the processing flight intent (including predicted trajectories of the surrounding aircraft) to the pilot and enables the required interaction to accept, modify or request replanning — which are the functions of the collaborative decision making. The Conflict Detection and Resolution procedure, which is proposed throughout this paper, which takes places in the Mid-term Trajectory Planning, may potentially be used in Negotiation Based Resolution, or in Conflict Resolution without negotiation requirements (seen in Figure 1.2). The 4DOD supports the pilot to follow the resolution advisories, which are generated by the integrated algorithm that we will present in this thesis.

In Short Term Collision Avoidance part of the thesis, we have integrated game theoretical approach into the sampling based algorithm that approximating the solution of the multi-thread pursuer problem. The control-driven approach based on random tree structure allows us to explore potential action maximizing collision time within the fixed-time horizon. This is a persistent procedure providing a "one-shot" plan to fly and updating itself upon new information (i.e. positional sharing via ADS-B) arriving in each fixed-time window. Whenever the algorithm finds a plan maximizing the collision time at each time intervals, it obviously ensures the maximizing collision time for the entire flight operation. The major concerns in that part become real-time

applicability such as that computational time should be as low as possible to provide a rapid response which also depends on the number of the threat. We aim to provide an algorithm which does not increase the computational time exponentially depending on the number of the threat. We use the "threat" as an "evader" or "intruder" interchangeably throughout this thesis. In 4D Trajectory Planning part, we aim to give a theoretically sound and practically efficient framework for solving the tactical optimal 4D trajectory generation problem in the evolving ATM realm. The proposed method involves a sophisticated aircraft performance model based on BADA 4 and recent algorithmic advances of probabilistic approaches to the motion planning problem that embed stochastic behavior of the effects that are inherent in air traffic (e.g. unpredictable weather conditions). It also embeds the operational cost objectives in the calculation of cost efficient trajectory segments through the predefined flight templates. These flight templates employ approximate trajectory optimization specific to themselves, which are introduced in performance definitions of BADA 4. Specifically, we have utilized two existing flight management models providing cost efficient local maneuver plans, which has been developed in ITU CAL and Boeing RTE. These local maneuver segments based on aircraft performance constitute the global trajectory plan. The trajectory planning algorithm, which relies on searching the airspace and providing proper separation through the local trajectory segments, guarantees asymptotic optimality under certain conditions while maintaining the same probabilistic completeness and computational efficiency of the purely randomized algorithms.

Moreover, we have integrated cross-entropy method, which transforms sampling problem into a stochastic optimization problem, and enables a more efficient and "smarter" way of sampling. The initialization of the problem exploits flight plans that compromised with potential conflict due to unpredicted intruders or the changing environmental conditions (such as wind speed change), where the new solution most likely to be close to the compromised flight plan in the parametric space. This practice is also inherent to ATM, where the strategic flight plan (reference business trajectory — RBT) already reflects many objectives of the stakeholders subject to comprehensive optimization, which is run on the ground systems. In a hypothetical worst-case scenario, where the new solution is far from the previous optimum, the provided

importance sampling converges gradually to a low-discrepancy uniform sampling, which is basically pseudo-random sampling. Otherwise, and mostly, the cross-entropy sampling rapidly converges to a delta function, in the other words, to the minimum cost trajectory. The integration of the proposed strategies provides us to solve challenging in-tactical conflict resolution problem within both the current and envisioned the future realm of the air traffic management.

The thesis also involves the development of innovative visual flight deck decision support avionics to meet the requirements of the future flight operations. These avionics are envisioned to aid pilots for conducting their new in-flight tasks such as a) collaborative tactical planning with intent negotiation/sharing, b) fully understanding situation in 4D (3D spatial + time) and analyzing/interpreting solutions with their alternatives, c) modifying commanded solution (ATC commands coming through data links) or generating new solutions subject to negotiation, and d) aware about delayed required response and allow a collision avoidance module to perform its automated evasive maneuver. This multi-level hybrid approach allows dynamic role assignment by switching defined autonomy level modes associated with the "required response time". These visual decision support tools and interfaces incorporating next-generation synthetic vision and augmented reality-based visualization 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 the 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.

1.1 Related works

The Traffic Alert and Collision Avoidance System (TCAS) (seen in Figure 1.5) is a widely-deployed safety system for reducing the risk of mid-air collision between aircraft. TCAS II [11] provides advisories to pilots to prevent potential conflicts in short-term time horizon. When TCAS detects a potential collision within the next 20 to 48 seconds (depending on altitude), it issues a traffic advisory (TA) in the cockpit. This advisory comes in the form of a spoken message, "traffic, traffic". The traffic icon also changes into a solid yellow circle. In the case of the TA alert, the pilot should search visually for the intruder and communicate with ATC on the progressing situation [11]. If the situation worsens, a resolution advisory (RA) is issued within 15 to 35 seconds before collision (depending on altitude). The RA includes aural command such as "climb, climb" and a graphical display of the target vertical rate for the aircraft. A pilot receiving an RA should disengage the autopilot and manually control the aircraft to follow the advisory of [11]. Collision avoidance alerts represent high-stress, time- critical interruptions to normal flight operations. These interruptions may lead to unnecessary maneuvering that depreciates the efficiency of flows and may also cause pilots to distrust the automation [1]. In the specific example, during an approach to closely spaced parallel runways (to increase the airport capacity) in good visibility conditions, pilots can maintain separation from parallel traffic by monitoring nearby aircraft visually. TCAS, however, does not know that visual separation is being used, and may distract the operation when pilots should be especially focused on performing approach procedures.

1.1.1 Next generation Airborne Collision Advisory System

With new SESAR/NextGen air traffic capabilities and procedures, it is likely that the TCAS II threat detection and resolution logic will require modification to meet newly evolved operational requirements and traffic capacities. Due to the complexity of the logic, modifying the logic may require a significant engineering effort citeKochenderfer:2011vw. The TCAS logic consists of several components:

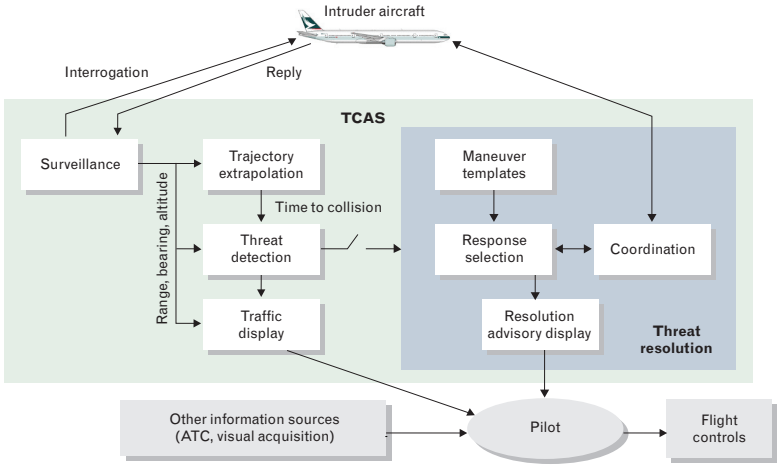

Figure 1.4: TCAS - Traffic Collision Avoidance System functional architecture [1].

threat detection, an initial sense selection, initial strength selection, and encounter monitoring and RA modification. Following figure demonstrates logical architecture of the TCAS.

Mid-term and short-term decision makers (Air Traffic Control, pilots and TCAS) use different information sources, and they work under different constraints and with different goals. In generally speaking, TCAS gets more accurate range or altitude information about an intruder than ATC, but TCAS cannot observe all the factors affecting traffic such as the location of hazardous weather, terrain, aircraft without transponders, or ATCo instructions – this is the major reason that TCAS is certified to operate only as an advisory system [1].

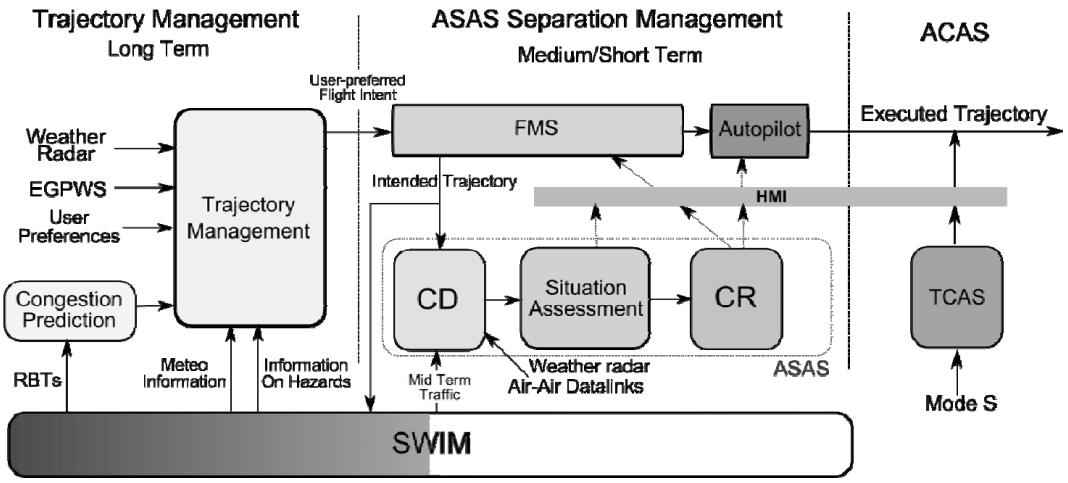

Figure 1.5: Trajectory management information flow and architecture [2].

MIT Lincoln Laboratory Air Traffic Monitoring Program group collected 200,000 flight hours (in 190 days) data within 60 nautical-mile coverage from June 2005 to January 2006 [1]. The group observed 1725 RA events, resulted in that 9 RA events per day, or one RA in every 116 flight hours. In processing this data, the following outcomes are proposed: Only 13% of pilot responses within 5 seconds and achieving a 1500 ft/min vertical rate (met the assumption used by TCAS). In 63% of the cases, the pilots maneuvered in the proper direction, but were not as aggressive as TCAS assumed. Pilots maneuvered in the opposite direction to the RA in 24% of the cases – some of these opposite responses are believed to be due to visual engaging with the intruder aircraft and the pilot's decision that following the RA was not necessary. Opposite response to the TCAS RA can result in exactly the kind of mid-air collision happened at Uberlingen. In Uberlingen accident, one aircraft flew opposite to its RA, and a reversal did not occur [12].

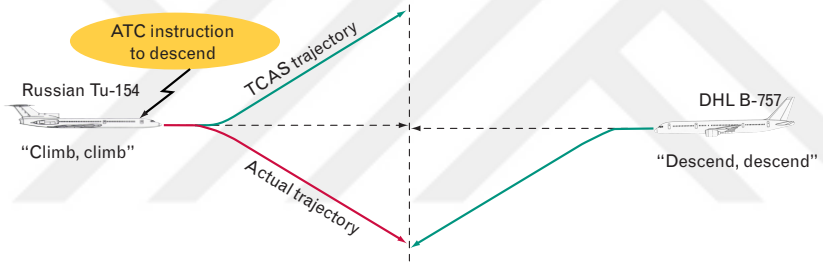

Figure 1.6: The Uberlingen mid-air collision occurred as Tu-154 pilot decided to follow ATCo instruction to descend rather than the TCAS advisory to climb.

A next generation air transport navigation systems should allow aircrafts to modify their flight plans during the flight without approval from a centralised control. Therefore Free Flight concept is extensively studied by the research community including decentralised peer-to-peer conflict detection and avoidance systems. It is possible to integrate some free flight methods as to support the pilots with conflict resolution advisory (with pilot decision support systems). NextGen is currently investigating more delegation of traffic separation responsibility to the pilot [13, 14]. Early ASAS experiments showed promising results of assisted separation operations [15, 16] with the system where pilots are assisted in predicting and resolving loss of separation by cockpit automation, known generally as Airborne Separation Assistance Systems ASAS [17, 18].

As a last note on the subject, future on-board alert systems will possibly integrate more information into flight deck. The next generation alert systems must cooperatively assess, prevent, detect and solve potential conflict situations when an aircraft could enter a Restricted Airspace Area (RAA), a Weather Hazard Area (WHA), a Terrain/Obstacle restriction or the Protected Airspace Zone (PAZ) of another aircraft. Therefore, newly generated conflict detection system should fuse all the information associated with flight safety to depict the complete projected picture. Figure 1.7 demonstrates various hazards that a conflict resolution system may account. A key enabler of this advanced collision avoidance concept of operation is a reliable communication network and information sharing system (e.g. System-Wide Information Management – SWIM [19]). With evolving SESAR and NextGen procedures, aircraft will be equipped with ADS-B Out technology to periodically broadcast their position, velocity and intent information to surrounding traffic and down to SWIM. SWIM will also broadcast surveillance information regarding neighbouring aircraft, current weather, forecasts, special use airspace and other areas to be avoided.

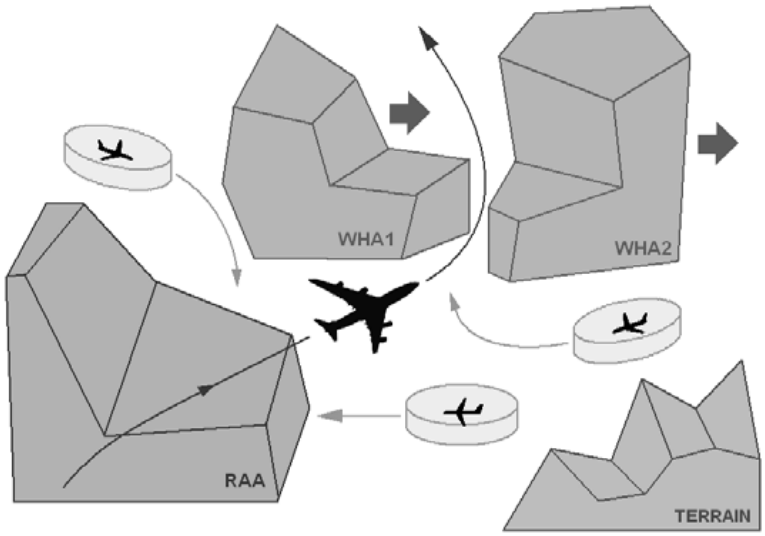

Figure 1.7: Various hazard factors that an on-board alert system should deal with [2].

1.2 Conflict Detection methods

Airborne Conflict Avoidance Systems (including TCAS) first collects state information about intruder aircraft, such as relative position and velocity, and convey the information to their conflict detection algorithms to determine whether a potential collision threat exists. For example; TCAS's threat detection algorithm defines intruders with four discrete parameter [11]. TCAS algorithm performs a linear extrapolation based on aircraft's estimated velocity to project an aircraft future position. Threat metrics decides whether an intruder is a potential threat by estimating vertical and slant-range separation. Tau is the another parameter represents the time to the closest point of potential collision engaging.

Conflict detection algorithms should predict the picture of the future to issue an appropriate alert, and these methods are distinguished according to how they detect potential collisions. Three exploration methods have been identified in the literature and [1] also gives their definitions: nominal, worst case and probabilistic. Figure 1.8 depicts these methods.

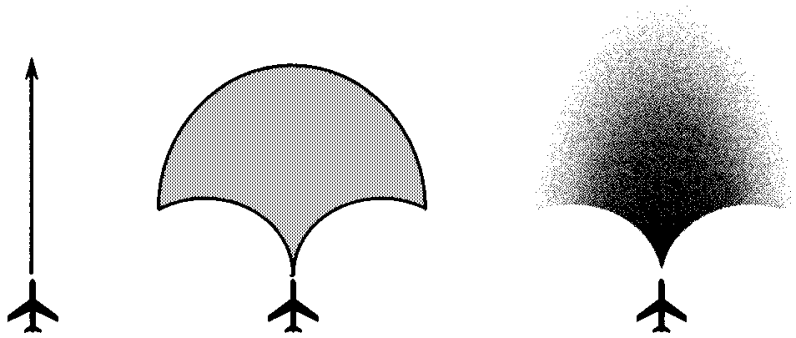

Figure 1.8: Current state projection into future for conflict detection; a) Nominal, b) Worst-case, c) Probabilistic [1].

Nominal projection methods extrapolate aircraft's future position based on the current velocity vector and current state information. These type of methods do not directly account uncertainties and irrationalities (e.g. possibility that an aircraft may not behave as expected). TCAS algorithm is part of this category.

In the worst-case projection, it is assumed that an aircraft may perform any of maneuvers can be done, and if any one of these maneuver could cause a conflict,

then it triggers conflict alert without considering intention of the aircraft motion [20]. This type conservative approach may issue unwanted alerts, often distract the safe operations and reduce overall traffic capacity due to a high false alarm rate [1].

A probabilistic approach contains both these methods; nominal and worst-case methods are special cases of the probabilistic approach. Nominal propagation assigns probability one to the most likely trajectory (linear motion) and worst-case propagation assigns equal probability to all maneuvers can be performed by aircraft. In a probabilistic approach, by assigning different likelihoods to different maneuvers, it is possible to reduce the false alarm rate while managing low-possible threats [21].

1.2.1 State uncertainty

Before reviewing probabilistic projection methods, it is better to give a survey on state uncertainty. Decision making with state uncertainty has been well studied by artificial intelligence communities. But few methods have been applied to airborne collision avoidance problem. In this sense, the main problematic issue is that complexity increases exponentially with the dimension of the state space. Some stochastic methods define the environment with the state uncertainty where it is a partially observable Markovian decision process (POMDP) [22–24]. The decision maker attempts to maximise expected utility with limited knowledge about the environment. POMDP remains short on large dimensional problems such as aircraft conflict resolution due to the complexity. In [25], an online approach for aircraft collision avoidance is proposed that relies on Monte Carlo sampling to search the belief space. The method uses a sample-based method to represent the state uncertainty similar to particle filter method citeThrunProb. Similarly, [26–28] use Monte Carlo technique for accounting state uncertainty. On the deterministic side, [29] suggests an analytic method in order to consider state uncertainty, while [30] is suggesting a numerical approach. Another work [31] uses numerical techniques by integrating Gaussian distributed states to derive a tight upper bound for the probability of conflict over a time period.

State uncertainty and uncertain sources will create an error between aircraft's projected position and actual position in the future. The FAA/EUROCONTROL joint document [3] defines these errors and their sources; cross track error on lateral plane, along-track

error on longitudinal plane, and vertical error. The summation of all these errors represent total error on prediction of the aircraft position. Following figure explicitly demonstrates these errors;

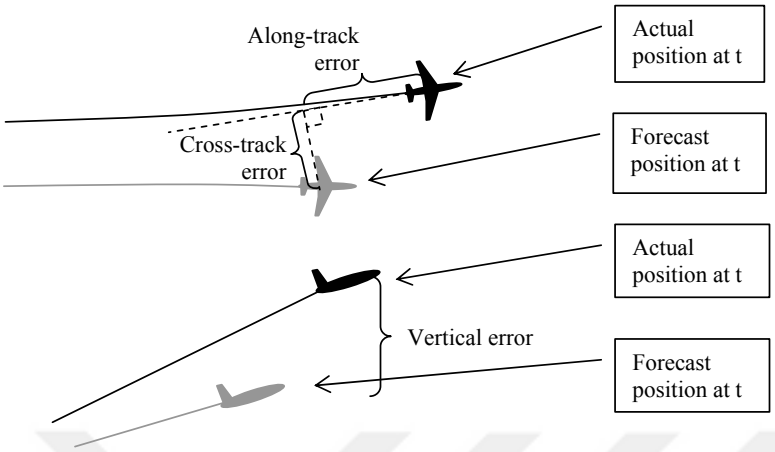

Figure 1.9: Definition of along-track, cross-track and altitude errors [3].

1.2.2 Probabilistic conflict detection

In the probabilistic approach, uncertainty is taken account by generating set of trajectory distribution and computing probability of the projected trajectories. Instead of giving equally distributed probabilities to projected trajectories, to avoid the conservativeness, probabilistic methods assign different likelihood based on admissibility associated with the current case. Various methods have been proposed to address the conflict probability estimation and to manage alarm rate. In the simulation-based approaches, recursive algorithm runs repeated simulations according to motion model and generates unbiased estimate of the probability of conflict. In this type of methods, computational time increases while complexity of the model is increasing. A multi-level approach for rare event probability estimation and particle filtering is used in [32] to reduce computational time while keeping certain accuracy and confidence level. Whenever new sensor measurement is arrived, standard simulation based approach updates the projected conflict probabilities according to a new initial position. In the particle filtering approach, namely Sequential Monte Carlo Method [33], probability of conflict is updated by re-using the previously generated trajectories, and their contributions are refreshed based on new position information.

1.2.3 Alert mechanisms

Because unnecessary alerts are undesirable, such systems must be able to determine whether an alert is required. The common approach in the warning systems involves computing probability of the undesirable event (such as collision) and giving an alert if the conflict probability exceeds the predefined thresholds. This approach has been used widely in airborne [34, 35] and automobile collision avoidance applications [36]. Threshold-based warning systems can be divided into two categories. Constant-threshold systems use fixed thresholds that remain fixed for all operations. Dynamic-threshold systems use varying thresholds that vary as environment evolves.

Three constant-threshold strategies are reviewed in [4]. In conservative policy, [35] suggests issuing an alert when the probability of conflict without alerting exceeds a set of threshold. In delay policy, issuing alert is delayed until the conflict probabilities for all actions exceed the thresholds. Conservative delay policy waits to alert until there is a unique alert that has a probability of conflict below the threshold; as expected, it issues an alert after conservative policy but before the delay policy. Following plot (Figure 1.10) shows the probability of conflict for the various actions.

In [34], a multi-staged threshold alerting system is used; low-probability threats produce passive alerts (e.g., changing the color of a traffic symbol) while high-probability threats produce aural alerts to indicate that evasive maneuver should be performed to resolve the conflict. In expected utility method [37], which is a dynamic-threshold method, the strategy maximises expected utility, where the utility is a function of the cost of issuing an alert and the cost of the event to be prevented. These type of approaches account all future sequences of decision made by the alerting system.

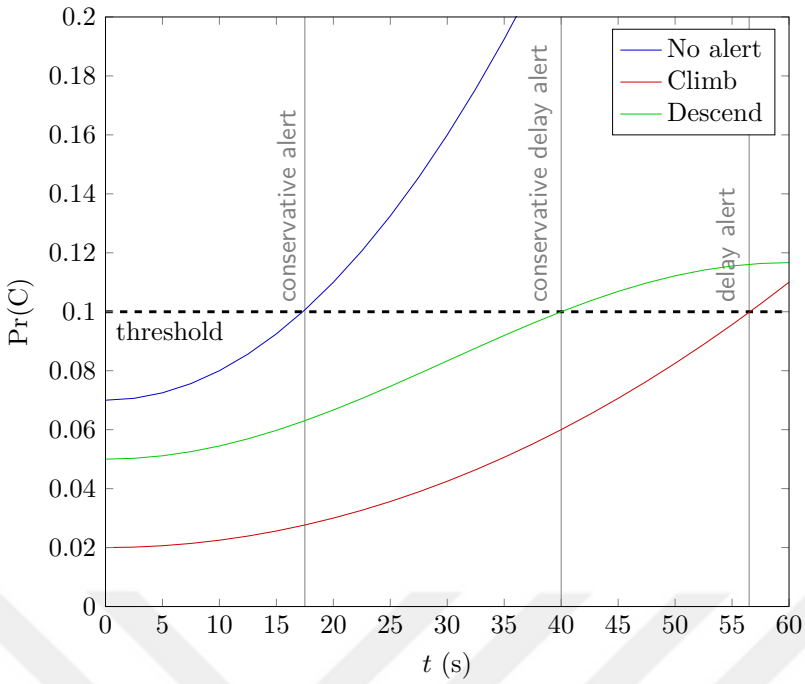

Figure 1.10: The probability of conflict for the various actions is plotted. The conservative strategy alerts as soon as the conflict probability when not alerting reaches the threshold. It issues a climb advisory because it provides the lowest probability of conflict at that point in time. The conservative delay strategy waits until there is a unique advisory (in this case, climb) that provides a conflict probability less than the threshold. The delay strategy waits until the moment all alerts meet or exceed the threshold before it issues a climb advisory [4].

1.3 Conflict Resolution methods

Aircraft collision avoidance systems aid pilots in resolving potential collision whenever collision alert system is triggered. Most algorithms for conflict detection and resolution (in last-minute collision avoidance or maintaining adequate en-route separation) systems, like TCAS, use some form of open-loop planning. Open-loop planners compute a "one-shot" trajectory projection and plans without considering how future information will affect the future actions. Algorithm first generates a sequence of action to take from the current state, when a new observation is arrived, plan is updated. This assumption may not be feasible for multiple-threat collision avoidance in air congestion, where frequent advisory changes may be operationally frustrating.

A comprehensive survey and characterization of the conflict detection and resolution are given in [38] up to its publication date. Many real-time conflict detection and resolution system uses 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 the future actions. The 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 [39], namely NextCAS II, provided a model-based solution that computes the alert thresholds that do not violate intruder's protection zone. In [40], 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 problem. [39] utilized the MILP method for solving conflicts arising among several aircraft, but considered only velocity changing action. [41] suggested a multi-layered open-loop "almost blind engaging" 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. The method exploits information uncertainty, communication delays and possible intruder action intents by using their probabilistic models.

Unlike open-loop methods, instead of generating static plans at each information update, a closed-loop method in [42, 43] has been used to generate an action sequence set that minimize the action cost by accounting future actions, and update alert likelihoods upon the new information availability. Online Markov Decision Process (MDP) algorithm in [25, 44] addressed 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 in [45] where the calculation for the expected utility of being in a particular belief state, and required action selected online; and action utilities are computed offline. A conflict resolution algorithm solving a parametric optimization problem of the point-mass model and utilizing formal definitions for the predefined trajectory parameterization of the aircraft intent has been given in [46].

1.3.1 Maneuver generation methods

Autonomous collision avoidance maneuver generation is still an open area for realistic cases. Motion planning problem for aerospace vehicles is complicated by the fact that, planners based on optimal performance begin to fail by means of computation, when one takes into account of constraints related with dynamical equations of aircraft. Due to fact that, aircrafts state space is at least 12 dimensional, input-state search becomes too complicated; therefore such planners are only successful for vehicles with small state space dimensions. To reduce the complexity of this problem, motion description languages and quantized control concept have been adopted into motion planning [47]. Motion description languages make use of classified combination of simplified control laws to track generalized outputs. Most of these languages are strongly connected with the concept of hybrid systems. In general, hybrid systems classify the motion by using discrete states and switch in between according to input and state information and each discrete state having its own continuous dynamics. A subclass of such languages, which are based on the classification of behaviour (or reaction) of the dynamic systems, has been successfully adapted for non-holonomic robotic systems [48]. More recently, closed loop hybrid control systems were developed based on linear temporal logic for the same purpose by [49]. For aerospace vehicles, a hybrid model for aircraft traffic management was developed in [50]. Study showed that, hybrid system representation gives an opportunity to calculate reachable sets of the system and design hybrid control laws to drive the system to safe states [50].

In real time applications, such as tactical conflict resolution, the principal concern is to determine 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 from the trajectory planning literature. As such, there has been an recent increasing interest (as demonstrated in [51–54]) to provably improve the quality of the sampling algorithm's solution as the 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 obstacles. This approach provides significant savings in computation time since collision check is performed when it is required. The connectivity of these samples is strongly connected with 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 [55]. Recently, the RRT algorithm and its variants were successfully demonstrated on different dynamical systems [56–60]. An important step towards efficient optimization using randomized planners was taken in [54] in which they have proven that the RRT algorithm converges to a non-optimal solution with probability one. Furthermore, they proposed a new algorithm, called RRT^*, and showed that it is globally asymptotically optimal while maintaining the same probabilistic completeness and computational efficiency of the baseline RRT. We have preferred to employ RRT^* in conflict resolution problem due to its superior properties to the other sampling based methods.

A common concern in randomized algorithms is their lack of repeatability, which makes them unable to certify their success and performance. Two successive runs of such algorithms may not produce the identical solution even under identical initial conditions; while a deterministic algorithm always performs exactly the same behavior. It is not possible to give a positive proof that any randomized algorithm solves a motion planning problem very quickly. Besides, [52] clearly showed that deterministic sampling strategies outperform purely random sampling in solving high dimensional problems. In practice, to address this issue, mostly a meta-heuristic monitors the growth of the number of samplings, and resets the search graph if its size exceeds a certain threshold since the processing complexity increases dramatically (e.g. finding nearest node) while the size of the search graph increases. However, the performance of the sampling based algorithms can be further improved by employing adaptive sampling that lead the algorithm to more efficient sampling. Several methods, such as [61–66], have been proposed to improve the sampling strategy of the algorithm by utilizing the knowledge that comes from previous loops of the sampling. [67] transformed the sampling problem of the motion planning into a stochastic optimization problem, where the cross-entropy (CE) method employs to estimate the parameter set of a distribution that guide the algorithm to sample around the optimal trajectory. The CE method was originally introduced by [68] for estimating

rare event probabilities. Then the method with its adaptations has become useful tools for multi-extremal nonlinear optimization. Specifically, [67] has integrated the CE method into the RRT^* to iteratively update the distribution in accordance with their cost, until the distribution concentrates around the optimum trajectory. In this work, we closely followed [67], and provide a more generalized form of the CE sampling will employ within the realm of the conflict resolution.

Motion planning problem for aerospace vehicles is also complicated by the fact that planners based on optimal performance begin to fail by means of computation, when one takes into account the constraints related to the dynamical equations of aircraft. To reduce the complexity of this problem, motion description languages and quantized control concept has been first proposed in [47]. Multi-modal maneuver modeling framework basically consists of decomposition of any trajectory into a set of maneuver modes and associating maneuver parameters. Complexity of maneuver planning part is significantly degraded by reducing the dimension of the problem (the modal sequence has strictly lower dimension than state space description), and parameter optimization part was relaxed by designing specific optimization procedures in each mode. The planner constructs any trajectory with proper modal sequence by switching them instead of performing a highly complex multi-objective parametric optimization over the full flight envelope. This approach is successfully applied in [69] for autonomous combat aerial vehicles, which involve complicated modeling and control. In the approach presented in [69], parameterized sub maneuvers build up complex maneuver sequences and make possible to cover almost any arbitrary maneuver and the entire flight envelope.

In parallel with these works, closed-loop hybrid control systems were developed based on linear temporal logic for the same purpose by [49]. For aerospace vehicles, a hybrid model for aircraft traffic management calculating the largest controlled invariant subset of each aircraft's protected zone was developed in [70], where relatively simplistic horizontal maneuver modes are used for algorithmic demonstration of a hybrid approach. Similarly, [71] suggested a maneuver automaton, which uses a number of feasible system trajectories to represent the building blocks of the motion plan of the aircraft, and a trajectory based control system, which asymptotically regulates the actual trajectory to the trajectory generated by the maneuver automaton.

However, motion plans and controllable trajectories are restricted to the library of the maneuver automaton. Such libraries are built by using interpolation between feasible trajectories as proposed in [72]. [73] extended this fashion for online planning of feasible trajectories in partially unknown environments by using receding horizon iterations. Maneuver Modes and 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 [74], 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 applied to that mode.

By addressing the computational complexity, Frazzoli [75] suggests a maneuver automaton, which uses a number of feasible system trajectories to represent the building blocks of the motion plan of the aircraft, and a trajectory based (based on maneuver regulation principle) 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 (similar to TCAS). Such libraries can be built by using interpolation between feasible trajectories [76]. In [75], autonomous agent chooses the best automaton using rapidly-expanding random trees (RRT). [77] extended this system with Mixed Integer Linear Programming for online planning of feasible trajectories in partially unknown environments by using receding horizon iterations. This method solves for an optimal policy in a given state, assuming some predicted future sequence of states, and chooses the current optimal action. This process is repeated when the environment changed. Description of aircraft dynamics from hybrid system point of view has been studied previously in [75, 78, 79] .These works have been successful in utilizing the advantages of hybrid system methodology in control of both single and multiple aircrafts. Uncertain state observation and stochastic state transition are also considered in collision avoidance problems. Partially Observable Markovian Decision Process (POMDP) [23], a decision theoretic planning framework, includes observation model, transition model, a reward function and action set, and the algorithm tries to maximise the future rewards of the agent. Beginning with the initial belief state, an aircraft receives observation on-the-fly, and updates its belief and choses its best

action using sensor models. [44] proposed a POMDP solution converges to an optimal policy (within policy set) by maximising the expected future reward, and results in reduced alert probability of unnecessary alerts compared to other algorithms. The algorithm involves Gaussian intruder process noise and intruder behavioural modes for belief state compression such as climbing, level, or descending. Consequently, POMDP solution remains on to use an associated set of possible maneuvers while the reward function is used to score their performance. Reward function may involve any objectives such as avoiding collision and minimising deviation from the planned path. However, these approaches do not include the full flight envelope dynamics of the aircraft. Specifically, both mode selection and controller design is strictly based on selected maneuvers; therefore controllability is limited to these predefined trajectories. In case of the multiple threat, in densely congested airspace, potential solution is limited to existing finite maneuver set. In the approach presented in [73], parameterized sub maneuvers build up complex maneuver sequences and makes possible to cover almost any arbitrary maneuver and the entire flight envelope by this approach.

On the side of the Game Theoretic approaches, aircraft conflict avoidance is grouped into two categories: cooperative and non-cooperative. In cooperative methods, the pilot of ownship can access the state and intent of any surrounding aircraft via a communication link (e.g. Mode S). Noncooperative process assumes that an aircraft do not know the flight intents of any adjacent aircraft. Because of the many uncertain factors such as wind and potential blunders, noncooperative approach tries to guess the potential conflict by projecting the situation to noncooperative game. In [70], noncooperative zero-sum game solution is proposed describing the boundaries of unsafe sets for continuous dynamical systems. The solution considers the worst case for two aircraft and uses predefined feasible maneuver set. [80] verifies that this approach is robust to potential blunders of other aircraft, and does not lead to a loss of separation in the application of closely spaced parallel approach.

Prescribed resolution maneuver solutions (it is reviewed in [35]) base on a library of predefined action plans as that GPWS issues a standard "pull up" warning when a conflict with terrain. Prescribed maneuvers may have the benefit that operators can response in a lower time since they are not faced with unexpected advisory as they

were trained before. Since there is no opportunity to modify the resolution maneuver, prescribed maneuvers are less effective than the maneuvers that are computed in real time [35]. In some case of conflicts, it may be necessary to adopt the resolution maneuver to account for unexpected events in the environment, or to reduce/increase the aggressiveness of the maneuver should the conflict be resolved easily/harder than first predicted.

1.3.2 Pilot response to advisory

Pilot Response is an another uncertainty to generate best maneuver action to solve the potential conflict. The TCAS logic also uses a deterministic pilot response model to forecast the future paths of the aircraft. In this deterministic model, it is assumed that pilot responses (after 5 seconds delay) to give advisories as exactly expected [11]. Under certain conditions, if the situation is continuing to degrade, TCAS can even reverse the sense of the RA. Coordination of this reversal with a TCAS-equipped intruder aircraft is performed through the Mode S data link. Sense reversal is especially challenging because only a few seconds may remain before collision [1]. Recorded radar data reviewed in [42] have shown that there is significant difference in the delay and strength of the pilot responses to advisories. In [5] Bayes nets and game theoretic approach is also used to predict the behaviour of hybrid systems involving both human and automated systems like collision advisory and pilot relation. In this approach, namely Semi Network-Form Game, utility function of pilots base on "happiness factor" such avoid collision, path deviation and obeying TCAS. Then algorithm tries to find the best response for the aircrafts by considering probabilistic human behaviour and advisory system model. Figure 1.11 shows predicted trajectories sampled from the solution of the method.

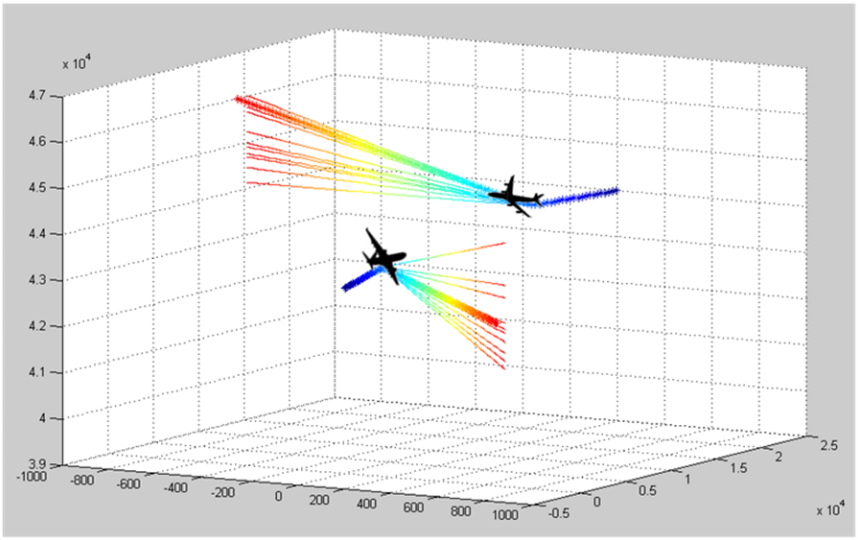

Figure 1.11: Each aircraft proceeds on a straight-line trajectory until the pilot receives an RA. At that point, the pilot uses level-K d-relaxed strategies to decide what vertical rate to execute. The resultant trajectories from 10 samples of the vertical rate are shown. The trajectory assumed by TCAS is shown as the thicker trajectory [5].

1.4 Decision Support Tool and Situational Awareness

Over the last 50 years, systems and instruments on flight decks have been changed or modified to support the pilot in performing flight tasks. Systems for modern flight deck designs are getting more automated and new design philosophies are emerging with crew alerting and messaging systems to support the flight crew in monitoring dynamical changes in the environment. These design philosophies and technology improvements provide for enhanced safety and reduced workload. The second-generation glass cockpit is started during the late 1980s, these new-generation cockpits were the introduction of primary flight displays, navigation displays, multifunction displays and systems displays, which were designed to maximize situational awareness for the flight crew [81]. Newly designed flight decks have provided more integrated information enabling greater safety, efficiency and reduced cost.

New functions like 4D trajectory, airport navigation systems or synthetic vision are expected to meet future mission management requirements. Due to the limited size of cockpit displays, the integration of these new applications on current displays

will saturate the crew with information. The ongoing ODICIS project [82] provides a single large, curved, seamless, avionic display with tactile interfaces. With the touchscreen-based display concept, ODICIS plans to create a self-configurable display space and touch screen surface enabling information to be presented for all types of aircraft and flight operations. Synthetic Vision and Augmented Reality Displays are emerging as enabling technologies within the future flight deck. Particularly these systems are all aimed to provide the flight crew with significantly improved and increased awareness. Integration of these technologies to the cockpit is envisioned to provide a potentially unlimited field-of-regard awareness for terrain, obstacles, traffic, and airspace for the flight crew. As such, NASA recently conducted a preliminary evaluation of a viable technology (as seen in Figure 1.12) to support the Enhanced Vision Operation (EVO) concept for approach and landing operations [6].

Figure 1.12: Head Worn Display - Google concept for augmented reality aided flight operations [6].

1.4.1 Situational Awareness

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 [83]. Progression of these layers, the level of Automation and the extent of SA does not exhibit a simple 1-1 relation. For example, inappropriate levels of automation can impact SA with results such as automation complacency, automation mistrust, increased workload, and automation transparency 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 [84]. When the pilot perceives the automation to be unreliable and gives excessive attention to monitor the automation, SA can also be diminished with a high workload and result in a phenomenon called attention tunneling [85]. 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 [86]. 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 [87]. In a fully transparent system, a pilot may be led to attending to "too much and too low level" system information, resulting in high workload and diminished SA [88].

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 [89]. Conflict resolution experiments are conducted in [90] 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 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.

In the persistent interfaces, the pilot gives her full attention to the interaction to keep track of the labels and pages visited in the process of discovering the correct action sequence [91]. This 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 an inordinate amount of precious time heads-down trying to manipulate 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 [92].

1.4.2 Next generation Pilot Decision Support Systems

For flight deck displays, there are currently a number of systems that provide different types of data such as Traffic Collision Avoidance System (TCAS), Enhanced Ground Proximity Warning System (EGPWS), Aircraft Communications Addressing and Reporting System (ACARS), and Weather Radar and Enhanced Vision System (EVS). As of today, these systems are not fully integrated / fused into a composite air, proximity and intent picture. This results in an increase in pilot workload and a less than optimum SA for potential hazards. These multiple systems carry only a primitive hierarchical integration and force the aircrew to sort through each system while simultaneously interfacing with ground controllers [91]. Therefore fused information display may indeed be the ideal solution to this problem, however with information fusion there are not only problems with tractable data quality and integrity, but also problems with cluttered screens with excessive amount of information types. For example, in the experiments [93] conducted by NASA Advanced Control Displays Unit, pilots provided a number of ideas for clutter mitigation. One thing that is explicitly reported is that during short-term collision avoidance operations, the non-conflicting aircraft or the aircraft away from a specified distance from the ownship should be removed from the display. This naturally suggests that allowing the pilot to choose and progress in time horizons is more in-line with strategic decision-making process.

Helmet-mounted displays (HMD or HUD) or head-worn displays (HWD) can create spatially integrated and large field-of-view augmentation (Figure 1.12). Although these display technologies are not new technologies, especially within the context of

military operations, component miniaturization and maturation are progressing to the point where they can be considered in commercial and business aircraft operations [6]. Transparent screens are another candidate technology for augmented reality integration to the flight deck. However, because of the collimation problem, it is not possible to directly use such devices without further improvements. The collimation refers to the concept of making the image appear to be coming from a distance much further than the display surface. In [94], collimation levels of the transparent screen as an ATC environment interface are compared with respect to response times of the operators. The experiments conducted proposed that operators using simulated collimation in transparent projection screens showed a significantly slower response when performing a visual search task, compared to performing such tasks on non-collimated display. On the other hand, it should be noted that transparent screens do not require additional head-tracking system while flight goggles need this technology for both pilots and copilots.

The response times of the input devices which allow the decisions and the tasks to be directed to the flight system are also critical. In the experiments conducted in [95], four technologies; rotary controller, trackball, track pad and touch screen are compared with regards to the required times as to input tasks. The outcome of the experiments supports the use of the touch screen for the control of menu-based tasks in the flight deck. However, in these experiments, physical location of the input device/display in relation to the pilot and the effects of vibration is not considered. Especially in a naturally vibrating flight environment, touch screen input devices may bring an additional workload to the pilots.

2. INTEGRATED TESTBED: FLIGHT DECK AND ATM SIMULATOR

The integrated system, including a B737-800 flight deck (Figure 2.2) and an ATM testbed (Figure 2.4), 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 flight deck structure utilizes two different autonomy levels and handles switching these autonomy level modes considering the required response time to an action. 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 to be incorporated into the flight deck to support the pilots and significantly enhance their situational awareness of the pilot. Figure 2.1 depicts whole integrated structure and its add-on modules. The software and communication structure of the entire testbed with their physical links is also given in Figure 2.3 for further understanding.

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 incorporate 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 such as drastic weather change, change in operational constraints, conflict detection, emergency situations or detection of an aircraft does not conform to

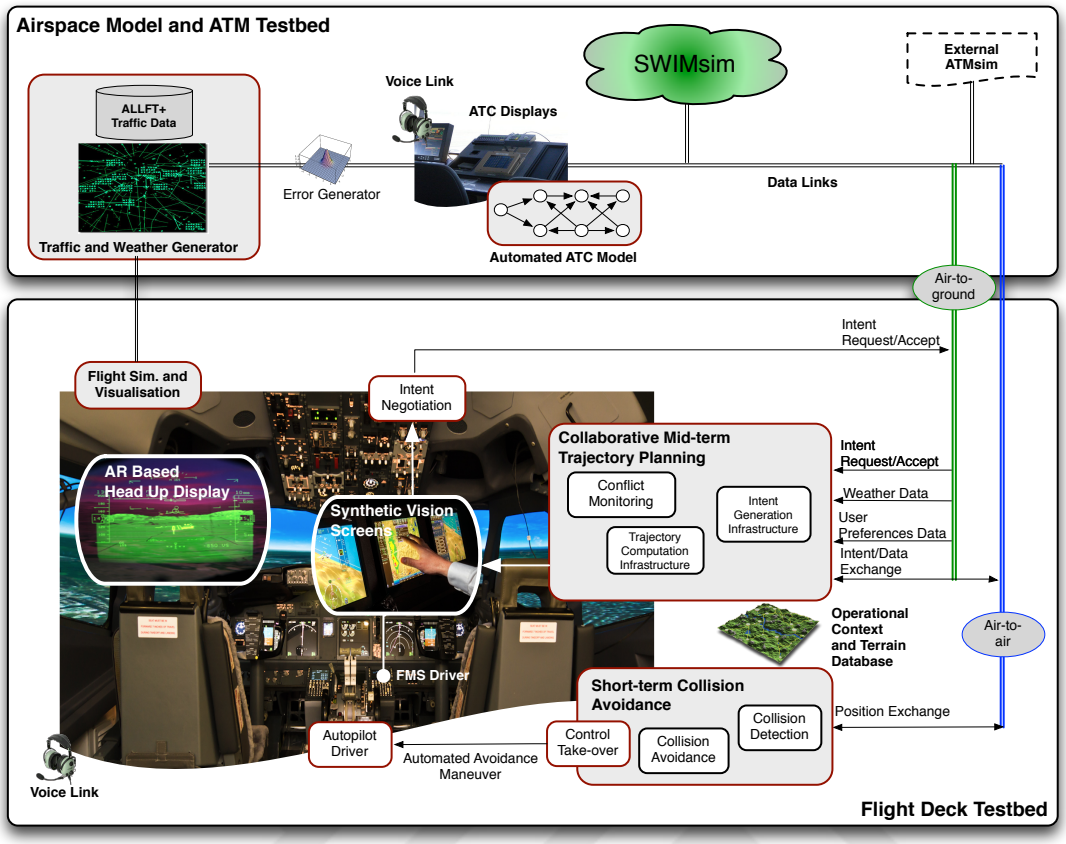

Figure 2.1: Architecture of the integrated next generation flight deck system with novel add-on modules.

the anticipated behavior. 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 initiate 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 validate the feasibility of the given intent data, and Conflict Monitoring block checks potential conflicts between the predicted trajectories in the traffic.

In both SESAR and NextGen visions, multi-layer structure will continue to perform a major role in ensuring safety for flight operations. Therefore, the flight decks will have to be equipped with multi-layer safety automation where at least one must work independently from the ground or air [8]. This will reduce reliance on the ground and isolate the system from common mode failures such that single data error would invalidate the entire system. By considering these facts, redundant collision avoidance systems will still be crucial when the collaborative separation assurance

Figure 2.2: Boeing 737-800 full replica flight deck testbed of ITU Aeronautics Research Center in a nominal tactical operation.

process fails. The limitation of this method is that the prediction error tends to grow quadratically with time; thus, these types of tools will still remain within the domain of the immediate to short-term collision avoidance.

The Short-Term Collision Avoidance module (seen in the Figure 2.1) is an isolated system from the intent data exchange and works independently. Thus, it provides redundancy in the flight deck system. This module only uses position data received from the aircraft in the surrounding traffic obtained via air-to-air data link. The Collision Detection (CD) block persistently monitors occurrence probabilities of potential collisions with other aircraft and terrain obstacles for bounded local region. The CD algorithm uses worst case approach and takes into account both uncertainties in position measurement and pilot actions (e.g blunders of pilots). Whenever the immediate threat(s) is/are detected (i.e. immediate response is required or late response is detected), the autonomous system takes over the flight control to solve the issue with required 3D avoidance maneuvers which is generated by Collision Avoidance block. The worst-case approaches could produce false alarms often due to their algorithmic natures. To address this issue, Short Term Collision Avoidance module delays taking-control action until it decides that "humanly response" may not be

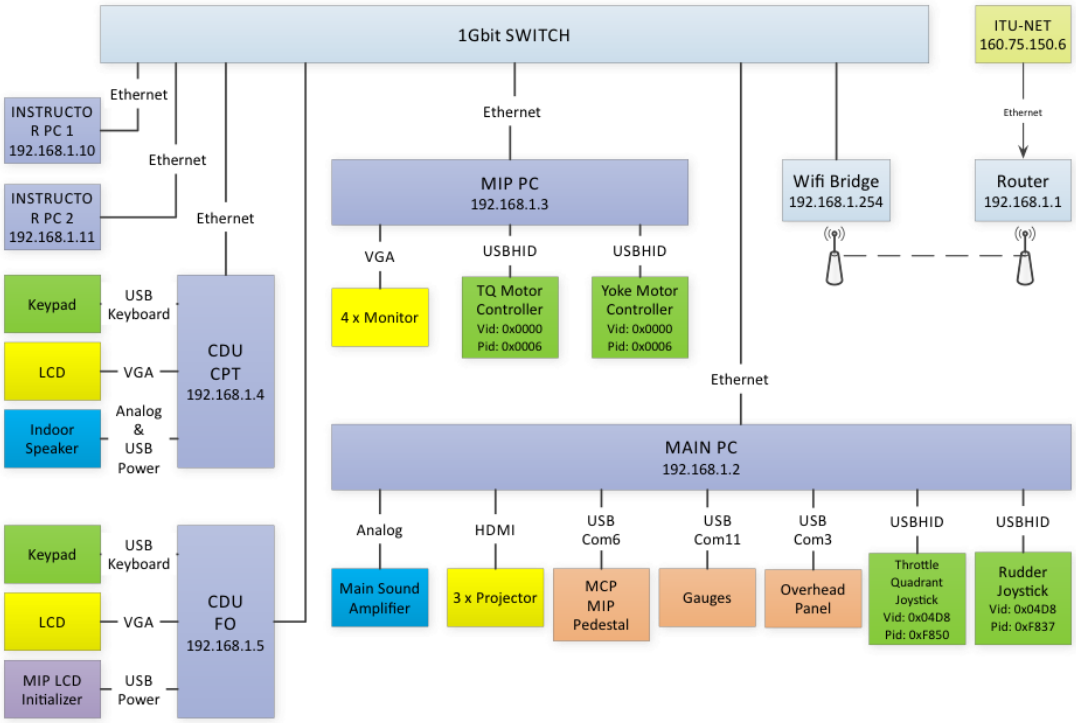

Figure 2.3: Software Architecture of the integrated System.

achieved within the required response time. Then it switches the system into the higher autonomy level (Parasuraman's autonomy level 6 [84]) where the avoidance maneuver is performed autonomously. The head-up-display (HUD), Synthetic Vision Display (SVD) and synthetic 4D Operational Display (4DOD) provides the pilot with an appropriate warning about the collision with visual timer countdown for "pessimistic required response time" before the possible automated avoidance maneuver execution.

Figure 2.4: Airspace Model and ATM Testbed.

Airspace Model and ATM Testbed 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 a complete airspace picture. Similarly, customized 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. An example screen-shot from the approach screen of ATM testbed can be seen in Figure 2.5.

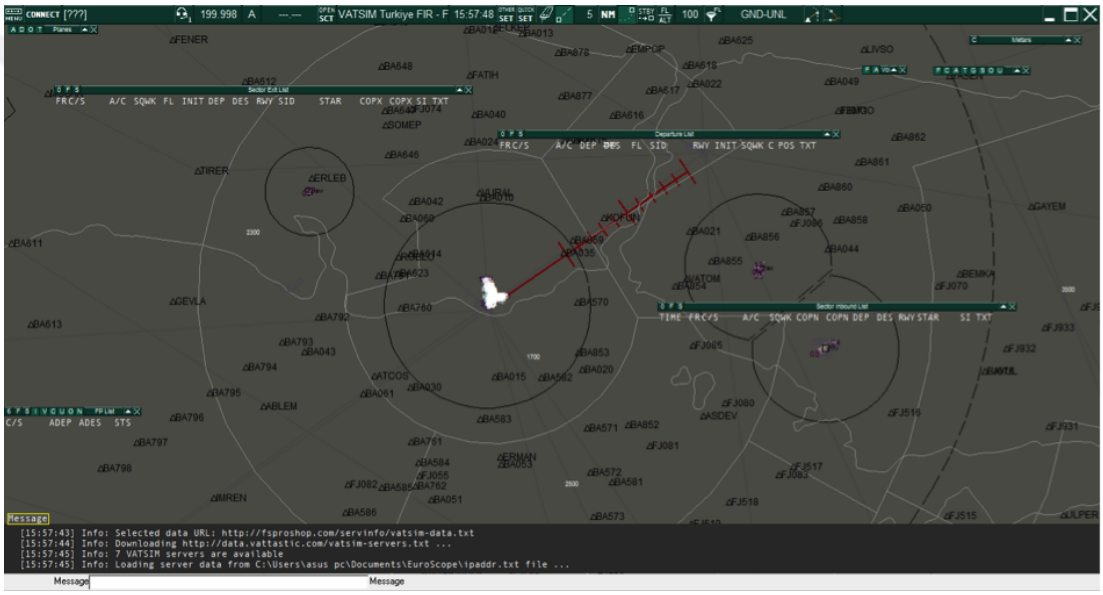

Figure 2.5: Example screen-shot from ATM Testbed: Approach screen.

2.1 Co-simulation with remote ATM system

The integrated system allows us to run cosimulation with remote systems through the Internet. In order to demonstrate the ability, we have established a network between ITU Boeing 737-800 Flight deck testbed and AU Air Traffic Controller, which enable to run joint scenarios. AU ATCo Training Center involves two controller working positions (CWP); radar control position and terminal area working position. Figure 2.6 shows both working positions of AU ATC Training Center for the controllers.

The general architecture of the working position is given in Figure 2.7. A circuit of test simulation includes controller positions, virtual pilots and a supervisor managing

Figure 2.6: ATCo Training Center of AU with radar position (left) and terminal area position (right).

the scenario. The ITU B737-800 simulator is gotten involved into a scenario through Co-Simulation Data/Voice Server which is deployed over one of the virtual pilot positions. This server enables bidirectional communication between ITU B737-800 Flight Deck and AU ATC Center. Hence, the pilot of B737-800 in ITU can hear the controller in AU ATC Center and vice versa. Similarly, the controller in AU ATC Center can see the B737-800 of ITU in radar screen, while the pilot of B737-800 can see the same traffic in his/her displays. Figure 2.8 shows a screen capture of the controller radar screen, which is taken during a co-simulation. The left-lower corner in Figure 2.8 also shows an operational screen (developed in ITU) of the flight deck.

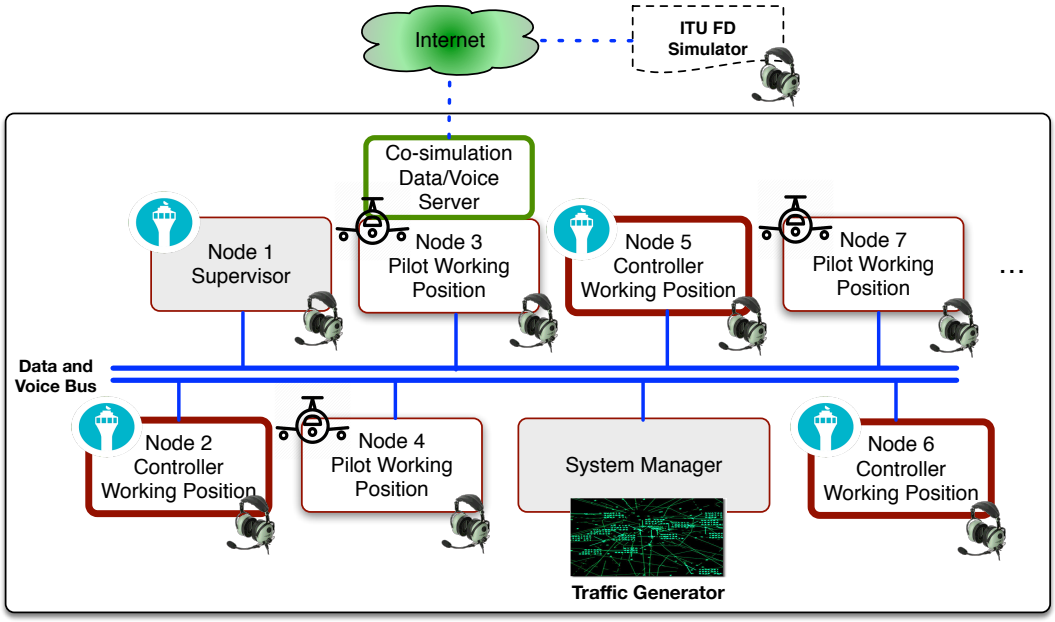

Figure 2.7: ATCo Training Center of AU with remote connection to ITU Flight Deck testbed.

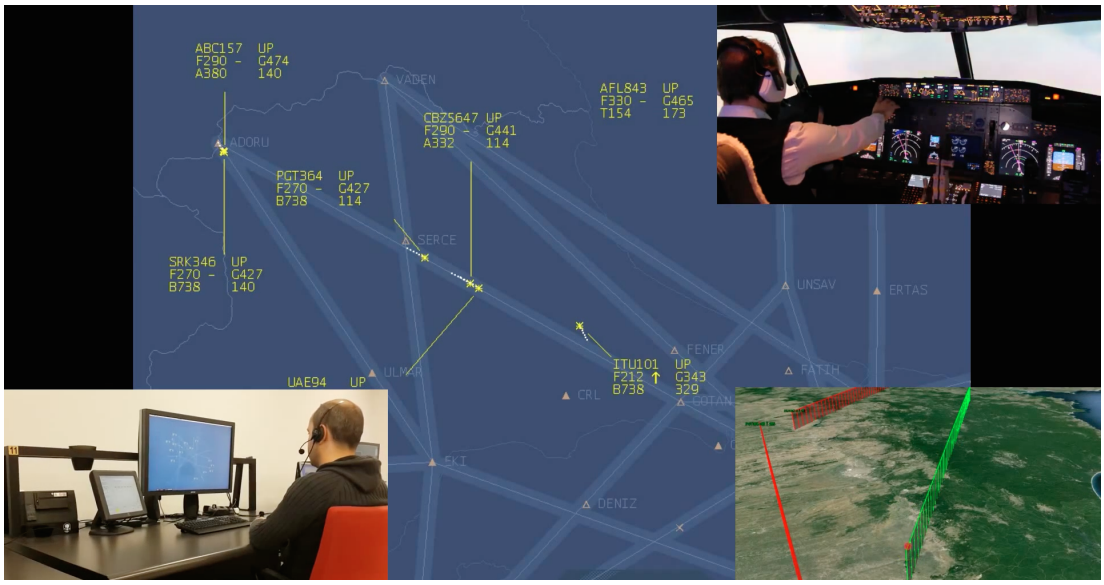

Figure 2.8: A radar screen capture during co-simulation for en-route separation scenario.

Following screen capture (Figure 2.9) is gotten during approach tests where the B737-800 of ITU is approaching to LTBA - Istanbul while a controller in AU ATC managing the terminal traffic.

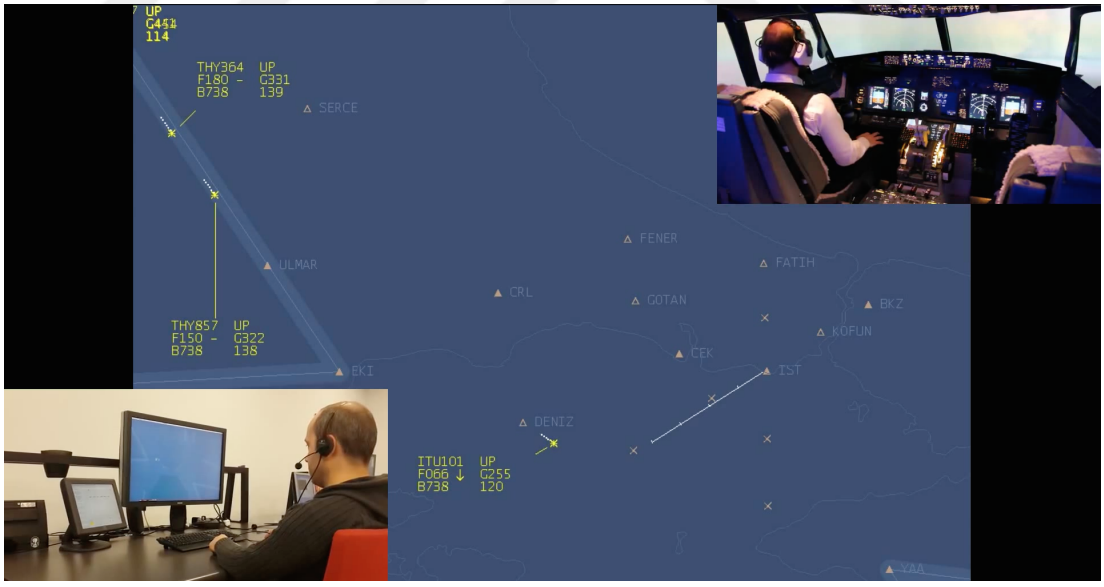

Figure 2.9: A radar screen capture during co-simulation for approach scenario.

The Figure 2.10 shows landing scenario during co-simulation. The pilot of B737-800 of ITU also utilizes the synthetic vision (developed in ITU) screen with tunnel-in-the-sky implementation during instrument landing.

Figure 2.10: Radar screen of the controller and synthetic vision screen of the pilot during co-simulation for landing scenario.

3. NOVEL FLIGHT DECK DECISION SUPPORT SYSTEMS

Monitoring the environment and analyzing the provided solution with their alternatives, in both space and time, within various constraints, is a complex task for the pilots, especially under high level stress. The crew cannot be expected to perform such a complex task without some new form of automation and decision support tools. However, inappropriate levels of automation, for instance, high levels of automation can create a case in which the pilot no longer actively processes information due to over-trust in the system. Such a case effectively diminishes the pilot's ability to recover from failure [84]. When the pilot perceives the automation to be unreliable and gives excessive attention to monitor the system, situational awareness of the pilot is diminished with a high workload and result in a phenomenon called "attention tunnelling" [85]. In a transparent system, where the underlying information behind the automation can be fully accessible, the pilot may be led to attend to "too much and too low level" system information, resulting in high workload and diminished situational awareness [88]. Considering these three cases, an expectation from a good decision support system is that it should provide transparency at a manageable workload level and allows the pilot to be in-the-loop in a cooperative manner [89]. Conflict resolution experiments are conducted in [90] support this proposition.

Over the last 50 years, systems and instruments on flight decks are getting more automated, and new design philosophies are emerging with crew alerting and situational visualization systems to support the flight crew in monitoring dynamical changes in the environment. These design philosophies and technological improvements are envisioned to provide for enhanced safety and reduced workload. New functionalities will be integrated in future cockpits such as en-route 4D trajectory implementations, low visibility operations or new data-link implementations (e.g. collaborative trajectory management) are expected to meet future operational requirements. For instance, System Wide Information Management (SWIM) implementations allow the pilot to obtain almost real-time information about the flight

operation as to enable decision making for evolving situations. The integration of these new applications on current flight decks will saturate the flight crew with a huge amount of information and interaction loads. Therefore, new interfaces and situational displays should emerge as enabling technologies in the future flight decks. The FAA PARC/CAST Flight Deck Automation Report [96] also emphasises that the new interfaces should be more understandable from the flight crew perspective, and should consider to human-centered design principles.

In this part of the work, novel visual decision support tools and interfaces, which incorporate next-generation synthetic vision and augmented reality-based visualization, is presented to support the flight crew. The 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 a 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; perception, comprehension and projection [83]. Progression of these layers, the level of Automation and the extent of SA does not indeed exhibit a simple 1-1 relation. For example, inappropriate levels of automation can impact SA with results such as automation complacency, automation mistrust,

Figure 3.1: B737 – 800 Flight Deck test platform in ITU CAL with experimental visual decision support tools for future ATM realm: Head up Display (HUD), Synthetic Vision Display (SVD) and 4D Operational Display (4DOD).

increased workload, and automation transparency [90]. Specifically, 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 [85]. 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 comprehensive evaluation of alternatives, and choices [86]. 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 [87]. 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 [89].

Figure 3.2: Synthetic Vision Display (SVD) and 4D Operational Display (4DOD) screens in the flight deck.

3.1 Next generation synthetic Vision Screens

Presented synthetic vision display includes two separate screens; 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 the 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 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, or pilot can choose to follow the trajectory manually with guidance of the tunnel-in-the-sky visualization on Synthetic Vision Display (SVD) and HUD.

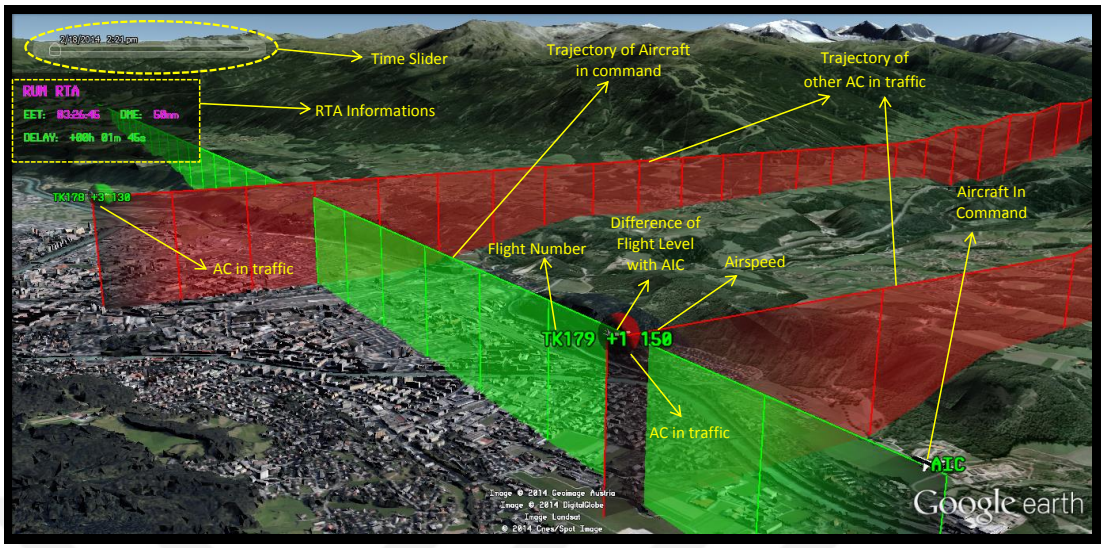

Figure 3.3: Definitions of the symbology in 4D Operational Display (4DOD).

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 time of arrival objectives, delays, estimated capacity of the airspace; and safety-related warnings such as conflict probability predictions. The display gives 3D visualization ability to the pilot as a 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 track pad and 3D navigator mouse; which both provide 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 choose to perform fast time simulation for entire or specific part of the flight. The Fugire 3.3 gives explanations 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 neutralized 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 modify existing trajectory by adding or removing fixes and then puts it on the ATC for acceptance.

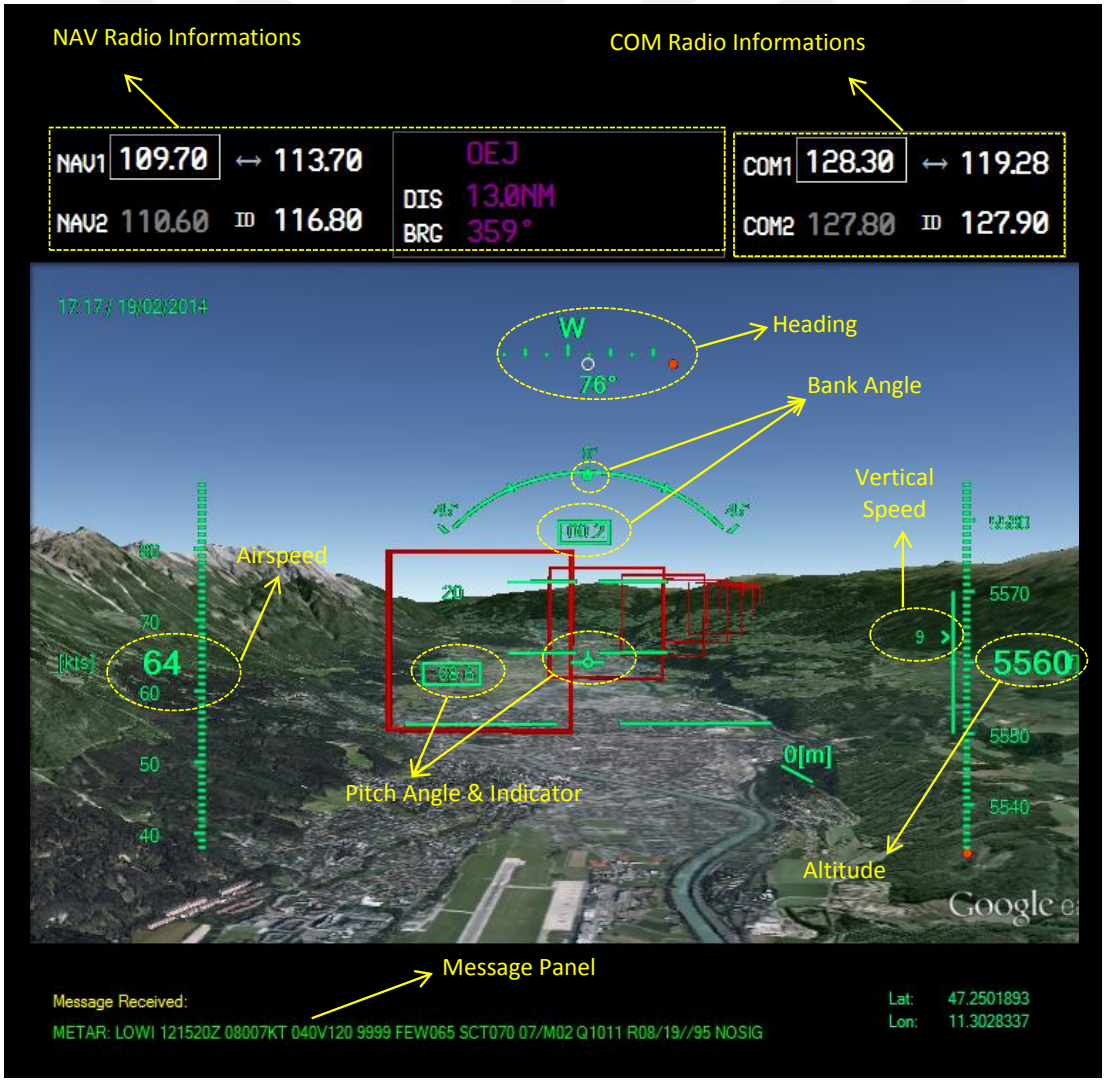

Figure 3.4: Definitions of the symbology in Synthetic Vision Display (SVD).

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 visualization also gives a continuous perception across the whole trajectory from surface operation to landing with glide slope. In addition to synthetic terrain visualization, It also enables to visualize 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 symbols in the Synthetic Vision Display (SVD) have been given in Figure 3.4.

3.2 Virtual Reality based Head-Up-Displays

The proposed structure of the Head-Up-Display (HUD) seen in reffig:transparentscreen 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, which all come from Flight Management System (FMS). The negotiated information at all phases (including land operations, take-off en-route and landing) is transformed into virtual tunnel visualization 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 on the corners of the tunnel frames provide the

Figure 3.5: Transparent Screen overlay for HUD augmented reality implementations.

pilot effective flight direction perception. The brief descriptions of the nominal HUD symbols can be seen in Figure 3.6.

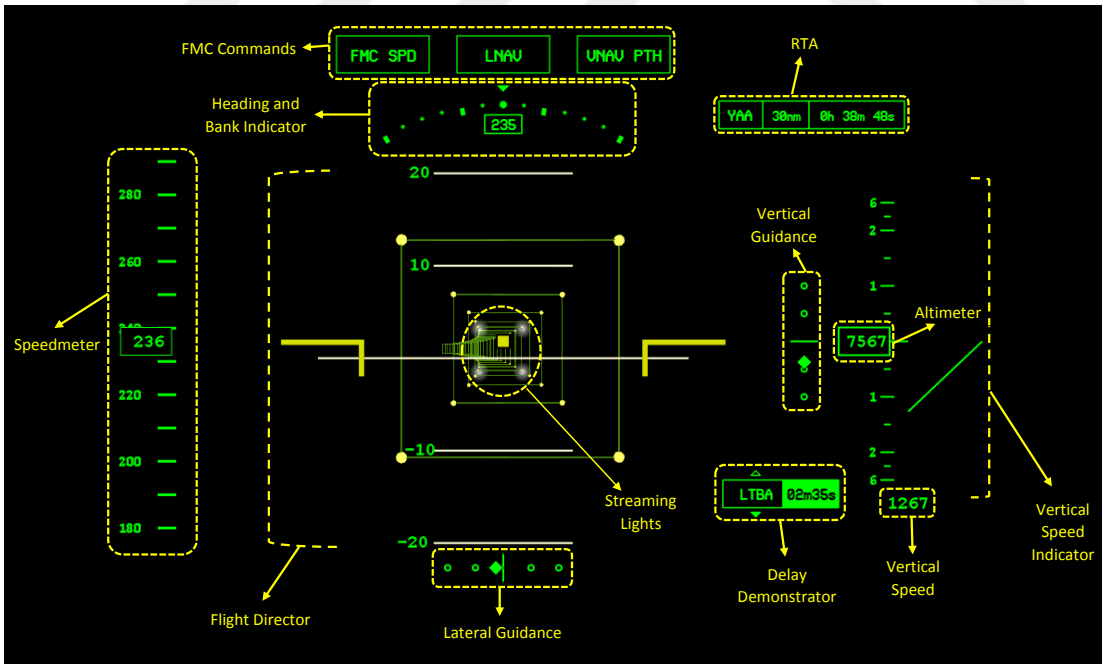

Figure 3.6: Definitions of the symbology in Head-Up-Display (HUD).

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 name, remaining distance and negotiated RTA. In addition to this, a coloured message box (i.e. green for positive and red for negative values) shows predicted delay times for the next waypoint. It also enables to demonstrate pop-up messages for the check lists according to related situations (e.g. engine start-up, emergency and required traffic and conflict avoidance messages, etc.)

4. TRAJECTORY COMPUTATION MODEL IN ATM CONTEXT

We can introduce the trajectory generation problem in two parts as it is defined in [54]. First part is the feasibility, which implies performable actions of the aircraft considering its dynamics such as limited maneuverability, performance limits, control input limits; and environmental constraints such that airspace limitations, operational constraints. Second part is the optimality, which considers certain costs in terms of the fuel or time.

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

$$\dot{x}(t) = f(x(t), u(t)), \quad x(0) = x_0, \tag{4.1}$$

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 predicted trajectory set and control input set for the reachable set of the surrounding aircraft. Let X_{obs} and X_{arr} represents the obstacle (static obstacles) region and arrival region respectively. Then we can define the conflict-free space depending on time (due to the 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 through a set of aircraft separation cylinders. So, a dynamically-feasible trajectory in X_{free} (we simply write X_{free} instead of $X_{free}(t)$) starts at x_{init} and ends in the arrival region X_{arr}. Formally;

Feasibility: The feasible trajectory generation problem can be defined as finding a feasible trajectory if one exists, report failure otherwise. 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 $X_{arr} \subset X$, a feasible trajectory is $x : [0, \tau] \to X_{free}$ 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 feasible trajectory with min cost. Formally;

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) $J(x^*) = min_{x \in \sum_{cl(X_{free})}} J(x)$,

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) $J(x_1|x_2) = J(x_1) + J(x_2)$.

Following section explains Aircraft Performance Model and Cost Efficient Trajectory Generation algorithm for local planing.

4.1 Aircraft Performance Model (APM) based on BADA 4

The following three degrees of freedom (3 DOF) equations of motion (presented in a similar fashion to [97] are considered sufficient to describe the aircraft dynamics in an air traffic management (ATM) context;

$$\dot{v_{TAS}} = \frac{T - D - W \sin \gamma_{TAS}}{m} - \dot{w_1} \quad \textbf{(4.2)}$$

$$\dot{\chi_{TAS}} = \frac{\frac{L \sin \mu_{TAS}}{m} + (\dot{w_3} \sin \mu_{TAS} - \dot{w_2} \cos \mu_{TAS})}{v_{TAS} \cos \gamma_{TAS}} \quad \textbf{(4.3)}$$

$$\dot{m} = -F \quad \textbf{(4.4)}$$

$$\dot{\lambda} = \frac{v_{TAS} \cos \gamma_{TAS} \sin \chi_{TAS} + w_2}{(N_c + h) \cos \varphi} \quad \textbf{(4.5)}$$

$$\dot{\varphi} = \frac{v_{TAS} \cos \gamma_{TAS} \cos \chi_{TAS} + w_1}{(M_c + h)} \quad \textbf{(4.6)}$$

$$\dot{h} = v_{TAS} \sin \gamma_{TAS}, \quad \textbf{(4.7)}$$

where $[v_{TAS}, \chi_{TAS}, m, \lambda, \varphi, h] \in X \subseteq \mathbb{R}^n$ denote the states of the aircraft and represent the true airspeed, true airspeed yaw, mass, latitude, longitude and altitude of aircraft respectively, and $[\gamma_{TAS}, \delta_T, \mu_{TAS}] \in U \subseteq \mathbb{R}^m$ are the control variables that represent the flight path angle, throttle parameter and aerodynamic bank angle respectively. W is the aircraft weight, D is the total drag, T is the total thrust, L is the total lift force and F is the fuel consumption rate. 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 the Wind Fixed System (WFS) axes.

Let ξ represent the bank angle with respect to the body fixed system (BFS) axes. For low angles of attack, BFS axes and WFS axes can be assumed to be aligned. Hence, the body bank angle is approximately equal to the wind axis bank angle; $\xi \approx \mu_{TAS}$. The dependence of the total lift L on the inputs U is given as:

$$L = \frac{W \cos \gamma_{TAS} - m\left(\dot{w}_3 \cos \mu_{TAS} + \dot{w}_2 \sin \mu_{TAS}\right)}{\cos \mu_{TAS}}. \tag{4.8}$$

The earth model is described by the vector $E = [\delta, \theta, g, w]$, where δ is the local pressure ratio, θ is the local temperature ratio, g is the local acceleration of gravity and w is the local wind speed vector. The equations for the drag force D and weight W are given as:

$$D = \frac{1}{2}\kappa p_0 \delta M^2 S C_D, \quad W = mg, \tag{4.9}$$

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. C_D is the drag coefficient, which is be defined as:

$$C_D = C_D(M, C_L, \Delta), \tag{4.10}$$

where $\Delta = [\delta_{HL}, \delta_{LG}, \delta_{SB}]$ is the configuration input set. 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 lift coefficient C_L is calculated as:

$$C_L = \frac{2L}{\kappa p_0 \delta M^2 S}. \tag{4.11}$$

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

$$T = W_{MTOW}\, \delta_T C_T \tag{4.12}$$

$$F = W_{MTOW} a_0 \delta_T \sqrt{\theta} C_F, \tag{4.13}$$

where $W_{MTOW} = m_{MTOW} g$ is the maximum take-off weight, a_0 is the speed of sound at sea level and $\delta_T \in [0,1]$ 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.

As it was mentioned in the introduction, we effectively utilize the BADA 4 dataset [98] in our APM development. The operational version of the BADA, namely BADA 3, does not include all the relevant physical dependencies, and provide little flexibility. Moreover, in this version of the dataset, dynamic calculation of forces acting on the aircraft, which are the functions of input variables used in APM, is not possible. Specifically, model specifications in BADA 3 only enable to evaluate maximum climb/take off and maximum cruise/descent thrust forces, and their nominal values such that they are not directly open to fuel and time cost efficiency calculations. The in-development version of BADA, i.e BADA 4, aims to meet advanced functional and precision requirements of the new ATM systems. For example, it provides a generalized thrust model as a function of the throttle parameter and Mach number. Such generalized expressions in BADA 4 allow us to develop advanced cost optimization procedures utilizing modal parametric definitions of the aircraft performance. We have rigorously studied on BADA 4 dataset, and integrated all the relevant functions into our 3 DOF equations of motion and cost-efficient trajectory calculations, which is essential to develop such effective cost minimization. Figure 4.1 demonstrates our parameter handling in trajectory generation relying on BADA 4.

Aircraft Performance Model (APM) involves details about the performance parameters of the aircraft including the 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; Aircraft Limitation Model (ALM) for identifying geometric, kinematic, dynamic and environmental operation limitations; Operation of Configuration Parameters Model (OPM) to define transition time for both the high-lift devices and

landing gear configurations [99]. The interactions between these different models can be seen in Figure 4.1.

The Aerodynamic Forces and Configurations Model, a part of the BADA 4 dataset, allows us to evaluate the lift and drag coefficients C_L and C_D. One way to obtain the lift coefficient C_L is to examine its relation with the angle of attack. Alternatively, C_L can be obtained directly from the lift force L. After the calculation of the lift coefficient C_L, the drag coefficient C_D can be evaluated by using the drag polar model. Due to the low angle of attack assumption, BADA 4 does not model the relation between the angle of attack and the lift coefficient C_L. Hence, we use the lift force L to calculate the lift coefficient C_L, as described in the Eq. 4.11. The drag coefficient C_D depends on the lift coefficient C_L, Mach number M and configuration inputs Δ. BADA 4 provides two different drag coefficient configurations. In the clean configuration, the configuration input set Δ is not active and C_D only depends on C_L and M. Following equations are valid for the clean configuration.

$$C_D = \left[C_0 + \left(C_2.{C_L}^2\right) + \left(C_6.{C_L}^6\right)\right], \tag{4.14}$$

where sc is a scaling coefficient and $C_i, i = 0,2,6$ are defined as;

$$C_0 = \sum_{k=0}^{4} \frac{d_{k+1}}{\left(1-M^2\right)^{\frac{k}{2}}} \tag{4.15}$$

$$C_2 = \sum_{k=0}^{4} \frac{d_{k+6}}{\left(1-M^2\right)^{\frac{3k}{2}}} \tag{4.16}$$

$$C_6 = d_{11} + \sum_{k=0}^{3} \frac{d_{k+12}}{\left(1-M^2\right)^{7+\frac{k}{2}}}. \tag{4.17}$$

$$\tag{4.18}$$

In the equations above $d_i, i = 1,\ldots,15$ are parameters defined in AFM part of BADA 4 [99]. In the Non-clean configuration, the drag coefficient C_D depends on the configuration input set Δ as well. In this case,

$$C_{D_{nonclean}} = \sum_{r=1}^{3} d_{r,\delta_{HL},\delta_{LG}} {C_L}^{r-1}, \tag{4.19}$$

where $d_{r,\delta_{HL},\delta_{LG}}$ is a parameter set defined in AFM [99]. In addition, if the speed brakes are applied, C_D is calculated as,

$$C_D = C_{D_{noSB}} + 0.03\delta_{SB}, \tag{4.20}$$

where $C_{D_{noSB}}$ is the drag coefficient calculated prior to applying speed breaks. The thrust coefficient C_T and fuel coefficient C_F are obtained from Propulsive Forces Model part of BADA 4. The thrust coefficient for a turbofan engine is evaluated from the throttle parameter and flight Mach number through the following expression given in [99];

$$C_T = \sum_{k=0}^{5} \left(\sum_{r=1}^{6} a_{r+6k} M^{r-1} \right) \delta_T^k. \tag{4.21}$$

Similarly, the fuel coefficient is evaluated through the following expression given in [99];

$$C_{F,gen} = \sum_{k=0}^{4} \left(\sum_{r=1}^{5} f_{r+5k} C_T^{r-1} \right) M^k, \tag{4.22}$$

where a_i and f_i are the parameters defined in BADA 4 for the corresponding aircraft. Note that, the thrust and fuel consumption model is valid only for a certain throttle value interval i.e. $0.4 \leq \delta_T \leq 1.0$. For the low throttle setting, i.e. $\delta_T < 0.4$, (descend mode), low-idle rating (LIDL) model is used for calculations.

To summarize, given the states $X(t_k)$ and inputs $U(t_k), \Delta(t_k)$ at time t_k, the derivatives in Eqs. 4.2-4.7 need be calculated to obtain the state configuration $X(t_{k+1})$ by integration. For the calculation of derivatives, first the lift force L is calculated using the Eq. 4.8. Once the lift L is calculated, the lift coefficient can be determined from the Eq. 4.11. Next, the drag coefficient is computed using the Eq. 4.14 or the Eq. 4.19, depending on whether the aircraft is in clean or non-clean configuration. Then the thrust and fuel coefficients are obtained from the Eqs. 4.21 and 4.22 respectively. Next, the drag D, the thrust T and the fuel consumption F are computed from Eqs. 4.9, 4.12 and 4.13 respectively. Once these forces are obtained, the state derivatives in Eqs. 4.2-4.7 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. We describe these operational limitations in the next subsection.

4.1.1 Aircraft Limitation Model

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

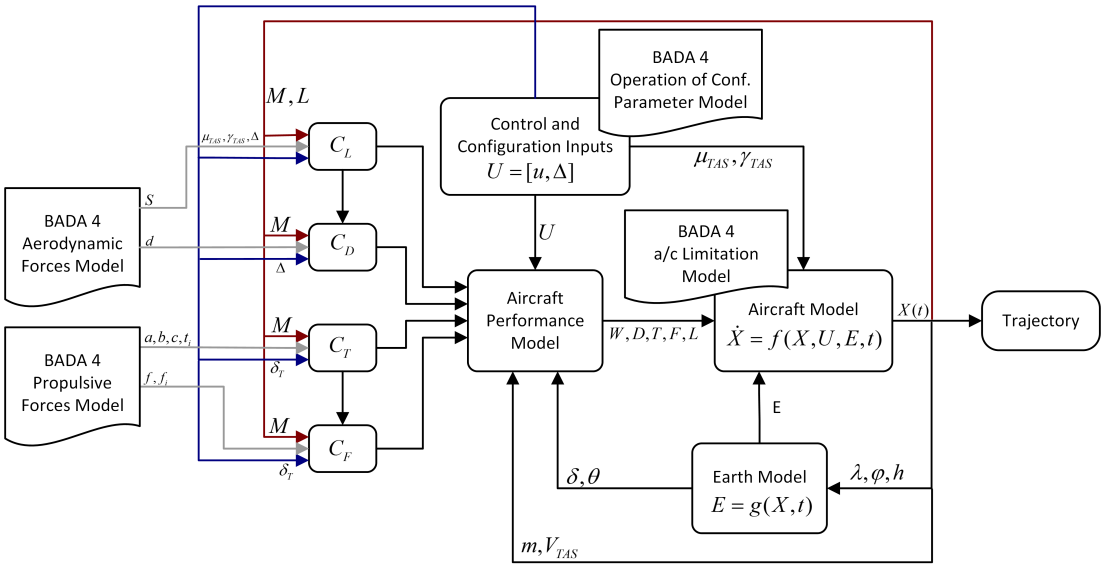

Figure 4.1: Aircraft state trajectory computation process based on BADA 4

models. These constraints are given as follows;

$$max[0, h_{as_{min}}] \leq h_{Hp} \leq min[h_{Hp_{max}}(\delta_{HL}), h_{as_{max}}] \quad \textbf{(4.23)}$$

$$M \leq M_{M_0}(\delta_{LG}) \quad \textbf{(4.24)}$$

$$\delta_{T_{min}} \leq \delta_T \leq \delta_{T_{max}} \quad \textbf{(4.25)}$$

$$V_{CASstall}(\delta_{HL}, \delta_{LG}) \leq V_{CAS} \leq V_{M_0}(\delta_{HL}, \delta_{LG}) \quad \textbf{(4.26)}$$

$$V_{CAS} \leq 250 \text{ kts for } h \leq 10,000 \text{ ft} \quad \textbf{(4.27)}$$

$$0 \leq C_L \leq C_{Lmax}(M, \delta_{HL}, \delta_{LG}) \quad \textbf{(4.28)}$$

$$m_{min} \leq m \leq m_{MTOW} \quad \textbf{(4.29)}$$

$$n_{min}(\delta_{HL}) \leq n \leq n_{max}(\delta_{HL}), \quad \textbf{(4.30)}$$

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, $h_{as_{max}}$ is the maximum altitude allowed in the airspace, $M_{M_0}(\delta_{LG})$ is the maximum operating Mach number depending on the landing gear configuration, $\delta_{T_{min}}$ is the minimum throttle setting, $\delta_{T_{max}}$ 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, $C_{Lmax}(M, \delta_{HL}, \delta_{LG})$ is the maximum lift coefficient depending on the depending on the Mach number, high-lift device and landing gear configuration, m_{min} is the minimum operating mass, m_{MTOW} is the maximum take-off mass, $n_{min}(\delta_{HL})$ is the minimum loading factor

depending on the high-lift device configuration and $n_{max}(\delta_{HL})$ is the maximum loading factor 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 or not. 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 in [99] as follows:

$$V_{TAS} = \sqrt{\frac{2p}{\mu\rho}\left\{\left(1+\frac{p_0}{p}\left[\left(1+\frac{\mu\rho_0}{2p_0}{V_{CAS}}^2\right)^{\frac{1}{\mu}}-1\right]\right)^{\mu}-1\right\}}. \tag{4.31}$$

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 positions. 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.

4.1.2 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;

$$J = c_f\delta m + c_t\tau + \sum_n c_{r_i}\delta d_i, \tag{4.32}$$

where c_f is the per lb fuel cost in cents, δm is the fuel consumption, 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 for services provided to airspace users by the air navigation service providers (ANSPs). The zone dependent charge notion

is defined in [100], such as $c_{r_i} = pt_i$ where t_i is the airspace dependent unit rate per kilometer and p is the weight factor, i.e. $p = \sqrt{m_{MTOW}/50}$. In Eq. 4.32, d_i denotes the great circle distance flown over the charging zones and expressed in kilometers. The entry and exit points to the zones are outlined as filed in the last 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 tactical trajectory optimization procedure. For the 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 [101], 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, e.g. [102] indicates that CI range is $0-500$ for the B737-800 and $0-999$ for B777. For instance, if the pilot inputs zero cost index through the FMC interface, the performance behavior yields max range airspeed, and the fuel consumption remains minimum by ignoring the time cost. If the cost index is maximized, the time of flight will be minimum with maximum climb/descent velocity and cruise Mach number by neglecting the fuel cost. Hence, this index parameter strongly forms the objectives of the flight operation where some researchers focus on benefiting from dynamic recalibration of CI parameter by considering evolving operational situations [9]. In summary, we take into account following tactical cost-to-go definition for a given trajectory segment;

$$J = c_f(\delta m + CI\tau) + C_r, \tag{4.33}$$

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

5. COLLISION AVOIDANCE PROBLEM: SHORT TERM

The immediate collision avoidance problem can be formulated as a perfect-information zero-sum differential game [103]. In that case, evader will be the ownship tries to maximise collision time, and other threads will be the pursuers try to minimise collision time as assuming the worst escape maneuvers are chosen by the intruders (worst case). If one can find a positive collision time at each attempt (as the "one-shot" plans) upon solving this problem, she guarantees that the collision will never occur. Let $x_e(t) \in X_e \subseteq \mathbb{R}^{n_e}$ be the state vector, and $u_e(t) \in U_e \subseteq \mathbb{R}^{m_e}$ be the control input vector of the evader (ownship) such that $n_e, m_e \in \mathbb{N}$ and the state $x_{0,e} \in X_e$ denotes the initial state. Similarly, Let $x_p(t) = \{x_{p,1}, x_{p,2}, ..., x_{p,k}\} \in X_p \subseteq \mathbb{R}^{k \times n_p}$ be the state vector, and $u_e(t) = \{u_{p,1}, u_{p,2}, ..., u_{p,k}\} \in U_e \subseteq \mathbb{R}^{k \times m_e}$ be the control input vector of the pursuer (multiple threats) such that $n_p, m_p, k \in \mathbb{N}$ and the state $x_{0,p} = \{x_{0,p,1}, x_{0,p,2}, ..., x_{0,p,k}\} \in X_e$ refers the initial states of the pursuers.

Recall the general form of the time-invariant dynamics of the aircraft, so the entire dynamics of the system will be:

$$\dot{x}(t) = \begin{bmatrix} \dot{x_e}(t) \\ \dot{x}_{p,1}(t) \\ \vdots \\ \dot{x}_{p,k}(t) \end{bmatrix} = \begin{bmatrix} f(x_e(t), u_e(t)) \\ f(x_{p,1}(t), u_{p,1}(t)) \\ \vdots \\ f(x_{p,k}(t), u_{p,k}(t)) \end{bmatrix}, \quad x(0) = \begin{bmatrix} x_{0,e} \\ x_{0,p,1} \\ \vdots \\ x_{0,p,k} \end{bmatrix}, \quad \forall t \in \mathbb{R}_{\geq 0}. \tag{5.1}$$

The states and the control inputs of the system in separable structure should satisfy the constraints coming from their dynamics and the environment. It can defined as:

$$C_\alpha(x_\alpha(t), u_\alpha(t)) \geq 0, \text{ where } \alpha \in \{e, p_1, p_2 \ldots, p_k\}. \tag{5.2}$$

The collision occurs if the state of the system falls into terminate condition that only present in time T, that is;

$$T = inf\{t \in \mathbb{R}_{\geq 0} : ||x_e(t,1:3) - x_{p,i}(t,1:3)|| \leq \varepsilon\},\ for\ any\ i \in \{1,2,\ldots,k\},\ and\ \varepsilon > 0. \tag{5.3}$$

This simply tells the collision (with ε distance) in 3D workspace with lat, long and altitude. The collision avoidance aims to keep the system at $T = +\infty$ which indicates "no collision". The objective of the evader aircraft is to maximise the T. Without knowing which action is to be taken by the other aircraft, it is assumed that the worst case can be happened such that the pursuers may choose the wrong (worst) avoidance maneuver. Thus, it is assumed that the pursuers minimize (unintentionally) the T. In that sense, this is a zero-sum game such that the only one objective will be;

$$J(u_e, u_{p,i}) = \begin{cases} T, & if\ ||x_e(T,1:3) - x_{p,i}(T,1:3)|| \leq \varepsilon \\ +\infty & \text{otherwise} \end{cases}. \tag{5.4}$$

If the the players (evader and pursuers) performs their feedback saddle-point strategies, the solution of the problem will be;

$$T^* = max_{u_e}\left\{min_{u_{p,i}}\left\{J(u_e, u_{p,i})\right\}\right\}\ for\ any\ i \in \{1,2,\ldots,k\}. \tag{5.5}$$

In order to solve this equation, it is need to be solving the Hamilton-Jacobi-Isaacs equation [104] which is not tractable except simple forms [105]. It will be considerably harder when multiple pursuer involved from a computational point of view. This prohibits real-time applications such as collision avoidance problem. To overcome this difficulty, sampling based algorithms [41, 106] has been proposed enabling real-time implementation.

5.1 Sampling-based Threat Avoidance algorithm: CA

In order to generate feasible approximate solution to this problem, we have introduced a sampling based procedure integrating sampling based motion planning algorithms. Specifically, objective of the evader (ownship) is to built as many as avoidance action and to select one maximising collision time. As introduced above, problem is rather to build real-time implementable action without considering maneuver cost. The approach holds following two assumption to relax the problem;

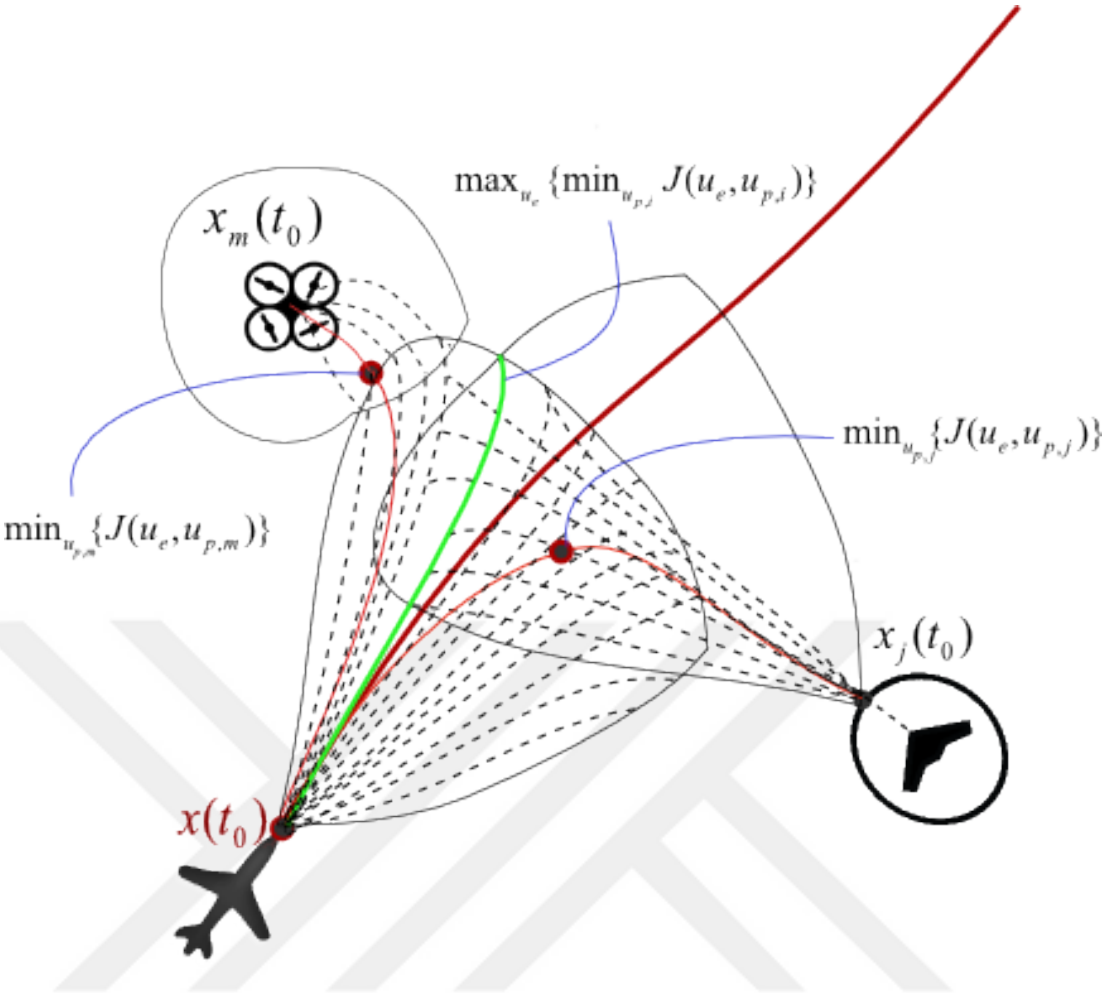

Figure 5.1: Collision check and resolution with max-min policy.

(i) Evader uses finite set of action combining fixed control inputs, that is $A_i : [0, T_h] \rightarrow U_e \subseteq \mathbb{R}^{m_e}$ for any $i = (1, 2, \ldots, \rho)$ such that $\rho < +\infty$,

(ii) The game has a fixed time horizon, and repeated when it passed. That is $+\infty > T_h > t_{comp}$.

(iii) The "one shot" action computation time t_{comp} is almost instantaneous and fixed.

First assumption relaxes the problem to make it tractable by using a finite set of control vectors. This approach also approximates TCAS as it gives fixed climb/descent rate advisories [11]. In simulations, we have also add fixed control input sets enabling horizontal maneuvers with coordinated turns. In the second assumption, the time window T_h ensures that the evader has enough time to generate its action and execute it. It bounds the size of the problem from the upper for the sake of the implementability. Last assumption holds that the computation time has an upper value for certain number of pursuer.

To generate many evasive action for the evader, we follow the control-driven Probabilistic Road Map (PRM) algorithm [107]. This algorithm spatially explores the space induced by the control inputs. This open-loop approach removes the burden of the solving computationally expensive trajectory generation problem as the other kinodynamic planners do (i.e. [56, 108]). It also allows to add uncertainty factors into the control inputs and environmental inputs such as wind speed. The approach provides an open-loop plan as it generates a set of trajectory projections by considering current situation. The main structure of the algorithm [107] is given in the following pseudo-code:

Algorithm 1: *Control-driven PRM*

$V \leftarrow (z_{init}, 0, +\infty)$, $E \leftarrow \emptyset$, $i \leftarrow 0$
$G \leftarrow (V, E)$
while $i < N$ **do**
    $(z_{rand}, t_{node}) \leftarrow SelectNode(G)$
    $(u_{new}, \tau) \leftarrow SelectControl(U)$
    $(x_{new}) \leftarrow Propagate(z_{rand}, u_{new}, \tau)$
    if $Conflict\_Free(x_{new})$ **then**
        $z_{new} \leftarrow x_{new}(\tau_{new})$ and $V \leftarrow V \cup \{z_{new}, (t_{node} + \tau), +\infty\}$
        $E \leftarrow E \cup \{x_{new}\}$
        $i \leftarrow i + 1$

In this algorithm, 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 $(v_i, v_{i+1}) \in E \ \forall i \geq 1$. *SelectNode()* function returns a randomly selected vertex $v = \{z_{node}, t_{node}, T_{col}\}$ with its state z_{node} and time stamp t_{node} from the graph. T_{col} denotes the collision time which is initially set to $+\infty$.

Let Ω be a finite set of control inputs sampled from $U \subseteq \mathbb{R}^m$. *SelectControl* function returns a control input vector u and terminal time τ such that $u(\tau) \in \Omega \subseteq U$ and $\tau \in [t_{comp}, T_h]$. The finite control input library Ω discretises input space and reduces size of dimension for the sake of the real-time implementability.

Let $z_{init} \in X$ be a state and $Propagate(z_{init}, u, \tau)$ function returns a trajectory segment x_{new} connecting z_{init} and z_{new} such that $x_{new}(0) = z_{init}$ and $x_{new}(\tau) = z_{new}$.

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 $X_{free}(t)$:

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

Generally, the algorithm extends the randomly selected node through the randomly selected control inputs (Lines 2-6). Whenever it finds a conflict free trajectories, add them into the graph $G = (V, E)$ (Lines 7-9).

To complete the problem introduced above, we need to approximate pursuers' actions as well. This part of the problem relies on to find minimum collision times for the threats (evaders).

Algorithm 2: *Sampling-based Threat Avoidance – CA*

1 $V_e \leftarrow (z_{init}, 0, +\infty)$, $E_e \leftarrow \emptyset$, $i \leftarrow 0$, $m \leftarrow 0$
2 $G_e \leftarrow (V_e, E_e)$
3 **while** $i < N$ **do**
4 $(z_{rand}, t_{node}) \leftarrow SelectNode(G_e)$
5 $(u_{new}, \tau) \leftarrow SelectControl(U_e)$
6 $(x_{new}) \leftarrow Propagate(z_{rand}, u_{new}, \tau)$
7 **if** $Conflict\_Free(x_{new})$ **then**
8 $z_{new} \leftarrow x_{new}(\tau_{new})$ and $V_e \leftarrow V_e \cup \{z_{new}, (t_{node} + \tau), +\infty\}$
9 $E_e \leftarrow E_e \cup \{x_{new}\}$ and $i \leftarrow i + 1$
10 $V_{p,j} \leftarrow (z_{p,j_{init}}, 0, +\infty)$, $E_{p,j} \leftarrow \emptyset$, $i \leftarrow 0$
11 $G_{p,j} \leftarrow (V_{p,j}, E_{p,j})$
12 **foreach** *pursuer* $j = 1, 2, \ldots, k$ **do**
13 **while** $i < M$ **do**
14 $(z_{rand}, t_{node}) \leftarrow SelectNode(G_{p,j})$
15 $(u_{new}, \tau) \leftarrow SelectControl(U_{p,j})$
16 $(x_{new}) \leftarrow Propagate(z_{rand}, u_{new}, \tau)$
17 **if** $Conflict\_Free(x_{new})$ **then**
18 $z_{new} \leftarrow x_{new}(\tau_{new})$ and $V_{p,j} \leftarrow V_{p,j} \cup \{z_{new}, (t_{node} + \tau), +\infty\}$
19 $E_{p,j} \leftarrow E_{p,j} \cup \{x_{new}\}$ and $i \leftarrow i + 1$
20 **while** $m < |V_e|$ **do**
21 $(z_{rand}, t_{node}, T_{col}) \leftarrow V_e(m)$ and $n \leftarrow 0$
22 **while** $n < |V_{p,j}|$ **do**
23 **if** $Reachable(V_{p,j}(n), z_{rand}, t_{node})$ **then**
24 $t_{col} \leftarrow t_{node}$
25 $V_{p,j}(n) \leftarrow \{z_{new}, t_{node}, t_{col}\}$ and $V_e(m) \leftarrow \{z_{new}, t_{node}, t_{col}\}$
26 $n \leftarrow n + 1$
27 $m \leftarrow m + 1$
28 $\mathscr{A} \leftarrow Sort(G_e)$

Table 5.1: TCAS resolution advisory commands

RA Type	Resolution Advisory	Target Rate(ft/min)
Positive	Climb/Descent	1500 to 2000
Positive	Crossing Climb/Descent	1500 to 2000
Positive	Maintain Climb/Descent	1500 to 4400
Negative	Do not Climb/Descent	~ 0
Negative	Do not Climb/Descent more than 500 ft/min	< 500
Negative	Do not Climb/Descent more than 1000 ft/min	< 1000
Negative	Do not Climb/Descent more than 1500 ft/min	< 1500

Generally *Threat Avoidance Algorithm* first propagates initial states of the evader(ownship) and the intruders(threats) using kinodynamic PRM implementation (Lines 1-9 and Lines 10-19 respectively). Then, algorithms tries to extend the graphs of the pursuers towards the maneuver set of the evader. Whenever the algorithm (Lines 20-27) finds that any vertex in the evader graph is *reachable* by any pursuer, writes the collision time on to it. Otherwise, collision time remains $+\infty$. *Sort()* function sorts the graph of the evader with regards to the collision times (Line 28) as the maneuver with the maximum collision time is preferred first. This algorithm provides "one-shot" evasive maneuvers that ensure maximising collision time, and is repeated for every T_h with updated information for the states.

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

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. *Reachable*(z_1, z_2, τ) function checks that whether z_2 is reachable from z_1 in time τ via trajectory x, such that;

(i) $\mathscr{R}_\varepsilon(z_1) = \{z_2 \in Z \mid \exists x \in X_{z,z'}$ such that $x(t) \in \mathscr{B}_\varepsilon(z_1)\ \forall t \in [0,\tau]\}$
(ii) $x(0) = z_1$ and $x(\tau) = z_2$.

Note that *Reachable* function can find many trajectories with many required control input set. To reduce the size of the problem, reachability check is done within the finite set of control input $\Omega_{p,j} \subseteq U_{p.j}$. The search withing finite maneuver set is

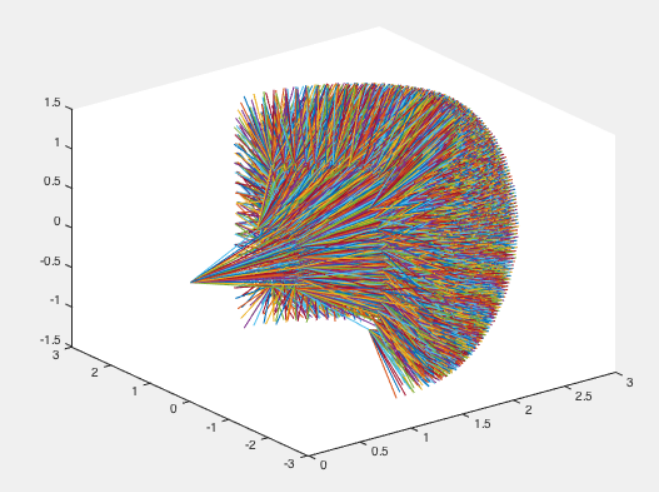

Figure 5.2: Trajectory projection for Dubins aircraft with with fixed airspeed, heading and climb/descent rates for fixed time steps.

.

common practice in the collision avoidance problem to reduce computational effort. In order to give an example, Table 5.1 summarizes the resolution advisories with bounded climb/descent rates of TCAS v7.1 that is the latest operational version.

In this study, by considering future needs and capabilities of the aircraft, we have incorporated also horizontal component of the trajectory, which typically provides 3D+time (4D) maneuvers. Figure 5.2 shows 4D envelope projection of a Dubins aircraft [109] with fixed airspeed, heading and climb/descent rates for fixed time steps.

5.2 Simulations

This section presents the simulation results on several scenarios for collision avoidance. In these scenarios, we consider standard collision avoidance for one-to-one and multiple aircraft. For the sake of simplicity, we have used fixed speed change rates (in reality depending on aircraft type), fixed bank angle rates and fixed climb/descent rates. Three scenarios have been shown in this section.

In the first scenario, in Figure 5.3, the aircraft detects an intruder aircraft coming from north-east. In Figure 5.3, black trajectories represents min time collision trajectories for the intruder. The evader aircraft generates many evasive maneuvers (green), which are maximizes collision time, by using sampling-based threat avoidance algorithm (CA). Note that the Figure 5.3 shows one-shot solution where the CA algorithm replans at each update.

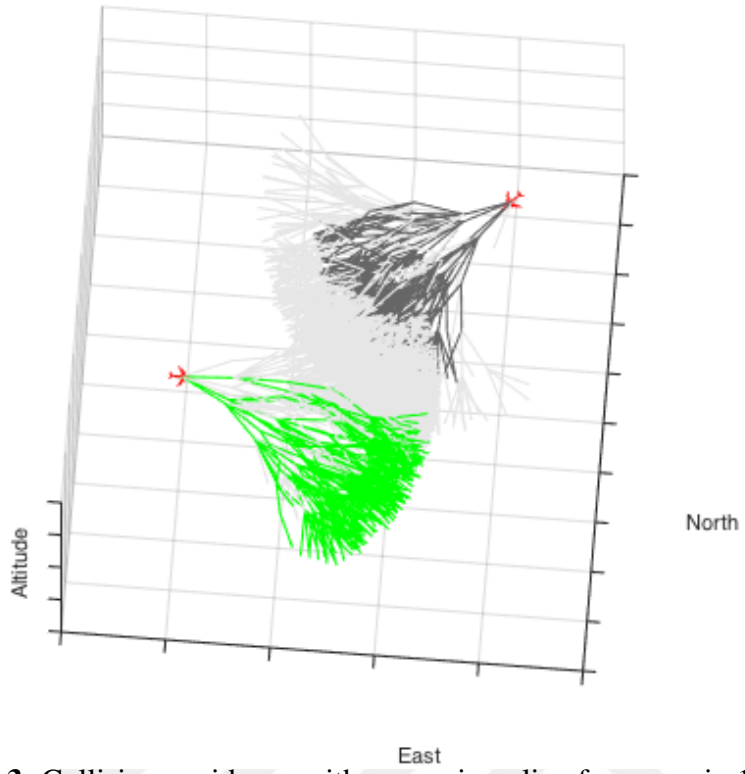

Figure 5.3: Collision avoidance with max-min policy for scenario 1.

.

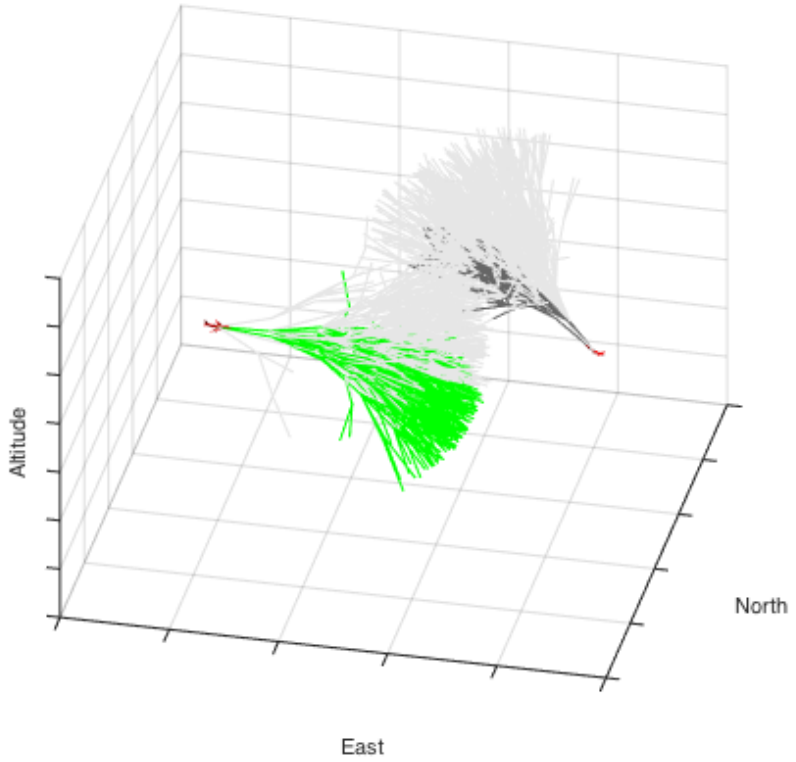

Figure 5.4: Collision avoidance with max-min policy for scenario 2.

.

In the second scenario, in Figure 5.4, the aircraft detects a climbing intruder aircraft coming from north-east. The evader aircraft generates many evasive maneuvers, mostly descending and heading towards north, which are maximizes collision time, by using sampling-based threat avoidance algorithm (CA). Note that, due to the fixed time steps, CA algorithm evaluates more than one action that maximizing collision in the simulations.

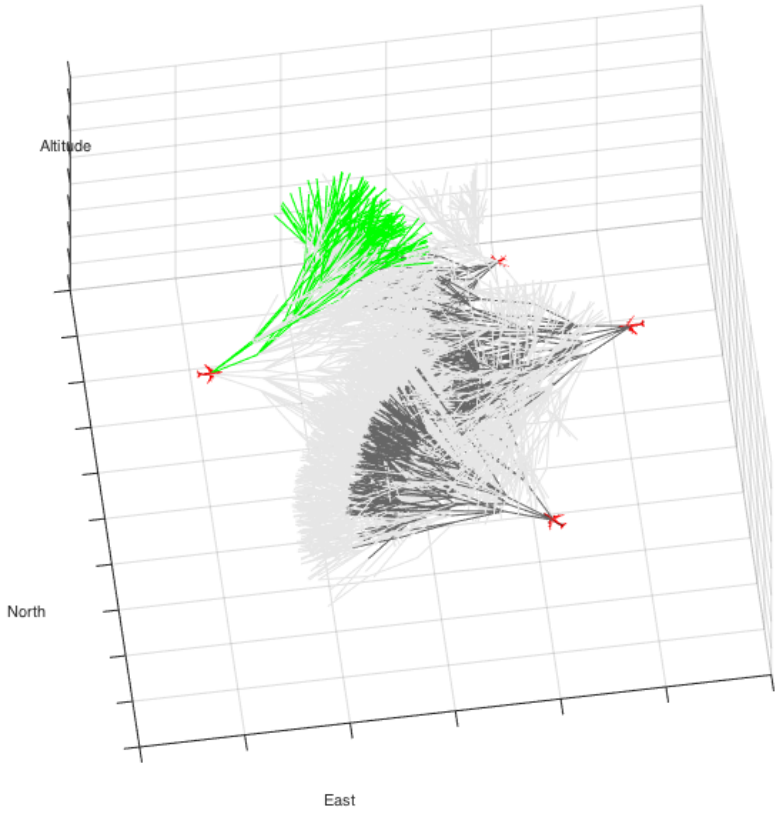

Figure 5.5: Collision avoidance with max-min policy for scenario 3.

.

Multiple-threat case is also considered in scenario 3. The solution to a multiple-threat case including climbing and descending intruders is shown in Figure 5.5. Note that the solution sequences involves many climb and turn actions, which are typically 3D maneuvers.

6. OPTIMAL 4D TRAJECTORY PLANNING

The Conflict Detection and Resolution (CDR) procedure is strongly related to the reachability notion due to the dynamic constraints of the aircraft. We suppose that ground systems are amenable to predict future trajectories based on computationally complex calculations with approximate performance models, while the airborne systems are obliged to frequent information sharing. It is expected that next generation on-board FMSs will exchange estimated parametric uncertainties for each state through the available data links.

6.1 Conflict Monitoring

Let $\chi_j(t)$ 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 onboard FMS due to information availability.

Suppose that Trajectory Generation Infrastructure of the ground segment has an 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:

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

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:

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

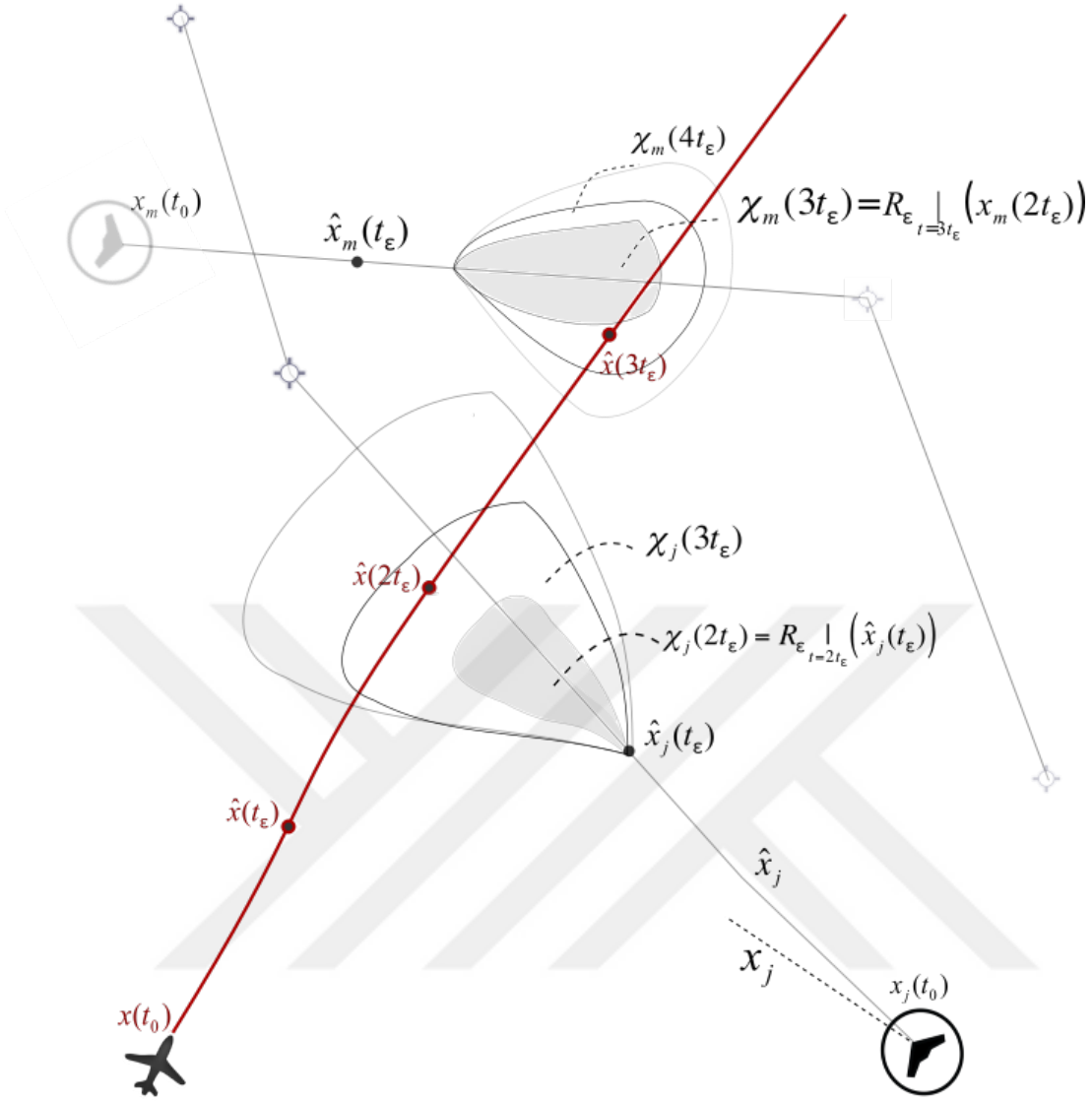

Figure 6.1: Ground perspective: conflict monitoring with flight intent and reachable sets associated with different performance models.

The best practice for 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 linear trajectories of the ownship and all other aircraft respectively, which is driven from current velocities. 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 depicts conflict check operation from the ground for non-homogenous airspace that involves many types of aircraft.

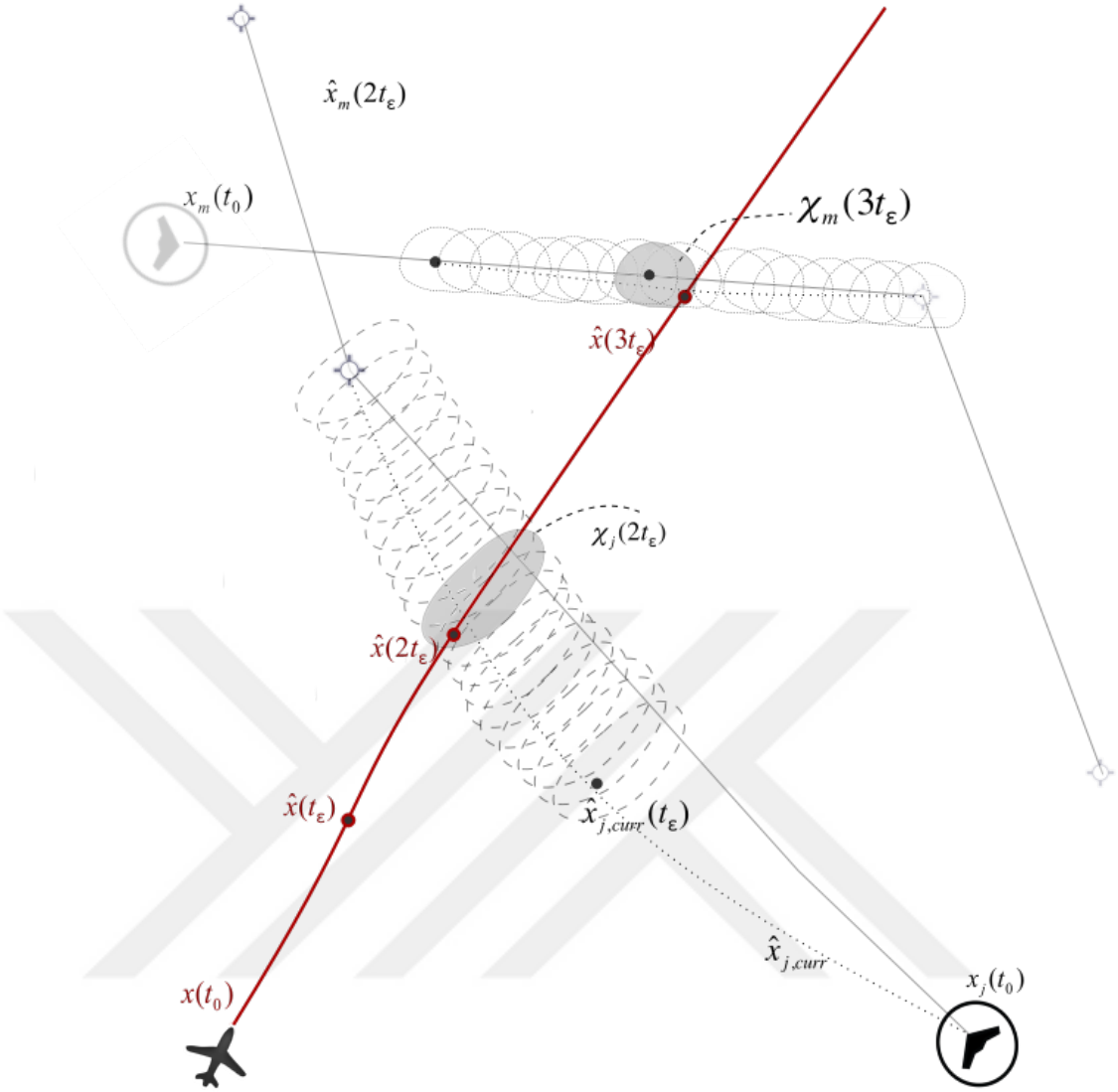

Figure 6.2: Airborne perspective: conflict monitoring with flight intent exchange and ADS-B.

.

Suppose that the APMs for the other aircraft are not available to the airborne Trajectory Generation Infrastructure of FMS, e.g. highly-heterogeneous airspace. While the flight intent provides future projection of the traffic, ADS-B/In (direct communications from surrounding aircraft) capability 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, i.e., horizontal position accuracy (NACp), horizontal velocity accuracy (NACv), and vertical accuracy (GVA). Their operational limits are given numerically in [110]. 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 $\hat{x_j}$ 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 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 for loss of separation for both the ground segment and onboard *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 avoidance and its recovery trajectory generation procedure.

6.1.1 Sampling-based Conflict Resolution: CR

Collision detection is a persistent process working in both ground systems and airborne FMS. The required intervention depends on required action time after detecting a potential collision. The ground system with human operators mostly monitors potential collisions and interprets the potential solution to modify operative flight plans with minimum deviation. If the required time for action is not enough, immediate action may be needed without human operator's involvement. It is envisioned that, in such case, the automated collision avoidance employs to fully control the aircraft or simply guide the pilot to follow required action.

In order to generate feasible trajectories for a given aircraft model, we chose to implement RRT^* algorithm prosed in [54] for its asymptotic optimality property in addition to the many other preferred properties that sampling-based strategies have. Asymptotic optimality implies 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 3: *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, *GE_Sample()* function returns a sample state by utilising Cross-Entropy sampling which its details are given in the following section. 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 to goal. In order

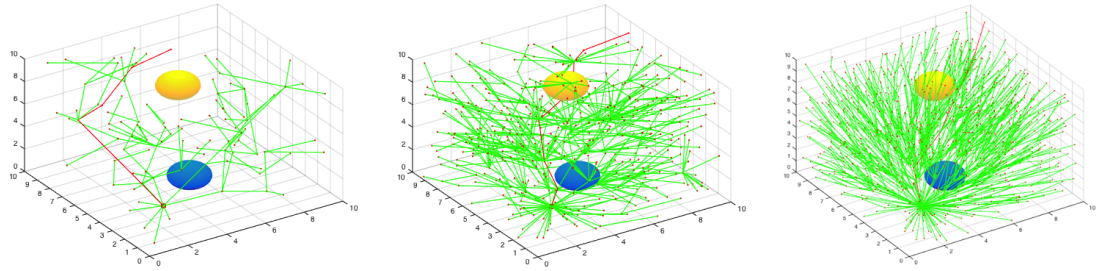

Figure 6.3: *RRT** algorithm solutions are shown after 100, 600 and 1200 vertices generation respectively.

.

to immediately build a non-empty set, the elite set π initially involves flight plan trajectory (FI) which is not conflict-free. The initial probability distribution in which cross-entropy sampling will employ is set over the quantized states of the reported flight plan. The *Quantize* function simply quantizes flight plan (FI) into set of states. The idea behind of this initialization is that the minimum cost trajectory most likely to be around the compromised flight plan trajectory. Note that the success of the proposed algorithm does not strongly depend on this assumption where the Cross-entropy sampling disperse over the search space until finding a new elite trajectory by adding variance smoothing. In addition, we have slightly modify the *RRT** algorithm to integrate Cross-Entropy sampling. *Near* function in Line 6 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 [111]) 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 4: *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 $\quad V' \leftarrow V' \cup \{z_{new}\}$
7 $\quad z_{min} \leftarrow z_{nearest}$
8 $\quad Z_{near} \leftarrow Near(G, z_{new}, |V|)$
9 $\quad$ **forall the** $z_{near} \in Z_{near}$ **do**
10 $\quad\quad (x_{near}, u_{near}, \tau_{near}) \leftarrow Generate(z_{near}, z_{new})$
11 $\quad\quad$ **if** $Conflict\_Free(x_{near})$ *and* $x_{near}(\tau_{near}) = z_{new}$ **then**
12 $\quad\quad\quad c' \leftarrow J(z_{near}) + J(x_{near})$
13 $\quad\quad\quad$ **if** $c' < J(z_{new})$ **then**
14 $\quad\quad\quad\quad z_{min} \leftarrow z_{near}$
15 $\quad E' \leftarrow E' \cup \{(z_{min}, z_{new})\}$
16 $\quad$ **forall the** $z_{near} \in Z_{near} \backslash \{z_{min}\}$ **do**
17 $\quad\quad (x_{near}, u_{near}, \tau_{near}) \leftarrow Generate(z_{new}, z_{near})$
18 $\quad\quad$ **if** $Conflict\_Free(x_{near})$ *and* $J(z_{near}) > J(z_{new}) + J(x_{near})$ *and* $x_{near}(T_{near}) = z_{near}$ **then**
19 $\quad\quad\quad z_{parent} \leftarrow Parent(z_{near})$
20 $\quad\quad\quad E' \leftarrow E' \backslash \{(z_{parent}, z_{near})\}$
21 $\quad\quad\quad E' \leftarrow E' \cup \{(z_{new}, z_{near})\}$
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 $x(t) : [0, \tau] \rightarrow X$ 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 $t \in [0, \tau]$.

This is where the trajectory planning frame uses local cost efficient trajectory segment generation depending on Aircraft Performance Model (APM) such that their details are explicitly given in the previous section. The optimality guarantee of the algorithm strongly depend on holding "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 (Lines 2-6). However, it generates a path

segment to the x_{new} from the vertex within X_{near} set, incurring minimum cost (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 (Lines 16-21). For instance, an example run of RRT^* is given in Figure 6.3 to demonstrate its asymptotic convergence as the sampling (i.e. through Halton sequence) increases.

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 $\alpha||z_1 - z_2||^n$

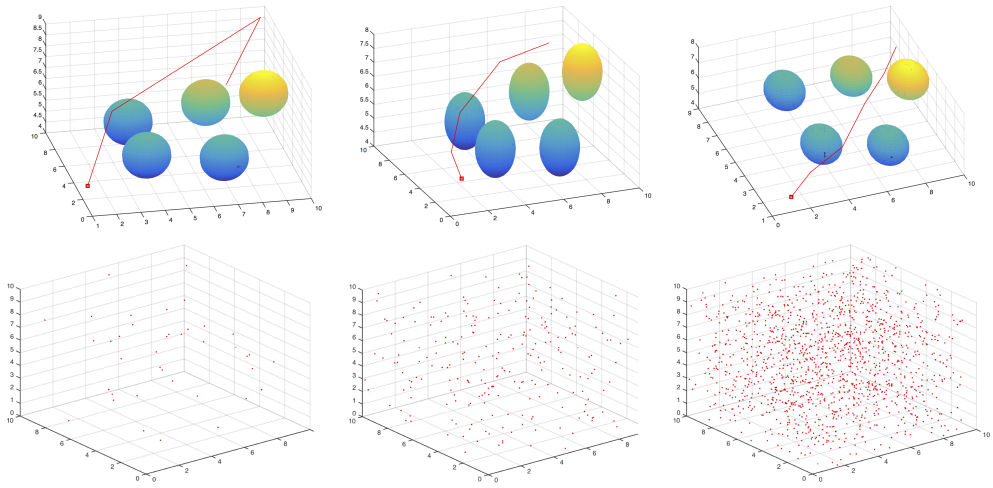

Figure 6.4: Pseudo-random sampling and asymptotic convergence in RRT^* with 40, 120 and 400 vertices.

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, e.g. 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:

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

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.

Proof – There exists a set of trajectories q such that $q(\varepsilon)$ is ε-collision-free $\forall \varepsilon > 0$. Consider an approximate trajectory $q(\varepsilon_i)$ and put overlapping balls with radius ε such that their centers are separated by a distance $l^{\frac{1}{p}}$, where $l = \beta \varepsilon_i$ for some constant β and p. Consider two balls centered at z_1 and z_2. *Local Controllability* assumption guarantees that ε-reachable set of z_1 has positive volume and has positive probability of being sampled. Conflict-free Trajectories assumption ensures that there exists a constant α such that a ball radius $\alpha||z_1 - z_2||^p = \alpha l$ centered at z_2 is reachable from z_1.

By the Lipschitz continuity, there exists a constant $\kappa > 1$ such that any state inside a ball of radius $\alpha l/2\kappa$ can be connected to any state inside the ball cantered at z_2 [112]. Hence note that, the volume of each such ball is $V_i := \xi_n \left(\frac{\alpha l}{2\kappa}\right)^n$ where ξ_n is the volume of a unit sphere in n dimension, and number of such balls would be most $\frac{L_i}{l^{1/p}} = \frac{L_i}{(\beta\varepsilon_i)^{1/p}}$ where L_i is the length of the trajectory.

If a sampling strategy generates any sample inside all such balls of radius $\alpha l/2\kappa$ by iteration i, where i is large enough and such that $V_i = \gamma\frac{logi}{i}$, then the RRT^* would necessarily connect the samples to build a trajectory that has cost is close to the cost of the optimal trajectory $q(\varepsilon)$. Consider an event A_i RRT^* fails to sample at least one such ball, $lim_{i\to\infty}\mathbb{P}(A_i)$ can be upper bounded by the number of balls that cover the trajectory $q(\varepsilon_i)$ times the probability that one ball does not include any vertex, such that;

$$\mathbb{P}(A_i) \leq \frac{L_i}{\alpha\gamma_1 l^{1/p}} \left(1 - \gamma_2 \frac{logi}{i}\right)^i \tag{6.6}$$

where γ_1 *and* γ_2 are constants. For large enough γ_1 *and* γ_2, it implies that $\sum_{i=0}^{\infty}\mathbb{P}(A_i) < \infty$. Then Borel-Cantelli lemma ensures that the event A_i can occur only finitely often with probability one. Hence, the RRT^* generates an approximately optimal trajectory infinitely often with probability one.

6.1.2 Importance Sampling with Cross-Entropy

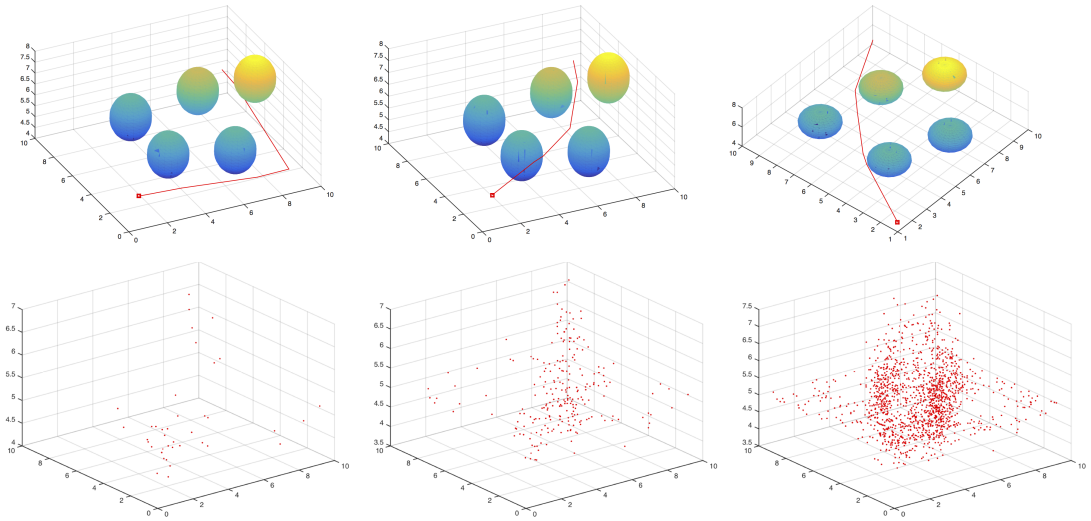

Figure 6.5: Importance sampling strategy of CE with 40, 120 and 400 vertices in RRT^*.

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 $Z = z_0, \ldots, z_{N-1}$ 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 of the P is given as;

$$D_N(P;\mathfrak{B}) = \sup_{B\in\mathfrak{B}} \left| \frac{|P \cap R|}{N} - \lambda_n(B) \right|. \tag{6.7}$$

The discrepancy notion can be defined as a quantitive measure for irregularity of sampling distribution, on the other words, it measures the largest volume estimation error [52]. 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 $Z = z_0, \ldots, z_{N-1} \in Z$, the dispersion notion is defined as;

$$d_N(P;Z) = \sup_{z\in Z} \min_{0\leq n<N} d(z, z_n), \tag{6.8}$$

where d denotes any distance metric, e.g. Euclidian 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 (the proof can be found in [113];

$$d_N(P;Z) \leq D_N(P;\mathfrak{B}). \tag{6.9}$$

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 optimization that bounds 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 [113]. Note that, this is the best possible point set dispersion, which depends on the dimension and constant number of points. In the motion planning problems, since we don't know in advance how much sample is needed, we prefer to use the lowest possible discrepancy and, so the lowest dispersion

in selection of random sampling strategy. For instance, Halton sequence sampling (pseudo-random) within the RRT^* algorithm is given in Figure 6.4. The figure demonstrates the convergence to optimal trajectory and sample distribution of RRT^* with 40, 120 and 400 number of vertices respectively.

The performance of sampling can be further improved with importance sampling such that the sampling distribution over the state space incrementally concentrates on promising regions. The sampling problem turns into solving stochastic optimization problem to find a proper sample set leading the algorithm to the minimum cost trajectories. For this purpose, we have integrated cross-entropy (CE) method. CE is an adaptive algorithm first introduced in [114] to estimate probabilities of rare events through the variance minimization. Cross-entropy uses an iterative procedure that first generates a set of sample from a specified distribution, then updates parameters associated with the distribution. This procedure continues until sample distribution approaches a delta function. In cross-entropy sampling integration in to the planning algorithm, we closely follow [114].

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;

$$\ell = \mathbb{E}[H(Z)] = \int_z H(z) f(z;v) dz \tag{6.10}$$

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;

$$\ell = \mathbb{P}_v(J(z) \leq \gamma) = \mathbb{E}_v[I_{\{J(z)\leq\gamma\}}] \tag{6.11}$$

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 too small subset of the entire space, namely rare-event. The alternative way to use importance-sampling, generates sample from a probability density function g defined on $\mathfrak{Z}$. Then estimator $\hat{\ell}$ becomes;

$$\hat{\ell} = \frac{1}{N}\sum_{i=1}^{N} I_{\{J(z)\leq\gamma\}}\frac{f(Z_i;\nu)}{g(Z_i)} \tag{6.12}$$

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

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

By applying this density in Eq. 6.12, we get;

$$\ell = I_{\{J(z)\leq\gamma\}}\frac{f(Z_i;\nu)}{g^*(Z_i)}, \quad \forall i \tag{6.14}$$

As seen in Eq. 6.14, g^* depends on ℓ, which is unknown as well. By choosing g in probability density $f(.;\nu)$, the problem turns into determining optimal parameter ν, such that distance between g^* and $f(.;\nu)$ should be minimal. The Kullback-Leibler (KL) distance between the two densities g and h, which is cross-entropy (CE) between g and h, is defined as;

$$\mathfrak{D}(g,h) = \mathbb{E}_g[ln\frac{g(Z)}{h(Z)}] = \int_z g(z)\ln g(z)dz - \int_z g(z)\ln h(z)dz \tag{6.15}$$

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

$$\underset{\nu}{\text{argmax}} \int g^*(z)\ln f(z;\nu)dz \tag{6.16}$$

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

$$\underset{\nu}{\text{argmax}}\,\mathfrak{D}(\nu) = \underset{\nu}{\text{argmax}}\,\mathbb{E}[I_{\{J(z)\leq\gamma\}}\ln f(Z,\nu)] \tag{6.17}$$

Numerical estimation of ν^* can be obtain, by solving following stochastic problem;

$$\hat{\nu}^* = \underset{\nu}{\text{argmax}}\,\frac{1}{N}\sum_{i=1}^{N} I_{\{J(z)\leq\gamma\}}\ln f(Z_i,\nu) \tag{6.18}$$

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

Cross-entropy optimization then becomes to find minimum cost value, i.e. $\gamma^* = \min_{z\in\mathfrak{Z}} J(z)$ associated to trajectories. If γ is very close to γ^*, then $f(z;\nu^*)$ accumulates

its probability mass around to x^* such that it approaches a delta distribution. On the selection of probabilistic densities, a Gaussian Mixture Model (GMM) [115] is preferred due to its ability to form smooth approximations of arbitrarily shaped sample distributions. For model parameter estimation, 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 [67], it is empirically addressed that the more than four components does not improves the solution. It is still open to study to find proper selection of k. CE Sampling strategy in this work follows closely [67] by generalizing in the context of the given problem.

Algorithm 5: $CE\_Sampling()$

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 $f(z;\nu^*) \leftarrow GMMFit(Z)$
6 $\nu^* \leftarrow Smoothing(\nu^*)$
7 **repeat**
8 $\quad z_{rand} \sim Sample\ f(z;\nu^*)$
9 **until** $Conflict\_Free(z)$
10 **return** z_{rand}

In this pseudo code, 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} (Line3). The elite set involves that the vertices with cost values lover than γ_ρ. In order to built 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 ν_{t-1} to ν_t , such that:

$$\hat{\nu}_t = \alpha\tilde{\nu}_t + (1-\alpha)\hat{\nu}_{t-1} \tag{6.19}$$

where $\tilde{\nu}_t$ is the parameter found in Eq. 6.17. $\alpha \in \mathbb{R}$ is the arbitrary smoothing parameter with $0.7 < \alpha \leq 1$, where $\alpha = 1$ indicates no-smoothing. Sample routine

picks a conflict-free sample from the constructed distribution and returns (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 while 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 in the Figure 6.6 and Figure 6.7 for an example 10x10 box environment. The results are obtained with repeated experiments, standard deviation in convergence to an optimal path as the number of sample increasing is shown with their error bars. As seen in the plot, the sampling algorithm with CE benefits from importance sampling as it requires a smaller number of nodes to converge to the optimal solution. The average computational efforts to generate a certain number of nodes are also analyzed. The required computational effort in importance sampling, as it is expected, increases faster than the pseudo-random sampling in the trade off the ability to converge to the optimal solution faster.

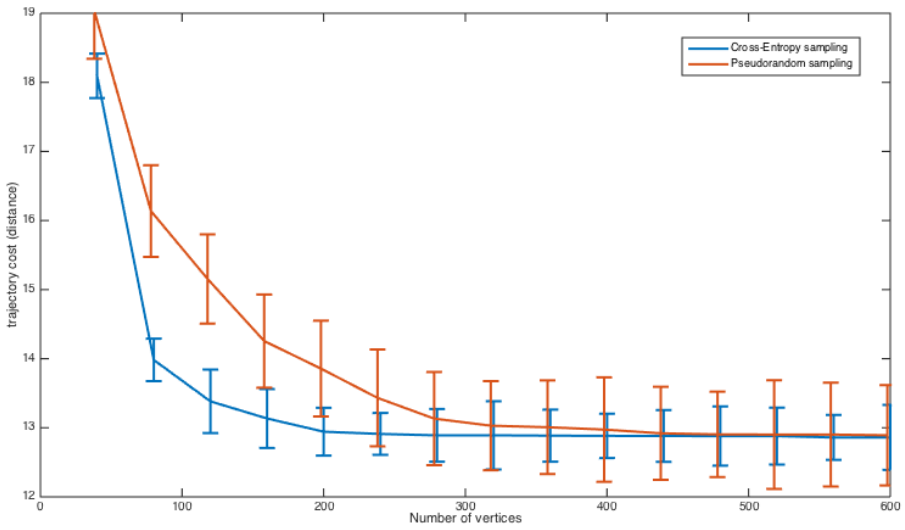

Figure 6.6: Trajectory cost convergence with the number of vertices in pseudo-random sampling and CE sampling.

To give a definition for the Halton sequence, it uses van der Corput sequences of n different bases, where n is the dimension [116]. Let $\{p_1, p_2, \ldots, p_n\}$ be n relative prime integers. The ith sample for the n dimension in the Halton sequence is:

$$\{r_{p1}(i), r_{p1}(i), \ldots r_{pn}(i)\} \quad i = 0, 1, 2, \ldots, \tag{6.20}$$

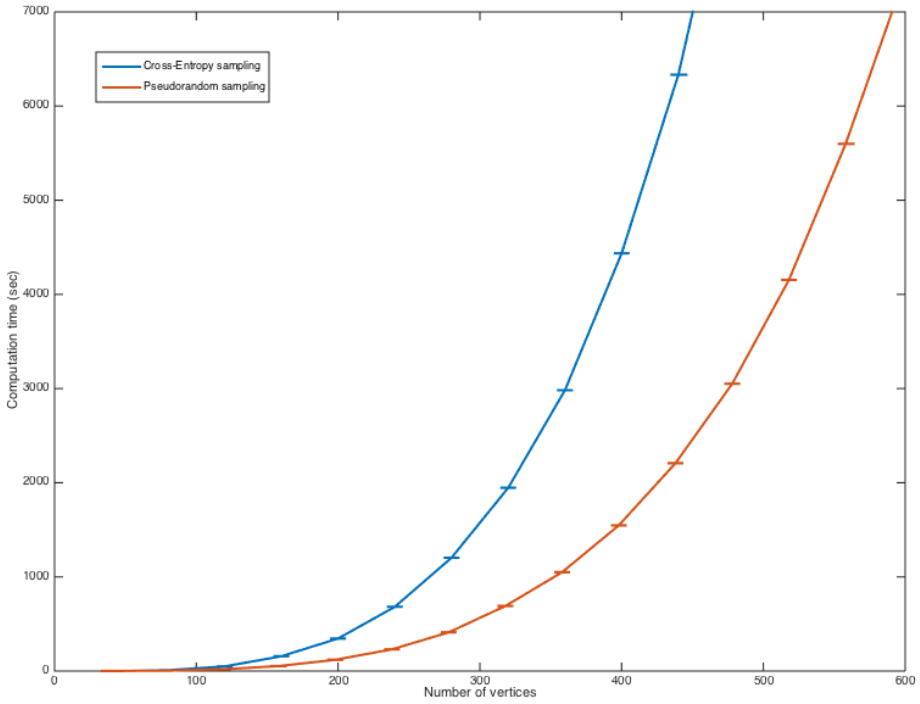

Figure 6.7: Computational effort with the number of vertices in pseudo-random sampling and CE sampling.

where each digit $a_{k,j} \in \{0,1,\ldots,p_k\}$ of base p_k are constructed from $i \in \mathbb{N}$, i.e. $i = \sum_{j=0}^{\infty} a_{k,j} p_k^j$. Hence, the elements of the generated sequence are defined as:

$$r_{pk}(i) = \sum_{j=0}^{\infty} = a_{k,j} b_k^{-j-1} \in [0,1]. \tag{6.21}$$

The Halton sequence replaces the standard random sample generation routines to benefit from its lower discrepancy and repeatability (with the same initial conditions) for software validation and certification purposes.

The result is a trajectory with state and effective time interval representations, which may be updated by dynamic collision check, that is sent to FMS to control the aircraft. That is:

$$\pi(t_0 : t_{end}) = \{(x_0, \tau_0), (x_1, \tau_1), \ldots, (x_{end}, \tau_{end})\}. \tag{6.22}$$

where $t_0 = \tau$, $\tau_{end} \leq t_{end} - t_\varepsilon$ and surely $t_{current} < t_0 - t_{min\_action}$. This final notion ensures that solution path begins with at least the minimum required time to perform a safe action before the first detected collision.

6.2 Simulations

This section presents the simulation results on several scenarios. In these scenarios, we consider standard separation problems with multiple aircraft. Trajectory projections up to ten minutes are determined through the reported speeds, which are updated at ~ 1 Hz in a simulation computer. For the sake of simplicity, we have used point-mass model for the intruder aircraft with fixed speed and fixed heading profiles. The model also enables to add wind disturbance, which provides additive uncertainty. Regarding the separation minima, we have applied horizontal radar separation in en-route airspace, which is 5 nmi, and 2,000 ft vertical separation. These minima boundaries are shown as cylinders around the intruder aircraft. On the other hand, we have ignored the semicircular (or hemispheric) rule, which applies east/west track split at certain flight levels. We rather aim to emulate future airspace, which most likely allows self-separation operations and enables self-regulated flight level to increase the capacity. Three scenarios have been demonstrated in this section.

In the first scenario, shown in Figure 6.9, potential conflict with two aircraft coming from different directions is resolved. The conflict detection and resolution (CDR) algorithm generates a trajectory tracking climb, descent and descent, climb templates. Note that, the maneuver sequence also involves a lateral component, which is seen in the bank angle profile. Note that, the entire maneuver sequence takes approximately 430 seconds. The CAS - Mach profile, Altitude - Bank angle profiles of the aircraft are demonstrated in Figure 6.9.

In the second scenario (in Figure 6.12), similar to scenario 1, potential conflict with two aircraft coming from different directions is detected, and this time CDR algorithm generates cascade climb maneuvers. The maneuver sequence also involves several lateral corrections with bank angle.

Third scenario (in Figure 6.15)simulates dense en-route airspace where many aircraft come from different directions, moreover, some perform climb and descent actions. In

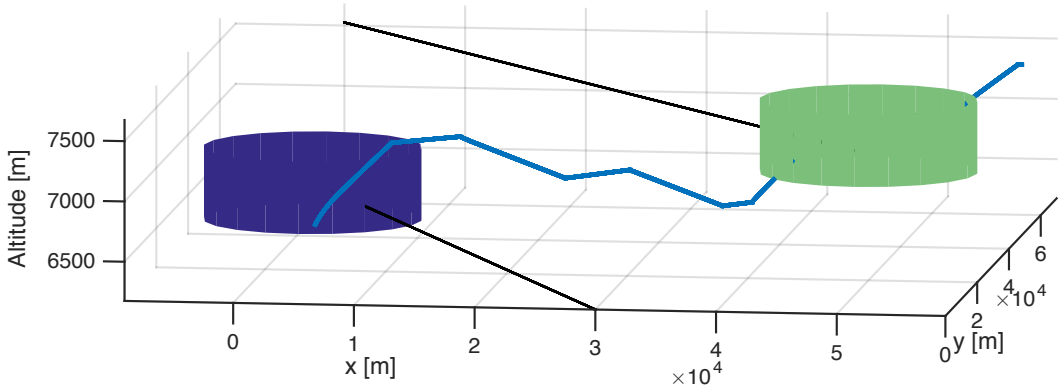

Figure 6.8: Conflict resolution trajectory for the scenario 1.

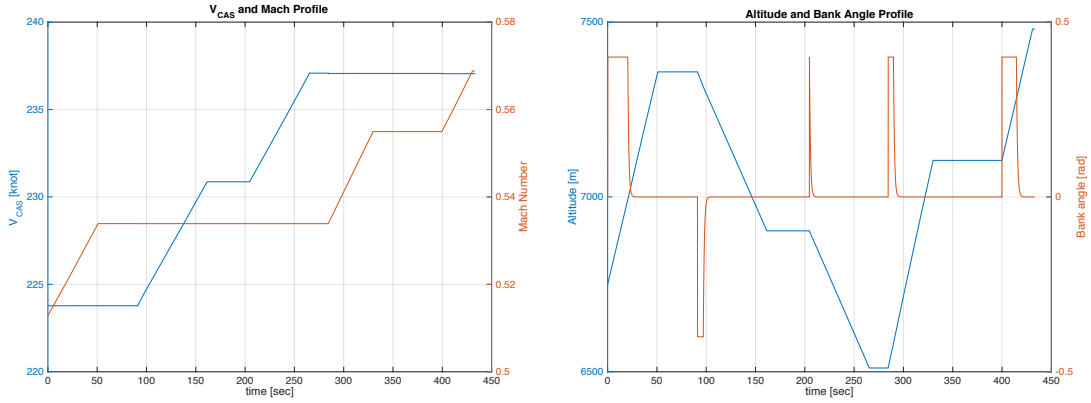

Figure 6.9: Conflict resolution trajectory with CAS-Mach and Altitude-Bank angle profile to the first scenario.

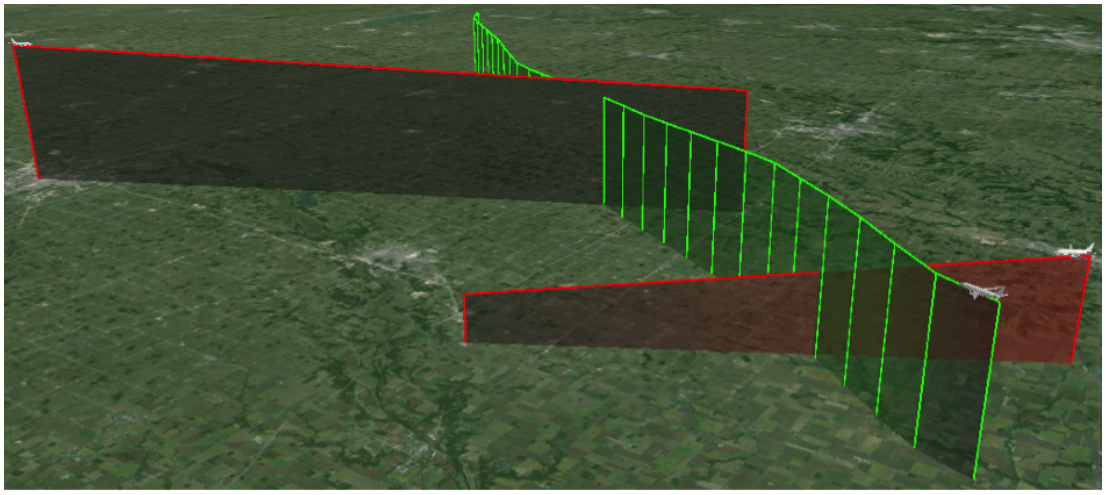

Figure 6.10: Conflict resolution trajectory in 4DOD screen for the first scenario.

this case, the CDR algorithm generates a maneuver sequence mostly involving lateral actions. At last, trajectory involves climb and accelerated cruise action in order to meet reference cruise Mach number.

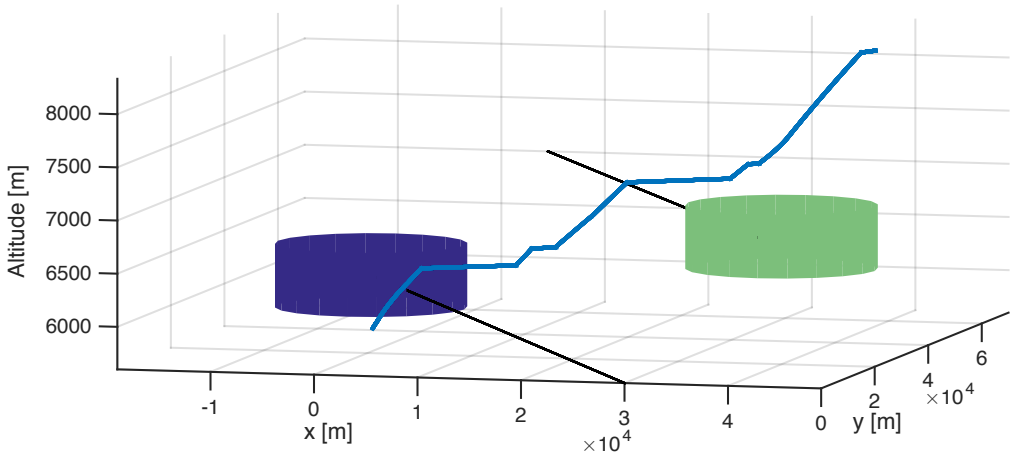

Figure 6.11: Conflict resolution trajectory for the scenario 2.

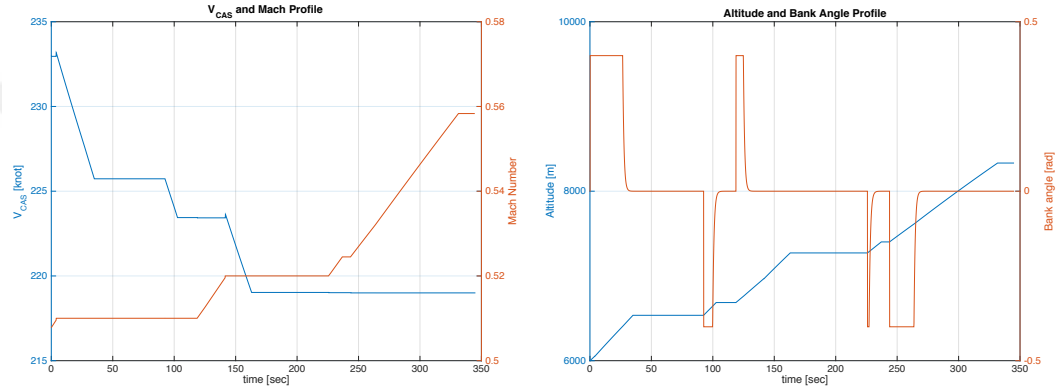

Figure 6.12: Conflict resolution trajectory with CAS-Mach and Altitude-Bank angle profile to the second scenario.

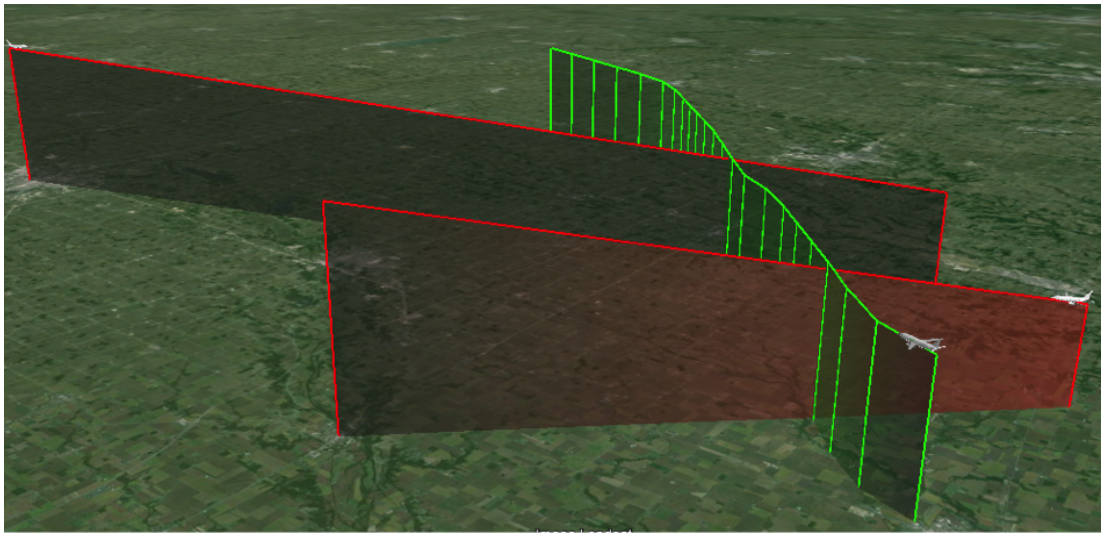

Figure 6.13: Conflict resolution trajectory in 4DOD screen for the second scenario.

Figure 6.10 and 6.13 gives examples where the solutions are demonstrated in 4D Operational Display (4DOD), which is a virtual pilot decision support display and explained in the following section.

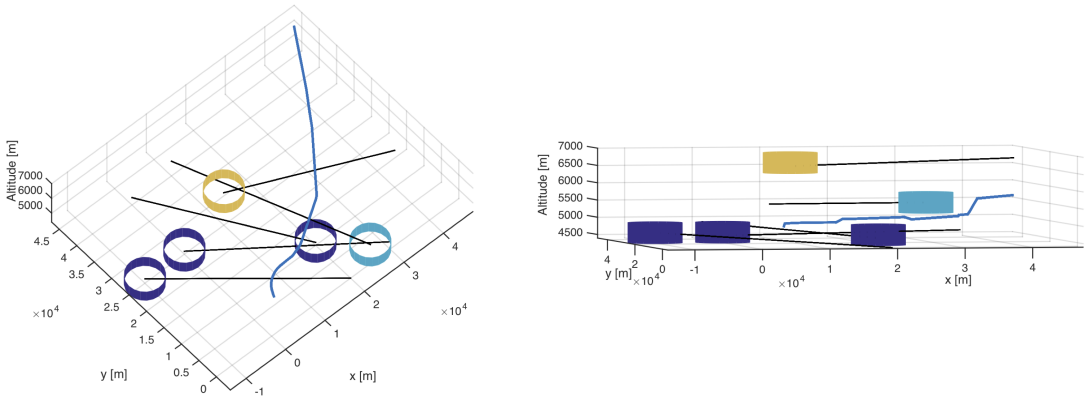

Figure 6.14: Conflict resolution trajectory for the scenario 3.

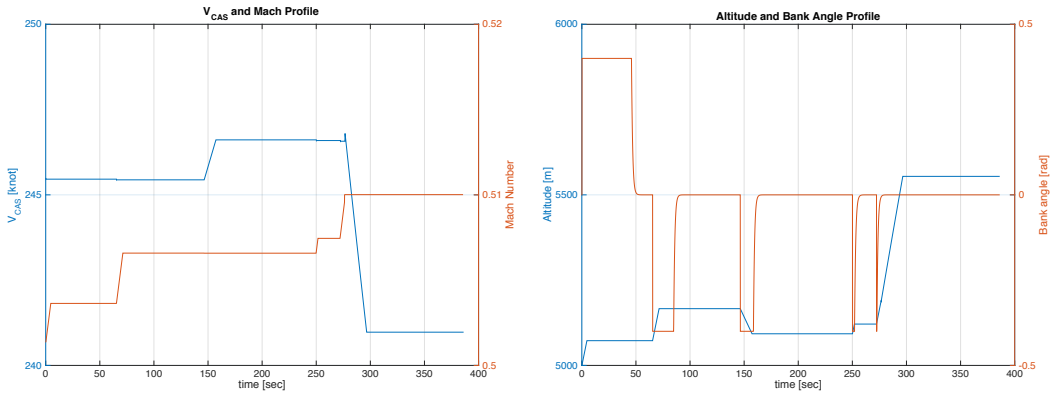

Figure 6.15: Conflict resolution trajectory with CAS-Mach and Altitude-Bank angle profile to the third scenario.

7. CONCLUSIONS

In this thesis, we have structured theoretical framework of the in-tactical trajectory generation problem of an aircraft, which equipped with automated tools can achieve airborne self-separation and collision avoidance. The proposed framework can also easily extend to the strategic planning problems. The architecture involves provably optimal sampling based algorithm (RRT^*) with embedded cross-entropy strategy, and advanced performance model relies on BADA 4.

As a local planner, we have utilized the flight templates with cruise, climb and descent templates. Each template employs an approximate trajectory optimization method specific to itself. This local trajectory generation algorithm provides a control input sequence for feasible trajectories, which is structured as compatible with the formal language of the current flight management systems (FMS). Thus, we envisioned to extend implementation of the proposed method to negotiation-based separation assurance through a data link.

In *Collision Avoidance* part, we have integrated game theoretical approach into the sampling based algorithm that approximating the solution of the multi-thread pursuer problem. The control-driven based random tree structure allowed us to explore potential action to choose the select maximizing collision time within the fixed-time horizon. This is a persistent procedure providing "one-shot" plans to fly and the it updated itself open new information (i.e positional sharing via ADS-B) arriving in each fixed-time window. Whenever the algorithm finds a plan maximizing the collision time for each time intervals, it obviously ensures the maximizing collision time for the entire flight operation.

In 4D trajectory planning, the sampling algorithm, namely RRT^*, instinctively embeds the stochastic nature of the effects which are inherent in air traffic realm, e.g. unpredictable weather conditions and uncertain intended action. It provably guarantees asymptotic optimality under certain conditions while maintaining the same probabilistic completeness and computational efficiency. Its local search routine

utilizes local trajectory generation algorithm, which employs flight templates. We have replaced the standard random (or quasi-random) sampling generation routine with the cross-entropy method in order to benefit from its importance sampling strategy based on the stochastic optimization. We have given our empirical analyzes such that CE rapidly converges to the optimal sampling set with a small number of node in the search graph. We have demonstrated performance of the developed procedure with the realistic simulations, which are given in the previous section.

Moreover, a multi-mode flight deck structure is presented through the three add-on modules. The Collaborative Mid-term Planning module involves collaborative flight operation functions such as intent negotiation and intent sharing implementations performed in the mid-term scale. In this mode, the required response time permits to maintain time consuming collaborative decision making processes. Therefore, this module aims to synchronize information level with the ground systems and other aircraft through the information sharing and intent negotiation processes. The Decentralized Short Term Planning module incorporates probabilistic methods and monitors probabilities of potential short-term collisions. By considering required response time, if it decides the immediate response is needed, the autonomous system (which is specific to this module) takes over the flight control to solve the situation. Through probabilistic search, it embeds uncertainty factors in both aircraft positions (obtained from air-to-air data link) and pilot actions. These two modules build a multi-level safety structure in the flight deck, where if the short term collision avoidance system issues an alert, it means that collaborative separation assurance process has failed before. In the mid-term planning mode, the flight crew is provided with the 2D and 3D information visualization through both the Augmented Reality Based Head-Up Display and the Synthetic Vision Displays using recent technological advances in visualization and interaction tools. During the intent negotiation, the Synthetic Vision Displays also allows to the flight crew to interact with the system at every level of the ongoing process.

REFERENCES

[1] **Kuchar, J.E. and Drumm, A.C.** (2007). The traffic alert and collision avoidance system, Lincoln Laboratory Journal, *16–2*(2), 277-289.

[2] **Weber, R.**, **Blom, H. and Cásek, P.** (2008). Safe, airborne self-separation operations in tomorrow's airspace, *The 26th International System Safety Conference*, Vancouver, Canada: 20-22 October

[3] **FAA** (2006). Aircraft Trajectory Prediction Errors: Including a Summary of Error Sources and Data (Version 0.2), FAA/Eurocontrol Action Plan 16 Common Trajectory Prediction Capabilities.

[4] **Kochenderfer, M.J.**, **Chryssanthacopoulos, J.P.**, **Kaelbling, L.P. and Lozano-Perez, T.** (2010). Model-Based Optimization of Airborne Collision Avoidance Logic, *Massachusetts Inst. of Technology, Lincoln Lab., Rept. ATC-360*, Cambridge, MA.

[5] **Lee, R. and Wolpert, D.**, (2012). Game Theoretic Modeling of Pilot Behavior during Mid-Air Encounters, In T. Guy, M. Kárný and D. Wolpert (Eds.), *Decision Making with Imperfect Decision Makers*, Springer Berlin Heidelberg, pp.75–111.

[6] **Arthur, J.J.**, **Prinzel, L.J.**, **Bailey, R.E.**, **Shelton, K.J.**, **Williams, S.P.**, **Kramer, L.J. and Norman, R.M.** (2008). Head-worn display concepts for surface operations for commercial aircraft, NASA/TP-2008-215321.

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

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

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

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

[11] **RTCA SC-147** (2013). Minimum Operational Performance Standards for Traffic Alert and Collision Avoidance System II (TCAS II), Technical Report for RTCA/DO-185B Change 2.

[12] **GFB-AAI** (2004). Investigation Report, Braunschweig: BFU, German Federal Bureau of Aircraft Accident Investigation AX001-1-2/02

[13] **Wing, D.** (2005). A potentially useful role for airborne separation in 4D-trajectory ATM operations, *5th AIAA Aviation Technology Integration and Operations (ATIO) Conference*.

[14] **Green, S.M. and Bilimoria, K.D.** (2001). Distributed air/ground traffic management for en route flight operations, *Air Traffic Control Quarterly*, *9*, 259–285.

[15] **Barhydt, R. and Kopardekar, P.** (2005). Joint NASA Ames/Langley experimental evaluation of integrated air/ground operations for en route free maneuvering, *6th USA/Europe Air Traffic Management Research and Development Seminar*. Baltimore, Maryland, USA, June 27-30.

[16] **MFFP** (2005). Mediterranean Free Flight Programme Final Report 2005, MFFP Technical Report D821.

[17] **Krishnamurthy, K.**, **Barmore, B. and Bussink, F.** (2005). Airborne precision spacing in merging terminal arrival routes, *6th USA/Europe Air Traffic Management R&D Seminar*. Baltimore, Maryland, USA, June 27-30.

[18] **Blom, H.A.P.**, **Obbink, B.K. and Bakker, G.I.** (2007). Safety risk simulation of an airborne self separation concept of operation, *7th AIAA Aviation Technology Integration and Operations (ATIO) Conference*. Belfast, Northern Ireland, September 19-20

[19] **Air Traffic Organization Technical Operations Services (ATO-W)** (2007). System Wide Information Management(SWIM) Artchitecture - Segment 1 Version 1.0, National Airspace System Enterprise Architecture (NASEA) Technical Report.

[20] **Bilimoria, K.D.** (2000). A Geometric Optimization Approach to Aircraft Conflict Resolution, *AIAA Guidance, Navigation, and Control Conf. and Exhibit*. Dever,CO,USA, August 14-17

[21] **Temizer, S.**, **Kochenderfer, M.J.**, **Kaelbling, L.P.**, **Lozano-pérez, T. and Kuchar, J.K.** (2010). Collision Avoidance for Unmanned Aircraft using Markov Decision Processes, *AIAA Guidance, Navigation and Control Conference*. Toronto, Ontario, Canada, 25-29 July

[22] **Sondik, E.J.** (1971). *The Optimal Control of Partially Observable Markov Processes*, Ph.D. Dissertation, Stanford, CA.

[23] **Kaelbling, L. P., Littman, M. L. and Cassandra, A.R.** (1998). Planning and Acting in Partially Observable Stochastic Domains, Artificial Intelligence, *101 – 1-2*, 99–134.

[24] **Thrun, S.**, **Burgard, W. and Fox, D.** (2006). *Probabilistic Robotics*, MIT Press, Cambridge, MA.

[25] **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.

[26] **Yang, L.**, **Yang, J.H.**, **Kuchar, J. and Feron, E.** (2004). A Real-Time Monte Carlo Implementation for Computing Probability of Conflict, *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Providence, RI.

[27] **Yang, L.C.** (2000). *Aircraft Conflict Analysis and Real-Time Conflict Probing Using Probabilistic Trajectory Modeling*, Ph.D. Dissertation, Massachusetts Inst. of Technology, Lexington, MA.

[28] **Blom, H.A.P.**, **Krystul, J.B.**, **G J, K.M.B. and Obbink, B.K.** (2007). Free Flight Collision Risk Estimation by Sequential MC Simulation, *Stochastic Hybrid Systems*, 249–281.

[29] **Paielli, R.A. and Erzberger, H.** (1997). Conflict Probability Estimation for Free Flight, *AIAA Journal of Guidance Control and Dynamics*, *20*, 588–596.

[30] **Jones, T.** (2012). Tractable Conflict Risk Accumulation in Quadratic Space for Autonomous Vehicles, *Journal of Guidance, Control, and Dynamics*,*29 – 1*, 39–48.

[31] **van Daalen, C.E. and Jones, T.** (2009). Fast conflict detection using probability flow, *Automatica*, *45*(8), 1903–1909.

[32] **Prandini, M. and Watkins, O.J.** (2005). Probabilistic aircraft conflict detection, HYBRIDGE IST-2001-32460, WP3 Project Deliverable D3.2.

[33] **Lymperopoulos, I. and Chaloulos, G.** (2010). An advanced particle filtering algorithm for improving conflict detection in Air Traffic Control, *International Conference on Research in Air Transportation (ICRAT)*, Budapest, Hungary, June 1-3.

[34] **Yang, L.C. and Kuchar, J.K.** (1997). Prototype Conflict Alerting System for Free Flight, *Journal of Guidance, Control, and Dynamics*, *20*, 768–773.

[35] **Carpenter, B. and Kuchar, J.** (1997). Probability-Based Collision Alerting Logic for Closely-Spaced Parallel Approach, *AIAA Meeting Papers on Disc*

[36] **Jansson, J. and Gustafsson, F.** (2008). A framework and automotive application of collision avoidance decision making, *Automatica*, *44*(9), 2347–2351.

[37] **Chryssanthacopoulos, J.P.**, **Kochenderfer, M.J.**, **Arino, T.**, **Erzberger, H.**, **Bethke, B.**, **Gustafsson, F. and Jones, T.** (2012). Hazard Alerting Based on Probabilistic Models, *Journal of Guidance, Control, and Dynamics*, *35*(2), 442–450.

[38] **Kuchar, J.** (2000). A review of conflict detection and resolution modeling methods, *IEEE Transactions on Intelligent Transportation Systems*, *1–4*.

[39] **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.

[40] **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.

[41] **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.

[42] **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.

[43] **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.

[44] **Winder, L.F.**, (2004). Hazard Avoidance Alerting with Markov Decision Processes, MIT ICAT Technical Reports.

[45] **Chryssanthacopoulos, J.P.** (2011). Accounting for state uncertainty in collision avoidance, *Journal of Guidance, Control, and Dynamics*, *34*(4), 951-960.

[46] **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.

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

[48] **Manikonda, V.**, **Krishnaprasad, P.S. and Hendler, J.** (1998). Languages, behaviors, hybrid architectures and motion control, *Mathematical Control Theory*. 199–226

[49] **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*, Seville, Spain. December 12-15.

[50] **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*.

[51] **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.

[52] **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.

[53] **Lindeman, S. and LaValle, S.M.**, (2005). Current Issues in Sampling-Based Motion Planning, *The Eleventh International Symposium Springer Tracts in Advanced Robotics*, *15*, 36–54.

[54] **Frazzoli, E. and Karaman, S.** (2010). Incremental sampling-based algorithms for optimal motion planning, *Robotics: Science and Systems (RSS)*, Zaragoza, Spain, June.

[55] **LaValle, S. and Kuffner, J.** (1999). Randomized kinodynamic planning, *Robotics and Automation, 1999. Proceedings, International Conference on*, (pp. 473–479), Detroit, MI, May 10-15.

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

[57] **Luders, B.D.**, **Karaman, S. and How, J.P.** (2013). Robust sampling-based motion planning with asymptotic optimality guarantees, *AIAA Guidance, Navigation, and Control Conference (GNC)*, Boston, MA, August 19-2.

[58] **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*, Puszczykowo, June 17-20.

[59] **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*, (pp.1478–1483), Shanghai, China, May 9-13.

[60] **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.

[61] **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*, (pp.1618–1623), Sendai, Japan, 28 Sept.-2 Oct.

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

[63] **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*, (pp.3874–3880), Barcelona, Spain, April 18-22.

[64] **Kalisiak, M. and van de Panne, M.** (2007). Faster motion planning using learned local viability models, *Robotics and Automation, 2007 IEEE International Conference on*, (pp.2700–2705), Roma, Italy, April 10-14.

[65] **Li, Y. and Bekris, K.E.** (2010). Balancing state-space coverage in planning with dynamics, *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, (pp.3246–3253), Anchorage, Alaska, May 3 - 8.

[66] **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*, (pp.2632–2639), San Francisco, CA, September 25-30.

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

[68] **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.

[69] **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.

[70] **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*.

[71] **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.

[72] **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.

[73] **Koyuncu, E.**, **Ure, N.K. and Inalhan, G.** (2010). Integration of Path/Maneuver Planning in Complex Environments for Agile Maneuvering UCAVs, *Journal of Intelligent and Robotics Systems*, *57*(1-4), 143–170.

[74] **Koo, T.J.**, **Pappas, G.J. and Sastry, S.** (2001). Multi-Modal Control of Systems with Constraints, *40th IEEE Conference Decision and Control*, (pp.2075–2081), Orlando, Florida, USA, December 4-7.

[75] **Frazzoli, E.**, **Dahleh, M.A. and Feron, E.** (2002). Real-time motion planning for agile autonomous vehicles, *AIAA Journal of Guidance and Control*, *25*(1), 116–129.

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

[77] **Schouwenaars, T.**, **How, J. and Feron, E.** (2004). Receding horizon path planning with implicit safety guarantees, *American Control Conference, 2004. Proceedings of the 2004*, volume 6, (pp.5576–5581).

[78] **Ghosh, R. and Tomlin, C.** (2000). Nonlinear Inverse Dynamic Control for Mode-Based Flight, *Proceedings of AIAA Guidance, Navigation and Control Conference and Exhibit*, Denver, CO, 14-17 August.

[79] **Oishi, M. and Tomlin, C.** (1999). Nonlinear control of a vstol aircraft, *The Proceedings of the 38th IEEE Conference on Decision and Control*, December 7-10.

[80] **Teo, R. and Tomlin, C.J.** (2000). Computing provably safe aircraft to aircraft spacing for closely spaced parallel approaches, *Digital Avionics Systems Conference, 2000. Proceedings. DASC. The 19th*, (pp.2D2–1 –2D2–9), vol.1., Philadelphia, PA.

[81] **Maurino, D. and Salas, E.** (2010). *Human Factors in Aviation*, Elsevier.

[82] **Bécouarn, L.**, **Dominici, J.**, **Bader, J.**, **Fabbri, M.**, **Pregnolato, M.**, **Sarayeddine, K.**, **Cuypers, D.**, **De Smet, H.**, **Alapetite, A.**, **Sgouros, N.** *et al.* (2012). ODICIS (One Display for a Cockpit Interactive Solution)-Final public progress report, ODICIS Consortium.

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

[84] **Parasuraman, R.**, **Sheridan, T.B. and Wickens, C.D.** (2000). A model for types and levels of human interaction with automation, *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, *30*(3), 286–297.

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

[86] **Endsley, M.R.** (2011). *Designing for situation awareness: An approach to user-centered design*, CRC Press.

[87] **Mark, G. and Kobsa, A.** (2005). The effects of collaboration and system transparency on CIVE usage: an empirical study and model, *Teleoperators and Virtual Environments*, *14*(1), 60–80.

[88] **Duggan, G.B.**, **Banbury, S.**, **Howes, A.**, **Patrick, J. and Waldron, S.M.** (2004). Too much, too little or just right: Designing data fusion for situation awareness, *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 48, SAGE Publications, pp.528–532.

[89] **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.

[90] **Dao, A.Q.V.**, **Brandt, S.L.**, **Battiste, V.**, **Vu, K.P.L.**, **Strybel, T. and Johnson, W.W.**, (2009). The impact of automation assisted aircraft separation on situation awareness, Human Interface and the Management of Information, *Information and Interaction, Springer*, pp.738–747.

[91] **Sherry, L.**, **Lard, J.**, **Fennell, K. and Feary, M.** (2009). Methodology for Estimation of Benefits of Human-Factors Engineering in NextGen/SESAR Development, *Eighth USA/Europe Air Traffic Management Research and Development Seminar*, Napa, California, USA.

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

[93] **Ligda, S.V.**, **Johnson, N.**, **Lachter, J. and Johnson, W.W.** (2009). Pilot Confidence with ATC Automation using Cockpit Situation Display Tools in a Distributed Traffic Management Environment, *Human Interface and the Management of Information, Information and Interaction*, (pp.816–825).

[94] **Peterson, S. and Pinska, E.** (2006). Human Performance with Simulated Collimation in Transparent Projection Screens, *International Conference on Research in Air Transportation*, Belgrade, Serbia and Montenegro.

[95] **Stanton, N.A.**, **Plant, K.L.**, **Harvey, C. and Bolton, L.** (2012). A study of input devices for menu navigation in the cockpit, *International conference on Ergonomics and Human Factors 2012*, Blackpool, UK, 16-19 April.

[96] **PARC/CAST**, (2013). Operational Use of Flight Path Management Systems, Technical Report of PARC/CAST.

[97] **Leonesl, J.L.** (2008). *Definition of an aircraft intent description language for air traffic management applications*, Ph.D. thesis, University of Glasgow.

[98] **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*, pp.2–2.B.4–1.

[99] **EUROCONTROL** (2014). User Manual for the Base of Aircraft Data (BADA) Family 4, EEC TechnicalScientific Report.

[100] **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.

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

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

[103] **Basar, T. and Olsder, G.J.** (1999). *Dynamic Noncooperative Game Theory*, volume 23, SIAM.

[104] **Isaacs, R.** (1999). *Differential games: a mathematical theory with applications to warfare and pursuit, control and optimization*, Courier Corporation.

[105] **Isler, V.**, **Sun, D. and Sastry, S.** (2005). Roadmap Based Pursuit-Evasion and Collision Avoidance., *Robotics: Science and Systems*, volume 1, pp.257–264.

[106] **Karaman, S. and Frazzoli, E.** (2010). Incremental Sampling-based Algorithms for a class of Pursuit-Evasion Games, *Ninth International Workshop on the Algorithmic Foundations of Robotics*, 1–16.

[107] **Hsu, D.**, **Kindel, R.**, **Latombe, J.C. and Rock, S.** (2002). Randomized Kinodynamic Motion Planning with Moving Obstacles, *International Journal of Robotics Research*, *21*(2), 233–255.

[108] **LaValle, S.M. and Kuffner, J.J.** (2001). Randomized kinodynamic planning, *The International Journal of Robotics Research*, *20*(5), 378–400.

[109] **Owen, M.**, **Beard, R.W. and McLain, T.W.**, (2014). Implementing Dubins Airplane Paths on Fixed-Wing UAVs*, *Handbook of Unmanned Aerial Vehicles*,(pp.1677–1701), Springer.

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

[111] **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.

[112] **Khalil, H.K.** (2002). *Nonlinear Systems, Third Ed.*, Prentice Hall.

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

[114] **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.

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

[116] **Halton, J.H.** (1960). On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals, *Numerische Mathematik*, *2*(1), 84–90.

APPENDICES

APPENDIX A.1 : Formal Intent Data Languages
APPENDIX A.2 : Local Trajectory Optimization

APPENDIX A.1: Formal Intent Data Languages

The Collaborative Mid-Term Planning module is envisioned to integrate intent sharing capability into the flight deck system. This module utilizes three-level formal description languages; AIDL, ICDL and FIDL; which are developed by Boeing Research and Technology Europe, in order to efficiently convey the intent data to other trajectory planners. These languages enable to define an action sequence of the aircraft dynamics or the flight plan with different levels of detail, fully or partially specifying some aspects of the aircraft motion and leaving others open for later optimization/specification/planning considering the constraints and the objectives.

The Aircraft Intent Description Language (AIDL) is a low level formal description employed to model the basic commands, guidance modes or control strategies at the disposal of the pilot/FMS to manage the aircraft. The AILD instructions basically fill each degrees of freedom of the mathematical description model describing the aircraft motion. The instructions set has been derived from a detailed analysis of the different primitive modes of operation that an aircraft may employ in the ATM context. The Figure A.1 shows an example AIDL sentence with six degrees of freedom threads. Any valid combination of the 35 predefined instructions (e.g. Hold Speed, Hold Altitude etc.) with their specifiers and execution intervals (bounded with end-triggers) describes the motion control objectives of the aircraft. The AIDL grammar is subjected to set of lexical and syntactical rules in order to create a valid sequence through six parallel threads.

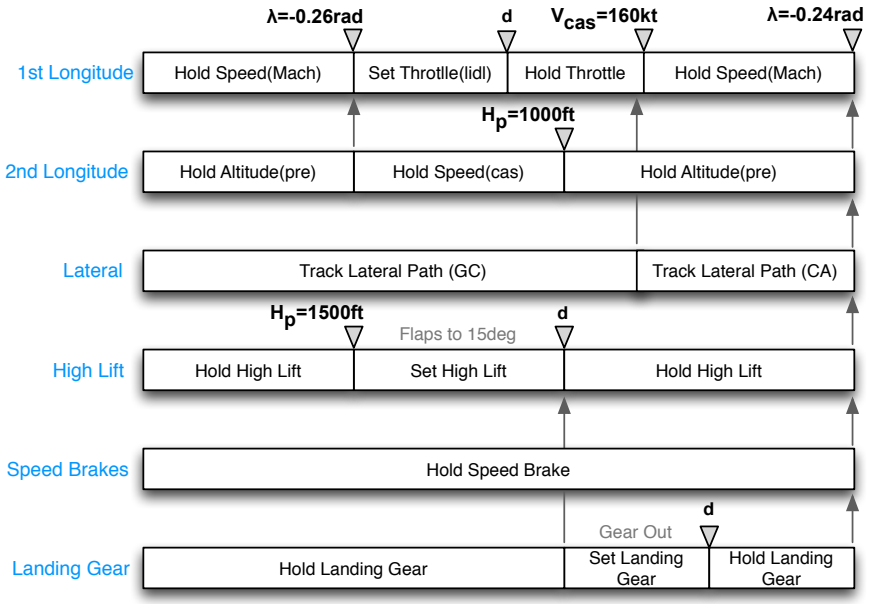

Figure A.1: Example AIDL instance with six parallel threads.

The higher level language, Flight Intent Description Language (FIDL), is seen as a basic plan for trajectory prediction where the details to be satisfied by the resulting trajectory are left unspecified. FIDL provides a high level directions (flight segments or AIDL compositions) on how a flight will be operated, and includes operation specific

constraints and objectives. In general, the flight intent does not determine a unique trajectory. A basic example for FIDL instruction is given in Figure A.2 where flight segment primitives defines certain waypoint tracking composites with their constraints and objectives. Flight Segment (FS) instructions may also include additional details about the lower level operation of flight if some aspects of the aircraft behaviour are defined. These are represented by the composites (ICDL) which are the template representation of a set of AIDL compositions such as Level Flight, Descent, Level Thrust Deceleration.

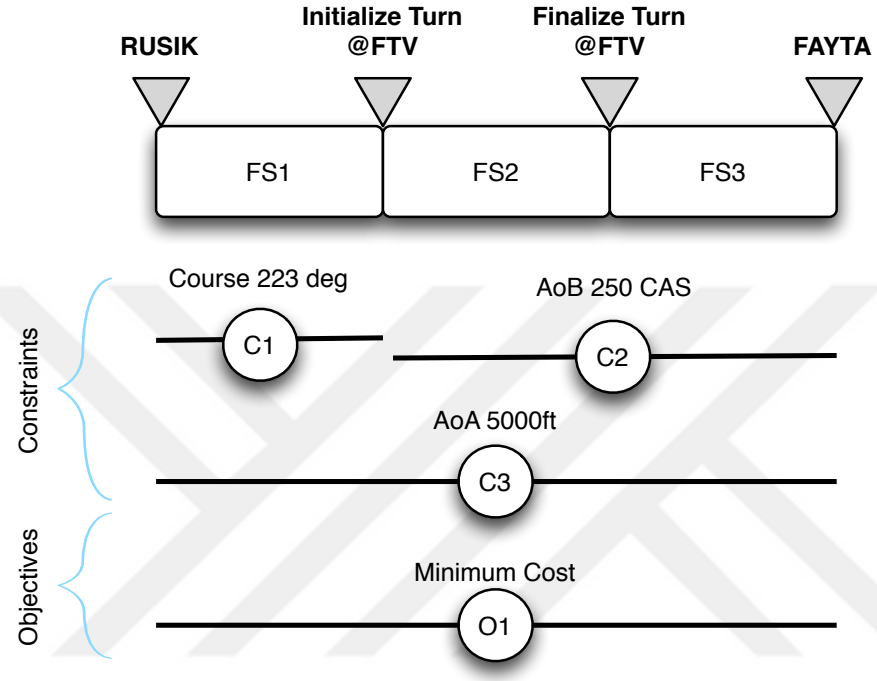

Figure A.2: Example FIDL instance with flight segments, constraints and objectives.

The Figure 1.2 demonstrates whole data handling process for the intent sharing/negotiation between the Trajectory Language Processing Engines (TLPE) that accepts AIDL and FIDL inputs. In this structure, air-to-ground data exchange procedure is handled through FIDL that is a higher level language enabling the pilot to easily interpret and modify. The ground based Conflict Monitoring and Trajectory Management blocks represent ATC functions including all autonomous and decision support tools for managing tactical flight operations at mid-term to short term domain. The flight decks have also equipped with the avionics enabling similar capabilities of TLPE for trajectory planning and intent negotiation handling. In addition to routine automated data exchange, any intervention (negotiation request) can be initiated through air-to-ground data-link when it is needed. The Conflict Monitoring functions in both air and flight segment monitor potential loss of separation situation within the prescribed time interval through predicted trajectories (TPs). These trajectories may also include uncertainty factors in a set of parameters (e.g. in aircraft performance, position, weather etc.) and their "what-if" extensions (e.g. considering unexpected behaviors) in a probabilistic manner. The Traffic Management function operates this intervention from the ground by attaching new constraints or objectives to the pre-negotiated FIDL sequence when it requires. Similarly, flight deck can also request (initiate) a flight intent negotiation, when the on-board Conflict Monitoring detects potential conflict or the pilot redefine existing or processing (during negotiation) flight

intent. Air-to-air intent data exchange procedure is handled through AIDL without negotiation. The AIDL sharing is the low-level "machine-to-machine" communication where the pilot can only observe through the predicted trajectories. In this case, the on-board Conflict Monitoring block monitors the potential conflicts between the predicted trajectories of the aircraft in the surveillance traffic through the Trajectory Computation Infrastructure.

The Intent Generation Infrastructure includes an Intent Generation Core Processor (IGCP), Aircraft Performance Model (APM) based on BADA models and a pair of databases, one storing a User Preferences Model (UPM) and one storing an Operational Context Model (OCM). The UPM involves the preferred operational strategies directing the aircraft such as the preferences of an airline, how to react to meteorological conditions, cost structure minimizing time of flight or cost of flight, maintenance costs, environmental impact, communication capabilities, and security considerations. The OCM involves standard constraints on the use of airspace. The Intent Generation Infrastructure accepts a FIDL sentence (including flight segments, constraints and objectives), and Initial State (IC) as inputs; then processes with UPM, OCM, APM in order to translate into a compatible AIDL through the Intent Generation Core Process (IGCP). The Trajectory Computation Infrastructure translates an AIDL sentence into unique predicted trajectory (TP). In this level, it is expected that different trajectory computation tools would result in the same predicted trajectory if they use the same inputs and models such as a) aircraft intent (AIDL) b) Initial conditions (IC) (aircraft state at the initial position and environmental condition at this altitude), c) Aircraft Performance Model (APM), d) Environmental Model (EM), and e) similar trajectory computation algorithms. Event though these premises may not be unattainable in practice, sharing the aircraft intent significantly contributes to achieve partial trajectory synchronization.

APPENDIX A.2: Local Trajectory Optimization

The problem to find an optimal trajectory $J(x^*) = min_{u \in U} J(x)$ in real-time applications is highly complicated due to the nonlinearity of the constraints. In this section, we present a multi-modal decomposition approach that solves the optimization problem approximately by trading-off the optimality with reduction in computational complexity. The developed trajectory generation algorithm accepts any initial state x_{init}, a reference state x_{wpt} and reference cost index CI as the required input set, then returns a dynamically feasible trajectory segment with the proper control input and time sequence, if it exists. This local planning approach ignores the obstacles and leaves the conflict-free check to the global planner, which is explained in the following section.

The basic idea behind the modal decomposition is to first determine the required flight template (i.e. cruise, climb or descent), then generate an appropriate maneuver sequence with the corresponding parameter set using a finite maneuver library (i.e. Level-Thrust Acceleration, Level-Thrust Deceleration, Level Flight, CAS Climb, Mach Climb, CAS Descent, Mach Descent). By following this approach, a proper maneuver sequence can steer the aircraft from the initial state x_{init} to a reference state x_{wpt}, if it is applicable. Maneuver sequences with their transition states and time schedule are determined by performing a gradient-descent search on 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 with finding cost-efficient switching parameters between the predefined maneuver modes, which significantly reduces computational complexity, and enables real-time local trajectory generation. This flight template selection and cost minimization formalism is parallel to the trajectory generation procedure õn the flight management systems (FMS). In our structure, the modal maneuver sequence selection in the flight templates differs due to the volume of the interest region, where our aim is to generate relatively small trajectory segments resolving potential conflicts, while the FMS calculates cost-efficient trajectories for entire flight.

The procedure for maneuver decomposition is represented with two-layer finite state automaton. At the higher level, which is shown in Figure B.1, the Flight Template automaton G_M takes place. TheFlight Template automaton involves four discrete states, i.e. $X_M = \{q_i : i = 0, \ldots, 3\}$. Note that the initial state q_0 and the final state x_M is the same, which practically represents the "stand-by" state of the trajectory generation procedure. Respectively, q_1, q_2 and q_3 represents the Cruise flight template, the Climb flight template and the Descent flight template. We have developed separate approximate optimization strategies for each flight template. These flight templates can be also directly mapped to the maneuver library, namely Maneuver Mode automaton, where X_m = { Level-Thrust Acceleration, Level-Thrust Deceleration, Level Flight, CAS Climb, Mach Climb, CAS Descent, Mach Descent}. The elements of the Maneuver Library are common maneuvers that are used in the Flight Management

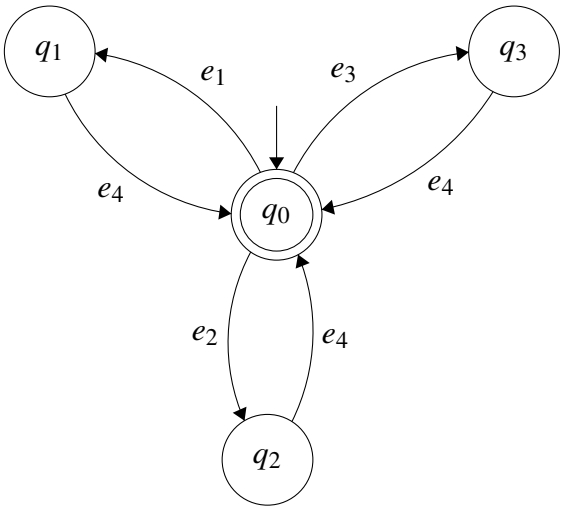

Figure B.1: Flight Template automaton with Cruise, Climb or Descent modes.

Computers on the commercial aircrafts. Each flight template X_{M_i} utilizes certain maneuver sequences, which they are element of the X_m.

Flight Template automaton has three different transition events, which are the functions of the aircraft′s states. These transition functions are Boolean, i.e. they return either 0 or 1. The finite set of events is;

$$E = \{e_i : i = 1, \ldots, 3\} \tag{A.1}$$

where e_1 is set to 1, if the initial altitude $h_{init} \in x_{init}$ is the same as the goal altitude $h_{goal} \in x_{wpt}$; e_2 is 1, if the initial altitude h_{init} is lower than the goal altitude h_{goal}; and e_3 is set to 1, if the initial altitude h_{init} is higher than the goal altitude h_{goal}. If the target state x_{wpt} is not reachable, e_4 is set to 1 and the automaton returns to its initial state. Note that the flight templates X_{M_i}s and the corresponding maneuver set X_m involves the longitudinal dynamics. To generate the 3D motion and keep the aircraft on the horizontal track, a lateral path controller that generates control input μ_{TAS} is coupled with these maneuver modes.

For integrating the equations of motion, the Euler discretization is utilized with a step size of $\Delta t = 0.1$ second. Cost-effective trajectory generation strategies for Cruise, Climb and Descent flight templates based on BADA 4 are described in the following subsections.

2.0.1 Cruise

If the given initial and goal altitudes h_{init} and h_{goal} are equal; the flight template automaton, which is shown in Figure B.1 moves to the Cruise template. In this template, the variations in speed and altitude are small, and hence they are ignored in the aircraft performance considerations. Hence, following simplifications are assumed for the cruise template;

$$L\cos\mu_{TAS} = W, \quad T = D \quad and \quad \gamma_{TAS} = 0 \tag{A.2}$$

In addition, we assume that the time interval $[t_I, t_1]$, which the lateral path controller generates a bank angle sequence $\mu_{TAS_k} \neq 0$ for all $k \in \mathbb{N}$ is quite smaller than

Table B.1: Summary of the local planning with flight template and maneuver library.

Flight Template	Optimized Parameter	Maneuvers	δ_T	γ_{TAS}
Climb	$M_{target,climb}$	CAS Climb	MCMB	Eq. **A.10**
		Mach Climb	MCMB	Eq. **A.10**
		Level flight	δ_{Tlvl}	0
Cruise	$M_{target,cruise}$	Level-Thrust Acceleration	δ_{Tmax}	0
		Level-Thrust Deceleration	LIDL	0
		Level flight	δ_{Tlvl}	0
Descent	$V_{target,CAS}$	Mach Descent	LIDL	Eq. **A.11**
		CAS Descent	LIDL	Eq. **A.11**
		Level flight	δ_{Tlvl}	0

the interval $[t_I, t_f]$ where t_I is the initial time, t_f is the final time of the trajectory respectively and $t_I < t_1 < t_f$. Hence, $L = W$ holds for the cruise mode. The Cruise flight template utilizes a set of maneuvers S_{cruise} = { Level-Thrust Acceleration, Level-Thrust Deceleration, Level Flight }. In this mode the optimum cruise Mach number $M_{target,cruise}$, which is also called the economic Mach number is calculated. Switching between these maneuvers is controlled by the automaton depicted in Figure B.2. If the initial Mach number $M_{init} \in x_{init}$ of the aircraft is not equal to the economic cruise Mach number $M_{target,cruise}$, the Level-Thrust Acceleration (q_1) or Level-Thrust Deceleration (q_2) maneuvers are executed. The automaton jumps to the state q_1, if $M_{target,cruise}$ is larger than the initial Mach number M_{init} (e_1). Similarly, the state of the automaton is switched to the state q_2, if the aircraft needs to decelerate to reach $M_{target,cruise}$ (e_3). The aircraft executes the Level Flight (q_3) maneuver, if $M_{init} = M_{target,cruise}$, which is represented by the event e_2. The events e_4 and e_5 corresponds to the aircraft reaching the economic cruise Mach number $M_{target,cruise}$ and the goal state c_{wpt} respectively. For the acceleration maneuver q_1, the throttle parameter δ_T is set to its maximum limit δ_{Tmax} to obtain the maximum thrust. On the other hand, LIDL (low idle) rating is used for generating lower thrust T than drag force D for deceleration. Eventually, the Level Flight mode follows each of these maneuvers where the throttle parameter δ_T is adjusted to a certain level, that is;

$$\delta_{Tlvl} = \left\{ \delta_T \in [\delta_{Tmin}, \delta_{Tmax}] | T(\delta_T, M) = D(M) \right\} \tag{A.3}$$

To obtain the economic cruise Mach number $M_{target,cruise}$, the cost function J needs to be reformulated as a function of the Mach number. This transformed is represented as minimizing the economy cruise cost function (ECCF), which is a method typically preformed in FMC [99]. Note that, the wind speed w is also taken into account in this minimization.

$$dJ = c_f F dt + c_t dt = \frac{c_f F + c_t}{(V_{TAS} + w)} dr \tag{A.4}$$

$$ECCF = \frac{dJ}{c_f dr} = \frac{c_f F + c_t}{c_f (V_{TAS} + w)} = \frac{CI + F}{V_{TAS} + w} \tag{A.5}$$

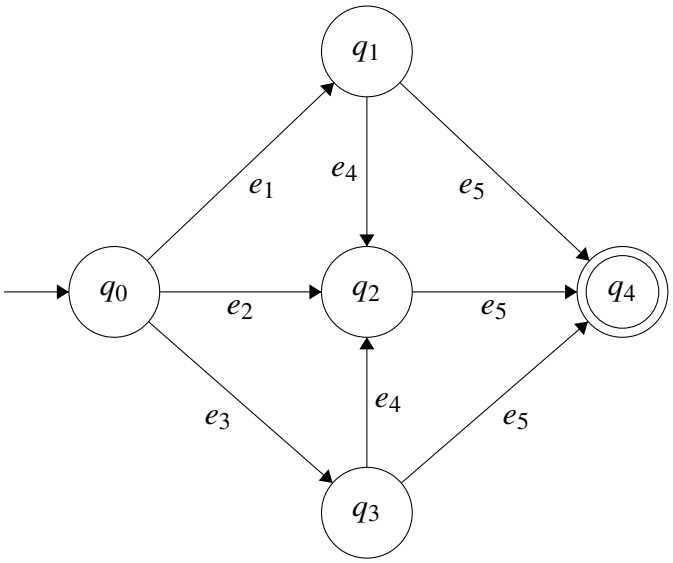

Figure B.2: Cruise flight template automaton.

Due to the holding assumptions that the cost index CI is constant during the flight, and wind speed is stationary, i.e. $\dot{w} = 0$, the fuel consumption F and true airspeed V_{TAS} can be expressed directly as a function of the Mach number M. The speed of sound a can easily be obtained depending on the geopotential pressure altitude and atmospheric conditions. Then, the true airspeed can be expressed as $V_{TAS} = \sqrt{\kappa RT}M$, where R is the natural gas constant.

The lift force L and the lift coefficient C_L can be obtained from $C_L = C_L(M)$, since the initial weight of the aircraft is known. Drag polar relationship between the lift coefficient C_L and the drag coefficient C_D in Eq. 4.10 results in a drag force formulation, which is expressed as a function of Mach number only. This expression is combined with Eq. **A.2**, and Eq. 4.12 yields the thrust coefficient as a function of the Mach number. Using Eq. 4.13 and Eq. 4.22, fuel consumption formulation is simplified into a function of one variable, i.e. Mach number. Figures B.3 and B.4 shows the variation of ECCF of a Boeing 737-800 aircraft with the Mach number for different cost index CI values. The lower curve refers to smaller cost index, where the fuel saving is more effective than the time cost reduction, and results in a lower cruise Mach number. The effect of the wind on the economy cruise Mach number can be seen in Figure B.5, where the increment in the true airspeed V_{TAS} due to tail winds results in a decrease in the optimum cruise speed, and head winds cause a higher cruise economy Mach number $M_{target,cruise}$.

2.0.2 Climb

If the given initial altitude h_{init} is at a lower flight level than the goal altitude h_{goal}, the Flight Template automaton goes into the Climb state. This mode uses a set of maneuvers, $S_{cmb} = \{CASClimb, MachClimb, LevelFlight\}, S_{cmb} \subset S_m$. The finite state automaton given in Figure B.6 controls the maneuver switching in Climb template. In this automaton; q_0, q_1 and q_2 represent the CAS Climb, Mach Climb and Level Flight maneuvers respectively, where q_3 is the final state. Cost efficient trajectory generation for Climb flight template involves speed profile scheduling with CAS/Mach Climb. CAS Climb and Mach Climb maneuvers with selected speed schedule reduce the flight

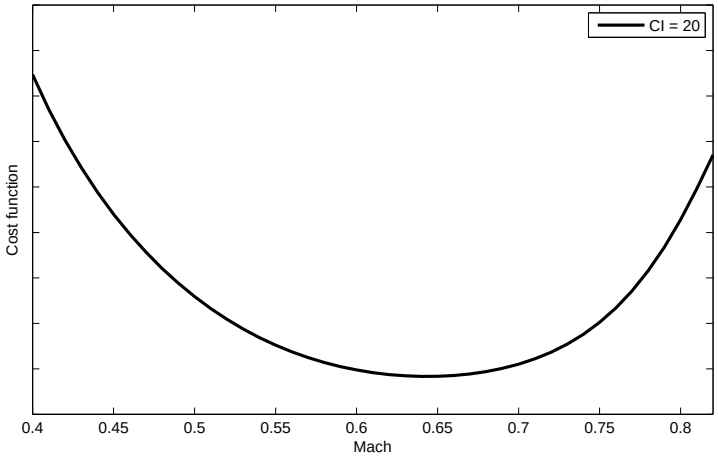

Figure B.3: Economy cruise cost function - Mach curve for $CI = 20$.

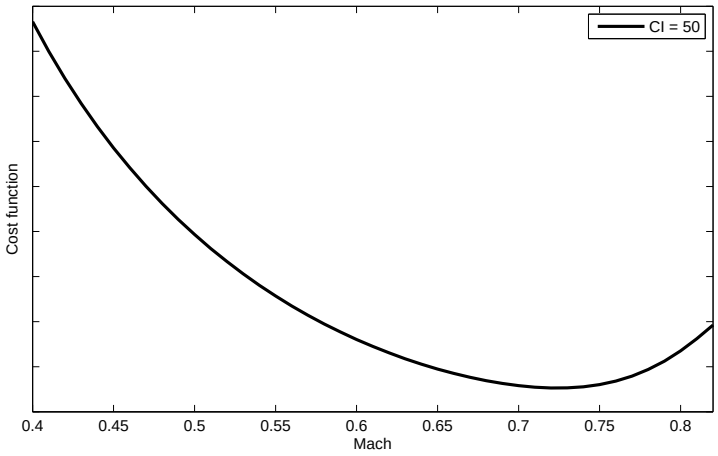

Figure B.4: Economy cruise cost function - Mach curve for $CI = 50$.

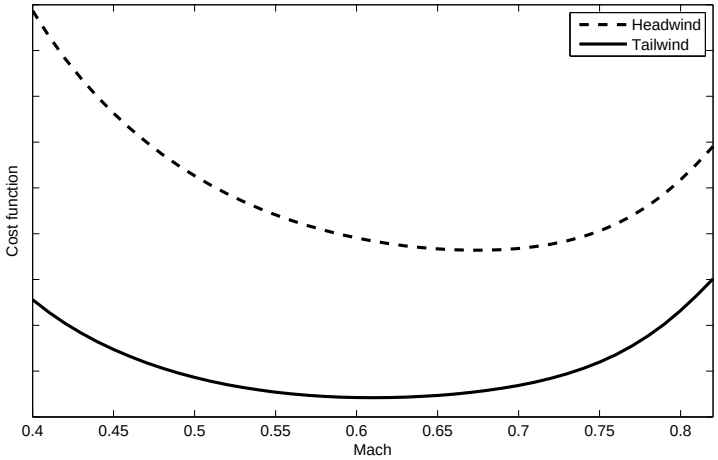

Figure B.5: Wind effect on optimum cruise Mach.

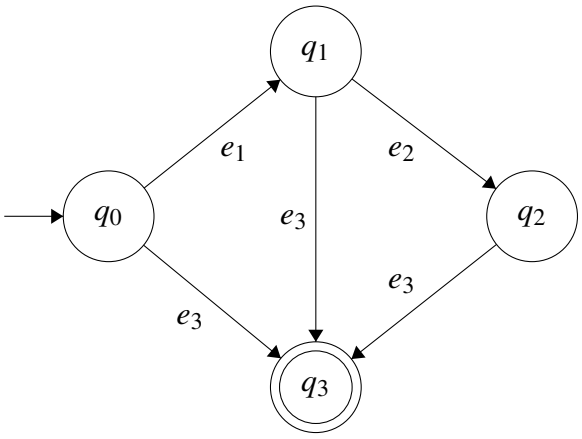

Figure B.6: Climb flight template automaton.

control effort, and it is easy to perform and follow the maneuver actions [99]. Similar to the Cruise flight template, objective of the cost reduction problem is to find a proper Mach number $M_{target,climb}$ that minimizes the total costs of Climb flight template. At the first stage, the aircraft starts climbing with a constant calibrated airspeed CAS (q_0), until it reaches the Mach number $M_{target,climb}$. The function e_1 refers whether the aircraft reached that Mach number. If it is set to 1, the aircraft continues to ascend while maintaining its Mach number $M_{target,climb}$ (q_2). Mach Climb maneuver proceeds until the goal altitude h_{goal} is reached, which corresponds to the e_2 transition. Automaton is terminated at any state if the aircraft reaches the goal state x_{wpt}, which is defined as the e_3 transition.

Throttle parameter δ_T is set to MCMB (Maximum Climb) rating for both CAS Climb and Mach Climb maneuvers. In BADA 4, this rating is formulated as:

$$\delta_T = \sum_{k=0}^{5} \left(\sum_{r=1}^{6} b_{r+6k} M^{r-1} \right) \delta^k. \tag{A.6}$$

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_{TAS} to evaluate required speed schedule. The equation with proper discretization depending on true airspeed V_{TAS} can be written as:

$$V_{TASk+1} - V_{TASk} = \Delta t \left(\frac{T_k - D_k}{m_k} - g sin\gamma_{TASk} - \dot{w}_1(k) \right) \tag{A.7}$$

To obtain the flight path angle in Eq. **A.7**, following assumptions are employed: a) the derivative of the wind speed $\dot{w}$ is very small in comparison with the aircraft speed profile, hence it is negligible., 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. Due to these assumptions, the target true airspeed $V_{TAS,target}$, whose value equals to the initial equivalent calibrated airspeed V_{CAS} for CAS Climb and corresponding Mach number M for Mach Climb modes can be estimated. Thus, the flight path angle γ_{TAS}, which remains an the only unknown parameter in Eq. **A.7**, can be evaluated from the following expression;

$$\Gamma = \frac{1}{g} \left(\frac{T_k - D_k}{m_k} - \frac{V_{TASk+1} - V_{TASk}}{\Delta t} \right) \tag{A.8}$$

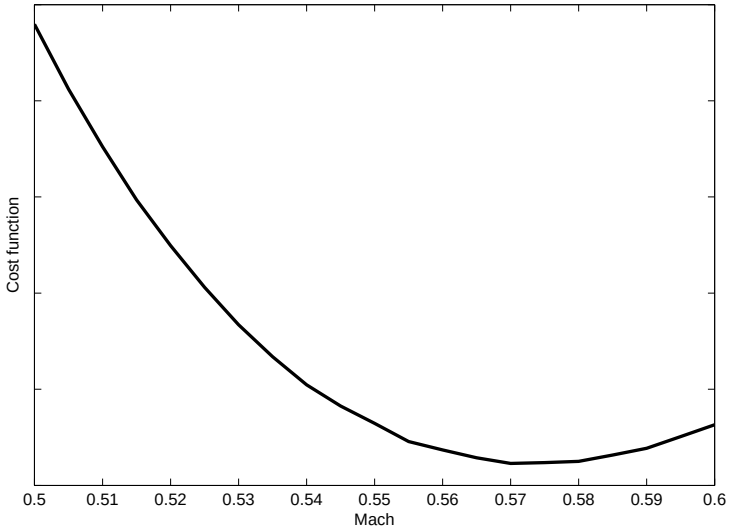

Figure B.7: Cost function - Mach curve for Climb flight template.

$$\gamma_{TAS_k} = sin^{-1}\Gamma \tag{A.9}$$

The flight path angle γ_{TAS} is also an initial input for the next step of the lift force calculation, i.e. Eq. 4.8, as it is required to compute other forces to fulfill the mathematical degree of the freedom in the Eq. **A.8**. Updated flight path angle is used for evaluating the aircraft's new state. If the solution goes towards infeasible or impractical flight path angles, saturation at $\gamma_{max} > 0$ is appended. The final form of the flight path angle becomes;

$$\gamma_{TAS} = \begin{cases} \gamma_{max} & \text{if } \sin^{-1}\Gamma > \gamma_{max} \text{ or } \Gamma > 1 \\ \sin^{-1}\Gamma & \text{o.w} \end{cases} \tag{A.10}$$

In order to demonstrate the cost management for the Climb flight template, an example test case where the aircraft climbs to FL200 from FL100 and travels over a distance of 25nm is simulated. Figure B.7 shows the relation between the cost function and Mach number. Figures B.8, B.9 and B.10 illustrate the Mach number M variation, calibrated airspeed CAS variation and altitude h change respectively. Effect of the presence of wind w is also demonstrated in Figure B.11 such that the optimum Mach number increases in the presence of the headwind and decreases in the presence of the tailwind.

2.0.3 Descent

If the given initial altitude h_{init} is at a higher flight level than the goal altitude h_{goal}, the Flight Template automaton goes to the Descent state. This template uses the following set of maneuvers;

$$S_{des} = \{MachDescent, CASDescent, LevelFlight\}, S_{des} \subset S_m$$

Switching between these maneuvers is controlled by the Descent template, which is shown in Figure B.12. The first maneuver q_0 is Mach Descent, where the descent action is applied by maintaining the initial Mach number, M_{init}. Respectively, q_1

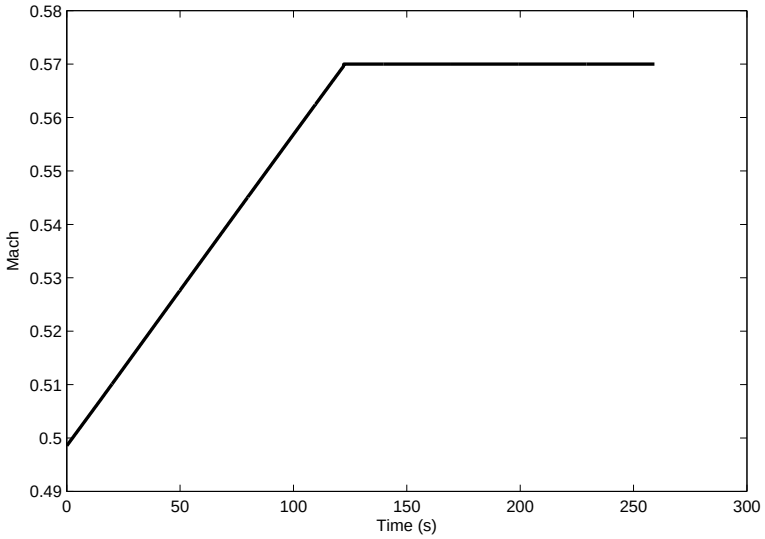

Figure B.8: Mach number variation with time.

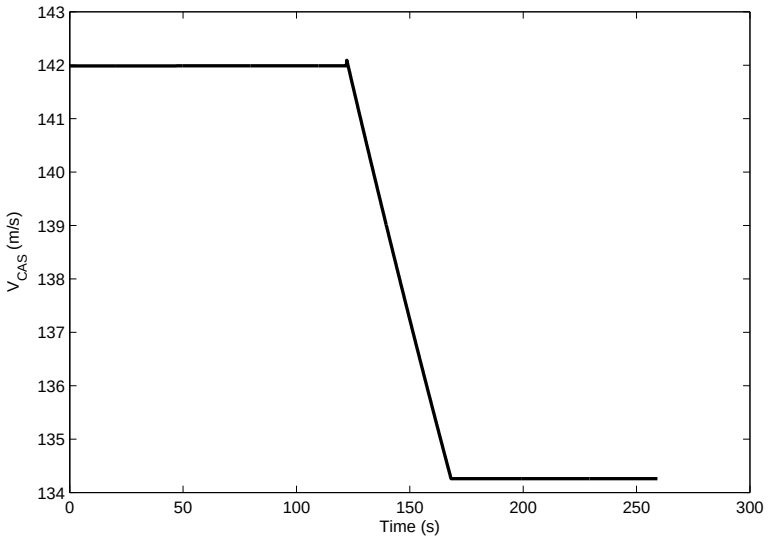

Figure B.9: CAS variation with time.

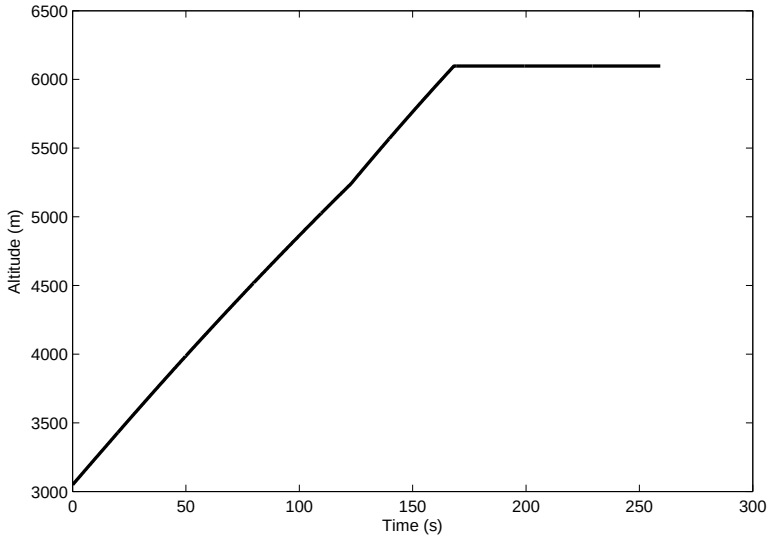

Figure B.10: Change in altitude with time.

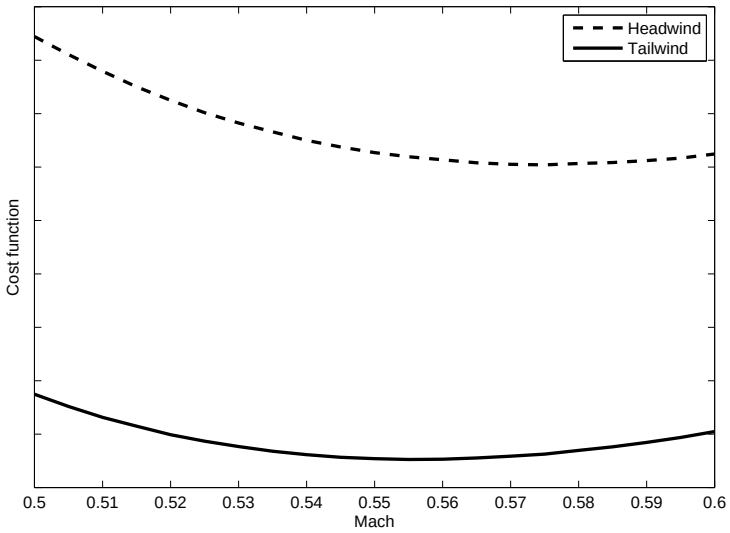

Figure B.11: Wind effect on optimum climb Mach.

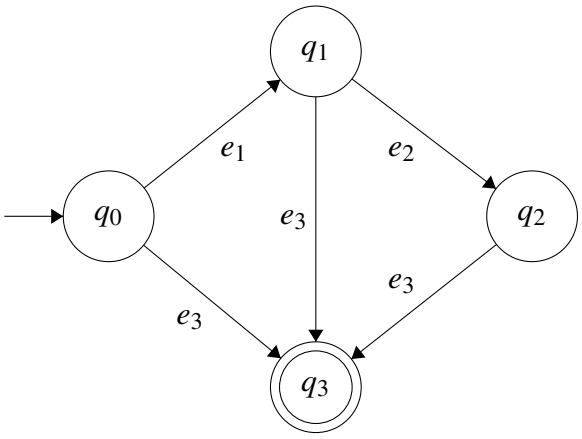

Figure B.12: Descent flight template automaton.

denotes the CAS Descent where the calibrated airspeed *CAS* is maintained at the descent action; q_2 denotes Level Flight where the aircraft maintains its altitude h and true airspeed V_{TAS}; q_3 denotes the goal state is reached. The transition function e_1 is set to 1, if the target calibrated airspeed$V_{target,CAS}$ is reached. e_2 is set to 1, if the goal altitude h_{goal} is reached, and e_3 is set to 1, if the goal state is reached.

Cost management in Descent flight template utilizes a similar strategy presented in the Climb template. The process is the determination of an optimum speed schedule composed of calibrated airspeed *CAS* and Mach number M. In this flight template, the aircraft starts to descent while maintaining its initial Mach number M_{init} until it reaches the target calibrated airspeed $V_{target,CAS}$ that minimizes the total flight cost. The maneuver mode set of this template includes Mach Descent, CAS Descent and Level Flight. Throttle parameter δ_T is set to LIDL 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 move its nose down. The flight path angle γ_{TAS} is computed at each step using the same approach presented in the Climb section. To prevent a steep descent maneuver, flight path angle γ_{TAS} is restricted to a value of $\gamma_{min} < 0$.

$$\gamma_{TAS} = \begin{cases} \gamma_{min} & \text{if } \sin^{-1}\Gamma < \gamma_{min} \text{ or } \Gamma < -1 \\ \sin^{-1}\Gamma & \text{o.w} \end{cases} \tag{A.11}$$

2.0.4 Lateral path control

A lateral path controller is coupled with the flight templates (i.e. Cruise, Climb and Descent) to turn aircraft's heading to the goal state. At each step, aircraft's bearing with respect to the horizontal location of the final state x_{wpt}, in other words, desired heading χ_{des} is calculated and necessary bank angle input μ_{TAS} sequence is evaluated, that is;

$$\psi = \frac{(\chi_{desk} - \chi_{TASk})(V_{TASk}\cos\gamma_{TASk})}{\Delta t L_k}$$

$$\mu_{TAS} = \sin^{-1}\psi$$

The equations above may give an infeasible solution if the deviation is too large. A saturation at $\phi_{lim} > 0$ is applied here to keep the bank angle μ_{TAS} in aircraft's limits. The final expression can be given as:

$$\mu_{TAS} = \begin{cases} \phi_{lim} & \text{if } \sin^{-1}\psi > \phi_{lim} \text{ or } \psi > 1 \\ -\phi_{lim} & \text{if } \sin^{-1}\psi < -\phi_{lim} \text{ or } \psi < -1 \\ \sin^{-1}\psi & \text{o.w} \end{cases}$$

The local approximate trajectory optimization, namely cost efficient trajectory generation through the flight templates and their modal maneuver library is summarized in the Table B.1. Following section gives Sampling-based Conflict Detection and Resolution algorithm, which utilizes this local planning to generate trajectory segments between the given states.

CURRICULUM VITAE

Name Surname: Emre Koyuncu

Place and Date of Birth: Ankara

Adress: Istanbul Teknik Universitesi, Ucak ve Uzay Bilimleri Fakultesi, Oda No: 312, Maslak, Istanbul, TURKEY

E-Mail: emre.koyuncu@itu.edu.tr

B.Sc.: Electrical Engineering (2005), Istanbul Technical University

M.Sc.: Mechatronics Engineering (2008), Istanbul Technical University

Professional Experience and Rewards:

- SESAR PhD Fellow, SESAR JU WP-E, HALA! Research Network, 2013-2015
- Visiting Researcher, Massachusetts Institute of Technology, Aero-Astro Department, 2014-2015
- Visiting Researcher, BOEING Research and Technology Europe, 2013 - 2014
- Research Fellow, SESAR WP-E Project SecureDataCloud – applying secure computation to ATM data, 2013-2016
- Research Fellow, Resilience2050 – New design principles fostering safety, agility and resilience for ATM, EU FP7-Transport Project, 2012-2015
- Research Fellow, TUBITAK N^{o}111M167 project — Design of Real-Time Automation and Decision-Support Systems for ATC Operators and Pilots Within The New Air Traffic Management Paradigm, 2012-2015
- Research Fellow, TUBITAK N^{o}108M523 project — Advanced Test Platform for Attitude Control of Nano Satellites, 2009-2012
- Teaching Assistant, Aeronautics Engineering Department, Istanbul Technical University, 2011 - cont.
- Research Assistant, Controls and Avionics Laboratories, Istanbul Technical University, 2006 - cont.
- Mechatronics R&D Engineer, Electronics and Mechatronics Department, ZER Robotics & Automation Ltd. Co., 2005-2006

List of Publications and Patents:

Journals and Book Chapters

- **Koyuncu E.**, Uzun M., Inalhan, G., Cross-Entropy based Cost Efficient 4D Trajectory Generation for Airborne Conflict Resolution, Journal of Aerospace Engineering, SAGE, 2015 (Submitted)

- **Koyuncu E.**, Tarhan A. F., Hasanzade M., Inalhan G., Collaborative Intent Exchange Based Flight Management System with Airborne Collision Avoidance for UAS, Journal of Intelligent and Robotics Systems, Springer, 2015 (Submitted)

- Karaman, S., **Koyuncu E.**, Inalhan, G., Innovative Collaborative Task Allocation for UAVs, Handbook of Unmanned Aerial Vehicles, Ed. Kimon P. Valavanis and George J. Vachtsevanos, 2013

- **Koyuncu E.**, Inalhan, G., Exploiting Delayed and Imperfect Information for Generating Approximate UAV Target Interception Strategy, Journal of Intelligent and Robotics Systems, Springer, January 2013, Volume 69, Issue 1-4, pp 313-329

- Karakas, H., **Koyuncu E.**, Inalhan, G., ITU Tailless UAV Design, Journal of Intelligent and Robotics Systems, DOI: 10.1007/s10846-012-9695-4, Springer, 2012

- **Koyuncu E.**, Ure, N. K., Inalhan, G., Integration of Path/Maneuver Planning in Complex Environments for Agile Maneuvering UCAVs, Journal of Intelligent and Robotics Systems, issn – 0921-0296 (Print) 1573-0409 (Online), doi – 10.1007/s10846-009-9367-1, Springer, January 2010

- **Koyuncu, E.**, Inalhan, G., Dynamically Feasible Probabilistic Motion Planning in Complex Environments for UAVs, Robotics 2010; Current and Future Challenges, Edited by Houssem Abdellatif, ISBN 978-953-7619-78-7, Publishing date: February 2010

Peer Reviewed Conferences

- Tarhan A. F., **Koyuncu E.**, Hasanzade M., Ozdemir U., Inalhan G., Formal Intent Based Flight Management System Design for Unmanned Aerial Vehicles, International Conference on Unmanned Aircraft Systems (ICUAS'14),pp. 984-992 27-30, Orlando, FL, USA, 2014

- Uzun M., Guner G., **Koyuncu E.**, Inalhan G., Integrated Flight Deck Testbed with Next Generation Visual Decision Support Tools, International Conference on Research in Air Transportation (ICRAT'14), May 26-30, Istanbul, 2014

- **Koyuncu, E.**, Garcia, E., and Inalhan, G., AUTOFLY-Aid: Multi-Modal Trajec- tory Projection Approach for Airborne Collision Detection and Avoidance, International Conference on Application and Theory of Automation in Command and Control Systems, 28-30 May, Naples, 2013

- **Koyuncu, E.**, Tokadli, G., Bahcivan, Z., Inalhan, G., Design of a Pilot-Centered Visual Decision-Support System for Airborne Collision Avoidance, 31st Digital Avionics Systems Conference (DASC'12), Williamsburg, VA, October 14-18, 2012

- **Koyuncu, E.**, Garcia, E., and Inalhan, G., Flight Deck Automation Support with Dynamic 4D Trajectory Management for ACAS: AUTOFLY-AID, Integrated

Communications Navigation and Surveillance (ICNS) Conference, pp. C4-1 - C4-9, Herndon, VA, 2012

- **Koyuncu, E.**, Garcia, E., and Inalhan, G., AUTOFLY-Aid: Flight Deck Automation Support with Dynamic 4D Trajectory Management for Responsive and Adaptive Airborne Collision Avoidance, International Conference on Application and Theory of Automation in Command and Control Systems, 29-31 May, London, 2012

- **Koyuncu, E.**, Karaman, S., Frazzoli, E., and Inalhan, G., Probabilistic 4D Conflict Detection and Avoidance Strategy in En-Route Air Traffic, International Conference on Application and Theory of Automation in Command and Control Systems, 26-27 May, Barcelona, 2011

- Bahcivan, Z., **Koyuncu, E.**, Aydin, S. S., Cinar, E., Cavcar, A., and Inalhan, G., ITU/AU Air Traffic Control Network Simulator for Design, Development and Testing of Automated ATM Systems, International Conference on Application and Theory of Automation in Command and Control Systems, 26-27 May, Barcelona, 2011

- **Koyuncu, E.**, and et. all, ITU-pSAT II: High-precision Nanosatellite ADCS Development Project, 5th International Conference on Recent Advances in Space Technologies, 9-11 June, Istanbul, 2011

Inalhan, G., **Koyuncu, E.**, et. all, Design and Development of ITU pSAT II: On orbit demonstration of a high-precision ADCS for nanosatellites, 8th International ESA Conference on Guidance and Navigation Control Systems, 5-10 June, Carlsbad, Czech Republic, 2011

- Inalhan G., **Koyuncu E.**, Üre N. K. ve Kurtulus C.,ITU PSAT II: Attitute Controlled Nano Satellite Platform Developing Project, III. National Conference on Aeronautics and Astronautics(UHUK'10), Eskisehir, Turkey, September 2010 (in Turkish)

- **Koyuncu E.** ve Inalhan G., Low Cost Star Tracker Development for Nano Satellites: ITU PSAT II Satellite Star Tracker System, III. National Conference on Aeronautics and Astronautics(UHUK'10), Eskisehir, Turkey, September 2010 (in Turkish)

- **Koyuncu E.**, Ure, N. K., Inalhan, G., Integration of Path/Maneuver Planning in Complex Environmentsfor Agile Maneuvering UCAVs, Proc. 2th Int. Symposium on Unmanned Aerial Vehicles (UAV'09), Reno, Nevada, June 2009

- **Koyuncu, E.**, Inalhan, G., A Probabilistic B-Spline Motion Planning Algorithm for Unmanned Helicopters Flying in Dense 3D Environments, Int. Conf. Intelligent Robots and Systems (IROS'08),Nice, France, September 2008

- **Koyuncu, E.**, Ure, N.K., Inalhan, G., A Probabilistic Algorithm for Mode Based Motion Planning of Agile Air Vehicles in Complex Environments, Int. Federation of Automatic Control World Congress (IFAC WC'08),Seoul, South Korea, June 2008

- Turanli, M., **Koyuncu, E.**, Inalhan, G., Multi-Purpose Testbed Design for Mobile Autonomous Robots: ITUCAL Robotic Testbed Platform, Turkish National Automatic Control Committee (TOK 2008), Istanbul, Turkey, September 2008 (in Turkish)

- **Koyuncu E.**, Demirok E., Artificial Neural Networks Controlled Maximum Power Point Tracker for Solar Vehicles in Rapidly Changing Conditions, Applications and Innovations in Intelligent Systems (ASYU), June 2006 (in Turkish)

▪ **Koyuncu E.**, Yeniçeri R., Ceylan O., Computer Vision Based Object Tracker Robot Arm Design, Turkish National Automatic Control Committee (TOK 2005), Istanbul, Turkey, June 2005 (in Turkish)

PUBLICATIONS/PRESENTATIONS ON THE THESIS

▪ **Koyuncu E.**, Inalhan, G., 2013: Exploiting Delayed and Imperfect Information for Generating Approximate UAV Target Interception Strategy, *Journal of Intelligent and Robotics Systems*, DOI: 10.1007/s10846-012-9693-6, Springer, January 2013, Volume 69, Issue 1-4, pp 313-329

▪ Uzun M., Guner G., **Koyuncu E.**, Inalhan G., Integrated Flight Deck Testbed with Next Generation Visual Decision Support Tools, *International Conference on Research in Air Transportation (ICRAT'14)*, May 26-30, Istanbul, 2014

▪ **Koyuncu E.**, Garcia, E., and Inalhan, G., AUTOFLY-Aid: Multi-Modal Trajectory Projection Approach for Airborne Collision Detection and Avoidance, *International Conference on Application and Theory of Automation in Command and Control Systems*, 28-30 May, Naples, 2013

▪ **Koyuncu E.**, Tokadli, G., Bahcivan, Z., Inalhan, G., Design of a Pilot-Centered Visual Decision-Support System for Airborne Collision Avoidance, *31st Digital Avionics Systems Conference (DASC'12)*, Williamsburg, VA, October 14-18, 2012

▪ **Koyuncu E.**, Garcia, E., and Inalhan, G., Flight Deck Automation Support with Dynamic 4D Trajectory Management for ACAS: AUTOFLY-AID, *Integrated Communications Navigation and Surveillance (ICNS) Conference*, pp. C4-1 - C4-9, Herndon, VA, 2012

▪ **Koyuncu E.**, Garcia, E., and Inalhan, G., AUTOFLY-Aid: Flight Deck Automation Support with Dynamic 4D Trajectory Management for Responsive and Adaptive Airborne Collision Avoidance, *International Conference on Application and Theory of Automation in Command and Control Systems*, 29-31 May, London, 2012

▪ **Koyuncu E.**, Karaman, S., Frazzoli, E., and Inalhan, G., Probabilistic 4D Conflict Detection and Avoidance Strategy in En-Route Air Traffic, *International Conference on Application and Theory of Automation in Command and Control Systems*, 26-27 May, Barcelona, 2011