





**ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE**  
**ENGINEERING AND TECHNOLOGY**

**OPTIMIZATION-BASED CONTROL OF  
COOPERATIVE AND NONCOOPERATIVE MULTI-AIRCRAFT SYSTEMS**

**Ph.D. THESIS**

**Bariş BAŞPINAR**

**Department of Aeronautics and Astronautics Engineering**

**Aeronautics and Astronautics Engineering Programme**

**FEBRUARY 2020**



**ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE**  
**ENGINEERING AND TECHNOLOGY**

**OPTIMIZATION-BASED CONTROL OF  
COOPERATIVE AND NONCOOPERATIVE MULTI-AIRCRAFT SYSTEMS**

**Ph.D. THESIS**

**Bariş BAŞPINAR  
(511152101)**

**Department of Aeronautics and Astronautics Engineering**

**Aeronautics and Astronautics Engineering Programme**

**Thesis Advisor: Asst. Prof. Dr. Emre KOYUNCU**

**FEBRUARY 2020**



**KOOPERATİF VE KOOPERATİF OLMAYAN ÇOKLU UÇAK SİSTEMLERİNİN  
OPTİMİZASYON TABANLI KONTROLÜ**

**DOKTORA TEZİ**

**Barış BAŞPINAR  
(511152101)**

**Uçak ve Uzay Mühendisliği Anabilim Dalı**

**Uçak ve Uzay Mühendisliği Programı**

**Tez Danışmanı: Asst. Prof. Dr. Emre KOYUNCU**

**ŞUBAT 2020**





Barış BAŞPINAR, a Ph.D. student of ITU Graduate School of Science Engineering and Technology 511152101 successfully defended the thesis entitled “OPTIMIZATION-BASED CONTROL OF COOPERATIVE AND NONCOOPERATIVE MULTI-AIRCRAFT SYSTEMS”, which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

**Thesis Advisor :**     **Asst. Prof. Dr. Emre KOYUNCU** .....  
Istanbul Technical University

**Jury Members :**     **Prof. Dr. İbrahim ÖZKOL** .....  
Istanbul Technical University

**Asst. Prof. Dr. Nazım Kemal ÜRE** .....  
Istanbul Technical University

**Asst. Prof. Dr. Uğur Özdemir** .....  
Eskisehir Technical University

**Dr. Cengiz PAŞAOĞLU** .....  
Personel Data Protection Authority

**Date of Submission :**   **02 January 2020**

**Date of Defense :**     **12 February 2020**





*To my family,*



## **FOREWORD**

I would like to begin with expressing my gratitude to my thesis advisor Prof. Emre Koyuncu for his support and guidance throughout this study. I would also like to thank Prof. Gokhan Inalhan for his guidance and Prof. Hamsa Balakrishnan for hosting and guiding me during my visiting research at MIT.

Throughout PhD education, The Scientific and Technological Research Council of Turkey (TÜBİTAK) has honoured and supported me with its prestigious domestic graduate scholarship and international doctoral research fellowship. Hence, I would like to thank TÜBİTAK for funding my studies during my PhD.

Finally, I thank my parents for being source of my motivation.

February 2020

Bariş BAŞPINAR  
(Aeronautical & Control Engineer)



## TABLE OF CONTENTS

	<u>Page</u>
<b>FOREWORD</b> .....	<b>ix</b>
<b>TABLE OF CONTENTS</b> .....	<b>xi</b>
<b>ABBREVIATIONS</b> .....	<b>xv</b>
<b>LIST OF TABLES</b> .....	<b>xvii</b>
<b>LIST OF FIGURES</b> .....	<b>xix</b>
<b>SUMMARY</b> .....	<b>xxi</b>
<b>ÖZET</b> .....	<b>xxiii</b>
<b>1. INTRODUCTION</b> .....	<b>1</b>
1.1 Purpose of Thesis .....	2
1.2 Literature Review .....	3
1.2.1 Aerial combat .....	3
1.2.2 Air traffic control .....	5
1.2.3 Mission specification with temporal logic.....	8
1.3 Overview .....	10
<b>2. HYBRID MANEUVER-BASED FRAMEWORK FOR EVALUATION OF AIR-TO-AIR COMBAT</b> .....	<b>15</b>
2.1 Purpose .....	15
2.2 Hybrid Automaton.....	16
2.3 Aircraft Model and Maneuvers for Military Aviation .....	17
2.3.1 Aircraft dynamics .....	17
2.3.2 Maneuver library .....	18
2.3.2.1 Aileron roll .....	18
2.3.2.2 Barrel roll .....	19
2.3.2.3 Loop .....	20
2.3.2.4 Break turn.....	21
2.3.2.5 Immelmann.....	21
2.3.2.6 Split-S.....	22
2.3.2.7 Vertical spiral.....	23
2.3.2.8 Spiral dive.....	24
2.3.2.9 Low yo-yo .....	24
2.3.2.10 High yo-yo.....	26
2.4 Maneuver-Based Optimal Strategy Generation for Autonomous Air Combat .....	27
2.4.1 Objective function for one-vs-one aerial combat .....	27
2.4.2 Evaluation strategy of zero-sum game .....	30
2.4.3 Air superiority for n-vs-m aerial combat.....	34
2.4.3.1 Offensive orientation score.....	34
2.4.3.2 Defensive orientation score .....	34

2.4.3.3	Range score .....	34
2.4.3.4	Speed score.....	35
2.4.3.5	Altitude score .....	35
2.4.3.6	Total scores for n-vs-m air combat.....	35
2.4.4	Security strategy to evaluate the air combat between multiple aircraft..	36
2.5	Simulation Results.....	38
2.5.1	Maneuver-based one-vs-one aerial combat .....	38
2.5.2	Maneuver-based two-vs-one aerial combat.....	41
<b>3.</b>	<b>OPTIMIZATION-BASED AUTONOMOUS AIR TRAFFIC CONTROL...</b>	<b>45</b>
3.1	Purpose .....	45
3.2	Aircraft Model and Maneuvers for Civil Aviation .....	45
3.2.1	Aircraft dynamics .....	46
3.2.2	Reference trajectory and speed schedule.....	47
3.2.3	Trajectory tracking guidance.....	48
3.2.3.1	Longitudinal controller.....	48
3.2.3.2	Lateral controller.....	48
3.2.3.3	Speed controller.....	52
3.2.4	Simulation and trajectory prediction .....	52
3.2.5	ATCo interventions.....	53
3.2.5.1	Speed change.....	53
3.2.5.2	Altitude change .....	53
3.2.5.3	Direct routing .....	54
3.2.5.4	Course change .....	54
3.2.5.5	Vector for spacing.....	55
3.2.5.6	Holding pattern.....	56
3.3	Maneuver-Based Optimal Strategy Generation for Autonomous Air Traffic Controller .....	57
3.4	Separation Assurance with Autonomous ATCo.....	63
3.5	Capacity Improvement with Autonomous ATCo .....	66
3.5.1	Stochastic traffic simulation environment .....	66
3.5.2	Breaking point analysis .....	68
3.5.3	Simulations in stochastic traffic environment.....	69
<b>4.</b>	<b>MISSION PLANNING AND CONTROL OF MULTI-AIRCRAFT SYSTEMS WITH SIGNAL TEMPORAL LOGIC SPECIFICATIONS .....</b>	<b>73</b>
4.1	Purpose .....	73
4.2	System Behavior and Signal Temporal Logic .....	73
4.2.1	Signal temporal logic.....	74
4.2.2	Robust STL specifications .....	75
4.3	Differential Flatness .....	75
4.4	Motion Planning for Flat Systems.....	76
4.5	Flat Description of Aircraft Dynamics for Civil Aviation.....	78
4.6	Differential Flatness-Based Optimal Strategy Generation for Multi-Aircraft Systems.....	80
4.6.1	Performance limits and initial conditions.....	81
4.6.2	STL operators .....	83
4.6.3	Continuous-time satisfiability.....	84



4.7 Illustrative Examples .....	85
4.7.1 Reach and avoid problem .....	85
4.7.2 Approach control and arrival sequencing .....	86
4.7.3 UAV fleet narrow-passage problem .....	89
<b>5. FORMULATION OF AERIAL COMBAT GAME VIA DIFFERENTIAL FLATNESS-BASED RECEDING HORIZON CONTROL.....</b>	<b>93</b>
5.1 Purpose .....	93
5.2 Flat Description of Aircraft Dynamics for Military Aviation.....	93
5.3 Differential Flatness-Based Optimal Strategy Generation for Aerial Combat	97
5.3.1 Aerial combat between aircraft .....	97
5.3.2 Reachable set of enemy aircraft .....	101
5.3.3 Initialization scheme.....	102
5.3.4 Solution method.....	104
5.3.5 Receding horizon control .....	105
5.4 Simulation Results.....	106
<b>6. CONCLUSION .....</b>	<b>111</b>
<b>REFERENCES.....</b>	<b>115</b>
<b>CURRICULUM VITAE .....</b>	<b>124</b>



## ABBREVIATIONS

<b>ACC</b>	: Area Control Center
<b>ATC</b>	: Air Traffic Control
<b>ATCo</b>	: Air Traffic Control Operator
<b>ATM</b>	: Air Traffic Management
<b>BADA</b>	: Base of Aircraft Data
<b>B-Spline</b>	: Basis Spline
<b>CI</b>	: Cost Index
<b>CTAS</b>	: Center-TRACON Automation System
<b>DD</b>	: Dynamic Density
<b>FDPS</b>	: Flight Data Processing System
<b>FL</b>	: Flight Level
<b>HJB</b>	: Hamilton-Jacobi-Bellman
<b>HJI</b>	: Hamilton-Jacobi-Isaacs
<b>ILP</b>	: Integer Linear Programming
<b>LOS</b>	: Line-of-Sight
<b>LTL</b>	: Linear Temporal Logic
<b>MILP</b>	: Mixed-Integer Linear Programming
<b>MIP</b>	: Mixed-Integer Programming
<b>MTL</b>	: Metric Temporal Logic
<b>NASA</b>	: National Aeronautics and Space Administration
<b>PLOS</b>	: Pursuit and Line-of-Sight
<b>PMM</b>	: Point Mass Model
<b>RHC</b>	: Receding Horizon Control
<b>RVSM</b>	: Reduced Vertical Separation Minima
<b>SQR</b>	: Sequential Quadratic Programming
<b>STAR</b>	: Standard Terminal Arrival Route
<b>STL</b>	: Signal Temporal Logic
<b>TL</b>	: Temporal Logic
<b>TRACON</b>	: Terminal Radar Approach Control Facilities
<b>UAV</b>	: Unmanned Aerial Vehicle
<b>URET</b>	: User Request and Evaluation Tool
<b>VFS</b>	: Vector for Spacing



## LIST OF TABLES

	<u>Page</u>
<b>Table 2.1</b> : Input Parameters for Simulations.....	38
<b>Table 2.2</b> : Initial Conditions for Scenarios.....	38
<b>Table 2.3</b> : Initial Conditions for All Scenarios. ....	41
<b>Table 3.1</b> : The Set of Maneuvers. ....	64
<b>Table 4.1</b> : Performance Evaluation. ....	86





## LIST OF FIGURES

	<u>Page</u>
<b>Figure 2.1</b> : Aileron Roll Maneuver: (a) Hybrid Dynamics, (b) Simulation. ....	19
<b>Figure 2.2</b> : Barrel Roll Maneuver: (a) Hybrid Dynamics, (b) Simulation.....	20
<b>Figure 2.3</b> : Loop Maneuver: (a) Hybrid Dynamics, (b) Simulation.....	21
<b>Figure 2.4</b> : Break Turn Maneuver: (a) Hybrid Dynamics, (b) Simulation. ....	22
<b>Figure 2.5</b> : Immelmann Maneuver: (a) Hybrid Dynamics, (b) Simulation. ....	23
<b>Figure 2.6</b> : Split-S Maneuver: (a) Hybrid Dynamics, (b) Simulation. ....	24
<b>Figure 2.7</b> : Vertical Spiral Maneuver: (a) Hybrid Dynamics, (b) Simulation. ....	25
<b>Figure 2.8</b> : Spiral Dive Maneuver: (a) Hybrid Dynamics, (b) Simulation. ....	26
<b>Figure 2.9</b> : Low Yo-Yo Maneuver: (a) Hybrid Dynamics, (b) Simulation. ....	27
<b>Figure 2.10</b> : High Yo-Yo Maneuver: (a) Hybrid Dynamics, (b) Simulation. ....	28
<b>Figure 2.11</b> : Combat Geometry and Scoring Parameters: (a) Aircraft Relative Geometry, (b) Range Score.....	29
<b>Figure 2.12</b> : Air Combat with Symmetric Initial Conditions: (a) 3D Positions, (b) Combat Angles for Blue, (c) Combat Angles for Red, (d) Range. ....	39
<b>Figure 2.13</b> : Air Combat with Asymmetric Initial Conditions: (a) 3D Positions, (b) Combat Angles for Blue, (c) Combat Angles for Red, (d) Range. ....	40
<b>Figure 2.14</b> : Air Combat for 1111 Configuration: (a) 3D Positions, (b) Combat Angles between $b_1$ and $r_1$ , (c) Combat Angles between $b_2$ and $r_1$ , (d) Ranges. ....	42
<b>Figure 2.15</b> : Air Combat for 1101 Configuration: (a) 3D Positions, (b) Combat Angles between $b_1$ and $r_1$ , (c) Combat Angles between $b_2$ and $r_1$ , (d) Ranges. ....	43
<b>Figure 2.16</b> : Air Combat for 1001 Configuration: (a) 3D Positions, (b) Combat Angles between $b_1$ and $r_1$ , (c) Combat Angles between $b_2$ and $r_1$ , (d) Ranges. ....	44
<b>Figure 3.1</b> : Flight Phase Transitions. ....	47
<b>Figure 3.2</b> : Straight-Line and Arc Geometry: (a) Straight Line Path, (b) Circular Motion.....	49
<b>Figure 3.3</b> : Transitions between Horizontal Controllers.....	52
<b>Figure 3.4</b> : Direct Routing Action. ....	54
<b>Figure 3.5</b> : Course Change.....	55
<b>Figure 3.6</b> : Vector for Spacing and Holding Pattern. ....	56

<b>Figure 3.7</b> : Example Scenarios: (a) Scenario 1: reference trajectories, (b) Scenario 2: reference trajectories, (c) Scenario 3: reference trajectories, (d) Scenario 1: simulation result with conflict resolution at $t = 50s$ , (e) Scenario 2: simulation result with conflict resolution at $t = 50s$ , (f) Scenario 3: simulation result with conflict resolution at $t = 50s$ .....	65
<b>Figure 3.8</b> : Flight Plans in ISTANBUL ACC for a Standard Day. ....	67
<b>Figure 3.9</b> : Airspace Capacity Estimation. ....	69
<b>Figure 3.10</b> : Distributions for Number of Aircraft in Airspace: (a) Real Throughput, (b) Simulated Throughput for Traffic Density = 30000. ....	70
<b>Figure 3.11</b> : Maneuver Distribution.....	71
<b>Figure 4.1</b> : Optimal Trajectories for a Reach and Avoid Problem with 20 Aircraft.....	86
<b>Figure 4.2</b> : Optimal Trajectories for Approach Control and Arrival Sequencing. ....	87
<b>Figure 4.3</b> : Solution to Approach Control and Arrival Sequencing: horizontal distances between aircraft pairs. ....	87
<b>Figure 4.4</b> : Solution to Approach Control and Arrival Sequencing: speeds of each aircraft stay within speed bounds. ....	88
<b>Figure 4.5</b> : Solution to Approach Control and Arrival Sequencing: bank angles of each aircraft stay within bank angle limits.....	88
<b>Figure 4.6</b> : Optimal Trajectories for the Quadrotor Fleet for the Narrow-Passage Problem.....	90
<b>Figure 4.7</b> : Solution to UAV Fleet Narrow-Passage Problem: horizontal distances between quadrotor pairs. ....	90
<b>Figure 4.8</b> : Solution to UAV Fleet Narrow-Passage Problem: speeds of each quadrotor stay within speed bounds.....	91
<b>Figure 5.1</b> : Initialization via Convex Hull of Enemy's Reachable Set. ....	103
<b>Figure 5.2</b> : Principle of Receding Horizon Control.....	105
<b>Figure 5.3</b> : Aerial Combat between two UAVs: Scenario 1 (a) 3D Positions, (b) Horizontal Positions, (c) Altitudes, (d) Speeds, (e) Total Load Factors.....	106
<b>Figure 5.4</b> : Aerial Combat between two UAVs: Scenario 2 (a) 3D Positions, (b) Horizontal Positions, (c) Altitudes, (d) Combat Angles, (e) Range. ....	107



# **OPTIMIZATION-BASED CONTROL OF COOPERATIVE AND NONCOOPERATIVE MULTI-AIRCRAFT SYSTEMS**

## **SUMMARY**

In this thesis, we mainly focus on developing methods that ensure autonomous control of cooperative and noncooperative multi-aircraft systems. Particularly, we focus on aerial combat, air traffic control problem, and control of multiple UAVs. We propose two different optimization-based approaches and their implementations with civil and military applications. In the first method, we benefit from hybrid system theory to present the input space of decision process. Then, using a problem specific evaluation strategy, we formulate an optimization problem in the form of integer/linear programming to generate optimal strategy. As a second approach, we design a method that generates control inputs as continuous real valued functions instead of predefined maneuvers. In this case, we benefit from differential flatness theory and flatness-based control. We construct optimization problems in the form of mixed-integer linear programming (MILP) and non-convex optimization problem. In both methods, we also benefit from game theory when there are competitive decision makers. We give the details of the approaches for both civil and military applications.

We present the details of the hybrid maneuver-based method for air-to-air combat. We use the performance parameters of F-16 to model the aircraft for military applications. Using hybrid system theory, we describe the basic and advanced fighter maneuvers. These maneuvers present the input space of the aerial combat. We define a set of metrics to present the air superiority. Then, the optimal strategy generation procedure is formulated as a linear program. Afterwards, we use the similar maneuver-based optimization approach to model the decision process of the air traffic control operator. We mainly focus on providing a scalable and fully automated ATC system and redetermining the airspace capacity via the developed ATC system. Firstly, we present an aircraft model for civil aviation applications and describe guidance algorithms for trajectory tracking. These model and algorithms are used to simulate and predict the motion of the aircraft. Then, ATCo's interventions are modelled as a set of maneuvers. We propose a mapping process to improve the performance of separation assurance and formulate an integer linear programming (ILP) that benefits from the mapping process to ensure the safety in the airspace. Thereafter, we propose a method to redetermine the airspace capacity. We create a stochastic traffic environment to simulate traffics at different complexities and define breaking point of an airspace with regards to different metrics. The approach is validated on real air traffic data for en-route airspace, and it is shown that the designed ATC system can manage traffic much denser than current traffic.

As a second approach, we develop a method that generates control inputs as continuous real valued functions instead of predefined maneuvers. It is also an optimization-based approach. Firstly, we focus on control of multi-aircraft systems. We utilize the

STL specifications to encode the missions of the multiple aircraft. We benefit from differential flatness theory to construct a mixed-integer linear programming (MILP) that generates optimal trajectories for satisfying the STL specifications and performance constraints. We utilize air traffic control tasks to illustrate our approach. We present a realistic nonlinear aircraft model as a partially differentially flat system and apply the proposed method on managing approach control and solving the arrival sequencing problem. We also simulate a case study with a quadrotor fleet to show that the method can be used with different multi-agent systems. Afterwards, we use the similar flatness-based optimization approach to solve the aerial combat problem. In this case, we benefit from differential flatness, curve parametrization, game theory and receding horizon control. We present the flat description of aircraft dynamics for military applications. We parametrize the aircraft trajectories in terms of flat outputs. By the help of game theory, the aerial combat is modeled as an optimization problem with regards to the parametrized trajectories. This method allows the presentation of the problem in a lower dimensional space with all given and dynamical constraints. Therefore, it speeds up the strategy generation process. The optimization problem is solved with a moving time horizon scheme to generate optimal combat strategies. We demonstrate the method with the aerial combats between two UAVs. We show the success of the method through two different scenarios.

## KOOPERATİF VE KOOPERATİF OLMAYAN ÇOKLU UÇAK SİSTEMLERİNİN OPTİMİZASYON TABANLI KONTROLÜ

### ÖZET

Hava muharebesi meydan okumanın dinamik doğası gereği uzmanlaşmanın en zor olduğu uçuş türüdür. Zıt amaca sahip farklı karar vericilerin aksiyonları rekabetin sonucunu etkiler. Bir savaş pilotunun karar süreci mevcut durumun değerlendirilmesi, düşman unsurun stratejisinin tahmini ve saldırı stratejisinin belirlenmesini kapsar. Uygulamada hava muharebe taktik ve teknolojilerinin analizi pahalıdır. Bu sebeple muharebe taktik ve teknolojilerinin analizi için matematiksel modellerden faydalanmak gündeme gelmektedir. Optimizasyon, oyun teorisi ve simulasyon bu amaçla en çok yararlanılan araçlardır. Havadan havaya muharebe modellemesi bu sebeplerle ele alınmaktadır. Muharebe sırasında optimum stratejileri üretebilen bir yöntem geliştirildikten sonra bu yöntem karar-destek sistemi olarak, otonom muharebe yapmak için, pilotların eğitilmesi amacıyla ve potansiyel muharebe senaryolarının değerlendirilmesi için kullanılabilir. Sivil havacılığa bakıldığında uçuş sayısının önümüzdeki 15 yıl içerisinde iki katına çıkması beklenmektedir. Fakat hava trafik yönetimindeki alt yapılar bu duruma hazırlıklı değildirler. Örneğin, mevcut hava sahası kapasitelerinin de hava trafiğindeki bu artışa uyum sağlamaları gerekir. Hava sahası kapasitelerinin artışının önündeki temel engellerden biri hava trafik kontrolörlerinin iş yüküdür. Bu sebeple gelecek hava trafik yönetimi operasyonlarında geliştirilmiş yüksek seviye otomasyon desteğinin gerekliliği gündeme gelmektedir. Görüldüğü gibi hem askeri hem de sivil havacılık operasyonlarında otomasyon desteği ve bazı sistemlerin otonom hale getirilmesi artık kaçınılmazdır. Bu tez kapsamında hem askeri hem de sivil uygulamalarda kullanılacak benzer formdaki problemleri çözen optimizasyon tabanlı metodlara odaklanılacaktır. Örneğin, sivil havacılıkta uçuş boyunca uçakların birbirlerine belirli bir mesafeden fazla yaklaşmamları istenirken, askeri havacılıkta esas amaç düşman uçağa sokularak onu bertaraf etmektir. Bu iki problem matematiksel açıdan aynı formda ifade edilebilirler. Bu tarz farklı öznelerin süreç içerisinde bulunduğu ve belirli amaçlarla sürecin çözümlenmeye çalışıldığı problemler oyun tarzında problemlerdir ve oyun teorisi kullanılarak değerlendirilebilmeleri mümkündür. Bu tez kapsamında havacılıktaki bu tarz problemlerin çözümlenebilmesi için iki farklı yöntem geliştirilerek ilgili süreçler için karar-destek sistemi oluşturulacaktır. Tez kapsamında özellikle havadan havaya muharebe ve hava trafik kontrolü problemlerine odaklanılmıştır.

İlk yaklaşımda bahsedilen problemlerin çözülebilmeleri için hibrit sistem teorisi ve oyun teorisinden faydalanılmıştır. Hibrit sistem teorisi kullanılarak ilgili problem için kullanılacak manevra kütüphanesindeki manevralar hibrit otomat olarak tasarlanmıştır. Amaç fonksiyonları ilgili oyuna göre türetilmiş ve oyun teorisinden türetilen değerlendirme stratejileri tasarlanmıştır. Karar sürecinin girdi uzayı tanımlanan uçak manevraları kümesiyle ifade edilmiştir. Bu manevralar oyuncunun olası seçimlerini

belirlemektedir. Bu yaklaşımın kullanılmasıyla problemlerin çözümleri gerçeğe daha uygun olmaktadır ve tasarlanan algoritmalar daha ölçeklenebilir olmaktadır. Otonom bir simülasyon ortamı tasarlanırken oyuncuların seçimlerini belirleme sürecinde iki temel unsur esastır. Bunlardan ilki oyuncuların amaçlarının sayısallaştırılması, ikincisi sayısallaştırılan parametrelerin değerlendirilme stratejisidir. Araştırmanın bir bölümü sayısallaştırılma sürecini kapsamaktadır. Bu süreç ilgili oyuna bağlıdır ve oyundan oyuna değişiklik arz etmektedir. İkinci kısım olan değerlendirme stratejisinin türetilmesi için oyun teorisi yaklaşımlarından faydalanılmıştır. Uçak manevraları, amaç fonksiyonları ve değerlendirme stratejilerinin birleştirilmesi ile ilgili problemler optimizasyon problemlerine dönüştürülmüştür. Optimizasyon problemlerinin çözülmesiyle ilgili oyun için optimum aksiyonlar üretilmektedir. Bahsedilen ilk yaklaşım hem havadan havaya muharebe hem de hava trafik kontrolü problemleri için detaylandırılmıştır. İlk problem olan havadan havaya muharebe için F-16 uçağının performans verileri kullanılarak bir uçak modeli oluşturulmuştur. Bu model vasıtasıyla temel savaş manevraları (örneğin ton, fiç ton, immelman dönüşü) ve bazı gelişmiş savunma (örneğin sarmal yükselme, sarmal dalış) ve hücum manevraları (örneğin alçak yo-yo, yüksek yo-yo) için hibrit sistem modelleri türetilmiştir. Bu manevralar muharebedeki pilotun olası seçimlerini sunmaktadır. Hava üstünlüğünü ifade eden bazı metrikler kullanılarak tercihler oyun teorisi aracılığı ile tanımlanmıştır. Ardından, optimum stratejiyi üreten bir doğrusal program formülize edilmiştir. Manevra tabanlı optimizasyon yaklaşımı hava trafik kontrolörünü otonom hale getirmek için de detaylandırılmıştır. İlk olarak sivil havacılık uygulamaları için uçak hareketini simule etmekte ve gelecek rotaları öngörmekte kullanılabilecek uçak modeli ve güdüm yöntemleri geliştirilmiştir. Bu süreçte hibrit sistem teorisinden yararlanılmıştır. Ardından, hava trafik kontrolörünün olası aksiyonlarını yansıtan bir manevra seti oluşturulmuştur. Tasarlanan haritalama süreci ile hava sahası ayrıklaştırılmış ve bu yapı optimizasyon problemini formülize ederken kullanılmıştır. Optimizasyon problemi tamsayı doğrusal programlama (integer linear programming) formunda inşa edilmiştir. Önerilen yapı sayesinde çakışma tespiti ve çözümü yüksek performansla hızlı bir şekilde gerçekleştirilebilmektedir. Ardından, hava sahası kapasitesinin yeniden belirlenmesi ile ilgili bir yöntem üzerinde durulmuştur ve tasarlanan optimizasyon tabanlı yapı ile hava sahasının yönetilmesi halinde kapasitenin mevcut durumunun çok üzerine çıkarılabileceği gösterilmiştir.

İkinci yaklaşım olarak kontrol girdilerini önceden tanımlanmış manevralar yerine sürekli zamandaki fonksiyonlar şeklinde üreten bir method üzerinde çalışılmıştır. Bu yöntem de bir optimizasyon tabanlı yaklaşıma dayanmaktadır. Bu yaklaşımda ilgili problemlerin çözülebilmesi için diferansiyel düzlük (differential-flatness) teorisi ve oyun teorisinden faydalanılmıştır. Optimizasyon problemleri karışık tamsayı programlaması (mixed-integer programming) ve konveks olmayan optimizasyon formunda ifade edilmiştir. Bu yöntem hem çoklu uçakların kooperatif kontrolü hem de havadan havaya muharebe problemlerinde otonom strateji üretilmesi için detaylandırılmıştır. Çoklu uçakların kontrolü sırasında uçakların görevleri zamansal sinyal mantık (signal temporal logic) tanımlamaları ile belirtilmiştir. Diferansiyel düzlük (differential-flatness) teorisi ve eğri parametrisasyonu aracılığıyla problem bir karışık tamsayı programlamasına (mixed-integer programming) dönüştürülmüştür. Optimizasyon probleminin çözümüyle kontrol girdileri sürekli zamanlı fonksiyonlar olarak üretilmektedir. Bu sayede hem görevleri sağlayan hem de performans limitlerine riayet eden uygulanabilir (feasible) rotalar elde edilmektedir. Sunulan

yöntemin uygulaması olarak yaklaşımdaki hava trafik kontrolü problemi ve quadrotor filosunun kontrolü problemleri çözümlenmiştir. Böylece metodun farklı çoklu uçak sistemleri ile uygulanabilir olduğu gösterilmiştir. Ardından, diferansiyel düzlük (differential-flatness) teorisine dayanan bu yaklaşım havadan havaya muharebe problemini çözümlenecek biçimde detaylandırılmıştır. Askeri havacılık uygulamalarında kullanılacak bir uçak modelinin diferansiyel düzlük formu türetilmiştir. Bu durumda uçak hareketi belirli bir değişken seti ve türevleri cinsinden sunulabilmektedir. Bu sayede uçağın gidebileceği rotalar diferansiyel denklem setinin integralinin alınmasına gerek kalmadan değerlendirilebilmektedir. Ardından uçuş rotaları eğriler vasıtasıyla parametrize edilmiş ve tanımlanan değişken seti ve türevleri cinsinden yazılmıştır. Oyun teorisinden faydalanarak havadan havaya muharebe parametrize edilen eğriler vasıtasıyla bir optimizasyon problemi olarak modellenmiştir. Bu yöntem verilen ve uçuş dinamiğinden gelen tüm kısıtları kullanarak problemin daha düşük boyutlu bir uzayda sunulmasına olanak sağlamaktadır. Bu sayede strateji üretme süreci hızlanmaktadır. Oluşturulan optimizasyon problemi hareketli bir zaman ufkunda oyun sonlanana kadar çözdürülmekte ve böylece muharebe stratejileri üretilmektedir. Sunulan yöntem insansız hava araçlarının savaştığı bir simülasyon ile uygulanmış ve yöntem değerlendirilmiştir.



## 1. INTRODUCTION

Decision making refers to the act of evaluating several alternatives and choosing the one most likely to achieve one or more goals [1]. Common examples include deciding for whom to vote, what to eat or buy, and which college to attend. Decision making plays a key role in many professions, such as public policy, medicine, and management. The related concept of judgment refers to the use of information, often from a variety of sources, to form an evaluation or expectation. In decision theory, an attempt is made to combine the sample information with other relevant aspects of the problem in order to make the best decision. In addition to the sample information, two other types of information are typically relevant [2]. The first is a knowledge of the possible consequences of the decisions. Often this knowledge can be quantified by determining the loss that would be incurred for each possible decision. The other one is prior information.

First consider the simple case of a single decision maker that must make the best decision. This leads to a familiar optimization problem, which is formulated as follows. A nonempty set  $U$  called the action space. Each  $u \in U$  is referred to as an action. A function  $L : U \rightarrow R \cup \{\infty\}$  called the cost function. A strategy simply consists of selecting the best action such that  $u^* = \arg \min_{u \in U} \{L(u)\}$ . Now suppose that there is a second decision maker that is a clever opponent that makes decisions in the same way that the first one would. This leads to a symmetric situation in which two decision makers simultaneously make a decision, without knowing how the other will act. This leads to game theory, in which all decision makers can be called players.

Game theory is the formal study of conflict and cooperation. Game theoretic concepts apply whenever the actions of several agents are interdependent. These agents may be individuals, groups, firms, or any combination of these. The internal consistency and mathematical foundations of game theory make it a prime tool for modeling and designing automated decision-making processes in interactive environments such as air combat and air traffic control in aviation. The process of formally modeling a

situation as a game requires the decision-maker to enumerate explicitly the players and their strategic options, and to consider their preferences and reactions.

In all game theoretic models the basic entity is a player. A player may be interpreted as an individual or as a group of individuals making a decision. Once we define the set of players, we may distinguish between two types of models: those in which the sets of possible actions of individual players are primitives and those in which the sets of possible joint actions of groups of players are primitives. Sometimes models of the first type are referred to as “noncooperative”, while those of the second type are referred to as “cooperative”.

The scope of the thesis is to generate optimization-based game theoretic approaches to analysis, evaluate and solve the game like problems in aviation such as determination of air superiority in an air combat and generation of solution strategies to prevent conflict between aircraft, which is the primary purpose of the air traffic control.

## **1.1 Purpose of Thesis**

Unmanned systems are becoming parts of civil and military aviation. They are replacing the manned systems in various aerial missions. However, it is a challenging issue to replace a manned platform by an unmanned system in aerial combat because of the dynamic nature of combat. An unmanned aerial vehicle (UAV) can be controlled remotely in an aerial combat. However, the UAV will be in a disadvantageous position against a manned platform because of the limited situation awareness of the UAV pilot, which originates from current pilot-vehicle interface technology. This limitation can be overcome via the automated combat maneuvering. Besides, an operator could improve vehicle performance and manage multiple UAVs in aerial combat by the help of automated combat maneuvering.

In spite of the advances made in missile technology and long range radar, modern fighter aircraft (e.g., F/A-22, F-15, and F-35) are still designed for close combat and military pilots are trained to effectively use in one-on-one aerial combat. The aerial combat has a vital role in military aviation. In air-to-air combat, the primary role of a combatant is destroying the enemy aircraft that is achieved by establishing air superiority over a battlefield. The key element that brings victory is always considered



as achieving and maintaining air superiority that are affected by several factors such as performance of the aircraft, skill of pilot and combat tactics [3], [4]. The decision making, which contains evaluation of these factors, is a challenging subject due to the multiple decision makers that have conflicted interests. Besides, analysis of aerial combat tactics and technologies are expensive tasks in practice. Hence, we benefit from mathematical models, which are based on optimization, game theory and control theory, to analyze combat tactics and technologies. After creating a strategy generation platform for aerial combat, this platform can be used for decision support, autonomous aerial combat, training of pilots and analyzing potential combat scenarios.

In civil aviation, it is expected that the number of commercial flights will almost double from 26 million to 48.7 million, and 13.5 trillion passenger-kilometer will be flown by 2030, which is almost three times what is flown by airlines today [5]. Therefore, the airspace capacity should also increase accordingly to accommodate the increase in the air traffic volume. One of the major barriers in the expansion of the airspace capacity is the workload on the air traffic controllers (ATCo). Therefore the future air traffic management (ATM) operations are going to require enhanced and high-level automation support for routine decision-making procedures, which contain the conflict detection and separation assurance.

In this thesis, we mainly focus on developing methods that ensure autonomous control of cooperative and noncooperative multi-aircraft systems. Particularly, we focus on aerial combat and air traffic control problems. We propose two different optimization-based approaches and present how these approaches can be implemented with civil and military applications.

## **1.2 Literature Review**

### **1.2.1 Aerial combat**

The modeling of aerial combat game has been studied from different aspects. The pursuit-evasion game problem is well known version of these modeling perspectives, where the combatants have fixed roles. In the papers [6], [7], the authors generate reachable sets via Hamilton-Jacobi-Isaacs (HJI) partial differential equation to find solution for the pursuit-evasion game. The study [7] shows that the level set method

can be used to calculate the backward reachable set for a two player differential game. However, this kind of methods suffers from curse of dimensionality. They are computationally inefficient. There are also alternative approaches to construct reachable sets. The zonotope-based reachability algorithms such as [8] and the methods based on approximations that use oriented rectangular hulls such as [9] have been proposed. These methods have not been implemented for the the pursuit-evasion game. However, if they are implemented, the inherited problems will still continue with these methods. The role of the combatant will be fixed, and the generated results will be a set instead of an optimal trajectory that can be used in an autonomous aerial combat. The pursuit-evasion game has been also evaluated using model predictive control. The studies [10], [11] present a nonlinear model predictive tracking control algorithm that provides evasive maneuvers to a fixed-wing aircraft when confronted by an airborne adversary of a priori type. In these studies, it is needed to encode proven aircraft maneuvering tactics into the cost functions, because the required behaviours cannot be produced by the algorithm. Besides, the method is not capable of switching between pursuit and evasion roles. The rule-based structure is an alternative to model the air combat as in study [12]. In this study, a rule-based logic is used to generate strategies. The influence diagram is another approach to solve the problem as presented in [13] that contains the modeling of the decision-making process of the pilot for one-on-one aerial combat using an influence diagram. In the works [14], [15], the combat between helicopters is simulated via a game theoretic approach that tries to select optimal maneuver from a set of elemental maneuver such as steady flight, max load factor turn, etc. The author of [16] uses approximate dynamic programming approach to produce maneuver inputs such as roll left, roll right and maintain the current bank for aerial combat on horizontal plane. However, none of the work presented above use the real air combat maneuvers as input space to determine the choice of pilot, most of time they use elemental maneuvers such as turn, go along etc. or subset of these elemental maneuvers as input space. And, none of them is capable of simulating two-vs-one air combats. Besides, when the input space is not discretized via maneuvers such as elemental or real air combat, the computation time is a challenging issue.

### 1.2.2 Air traffic control

There are several studies in the literature associated with conflict detection and separation assurance problems. All of these studies focus on either free flight concept or ground based operation. The free flight concept does not contain a centralized traffic controller and aircraft are responsible from their conflict resolutions that are performed airborne. The ground based operation corresponds to a centralized air traffic control mechanism. The main operational difference between the free flight and centralized mechanism is that the flight path intent information is not used in the free flight approach, whereas it is utilized in the ground based operation during conflict detection and resolution processes.

In the literature, some studies are centred on conflict detection problem without conflict resolution process. Most of these works are based on the free flight concept. The work in [17] defines protected zones around aircraft, and then uses these zones for conflict detection. These zones are propagated for a fixed time horizon to check the potential collisions. The propagation process is limited to simplified aircraft dynamics. Shewchun et al. [18] consider more complex dynamics, such as along track and cross track fluctuations, which translates to bearing and acceleration uncertainties. The authors benefit from Linear Matrix Inequalities and positive semi-definite programming to solve the conflict detection problem. The probability theory is also studied in free flight to present the conflict detection as a probability distribution. The work in [19] models the trajectory prediction error as a normal distribution, with zero mean and a covariance matrix with eigenvectors in the along-track and cross-track directions. The conflict probability is computed in the horizontal plane according to these stochastic error dynamics. In the study [20], the authors utilize the predicted trajectories during conflict detection process in horizontal plane by focusing on the ground based operation unlike the aforementioned probabilistic approach. They focus on finding possible conflict locations. They use the idea that the conflict detection process can be speed up via a transformation from trajectories to bins. Vink et al. [21] also utilize the predicted trajectories for conflict detection. In addition, the authors present unpredictable aircraft dynamics as uncertainty regions around the trajectories and the conflict detection can be achieved for 3D trajectories. Both studies assume that predicted trajectories are received from another source such as Flight Data Processing

System (FDPS), hence they mainly work on generated trajectories without using an aircraft model. The probabilistic methods have also been applied to the ground-based approaches. For instance, the work [22] constructs a mathematical model by solving a partial differential equation with Dirichlet boundary conditions to calculate the conflict probability. Note that the aforementioned works are limited to conflict detection, while the automation of the air traffic control system would require algorithms that can perform both conflict detection and resolution.

There are also some methods that focus only on the separation assurance without conflict detection. The work in [23] benefits from the potential field method to ensure the safety in free flight. The approach is computationally cheap, however the potential field methods have inherent limitations, such as being stuck in the local minima and oscillating solutions in the presence of narrow passages and dense environments [24]. Therefore, it can cause loss of separation in realistic ATM scenarios. Tomlin's works [25], [26] present hybrid system frameworks for conflict resolution in horizontal plane. Although the paper does not directly address the conflict detection problem, the alert zones and protected zones terms are defined to acknowledge the issue. The resolution maneuvers are generated by solving the Hamilton-Jacobi-Bellman (HJB) equation. The authors use simplified aircraft dynamics and consider only conflict resolution in the horizontal plane because of high computational complexity of solving the HJB equation. In addition to these studies, the work [27] also develops a hybrid automaton for separation assurance in the horizontal plane. The aforementioned work is mainly about air traffic flow, however the presented automaton is used to ensure the separation. Overall, these methods have limited scalability because of the high computational demand, thus it is difficult to use these algorithms in real airspace. Furthermore, generating resolution maneuvers only on horizontal plan (2D) cause a bottleneck in the airspace capacity and it is far from real ATM operation. In many practical situations, vertical maneuvers might be the only option to ensure the safety. Hence, it is important for an automated ATC system to operate in 3D for realistic applications.

There are also several studies that address conflict detection and resolution together. In the study [28], the authors use quadratic programming to model the conflict resolution process, and then the optimization problem is solved via semi-definite programming

combined with a randomization scheme. The algorithm is capable of conflict detection and separation assurance and also uses flight plans during optimization process. However, the algorithm has exponential complexity with respect to the number of aircraft, which limits its applicability to large-scale ATM scenarios. The work [29] follows a similar approach that performs separation assurance via solving a quadratic program. The problem is also formulated as a mixed-integer programming in the study [30]. In these two studies, the problem is limited to the horizontal plane and simplified aircraft dynamics is used to construct the optimization problem. Overall, it can be observed that most of the algorithmic approaches either tend to use simplified aircraft dynamics and limit the conflict resolution maneuvers to horizontal plane for the sake of reducing the computational complexity, or tend to incorporate more realistic conflict resolution maneuvers and aircraft dynamics by sacrificing computational complexity and hence limiting the scalability. There are also some studies that use heuristics to improve the scalability. In the works [31], [32], the authors create algorithms that emulate the decision process of an air traffic controller based on the finite state machine. The developed algorithms detect conflicts and ensure separations in 3D without an optimization process.

In addition to the algorithmic works above, there are also software tools developed for providing decision support. The tool developed by Yang and Kuchar [33] is based on the free flight concept. The authors benefit from the algorithm in [19] to achieve conflict detection. The tool supports the decision process of the pilot with a probability map of conflicts and suggests elemental maneuvers such as heading change, speed change, climb or descent to resolve the potential conflicts. For the ground-based control systems, NASA's Center-TRACON Automation System (CTAS) [34] and MITRE's URET [35] are developed for providing decision support to air traffic controllers. Both tools have conflict detection capability without resolution advisory and use flight plans to assist the conflict detection process.

The other issue is the estimation of the airspace capacity, which is limited by controller workload. In the literature, there are several studies focus on evaluating the controller workload via complexity metrics. Complexity is described as "hard to separate, analyze, or solve" harmonizing with most people's intuition [36]. According to [37], complexity is composition of situation complexity and cognitive complexity. Situation

complexity is mainly about configuration of traffic, whereas cognitive complexity is closely related to the decision making process. Cognitive complexity is also affected from situation complexity. According to [36], [38], complexity drives the workload of an air traffic controller that is a limiting factor on the airspace capacity. The studies in the literature are mainly focused on situation complexity. Among all the complexity metrics given in the literature, traffic density is the most basic and most associated with complexity. As stated by [36], the body of literature seems at the same time to praise the concept of traffic density as the best available indicator of complexity, and to criticize it that it does not capture the richness of what controllers find complex. As an alternative, the model of Dynamic Density (DD) is described as aggregation of different complexity metrics. DD models are presented as either linear combination of different complexity metrics [39], [40] or non-linear combination of complexity factors as in work [41], which uses neural networks for non-linear regression. These studies represent several complexity metrics to construct more detailed mathematical model representations of situation complexity, where traffic density is one of complexity factors. Besides them, the work [42] proposes fractal dimension as a measure of the situation complexity of the traffic pattern. The fractal dimension evaluates the number of degree of freedom in the traffic flow. In the [43], air traffic complexity is defined as the required control effort to ensure the safety when a new aircraft enters into the airspace. The authors describe an input-output framework and present a complexity map to evaluate the complexity of a traffic situation. Despite this lack of inclusion in metrics, the airspace structure is considered an important factor for understanding complexity.

### **1.2.3 Mission specification with temporal logic**

Mission planning and control of multi-aircraft systems involve several temporal and logical constraints. These constraints can naturally be specified using Temporal Logic (TL), a system of rules and symbolism. Signal Temporal Logic (STL) is an extension of TL in which temporal operators also contain timing constraints for specifying properties of real-valued signals [44]. When dealing with continuous systems, STL is convenient to specify these constraints.

Temporal logic has been studied as a formal language for specifying system behaviors and complex tasks. Linear Temporal Logic (LTL) has been employed as a tool for specifying the restrictions in discrete-time dynamical systems [45]. The LTL specifications are represented as mixed-integer linear constraints to generate optimal control strategy for a discrete-time system via Mixed-Integer Programming (MIP). STL specifications have been encoded as mixed-integer linear constraints to specify system's properties such as safety and response for model predictive control of discrete-time systems [46]. In both the above studies, it is required to add decision variables at every time step for a specific temporal logic constraint. Therefore, these approaches do not scale well. There are also studies that present heuristics in order to reduce MIP complexity in these approaches. The authors of [47] propose a heuristic to add constraints when necessary instead of adding auxiliary decision variables at every time step. There are also studies such as [48] that evaluate the problem as a non-convex optimization problem to generate trajectories with Metric Temporal Logic (MTL) specifications. Consequently, the system can have nonlinear dynamics but it should be discretized. Stochastic heuristics have also been proposed for finding system behaviors that falsify a temporal logic property. For example, the study [49] presents a Monte-Carlo technique for finding counterexamples to MTL properties. The algorithm can be applied to nonlinear dynamical systems in continuous-time. However, it is a sampling-based method without guarantees and it is hard to apply this approach to multi-agent systems. Temporal logic specifications have also been studied with multi-agent systems. The desired behavior of a group of agents is specified with variants of TL in the studies [50], [51] that evaluate the problem in a grid-based environment using discrete abstractions of the system dynamics. It is common practice in many studies to use discrete abstractions when dealing with temporal logic for multi-agent systems. These studies provide correctness guarantees for the discrete behavior. However, it is a simplified version of the real system. In the study [52], the authors focus on mission planning of multi-quadrotor systems. Using a trajectory generator, they construct a non-convex optimization problem to obtain trajectories that satisfy the STL specifications in continuous-time. However, the study mainly focuses on quadrotors, and it decouples the equations of motion along three orthogonal axes. Because of this decoupling, the presented method cannot be implemented to fixed-wing aircraft. Moreover, the study mainly focuses on the position information,

and STL specifications cannot be given in terms of the angles. However, sometimes it may be necessary to specify orientations. For example, in arrival sequencing, it may be necessary to align the aircraft with respect to the runway's direction.

### 1.3 Overview

As mentioned before, the primary purpose of this thesis is the development of the methods that ensure autonomous control of cooperative and noncooperative multi-aircraft systems. However, the proposed methods can also be used in semi-autonomous systems for providing decision support to the human operators. Especially, we focus on aerial combat and air traffic control problems. We propose two different optimization-based approaches and their implementations with civil and military applications. In the first method, we benefit from hybrid system theory to present the input space of decision process. Then, using a problem specific evaluation strategy, we formulate an optimization problem in the form of integer/linear programming to generate optimal strategy. As a second approach, we design a method that generates control inputs as continuous real valued functions instead of predefined maneuvers. In this case, we benefit from differential flatness theory and flatness-based control. We construct optimization problems in the form of mixed-integer linear programming (MILP) and non-convex optimization problem. In both methods, we also benefit from game theory when there are competitive decision makers. Overall, Chapter 2 and Chapter 3 explains the first method for air-to-air combat and air traffic control applications, respectively, whereas Chapter 4 and Chapter 5 consist of the details of the second method.

Chapter 2 presents the hybrid maneuver-based method to assist the decision making procedure in an air-to-air combat. It is mainly about engagement phase of the aerial combat. It consists of the aircraft performance model that is implemented with F-16 performance parameters and air combat maneuvers. The basic fighter maneuvers (e.g. aileron roll, barrel roll, immelmann) and some advanced defensive (e.g. vertical spiral, spiral dive) and offensive maneuvers (e.g. low yo-yo, high yo-yo) are described as hybrid system models. These maneuvers are possible choices of pilots in an air combat. The preferences are defined by a game theoretic approach using some metrics that present the air superiority. Then, a linear program is formulated to generate optimal



strategy. Contrary to existing approaches, the real aerial combat maneuvers are used to determine the choices of the combatants. While previous studies use elemental maneuvers such as turn, go along etc. or a subset of these elemental maneuvers as input space, the proposed method use more realistic air combat maneuvers. And, the proposed approach is capable of simulating N-vs-M air combats.

In Chapter 3, we use the similar maneuver-based optimization approach presented in Chapter 2 to solve the air traffic control problem. Chapter 3 mainly focuses on providing a scalable and fully automated ATC system that can be integrated seamlessly into the existing ATM system and redetermining the airspace capacity by the help of designed ATC system. To achieve these objectives, first of all, we design an aircraft model and guidance procedure to simulate and predict the motion of the aircraft. Secondly, we model a set of maneuvers that represents the possible choices of ATCo during decision process. Then, we develop a mapping process based on discretization of the airspace to improve the performance of conflict detection and resolution and create an integer linear programming (ILP) that uses the sets and matrices derived from mapping process to ensure the safety in the airspace. Next, we focus on a methodology to redetermine the airspace capacity that consists of creating a stochastic traffic environment to simulate traffics at different complexities and defining breaking point of an airspace in terms of different metrics. In consequence of these approaches, the designed autonomous ATC system has better scalability than previous algorithmic approaches such as [26], [29], [28]. Moreover, it presents a more realistic approach to the automated air traffic control problem by generating separation maneuvers in 3D and using detailed aircraft motion models in opposition to some aforementioned studies. It also contains an optimization-based approach to improve the efficiency of the ATC process. Furthermore, the effects of the designed system on the airspace capacity are also investigated, and it is shown that the designed ATC system can manage traffic much denser than current traffic. Besides, the sub-components of the proposed system can be used individually in different ATM applications. For example, the trajectory prediction model can be used when dealing with trajectory management, or the mapping process can be utilized when dealing with mitigation of airspace congestion.

In Chapter 4, we have focused on designing an alternative method that generates control inputs as continuous real valued functions instead of predefined maneuvers. It is also an optimization-based approach to control a multi-aircraft system in which each aircraft has several mission objectives. We encode the missions of the multiple aircraft as STL specifications. Then, using differential flatness theory, we construct an optimization process to generate optimal strategies for multiple aircraft to satisfy the STL specifications, which corresponds to completing the assigned tasks. The proposed method generates control inputs as continuous real valued functions, and it generates feasible trajectories that satisfy the missions and performance limitations. We focus on air traffic control tasks using a realistic nonlinear aircraft model to illustrate our approach. We also simulate a case study with a quadrotor fleet to show the generalizability of the proposed method to other multi-agent systems. As mentioned before, the majority of the existing studies such as [45], [46], [47], [48] discretize the system dynamics to ensure the STL specifications in discrete-time. One of the contributions of this study is that the proposed method generates feasible trajectories in continuous-time that satisfy the tasks described via STL, without discretizing the system dynamics. Although some of the constraints in the optimization problem are enforced only at the sampled times, this sampling does not jeopardize continuous-time satisfiability. Compared to the existing MILP-based approaches such as [45], [46], the developed method has better scalability to deal with nonlinear system dynamics, because it is not necessary to add excessive auxiliary decision variables. The method can be used with different multi-agent systems such as a fleet of fixed-wing aircraft or multi-quadrotor systems. Because of this generalizability, it overcomes the restriction of the studies such as [52] that are developed for specific systems. The proposed method fills a gap in the literature by showing that a MILP based approach works well for the STL satisfaction of differentially flat nonlinear systems. The study presents a convenient way to use the flatness property of a nonlinear dynamical system to satisfy STL specifications. To the best of our knowledge, no other study in the literature presents a convenient way to form a MILP that is used to guarantee the continuous-time satisfiability of the STL specifications for nonlinear continuous systems. Moreover, the method enables us to use realistic nonlinear dynamical models when evaluating the complex missions of multi-agent systems, contrary to many existing studies such as [50], [51] that use discrete abstractions of the system dynamics and grid-based

environments for mission planning of multi-agent systems. The presentation of the partially-flat aircraft model that can be used in Air Traffic Management (ATM) applications is also a contribution.

In Chapter 5, we use the similar flatness-based optimization approach presented in Chapter 4 to solve the aerial combat problem. Chapter 5 presents the method to model aerial combat that is based on differential-flatness, curve parametrization, game theoretic evaluation and receding horizon control. The flat description of aircraft dynamics, which contains movements on horizontal and vertical planes, is presented that allows to represent the aircraft motion in terms of a set of specific variables and their derivatives. This property provides an opportunity to evaluate aircraft motion without integration of a set of differential equations. This leads to parametrization of aircraft trajectory in terms of these specific variables and their derivatives. By the help of game theory, the aerial combat is modeled as an optimization problem in terms of parametrized trajectories and performance constraints are also presented in this optimization scheme to generate feasible strategies for combatants. This method allows the presentation of the problem in a lower dimensional space with all given and dynamical constraints that speed up the strategy generation process. The created optimization problem is solved over a moving time horizon till the end to generate the combat strategies. For simulation purpose, we have implemented the algorithm to aerial combat between two UAVs. We demonstrated the success of the algorithm through two different scenarios. In Chapter 2, we focus on a game theoretic hybrid system modeling approach that uses real aerial combat maneuvers to obtain strategies for aerial combat. However, when the input space is not discretized via maneuvers, the computation time is a challenging issue. The proposed method in Chapter 5 overcomes this issue.



## **2. HYBRID MANEUVER-BASED FRAMEWORK FOR EVALUATION OF AIR-TO-AIR COMBAT**

### **2.1 Purpose**

The air superiority refers to constituting sovereignty on airspace over enemy's air force. When a combat team outperforms the enemy's force and gets the air superiority, it takes a chance to shoot down the opponent. Therefore, air combatants derive various strategies to get air superiority that could bring victory. However, during combat, the enemy's force also tries some strategies to take advantage. The fighter team should predict the enemy's maneuvers and define their maneuvers according to the opponent's possible strategies. Thus, it is an active process that contains the prediction of challenger's possible strategies and reevaluation of choices. In this process, the main question is how can the combat team evaluate the opponent's actions and update their strategies? Air combat modeling is about this issue. It contains the evaluation of the enemy's movements and generation of actions to get air superiority. When the modeling environment is constructed, it can be used for decision support to fighter pilots, autonomous air combat and analyzing combat scenarios.

In this chapter, we present a simulation platform and maneuver selection methodology for air-to-air combats. It consists of the aircraft performance model, aerial combat maneuvers and game theoretical decision strategy. The performance parameters of F-16 are used to implement the aircraft dynamics. The hybrid system theory underlies the modeling methodology of the aerial combat maneuvers, which are the possible options of the combatants. The decision strategy is derived from security policy approach in game theory using some metrics that present the air superiority. The definitions of these metrics and evaluation strategy are presented. The verification of the models and developed decision processes are given through the simulations.

## 2.2 Hybrid Automaton

A hybrid automaton is a model and specification language for hybrid system. It can be used to evaluate the system dynamics with phased operations.

A hybrid automaton  $H$  is a collection  $H = (Q, X, f, I, D, E, G, R)$  [53], [54], with

- $Q$  is a set of discrete states
- $X$  is a set of continuous variables
- $f : Q \times X \rightarrow TX$  is a vector field
- $I \subseteq Q \times X$  is a set of initial states
- $D : Q \rightarrow P(X)$  is a domain
- $E \subset Q \times Q$  is a set of discrete transitions
- $G : E \rightarrow P(X)$  assigns to each  $e = (q, q') \in E$  a guard
- $R : E \times X \rightarrow P(X)$  assigns to each  $e = (q, q') \in E$  and  $x \in X$  a reset relation

where  $P(X)$  indicates the power set or set of all subsets of  $X$ . Hybrid automata define possible evaluations of their state. Roughly speaking, an automaton starts from an initial value  $(q_0, x_0) \in I$  and continuous variable  $x$  is updated according to differential equation

$$\dot{x} = f(q_0, x) \quad (2.1)$$

$$x(0) = x_0 \quad (2.2)$$

the discrete state  $q$  remains constant as  $q(t) = q_0$  until continuous variable  $x$  reaches the guard  $G(q_0, q_1) \subseteq \mathbb{R}^n$  of some edge  $(q_0, q_1) \in E$ . Then, the discrete state changes value to  $q_1$  and the continuous variable may get reset to some value in  $R(q_0, q_1, x) \subseteq \mathbb{R}^n$ . After this discrete transition, continuous evaluation resumes and the whole process is repeated.

## 2.3 Aircraft Model and Maneuvers for Military Aviation

Aircraft performance model and hybrid descriptions of air combat maneuvers that will be used as input space in simulation environment are presented in this section.

### 2.3.1 Aircraft dynamics

A point mass model (PMM) is used for modeling the aircraft dynamics. The model is a nonlinear dynamical system with three control inputs ( $[\delta, \mu, \gamma]$ ) and five state variables ( $[x_1, x_2, h, V, \psi]$ ). The equations of aircraft motion are:

$$\dot{x}_1 = V \cos(\psi) \cos(\gamma) \quad (2.3)$$

$$\dot{x}_2 = V \sin(\psi) \cos(\gamma) \quad (2.4)$$

$$\dot{h} = V \sin(\gamma) \quad (2.5)$$

$$\dot{V} = -\frac{C_D S \rho V^2}{2m} - g \sin(\gamma) + \frac{T \cos(\alpha)}{m} \quad (2.6)$$

$$\dot{\psi} = \left( \frac{C_L S \rho V}{2m} + \frac{T \sin(\alpha)}{mV} \right) \frac{\sin(\mu)}{\cos(\gamma)} \quad (2.7)$$

where the thrust ( $T$ ) is generated according to throttle level ( $\delta$ ). In this study, performance parameters of F-16 are used for simulations. The engine model for F-16 is presented in NASA report [55] and Steven's book [56]. It has an after-burning turbofan engine. The control command is the throttle level ( $\delta$ ) that has a value between 0 and 1. When it comes to 0.77, engine reaches the military power level. There is a linear relationship between throttle level and commanded engine power level, with a change in slope when it reaches military power level. In original model, commanded engine level has a first-order lag that is not used in our model. The thrust is calculated according to power level, Mach number and altitude. There are three look-up tables: idle, military and maximum. According to power level, thrust value is generated from these tables. These tables can be found in NASA report [55] and Steven's book [56]. In addition to engine model, there is a proportional controller that generates necessary throttle input to hold the speed at desired value, which symbolized as:

$$\delta = f_{spd}(V_{desired}) \quad (2.8)$$

In model,  $C_L$  is calculated as follows:

$$C_L = \frac{2mg \cos(\gamma)}{\rho V^2 S \cos(\phi)} \quad (2.9)$$

$C_D$  and  $\alpha$  are specified as functions of the  $C_L$  as shown in (2.10) and (2.11). The correlations between  $C_D$ - $C_L$  and  $\alpha$ - $C_L$  are extracted using aerodynamic data in NASA report [55].

$$C_D = f_{C_D}(C_L) \quad (2.10)$$

$$\alpha = f_{\alpha}(C_L) \quad (2.11)$$

For the rest of the chapter, this aircraft dynamic model is denoted as  $\dot{x} = f(x, u)$ , where  $x$  is the vector of state variables and  $u$  is the vector of control inputs.

### 2.3.2 Maneuver library

The dynamics of a combat maneuver can be encoded by a hybrid automaton.

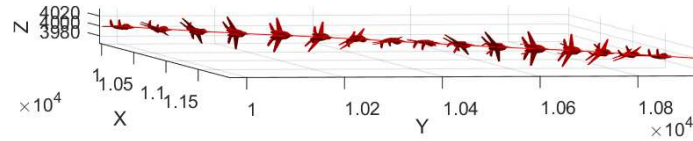
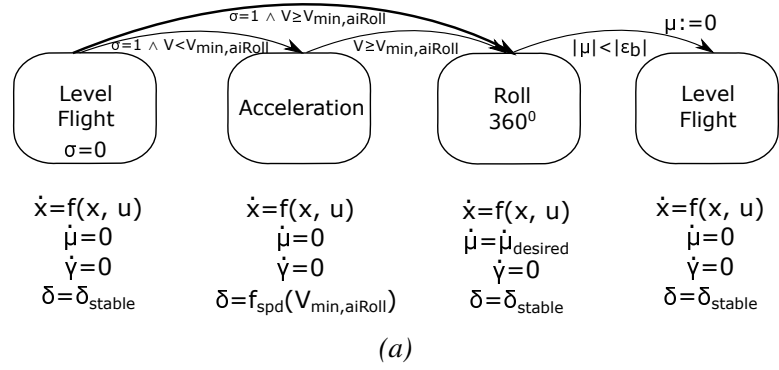
Air combat maneuvering is the combination of several simple maneuvers that must be learned before more complicated offensive and defensive maneuvers. In this study, maneuver library consists of 10 different combat maneuvers that cover basic combat maneuvers [3], [4] and some of advanced maneuvers [3], [4]. These maneuvers and hybrid automata of them are presented at the rest of this section.

#### 2.3.2.1 Aileron roll

An aileron roll refers to a full  $360^\circ$  revolution about the longitudinal axis while maintaining the flight level and flight vector. The hybrid automaton of the aileron roll and its implementation are presented in Figure 2.1.

In this hybrid automaton,  $Q = \{LevelFlight, Acceleration, Roll360^\circ, LevelFlight\}$  and the set of differential equations for each discrete state are illustrated at the below of related discrete state in Figure 2.1a. Transitions, guards and reset relations are also portrayed and reset relations are symbolized as  $:=$ . The initiation of the maneuver is controlled by a discrete input variable  $\sigma$ . The other input parameters are  $\dot{\mu}_{desired}$  and  $\delta_{stable}$ .  $\epsilon_b$  corresponds to error tolerance for bank angle and it is defined as  $\dot{\mu}_{desired}\Delta t$ .  $\delta_{stable}$  denotes the throttle value that is necessary to keep the aircraft's speed at desired value as stable.



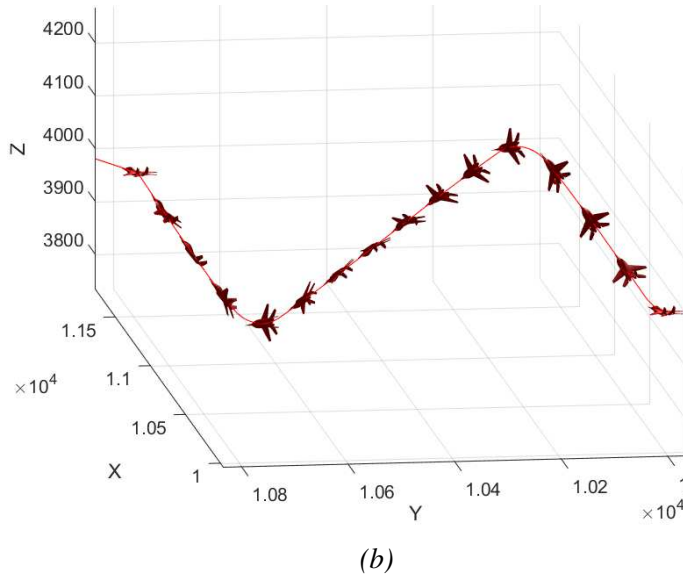
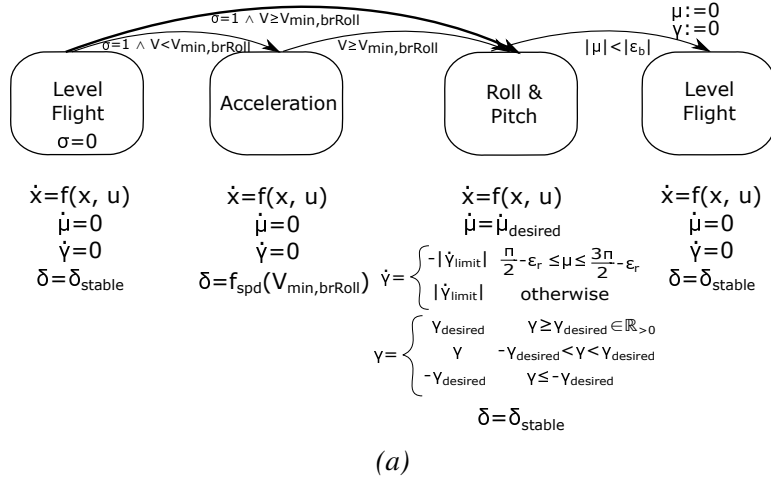


**Figure 2.1** : Aileron Roll Maneuver: (a) Hybrid Dynamics, (b) Simulation.

### 2.3.2.2 Barrel roll

In a barrel roll, the aircraft rotates both in its longitudinal and lateral axes, while in case of the aileron roll, the rotation is only about the longitudinal axis. Because of the rotation, when compared with level flight without roll, more drag is generated; consequently, the barrel roll is used in many maneuvers to reduce the aircraft's speed. The hybrid automaton of the barrel roll and its implementation are presented in Figure 2.2.

In the hybrid automaton of the barrel roll,  $Q = \{LevelFlight, Acceleration, Roll\&Pitch, LevelFlight\}$ , the differential equations and other parameters are presented in Figure 2.2a. The initiation of the maneuver is controlled by a discrete input variable  $\sigma$ . The other input parameters are  $\dot{\mu}_{desired}$ ,  $\dot{\gamma}_{limit}$  and  $\delta_{stable}$ .  $\gamma_{desired}$  is standardized as  $\pi/6$  for the barrel roll.  $\dot{\gamma}_{limit}$  corresponds to maximum or minimum value of allowed rate of path angle. It can be positive or negative. It can be taken according to aircraft performance limitations or can be defined by user that must be smaller than aircraft performance limits.

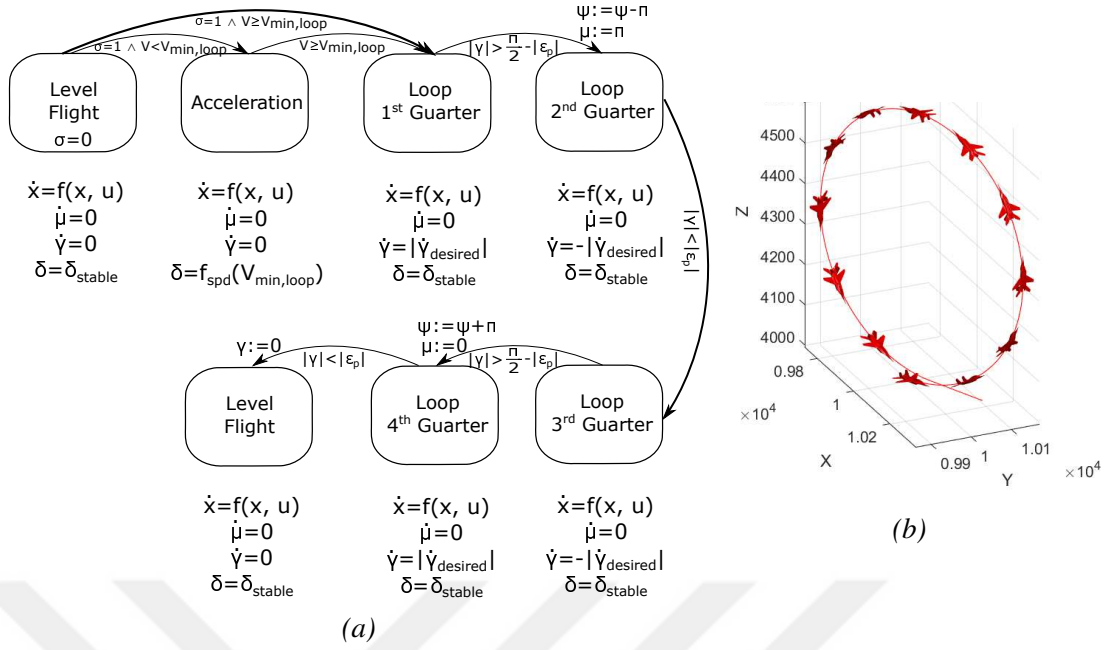


**Figure 2.2** : Barrel Roll Maneuver: (a) Hybrid Dynamics, (b) Simulation.

### 2.3.2.3 Loop

In a loop maneuver, aircraft follows a circular path in vertical plane, while the lateral axis of aircraft remains horizontal. The hybrid automaton of the loop maneuver and its implementation are presented in Figure 2.3.

In the hybrid automaton of the loop,  $Q = \{LevelFlight, Acceleration, Loop1^{st}Quarter, Loop2^{nd}Quarter, Loop3^{rd}Quarter, Loop4^{th}Quarter, LevelFlight\}$ , the rest of the information of the hybrid automaton are represented in Figure 2.3a. Input variable  $\sigma$  is used to initialize the maneuver. The other input parameters are  $\dot{\gamma}_{desired}$  and  $\delta_{stable}$ .  $\epsilon_p$  is the error tolerance for flight path angle and it is defined as  $\dot{\gamma}_{desired}\Delta t$ . Additionally, this hybrid automaton has reset relations at the transitions from  $Loop1^{st}Quarter$  to  $Loop2^{nd}Quarter$  and from  $Loop3^{rd}Quarter$  to  $Loop4^{th}Quarter$ . In the first one, heading angle is decreased by  $\pi$  and bank angle is set to  $\pi$ ; while, heading angle is



**Figure 2.3** : Loop Maneuver: (a) Hybrid Dynamics, (b) Simulation.

increased by  $\pi$  and bank angle is set to 0 for second reset relation. These reset relations simulate the instantaneous heading change because of rotation on vertical plane.

### 2.3.2.4 Break turn

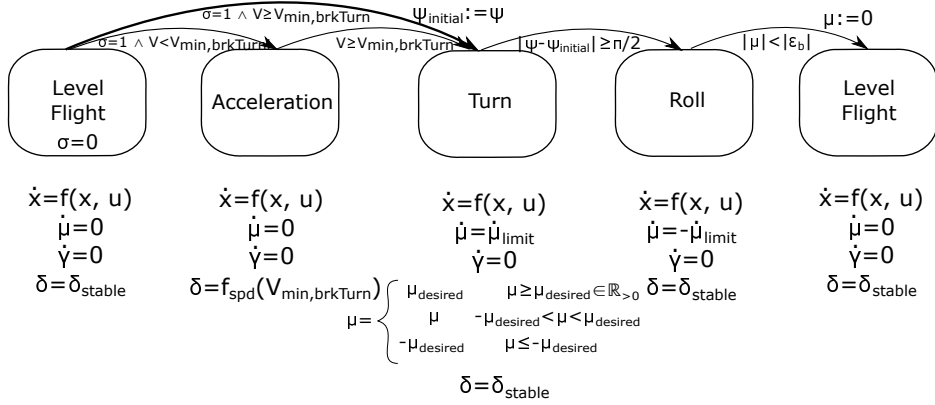
A break turn is recommended to change direction as quick as possible. Often used to avoid missiles or to outmaneuver enemy aircraft. The hybrid dynamics of the break turn as an automaton and its implementation are illustrated in Figure 2.4.

In this hybrid automaton,  $Q = \{LevelFlight, Acceleration, Turn, Roll, LevelFlight\}$  and the set of differential equations and other information are represented in Figure 2.4a. The input variable  $\sigma$  is used to initialize the maneuver. The other input parameters are  $\dot{\mu}_{limit}$  and  $\delta_{stable}$ .  $\mu_{desired}$  is standardized as  $0.89\pi/2$  for the break turn. According to sign of  $\dot{\mu}_{limit}$ , turn direction is defined.

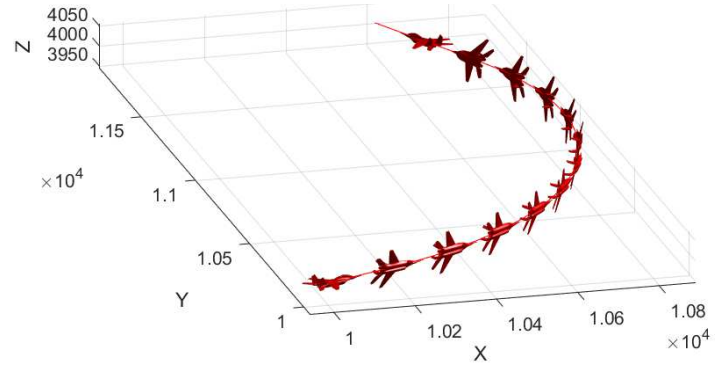
### 2.3.2.5 Immelmann

An Immelmann is used to quickly obtain a reverse heading and bleed off excess speed or gain altitude. The hybrid automaton of the Immelmann and its implementation are represented in Figure 2.5.

In Figure 2.5a, the hybrid automaton of the Immelmann is illustrated with  $Q, E, X, Init, Flow, Inv, G, R$  parameters. The input variables are  $\sigma, \dot{\gamma}_{desired}, \dot{\mu}_{limit}$  and



(a)



(b)

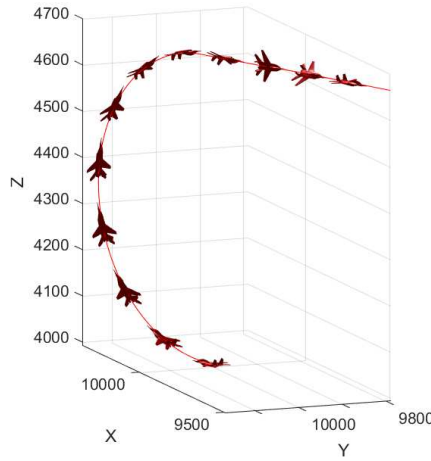
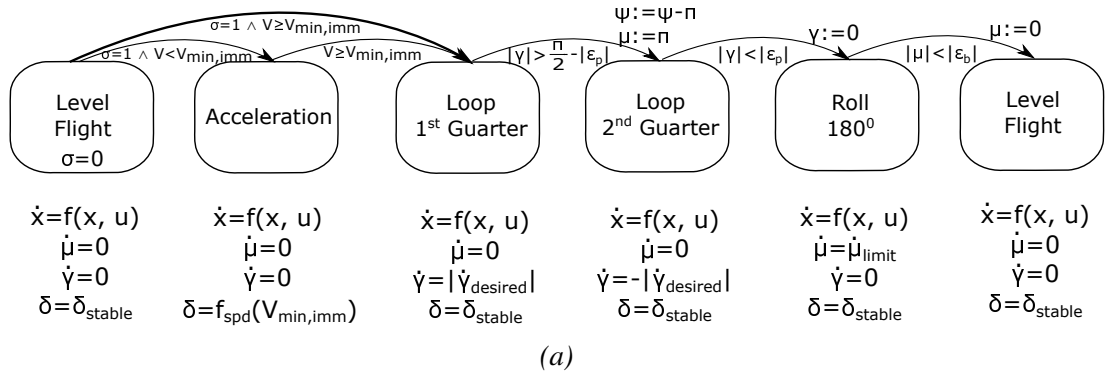
**Figure 2.4** : Break Turn Maneuver: (a) Hybrid Dynamics, (b) Simulation.

$\delta_{stable}$ . The expressions  $\psi := \psi - \pi$  and  $\mu = \pi$  are used as reset relations to obtain reverse heading.

### 2.3.2.6 Split-S

A Split-S is the opposite of an Immelmann. While it quickly makes 180° turn as Immelmann, it sacrifices altitude for speed. It can be used to quickly lose altitude while making a 180° turn. The hybrid automaton of the Split-S and its implementation are presented in Figure 2.6.

In Figure 2.6a, the hybrid automaton of the Split-S is illustrated with  $Q, E, X, Init, Flow, Inv, G, R$  parameters. The input variables are  $\sigma, \dot{\gamma}_{desired}, \dot{\mu}_{limit}$  and  $\delta_{stable}$ . The expressions  $\psi := \psi - \pi$  and  $\mu = 0$  are used as reset relations to obtain reverse heading.

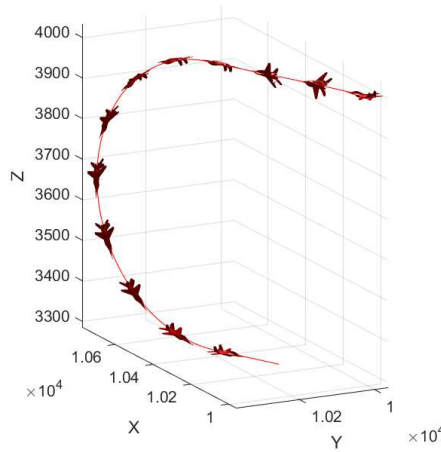
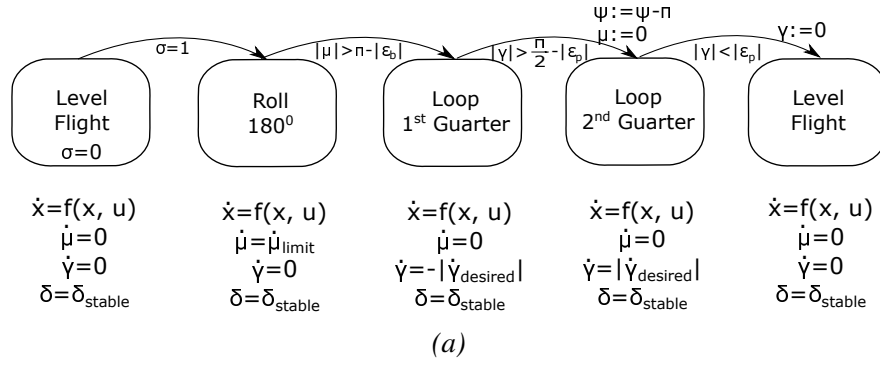


**Figure 2.5** : Immelmann Maneuver: (a) Hybrid Dynamics, (b) Simulation.

### 2.3.2.7 Vertical spiral

Most of the time a vertical spiral is a defensive maneuver. The maneuver consists of rising the nose up during the turn and going in to a spiral. The defender’s goal is to stay out of phase with the attacker. When the defender’s aircraft has superiority in terms of climb and hard turn, then it will create an advantage. The hybrid automaton of the vertical spiral and its implementation are presented in Figure 2.7.

In this hybrid automaton,  $Q = \{LevelFlight, ClimbWithRoll\&Pitch, EndOfClimb, LevelFlight\}$  and the set of differential equations and other information are represented in Figure 2.7a. The input variable  $\sigma$  is used to initialize the maneuver. The other input parameters are  $\dot{\mu}_{limit}$ ,  $\dot{\gamma}_{limit}$ ,  $\mu_{desired}$ ,  $\gamma_{desired}$ ,  $\delta_{stable}$  and  $\Delta h_{desired}$ .  $\Delta h_{desired}$  denotes the desired altitude change during maneuver.



**Figure 2.6** : Split-S Maneuver: (a) Hybrid Dynamics, (b) Simulation.

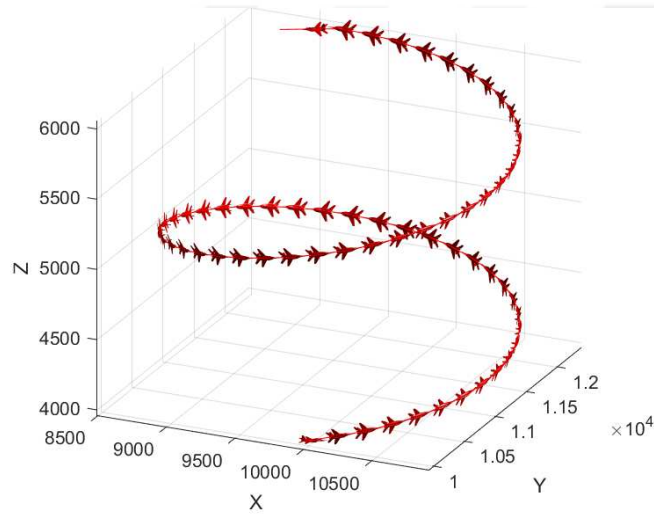
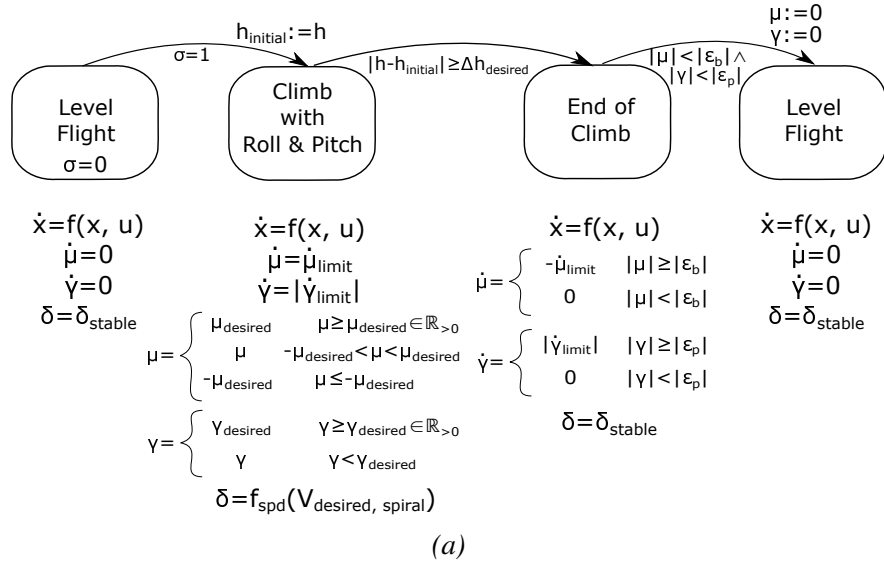
### 2.3.2.8 Spiral dive

Most of the time a spiral dive is a defensive maneuver. It is used by the defender when the kinetic energy become depleted. The maneuver consists of dropping the nose low during the turn and going in to a spiral. The defender's goal is to stay out of phase with the attacker. The hybrid automaton of the spiral dive and its implementation are presented in Figure 2.8.

In this hybrid automaton,  $Q = \{LevelFlight, DescentWithRoll\&Pitch, EndOfDescent, LevelFlight\}$  and the set of differential equations and other information are represented in Figure 2.8a. The input variable  $\sigma$  is used to initialize the maneuver. The other input parameters are  $\dot{\mu}_{limit}$ ,  $\dot{\gamma}_{limit}$ ,  $\mu_{desired}$ ,  $\gamma_{desired}$ ,  $\delta_{stable}$  and  $\Delta h_{desired}$ .  $\Delta h_{desired}$  denotes the desired altitude change during maneuver.

### 2.3.2.9 Low yo-yo

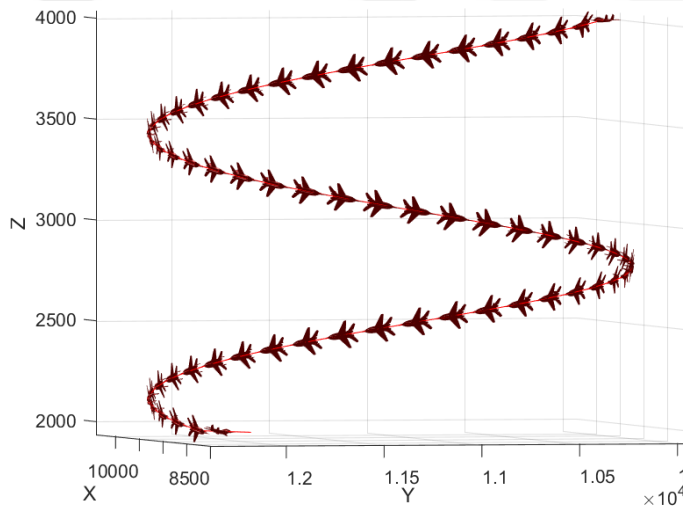
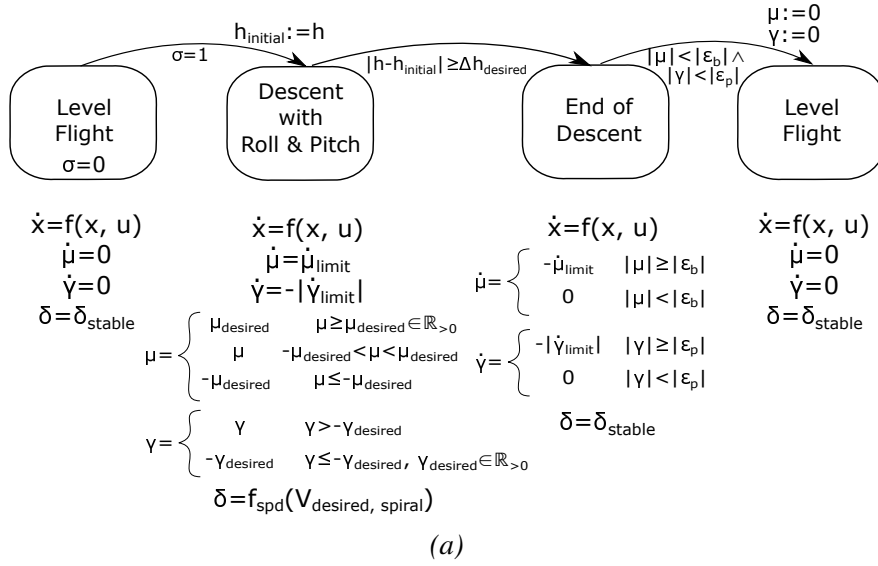
Most of the time a low yo-yo is an offensive maneuver. This maneuver is accomplished by rolling with the nose down into the turn and climbing back to the defender's height.



**Figure 2.7** : Vertical Spiral Maneuver: (a) Hybrid Dynamics, (b) Simulation.

It is an out-of-plane maneuver. It sacrifices altitude for an instantaneous increase in speed. The low yo-yo is often followed by a high yo-yo, to prevent an overshoot, or several small low yo-yo's can be used instead of one large maneuver.

In Figure 2.9a, the hybrid automaton of the low yo-yo is portrayed with  $Q, E, X, Init, Flow, Inv, G, R$  parameters. The input variables are  $\sigma, \dot{\mu}_{limit}, \dot{\gamma}_{limit}, \mu_{desired}, \gamma_{desired}, \delta_{stable}$  and  $\Delta h_{desired}$ .  $\Delta h_{desired}$  corresponds to the desired altitude change during maneuver.  $\epsilon_h$  is the error tolerance for altitude and it is defined as  $5m$ .



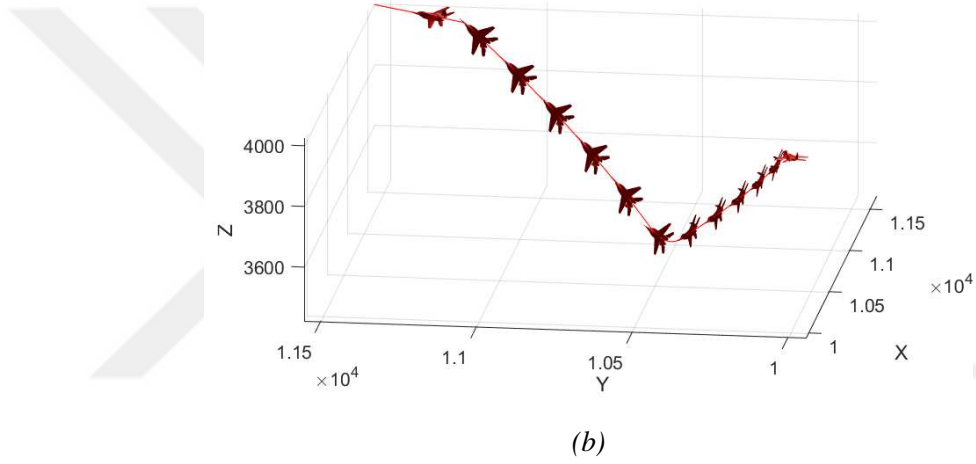
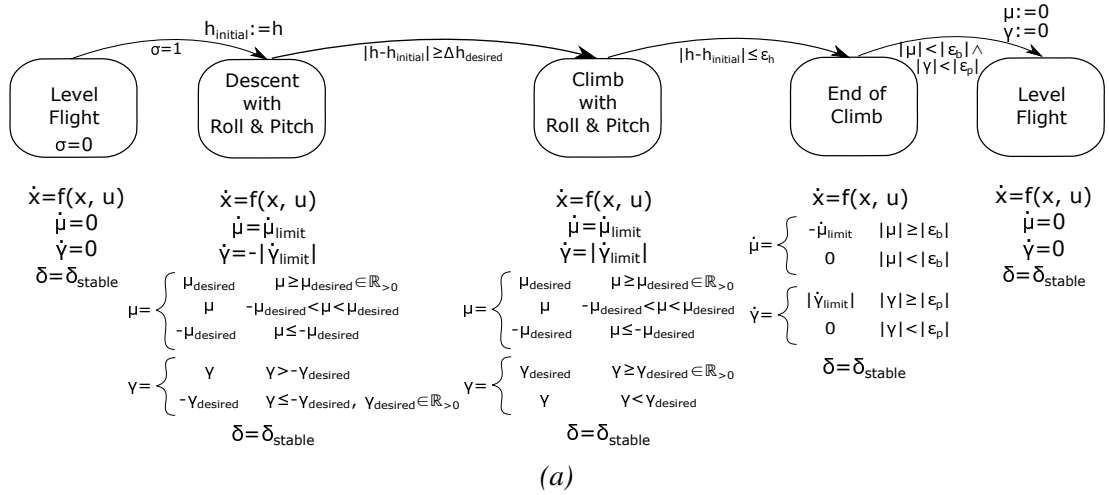
**Figure 2.8** : Spiral Dive Maneuver: (a) Hybrid Dynamics, (b) Simulation.

### 2.3.2.10 High yo-yo

Most of the time a high yo-yo is an offensive maneuver. This maneuver is accomplished by rolling with the nose up into the turn and diving back to the defender's height. It is an out-of-plane maneuver. The maneuver is used to slow the approach of a fast moving attacker while conserving the airspeed energy.

In Figure 2.10a, the hybrid automaton of the high yo-yo is portrayed with  $Q, E, X, Init, Flow, Inv, G, R$  parameters. The input variables are  $\sigma, \dot{\mu}_{limit}, \dot{\gamma}_{limit}, \mu_{desired}, \gamma_{desired}, \delta_{stable}$  and  $\Delta h_{desired}$ .  $\Delta h_{desired}$  corresponds to the desired altitude change during maneuver.  $\epsilon_h$  is the error tolerance for altitude and it is defined as 5m.





**Figure 2.9** : Low Yo-Yo Maneuver: (a) Hybrid Dynamics, (b) Simulation.

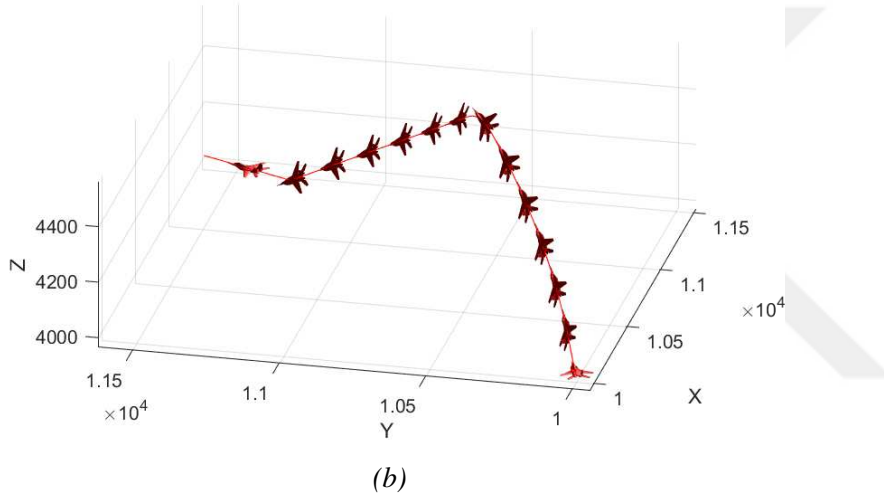
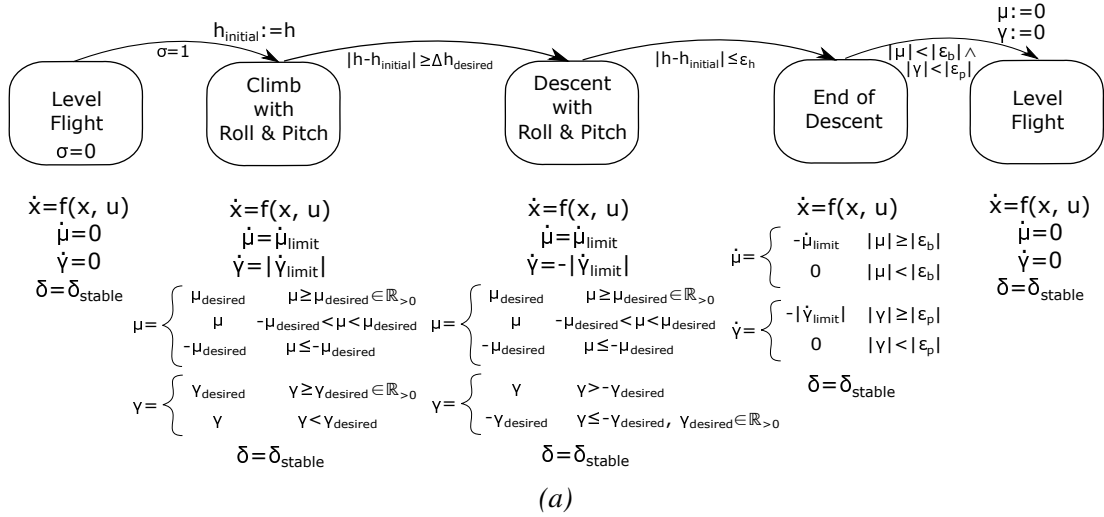
## 2.4 Maneuver-Based Optimal Strategy Generation for Autonomous Air Combat

One of the crucial parts for designing an autonomous air combat environment is to develop a methodology for quantification of combatant's objectives. The other is the evaluation strategy of these quantified parameters to simulate the battle. In this section, scoring function methodology and game theoretical evaluation strategy are explained.

### 2.4.1 Objective function for one-vs-one aerial combat

When discussing one aircraft's position relative to another, range, aspect angle (AA) and bearing angle (BA) are used to describe angular relationships. These three factors dictate which aircraft has air superiority against the other.

As shown in Figure 2.11a, the aspect angle of the blue aircraft ( $AA_b$ ) describes the angle from the tail of the red aircraft to the position of the blue aircraft. The bearing

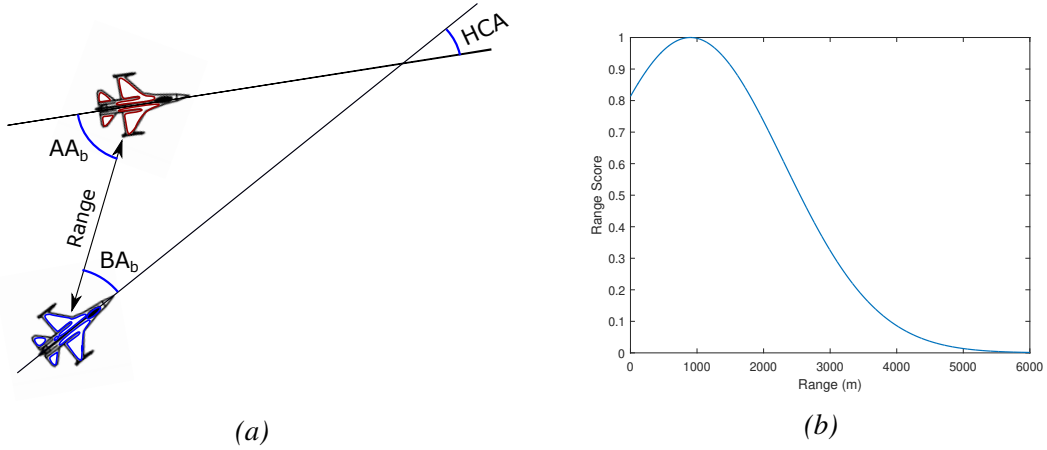


**Figure 2.10** : High Yo-Yo Maneuver: (a) Hybrid Dynamics, (b) Simulation.

angle of the blue aircraft ( $BA_b$ ) denotes the angle from the head of the blue aircraft to the position of the red aircraft. The range is the distance between two aircraft.

For air superiority, each aircraft tries to locate at the rear of the other aircraft with same heading angle. When  $AA_b$  and  $BA_b$  are zero, blue's orientation is perfect due to the alignment with red and also from behind. When  $AA_b$  and  $BA_b$  are  $\pi$ , it is the worst case for blue. The sum of  $AA_b$  and  $BA_b$  must be minimized for air superiority of blue. Then, the orientation score is described by the following relationship:

$$S_o = 1 - \frac{AA_b + BA_b}{\pi} \quad (2.12)$$



**Figure 2.11** : Combat Geometry and Scoring Parameters: (a) Aircraft Relative Geometry, (b) Range Score.

This orientation score must be maximized for blue and be minimized for red. The score takes value between 1 and  $-1$ . Blue is on red's tail when score is 1, and red is on blue's tail when it is  $-1$ .

The other scoring function is the range score. During combat, the fighter tries to shoot down the other one. For shooting, a safety distance is necessary and this distance also must be smaller than the range of missile or gun. To define desired range interval, following range score is used.

$$S_r = ae^{-\frac{(R-R_d)^2}{2\sigma^2}} \quad (2.13)$$

which is a gaussian distribution multiplied by the normalization parameter  $a$ .  $R \in \mathbb{R}_{>0}$  is the range between two aircraft.  $R_d$  is the desired range and  $\sigma$  is the standard deviation. The shooting range for a short range air-to-air missile can be from 900m to 5km.  $R_d = 900$  and  $\sigma = 1400$  are chosen to construct these range limitations. The illustration of the range score with these parameters is given in Figure 2.11b.

The objective function is the combination of these two scoring functions as follows:

$$S_t = S_o S_r \quad (2.14)$$

While blue's objective is to maximization of  $S_t$ , red's objective is to minimization of this parameter. Aerial combat with this objective function is a zero-sum game.

### 2.4.2 Evaluation strategy of zero-sum game

A two person zero-sum game [57], [58] is given by a collection  $(X, Y, A)$ , where

- $X$  is a set of the strategies of Player 1 (P1)
- $Y$  is a set of the strategies of Player 2 (P2)
- $A$  is payoff function defined on  $X \times Y$

In this game, simultaneously,  $P1$  chooses  $x \in X$  and  $P2$  chooses  $y \in Y$ , the choice of the other is unknown for each.  $A(x, y)$  denotes the outcomes of the game that are the gains of  $P1$ , and the losses of  $P2$ .

In a finite two person zero-sum game, if  $X = \{x_1, \dots, x_m\}$  and  $Y = \{y_1, \dots, y_n\}$ , then the game matrix or payoff matrix is given as:

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} \quad \text{where } a_{ij} = A(x_i, y_j)$$

A mixed strategy for  $P1$  can be denoted by an  $m$ -tuple,  $p = (p_1, p_2, \dots, p_m)^T$  of probabilities that add to 1. Similarly, a mixed strategy for  $P2$  is represented as an  $n$ -tuple,  $q = (q_1, q_2, \dots, q_n)^T$ .

Suppose that  $P2$  selects a column at random using  $q \in Y^*$ . If  $P1$  prefers row  $i$ , the  $P1$ 's average payoff is calculated as

$$\sum_{j=1}^n a_{ij}q_j = (Aq)_i \quad (2.15)$$

where,  $(Aq)_i$  corresponds to the  $i^{th}$  component of the vector  $Aq$ . Similarly, if  $P1$  chooses  $p \in X^*$  and  $P2$  prefers column  $j$ , then the average payoff to  $P1$  becomes

$$\sum_{i=1}^m p_i a_{ij} = (p^T A)_j \quad (2.16)$$

where,  $(p^T A)_j$  denotes the  $j^{th}$  component of the vector  $p^T A$ . More generally, if  $P1$  uses  $p \in X^*$  and  $P2$  chooses  $q \in Y^*$ , the average payoff to  $P1$  is

$$\sum_{i=1}^m \left( \sum_{j=1}^n a_{ij}q_j \right) p_i = p^T Aq \quad (2.17)$$

Suppose it is known that Player 2 is going to use a particular strategy  $q \in Y^*$ . Then  $P1$  would prefer that row  $i$  that maximizes (2.15); or, equivalently, he would choose that  $p \in X^*$  that maximizes (2.17). The average payoff to  $P1$  would be

$$\max_{1 \leq i \leq m} \sum_{j=1}^n a_{ij}q_j = \max_{p \in X^*} p^T Aq \quad (2.18)$$

To see that these quantities are equal, note that the left side is the maximum of  $p^T Aq$  over  $p \in X^*$ , and so, since  $X \subseteq X^*$ , must be less than or equal to the right side.

The solution  $p \in X^*$  of the equality (2.18) is called a best response or a Bayes strategy against  $q$ . In particular, when the maximum of (2.15) is achieved by row  $i$ , this strategy  $i$  is called as a pure Bayes strategy against  $q$ .

Similarly, if the preferred strategy  $p \in X^*$  of  $P1$  is known, then  $P2$  would choose that column  $j$  that minimizes (2.16), or, equivalently, that  $q \in Y^*$  that minimizes (2.17). The average payoff to  $P2$  would be

$$\min_{1 \leq j \leq n} \sum_{i=1}^m p_i a_{ij} = \min_{q \in Y^*} p^T Aq \quad (2.19)$$

Any  $q \in Y^*$  that achieves the minimum in (2.19) is called a best response or a Bayes strategy for  $P2$  against  $p$ . The notion of best response presents a practical way of playing a game: Make a guess at the probabilities that you think your opponent will play his/her various pure strategies, and choose a best response against this.

According to **The Minimax Theorem** [57], [58], for every finite two-person zero-sum game,

- there is a number  $V$ , called the value of the game
- there is a mixed strategy for Player 1 such that Player 1's average gain is at least  $V$  no matter what Player 2 does, and
- there is a mixed strategy for Player 2 such that Player 2's average loss is at most  $V$  no matter what Player 1 does, and

The minimum of (2.18) over all  $q \in Y^*$  is denoted by  $\bar{V}$  and called the *upper value of the game*  $(X, Y, A)$ .

$$\bar{V} = \min_{q \in Y^*} \max_{1 \leq i \leq m} \sum_{j=1}^n a_{ij}q_j = \min_{q \in Y^*} \max_{p \in X^*} p^T Aq \quad (2.20)$$

Any  $q \in Y^*$  that achieves the minimum in (2.20) is called a minimax strategy for  $P2$ . It minimizes  $P2$ 's maximum loss.

Similarly, the quality  $\underline{V}$  is called as the *lower value of the game*  $(X, Y, A)$

$$\underline{V} = \max_{p \in X^*} \min_{1 \leq j \leq n} \sum_{i=1}^m p_i a_{ij} = \max_{p \in X^*} \min_{q \in Y^*} p^T A q \quad (2.21)$$

It is the maximum amount that  $P1$  can guarantee himself no matter what  $P2$  does. Any  $p \in X^*$  that achieves the maximum in (2.21) is called a minimax strategy for  $P1$ . Perhaps maximin strategy would be more appropriate terminology in view of (2.21), but from symmetry the same word to describe the same idea may be preferable.

For the solution [57] of minimax strategies, linear programming can be used. Let us consider the game problem from  $P1$ 's point of view. He wants to choose  $p = (p_1, p_2, \dots, p_m)^T$  to maximize (2.19) subject to the constraint  $p \in X^*$ . This becomes the mathematical program: choose  $p = (p_1, p_2, \dots, p_m)^T$

$$\text{maximize} \quad \min_{1 \leq j \leq n} \sum_{i=1}^m p_i a_{ij} \quad (2.22)$$

subject to:

$$p_1 + \dots + p_m = 1 \quad (2.23)$$

$$p_i \geq 0 \text{ for } i = 1, \dots, m \quad (2.24)$$

Because of the min operator on objective function, this is not a linear programming. However, it can be changed through a trick. Variable  $v$  is added to the list of  $P1$ 's variables and it is restricted to be less than objective function,  $v \leq \min \sum_{i=1}^m p_i a_{ij}$ , and it is tried to make  $v$  as large as possible. The problem becomes:

Choose  $v$  and  $p = (p_1, p_2, \dots, p_m)^T$  to

$$\text{maximize} \quad v \quad (2.25)$$

subject to:

$$\sum_{i=1}^m p_i a_{i1} \geq v \quad (2.26)$$

$$\vdots \quad (2.27)$$

$$\sum_{i=1}^m p_i a_{in} \geq v \quad (2.28)$$

$$p_1 + \dots + p_m = 1 \quad (2.29)$$

$$p_i \geq 0 \text{ for } i = 1, \dots, m \quad (2.30)$$

This is a standard linear programming. In a similar way,  $P2$ 's problem can be formulated as:

Choose  $w$  and  $q = (q_1, q_2, \dots, q_n)^T$  to

$$\text{minimize } w \quad (2.31)$$

subject to:

$$\sum_{j=1}^n a_{1j}q_j \leq w \quad (2.32)$$

$$\vdots \quad (2.33)$$

$$\sum_{j=1}^n a_{mj}q_j \leq w \quad (2.34)$$

$$q_1 + \dots + q_n = 1 \quad (2.35)$$

$$q_j \geq 0 \text{ for } j = 1, \dots, n \quad (2.36)$$

The solutions of these two linear programmings give the mixed strategies of  $P1$  and  $P2$  as  $p = (p_1, \dots, p_m)^T$  and  $q = (q_1, \dots, q_n)^T$ , respectively.

In our problem, when two fighters are engaged a  $10 \times 10$  payoff matrix is generated that corresponds to  $A$ . The score function  $S_t$  that was presented previously is used to calculate the payoff matrix. Each aircraft has 10 maneuver choices. These maneuvers are compared one by one, and an average score is calculated for each maneuver pair to generate payoff matrix. After payoff matrix is calculated, linear programs are used to generate the probability vectors  $p = (p_1, \dots, p_m)^T$  for the blue aircraft and  $q = (q_1, \dots, q_n)^T$  for the red aircraft. These two vectors present the probabilities to chose a maneuver for each aircraft with optimal strategy. There are two possibilities in simulation environment, the simulation can be executed in stochastic mode according to probability vector or deterministic mode that uses the maneuver which has maximum probability in probability vector for each fighter. After an aircraft completes the chosen maneuver, then the process is repeated by considering the trajectory of other aircraft that stays from previous choice. The simulation is finished when an aircraft shoots down the other or predefined simulation time ends.

### 2.4.3 Air superiority for n-vs-m aerial combat

The relative position, angular relationship, speed and altitude are main factors that affect the air superiority. The objectives of the combatants can be quantified as a function of these variables.

#### 2.4.3.1 Offensive orientation score

In offensive manner, each aircraft tries to locate at the rear of the other aircraft with same heading angle for air superiority. When  $AA_b$  and  $BA_b$  are zero, blue's orientation is perfect due to the alignment with red and also from behind. When  $AA_b$  and  $BA_b$  are  $\pi$ , it is the worst case for blue. The sum of  $AA_b$  and  $BA_b$  must be minimized for air superiority of the blue aircraft. Then, the offensive orientation score for blue is described by the following relationship:

$$S_{o_1} = 1 - \frac{AA_b + BA_b}{\pi} \quad (2.37)$$

This orientation score must be maximized for the blue aircraft. The score takes value between 1 and  $-1$ . Blue is on red's tail when score is 1, and red is on blue's tail when it is  $-1$

#### 2.4.3.2 Defensive orientation score

In defensive manner, an aircraft tries to escape from dangerous orientations that correspond to both of aspect and bearing angles of opponent are smaller than  $30^\circ$ . The defensive orientation score for blue is described as follows:

$$S_{o_2} = \begin{cases} 1 & \text{If } AA_r > \frac{\pi}{6} \text{ or } BA_r > \frac{\pi}{6} \\ 0 & \text{otherwise} \end{cases} \quad (2.38)$$

#### 2.4.3.3 Range score

The other scoring function is the range score. During combat, a fighter tries to shoot down its enemy. For shooting, a safety distance is necessary and this distance also must be smaller than the range of missile or gun. To define desired range interval, following range score is used

$$S_{rg} = a_1 e^{\frac{-(R-R_d)^2}{2\sigma^2}} \quad (2.39)$$

which is a gaussian distribution multiplied by the normalization parameter  $a_1$ .  $R \in \mathbb{R}_{>0}$  is the range between two aircraft.  $R_d$  is the desired range and  $\sigma$  is the standard



deviation. The shooting range for a short range air-to-air missile can be from 900m to 5km.  $R_d = 900$  and  $\sigma = 1400$  are chosen to construct these range limitations.

#### 2.4.3.4 Speed score

The total energy level of an aircraft is crucial in an air combat. So, the speed of the aircraft must be kept at desired level to prevent the high amount of speed decrease. The scoring function for speed is given as follows:

$$S_v = \begin{cases} a_2 & \text{If } \left| \frac{V_{cur} - V_{ref}}{V_{ref}} \right| > 0.5 \\ a_2 \left| \frac{V_{cur} - V_{ref}}{V_{ref}} \right| & \text{otherwise} \end{cases} \quad (2.40)$$

where,  $a_2 \in \mathbb{R}_{<0}$  is the scale parameter and is chosen as  $-0.8$  in this study.  $V_{cur}$  and  $V_{ref}$  are the current speed and reference/desired speed, respectively.

#### 2.4.3.5 Altitude score

The other factor is the altitude. Following scoring function is used to prevent the undesirable altitude decrease

$$S_h = \begin{cases} a_3 & \text{If } |H_{cur} - H_{ref}| \leq 2000 \\ 0 & \text{otherwise} \end{cases} \quad (2.41)$$

where,  $a_3 \in \mathbb{R}_{>0}$  is the scale parameter and is chosen as  $0.5$  in this study.  $H_{cur}$  and  $H_{ref}$  are the current altitude and reference/desired altitude, respectively.

#### 2.4.3.6 Total scores for n-vs-m air combat

The objective function is the combination of these scoring functions. In  $n - vs - m$  combat, the blue team has  $n$  combatants, while the red team has  $m$ . However, team members have cooperation. So, only one objective function is developed for each team. Let us focus on  $2vs1$  aerial combat. The general combined objective function is at the form of (2.42).

$$S_t = S_o S_{rg} + S_v + S_h \quad (2.42)$$

The objective is to maximization of  $S_t$ .  $S_{o_1}$  is used as  $S_o$  in offensive mode, while  $S_{o_2}$  is used as  $S_o$  in defensive mode. When both of the fighters are offensive against enemy, the total scoring function for the blue team in a  $2vs1$  combat is given by the following equation.

$$S_{b,t} = S_{o_1, b_1 r_1} S_{rg, b_1 r_1} + S_{o_1, b_2 r_1} S_{rg, b_2 r_1} + S_v + S_h \quad (2.43)$$

where,  $b_1, b_2$  and  $r_1$  correspond to 1<sup>st</sup> fighter in the blue team, 2<sup>nd</sup> fighter in the blue team and the combatant in the red team, respectively.  $S_V = S_{v,b_1} + S_{v,b_2} - S_{v,r_1}$  is the combined speed score of  $b_1, b_2$  and  $r_1$ .  $S_H = S_{h,b_1} + S_{h,b_2} - S_{h,r_1}$  is the combined altitude score.

The objective function is defined according to strategy against other team. Let us assume that  $b_1$  is offensive, while  $b_2$  is defensive against  $r_1$ , then the objective function of blue is given as follows:

$$S_{b,t} = S_{o_1,b_1r_1}S_{rg,b_1r_1} + S_{o_2,b_2r_1}S_{rg,b_2r_1} + S_V + S_H \quad (2.44)$$

In this scoring function, defensive orientation score ( $S_{o_2}$ ) is used instead of offensive orientation ( $S_{o_1}$ ) for  $b_2$ .

The objective function for the red team is also determined as same manner. When  $r_1$  is defensive against  $b_1$  and  $r_1$  is offensive against  $b_2$ , the total scoring function is given as follows:

$$S_{r,t} = S_{o_2,r_1b_1}S_{rg,r_1b_1} + S_{o_1,r_1b_2}S_{rg,r_1b_2} - S_V - S_H \quad (2.45)$$

In this concept, it can be said that aggressiveness of the fighters determines the total score functions. Aggressiveness of a combatant against to an enemy can be low (defensive) or high (offensive). It can be presented as boolean. Defensive manner corresponds to 0, while offensive manner corresponds to 1. So, combination of aggressiveness in a 2vs1 combat can be presented with a 4-digit binary number. E.g. lets  $b_1$  and  $b_2$  are offensive against to  $r_1$ ,  $r_1$  is defensive against to  $b_1$  and  $r_1$  is offensive against to  $b_2$ , then combination of aggressiveness is presented as 1101. This presentation will be used at the rest of the study.

#### 2.4.4 Security strategy to evaluate the air combat between multiple aircraft

A two person nonzero-sum game [57], [58] is given by a collection  $(X, Y, A, B)$ , where

- $X$  is a set of the strategies of Player 1 (P1)
- $Y$  is a set of the strategies of Player 2 (P2)
- $A$  is payoff function of P1 defined on  $X \times Y$

- $B$  is payoff function of  $P2$  defined on  $X \times Y$

In this game, simultaneously,  $P1$  chooses  $x \in X$  and  $P2$  chooses  $y \in Y$ , the choice of the other is unknown for each. In this finite two person nonzero-sum game or bimatrix game, if  $X = \{x_1, \dots, x_m\}$  and  $Y = \{y_1, \dots, y_n\}$ , then the game matrices are given as  $A, B$  and  $a_{ij} = A(x_i, y_j)$ ,  $b_{ij} = B(x_i, y_j)$  are the gains, when  $P1$  chooses row  $i$  and  $P2$  chooses column  $j$ , respectively.

As in zero-sum games, security strategies can be introduced for the players in bimatrix games. The security strategy for a player is presented such that the player's average gain is at least  $V$  no matter what the other player does. The strategy creates a lower bound for the gain of the player that is independent from the choice of the other player. It should be emphasized that, in a bimatrix game  $(A, B)$ , the security strategy of  $P1$  involves only the entries of matrix  $A$ , while the entries of matrix  $B$  are used to generate the security strategy of  $P2$ .

In a 2vs1 air combat, when the fighters are engaged, two different  $100 \times 10$  payoff matrices are generated, which correspond to  $A$  and  $B$  according to the score functions  $S_{b,t}$  and  $S_{r,t}$ , respectively. The score functions are determined according to predefined 4-digit binary number, which presents the combination of aggressiveness for the game. Each aircraft has 10 maneuver choices. Hence, the blue team has 100 maneuver combinations, while the red team has 10 maneuver choices. These maneuvers are compared one by one, and an average score is calculated for each maneuver combination to generate payoff matrices. After payoff matrices are calculated, linear programs are used to generate the probability vectors  $p = (p_1, \dots, p_m)^T$  for the blue team and  $q = (q_1, \dots, q_n)^T$  for the red team. These two vectors present the probabilities to chose a maneuver combination for each team with optimal strategy. There are two possibilities in simulation environment, the simulation can be executed in stochastic mode according to probability vector or deterministic mode that uses the maneuver which has maximum probability in probability vector for each team. After a team completes the chosen maneuvers, the process is repeated by considering the trajectory of other aircraft that stays from previous choice. The simulation is finished when a team shoots down the other or predefined simulation time ends.

**Table 2.1** : Input Parameters for Simulations.

	$\mu_{desired}$	$\gamma_{desired}$	$\dot{\mu}_{desired}$	$\dot{\gamma}_{desired}$	$\dot{\mu}_{limit}$	$\dot{\gamma}_{limit}$	$\Delta h_{desired}$
Aileron Roll	-	-	$40^\circ/s$	-	-	-	-
Barrel Roll	-	$30^\circ$	$40^\circ/s$	-	-	$90^\circ/s$	-
Loop	-	-	-	$40^\circ/s$	-	-	-
Break Turn	$80^\circ$	-	-	-	$90^\circ/s$	-	-
Immelmann	-	-	-	$40^\circ/s$	$90^\circ/s$	-	-
Split S	-	-	-	$40^\circ/s$	$90^\circ/s$	-	-
Vertical Spiral	$80^\circ$	$30^\circ$	-	-	$90^\circ/s$	$90^\circ/s$	500 m
Spiral Dive	$80^\circ$	$30^\circ$	-	-	$90^\circ/s$	$90^\circ/s$	500 m
Low Yo-Yo	$65^\circ$	$30^\circ$	-	-	$90^\circ/s$	$90^\circ/s$	500 m
High Yo-Yo	$65^\circ$	$30^\circ$	-	-	$90^\circ/s$	$90^\circ/s$	500 m

**Table 2.2** : Initial Conditions for Scenarios.

		$x_1$	$x_2$	$h$	$V$	$\psi$
Scenario 1	Blue	10000	10000	6000	250	$\pi/12$
	Red	13000	13000	6000	250	$13\pi/12$
Scenario 2	Blue	10000	10000	6000	250	$\pi/8$
	Red	14000	14000	6000	250	$\pi/8$

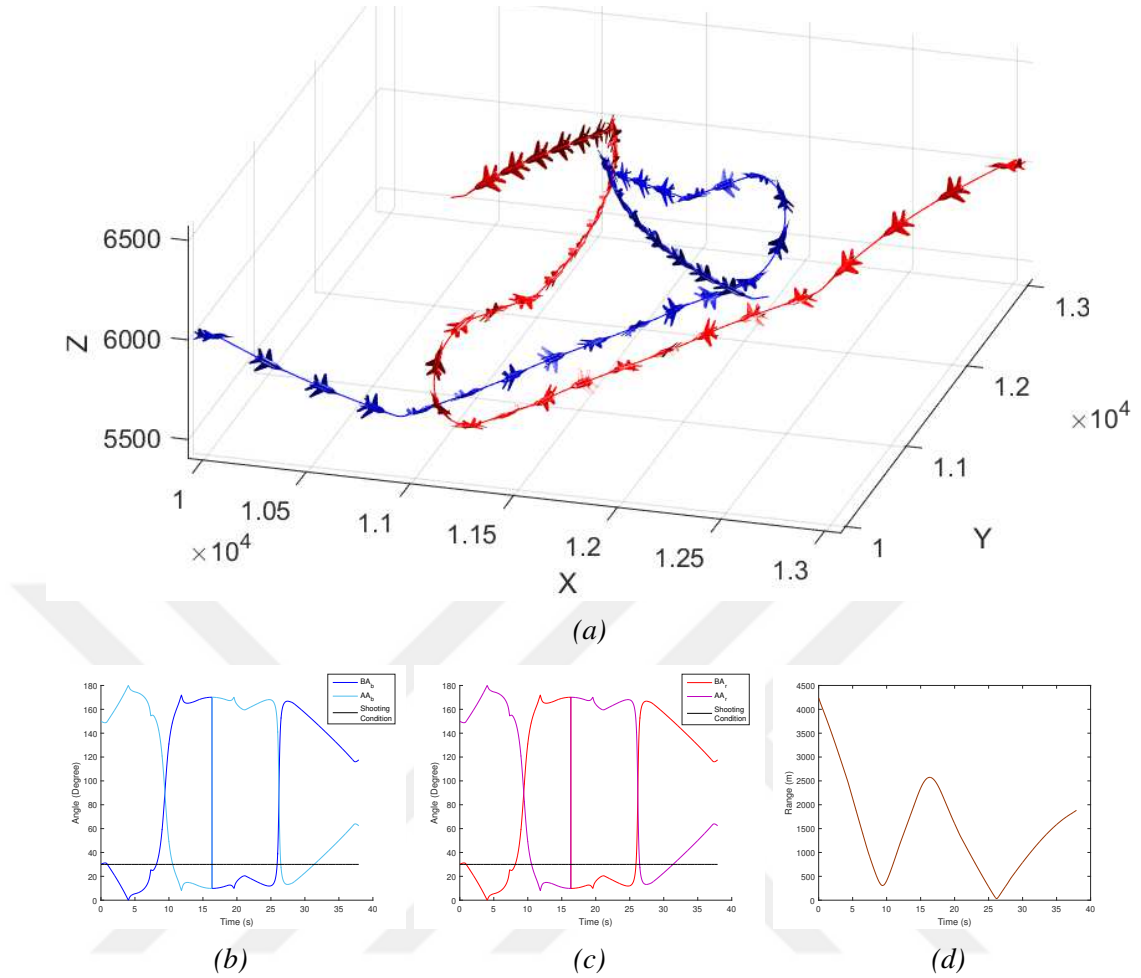
## 2.5 Simulation Results

### 2.5.1 Maneuver-based one-vs-one aerial combat

To simulate a maneuver in the air combat maneuver library, some input parameters are required. For example,  $\dot{\mu}_{desired}$  is necessary as an input for an Aileron Roll and it determines the movement space of the maneuver. An Immelmann can cause high altitude increment because of small path angle rate. In the implementation phase, input parameters for maneuvers are standardized as given in Table 2.1.

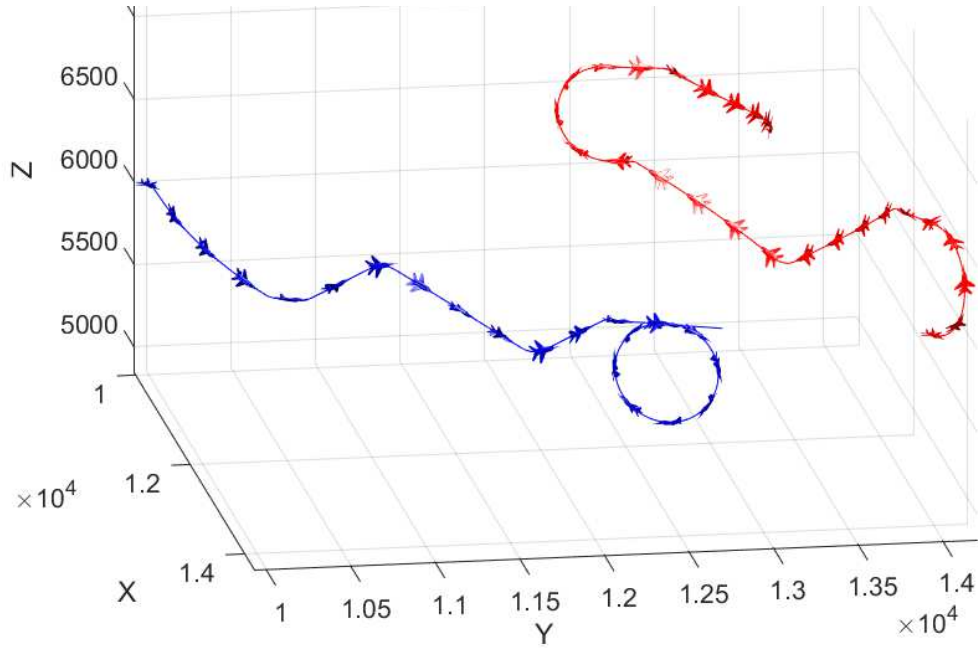
Two different scenarios are analyzed in deterministic mode as implementations. These scenarios are about one-on-one air combats at initial symmetric conditions and initial asymmetric conditions. The initial conditions for fighters are presented in Table 2.2.

In the first scenario, the heading angle of the blue aircraft is  $\pi/12$ , while the red's heading is  $13\pi/12$ . They are in head-to-head positions at the beginning of the air combat. The velocities and altitudes are also same. Hence, they are totally equivalent in this scenario and they are in symmetric positions according to each other as the initial condition. The simulation results are represented in Figure 2.12. Both of the

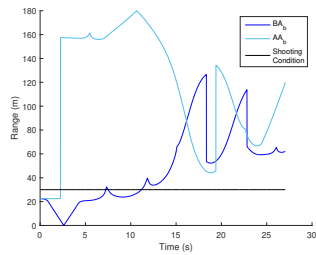


**Figure 2.12** : Air Combat with Symmetric Initial Conditions: (a) 3D Positions, (b) Combat Angles for Blue, (c) Combat Angles for Red, (d) Range.

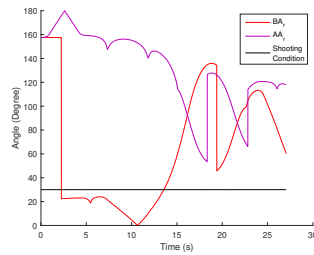
aircraft choose same sequence of maneuvers as Spiral Dive, Aileron Roll, Immelmann, High Yo-Yo, which are illustrated in Figure 2.12a. During the combat, the fighters' movements are totally symmetric. Because of this pattern, the bearing and aspect angles of two fighters are entirely same during the air combat as shown in Figure 2.12b and Figure 2.12c. The range between the aircraft is also presented in Figure 2.12d. As expected, it fluctuates around  $1500m$ . In the objective function, the ranges between  $500m$  and  $2000m$  have high scores, so it is around these values as desired. The symmetric behavior of the fighters makes sense. When any of the aircraft has no advantage and the initial conditions are symmetric, the maneuvers should be symmetric. This pattern is proof of the validity of the evaluation strategy. The players choose optimal strategies to prevent the other's air superiority and get the other down. The second scenario is about asymmetric initial conditions, both of the fighters have same heading angle as  $\pi/8$ . This means that the blue aircraft has an advantage at



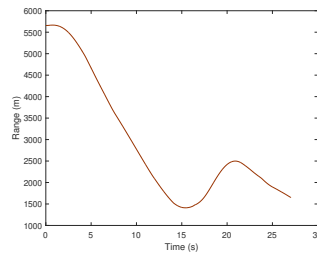
(a)



(b)



(c)



(d)

**Figure 2.13** : Air Combat with Asymmetric Initial Conditions: (a) 3D Positions, (b) Combat Angles for Blue, (c) Combat Angles for Red, (d) Range.

the beginning of the battle because of the orientation, while the red aircraft has safe separation from the blue aircraft. We assume that minimum  $3000m$  is necessary for a shooting and both of the aspect angle and bearing angle of the attacker must be less than  $30^\circ$  for a shooting. The simulation results are illustrated in Figure 2.13. The sequence of maneuvers is Spiral Dive, Barrel Roll, Split S, Immelmann for the blue aircraft and Immelmann, Low Yo-Yo, Immelmann, Spiral Dive for the red aircraft. As seen in Figure 2.13d, the range is greater than  $3000m$  during first  $9s$ . So, none of them has a chance for a shooting during this period. The blue fighter has orientation advantage at the beginning of the combat around  $2s$  as seen in Figure 2.13b. Both of  $BA_b$  and  $AA_b$  are less than  $30^\circ$  during advantageous period, however, blue is not capable of shooting because of the range. Then, red turns with a Split S and blue loses the orientation

**Table 2.3** : Initial Conditions for All Scenarios.

	$x_1$	$x_2$	$h$	$V$	$\psi$
Blue 1	10000	10000	6000	250	$\pi/3$
Blue 2	14000	10000	6000	250	$2\pi/3$
Red 1	12000	10000	6000	250	$\pi/2$

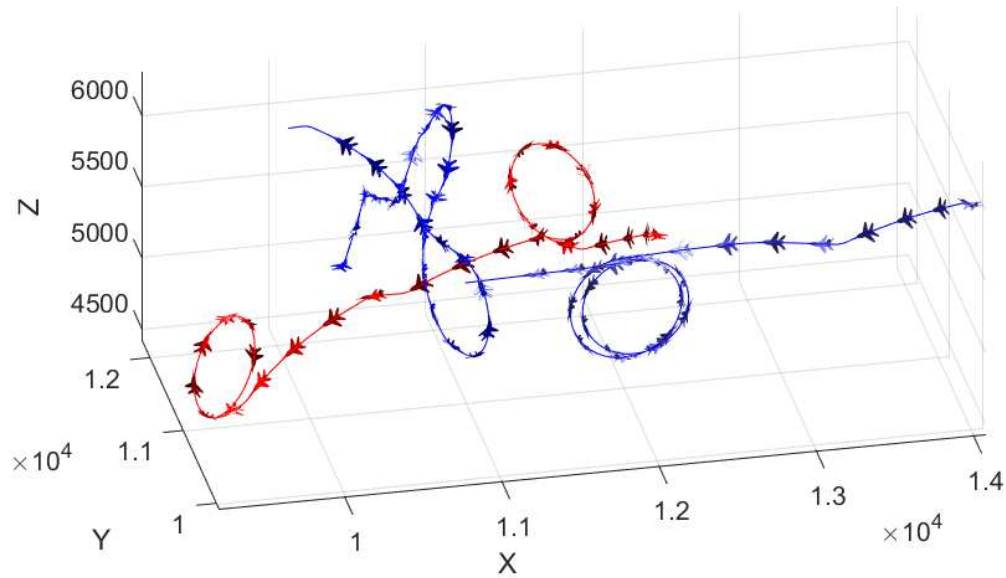
advantage. After first maneuvers, none of them has required orientations for a shooting as shown in Figure 2.13b and Figure 2.13c.

### 2.5.2 Maneuver-based two-vs-one aerial combat

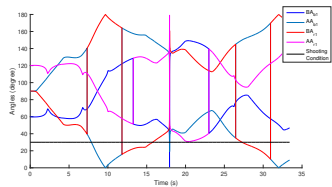
Three different scenarios are analyzed in deterministic mode as implementations. These scenarios are about two-on-one air combats at different combinations of aggressiveness. The initial conditions for the fighters are presented in Table 2.3. The combinations of aggressiveness for the scenarios are 1111, 1101 and 1001, respectively.

In all of the scenarios, the initial conditions except the aggressiveness are same. In the first scenario, all of the fighters are offensive against the other enemies. We assume that minimum  $3000m$  is necessary for a shooting and both of the aspect angle and bearing angle of the attacker must be less than  $30^\circ$  for a shooting. The simulation results are presented in Figure 2.14. Blue 1 chooses the sequence of maneuvers as Barrel Roll, Split S, Spiral Dive, Immelmann, Vertical Spiral, while Blue 2 chooses the sequence as Spiral Dive, Break Turn, Split S, Loop, Immelmann, Break Turn. The sequence of maneuvers is Spiral Dive, Loop, Spiral Dive, Spiral Dive, Loop and Vertical Spiral for the red aircraft. The bearing and aspect angles of the teams are illustrated in Figure 2.14b and Figure 2.14c, and the ranges between aircraft are also presented in Figure 2.14d. During the aerial combat, there is no air superiority between Blue 1 and Red, while Blue 2 has  $1.39s$  air superiority, which begins at  $6s$  of the simulation. During this superiority period, the ranges between Blue 2 and Red aircraft are around  $1500m$  that is sufficient for a shooting.

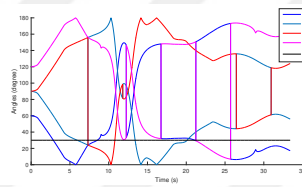
In the second scenario, Red is defensive against Blue 1, while it is offensive against Blue 2. And, both members of the blue team are offensive. The simulation results are illustrated in Figure 2.15. The sequence of maneuvers is Barrel Roll, Immelmann,



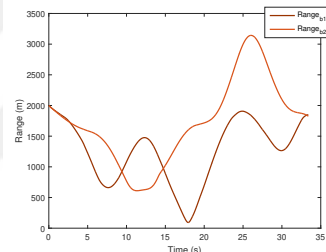
(a)



(b)



(c)



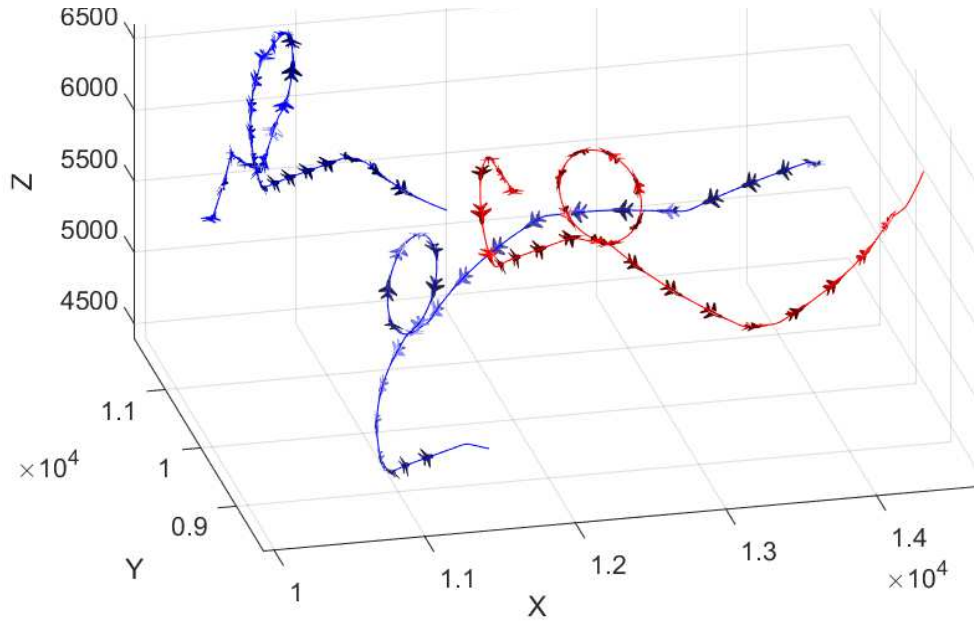
(d)

**Figure 2.14** : Air Combat for 1111 Configuration: (a) 3D Positions, (b) Combat Angles between  $b_1$  and  $r_1$ , (c) Combat Angles between  $b_2$  and  $r_1$ , (d) Ranges.

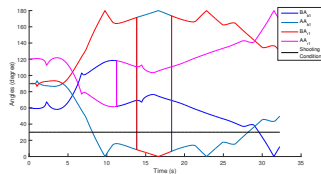
Low Yo-Yo, Spiral Dive for Blue 1 and it is Spiral Dive, Break Turn, Spiral Dive, Loop, Spiral Dive, Vertical Spiral for Blue 2. The maneuver set is Split S, Vertical Spiral, Loop, Spiral Dive, Vertical Spiral, Vertical Spiral for Red. During the combat, none of the combatants has a chance for a shooting because of insufficient orientations as seen in Figure 2.15b and Figure 2.15c. So, this aggressiveness strategy is better than 1111 from perspective of the red team, whereas 1111 is better for the blue team.

In the third scenario, Blue 1 and Red are offensive against Red and Blue 2, whereas Blue 2 and Red are defensive against Red and Blue 1, respectively. The simulation results are illustrated in Figure 2.16. Blue 1 chooses the sequence of maneuvers as Barrel Roll, Split S, Spiral Dive, Spiral Dive, whereas Blue 2 chooses the sequence as Vertical Spiral, Break Turn, Low Yo-Yo and Spiral Dive. The maneuver set is Split S, Loop, Spiral Dive and Aileron Roll for Red. As seen in Figure 2.16b, Blue 1 has air superiority against Red during 3s, then the game is terminated. While 1101 is better

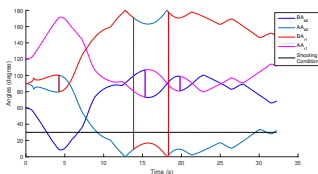




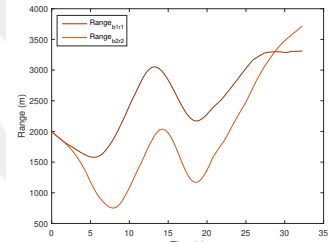
(a)



(b)



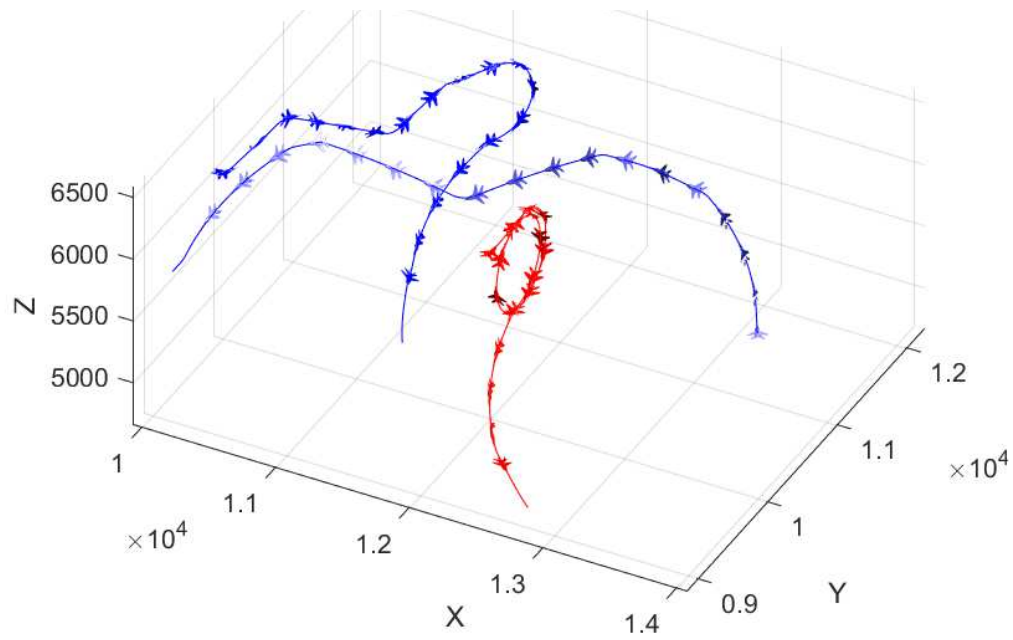
(c)



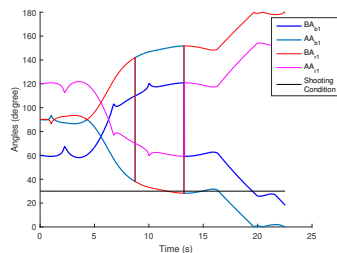
(d)

**Figure 2.15** : Air Combat for 1101 Configuration: (a) 3D Positions, (b) Combat Angles between  $b_1$  and  $r_1$ , (c) Combat Angles between  $b_2$  and  $r_1$ , (d) Ranges.

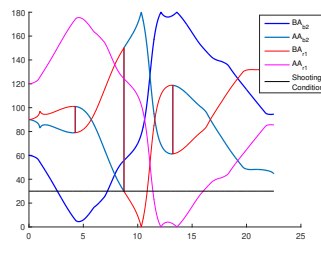
for the red team, 1001 is best for the blue team. Changing the aggressiveness of Blue 2 from 1101 to 1001, the blue team can get the advantage. This strategy is totally related to the blue team. Changing only one of the combatant's mood, the air combat can be won. In the 1111 combination, the blue team has air superiority for 1.39s time period. Changing the mood of Red from 1111 to 1101, the air combat can be ended in a draw, which is more desirable for the red team.



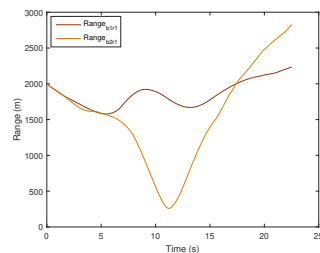
(a)



(b)



(c)



(d)

**Figure 2.16** : Air Combat for 1001 Configuration: (a) 3D Positions, (b) Combat Angles between  $b_1$  and  $r_1$ , (c) Combat Angles between  $b_2$  and  $r_1$ , (d) Ranges.

### **3. OPTIMIZATION-BASED AUTONOMOUS AIR TRAFFIC CONTROL**

#### **3.1 Purpose**

In order to handle increasing demand in air transportation, high-level automation support seems inevitable. This chapter presents an optimization-based autonomous air traffic control (ATC) system and the determination of airspace capacity with respect to the proposed system. We use the similar maneuver-based approach presented in the previous chapter. We model aircraft dynamics and guidance procedures for simulation of aircraft motion and trajectory prediction. The predicted trajectories are used during decision process and simulation of aircraft motion is the key factor to create a traffic environment for estimation of airspace capacity. We define the interventions of an air traffic controller (ATCo) as a set of maneuvers that is appropriate for real air traffic operations. The decision process of the designed ATC system is based on integer linear programming (ILP) constructed via a mapping process that contains discretization of the airspace with predicted trajectories to improve the time performance of conflict detection and resolution. We also present a procedure to estimate the airspace capacity with the proposed ATC system. This procedure consists of constructing a stochastic traffic simulation environment that includes the structure of the evaluated airspace. The approach is validated on real air traffic data for en-route airspace, and it is also shown that the designed ATC system can manage traffic much denser than current traffic.

#### **3.2 Aircraft Model and Maneuvers for Civil Aviation**

An aircraft follows its planned trajectory according to flight plan and the interventions of air traffic controller. To simulate and predict this process, an aircraft model and trajectory tracking algorithms are required. This section presents the aircraft model and guidance algorithms that will be used both simulation and trajectory prediction processes.

### 3.2.1 Aircraft dynamics

General form of the time-invariant dynamics of the aircraft is considered as:

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

where the  $x(t) \in X \subseteq \mathbb{R}^n$ ,  $u(t) \in U \subseteq \mathbb{R}^m$  such that  $n, m \in \mathbb{N}$  and the state  $x_0 \in X$  is called the initial state of the aircraft. A point mass model is used for modeling the aircraft dynamics. The model is a nonlinear dynamical system with three control inputs and six state variables. The state variables of the aircraft are the horizontal position ( $x$  and  $y$ ), altitude ( $h$ ), the true airspeed ( $v$ ), the heading angle ( $\psi$ ) and the mass of the aircraft ( $m$ ). The control inputs of the aircraft are the engine thrust ( $T$ ), the bank angle ( $\phi$ ) and the flight path angle ( $\gamma$ ). The wind acts as a disturbance on the aircraft dynamics, which is modeled by the wind speed,  $W = (w_1, w_2, w_3)$ . The equations of aircraft motion are:

$$\dot{x} = v \cos(\psi) \cos(\gamma) + w_1 \quad (3.2)$$

$$\dot{y} = v \sin(\psi) \cos(\gamma) + w_2 \quad (3.3)$$

$$\dot{h} = v \sin(\gamma) + w_3 \quad (3.4)$$

$$\dot{v} = -\frac{C_D S \rho v^2}{2m} - g \sin(\gamma) + \frac{T}{m} \quad (3.5)$$

$$\dot{\psi} = \frac{C_L S \rho v \sin(\phi)}{2m \cos(\gamma)} \quad (3.6)$$

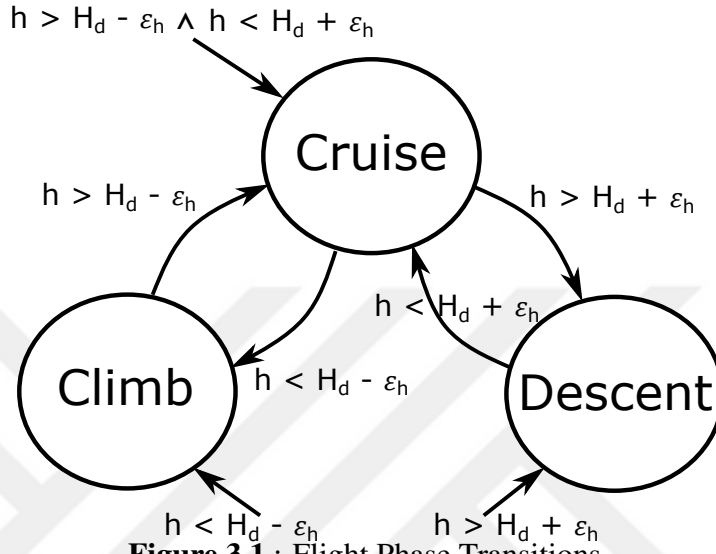
$$\dot{m} = -F \quad (3.7)$$

In the equation set above, aerodynamic lift and drag coefficients are denoted by  $C_L$  and  $C_D$ , total wing surface area is  $S$ , air density is indicated as  $\rho$  and the fuel consumption is indicated as  $F$ . These coefficients and other parameters such as bounds on the speed and mass are obtained from the Base of Aircraft Data (BADA) as in paper [59].

In BADA, the thrust and fuel consumption are presented as functions of altitude and speed according to flight phase and engine type. The fuel consumption is also affected from the thrust. For a specific type of aircraft, these functions differ with respect to the flight phase. So, it is necessary to describe the flight phase of aircraft as a discrete state that helps to use the appropriate functions. In this case, the system involves the interaction of continuous and discrete dynamics, which is the fundamental property of a hybrid system [53].

The hybrid automaton for aircraft dynamics with flight phases is presented in Figure 3.1. The set of discrete states  $Q$  is  $\{Climb, Cruise, Descent\}$  and the set of guard

conditions  $G$  is described in terms of the altitude of aircraft  $h$ , the desired altitude  $H_d$  and the buffer for altitude transitions  $\varepsilon_h$ . The same set of equations (3.2)-(3.7) is used for vector field  $f$  in each flight phase, however the functions that calculate the thrust and fuel consumption in  $f$  are different from each other. The detailed information about these functions can be found in [60], [61].



**Figure 3.1** : Flight Phase Transitions.

### 3.2.2 Reference trajectory and speed schedule

The reference trajectory corresponds to a set of way-points  $WP = \{wp_0, wp_1, \dots, wp_k\}$  and  $wp_i \in \mathbb{R}^3$ , which contains the position information. During the flight, an aircraft follows its reference trajectory and this trajectory can be modified by Air Traffic Control Operator (ATCo). The speed schedule defines the desired speed of the aircraft. In BADA [60], the airline procedure model is defined according to engine model that gives the speed schedule as a function of altitude and flight phase. In this study, desired speed is defined as the combination of the BADA airline procedure model and the speed intervention of ATCo as follows:

$$V_d = f_v(h, q) + V_c \quad (3.8)$$

where,  $V_d$  and  $V_c$  are the desired speed and the speed change, respectively.  $f_v(h, q)$  corresponds to the speed schedule of the airline according to altitude and flight phase. In the rest of the study, the flight plan term implies both reference trajectory and speed schedule.  $H_d$  corresponds to the desired altitude. When an aircraft flies between  $wp_i$

and  $wp_{i+1}$ ,  $H_d$  defines the sum of the altitude information in  $wp_{i+1}$  and the altitude change  $H_c$  given by ATCo.

### 3.2.3 Trajectory tracking guidance

Trajectory Tracking corresponds to generate the required control inputs (e.g. bank angle and flight path angle) for an aircraft to follow the reference trajectory. In trajectory tracking, two control functions are combined for lateral and longitudinal motions. In the longitudinal part, the desired velocity and target altitude are reached by calculating the required flight path angle. The lateral control part includes the straight line controller, turn controller and heading controller. These controllers calculate required bank angle to follow a horizontal path. In addition to them, a speed controller adjust thrust of the aircraft to keep the speed at the desired level during cruise.

#### 3.2.3.1 Longitudinal controller

The longitudinal controller determines the flight path angle. There are two different controllers and one of them is used as longitudinal controller according to the flight phase. In climb and descent phases, the controller calculates the flight path angle according to desired speed with the following equation:

$$\gamma = \max \left\{ \gamma_{min}, \min \left\{ \gamma_{max}, \left( \frac{T - D}{m} - \frac{V_d - v}{\Delta t} \right) \frac{1}{g} \right\} \right\} \quad (3.9)$$

where,  $D$  is the drag.  $V_d$  is a function of  $h$ , so this controller calculates the necessary flight path angle to reach the desired altitude and keep the desired speed.

In cruise phase, there is a proportional controller to compensate the steady state error of the altitude as follows:

$$\gamma = \max \left\{ \gamma_{min}, \min \left\{ \gamma_{max}, k_a(H_d - h) \right\} \right\} \quad (3.10)$$

#### 3.2.3.2 Lateral controller

This controller calculates the bank angle to follow the path on the horizontal plane. Two different controllers are used with a switching mechanism as lateral controller to generate the bank angle, which are straight-line controller and turn controller. In addition to them, there is a heading controller that is used in a special case.

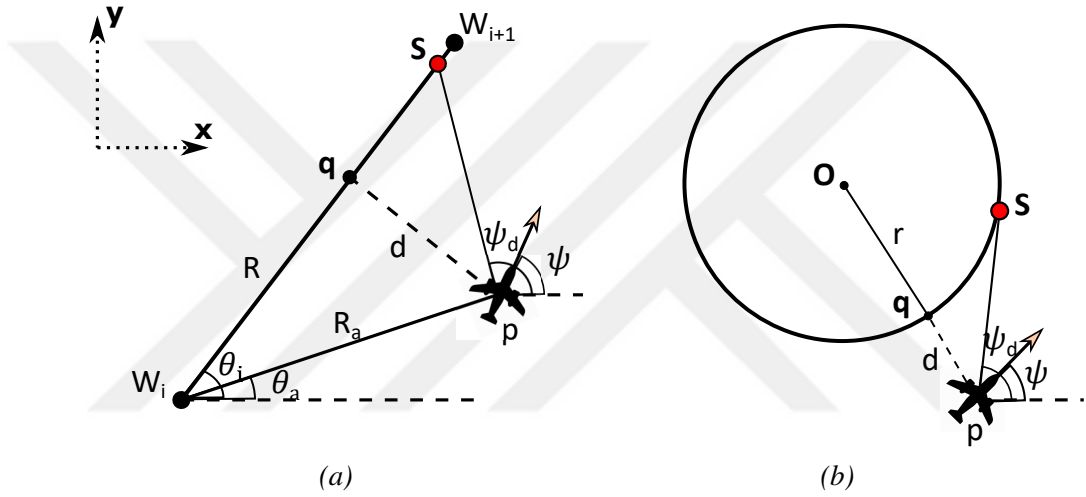
**Straight-line controller:** This controller generates required bank angle,  $\phi$ , to follow the straight line between two way-points in horizontal plane. PLOS (pursuit and

line-of-sight) guidance algorithm [62] is used to perform this controller. In Figure 3.2a, the straight-line between two way-points,  $W_i = (x_i, y_i)$  and  $W_{i+1} = (x_{i+1}, y_{i+1})$ , and the position of aircraft,  $p = (x, y)$ , are shown.  $S$  is the target location that aircraft tries to reach. If  $S$  is taken as  $W_{i+1}$ , heading angle rate,  $\dot{\psi}_1$ , can be expressed as (3.12) by pure pursuit guidance law.

$$\psi_d = \arctan 2(y_{i+1} - y, x_{i+1} - x) \quad (3.11)$$

$$\dot{\psi}_1 = k_1(\psi_d - \psi) \quad (3.12)$$

where  $k_1$  is the gain.



**Figure 3.2** : Straight-Line and Arc Geometry: (a) Straight Line Path, (b) Circular Motion.

The LOS (line-of-sight) guidance law ensures that the angle between  $W_i$  and the aircraft is the same as that of the angle between  $W_i$  and  $W_{i+1}$ . The LOS guidance law is given by expression (3.17) with gain  $k_2$ .

$$\theta_i = \arctan 2(y_{i+1} - y_i, x_{i+1} - x_i) \quad (3.13)$$

$$\theta_a = \arctan 2(y - y_i, x - x_i) \quad (3.14)$$

$$R_a = \|W_i - p\| \quad (3.15)$$

$$d = R_a \sin(\theta_i - \theta_a) \quad (3.16)$$

$$\dot{\psi}_2 = k_2 d \sin(\theta_i - \theta_a) \quad (3.17)$$

The combined guidance law is given as (3.18) and bank angle is calculated as (3.19).

$$\dot{\psi} = \dot{\psi}_1 + \dot{\psi}_2 \quad (3.18)$$

$$\phi = \max \left\{ -\phi_{max}, \min \left\{ \phi_{max}, \arctan \left( \frac{\dot{\psi}_v}{g} \right) \right\} \right\} \quad (3.19)$$

**Turn controller:** Turn controller provides necessary bank angle to follow a circular trajectory that is centered in  $W_o = (x_o, y_o)$  with radius  $r$  as shown in Figure 3.2b where the position of aircraft is  $p = (x, y)$  and  $S$  is the target location that aircraft tries to reach. If  $S$  is taken as tangent point to the line joining the aircraft and the loiter circle center, then desired heading angle is calculated by expression (3.25). Then, heading angle rate,  $\dot{\psi}_1$ , for desired heading is ensured by expression (3.26).

$$\theta_a = \arctan 2(y - y_o, x - x_o) \quad (3.20)$$

$$\alpha = \begin{cases} \arccos\left(\frac{r}{d+r}\right) & \text{for left turn} \\ -\arccos\left(\frac{r}{d+r}\right) & \text{for right turn} \end{cases} \quad (3.21)$$

$$S = (x_s, y_s) \quad (3.22)$$

$$x_s = x_o + r \cos(\theta_a + \alpha) \quad (3.23)$$

$$y_s = y_o + r \sin(\theta_a + \alpha) \quad (3.24)$$

$$\psi_d = \arctan 2(y_s - y_o, x_s - x_o) \quad (3.25)$$

$$\dot{\psi}_1 = k_1(\psi_d - \psi) \quad (3.26)$$

Another heading angle rate,  $\dot{\psi}_2$ , is generated to decrease the cross-track error ( $d$ ) with gain  $k_2$  as expressed below.

$$d = \|W_o - p\| - r \quad (3.27)$$

$$\dot{\psi}_2 = k_2 d \quad (3.28)$$

The combined heading angle rate is provided as (3.29) and bank angle is calculated as (3.30).

$$\dot{\psi} = \dot{\psi}_1 + \dot{\psi}_2 \quad (3.29)$$

$$\phi = \max \left\{ -\phi_{max}, \min \left\{ \phi_{max}, \arctan \left( \frac{\dot{\psi}_v}{g} \right) \right\} \right\} \quad (3.30)$$

**Heading controller:** Heading controller turns the aircraft to desired heading by arranging the necessary bank angle and keeps the this heading value as stable. The heading angle rate and bank angle are calculated as follows:

$$\dot{\psi} = k(\psi_d - \psi) \quad (3.31)$$



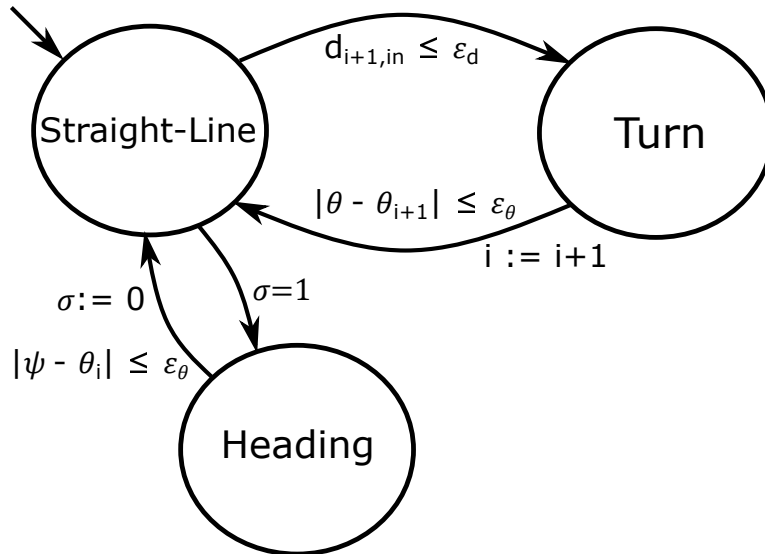
$$\phi = \max \left\{ -\phi_{max}, \min \left\{ \phi_{max}, \arctan \left( \frac{\dot{\psi}v}{g} \right) \right\} \right\} \quad (3.32)$$

This controller is only used to direct the aircraft to a new way-point that is not found in original flight plan.

**Transitions between horizontal controllers:** In any stage of the flight, only one of the straight-line, turn and heading controllers is operated as lateral controller as presented in Figure 3.3. The system is initialized with straight-line controller. Assume that the aircraft has a list of way-points and it is flying between  $wp_i$  and  $wp_{i+1}$ , where  $i$  symbolizes the current flight segment in the flight plan. All of the segments of the flight can be presented via lines between consecutive way-points and the transition between two lines can be presented as an arc as in fly-by transition approach. In this case, the reference trajectory consists of lines and arcs. The position  $(x_{i+1,in}, y_{i+1,in})$  defines the entry point of the arc between line  $l_{wp_i, wp_{i+1}}$  and line  $l_{wp_{i+1}, wp_{i+2}}$ . The straight-line controller calculates the bank angle to follow the line  $l_{wp_i, wp_{i+1}}$  until the transition condition  $d_{i+1,in} \leq \epsilon_d$ , which describes reaching the entry point of the arc and  $d_{i+1,in}$  is computed as follows:

$$d_{i+1,in} = \sqrt{(x_{i+1,in} - x)^2 + (y_{i+1,in} - y)^2} \quad (3.33)$$

where,  $(x, y)$  is the current position of the aircraft. Then the turn controller is activated and it produces the bank angle to follow the arc until catching the course of the next line  $l_{wp_{i+1}, wp_{i+2}}$ , which is symbolized as  $\theta_{i+1}$ . There is also a reset relation  $i := i + 1$  in the transition from *Turn* to *Straight – Line* that specifies passing the current trajectory segment. The heading controller is only triggered when a new way-point is added to the reference trajectory as the next way-point  $wp_{i+1}$  or the next way-point  $wp_{i+1}$  is modified. The switch  $\sigma$  turns to 1 when the next way-point  $wp_{i+1}$  is changed, then the heading controller changes the direction of the aircraft towards to the next way-point until the heading of the aircraft gets closer to the course of the line  $l_{wp_i, wp_{i+1}}$ . Afterwards, the switch of the special case  $\sigma$  turns to 0 and the straight-line controller takes over as the lateral controller. When the aircraft reaches the last way-point of the reference trajectory, the controllers are deactivated and the process is terminated.



**Figure 3.3 :** Transitions between Horizontal Controllers.

### 3.2.3.3 Speed controller

The last controller aims to hold the speed at the desired value for the cruise phase. It arranges the thrust to keep the speed with proportional control. The speed controller is only active during cruise phase because the longitudinal controller already adjusts the speed in climb and descent phases.

### 3.2.4 Simulation and trajectory prediction

It is important to emphasize the difference between simulation and trajectory prediction from the stand point of this study. The simulation refers to the proceed one step at a time. Let us consider a simulation environment that contains several aircraft, which fly according to different flight plans. The simulation process contains movements of the all aircraft in the environment and ATCo can intervene any aircraft at any time step of the simulation. During the decision process, ATCo uses the trajectory prediction as a tool to calculate the future states of the aircraft and decide which action is better for an aircraft. The trajectory prediction can be performed via reference trajectory to check the possible conflicts or modified trajectory to forecast the consequences of an action. Of course we will use same aircraft model and trajectory tracking algorithms for both simulation and trajectory prediction. However, they correspond to the different parts of the real operation. While the simulation environment corresponds to the virtual version of the real operation in an airspace,

the trajectory prediction refers to the calculations of the future states of the aircraft during decision process.

### **3.2.5 ATCo interventions**

The purpose of air traffic control is to ensure the safe and efficient flow of air traffic. To prevent collisions, an air traffic controller (ATCo) ensures that each aircraft maintains a minimum amount of empty space around it at all times. The interventions of the ATCo are denoted as maneuvers. At any time during the flight, the ATCo can give a maneuver to an aircraft to ensure its safety. In this section, we define the set of maneuvers that will be used for separation assurance. Note that the term 'Null Action' is used to refer to following the current trajectory without any change; it can also be given by an ATCo as a maneuver.

#### **3.2.5.1 Speed change**

A speed change corresponds to the modification of the desired speed. As presented before, the speed schedule contains the speed information according to flight level and the desired speed of the aircraft is the combination of the speed schedule and the speed change as in Eq. (3.8). At any time during the flight, ATCo can give a speed change to an aircraft by adjusting the  $V_c$ . The desired true airspeed of the aircraft is updated according to the speed change and the true airspeed is held at the desired value with the controllers.

#### **3.2.5.2 Altitude change**

As the speed change, the altitude change includes the adjustment of the schedule. Let us consider that the aircraft is flying between  $wp_i$  and  $wp_{i+1}$ , then planned altitude of the aircraft is presented in  $wp_{i+1}$  as  $h_{i+1}$ . The desired altitude of the aircraft refers to the sum of  $h_{i+1}$  and the altitude change  $H_c$ . When ATCo changes the value of  $H_c$  or  $V_c$  for an aircraft, this modification continuous until the end of the flight without affected from the other interventions. These variables can be set to another value only with another intervention of the ATCo to these variables.

### 3.2.5.3 Direct routing

An ATCo can make an aircraft skip a sequence of way-points, a maneuver known as Direct Routing. Direct routing can involve skipping next way-point or the next 2-3 way-points as shown in Figure 3.4. In Figure 3.4, three different scenarios are illustrated. An action is given to the aircraft at time  $t = 40s$  for each scenario. These actions, 'Null Action', 'Direct Routing 1' and 'Direct Routing 2' correspond to following the current trajectory, skipping the next way-point, and skipping the next two consecutive way-points, respectively. Before implementing the action, feasibility is always checked. If the action is not feasible, then 'Null Action' is implemented instead of given action. For example, it is not possible to skip the next four consecutive way-points in the scenario in Figure 3.4, so 'Null Action' is performed when 'Direct Routing 4' is requested.

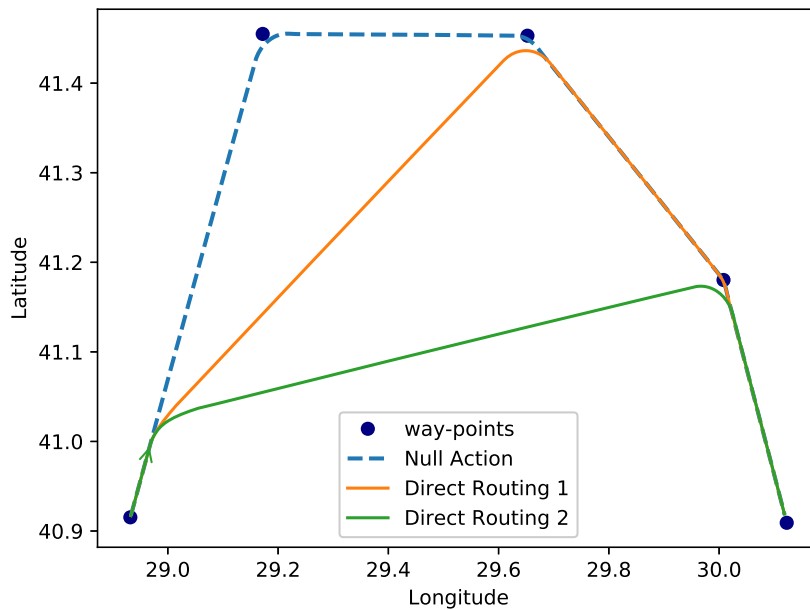


Figure 3.4 : Direct Routing Action.

### 3.2.5.4 Course change

Course change refers to changing the current reference course of the aircraft. It is implemented by modifying the next way-point  $w_{p_{i+1}}$  in the flight plan, using Equation

(3.38).

$$\theta_i = \arctan 2(y_{i+1} - y_i, x_{i+1} - x_i) \quad (3.34)$$

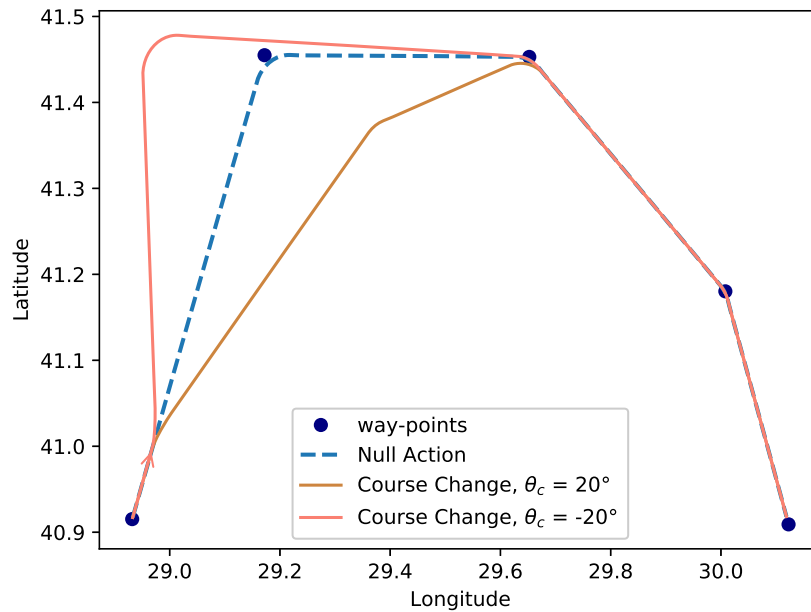
$$d_{i+1} = \sqrt{(x_{i+1} - x)^2 + (y_{i+1} - y)^2} \quad (3.35)$$

$$x_{new} = x + d_{i+1} \cos(\theta_i - \theta_c) \quad (3.36)$$

$$y_{new} = y + d_{i+1} \sin(\theta_i - \theta_c) \quad (3.37)$$

$$wp_{i+1} = (x_{new}, y_{new}, h_{i+1}) \quad (3.38)$$

where,  $\theta_i, d_{i+1}$  and  $\theta_c$  define the current reference course, distance from the aircraft to the next way-point, and course change, respectively. The course change  $\theta_c$  can be positive or negative depending on the direction of the motion. While one of them decreases the length of the path and acts like a direct routing action, the other one causes delay, as illustrated in the scenarios in Figure 3.5.



**Figure 3.5 : Course Change.**

### 3.2.5.5 Vector for spacing

Vector for spacing (VFS) is a kind of delaying motion. It deviates the aircraft from the reference trajectory to cause a delay. It is implemented by inserting a new way-point between the current  $wp_i$  and  $wp_{i+1}$ , as calculated via Equation (3.41). After adding

the new way-point to the flight plan, the new way-point will be the  $wp_{i+1}$ .

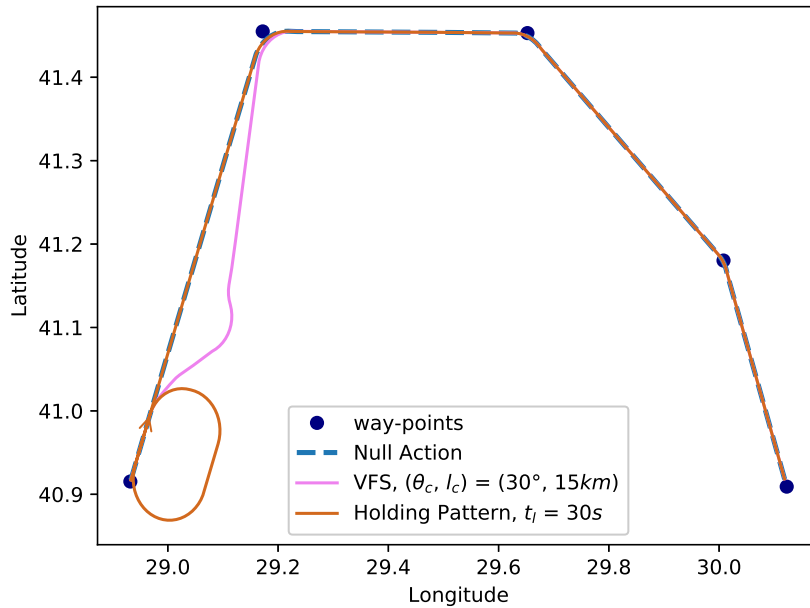
$$x_{new} = x + l_c \cos(\theta_i - \theta_c) \quad (3.39)$$

$$y_{new} = y + l_c \sin(\theta_i - \theta_c) \quad (3.40)$$

$$wp_{new} = (x_{new}, y_{new}, h_{i+1}) \quad (3.41)$$

where,  $l_c$  and  $\theta_c$  symbolize the distance from the aircraft to the new way-point and course change, respectively. Both  $l_c$  and  $\theta_c$  are the inputs of VFS action. An example scenario is illustrated in Figure 3.6. This maneuver always contains a feasibility assessment before implementation. Before adding  $wp_{new}$  to the flight plan, the inequality (3.42) is checked. If this inequality is ensured then the action is given, otherwise 'Null Action' is implemented.

$$\sqrt{(x_{i+1,in} - x)^2 + (y_{i+1,in} - y)^2} \geq 1.5l_c \quad (3.42)$$



**Figure 3.6 :** Vector for Spacing and Holding Pattern.

### 3.2.5.6 Holding pattern

Holding defines a racetrack pattern based on a reference position. In this study, this reference position is given as the position of the aircraft when the holding maneuver is activated. The procedure begins with a semi circle, followed by a straight-line, a semi-circle and another straight-line. A standard holding pattern uses the right-side as

the turn direction and takes approximately 4 minutes to complete. Each semi-circle takes one minute; in the same way, we also standardize the flight duration for the  $180^\circ$  turn as approximately one minute via appropriate turn radius, which is calculated using Equation (3.44). However, the flight duration of straight ahead sections  $t_l$  is given as input parameter for the pattern. The example scenario presented in Figure 3.6 contains a holding pattern for  $t_l = 30s$ , which takes approximately 3 minutes.

$$\phi_r = \max \left\{ -\phi_{max}, \min \left\{ \phi_{max}, \arctan \left( \frac{\psi_r v}{g} \right) \right\} \right\} \quad (3.43)$$

$$r_h = \frac{v^2}{g \tan(\phi_r)} \quad (3.44)$$

$$l_h = vt_l \quad (3.45)$$

where,  $\phi_r$  and  $\psi_r$  are the reference bank angle and turn rate for the calculation of the radius.  $\psi_r$  is chosen as  $3^\circ$  per second to complete  $180^\circ$  turn in one minute.  $l_h$  defines the length of the straight ahead sections, which is adjusted via  $t_l$ . The holding pattern is implemented in level flight, whereas all of the other maneuvers can be implemented during climb, descent or cruise. The aircraft can climb or descent after completing the holding pattern.

### 3.3 Maneuver-Based Optimal Strategy Generation for Autonomous Air Traffic Controller

An airspace can be discretized using a grid structure to speed up the conflict detection and resolution process. In an en-route airspace, the vertical separation minima  $d_s^v$  is  $1000ft$  where Reduced Vertical Separation Minima (RVSM) apply and the horizontal separation minima  $d_s^h$  is  $5nm$  that are also used as minimum separation distances in this study. Therefore, in discretized airspace, the flight levels are defined in thousands of feet (e.g., FL290, FL300, FL310 etc.) and the edge length of the square cells is chosen as  $10nm$ . In this study, we use double-layer grids for each flight level in sector to discretize the airspace. A double-layer grid consists of two non-overlapping grids where the edge lengths are equal, and the corners of the cells in the first grid overlap the centers of the cells in the second grid. Assume that two aircraft are flying north south direction at two neighboring cells in the red grid with violation of the separation,

in this case, the conflict can not be detected by checking cell occupancy in the red grid. Thus, we use double-layer grids to catch overlooked loss of separation at the borders of the cells. We map a predicted trajectory into the grid structure via matrices, where each element of a matrix corresponds to a specific cell in the grid. We introduce the mapping process for a grid, then the same process can be applied for the other grid. We focus on the mapping process, and then continue with formulating the optimization problem.

The trajectory tracking algorithms compute an action trajectory  $\tilde{u}$ , which is a function of the current situation and flight plan. Starting from some initial state  $x(t_0)$  at time  $t_0$ , a state trajectory is derived from an action trajectory  $\tilde{u}$  as:

$$x(t_0 + T) = x(t_0) + \int_{t_0}^{t_0+T} f(x(t), u(t)) dt \quad (3.46)$$

which integrates the state transition equation  $\dot{x} = f(x, u)$  from the initial condition  $x(t_0)$ . Let  $\mathcal{A}$  be the set of aircraft and  $\mathcal{I}$  be the set of interventions or maneuvers. For an aircraft  $a \in \mathcal{A}$ , the  $\tilde{u}$  is also affected from maneuver  $m \in \mathcal{I}$  at time  $t_0$ . Let  $\tilde{x}(a, m)$  denote the state trajectory or predicted trajectory over the interval  $[t_0, t_0 + T]$ , obtained by integrating (3.46), for the aircraft  $a \in \mathcal{A}$  with the maneuver  $m \in \mathcal{I}$ , where the planning horizon is  $T$ .

Let  $\mathcal{M}_{k \times l}$  be the set of  $k \times l$  real matrices. A matrix  $A \in \mathcal{M}_{k \times l}$  is called a Boolean matrix if its entries  $a_{i,j} \in \mathcal{D}$ , where  $\mathcal{D} = \{1, 0\}$ . The set of  $k \times l$  Boolean matrices is denoted by  $\mathcal{B}_{k \times l}$ . Let  $A = (a_{i,j}), B = (b_{i,j}) \in \mathcal{B}_{k \times l}$ . Then,  $\neg A = (-a_{i,j})$ ,  $A \wedge B = (a_{i,j} \wedge b_{i,j})$  and  $A \vee B = (a_{i,j} \vee b_{i,j})$ . Let  $L = (l_{i,j}) \in \mathcal{M}_{k \times l}$ , then the Hadamard product  $B \circ L$  is given as  $B \circ L = (b_{i,j} \cdot l_{i,j})$ . Let  $N = (n_{i,j}) \in \mathcal{M}_{k \times l}$ , then the min and max operations between  $L$  and  $N$  are described as  $\min(L, N) = (\min(l_{i,j}, n_{i,j}))$  and  $\max(L, N) = (\max(l_{i,j}, n_{i,j}))$ , respectively.

Let  $\mathcal{F}$  be the set of flight levels (FL). Let us define  $g_{a,m}^f \in \mathcal{B}_{k \times l}$ ,  $s_{a,m}^f \in \mathcal{M}_{k \times l}$  and  $e_{a,m}^f \in \mathcal{M}_{k \times l}$  that contain location, entry time and exit time information for the corresponding cell of the grid, respectively. These matrices are calculated for the aircraft  $a \in \mathcal{A}$  using  $\tilde{x}(a, m)$ . Let  $\mathcal{G}$  denote a grid and  $c_{i,j,k}$  denote a specific cell in this grid, where  $(i, j)$  corresponds to horizontal location of the cell and  $k$  corresponds to flight level  $f \in \mathcal{F}$ . The cell is a three-dimensional object and  $k$  specifies the center of the object along altitude axis. The height of the cell is  $2\Delta h - 2\varepsilon_h$ , where  $\Delta h$  defines the



altitude difference between two consecutive flight levels and  $\varepsilon_h$  symbolizes the altitude buffer. In this case, the two consecutive cells  $c_{i,j,k}$  and  $c_{i,j,k+1}$  have overlapping volume between  $FL_k$  and  $FL_{k+1}$ . Therefore, an aircraft is assigned to both cells when climbing or descending between  $FL_k$  and  $FL_{k+1}$ . Let the entry in the  $i^{th}$  row and  $j^{th}$  column of the matrix  $g_{a,m}^f$ ,  $s_{a,m}^f$  or  $e_{a,m}^f$  be denoted as  $(\cdot)_{i,j}$ . The entries of these matrices are expressed as follows:

$$(g_{a,m}^f)_{i,j} = \begin{cases} 1 & \tilde{x}(a,m) \cap c_{i,j,f} \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \quad (3.47)$$

$$(s_{a,m}^f)_{i,j} = \begin{cases} t_{i,j,f}^{in} - t_b & \tilde{x}(a,m) \cap c_{i,j,f} \neq \emptyset \\ M & \text{otherwise} \end{cases} \quad (3.48)$$

$$(e_{a,m}^f)_{i,j} = \begin{cases} t_{i,j,f}^{out} + t_b & \tilde{x}(a,m) \cap c_{i,j,f} \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \quad (3.49)$$

where  $M$  is a large enough number, typically chosen greater than an upper bound on the planning completion time. The parameters  $t_{i,j,f}^{in}$  and  $t_{i,j,f}^{out}$  denote the entry time to the cell  $c_{i,j,f}$  and exit time from the cell  $c_{i,j,f}$ , respectively. The parameter  $t_b$  is the time buffer.

Let us consider the case in which maneuver is the null action, which is symbolized as  $m_0$ . Let  $a_p$  and  $a_q$  be the different aircraft in the airspace. The matrix  $g_{a_p,q,m_0}^f$  presents the intersections of the predicted trajectories without any interventions, which is given as follows:

$$g_{a_p,q,m_0}^f = g_{a_p,m_0}^f \wedge g_{a_q,m_0}^f \quad (3.50)$$

Let us introduce two new matrices  $s_{a_p,q,m_0}^f$  and  $e_{a_p,q,m_0}^f$  that contain the maximum values of the entry times and the minimum values of the exit times for the simultaneously occupied cells. The matrix  $\delta_{a_p,q,m_0}^f$  contains the time differences between  $s_{a_p,q,m_0}^f$  and  $e_{a_p,q,m_0}^f$ . These matrices are calculated as follows:

$$s_{a_p,q,m_0}^f = \max(g_{a_p,q,m_0}^f \circ s_{a_p,m_0}^f, g_{a_p,q,m_0}^f \circ s_{a_q,m_0}^f) \quad (3.51)$$

$$e_{a_p,q,m_0}^f = \min(g_{a_p,q,m_0}^f \circ e_{a_p,m_0}^f, g_{a_p,q,m_0}^f \circ e_{a_q,m_0}^f) \quad (3.52)$$

$$\delta_{a_p,q,m_0}^f = s_{a_p,q,m_0}^f - e_{a_p,q,m_0}^f \quad (3.53)$$

The entry  $(\delta_{a_p, a_q, m_0}^f)_{i,j}$  of the matrix  $\delta_{a_p, a_q, m_0}^f$  corresponds to the time difference between aircraft  $a_p$  and  $a_q$  for the occupation of the cell  $c_{i,j,f}$ . If  $(\delta_{a_p, a_q, m_0}^f)_{i,j} \geq 0$ , then there is no conflict between them in the cell  $c_{i,j,f}$ , because they are not in the cell simultaneously. In the case of  $(\delta_{a_p, a_q, m_0}^f)_{i,j} < 0$ , there may be a conflict between them in the corresponding cell. Let  $\tau_{a_p, a_q, m_0}^f$  and  $\tau_{a_p, a_q, m_0}$  denote the sets of ordered pairs  $(t_1, t_2)$  with first element from  $s_{a_p, a_q, m_0}^f$  and second element from  $e_{a_p, a_q, m_0}^f$  for  $f \in F$ . These sets are defined as follows:

$$\tau_{a_p, a_q, m_0}^f = \{((s_{a_p, a_q, m_0}^f)_{i,j}, (e_{a_p, a_q, m_0}^f)_{i,j}) : (\delta_{a_p, a_q, m_0}^f)_{i,j} < 0\} \quad (3.54)$$

$$\tau_{a_p, a_q, m_0} = \bigcup_{f \in F} \tau_{a_p, a_q, m_0}^f \quad (3.55)$$

The set  $\tau_{a_p, a_q, m_0}$  contains all time pairs that are candidate time intervals for the loss of separation between the aircraft  $a_p$  and  $a_q$  with  $m_0$ . If  $\tau_{a_p, a_q, m_0} = \emptyset$ , then the two aircraft have already necessary separation for the planning horizon  $T$ , otherwise there may be a loss of separation at the candidate time intervals between the aircraft  $a_p$  and  $a_q$ . The set of conflicted aircraft  $\mathcal{A}_c$  is described as follows:

$$\begin{aligned} \mathcal{A}_c = & \{a_p : \forall a_p \in \mathcal{A}, \forall a_q \in \mathcal{A} \setminus \{a_p\}, \forall (t_1, t_2) \in \tau_{a_p, a_q, m_0}, \\ & \exists t \in [t_1, t_2] (\|h_{a_p, m_0}(t) - h_{a_q, m_0}(t)\| < d_s^v \wedge \\ & \|(x(t), y(t))_{a_p, m_0} - (x(t), y(t))_{a_q, m_0}\| < d_s^h)\} \end{aligned} \quad (3.56)$$

where,  $d_s^v$  and  $d_s^h$  symbolize vertical separation minima and horizontal separation minima, respectively. The parameters  $(x(t), y(t), h(t))_{a_p, m_0}$  and  $(x(t), y(t), h(t))_{a_q, m_0}$  belong to  $\tilde{x}(a_p, m_0)$  and  $\tilde{x}(a_q, m_0)$ , respectively. The set of separated aircraft  $\mathcal{A}_s$ , which have separation without ATCo's intervention, can be presented as the relative complement of  $\mathcal{A}_c$  with respect to the set  $\mathcal{A}$  as follows:

$$\mathcal{A}_s = \mathcal{A} \setminus \mathcal{A}_c \quad (3.57)$$

Let  $a_n \in \mathcal{A}_c$  and  $a_r \in \mathcal{A}_c$  be two different aircraft in the airspace. Let matrix  $g_{a_n}^f$  be the logical disjunction of the all matrices  $g_{a_n, m}^f$  as in Eq. (3.58) that presents the all visitable cells at the flight level  $f$  for the aircraft  $a_n$  with the all possible maneuvers in the set  $\mathcal{I}$ . By using the matrices  $g_{a_n}^f$  and  $g_{a_r}^f$ , the matrix  $g_{a_n, a_r}^f$  can be expressed as

in (3.59), which contains intersections of the all possible trajectories of the aircraft  $a_n$  and  $a_r$  at flight level  $f$  via the maneuver set  $\mathcal{I}$ .

$$g_{a_n}^f = \bigvee_{m \in \mathcal{I}} g_{a_n, m}^f \quad \text{and} \quad g_{a_r}^f = \bigvee_{m \in \mathcal{I}} g_{a_r, m}^f \quad (3.58)$$

$$g_{a_n, r}^f = g_{a_n}^f \wedge g_{a_r}^f \quad (3.59)$$

Let  $m_u \in \mathcal{I}$  and  $m_v \in \mathcal{I}$  be interventions for the aircraft  $a_n$  and  $a_r$ , respectively. Then, the time matrices  $s_{a_n, r, m_u, v}^f$ ,  $e_{a_n, r, m_u, v}^f$  and  $\delta_{a_n, r, m_u, v}^f$  can be calculated via  $g_{a_n, r}^f$  as follows:

$$s_{a_n, r, m_u, v}^f = \max(g_{a_n, r}^f \circ s_{a_n, m_u}^f, g_{a_n, r}^f \circ s_{a_r, m_v}^f) \quad (3.60)$$

$$e_{a_n, r, m_u, v}^f = \min(g_{a_n, r}^f \circ e_{a_n, m_u}^f, g_{a_n, r}^f \circ e_{a_r, m_v}^f) \quad (3.61)$$

$$\delta_{a_n, r, m_u, v}^f = s_{a_n, r, m_u, v}^f - e_{a_n, r, m_u, v}^f \quad (3.62)$$

As mentioned before, there is no conflict in the corresponding cell between the aircraft  $a_n$  and  $a_r$  with the maneuver  $m_u$  and  $m_v$  if  $(\delta_{a_n, r, m_u, v}^f)_{i, j} \geq 0$ , there may be a loss of separation otherwise. Let us consider a special case that the aircraft  $a_n$  visits a specific cell  $c_{i, j, f}$  with a specific maneuver  $m_1$ , whereas it does not visit this cell with another maneuver  $m_2$  and the aircraft  $a_r$  also visits this cell with maneuver  $m_v$ . In this case, the  $(g_{a_n, r}^f)_{i, j}$  turns to 1. When aircraft  $a_n$  flies with maneuver  $m_2$ , this situation does not cause any problem in terms of conflict detection because of the large number  $M$  in the  $(s_{a_n, m_2}^f)_{i, j}$ . In any case,  $(\delta_{a_n, r, m_2, v}^f)_{i, j} \geq 0$  because of the large number  $M$  in the  $(s_{a_n, m_2}^f)_{i, j}$ , where the aircraft  $a_n$  does not visit the cell  $c_{i, j, f}$  with maneuver  $m_2$ .

The set  $\tau_{a_n, r, m_u, v}$  can be obtained as previously presented:

$$\tau_{a_n, r, m_u, v}^f = \{((s_{a_n, r, m_u, v}^f)_{i, j}, (e_{a_n, r, m_u, v}^f)_{i, j}) : (\delta_{a_n, r, m_u, v}^f)_{i, j} < 0\} \quad (3.63)$$

$$\tau_{a_n, r, m_u, v} = \bigcup_{f \in F} \tau_{a_n, r, m_u, v}^f \quad (3.64)$$

Let us define a new set  $P_{cc}$  that consists of ordered quadruples. Each quadruple contains a specific maneuver pair for two aircraft that cause a loss of separation when these

maneuvers are implemented together. The set  $P_{cc}$  is given by the following expression:

$$\begin{aligned}
\mathcal{P}_{cc} = & \{(a_n, a_r, m_u, m_v) : \forall a_n \in \mathcal{A}_c, \forall a_r \in \mathcal{A}_c \setminus \{a_n\}, \\
& \forall m_u \in \mathcal{I}, \forall m_v \in \mathcal{I}, \forall (t_1, t_2) \in \tau_{a_n, r, m_u, v}, \\
& \exists t \in [t_1, t_2] (\|h_{a_n, m_u}(t) - h_{a_r, m_v}(t)\| < d_s^v \wedge \\
& \|(x(t), y(t))_{a_n, m_u} - (x(t), y(t))_{a_r, m_v}\| < d_s^h)\}
\end{aligned} \tag{3.65}$$

Let us consider the case in which the aircraft  $a_n \in \mathcal{A}_c$ , whereas  $a_r \in \mathcal{A}_s$ . ATCo prefers null action than other maneuvers when an aircraft has separation. Therefore, in this study, the aircraft  $a_r \in \mathcal{A}_s$  takes null action, while one of the maneuvers in the set  $\mathcal{I}$  can be given to the aircraft  $a_n \in \mathcal{A}_c$ . The matrix  $g_{a_n, r}^f$  can be calculated with the Eq. (3.59) as previously presented. However, the matrix  $g_{a_r}^f$  should be described as in the expression (3.66) instead of the expression (3.58). The rest of the matrices and sets are calculated via expressions from (3.60) to (3.64), where  $m_v$  is taken as  $m_0$ .

$$g_{a_n}^f = \bigvee_{m \in \mathcal{I}} g_{a_n, m}^f \quad \text{and} \quad g_{a_r}^f = g_{a_r, m_0}^f \tag{3.66}$$

Then, the new set  $P_{cs}$  that consists of the restricted maneuvers for specific aircraft in the set  $\mathcal{A}_c$  because of the violation of the separation with an aircraft  $a_r \in \mathcal{A}_s$  can be described as follows:

$$\begin{aligned}
\mathcal{P}_{cs} = & \{(a_n, m_u) : \forall a_n \in \mathcal{A}_c, \forall a_r \in \mathcal{A}_s, \forall m_u \in \mathcal{I}, \\
& \forall (t_1, t_2) \in \tau_{a_n, r, m_u, 0}, \exists t \in [t_1, t_2] \\
& (\|h_{a_n, m_u}(t) - h_{a_r, m_0}(t)\| < d_s^v \wedge \\
& \|(x(t), y(t))_{a_n, m_u} - (x(t), y(t))_{a_r, m_0}\| < d_s^h)\}
\end{aligned} \tag{3.67}$$

By using the defined sets and matrices, an integer linear programming (ILP) is formulated to ensure the safety in the airspace, while optimizing the separation assurance process as follows:

$$\min \sum_{m \in \mathcal{I}} \sum_{a \in \mathcal{A}_c} (f_{a, m} + \mathcal{C} \mathcal{I} t_{a, m}) x_{a, m} - \sum_{a \in \mathcal{A}_c} c_0 x_{a, m_0} \tag{3.68}$$

subject to

$$x_{a_n, m_u} + x_{a_r, m_v} \leq 1 \quad \forall (a_n, a_r, m_u, m_v) \in P_{cc} \quad (3.69)$$

$$x_{a_n, m_u} = 0 \quad \forall (a_n, m_u) \in P_{cs} \quad (3.70)$$

$$\sum_{m \in \mathcal{I}} x_{a, m} = 1 \quad \forall a \in \mathcal{A}_c \quad (3.71)$$

$$x_{a, m} \in \{0, 1\} \quad \forall a \in \mathcal{A}_c, \forall m \in \mathcal{I} \quad (3.72)$$

where,  $x_{a, m}$  for all  $a \in \mathcal{A}_c$  and  $m \in \mathcal{I}$  are binary decision variables such that the variable  $x_{a, m}$  is equal to 1 if aircraft  $a$  takes maneuver  $m$  as an action, and zero otherwise. The parameters  $f_{a, m}$  and  $t_{a, m}$  define the fuel consumption and travel duration, respectively. The parameter  $CI$  is the cost index. The combination of the  $f_{a, m}$  and  $t_{a, m}$  is defined as the total cost of the flight and presented as the first term of the objective function (3.68), whereas the second term of the objective function prioritizes the null action. In implementations, the  $c_0$  will be chosen much greater than  $CI$  to prevent intervention if it is unnecessary in terms of separation. Constraint (3.69) ensure that restricted maneuver pair, which cause loss of separation, are not given together. Constraint (3.70) ensure that an action that cause a conflict with an aircraft  $a_r \in \mathcal{A}_s$ , which already has separation, is not chosen.

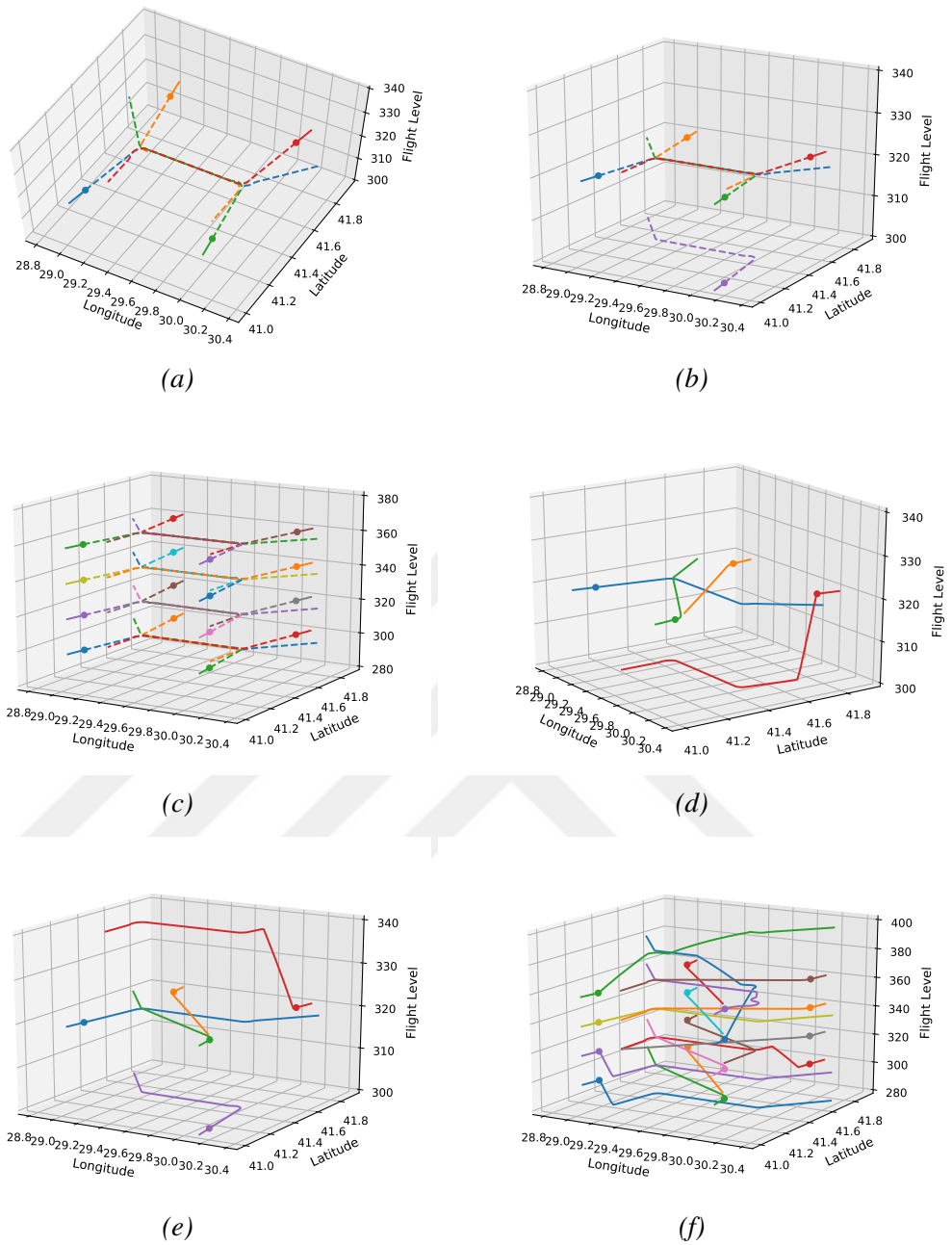
### 3.4 Separation Assurance with Autonomous ATCo

In this section, several example scenarios are analyzed to show the working principle and validity of the optimization-based autonomous ATC system. In simulations, the performance parameters of Boeing 737-800 are used for all aircraft in the sector. During the implementations, the set of interventions  $\mathcal{I}$  consists of 13 different maneuvers as presented in Table 3.1. This set can be rearranged by adding new maneuvers or removing some of the maneuvers. The chosen maneuvers are rational and applicable to the real ATC system. When adding a new maneuver, it is important to consider the constraints in the real system. For example, the speed change can not be high because of the dynamic limitations and high fuel consumption or the altitude change should be proportional to the amount of 2000ft because of the fact that eastbound flights use odd flight levels, whereas westbound flights use even flight levels with 1000ft vertical separation during standard cruise operation.

**Table 3.1** : The Set of Maneuvers.

Maneuver ID	Maneuver
0	Null Action
1	Direct Routing 1
2	Direct Routing 2
3	Speed Change, $V_c = 10 \text{ m/s}$
4	Speed Change, $V_c = -10 \text{ m/s}$
5	Altitude Change, $H_c = 2000 \text{ ft}$
6	Altitude Change, $H_c = 4000 \text{ ft}$
7	Altitude Change, $H_c = -2000 \text{ ft}$
8	Vector for Spacing, $(\theta_c, l_c) = (30^\circ, 15 \text{ km})$
9	Vector for Spacing, $(\theta_c, l_c) = (45^\circ, 15 \text{ km})$
10	Course Change, $\theta_c = 20^\circ$
11	Course Change, $\theta_c = -20^\circ$
12	Holding Pattern (Circular)

We perform three different scenarios to demonstrate the basic working principles and validity of the autonomous ATCo as illustrated in Figure 3.7. In the scenario 1, four aircraft fly towards each other in the *FL320* as shown in Figure 3.7a, where dashed lines correspond to reference trajectories of the aircraft. The ATCo is activated at  $t = 50\text{s}$  and it generates the optimum actions, then aircraft follow the intervened trajectories after ATCo action request that are direct routing for the green and orange aircraft and altitude change for the red aircraft as illustrated in Figure 3.7d. After ATCo's intervention, the conflicts are resolved and there is no collision during simulation. As presented in Figure 3.7d, ATCo decreases the red aircraft to prevent the collision. It prefers descent to climb because of less fuel consumption during descent. The scenario 2 contains an additional aircraft in the airspace as shown in Figure 3.7b and this aircraft is in the set of separated aircraft  $\mathcal{A}_s$ , which has separation with the rest of the traffic. As presented in Figure 3.7e, ATCo increases the altitude of the red aircraft instead of decrease as in previous scenario. It is shown in this scenario that the intervention to a conflicted aircraft does not cause a loss of separation with a separated aircraft. In the scenario 3, which is illustrated in Figure 3.7c, we perform a complex scenario that consists of four conflicted aircraft in the four different flight levels and altitude change of an aircraft can cause another conflict with an aircraft in the other flight levels if this situation is not evaluated during conflict resolution process. The autonomous ATCo resolves the conflicts in this scenario at  $t = 50\text{s}$  and there is no



**Figure 3.7 :** Example Scenarios: (a) Scenario 1: reference trajectories, (b) Scenario 2: reference trajectories, (c) Scenario 3: reference trajectories, (d) Scenario 1: simulation result with conflict resolution at  $t = 50s$ , (e) Scenario 2: simulation result with conflict resolution at  $t = 50s$ , (f) Scenario 3: simulation result with conflict resolution at  $t = 50s$ .

collision during simulation after conflict resolution as presented in Figure 3.7f. Let us focus on the aircraft that operate in  $FL360$ . The blue aircraft is climbing from  $FL340$  to  $FL380$  that has a trajectory intersection with the purple aircraft in  $FL360$ . The potential collision between them is prevented by giving vector for spacing to the purple aircraft. Because of the vector for spacing, the purple aircraft has also separation with the brown aircraft and the red aircraft in the same flight level. As presented in this

example, the conflict resolution mechanism considers climbing and descending traffic when ensuring the safety in a flight level and an intervention to a specific aircraft does not cause an additional conflict with another aircraft. Moreover, as shown in this scenario, autonomous ATCo can manage a complex traffic that is really challenging for a human operator.

### 3.5 Capacity Improvement with Autonomous ATCo

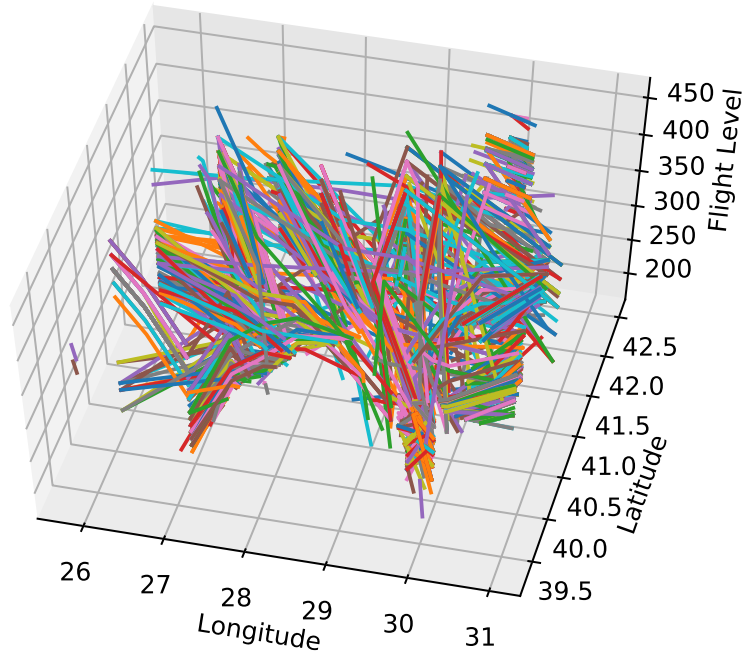
In the current ATM system, the maximum amount of workload that ATCos are able to sustain imposes the limits on the capacity of the airspace. In an automated ATM system, the capacity of the airspace should be redetermined according to designed algorithms. This section presents the identification method of the airspace capacity according to developed autonomous air traffic control system. The capacity estimation bases on making simulations for different traffic densities to obtain the critical capacity of the system.

#### 3.5.1 Stochastic traffic simulation environment

We construct a traffic simulation environment to simulate the daily traffic in an airspace with desired throughput. The traffic simulator consists of two stochastic processes. The first process determines which reference trajectory is followed by an aircraft and the second stochastic process generates the entry time of the aircraft to the airspace.

Let  $\mathcal{R}_t$  be the set of the reference trajectories such that  $\mathcal{R}_t = \{WP_1, WP_2, \dots, WP_M\}$ , where  $WP_j = \{wp_0, wp_1, \dots, wp_k\}$  and  $wp_i \in \mathbb{R}^3$ . Let  $\mathcal{R}_p$  denote the set of usage probabilities of the reference trajectories in  $\mathcal{R}_t$  such that  $\mathcal{R}_p = \{p_1, p_2, \dots, p_M\}$  and  $\sum_{i=1}^M p_i = 1$ . A reference trajectory  $r_t \in \mathcal{R}_t$  is assigned to the aircraft  $a$  randomly based on the probabilities in  $\mathcal{R}_p$ . The set  $\mathcal{R}_t$  and  $\mathcal{R}_p$  are extracted from the real flight plan data of the aircraft, which operate in the corresponding airspace. In Figure 3.8, the reference trajectories of the all aircraft that operate during a standard day in the ISTANBUL ACC (area control centre) are illustrated. Some of the reference trajectories are flown more than others. The utilization frequency of a reference trajectory is used to generate the usage probability of this flight plan and all of the reference trajectories in the real operation are assigned to the set  $\mathcal{R}_t$  by removing duplicates.





**Figure 3.8** : Flight Plans in ISTANBUL ACC for a Standard Day.

After constructing the simulation environment for a sector with the sets  $\mathcal{R}_t$  and  $\mathcal{R}_p$ , the only input for the environment is the traffic density  $N$  that denotes the total number of flights during the day. The second stochastic process determine the entry times of the flights into the airspace according to  $N$ . We want to keep the workload of the ATCo approximately at the same level during operation and operating with a flat demand profile rather than having peak and off-peak hours is a fact when the demand is close to the capacity of the airspace. Therefore, the entry times of the flights during the day are modelled as a Poisson process. It is more convenient to define a Poisson process in terms of the sequence of interarrival times,  $X_1, X_2, \dots$ , which are defined to be independent and identically distributed (i.d.d.). A renewal process is an arrival process for which the sequence of interarrival times is a sequence of i.d.d random variables and a Poisson process is a renewal process in which the interarrival intervals have an exponential distribution function; i.e., for some real  $\lambda > 0$  each  $X_i$  has the density  $f_X(x) = \lambda \exp(-\lambda x)$  for  $x \geq 0$ . To simulate a Poisson process with rate  $\lambda$ , the i.d.d. random variables  $X_1, X_2, \dots, X_N$  are generated, where  $X_i \sim Exponential(\lambda)$ . Then, the arrival times are given by  $T_i = \sum_{k=1}^i X_k$ . In our case, the arrival times correspond to the entry times of the aircraft into the airspace and  $\lambda$  equals to  $86400/N$ , where  $N$  is the traffic density and 86400 refers to 24 hours in terms of seconds.

In the simulation environment, each aircraft enters into the airspace at its entry time, however there is also a collision detection mechanism at the border, which shifts the entry time of a problematic aircraft. If the incoming aircraft cause a collision with another aircraft when it enters into the airspace, the incoming aircraft is held at the entrance until a clearance.

### 3.5.2 Breaking point analysis

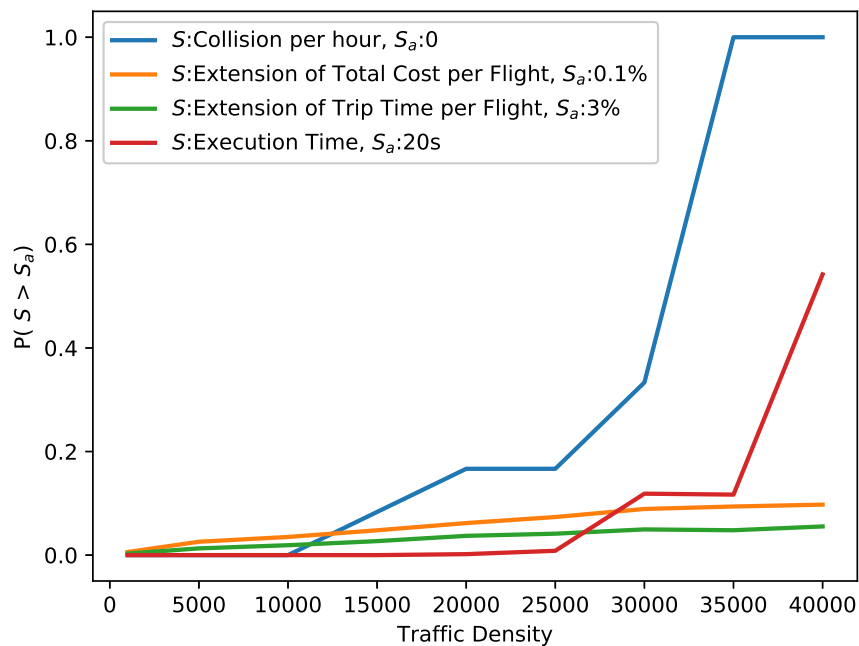
The breaking point corresponds to the critical capacity of the airspace in terms of a specific metric, where the probability of the metric greater than the acceptable metric level has a sharp transition. This metric can be about safety, performance, efficiency, etc. Let us  $S$  be a specific metric and  $S_a$  be the acceptable level of this metric. The number of aircraft  $n$  is the breaking point or the metric-specific capacity of the airspace if  $P(S(n-1) > S_a) < b$  and  $P(S(n+1) > S_a) > 1-b$ , where  $b \in (0, 0.5)$  and  $S(n)$  is a non-decreasing function [63]. The capacity of an airspace can be affected from several metrics. In this case, a critical capacity value  $n_i$  is generated for each metric  $S_i$  and minimum of them is taken as the capacity of the airspace,  $n = \min(n_1, n_2, \dots, n_l)$ . It is also possible to define critical capacity as a range instead of a single number. In this case, the  $n$  is defined as a range  $[n^l, n^u]$  and it corresponds to the critical capacity of the airspace if  $P(S(n^l-1) > S_a) < b$  and  $P(S(n^u+1) > S_a) > 1-b$ , where  $b \in (0, 0.5)$ .

In this study, we use four different metrics to determine the airspace capacity. The collision per hour is presented as safety metric that corresponds to the number of collision in the airspace during an hour and the acceptable limit of this metric is taken as 0. The second and third metrics are about efficiency that are defined as extension of total cost per flight and extension of travel duration per flight. As previously presented, the total cost corresponds to the combination of the fuel consumption and travel duration as  $f_{a,m} + \mathcal{C} \mathcal{I} t_{a,m}$ . The acceptable limits of them are defined as 0.1% and 3%, respectively. The last metric is about the time performance of the optimization process. In the simulation environment, the presented optimization problem is repeatedly solved with time period  $\delta$  to ensure the safety in the airspace. If the execution time of the optimization process is smaller than  $\delta$  seconds, it will be applicable with time period  $\delta$ . Therefore, the last metric is chosen as execution time of the optimization process and the acceptable limit is taken as  $\delta$  seconds. In

implementations, the optimization-based ATCo generates actions every 20 seconds, so the acceptable limit is taken as 20 seconds. If the time period of the algorithm creates a bottleneck in the capacity, it can be increased easily or more powerful hardware can be used to improve the time performance of the algorithm. However, evaluation of the time performance is necessary to check the applicability of the chosen time period.

### 3.5.3 Simulations in stochastic traffic environment

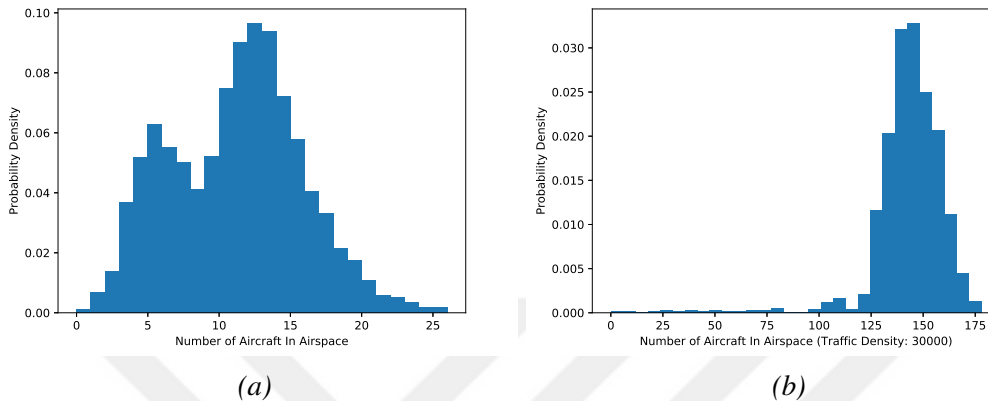
We perform simulations for ISTANBUL ACC in the presented stochastic traffic environment with traffic densities from 1000 to 40000 flights per day. During simulations, optimization-based autonomous ATCo intervenes the traffic every 20 seconds.



**Figure 3.9 :** Airspace Capacity Estimation.

The probabilities of the presented metrics with respect to traffic density are obtained from the results of the simulations to redefine the airspace capacity as illustrated in Figure 3.9. As shown in this figure, second and third metrics, which are extension of total cost per flight and extension of trip time per flight, do not have a breaking point in the interval between 1000 and 40000 flights per day. Thus, they do not cause a bottleneck to the airspace capacity. The main metric that defines the critical capacity is about safety. The collision per hour has a breaking point around 30000 flights per

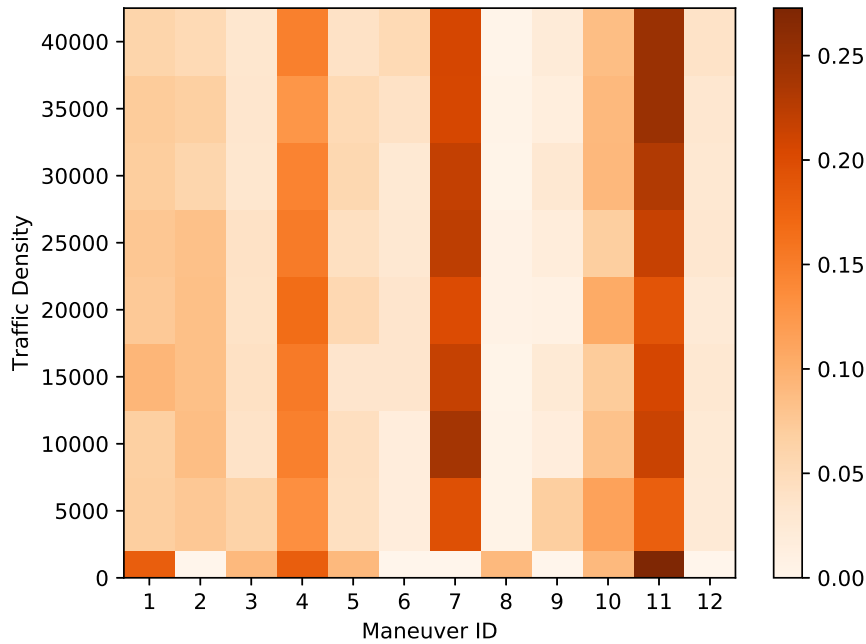
day. It has a sharp transition in probability for the traffic density range [30000, 35000], whereas the time performance metric has this transition for the range [35000, 40000]. Therefore, the critical capacity of the airspace can be defined as 30000 that is limited by the safety.



**Figure 3.10 :** Distributions for Number of Aircraft in Airspace: (a) Real Throughput, (b) Simulated Throughput for Traffic Density = 30000.

The distributions of the number of aircraft that ATCo operates in the airspace (ISTANBUL ACC) are presented in Figure 3.10. The real throughput of the airspace during a standard day is presented in Figure 3.10a, where the mean of the distribution is 10.7. In Figure 3.10b, the distribution of the throughput during simulations for the breaking point 30000 flights per day is illustrated and the mean of the simulated throughput is 143 aircraft. As a result of the analyses of the distributions, the optimization-based autonomous ATCo can manage traffic approximately 10 times denser than current traffic.

To analyze the efficiency of the chosen maneuvers, the distribution of the given maneuvers during simulations is illustrated in Figure 3.11. The traffic density almost has no impact on this distribution. The priority of ATCo is the maneuver 11, which is a course change. As mentioned before, a course change can be used to decrease the length of the trajectory while preventing collision. ATCo usually prefer this maneuver to minimize the cost function. The maneuver 10 is also a course change. However, the sign of the course change angle and the direction of the reference trajectory define the characteristic of the maneuver. It can act like a direct routing or a delaying motion. In analyzed airspace, maneuver 11 is preferred because of the directions of the reference trajectories, however, ATCo can prefer maneuver 10 to maneuver 11 in



**Figure 3.11** : Maneuver Distribution.

another airspace. Therefore, it is important to keep two different course change actions that have opposite signs in the set of maneuvers. The second most frequent maneuver is maneuver 7, which corresponds to decrease of altitude. ATCo always prefers descent to the climb because of less fuel consumption and sometimes it is preferred because of performance limitations. The third most preferred maneuver is decrease of speed as maneuver 4. Autonomous ATCo usually prefers decrease of speed to holding pattern or vector for spacing as a delaying motion because of the less impact on the cost. As seen in Figure 3.11, the least preferred actions are maneuver 8, 9 and 6. Both maneuver 8 and 9 are vector for spacing and speed change dominates them. The maneuver 6 is increase of the altitude as  $H_c = 4000ft$  and this action is dominated by other altitude change maneuvers. The least used maneuvers, especially maneuver 8, can be removed from the set of maneuvers or changed with more efficient maneuvers to improve the performance of the optimization process and increase the airspace capacity.



## 4. MISSION PLANNING AND CONTROL OF MULTI-AIRCRAFT SYSTEMS WITH SIGNAL TEMPORAL LOGIC SPECIFICATIONS

### 4.1 Purpose

We have focused on designing an alternative method that generate control inputs as continuous real valued functions instead of predefined maneuvers. It is also an optimization-based approach to control a multi-aircraft system in which each aircraft has several mission objectives. In the method, Signal Temporal Logic (STL) is used to express the mission specifications that combine temporal and logical constraints. An optimization problem is constructed in the form of Mixed-Integer Linear Programming (MILP) by using the differential flatness property of a nonlinear dynamical system and STL specifications to generate feasible trajectories. Contrary to general implementations of Temporal Logic to discrete-time systems, the proposed method deals with continuous-time systems. It can be used to find optimal control strategies to achieve the assigned tasks for nonlinear dynamical systems without discretizing the system dynamics. As an illustration, we present an air traffic control example. The nonlinear dynamical model for the aircraft is represented as a partially differentially flat system, and the presented method is applied to manage approach control and to solve the arrival sequencing problem. The method is also applied with a quadrotor fleet to show that the method can be used with different multi-agent systems.

### 4.2 System Behavior and Signal Temporal Logic

We consider the continuous-time dynamical systems of the form:

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

where  $x(t) \in \mathbb{R}^n$  is the vector of system states,  $u(t) \in \mathbb{R}^m$  is the vector of control inputs and  $x_0 \in \mathbb{R}^n$  is the initial state of the system. A state trajectory  $\mathbf{x}$  is a vector of continuous-time signals and this trajectory is derived from an action trajectory by running the system model (4.1). An action trajectory contains the control inputs for

a specific time period  $[0, T]$  and the state trajectory is generated for this finite time period.

#### 4.2.1 Signal temporal logic

The desired system behaviors can be specified using Signal Temporal Logic (STL) [44]. In this study, we use the future fragment of STL, which does not contain the since operator. The set of formulas of STL can be recursively defined by:

$$\psi ::= \top \mid \mu \mid \neg\psi \mid \psi_1 \wedge \psi_2 \mid \psi_1 \mathcal{U}_{[a,b]} \psi_2$$

where  $\psi$  is an STL formula, and  $\mu$  is an atomic predicate whose value depends on the sign of a function of  $\mathbf{x}$ .  $\top$  is the Boolean True.  $\neg$ ,  $\wedge$ , and  $\mathcal{U}$  are the negation, conjunction, and until operators, respectively. The other connectives can be defined with regard to these operators. The following identity allows to define the disjunction ( $\vee$ ) in terms of the negation and the conjunction,  $\psi_1 \vee \psi_2 = \neg(\neg\psi_1 \wedge \neg\psi_2)$ . The operators eventually ( $\diamond_{[a,b]}$ ) and always ( $\square_{[a,b]}$ ) can be defined as  $\diamond_{[a,b]}\psi = \top \mathcal{U}_{[a,b]}\psi$  and  $\square_{[a,b]}\psi = \neg\diamond_{[a,b]}\neg\psi$ , respectively. Additionally, the operators implication ( $\Rightarrow$ ) and equivalency ( $\Leftrightarrow$ ) can be presented as  $\psi_1 \Rightarrow \psi_2 = \neg\psi_1 \vee \psi_2$  and  $\psi_1 \Leftrightarrow \psi_2 = (\psi_1 \Rightarrow \psi_2) \wedge (\psi_2 \Rightarrow \psi_1)$ , respectively.

The validity of a formula  $\psi$  with respect to signal  $\mathbf{x}$  at time  $t$  is defined as follows:

$$\begin{aligned} (\mathbf{x}, t) \models \top & \text{ iff } \top \\ (\mathbf{x}, t) \models \mu & \text{ iff } \mu(\mathbf{x}(t)) \geq 0 \\ (\mathbf{x}, t) \models \neg\psi & \text{ iff } (\mathbf{x}, t) \not\models \psi \\ (\mathbf{x}, t) \models \psi_1 \wedge \psi_2 & \text{ iff } (\mathbf{x}, t) \models \psi_1 \text{ and } (\mathbf{x}, t) \models \psi_2 \\ (\mathbf{x}, t) \models \psi_1 \mathcal{U}_{[a,b]} \psi_2 & \text{ iff } \exists s \in [t+a, t+b], (\mathbf{x}, s) \models \psi_2 \\ & \text{ and } \forall s' \in [t, s], (\mathbf{x}, s') \models \psi_1 \end{aligned}$$

The trajectory  $\mathbf{x}$  satisfies the formula  $\psi$  if and only if  $(\mathbf{x}, t) \models \psi$ . Additionally, the semantics of the operators eventually and always can be given as follows:

$$\begin{aligned} (\mathbf{x}, t) \models \diamond_{[a,b]}\psi & \text{ iff } \exists s \in [t+a, t+b], (\mathbf{x}, s) \models \psi \\ (\mathbf{x}, t) \models \square_{[a,b]}\psi & \text{ iff } \forall s \in [t+a, t+b], (\mathbf{x}, s) \models \psi \end{aligned}$$



## 4.2.2 Robust STL specifications

The robust semantics of STL ([64], [65]) can be used to give the system the ability of tolerating perturbations. The robustness of STL formula  $\psi$  can be specified via a function  $\rho^\psi(\mathbf{x}, t)$  that is defined recursively as follows:

$$\begin{aligned}\rho^\mu(\mathbf{x}, t) &= \mu(\mathbf{x}(t)) \\ \rho^{-\psi}(\mathbf{x}, t) &= -\rho^\psi(\mathbf{x}, t) \\ \rho^{\psi_1 \wedge \psi_2}(\mathbf{x}, t) &= \min(\rho^{\psi_1}(\mathbf{x}, t), \rho^{\psi_2}(\mathbf{x}, t)) \\ \rho^{\psi_1 \mathcal{U}_{[a,b]} \psi_2}(\mathbf{x}, t) &= \max_{s \in [t+a, t+b]} \left( \min(\rho^{\psi_2}(\mathbf{x}, s), \right. \\ &\quad \left. \min_{s' \in [t, s]}(\rho^{\psi_1}(\mathbf{x}, s')) \right)\end{aligned}$$

For any signal  $\mathbf{x}$  and STL formula  $\psi$ ,  $\mathbf{x}$  satisfies  $\psi$  at time  $t$  if  $\rho^\psi(\mathbf{x}, t) > 0$  such that  $\rho^\psi(\mathbf{x}, t) > 0 \Rightarrow (\mathbf{x}, t) \models \psi$ . The magnitude of the  $\rho^\psi(\mathbf{x}, t)$  quantifies the robustness for the formula  $\psi$ .

## 4.3 Differential Flatness

The dynamical system (4.1) is differentially flat if there exist relations ([66], [67], [68])

$$\begin{aligned}\zeta &: \mathbb{R}^n \times (\mathbb{R}^m)^{r+1} \rightarrow \mathbb{R}^m, \\ \eta &: (\mathbb{R}^m)^r \rightarrow \mathbb{R}^n, \text{ and} \\ \kappa &: (\mathbb{R}^m)^{r+1} \rightarrow \mathbb{R}^m.\end{aligned}\tag{4.2}$$

such that

$$z = \zeta(x, u, \dot{u}, \dots, u^{(r)}),\tag{4.3}$$

$$x = \eta(z, \dot{z}, \dots, z^{(r-1)}), \text{ and}\tag{4.4}$$

$$u = \kappa(z, \dot{z}, \dots, z^{(r-1)}, z^{(r)}).\tag{4.5}$$

where  $\zeta, \eta, \kappa$  are smooth functions, and  $z$  is the flat output vector. This means that all system dynamics can be expressed as a function of the flat outputs and their derivatives. This model is equivalent to (4.1) and can be used to efficiently generate trajectories. The equations (4.4) and (4.5) yield that for every given trajectory of the flat output  $t \mapsto z(t)$ , the evolution of all other variables of the system  $t \mapsto x(t)$  and  $t \mapsto u(t)$  is also determined without integration of the system of differential equations. Moreover,

given a sufficiently smooth trajectory for the flat output  $t \mapsto z^*(t)$ , equation (4.5) can be used to generate the corresponding feedforward  $u^*$  directly.

Let us suppose that all system variables cannot be expressed as a function of the flat outputs and their derivatives. The dynamical system (4.1) is partially differentially flat if a partition of the system variables  $(x_d, u_d)$  can be expressed as in equation (4.4) and (4.5) via the set of smooth functions  $\eta_d, \kappa_d$ , while the rest of the system variables  $(x_n, u_n)$  are presented in the following form:

$$\dot{x}_n = \alpha(x_n, u_n, z, \dot{z}, \dots, z^{(r-1)}), \text{ and} \quad (4.6)$$

$$u_n = \beta(x_n, z, \dot{z}, \dots, z^{(r-1)}, z^{(r)}). \quad (4.7)$$

In this case, it is also possible to generate trajectories for the system variables  $(x_d, u_d)$  such that  $t \mapsto x_d(t)$  and  $t \mapsto u_d(t)$  from a given trajectory  $t \mapsto z(t)$  without numerical integration. However, it is necessary to integrate the equation (4.6) to generate the trajectories for the system variables  $(x_n, u_n)$  such that  $t \mapsto x_n(t)$  and  $t \mapsto u_n(t)$ .

#### 4.4 Motion Planning for Flat Systems

The motion planning problem corresponds to finding a trajectory  $t \mapsto (x(t), u(t))$  from a set of specific initial conditions to a defined final state while satisfying the system dynamics  $\dot{x} = f(x, u)$ . If some STL specifications are added as constraints on the trajectory, the problem is transformed into a motion planning with constraints. In the general case, this problem can be quite difficult because it requires the integration of the system equations to find the sequence of control inputs that satisfies the initial conditions, final conditions and constraints. For nonlinear systems, it may pose some additional problems [69]. The trajectory generation is particularly easy for the differentially flat systems.

Let us define each element of the flat output  $z$  as a linear combination of certain basis functions of the time, i.e.,

$$z_i = \sum_{j=1}^{N_i} c_{ji} \phi_{ji}(t) \quad (4.8)$$

where,  $c_{ji}$  is a weighting coefficient or control point,  $\phi_{ji}(t)$  is a basis function and  $z = [z_1, z_2, \dots, z_s]^T$ .

There are several candidates that can be used as basis functions. In this study, B-spline basis functions are used to present the flat outputs. Let  $p$  be a nonnegative integer and let  $\mathbb{T} = \{\tau_0, \tau_1, \dots, \tau_m\}$ , the knot vector, be a nondecreasing sequence of real numbers. The  $q^{th}$  B-spline basis function of  $p$ -degree, denoted by  $N_{q,p}(t)$ , is defined as [70]:

$$N_{q,0}(t) = \begin{cases} 1 & \text{if } \tau_q \leq t < \tau_{q+1} \\ 0 & \text{otherwise} \end{cases} \quad (4.9)$$

$$N_{q,p}(t) = \frac{t - \tau_q}{\tau_{q+p} - \tau_q} N_{q,p-1}(t) + \frac{\tau_{q+p+1} - t}{\tau_{q+p+1} - \tau_{q+1}} N_{q+1,p-1}(t) \quad (4.10)$$

The B-spline functions yield some geometric properties [70] that affect the performance of the trajectory generation positively when dealing with the dynamical systems. For example, a B-spline curve such as  $z_i$  in (4.8) is  $C^{p-1}$  in any  $t \in \mathbb{T}$ , and  $C^\infty$  otherwise. It is a continuous and differentiable function. The higher order derivatives of B-spline basis functions can be presented as linear combinations of B-splines of lower order, and the weighting coefficients (or control points) have linear impact on the higher order derivatives of the curve such that  $z_i^{(k)} = \sum_{j=1}^{N_i} c_{ji} \phi_{ji}^{(k)}(t)$ , where  $\phi_{ji}^{(k)}(t)$  is  $k$ -order derivative of basis function. Moreover, the B-spline functions satisfy that  $N_{q,p}(t) \geq 0$  and  $\sum_{q=0}^n N_{q,p}(t) = 1, \forall t \in [\tau_0, \tau_m]$ , and they have local support such that  $N_{q,p}(t) \neq 0$  iff  $t \in [\tau_q, \tau_{q+p}]$ . Hence, the adjustment of a specific weighting coefficient leads to change shape of a specific region of the curve without affecting the rest of it.

Let  $C$  be the set of the control points that configure the flat output for the dynamical system (4.1). Equation (4.8) and a given  $C$  construct the trajectory of the flat output  $t \mapsto z^C(t)$ . The trajectory  $t \mapsto z^C(t)$  can be modified by changing the coefficients in  $C$ , where each coefficient is a control point that shapes a specific region of the trajectory. However, some coefficient sets can generate infeasible trajectories because of the violation of the dynamical constraints or STL formulas. Using the analytical expressions in (4.4) and (4.5), the trajectories of the system variables  $t \mapsto x^C(t)$  and  $t \mapsto u^C(t)$  can be expressed in terms of  $C$ . Then, the violation of any constraint can be checked via these system trajectories. By tuning the coefficients, these violations can be removed. Because of the local support property, each coefficient has impact on a specific region of the trajectory. Therefore, the violations can be removed by changing only some specific coefficients without modifying all of them. The tuning process can be achieved via an algorithm or an optimization problem. In this study, we construct an optimization problem to find the proper control points that generate

feasible trajectories with STL specifications. When dealing with partially differentially flat systems, we avoid presenting the dynamical constraints and STL specifications in terms of the system variables  $(x_n, u_n)$  during the trajectory planning to prevent the costs that originate from integrating the differential equations. All constraints and STL specifications are expressed with respect to the system variables  $(x_d, u_d)$  that form the differentially flat part of the system. However, the system variables  $(x_n, u_n)$  are generated from the flat output's trajectory  $t \mapsto z^C(t)$  after the set  $C$  is determined.

#### 4.5 Flat Description of Aircraft Dynamics for Civil Aviation

In this study, we use the following set of equations as the dynamical system (4.1) to model the aircraft dynamics that was used in [31], [32], [71] for air traffic control applications.

$$\dot{x}_1 = x_4 \cos(x_5) \cos(u_3) \quad (4.11)$$

$$\dot{x}_2 = x_4 \sin(x_5) \cos(u_3) \quad (4.12)$$

$$\dot{x}_3 = x_4 \sin(u_3) \quad (4.13)$$

$$\dot{x}_4 = -\frac{C_D S \rho x_4^2}{2x_6} - g \sin(u_3) + \frac{u_1}{x_6} \quad (4.14)$$

$$\dot{x}_5 = \frac{C_L S \rho x_4 \sin(u_2)}{2x_6 \cos(u_3)} \quad (4.15)$$

$$\dot{x}_6 = -F \quad (4.16)$$

The model is a nonlinear dynamical system, where the control inputs are the engine thrust ( $u_1$ ), bank angle ( $u_2$ ), and flight path angle ( $u_3$ ), and the state variables are the horizontal position ( $x_1$  and  $x_2$ ), altitude ( $x_3$ ), true airspeed ( $x_4$ ), heading angle ( $x_5$ ), and the mass of the aircraft ( $x_6$ ). In the equation set above, aerodynamic lift and drag coefficients are denoted by  $C_L$  and  $C_D$ , gravitational acceleration is  $g$ , total wing surface area is  $S$ , air density is indicated as  $\rho$ , and the fuel consumption is indicated as  $F$ . These coefficients and other parameters such as bounds on the speed, flight path angle and mass are obtained from the Base of Aircraft Data (BADA) [59].

The lift coefficient and drag coefficient are expressed as follows:

$$C_L = \frac{2mg \cos(\gamma)}{\rho V^2 S \cos(\phi)} \quad (4.17)$$

$$C_D = C_{D0} + C_{D2} C_L^2 \quad (4.18)$$

where  $C_{D0}$  and  $C_{D2}$  are constants specified in the database.  $\gamma, \phi$  and  $V$  are the flight path angle, bank angle and speed, respectively.

The dynamical system (4.11)-(4.16) is partially differentially flat, when the flat output  $z$  is expressed as follows:

$$z = [x_1, x_2, x_3]^T \quad (4.19)$$

The differentially flat part of the system consists of the system variables  $x_1, x_2, x_3, x_4, x_5, u_2$  and  $u_3$ . Let us give the flat descriptions of these variables. The first three variables directly correspond to the flat outputs. Using Equation (4.11), (4.12) and (4.13), the fourth variable  $x_4$  can be expressed as the Euclidean norm of the derivatives of the flat outputs as follows:

$$x_4 = \sqrt{\dot{z}_1^2 + \dot{z}_2^2 + \dot{z}_3^2} \quad (4.20)$$

The variable  $x_5$  can be described in terms of the derivatives of the first two flat outputs by dividing Equation (4.12) to Equation (4.11), as:

$$x_5 = \arctan\left(\frac{\dot{z}_2}{\dot{z}_1}\right) \quad (4.21)$$

By modifying Equation (4.13), the variable  $u_3$  can be given by:

$$u_3 = \arcsin\left(\frac{\dot{z}_3}{\sqrt{\dot{z}_1^2 + \dot{z}_2^2 + \dot{z}_3^2}}\right) \quad (4.22)$$

The variable  $\dot{x}_5$  can be obtained from Equation (4.21) and the lift coefficient  $C_L$  is presented in Equation (4.17). By writing the expressions for  $\dot{x}_5$  and  $C_L$  into Equation (4.15), the variable  $u_2$  can be formulated as follows:

$$u_2 = \arctan\left(\frac{(\ddot{z}_2\dot{z}_1 - \dot{z}_2\ddot{z}_1)\sqrt{\dot{z}_1^2 + \dot{z}_2^2 + \dot{z}_3^2}}{g(\dot{z}_1^2 + \dot{z}_2^2)}\right) \quad (4.23)$$

The rest of the system variables  $u_1, x_6$  cannot be expressed only in terms of the flat outputs and their derivatives. These variables can be presented in the form of (4.6) and (4.7). The acceleration  $\dot{x}_4$  can be derived from Equation (4.20) and the variables  $x_4, u_3$  are presented before. By putting these expressions into Equation (4.14), the engine thrust  $u_1$  can be given as follows:

$$u_1 = x_6\left(\frac{\dot{z}_1\ddot{z}_1 + \dot{z}_2\ddot{z}_2 + \dot{z}_3(g + \ddot{z}_3)}{\sqrt{\dot{z}_1^2 + \dot{z}_2^2 + \dot{z}_3^2}}\right) + 0.5C_{DS}(\dot{z}_1^2 + \dot{z}_2^2 + \dot{z}_3^2). \quad (4.24)$$

The value of  $u_1$  depends on the variable  $x_6$ . To calculate  $x_6$ , the fuel consumption  $F$  must be expressed. In BADA, there are different functions that are used to determine the fuel consumption, and these functions are presented with regard to the aircraft type and flight phase. For a jet aircraft in the descent phase, the fuel consumption is as follows:

$$F = C_{f3} \left(1 - \frac{H_p}{C_{f4}}\right) \quad (4.25)$$

where  $C_{f3}$  and  $C_{f4}$  are constants, and  $H_p$  is geopotential pressure altitude. Using this expression, the derivative of the mass can be presented as a function of the altitude:

$$\dot{x}_6 = \alpha_{11}(z_3) \quad (4.26)$$

A function of the altitude  $z_3$ , speed  $x_4$  and thrust  $u_1$  can be used to generalize the calculation of the fuel consumption. Then, the following equation covers the operation in any flight phase:

$$\dot{x}_6 = \alpha_{12}(z_3, \dot{z}_1, \dot{z}_2, \dot{z}_3, u_1) \quad (4.27)$$

Both Equation (4.26) and Equation (4.27) are in the form of (4.6). Therefore, the dynamical system (4.11)-(4.16) is a partially differentially flat system, where  $x_d = [x_1, x_2, x_3, x_4, x_5]$ ,  $u_d = [u_2, u_3]$ ,  $x_n = x_6$ ,  $u_n = u_1$  with the flat descriptions (4.19)-(4.27).

#### 4.6 Differential Flatness-Based Optimal Strategy Generation for Multi-Aircraft Systems

This section presents the optimization problem that is used to generate the feasible trajectories with STL specifications to control multiple aircraft.

Let  $A$  be the set of aircraft, whose dynamics are presented with the flat descriptions (4.19)-(4.27), and each flat output is expressed as in Equation (4.8). Let  $C^A$  be the set that contains all control points for all aircraft in  $A$ . Then, by specifying the control points in  $C^A$  as decision variables, the following optimization problem can be formulated to generate the optimum trajectories and control inputs that ensure the given STL formulas:

$$\max_{C^A} \rho^\Psi(z) \quad (4.28)$$

subject to

$$d(z) \geq 0 \quad (4.29)$$

$$b(z) = 0 \quad (4.30)$$

$$\mu_\psi(z) \geq 0 \quad (4.31)$$

where  $\rho^\psi(z)$  symbolizes the robust specification of the STL formula,  $\mu_\psi(z)$  symbolizes the vector of the boolean specifications that construct the formula  $\psi$ , the vector  $d(z)$  denotes the dynamical constraints, and the vector  $b(z)$  denotes the initial conditions or other equality constraints that are presented in the flat space. The solution of this optimization problem corresponds to the values of the control points for all aircraft. After obtaining the control points, the state trajectories and control inputs are generated in continuous-time by using the flat descriptions (4.19)-(4.27).

This optimization problem can be expressed as a non-convex optimization problem or a mixed-integer linear programming (MILP). In this study, we focus on the second case and present the objective and all constraints as linear expressions.

#### 4.6.1 Performance limits and initial conditions

Any performance limit can be presented in the constraint set (4.29). Let us consider the restriction of the aircraft speed. By using the expression (4.20), the aircraft speed can be bounded with the following constraints:

$$V_{min}^2 \leq \dot{z}_1^2 + \dot{z}_2^2 + \dot{z}_3^2 \leq V_{max}^2 \quad (4.32)$$

where  $V_{min}$  and  $V_{max}$  are the bounds for the speed. However, this expression is nonlinear. In a non-convex optimization problem, this expression can be directly used, whereas it should be presented as linear constraints in MILP formulation. Let us evaluate the vertical and horizontal speeds separately. The rate of climb/descent  $\dot{z}_3$  can be directly bounded such that  $V_{min}^v \leq \dot{z}_3 \leq V_{max}^v$ . To bound the horizontal speed, we approximate the Euclidean norm  $\|[\dot{z}_1 \ \dot{z}_2]^T\|$  by the edges of an N-sided polygon that

can be captured by the following inequalities [72]:

$$\begin{aligned}
\dot{z}_1 \sin\left(\frac{2\pi n}{N}\right) + \dot{z}_2 \cos\left(\frac{2\pi n}{N}\right) &\leq V_{max}^h, \quad n = 1, \dots, N \\
\dot{z}_1 \sin\left(\frac{2\pi n}{N}\right) + \dot{z}_2 \cos\left(\frac{2\pi n}{N}\right) &\geq V_{min}^h - Ma_n \\
\sum_{n=1}^N a_n &\leq N - 1 \\
a_n &\in \{0, 1\}
\end{aligned} \tag{4.33}$$

where  $V_{min}^h$  and  $V_{max}^h$  are the bounds for the horizontal speed, and  $M$  is a large enough number. The other performance variables can also be restricted. To bound the acceleration, the N-sided polygon approach can also be used. In the optimization problem, this kind of constraint is enforced at the predefined points that are chosen uniformly over the time interval  $[t_0, t_f]$ .

The constraint set (4.30) consists of the initial conditions. Let  $[x_0, y_0, h_0]^T$  be the initial position of the aircraft at  $t_0$ . Then, the initial position is defined in the constraints as follows:

$$z_1(t_0) = x_0, \text{ and } z_2(t_0) = y_0, \text{ and } z_3(t_0) = h_0 \tag{4.34}$$

The other system variables can also be assigned to a certain value at a specific time. Let us focus on the heading angle that is presented in the equation (4.21). Let  $\theta$  be the assigned value for the heading angle  $x_5$  at time  $t_s$  such that  $x_5(t_s) = \theta$ . By using the expression (4.21), this condition can be presented as follows:

$$\begin{aligned}
\dot{z}_2(t_s) - \dot{z}_1(t_s) \tan(\theta) &= 0, \text{ and} \\
\dot{z}_1(t_s) &\leq 0 \text{ if } \pi/2 \leq \theta < 3\pi/2 \\
\dot{z}_1(t_s) &\geq 0 \text{ otherwise}
\end{aligned} \tag{4.35}$$

When it is necessary to specify the values of the several system variables, the combination of them can be used to form linear constraints. Let us consider that the initial speed, initial heading angle and initial path angle are known, and we want to set  $x_4, x_5, u_2$  to these specific values at time  $t_0$ . Instead of setting them separately via equations (4.20), (4.21), (4.23), these initial values are used to calculate  $\dot{x}_0, \dot{y}_0, \dot{h}_0$  via equations (4.11), (4.12), (4.13), and these derivatives are assigned as linear constraints:

$$\dot{z}_1(t_0) = \dot{x}_0, \text{ and } \dot{z}_2(t_0) = \dot{y}_0, \text{ and } \dot{z}_3(t_0) = \dot{h}_0 \tag{4.36}$$



## 4.6.2 STL operators

The tasks of the aircraft such as reaching particular regions, avoiding obstacles, ensuring appropriate separations can be described by the help of the convex hulls, where the faces of a convex hull are defined in terms of the affine expressions. These tasks can be evaluated as lying inside or outside the convex hulls. For example, obstacle avoidance refers to staying outside of the corresponding convex hull or lying in at least one of the outer halfspaces determined by the faces of the convex hull that can be formulated via disjunction operator. In like manner, the conjunction operator can be used to enforce the arriving a particular region. These missions can be specified with STL formalism. The STL operators should be described as MILP constraints to present these missions for the optimization problem.

Let  $\mu$  be a predicate such that it holds at time  $t$  if and only if  $\mu(\mathbf{x}, t) \geq 0$ . When the predicate  $\mu$  is an affine expression, this condition of the STL formulas can be directly described as a MILP constraint:

$$\mu(\mathbf{x}, t) \geq 0 \quad (4.37)$$

The other connectives in the STL formulas can also be described as MILP constraints. The negation of the predicate  $\mu$  at time  $t$  can be presented as  $-\mu(\mathbf{x}, t) \geq 0$ . The conjunction  $\bigwedge_{i=1}^k \mu^i(\mathbf{x}, t_i)$  is enforced with the following constraints:

$$\mu^i(\mathbf{x}, t_i) \geq 0, \quad i = 1, \dots, k \quad (4.38)$$

The disjunction  $\bigvee_{i=1}^k \mu^i(\mathbf{x}, t_i)$  can also be ensured as follows:

$$\mu^i(\mathbf{x}, t_i) \geq -Mb_i, \quad i = 1, \dots, k \quad (4.39)$$

$$\sum_{i=1}^k b_i \leq k - 1 \quad (4.40)$$

where  $M$  is a large enough number and  $b_i$ 's are binary variables.

The operators eventually  $\diamond_{[\alpha, \beta]} \mu$  and always  $\square_{[\alpha, \beta]} \mu$  can be described in terms of the conjunction and disjunction operators. The operator eventually can be presented as  $\bigvee_{\tau=\alpha}^{\beta} \mu(\mathbf{x}, \tau)$ , and MILP formulation of the disjunction operator is given in (4.39)-(4.40). In a similar manner, the operator always can be described as  $\bigwedge_{\tau=\alpha}^{\beta} \mu(\mathbf{x}, \tau)$ .

The robust definition of STL contains the min and max operators. Let us focus on the MILP formulation of conjunction operator in the robust setting such that  $\rho^\Psi = \bigwedge_{i=1}^k \rho^{\Psi_i}(\mathbf{x}, t_i)$ . The following set of constraints can be used to obtain the robust conjunction:

$$\rho^\Psi \leq \rho^{\Psi_i}(\mathbf{x}, t_i), \quad i = 1, \dots, k \quad (4.41)$$

$$M(b_{t_i}^{\Psi_i} - 1) \leq \rho^\Psi - \rho^{\Psi_i}(\mathbf{x}, t_i) \leq M(1 - b_{t_i}^{\Psi_i}) \quad (4.42)$$

$$\sum_{i=1}^k b_{t_i}^{\Psi_i} = 1 \quad (4.43)$$

$$b_{t_i}^{\Psi_i} \in \{0, 1\}, \quad i = 1, \dots, k \quad (4.44)$$

where the variables  $b_{t_i}^{\Psi_i}$  are auxiliary binary variables that are used to enforce that  $\rho^\Psi = \min_i(\rho^{\Psi_i}(\mathbf{x}, t_i))$ . The robust disjunction can also be presented in a similar manner. By replacing the first inequality with  $\rho^\Psi \geq \rho^{\Psi_i}(\mathbf{x}, t_i)$ , the set of constraints enforces that  $\rho^\Psi = \max_i(\rho^{\Psi_i}(\mathbf{x}, t_i))$ . The negation of  $\rho^\Psi(\mathbf{x}, t)$  was also presented as  $-\rho^\Psi(\mathbf{x}, t)$ . Then, the rest of the operators can be obtained from these three operators.

### 4.6.3 Continuous-time satisfiability

It is stated in [64] that satisfying an STL formula for a sampled trajectory does not imply continuous-time satisfiability unless the formula is strictified. In order to guarantee the continuous-time satisfiability, a given formula  $\phi$  is strengthened via a function such that  $\mathbf{str} : \phi \rightarrow \phi_s$ . Firstly, an appropriate sampling period  $\Delta t$  must be chosen that satisfy a set of predefined conditions and guarantee the existence of at least one sampling point within each timing interval of the temporal operators. Secondly, the trajectory must have conservative bounds between two consecutive samples which can be satisfied as  $\|\mathbf{x}(t) - \mathbf{x}(t + \Delta t)\| \leq \mathcal{E}\Delta t$ , where  $\mathcal{E} \geq 0$ . Note that B-spline curves ensure this property. Then, the following relation holds [64, Theorem 5.3.1]:  $\rho^{\phi_s}(\mathbf{x}^{samp}, t) > \mathcal{E} \Rightarrow \rho^\phi(\mathbf{x}, t) \models \top$ , where  $\mathbf{x}^{samp}$  is the sampled trajectory and  $\mathbf{x}$  is the continuous-time trajectory. More detailed information can be found in Ch. 5 of the study [64]. For an appropriate sampling period, the value of  $\mathcal{E}$  can be calculated when the performance limits are given, and this value can be used in optimization problem as a buffer. In this way, the continuous-time satisfiability of an STL formula can be guaranteed, although the conditions are enforced for the sampled times. This strategy

is also valid for the performance constraints in the optimization problem such as speed or acceleration.

## 4.7 Illustrative Examples

In this section, we present the experimental results to evaluate the performance of the proposed method and show the validity of the method for air traffic control and UAV applications. In the applications, the air vehicles are controlled by a centralized mechanism. In the ATC examples, we use the performance parameters of Boeing 737-800 for all aircraft, and we specify the minimum horizontal separation as 3nm, which is used in the real approach control operations. In the last case study, we focus on the control of a UAV fleet and specify the minimum separation as 5m. We construct the MILP problems in Python programming language using the PuLP library [73] and generate the solutions using the solver Gurobi<sup>1</sup>. All optimization problems are solved on a laptop with an Intel i7 processor and 16GB RAM.

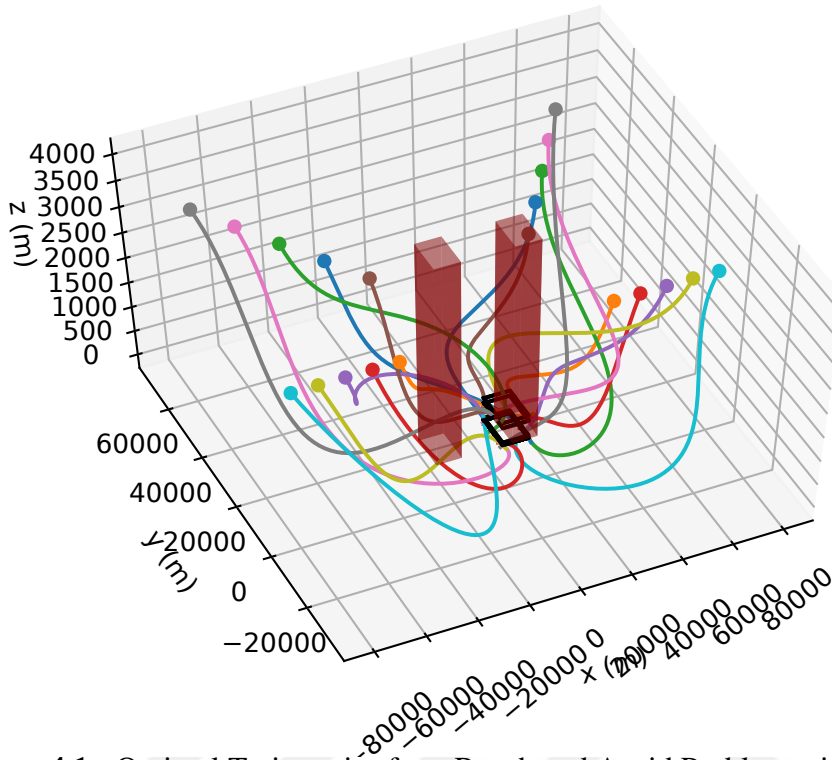
### 4.7.1 Reach and avoid problem

Firstly, we analyse the simulation results for the reach-avoid problem. In this problem, all of the aircraft try to reach a target region, while avoiding conflicts with the other aircraft and obstacles. We consider this problem as an ATC application and use the performance parameters of Boeing 737-800. An example scenario for the control of 20 aircraft is presented in Figure 4.1. The computation times for the solutions of the reach-avoid problems with different number of aircraft are presented in Table 4.1. The solutions are generated through two different modes. The Boolean mode corresponds to solving the optimization problem to generate trajectories that satisfy STL specifications without maximizing robustness, whereas the Robust mode refers to the maximization of the robustness of the STL specifications. It is observed that the optimization problem is always solved faster in the Boolean mode than the Robust mode. As presented in Table 4.1, the performance of the designed method is computationally tractable for the control of multiple aircraft. Although the problem cannot be solved in the Robust mode for 20 aircraft within feasible time limit, the

---

<sup>1</sup><http://www.gurobi.com/>

solution is generated efficiently in the Boolean mode. The method can be used for the real-time applications.



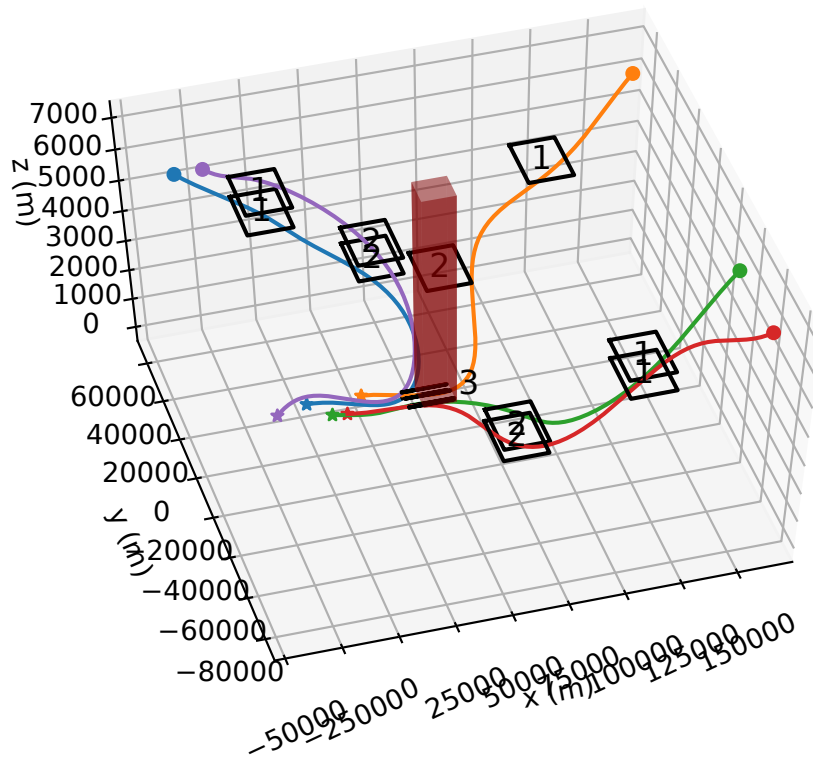
**Figure 4.1** : Optimal Trajectories for a Reach and Avoid Problem with 20 Aircraft.

**Table 4.1** : Performance Evaluation.

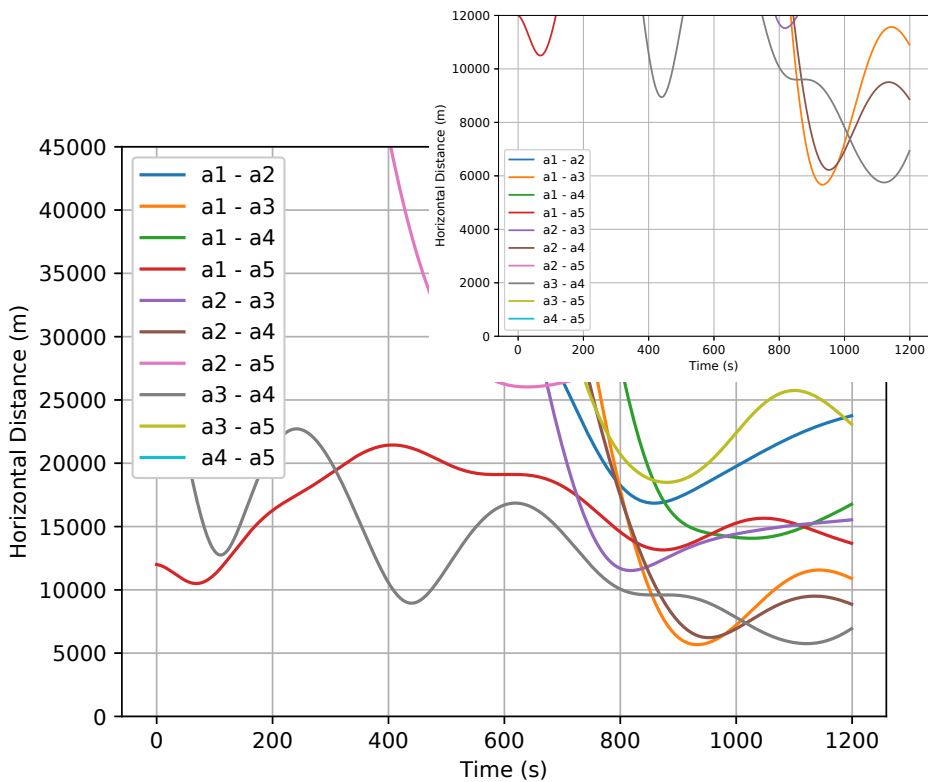
# aircraft	Boolean Mode		Robust Mode	
	Construction Time (s)	Solution Time (s)	Construction Time (s)	Solution Time (s)
2	$0.33 \pm 0.12$	$0.18 \pm 0.06$	$0.35 \pm 0.14$	$0.21 \pm 0.08$
5	$0.97 \pm 0.11$	$0.65 \pm 0.03$	$1.01 \pm 0.37$	$0.84 \pm 0.27$
10	$2.41 \pm 0.27$	$1.17 \pm 0.34$	$3.34 \pm 0.61$	$3.57 \pm 0.52$
20	$7.36 \pm 0.91$	$3.40 \pm 0.63$	$7.96 \pm 1.12$	-

#### 4.7.2 Approach control and arrival sequencing

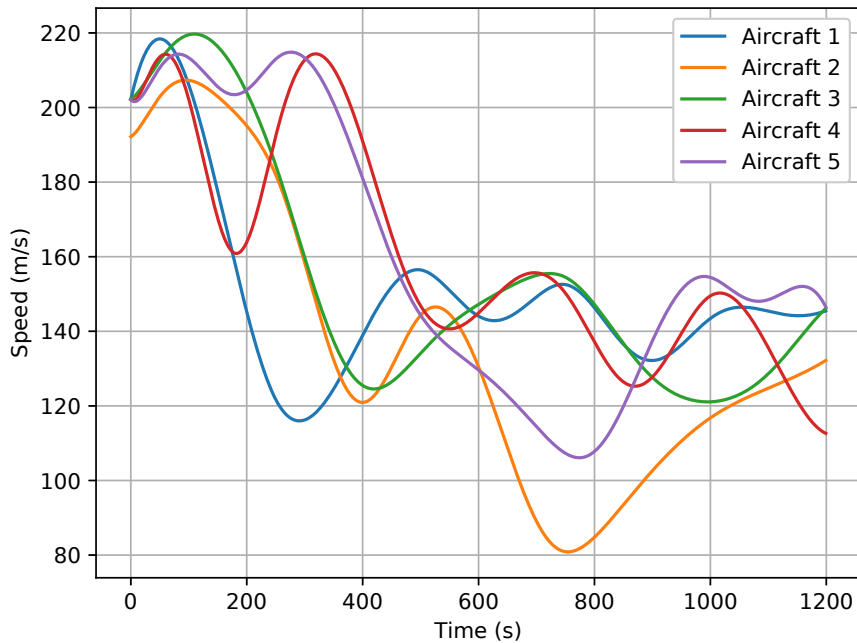
The second case is the application of the method to a realistic approach control scenario. This scenario contains the control of multiple aircraft in arrival traffic and sequencing them at 1000m. In this case study, all aircraft have the same performance limits except second aircraft (orange). The speed limits and initial speed of the second aircraft are  $10m/s$  lower than the others to show the capability of the proposed method dealing with heterogeneous aircraft performance. In the scenario, each aircraft should visit three predefined regions at the specified times that refer to a standard terminal



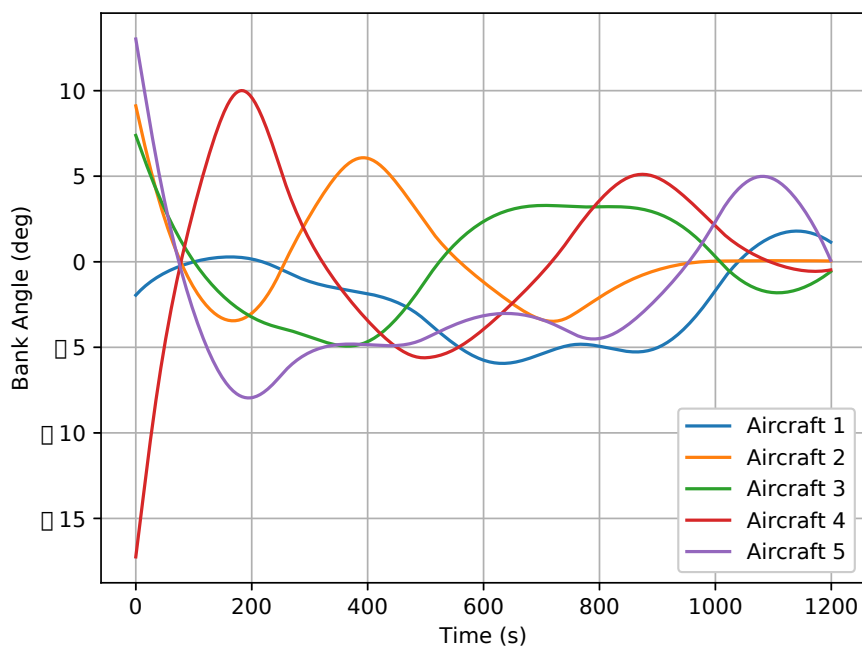
**Figure 4.2 :** Optimal Trajectories for Approach Control and Arrival Sequencing.



**Figure 4.3 :** Solution to Approach Control and Arrival Sequencing: horizontal distances between aircraft pairs.



**Figure 4.4** : Solution to Approach Control and Arrival Sequencing: speeds of each aircraft stay within speed bounds.

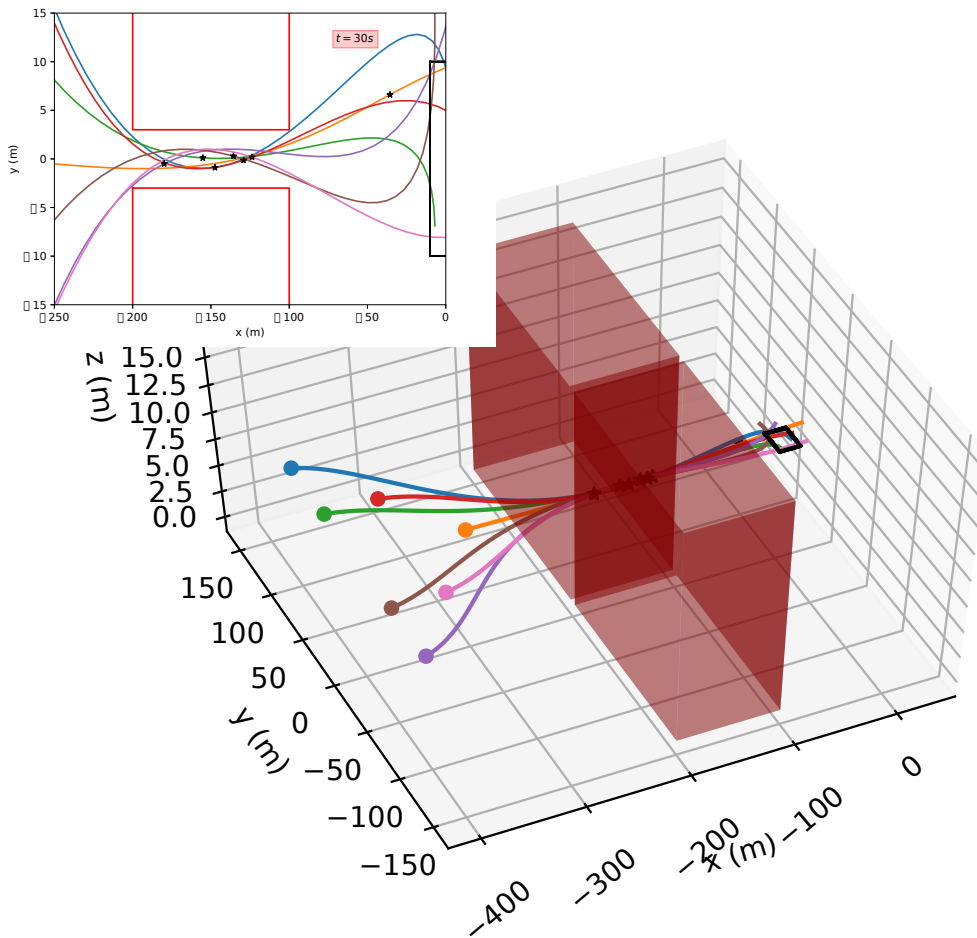


**Figure 4.5** : Solution to Approach Control and Arrival Sequencing: bank angles of each aircraft stay within bank angle limits.

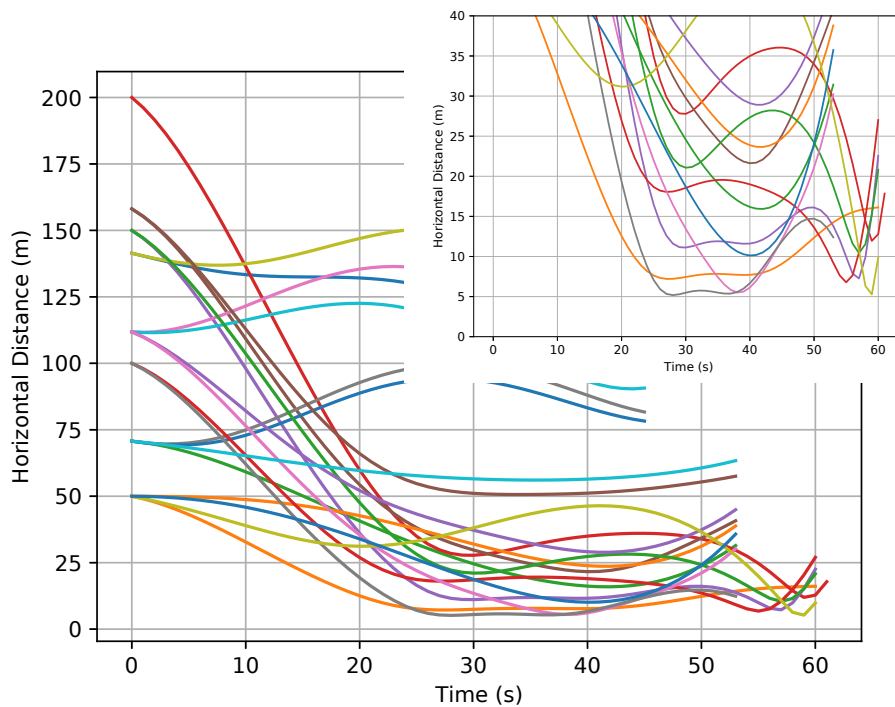
arrival (STAR) procedure in the real operation. After reaching the last region, they should arrange their headings according to the runway's direction. In this phase, the aircraft are sequenced while the minimum separation requirements are satisfied and their headings are arranged according to the runway's direction. The simulation results for the control of five aircraft are presented in Figure 4.2. The observation is that the aircraft visit the predefined regions while avoiding obstacles and each other, and obtain the necessary heading angles according to the runway's direction after reaching the last region. Furthermore, they are sequenced with minimum horizontal separation requirement. The results show that all aircraft always ensure the minimum separation requirement (3nm/5556meters) as illustrated in Figure 4.3. The generated trajectories are also feasible in terms of performance limitations. For example, the speeds are always within limits during operation as shown in Figure 4.4. The speed of the second aircraft is also always within its limits, and its speed is often lower than the other aircraft. This case study also shows that the proposed method can handle heterogeneous aircraft performance during approach control. As mentioned before, the control inputs are also generated at the end of the optimization. The bank angles are illustrated as an example in Figure 4.5. It is shown that the bank angles are also within limits, which are  $[-25^\circ, 25^\circ]$  during take-off and landing and  $[-45^\circ, 45^\circ]$  during other flight phases for civil flight [60], and they are tractable by a Boeing 737-800 because of the soft angle changes. Moreover, the time performance of the method is also practicable. For this case study, the construction time of the MILP is approximately 1.8s, whereas the solution time is around 3.9s.

### 4.7.3 UAV fleet narrow-passage problem

In the last example, we solve a narrow-passage problem for a quadrotor fleet, where the quadrotors aim to reach a delivery point by passing a narrow-passage. We aim to show that the method can also be used in UAV applications. The flat descriptions of a quadrotor are presented in [74]. The problem is solved for a fleet that has 7 quadrotors as presented in Figure 4.6, which also contains the positions of the quadrotors while passing the passage at time  $t = 30s$ . As presented in Figure 4.7, the quadrotor pairs always ensure the minimum separation requirement, which is 5m, during operation. These figures show that the fleet satisfies the mission specifications. The system

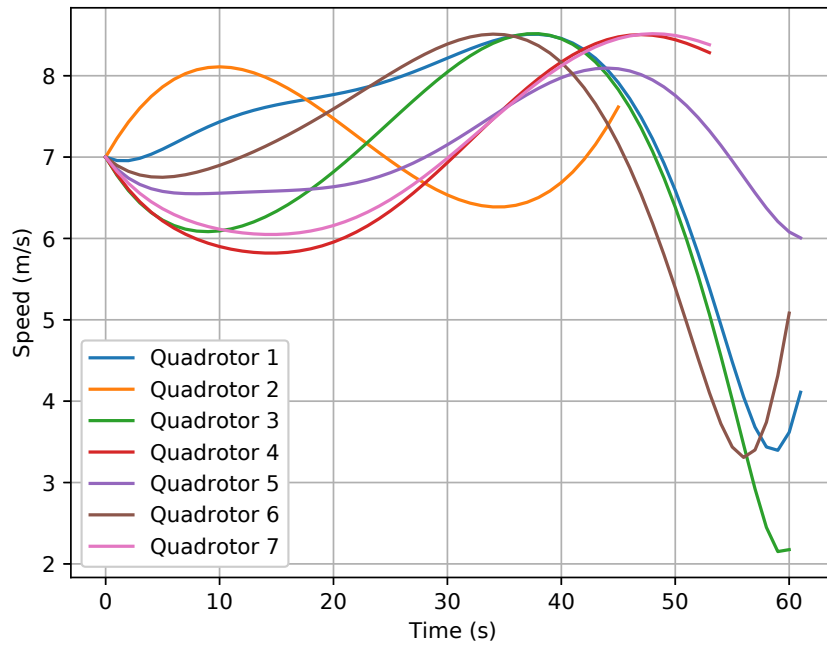


**Figure 4.6 :** Optimal Trajectories for the Quadrotor Fleet for the Narrow-Passage Problem.



**Figure 4.7 :** Solution to UAV Fleet Narrow-Passage Problem: horizontal distances between quadrotor pairs.





**Figure 4.8** : Solution to UAV Fleet Narrow-Passage Problem: speeds of each quadrotor stay within speed bounds.

variables are also within limits during operation. For example, the speeds of the UAVs are smaller than the maximum speed 8.5m/s as presented in Figure 4.8.



## **5. FORMULATION OF AERIAL COMBAT GAME VIA DIFFERENTIAL FLATNESS-BASED RECEDING HORIZON CONTROL**

### **5.1 Purpose**

This chapter focuses on modeling of air-to-air combat via flatness-based control and game theoretic approach. The flatness property of the dynamical system is used to present aircraft dynamics in terms of specific variables and their derivatives that can be described via a set of parameterized curves. By the help of game theory, the combat between two aircraft is translated into an optimization problem in terms of these parameterized curves. The optimization problem is solved with a moving time horizon scheme to generate optimal strategies for aircraft in the aerial combat. In this way, the algorithm produces optimal feasible strategies that meet all the given and dynamical constraints. The method is demonstrated with aerial combat between two UAVs. Two different case studies are analyzed to show and validate the method.

### **5.2 Flat Description of Aircraft Dynamics for Military Aviation**

This section contains the aircraft model for military aviation, and the presentation of this model as a differentially flat system. Differential flatness is a structural property of a class of nonlinear dynamical systems, denoting that all system variables can be written in terms of a set of specific variables and their derivatives. By using this property, aircraft dynamics can be evaluated via a set of algebraic equations without numerical integration.

The motion of aircraft can be expressed by the following set of differential equations:

$$\dot{x} = V \cos(\psi) \cos(\gamma) \quad (5.1)$$

$$\dot{y} = V \sin(\psi) \cos(\gamma) \quad (5.2)$$

$$\begin{aligned} \dot{h} &= V \sin(\gamma) \\ \dot{V} &= \frac{T \cos(\alpha) - D}{m} - g \sin(\gamma) \end{aligned} \quad (5.3)$$

$$\dot{\psi} = \frac{L + T \sin(\alpha)}{mV} \frac{\sin(\mu)}{\cos(\gamma)} \quad (5.4)$$

$$\dot{\gamma} = \frac{L + T \sin(\alpha)}{mV} \cos(\mu) - \frac{g \cos(\gamma)}{V} \quad (5.5)$$

where  $x$ ,  $y$  and  $h$  define the position of aircraft. The rest of the state variables are the speed  $V$ , the heading angle  $\psi$  and the flight path angle  $\gamma$ . The control variables are defined as the bank angle  $\mu$ , the angle of attack  $\alpha$  and the thrust  $T$ .  $L$  and  $D$  refer to lift and drag force, respectively. The mass of aircraft is symbolized as  $m$ . The gravity is presented as  $g$ .

The longitudinal load factor  $n_x$  and normal load factor  $n_z$  can be described as follows:

$$n_x = \frac{T \cos(\alpha) - D}{mg}; \quad n_z = \frac{L + T \sin(\alpha)}{mg} \quad (5.6)$$

The equations of aircraft's motion can be reorganized by using load factors as follows:

$$\dot{x} = V \cos(\psi) \cos(\gamma) \quad (5.7)$$

$$\dot{y} = V \sin(\psi) \cos(\gamma) \quad (5.8)$$

$$\dot{h} = V \sin(\gamma) \quad (5.9)$$

$$\dot{V} = g(n_x - \sin(\gamma)) \quad (5.10)$$

$$\dot{\psi} = \frac{gn_z \sin(\mu)}{V \cos(\gamma)} \quad (5.11)$$

$$\dot{\gamma} = \frac{g}{V}(n_z \cos(\mu) - \cos(\gamma)) \quad (5.12)$$

where the longitudinal load factor  $n_x$ , the normal load factor  $n_z$  and the bank angle  $\mu$  are the control inputs. We will use the set of differential equations (5.7)-(5.12) as aircraft model to represent the motion of a combatant during aerial combat.

A system  $\dot{x} = f(x, u)$  with state vector  $x \in \mathbb{R}^n$ , input vector  $u \in \mathbb{R}^m$  where  $f$  is a smooth vector field, is differentially flat if there exists a vector  $z \in \mathbb{R}^m$  in the form [66], [67]:

$$z = \eta(x, u, \dot{u}, \dots, u^{(r)}) \quad (5.13)$$

such that

$$\begin{aligned}x &= \alpha(z, \dot{z}, \dots, z^{(q)}) \\u &= \beta(z, \dot{z}, \dots, z^{(q)})\end{aligned}\quad (5.14)$$

where  $\eta, \alpha, \beta$  are smooth functions. The new system is expressed via the  $m$  algebraic variables  $z_i, i = 1, 2, \dots, m$  that correspond to flat outputs. As shown in the equation (5.14), all of the variables in the differentially flat system can be described as function of the flat outputs and their derivatives. In the study of differential flatness, the basic question is whether there exists a function  $z = \eta(x, u, \dot{u}, \dots, u^{(r)})$  for the system  $\dot{x} = f(x, u)$  such that the state and control input vectors can be expressed in terms of  $z$  and its derivatives as in Eq.(5.14).

Let us consider the aircraft model in (5.7)-(5.12). This system is differentially flat, if flat outputs are chosen as  $x, y$  and  $h$ :

$$z = [z_1 \ z_2 \ z_3]^T = [x \ y \ h]^T \quad (5.15)$$

The other state variables  $V, \psi$  and  $\gamma$  can be described in terms of the flat outputs. The equation (5.7), (5.8) and (5.9) can be used to calculate the speed  $V$  in terms of the flat outputs. By squaring the both sides of these equations and summing the squared equations, the square of the speed can be obtained. Then the speed  $V$  can be presented as follows:

$$V = \sqrt{\dot{z}_1^2 + \dot{z}_2^2 + \dot{z}_3^2} \quad (5.16)$$

The heading angle  $\psi$  can be expressed using the equation (5.7) and (5.8). Dividing eq. (5.8) by eq.(5.7), the tangent of the heading angle can be obtained. Then, the heading angle  $\psi$  is represented as follows:

$$\psi = \arctan\left(\frac{\dot{z}_2}{\dot{z}_1}\right) \quad (5.17)$$

By using the equation (5.9), the flight path angle  $\gamma$  is described in terms of flat outputs with the equation below.

$$\gamma = \arcsin\left(\frac{\dot{z}_3}{V}\right) \quad (5.18)$$

where the speed  $V$  was already represented in terms of flat outputs in the eq. (5.16).

The derivatives of the speed  $V$ , the heading angle  $\psi$  and the flight path angle  $\gamma$  can be calculated by taking the derivative of the equation (5.16), (5.17) and (5.18), respectively, that correspond to the following equations:

$$\dot{V} = \frac{\dot{z}_1\ddot{z}_1 + \dot{z}_2\ddot{z}_2 + \dot{z}_3\ddot{z}_3}{\sqrt{\dot{z}_1^2 + \dot{z}_2^2 + \dot{z}_3^2}} \quad (5.19)$$

$$\dot{\psi} = \frac{\ddot{z}_2\dot{z}_1 - \dot{z}_2\ddot{z}_1}{\dot{z}_1^2 + \dot{z}_2^2} \quad (5.20)$$

$$\dot{\gamma} = \frac{\ddot{z}_3V - \dot{V}\dot{z}_3}{V\sqrt{\dot{z}_1^2 + \dot{z}_2^2}} \quad (5.21)$$

The control inputs can also be formulated in terms of flat variables via the equation (5.10), (5.11) and (5.12). The bank angle  $\mu$  can be obtained using the equation (5.11) and (5.12). In eq. (5.11),  $\sin(\mu) = \dot{\psi}V \cos(\gamma)/gn_z$  and, in eq. (5.12),  $\cos(\mu) = \dot{\gamma}V + g \cos(\gamma)/gn_z$ . Then  $\mu$  can be found by dividing the first equation by the second equation as follows:

$$\mu = \arctan\left(\frac{V\dot{\psi}\cos(\gamma)}{V\dot{\gamma} + g\cos(\gamma)}\right) \quad (5.22)$$

where state variables  $V, \psi, \gamma$  and their derivatives were already presented in terms of flat variables in the equations (5.16)-(5.21). The longitudinal load factor  $n_x$  can be described in terms of flat outputs by reorganizing the eq. (5.10) as follows:

$$n_x = \frac{\dot{V}}{g} + \sin(\gamma) \quad (5.23)$$

The last control variable  $n_z$  can be expressed by reorganizing the eq. (5.12) as follows:

$$n_z = \frac{\dot{\gamma}V}{g\cos(\mu)} + \frac{\cos(\gamma)}{\cos(\mu)} \quad (5.24)$$

The set of equations (5.15)-(5.24) presents the flat description of the aircraft model. The differential flatness relies on the representation of system variables in terms of the flat outputs and their derivatives. This characteristic provides an advantage when dealing with trajectories. Because of this property, the system dynamics can be described as analytic expressions in the flat output space without integrating any differential equation. Then, the objective of trajectory planning, which is to find a trajectory going from point  $a$  to  $b$ , can be reduced to finding smooth curves that satisfy a set of defined conditions in the flat output space. In this methodology, firstly, all

dynamical and environmental constraints are expressed in terms of flat outputs and their derivatives, then the trajectory planning problem is solved in the flat output space, and in the end, the variables are converted back to the original state and input space. In this way, the number of variables in the optimization problem is reduced to speed up the process.

### 5.3 Differential Flatness-Based Optimal Strategy Generation for Aerial Combat

This section contains the construction of the optimization problem that models the decision process of a fighter in the presence of another decision maker. The decision makers are intelligent opponents who make decisions according to their own self-interest. This is the topic of game theory that focuses on decision process of players with conflicted interests [58].

#### 5.3.1 Aerial combat between aircraft

Let us consider that two combatants simultaneously make a decision without knowing how to other will act, where the fighters have completely opposing interests. This corresponds to a zero-sum game in which a gain for one decision maker leads to a loss for the other, and vice versa.

Suppose that the cost functions of the players are symbolized as  $L_i : U_b \times U_r \rightarrow \mathbb{R}$  for  $i = 1, 2$ , where  $U_b$  and  $U_r$  correspond to action spaces of the blue and red combatants, respectively. An important constraint for zero-sum games is:

$$L_1(u_b, u_r) = -L_2(u_b, u_r) \quad (5.25)$$

which means that the opponents have completely conflicted interests and a loss for one player corresponds to the reward for other. The name of zero-sum originates from this property, which means that the sum of opponents' costs is zero. Instead of defining two different cost functions, it is common to define a cost function  $L$  for first player. In this case, the goal of the blue fighter is to minimize  $L$ , whereas the red aircraft tries to maximize  $L$ . Thus  $L$  can be considered as a reward for red, but a cost for blue.

There are some assumptions on the game setting. The players have knowledge of each other's cost functions. This implies that each player completely understands the opponent's motivation. The other assumption is that the players are rational,

which corresponds to trying to get best reward whenever possible. The first player will not prefer a higher cost action to a lower cost action when it is available. Likewise, the second player will not choose an action that leads to lower cost. The final assumption is that both players make their decisions simultaneously without knowledge of opponents' decisions. For any player, there is no information about the decision of opponent that can be exploited by the player.

Let us consider the game from the stand point of blue. Applying worst-case analysis is rational that contains the evaluation of possible actions of the opponent to make a decision. This corresponds to selecting an action with minimum cost that the first player can guarantee itself no matter what the second player does. Let us define this selection as a security strategy for blue. A security strategy,  $u_b^*$ , for the blue aircraft can be formalized as:

$$u_b^* = \arg \min_{u_b} \left\{ \max_{u_r} \left\{ L(u_b, u_r) \right\} \right\} \quad (5.26)$$

Let us focus on the game from perspective of red, which aims to maximize  $L$ . It can also use worst-case analysis, which means that it chooses an action that guarantees a high cost, in spite of the action of the first player. A security strategy,  $u_r^*$ , for red is described as:

$$u_r^* = \arg \max_{u_r} \left\{ \max_{u_b} \left\{ L(u_b, u_r) \right\} \right\} \quad (5.27)$$

The upper value,  $\bar{L}^*$ , and lower value,  $\underline{L}^*$ , of the game are defined as follows:

$$\bar{L}^* = \max_{u_r} \left\{ L(u_b^*, u_r) \right\}, \quad \underline{L}^* = \min_{u_b} \left\{ L(u_b, u_r^*) \right\} \quad (5.28)$$

An interesting relationship between the upper and lower values is that  $\underline{L}^* \leq \bar{L}^*$  for a zero-sum game. This is shown by observing that

$$\underline{L}^* = \min_{u_b} \left\{ L(u_b, u_r^*) \right\} \leq L(u_b^*, u_r^*) \leq \max_{u_r} \left\{ L(u_b^*, u_r) \right\} = \bar{L}^* \quad (5.29)$$

in which  $L(u_b^*, u_r^*)$  is the cost as a result of players' security strategies. If the players are rational decision makers, then the resulting cost always lies between  $\underline{L}^*$  and  $\bar{L}^*$ . If  $\underline{L}^* = \bar{L}^*$ , the security strategies are called a saddle point that refers to an equilibrium. For some games, the values are equal, but for many  $\underline{L}^* < \bar{L}^*$  [75]. It is the property of the game.



Let us consider the cost function in an aerial combat. The cost can be presented as combination of angular relationship and range score. In air combat, a combatant tries to get an advantageous position to shoot the enemy. The most advantageous position for a fighter is the rear of the opponent. Therefore, combatants try to locate behind of their enemies. The aspect angle ( $AA$ ) and bearing angle ( $BA$ ) of an aircraft can be used to quantify this objective. Let us focus on the blue combatant. The blue aircraft would like to minimize both angles to get air superiority. When  $AA_b$  and  $BA_b$  equal to zero, the blue aircraft's orientation is perfect for shooting. The orientation goal can be presented via  $AA_b$  and  $BA_b$  as follows:

$$L_o = \frac{AA_b + BA_b}{\pi} - 1 \quad (5.30)$$

The blue aircraft tries to minimize  $L_o \in [-1, 1]$ , whereas the red aircraft would like to maximize this score. The second objective is about range between combatants. The distance between aircraft must be smaller than a threshold, which corresponds to missile or gun range, for shooting. The range objective is described as a Gaussian distribution that is given by:

$$L_r = ae^{-\frac{(R-R_d)^2}{2\sigma^2}} \quad (5.31)$$

where  $R \in \mathbb{R}_{>0}$  denotes the range between combatants.  $R_d$  and  $\sigma$  correspond to the desired range and standard deviation, respectively. In this study, the shooting range is taken as 300m.  $R_d = 300$  and  $\sigma = 100$  are chosen to construct the range score. The objective of combat can be represented as combination of these two functions as follows:

$$L = L_o L_r \quad (5.32)$$

The blue combatant desires to minimize the  $L$ , while the red fighter aspires to maximize it. As presented before, the blue aircraft applies the security strategy as follows:

$$\min_{u_b} \max_{u_r} L \quad (5.33)$$

The blue combatant tries to minimize his maximum loss no matter what the red combatant does. This strategy gives the minimum amount of score that blue can guarantee himself.

Let us consider the case that the dynamical system is presented in terms of flat outputs and these outputs are described via B-spline curves. In this case, the objective can be formulated in terms of B-splines' parameters:

$$\min_{P_b} \max_{P_r} F_L(t) \quad (5.34)$$

where  $P_b$  and  $P_r$  are control points of B-splines that present the trajectory of the blue and red aircraft, respectively.  $F_L(t)$  refers to the objective  $L$  in terms of B-splines. Let  $w$  be a new variable that is used to represent the objective function in standard optimization form. The new variable is restricted to be greater than objective function,  $w \geq \max F_L(t)$ , and it is tried to make  $w$  as small as possible. The problem is represented as:

Choose  $P_b$  and  $w$  to

$$\text{minimize } w \quad (5.35)$$

subject to:

$$F_L^{\tilde{x}(x_0^r, \tilde{u})}(t) \leq w \quad \forall \tilde{u} \in \tilde{U} \quad (5.36)$$

$$lb \leq c(z) \leq ub \quad (5.37)$$

$$e(z) = v \quad (5.38)$$

where  $\tilde{u}$  corresponds to an action trajectory and  $\tilde{U}$  contains samples for action trajectories. The state trajectory for red is symbolized as  $\tilde{x}(x_0^r, \tilde{u})$ , which is derived from  $\tilde{u}$ .  $F_L^{\tilde{x}(x_0^r, \tilde{u})}(t)$  denotes the value of score for sampled state trajectory of red  $\tilde{x}(x_0^r, \tilde{u})$  with respect to the set of decision variables  $P_b$  at time  $t$ . The set of all state trajectories derived from action trajectories in  $\tilde{U}$  creates the reachable set of the red aircraft. By evaluating each state in the sampled reachable set of the red aircraft, blue tries to find its security strategy. The vector  $c(z)$  consists of the inequality constraints in output space such as speed constrains, orientation constraints etc..  $ub$  and  $lb$  are the upper and lower bounds for associated variables, respectively. The vector  $e(z)$  contains the equality constraints in output space such as initial conditions, way-points etc.. The solution of this optimization problem gives the best strategy for the blue fighter. By constructing and solving the same problem from perspective of the red combatant, the optimal strategy for the red aircraft can also be generated.

### 5.3.2 Reachable set of enemy aircraft

The set of constraints in Eq. (5.36) contains the reachable set of the enemy aircraft. The reachable set of an aircraft can be generated via a sampling-based method. A state trajectory can be derived from an action trajectory  $\tilde{u}$  as follows:

$$x(T) = x(0) + \int_0^T f(x(t), u(t)) dt \quad (5.39)$$

where  $x(0)$  is the initial state at time  $t = 0$ . Let  $\tilde{x}(x_0, \tilde{u})$  denote the state trajectory over all time, obtained via integration (5.39). Let  $U$  be the set of all permissible action trajectories on the time interval  $[0, \infty)$ . A state trajectory  $\tilde{x}(x_0, \tilde{u})$  can be generated from each  $\tilde{u} \in U$  via equation (5.39). The main question is which states in  $X$  are visited by these trajectories? In general, it may not be possible to visit some states because of dynamical constraints. The set of states that can be reached from an initial condition is described as reachable set of the dynamical system.

Let  $R(x_0, U) \subseteq X$  denote the reachable set from  $x_0$ , which is the set of all states that are visited by any trajectories that start at  $x_0$  and are obtained from some  $\tilde{u} \in U$  by integration. This can be expressed as

$$R(x_0, U) = \{x_1 \in X : \exists \tilde{u} \in U \text{ and } \exists t \in [0, \infty) \text{ such that } x(t) = x_1\} \quad (5.40)$$

where  $x(t)$  is given by (5.39) and  $x(0)$  is symbolized as  $x_0$ .

Consider the generation of a reachable set for a fixed time period. Let the time-limited reachable set  $R(x_0, U, t)$  be the subset of  $R(x_0, U)$  that is reached up to time  $t$ . Formally, this is

$$R(x_0, U, t) = \{x_1 \in X : \exists \tilde{u} \in U \text{ and } \exists t' \in [0, t) \text{ such that } x(t') = x_1\} \quad (5.41)$$

Under differential constraints, a sampling-based approach can be used by sampling the space of action trajectories to generate the time-limited reachable set. This results in a reduced set of possible action trajectories. The main difference in the current setting is that the approach here work with a sample sequence over  $U$  to generate the reachable set.

### 5.3.3 Initialization scheme

An optimization problem is said to be convex if the objective is a convex function, as well as the constraint set is a convex set. An optimization problem that violates either one of these conditions, i.e., one that has a non-convex objective, or a non-convex constraint set, or both, is called a non-convex optimization problem. A function  $f : S \rightarrow \mathbb{R}^{n_f}$  is convex if its domain  $S$  is a convex set and

$$\forall s_1, s_2 \in S, 0 \leq \alpha \leq 1 \Rightarrow f(\alpha s_1 + (1 - \alpha)s_2) \leq \alpha f(s_1) + (1 - \alpha)f(s_2) \quad (5.42)$$

In our problem, objective function and some of the constraints are non-convex. When dealing with a non-convex optimization problem, initial guess is important that can improve the performance of algorithm. The bounds of the decision variables should also be specified. In our case, the decision variables are control points that parametrize the location of the aircraft in  $x, y$  and  $h$ . The bounds of control points can be specified via maximum speed and climb/descent ratio of the aircraft. Let  $T$  be the planning horizon, then the bounds for control points on  $x$ -axis and  $y$ -axis can be expressed as follows:

$$\begin{aligned} (x_{min}^b, x_{max}^b) &= (x_0 - 2V_{max}T, x_0 + 2V_{max}T) \\ (y_{min}^b, y_{max}^b) &= (y_0 - 2V_{max}T, y_0 + 2V_{max}T) \end{aligned} \quad (5.43)$$

the control points that parametrize the  $z_1$  should be on the interval  $[x_{min}^b, x_{max}^b]$ , while the control points for  $z_2$  should be between  $y_{min}^b$  and  $y_{max}^b$ . By setting bounds of each control point to these values, all of the feasible trajectories can be evaluated during optimization process without focusing on control points that violate the aircraft dynamics. As in the horizontal plane, the control points that specify the altitude can also be bounded via maximum rate of climb/descent:

$$(h_{min}^b, h_{max}^b) = (h_0 - 2\dot{h}_{max}T, h_0 + 2\dot{h}_{max}T) \quad (5.44)$$

The initial guess of the control points can be obtained via convex hull of the enemy's reachable set. Let us define the convex set and convex hull. A set  $S \subseteq \mathbb{R}^{n_s}$  is a convex set if the line segment connecting any pair of points of  $S$  lies entirely in  $S$ , i.e.

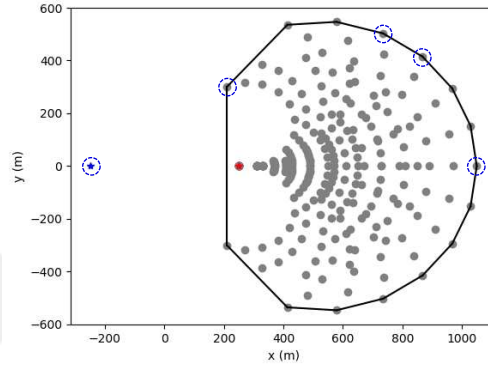
$$s_1, s_2 \in S, 0 \leq \alpha \leq 1 \Rightarrow \alpha s_1 + (1 - \alpha)s_2 \in S \quad (5.45)$$

The convex hull of a set  $S \subseteq \mathbb{R}^{n_s}$  is the smallest convex set containing  $S$  [76], i.e.

$$\text{conv}(S) := \bigcap \left\{ \tilde{S} \subseteq \mathbb{R}^{n_s} : S \subseteq \tilde{S}, \tilde{S} \text{ is convex} \right\} \quad (5.46)$$

For any finite convex set  $S = \{s_1, \dots, s_k\}$  with  $k \in \mathbb{N}$  it follows that

$$\text{conv}(S) = \left\{ \sum_{i=1}^k \alpha_i s_i : \forall s_i \in S, \alpha_i \geq 0, i = 1, \dots, k, \sum_{i=1}^k \alpha_i = 1, k \in \mathbb{N} \right\} \quad (5.47)$$



**Figure 5.1** : Initialization via Convex Hull of Enemy's Reachable Set.

The control points can be initialized using the elements of the convex hull of enemy's reachable set. The first control points are taken as current position of the fighter. The coordinates of the closest point on the convex hull to the current position of the aircraft are defined as second control points. The farthest point on the convex hull to the current location is used to specify the last control points. The other control vertices are taken as some of the points on the convex hull between closest and farthest points. Let us consider an illustrative example that the positions of the blue and red fighters are  $[-250, 0, 3000]$  and  $[250, 0, 3000]$  with  $0$  heading angle, respectively, as shown in Figure 5.1. Assume that the performance constraints are as follows:  $V_{min} = 30m/s$ ,  $V_{max} = 40m/s$ ,  $\mu_{min} = -30^\circ$ ,  $\mu_{max} = 30^\circ$ , the number of control points that construct the curve for a flat output is 5 and the planning horizon is  $20s$ . In this case, the reachable set of the red combatant consists of the grey points in the Figure 5.1. The convex hull that contains the reachable set of red is illustrated via the black line in the Figure 5.1. And, the initial control points on the horizontal plane for the blue aircraft are indicated via dashed blue circles that consist of the current position of the blue aircraft and the vertices on the convex hull of the red's reachable set. These initial control points construct a curve that lies partly in the convex hull of the enemy's reachable set. From perspective of aerial combat, this initialization strategy is logical

that contains staying close to the enemy and intercepting enemy's trajectory with an orientation advantage. If the aircraft has a disadvantage at the beginning of the combat, the strategy will also be logical that contains an initialized trajectory that closes to the dynamic limitations of the enemy. Therefore, the disadvantageous situation can also be resolved around this initialization.

### 5.3.4 Solution method

In this study, we use a Sequential Quadratic Programming (SQP)-based method that is presented in the paper [77] to solve the optimization problem in (5.35)-(5.38). Let us present the general nonlinear programming problem:

$$\begin{aligned}
 & \min_{x \in \mathbb{R}^n} f(x) \\
 & \text{subject to:} \\
 & g_i(x) = 0, \quad i = 1, \dots, m_e \\
 & g_i(x) \geq 0, \quad i = m_e + 1, \dots, m \\
 & x_l \leq x \leq x_u
 \end{aligned} \tag{5.48}$$

In the SQP-based methods, this problem is solved iteratively by updating the value of the vector  $x$ , which is initialized as  $x^0$ , via the following equation:

$$x^{k+1} := x^k + \alpha^k d^k \tag{5.49}$$

where superscript  $k$  indicates the number of step.  $d^k$  and  $\alpha^k$  correspond to the search direction and the step length at the  $k^{\text{th}}$  step, respectively. A quadratic programming sub-problem is solved to determine the search direction. This sub-problem is formulated by a quadratic approximation of the Lagrange function of the optimization problem and a linear approximation of the constraints  $g_i$ , where the Lagrange function is presented as follows:

$$L(x, \lambda) = f(x) - \sum_{i=1}^m \lambda_i g_i(x) \tag{5.50}$$

This sub-problem is of the following standard form of quadratic programming:

$$\min_{d \in \mathbb{R}^n} \frac{1}{2} d^T B^k d + \nabla f(x^k) d$$

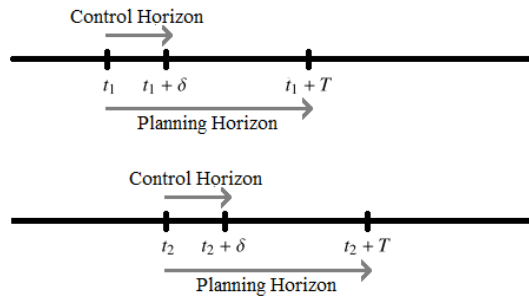
subject to:

$$\begin{aligned} \nabla g_i(x^k)d + g_i(x^k) &= 0, \quad i = 1, \dots, m_e \\ \nabla g_i(x^k)d + g_i(x^k) &\geq 0, \quad i = m_e + 1, \dots, m \end{aligned} \quad (5.51)$$

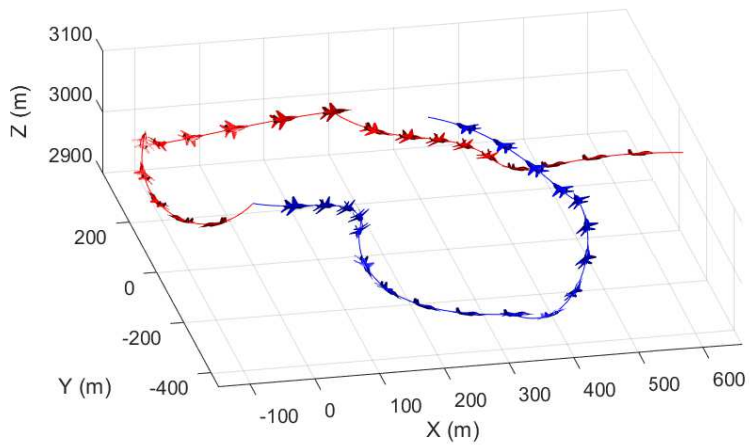
in a general SQP scheme,  $B$  is taken as  $\nabla_{xx}^k L(x, \lambda)$ . In the paper [77], which is the used algorithm in this study, the objective function is replaced by a linear least squares term that utilizes the stable factorization of the matrix  $B$ . The details about the update and approximation strategies can be found in [77].

### 5.3.5 Receding horizon control

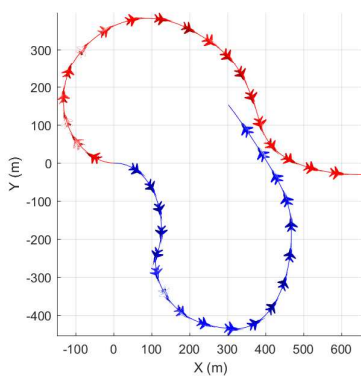
Receding horizon control (RHC) corresponds to solving an optimization problem in a receding time horizon to generate control actions for a system, which is illustrated in Figure 5.2. Using system model and current state information, RHC generates the control strategy over a planning horizon via solving constrained optimization problem, but only implements the control actions for the control horizon. The optimization problem is solved at the time  $t$  over the interval  $[t, t + T]$ , where  $T$  denotes the planning horizon. Then, the optimal trajectory is used to drive the system from time  $t$  to time  $t + \delta$ , where the  $\delta$  corresponds to control horizon. Every  $\delta$  seconds, this process is repeated to find a new optimal trajectory. In this study, we use the receding horizon approach to produce the strategies of the combatants. At every  $\delta$  seconds, using initial conditions at the current time  $t$ , the optimization problem in (5.35)-(5.38) is solved over the interval  $[t, t + T]$  for blue and also red, separately, and then the strategies for both fighters are implemented during  $\delta$  seconds.



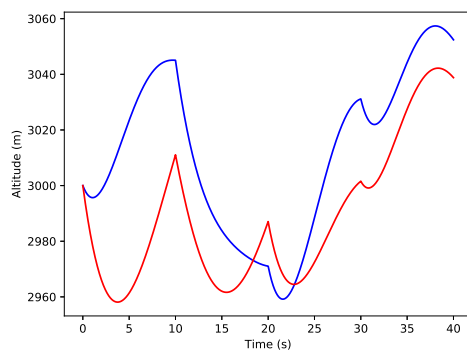
**Figure 5.2 :** Principle of Receding Horizon Control.



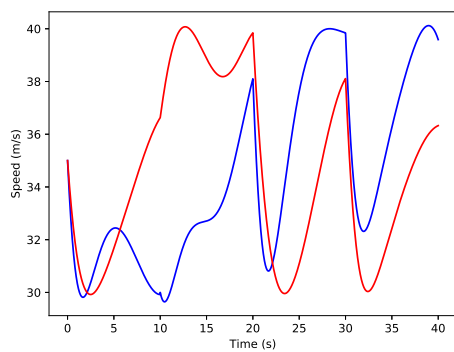
(a)



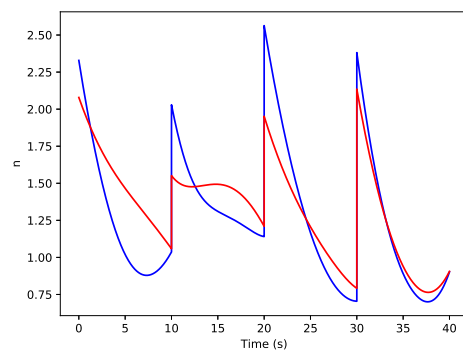
(b)



(c)



(d)



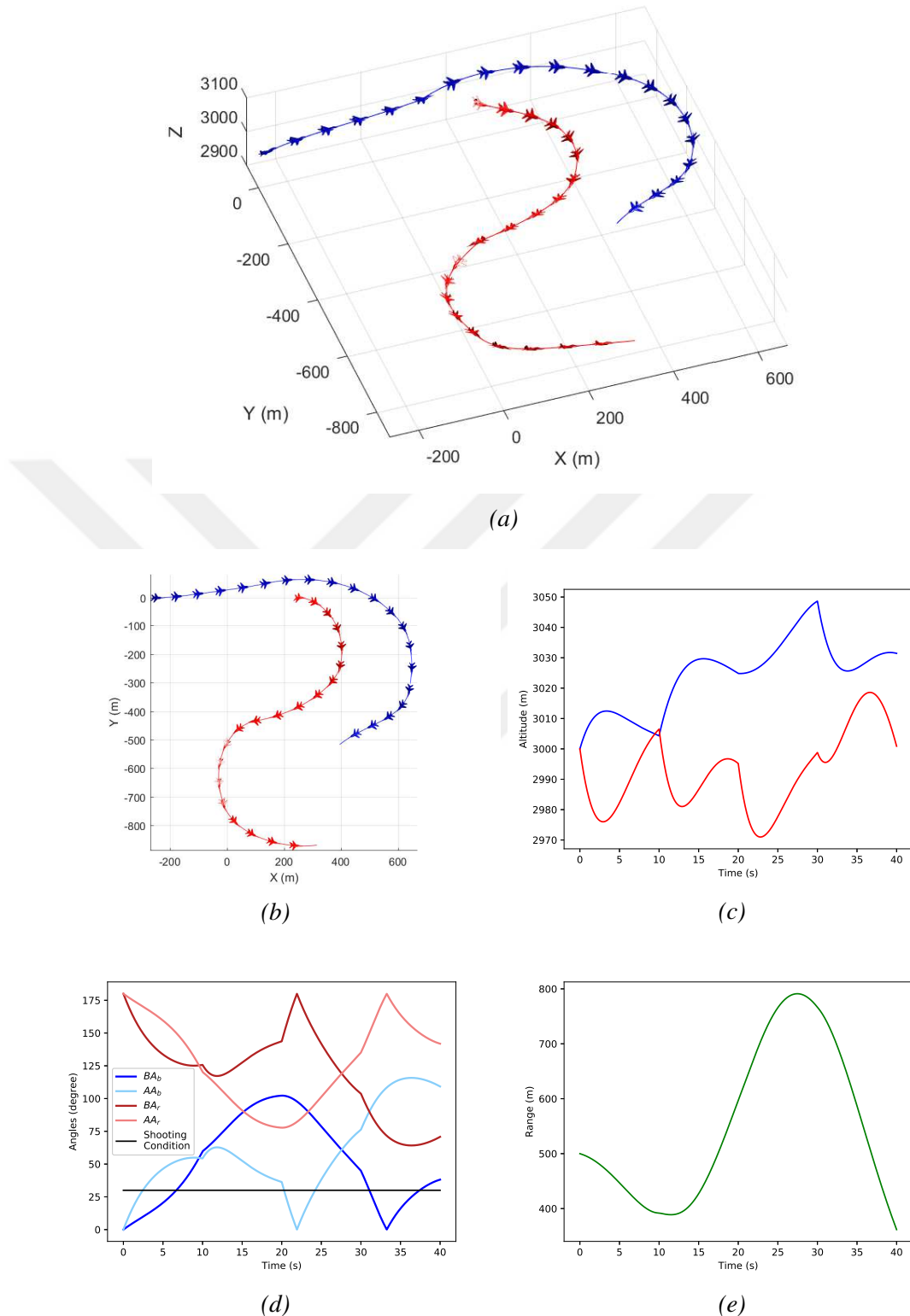
(e)

**Figure 5.3** : Aerial Combat between two UAVs: Scenario 1 (a) 3D Positions, (b) Horizontal Positions, (c) Altitudes, (d) Speeds, (e) Total Load Factors.

## 5.4 Simulation Results

We analyze two different scenarios to show the validity of the method. The first scenario corresponds to equivalent initial conditions without any air superiority, while





**Figure 5.4** : Aerial Combat between two UAVs: Scenario 2 (a) 3D Positions, (b) Horizontal Positions, (c) Altitudes, (d) Combat Angles, (e) Range.

the second case examines the effect of a fighter's initial orientation advantage during aerial combat. The performance limitations of the combatants are defined as  $V_{min} = 30m/s$ ,  $V_{max} = 40m/s$ ,  $\mu_{min} = -30^\circ$ ,  $\mu_{max} = 30^\circ$  and  $n_{max}^2 = 7$ . The degree of the

B-spline curves is chosen as 4. The shooting clearance is described via combat angles and the range between the combatants. The range, aspect angle and bearing angle thresholds are taken as  $300m$ ,  $30^\circ$  and  $30^\circ$ , respectively. It is assumed that the attacker has a shooting clearance to win the combat, when the range, attacker's aspect angle and attacker's bearing angle are smaller than the defined threshold values. During the implementations, the planning horizon  $T$  is taken as 20 seconds and the control horizon  $\delta$  is defined as 10 seconds. Hence, at every 10 seconds, optimal strategies for both blue and red are generated by solving the optimization problem over the 20 seconds time horizon and the strategies are used over the interval  $[t, t + 10]$  for each solution.

The first scenario begins with tail-to-tail position of the combatants at  $(x_0, y_0, h_0) = (0, 0, 3000)$ . Both combatants have  $35m/s$  speed at the beginning of the combat. None of aircraft has air superiority with these initial conditions. The results of the aerial combat are illustrated in Figure 5.3. As shown in Figure 5.3d and 5.3e, the constrained variables are within limits during aerial combat. The speed, load factor and also bank angle do not violate the bounds. The algorithm generates feasible strategies. The positions of the combatants are presented in Figure 5.3a, Figure 5.3b and Figure 5.3c. At the beginning of the combat, the fighters are in symmetric conditions without any air superiority. This equivalent situation continues until the end of the implementation and the combatant comes symmetric orientation again around the end of implementation as shown in Figure 5.3b. The combatants make circular motions to get a clearance for shooting and also try to stay close during combat due to the range score. They select optimal strategies to prevent the opponent's victory and get an advantageous position to win the combat. However, none of them gets air superiority during combat and they end the combat at equivalent conditions. These results show that the algorithm is capable of generating rational outcomes for aerial combat between two equivalent aircraft in the symmetric initial conditions without any air superiority.

In the second scenario, the blue aircraft has an orientation advantage at the beginning of the combat. The initial conditions are specified as  $(x_0^b, y_0^b, h_0^b) = (-250, 0, 3000)$ ,  $\psi_0^b = 0$ ,  $V_0^b = 35$  and  $(x_0^r, y_0^r, h_0^r) = (250, 0, 3000)$ ,  $\psi_0^r = 0$ ,  $V_0^r = 35$ . The security strategies are generated as solutions of the optimization problems with presented receding horizon scheme and the results are illustrated in Figure 5.4. At the beginning of the combat, the blue fighter has air superiority without sufficient range for a shooting. This orientation

advantage continues for 3 seconds with lack of range feasibility for shooting as shown in Figure 5.4d and Figure 5.4e. The position information of the fighters are represented in Figure 5.4a, Figure 5.4b and Figure 5.4c. During combat, the blue aircraft pursues the red combatant to obtain an advantageous position for shooting, while red tries to avoid the disadvantageous orientation. Towards the end of the simulation, the range between aircraft is close to the threshold of the feasible range and blue has sufficient bearing angle, but it loses the necessary aspect angle for shooting as shown in Figure 5.4d. As presented in the results, the red combatant manages to escape from blue's shooting via optimal strategy that causes either orientation infeasibility or range infeasibility for a shooting.





## 6. CONCLUSION

In this thesis, we have focused on autonomous control of cooperative and noncooperative multi-aircraft systems. We have presented two different optimization-based methodology and their implementations to the problems in civil and military aviation. Whereas the first method was based on hybrid system theory, the second one was mainly about differential flatness theory.

First, we have focused on hybrid maneuver-based optimization approach to assist the decision making procedure in an air-to-air combat. An aircraft performance model was constructed and implemented using the performance parameters of F-16. Known air combat maneuvers were formulated through hybrid automaton spanning maneuver space of the aircraft. Objective functions for air superiority were presented, and game theoretical evaluation strategy was expressed. The working principle of the methods was presented, and the validity of the algorithm was demonstrated with implementations. Designed platform can be used to simulate air combat to assist the decision-making procedure. It can be used for decision support to fighter pilots, autonomous air combat, training of the fighter pilots and survivability analysis.

Second, we have presented how a similar hybrid maneuver-based optimization approach can be used to model an autonomous air traffic control system to improve airspace capacity. Although the proposed system was presented as a fully autonomous system, it can also be used as a semi-autonomous system for providing decision support to human air traffic operators. The proposed system is based on an integer linear programming formulation, constructed via a mapping process that discretizes the airspace, with predicted trajectories to ensure the safety. A trajectory prediction method was also introduced. We showed that the proposed approach is scalable for large-scale ATM scenarios. The sub-components of the proposed system can also be used individually in different ATM applications. For example, the trajectory prediction model can be used when dealing with trajectory management, or the mapping process can be utilized when dealing with mitigation of airspace congestion. Moreover, we

defined an airspace capacity estimation procedure to determine the capacity of an airspace with the proposed ATC system. We showed that the proposed approach can manage traffic approximately 10 times denser than current traffic.

Then, we have focused on the second method that generates control inputs as continuous real valued functions instead of predefined maneuvers. We presented the optimization-based method for the mission planning and control of multi-aircraft systems with STL specifications. We specified the missions of aircraft via an STL formalism, and described the system dynamics as a (partially) differentially flat system. We then constructed a MILP-based formulation to generate optimal trajectories that satisfy STL specifications. In our examples, we used a realistic aircraft model with performance parameters of Boeing 737-800 and realistic conditions to simulate arrival traffic, and evaluated the performance of the proposed method. Moreover, we showed that the method can be applied to other multi-agent systems, such as mission planning and control of multiple unmanned aircraft.

We also have presented how this approach can be utilized to solve aerial combat problem. We described the flat(meaning acyclic) optimal-maneuver generation algorithm for aerial combat games. This approach enabled us to determine the control input and the state sequence with regards to the desired output trajectory and its derivatives, which were formulated with b-spline functions. Then, the air combat problem has been translated into an optimization problem through a cost function with aspect angle, bearing angle and range between two aircraft. At this level, the optimization problem has been set to satisfy the given and dynamical constraints. For the simulation and demonstration purposes, we have applied this methodology to air combat between two UAVs and shown the results for the example scenarios. The results state that the proposed methodology is able to provide solutions to the air combat of autonomous aerial vehicles.

We have developed two different methods for the autonomous control of cooperative and noncooperative multi-aircraft systems. While the first method is based on hybrid system theory, the second method is mainly based on differential flatness theory. Both methods generate control inputs as a result of an optimization process. We have shown that both methods have different contributions. Basically, the main advantages of the second method are that the control inputs are generated in continuous-time, the

complex tasks can be assigned to the aircraft and the method can be used with different multi-agent systems, whereas the first method has better scalability.







## REFERENCES

- [1] **Baumeister, R.F. and Vohs, K.D.** (2007). *Encyclopedia of social psychology*, volume 1, Sage.
- [2] **Berger, J.O.** (2013). *Statistical decision theory and Bayesian analysis*, Springer Science & Business Media.
- [3] **Shaw, R.L.** (1985). *Fighter Combat*, Naval Institute Press.
- [4] **Sheffield, R.G.** (1990). *The Official F-15 Strike Eagle Handbook*, Compute Pu.
- [5] **Airbus** (2016). Global Market Forecast: Mapping Demand 2016 - 2035, **Technical ReportD14029463**, Airbus.
- [6] **Mitchell, I.M. and Tomlin, C.J.** (2003). Overapproximating reachable sets by Hamilton-Jacobi projections, *journal of Scientific Computing*, 19(1), 323–346.
- [7] **Mitchell, I.M., Bayen, A.M. and Tomlin, C.J.** (2005). A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games, *IEEE Transactions on automatic control*, 50(7), 947–957.
- [8] **Girard, A., Le Guernic, C. and Maler, O.** (2006). Efficient computation of reachable sets of linear time-invariant systems with inputs, *International Workshop on Hybrid Systems: Computation and Control*, Springer, pp.257–271.
- [9] **Stursberg, O. and Krogh, B.H.** (2003). Efficient representation and computation of reachable sets for hybrid systems, *International Workshop on Hybrid Systems: Computation and Control*, Springer, pp.482–497.
- [10] **Sprinkle, J., Eklund, J.M., Kim, H.J. and Sastry, S.** (2004). Encoding aerial pursuit/evasion games with fixed wing aircraft into a nonlinear model predictive tracking controller, *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*, volume 3, IEEE, pp.2609–2614.
- [11] **Eklund, J.M., Sprinkle, J. and Sastry, S.** (2005). Implementing and testing a nonlinear model predictive tracking controller for aerial pursuit/evasion games on a fixed wing aircraft, *Proceedings of the 2005, American Control Conference, 2005.*, IEEE, pp.1509–1514.
- [12] **Burgin, G.H. and Sidor, L.** (1988). Rule-based air combat simulation, **Technical Report**, TITAN SYSTEMS INC LA JOLLA CA.

- [13] **Virtanen, K., Karelahti, J. and Raivio, T.** (2006). Modeling air combat by a moving horizon influence diagram game, *Journal of Guidance, Control, and Dynamics*, 29(5), 1080–1091.
- [14] **Austin, F., Carbone, G., Falco, M., Hinz, H. and Lewis, M.** (1987). Automated maneuvering decisions for air-to-air combat, *Guidance, Navigation and Control Conference*, p.2393.
- [15] **Austin, F., Carbone, G., Hinz, H., Lewis, M. and Falco, M.** (1990). Game theory for automated maneuvering during air-to-air combat, *Journal of Guidance, Control, and Dynamics*, 13(6), 1143–1149.
- [16] **McGrew, J.S.** (2008). Real-time maneuvering decisions for autonomous air combat, *Ph.D. thesis*, Massachusetts Institute of Technology.
- [17] **Kelly III, W.E.** (1999). Conflict detection and alerting for separation assurance systems, *Digital Avionics Systems Conference, 1999. Proceedings. 18th*, volume 2, IEEE, pp.6–D.
- [18] **Shewchun, J.M., Oh, J.H. and Feron, E.** (1997). Linear matrix inequalities for free flight conflict problems, *Decision and Control, 1997., Proceedings of the 36th IEEE Conference on*, volume 3, IEEE, pp.2417–2422.
- [19] **Paielli, R.A. and Erzberger, H.** (1997). Conflict Probability for Free Flight, *Journal of Guidance, Control, and Dynamics*, 20(3), 588–596.
- [20] **Sridhar, B. and Chatterji, G.** (1997). Computationally efficient conflict detection methods for air traffic management, *American Control Conference, 1997. Proceedings of the 1997*, volume 2, IEEE, pp.1126–1130.
- [21] **Vink, A., Kauppinen, S., Beers, J. and de Jong, K.** (1997). Medium term conflict detection in EATCHIP phase III, *Digital Avionics Systems Conference, 1997. 16th DASC., AIAA/IEEE*, volume 2, IEEE, pp.9–3.
- [22] **Bakker, G. and Blom, H.A.** (1993). Air traffic collision risk modelling, *Decision and Control, 1993., Proceedings of the 32nd IEEE Conference on*, IEEE, pp.1464–1469.
- [23] **Eby, M.S.** (1994). A Self-Organizational Approach for Resolving Air Traffic Conflicts., *Lincoln Laboratory Journal*.
- [24] **Koren, Y. and Borenstein, J.** (1991). Potential field methods and their inherent limitations for mobile robot navigation, *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, IEEE, pp.1398–1404.
- [25] **Tomlin, C., Pappas, G.J. and Sastry, S.** (1998). Conflict resolution for air traffic management: A study in multiagent hybrid systems, *Automatic Control, IEEE Transactions on*, 43(4), 509–521.
- [26] **Tomlin, C., Mitchell, I. and Ghosh, R.** (2001). Safety verification of conflict resolution manoeuvres, *Intelligent Transportation Systems, IEEE Transactions on*, 2(2), 110–120.

- [27] **Bayen, A.M., Grieder, P. and Tomlin, C.J.** (2002). A control theoretic predictive model for sector-based air traffic flow.
- [28] **Durand, N., Alliot, J.M. and Granger, G.** (1997). Optimal resolution of en route conflicts, *Proceedings of the 1st USA/Europe Seminar*.
- [29] **Frazzoli, E., Mao, Z.H., Oh, J.H. and Feron, E.** (2001). Resolution of conflicts involving many aircraft via semidefinite programming, *Journal of Guidance, Control, and Dynamics*, 24(1), 79–86.
- [30] **Pallottino, L., Feron, E.M. and Bicchi, A.** (2002). Conflict resolution problems for air traffic management systems solved with mixed integer programming, *Intelligent Transportation Systems, IEEE Transactions on*, 3(1), 3–11.
- [31] **Baspinar, B., Pasaoglu, C., Ure, N.K. and Inalhan, G.** (2015). Infrastructure development for ground-based separation assurance with optional automation, *Proceedings of the 5th International Conference on Application and Theory of Automation in Command and Control Systems*, ACM, pp.65–74.
- [32] **Pasaoglu, C., Baspinar, B., Ure, N.K. and Inalhan, G.** (2016). Hybrid systems modeling and automated air traffic control for three-dimensional separation assurance, *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 230(9), 1788–1809.
- [33] **Yang, L.C. and Kuchar, J.K.** (1997). Prototype conflict alerting system for free flight, *Journal of Guidance, Control, and Dynamics*, 20(4), 768–773.
- [34] **Erzberger, H., Davis, T.J. and Green, S.** (1993). Design of center-TRACON automation system.
- [35] **Brudnicki, D. and McFarland, A.** (1997). User Request Evaluation Tool (URET) conflict probe performance and benefits assessment, *MITRE CAASD, MP, 1130065520*.
- [36] **Hilburn, B.** (2004). Cognitive complexity in air traffic control: A literature review, *EEC note*, 4(04).
- [37] **Warfield, J.N. and Cárdenas, A.R.** (1994). *A handbook of interactive management*, Iowa State University Press Ames.
- [38] **Histon, J.M. and Hansman, R.J.** (2008). Mitigating complexity in air traffic control: the role of structure-based abstractions, *Ph.D. thesis*, Massachusetts Institute of Technology, Department of Aeronautics and . . . .
- [39] **Laudeman, I.V., Shelden, S., Branstrom, R. and Brasil, C.** (1998). Dynamic density: An air traffic management metric.
- [40] **Kopardekar, P., Schwartz, A., Magyarits, S. and Rhodes, J.** (2007). Airspace complexity measurement: An air traffic control simulation analysis, *7th USA/Europe Air Traffic Management R&D Seminar, Barcelona, Spain*.

- [41] **Chatterji, G. and Sridhar, B.** (2001). Measures for air traffic controller workload prediction, *1st AIAA, Aircraft, Technology Integration, and Operations Forum*, p.5242.
- [42] **Mondoloni, S.** (2001). Airspace Fractal Dimensions and Applications, *The 4th ATM Seminar, 2001*.
- [43] **Lee, K., Feron, E. and Pritchett, A.** (2007). Air traffic complexity: an input-output approach, *American Control Conference, 2007. ACC'07*, IEEE, pp.474–479.
- [44] **Maler, O. and Nickovic, D.** (2004). Monitoring temporal properties of continuous signals, *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*, Springer, pp.152–166.
- [45] **Karaman, S., Sanfelice, R.G. and Frazzoli, E.** (2008). Optimal control of mixed logical dynamical systems with linear temporal logic specifications, *2008 47th IEEE Conference on Decision and Control*, IEEE, pp.2117–2122.
- [46] **Raman, V., Donzé, A., Maasoumy, M., Murray, R.M., Sangiovanni-Vincentelli, A. and Seshia, S.A.** (2014). Model predictive control with signal temporal logic specifications, *53rd IEEE Conference on Decision and Control*, IEEE, pp.81–87.
- [47] **Saha, S. and Julius, A.A.** (2016). An MILP approach for real-time optimal controller synthesis with metric temporal logic specifications, *2016 American Control Conference (ACC)*, IEEE, pp.1105–1110.
- [48] **Pant, Y.V., Abbas, H. and Mangharam, R.** (2017). Smooth operator: Control using the smooth robustness of temporal logic, *2017 IEEE Conference on Control Technology and Applications (CCTA)*, IEEE, pp.1235–1240.
- [49] **Abbas, H., Fainekos, G., Sankaranarayanan, S., Ivančić, F. and Gupta, A.** (2013). Probabilistic temporal logic falsification of cyber-physical systems, *ACM Transactions on Embedded Computing Systems (TECS)*, 12(2s), 95.
- [50] **Saha, I., Ramaithitima, R., Kumar, V., Pappas, G.J. and Seshia, S.A.** (2014). Automated composition of motion primitives for multi-robot systems from safe LTL specifications, *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, pp.1525–1532.
- [51] **DeCastro, J.A., Alonso-Mora, J., Raman, V., Rus, D. and Kress-Gazit, H.** (2018). Collision-free reactive mission and motion planning for multi-robot systems, *Robotics Research*, Springer, pp.459–476.
- [52] **Pant, Y.V., Abbas, H., Quaye, R.A. and Mangharam, R.** (2018). Fly-by-logic: Control of multi-drone fleets with temporal logic objectives, *Proceedings of the 9th ACM/IEEE International Conference on Cyber-Physical Systems*, IEEE Press, pp.186–197.
- [53] **Lygeros, J., Tomlin, C. and Sastry, S.** (1999). Hybrid systems: modeling, analysis and control, *preprint*.

- [54] **Raskin, J.F.**, (2005). An introduction to hybrid automata, *Handbook of networked and embedded control systems*, Springer, pp.491–517.
- [55] **Nguyen, L.T., Ogburn, M.E., Gilbert, W.P., Kibler, K.S., Brown, P.W. and Deal, P.L.** (1979). Simulator study of stall/post-stall characteristics of a fighter airplane with relaxed longitudinal static stability.[F-16].
- [56] **Stevens, B.L., Lewis, F.L. and Johnson, E.N.** (2015). *Aircraft control and simulation: dynamics, controls design, and autonomous systems*, John Wiley & Sons.
- [57] **Ferguson, T.S.**, (2014), *Game Theory*.
- [58] **Başar, T. and Olsder, G.J.** (1998). *Dynamic noncooperative game theory*, SIAM.
- [59] **Baspinar, B., Uzun, M., Faik, A., Basturk, T., Tasdelen, I., Koyuncu, E. and Inalhan, G.** (2017). A 4D Trajectory Generation Infrastructure Tool for Controller Working Position, *36th Digital Avionics System Conference*, AIAA/IEEE.
- [60] (2013). User Manual for the Base of Aircraft Data (BADA) Revision 3.11, **Technical Report**, Eurocontrol Experimental Centre.
- [61] (2014). User Manual for the Base of Aircraft Data (BADA) Family 4, **Technical Report**, Eurocontrol Experimental Centre.
- [62] **Sujit, P., Saripalli, S. and Sousa, J.** (2013). An evaluation of UAV path following algorithms, *Control Conference (ECC), 2013 European*, IEEE, pp.3332–3337.
- [63] **Bulusu, V. and Polishchuk, V.** (2017). A threshold based airspace capacity estimation method for UAS traffic management system, *IEEE Systems Conference*.
- [64] **Fainekos, G.E. and Pappas, G.J.** (2009). Robustness of temporal logic specifications for continuous-time signals, *Theoretical Computer Science*, 410(42), 4262–4291.
- [65] **Donzé, A. and Maler, O.** (2010). Robust satisfaction of temporal logic over real-valued signals, *International Conference on Formal Modeling and Analysis of Timed Systems*, Springer, pp.92–106.
- [66] **Fliess, M., Lévine, J., Martin, P. and Rouchon, P.** (1995). Flatness and defect of non-linear systems: introductory theory and examples, *International journal of control*, 61(6), 1327–1361.
- [67] **Fliess, M., Lévine, J., Martin, P. and Rouchon, P.** (1999). A lie-backlund approach to equivalence and flatness of nonlinear systems, *IEEE Transactions on automatic control*, 44(5), 922–937.
- [68] **Baspinar, B. and Koyuncu, E.** (2019). Differential Flatness-based Optimal Air Combat Maneuver Strategy Generation, *AIAA Scitech 2019 Forum*, p.1985.

- [69] **Levine, J.** (2009). *Analysis and control of nonlinear systems: A flatness-based approach*, Springer Science & Business Media.
- [70] **Piegl, L. and Tiller, W.** (2012). *The NURBS book*, Springer Science & Business Media.
- [71] **Uzun, M., Başpınar, B., Koyuncu, E. and İnalhan, G.** (2017). Takeoff weight error recovery for tactical trajectory prediction automaton of air traffic control operator, *2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)*, IEEE, pp.1–7.
- [72] **Schouwenaars, T.** (2006). Safe trajectory planning of autonomous vehicles, *Ph.D. thesis*, Massachusetts Institute of Technology.
- [73] **Mitchell, S., OSullivan, M. and Dunning, I.** (2011). PuLP: a linear programming toolkit for python, *The University of Auckland, Auckland, New Zealand*.
- [74] **Formentin, S. and Lovera, M.** (2011). Flatness-based control of a quadrotor helicopter via feedforward linearization, *2011 50th IEEE Conference on Decision and Control and European Control Conference*, IEEE, pp.6171–6176.
- [75] **LaValle, S.M.** (2006). *Planning algorithms*, Cambridge university press.
- [76] **Christophersen, F.** (2007). *Optimal control of constrained piecewise affine systems*, volume359, Springer.
- [77] **Kraft, D.** (1988). A software package for sequential quadratic programming, *Forschungsbericht- Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt*.

## CURRICULUM VITAE



**Name Surname:** Bariş Bařpınar

**Place and Date of Birth:** Elazığ, June 1990

**E-Mail:** baspınarb@itu.edu.tr

### EDUCATION:

- **B.Sc.:** 2013, Istanbul Technical University, Faculty of Aeronautics and Astronautics, Aeronautical Engineering
- **B.Sc.:** 2013, Istanbul Technical University, Faculty of Electrical and Electronics Engineering, Control Engineering
- **M.Sc.:** 2015, Istanbul Technical University, Graduate School of Science Engineering and Technology, Aerospace Engineering

### PROFESSIONAL EXPERIENCE AND REWARDS:

- 2018-2019 Visiting Researcher, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology.
- 2017-2020 Teaching Assistant, Department of Aeronautical Engineering, Istanbul Technical University.
- 2013-2020 Research Assistant, Controls and Avionics Research Group, ITU Aerospace Research Center.
- 2018-2019 TUBITAK International Doctoral Research Fellowship.
- 2013-2018 TUBITAK Graduate Research Fellowship.
- 2013 Honored by ITU for completing Double Major in 4 years.
- 2013 Aeronautical Engineering Second Best Student Award.
- 2013 Control Engineering Best Student Award.

### PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Baspınar, B.,** Koyuncu, E., 2020. Assessment of Aerial Combat Game via Optimization-Based Receding Horizon Control. *IEEE Access*, 8, 35853-35863.
- **Baspınar, B.,** Balakrishnan, H., Koyuncu, E., 2019. Mission Planning and Control of Multi-Aircraft Systems with Signal Temporal Logic Specifications. *IEEE Access*, 7, 155941-155950.

- **Baspinar, B.,** Koyuncu, E., 2019. Evaluation of Two-vs-One Air Combats Using Hybrid Maneuver-Based Framework and Security Strategy Approach. *Journal of Aeronautics and Space Technologies*, 12(1), 95-107.
- **Baspinar B.,** Koyuncu E., 2019. Differential Flatness-based Optimal Air Combat Maneuver Strategy Generation. *AIAA SciTech Forum and Exposition: Modelling and Simulation Technologies*, January 07-11, 2019 San Diego, California, USA.
- **Baspinar B.,** Koyuncu E., 2018. Aerial Combat Simulation Environment for One-on-One Engagement. *AIAA SciTech Forum and Exposition: Modelling and Simulation Technologies*, January 08-12, 2018 Kissimmee, Florida, USA.
- **Baspinar B.,** Uzun M. et al, 2017. A 4D Trajectory Generation Infrastructure Tool for Controller Working Position. *36th IEEE/AIAA Digital Avionics Systems Conference*, September 17-21, 2017 St. Petersburg, Florida, USA.





## OTHER PUBLICATIONS, PRESENTATIONS AND PATENTS:

- **Baspinar B.**, Koyuncu E., 2018. Survivability based Optimal Air Combat Mission Planning with Reinforcement Learning. *2nd IEEE Conference on Control Technologies and Applications (IEEE CCTA 2018)*, August 21-24, 2018 Copenhagen, Denmark.
- Uzun M., **Baspinar B.**, Koyuncu E., Inalhan G., 2017. Take-off Weight Error Recovery Tool for Tactical Trajectory Prediction Automaton of Air Traffic Control Operator. *36th IEEE/AIAA Digital Avionics Systems Conference*, September 17-21, 2017 St. Petersburg, Florida, USA.
- **Baspinar B.**, Koyuncu E., 2017. Demand and Capacity Balancing Through Probabilistic Queuing Theory and Ground Holding Program for European Air Transportation Network. *Journal of Science and Technology A - Applied Sciences and Engineering*, 18(2), 360-374.
- **Baspinar B.**, Koyuncu E., Inalhan, G., 2017. Large Scale Data-Driven Delay Distribution Models of European Air Traffic Flow Network. *Transportation Research Procedia*, 22, 499-508.
- **Baspinar B.**, Koyuncu E., 2017. Managing Air Transportation Demand and Capacity via Stochastic Modelling Approach. *9th Ankara International Aerospace Conference*, September 20-22, 2017 Ankara, Turkey.
- **Baspinar B.**, Koyuncu E., 2016. A Data-Driven Air Transportation Delay Propagation Model Using Epidemic Process Models. *International Journal of Aerospace Engineering*, DOI: 10.1155/2016/4836260.
- Koyuncu E, **Baspinar B.**, Guney G. et al, 2016. Implementing Dynamic Air Transport Slot Trading Through Secure Auction Mechanism. *7th International Conference on Research in Air Transportation*, June 20-24, 2016 Philadelphia, Pennsylvania, USA.
- **Baspinar B.**, Ure N.K., Koyuncu E., Inalhan G., 2016. Analysis of Delay Characteristics of European Air Traffic through a Data-Driven Airport-Centric Queuing Network Model. *14th IFAC Symposium on Control in Transportation Systems*, May 18-20, 2016 Istanbul, Turkey.
- Guner G., **Baspinar B.**, Koyuncu E., et. al., 2015. A Flight Delay Reporting and Analysis Platform Through Secure Information Sharing. *Fifth SESAR Innovation Days*, December 1-3, 2015 Bologna, Italy.
- Pasaoglu C., **Baspinar B.**, Ure N.K. and Inalhan G., 2015. Hybrid Systems Modelling and Automated Air Traffic Control for Three Dimensional Separation Assurance. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 230(9) 1788-1809.
- **Baspinar B.**, Pasaoglu C., Ure N.K., Inalhan, G., 2015. Infrastructure Development for Ground-Based Separation Assurance with Optional Automation. *International Conference on Application and Theory of Automation in Command and Control Systems*, September 30 - October 2, 2015 Toulouse, France.

- **Baspinar B.**, Pasaoglu C., Inalhan G., 2014. Hybrid Modelling and Automation of Air Traffic Controller Decision Process: Separation Assurance, *6th International Conference on Research in Air Transportation*, May 26-30, 2014 Istanbul, Turkey.
- **Baspinar B.**, Koyuncu E., Inalhan G., 2016. Modelling of Air Traffic Flow Network and Analysis of Delay Characteristics because of Capacity Reductions in Airports. *VI. National Conference on Aeronautics and Astronautics (UHUK'16)*, September 28-30, 2016 Kocaeli, Turkey.
- **Baspinar B.**, Pasaoglu C. et. al., 2014. Hybrid Automata Models for Separation Assurance in Air Traffic Control. *V. National Conference on Aeronautics and Astronautics (UHUK'14)*, September 8-10, 2014 Kayseri, Turkey.
- **Baspinar B.**, Temeltas H., 2013. Navigation with EKF based INS/GPS Integration, *National Automatic Control Committee (TOK'13)*, September 26-28, 2013 Malatya, Turkey.

