AIR TRAFFIC SURVEILLANCE AND CONTROL USING HYBRID ESTIMATION AND PROTOCOL-BASED CONFLICT RESOLUTION

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF
AERONAUTICS AND ASTRONAUTICS
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Inseok Hwang
December 2003

© Copyright by Inseok Hwang 2004
All Rights Reserved

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Claire Tomlin
(Aeronautics and Astronautics)
(Principal Adviser)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Stephen Rock
(Aeronautics and Astronautics)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

David Dill
(Computer Science)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Sanjay Lall
(Aeronautics and Astronautics)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Andrew Ng
(Computer Science)

Approved for the University Committee on Graduate Studies:

Abstract

Continued growth of air travel and recent advances in new technologies for navigation, surveillance, and communication have led to proposals by the Federal Aviation Administration (FAA) to provide reliable and efficient tools to aid Air Traffic Control (ATC) in performing their tasks. The current ATC has a rigid and centralized structure: aircraft fly along predefined airways between waypoints, and ground controllers direct aircraft using radar track and flight information from plan view displays and voice communication over radio channels. New technologies such as the Global Positioning System (GPS) for navigation and Automatic Dependent Surveillance-Broadcast (ADS-B) for communication, will enable automation of some of the ATC functions. These functions include multiple-target tracking and identity management for air traffic surveillance, and conflict detection and resolution between aircraft for air traffic control. In this thesis, we address four problems frequently encountered in air traffic surveillance and control. We develop a set of algorithms and tools with provable properties: these algorithms may be used either in a fully autonomous way, or as supporting tools to increase controllers' situational awareness and to reduce their work load.

Firstly, we develop a multiple-maneuvering-target tracking and identity management (MTIM) algorithm which can keep track of maneuvering aircraft in noisy environments and of their identities. The algorithm uses a modified approximate Joint Probabilistic Data Association algorithm for associating measurements with targets, an Identity Management algorithm based on Identity-Mass Flow, and a multiple-target tracking algorithm based on the Residual-Mean Interacting Multiple Model algorithm for hybrid state estimation. The MTIM algorithm incorporates suitable local information, when available, in a manner that

decreases the uncertainty, as measured by system entropy. In situations in which local information is not explicitly available, a version of local information incorporation based on Multiple Hypothesis Tracking is included to improve identity management.

Secondly, we propose a hybrid probabilistic conflict detection algorithm between multiple aircraft which uses *flight mode estimates* as well as aircraft current state estimates. Our algorithm is based on *hybrid* models of aircraft, which incorporate both continuous dynamics and discrete mode switching. As such, the algorithm can provide advantages over existing conflict detection algorithms which use current state estimates from continuous dynamics only. This algorithm can also be used for blunder detection, that is, the detection of a possible incursion maneuver of another aircraft into the path of the first.

Thirdly, we develop an algorithm for multiple (greater than two) aircraft conflict avoidance. The algorithm is based on a closed-form analytic solution. Heading change is the main control input for conflict resolution, yet velocity change may be also used. Constructing a finite partition of the airspace around the possible conflict and using our analytic solution, we derive a protocol for resolving the worst-case conflict within each partition. The result is a multiple-aircraft conflict resolution protocol: a simple rule which is easily understandable and implementable by all aircraft involved in the conflict, and which provides guarantees of safety. We show that this solution is robust to uncertainties in the aircraft's position, heading, and velocity, as well as to small changes in the aircraft trajectory, and asynchronous maneuvers. We present simulation results using a dynamic aircraft model for various multiple aircraft conflict scenarios derived from actual air traffic data (Enhanced Traffic Management System data).

Finally, we consider the problem of safety verification of control laws for safety critical systems, with application to air traffic control systems. We approach safety verification through reachability analysis, which is a computationally expensive problem. We develop an over-approximate method for reachable set computation using polytopic approximation methods and dynamic optimization. We also successfully solve a two aircraft conflict problem in real time.

Acknowledgments

First and foremost, I would like to thank my advisor, Professor Claire J. Tomlin, for providing me an excellent environment for research and study. Her keen insights and knowledge, enthusiasm, clear guidance, and care for her students have set a wonderful example for me and made it privilege and a pleasure to work with her.

I would like to acknowledge the excellent instruction I have received from all of my professors at Stanford. In particular, I would like to thank my thesis reading committee, comprising of Professor Stephen Rock, Professor David Dill, Professor Sanjay Lall, and Professor Andrew Ng. Their invaluable advice, comments, and support helped me pull together various concepts into a coherent picture and enhanced my insights and knowledge on control systems theory and applications.

It is a pleasure to acknowledge the research collaborations with my colleagues at Stanford. My special thanks go to Hamsa Balakrishnan, with whom I have worked on hybrid estimation, multiple-target tracking and identity management, and conflict detection. I also would like to thank Dr. Dusan Stipanović, with whom I have worked on approximate methods on reachable set computation. They are not only good researchers but also good friends. I have learned a lot from them through invaluable discussions and talks. Multiple-target tracking and identity management was joint work with Kaushik Roy and Jaewon Shin. For conflict detection and resolution, I would like to thank Jaegyom Kim and Jesse Hwang. Observability of discrete event systems and its application to safe interface design was joint work with Meeko Oishi, and predicate abstraction and its application to biological systems was joint work with Ronojoy Ghosh. In addition, I would like to thank my colleagues in the Hybrid Systems Laboratory including Gokhan Inalhan, Rodney Teo, Jung Soon Jang,

and Alex Bayen for their support and discussions. I'm pleased to thank Ray Deutsch and the KAASA members who made my Stanford life rich and happy.

I would like to acknowledge the financial support from DARPA, MURI and NSF, and to thank Stanford University for the Stanford Graduate Fellowship and the Korean government for the Korean Government Scholarship.

Last but not least, my thanks go to my family. I would like to thank my parents for their endless love and support. I also would like to thank my wife Haejeong and my son Myunghwan for always being beside me and encouraging me.

Contents

Abstract **v**

Acknowledgments **vii**

1 Introduction **1**

1.1 Overview 7

I Hybrid Estimation **9**

2 Hybrid Estimation Algorithm **11**

2.1 Properties of Discrete-Time Stochastic Linear Hybrid Systems 16

2.1.1 Observability of discrete-time stochastic linear hybrid systems 16

2.1.2 Exponential convergence condition for hybrid estimation algorithms 23

2.1.3 Aircraft model for tracking 28

2.2 Performance analysis of hybrid estimation algorithms 32

2.2.1 Discrete-time stochastic linear hybrid systems with the Markov discrete state dynamics 32

2.2.2 Multiple Model Adaptive Estimation (MMAE) algorithm 33

2.2.3 Sojourn time analysis for stochastic linear hybrid systems 35
2.2.4 Steady-state and transient analysis for hybrid estimation 37
2.2.5 Performance comparison between hybrid estimation algorithms . . . 44
2.3 The RMIMM Algorithm 49
2.3.1 Example: Tracking a single aircraft 53

II Air Traffic Surveillance and Control 57

3 Multiple-Target Tracking and Identity Management 59

3.1 The MTIM algorithm 62
3.1.1 Data association algorithms 64
3.1.2 Identity management algorithm 68
3.1.3 Incorporation of Local Information 69
3.2 MTIM in an environment without clutter 72
3.2.1 Simulation results 73
3.3 MTIM in clutter 74
3.3.1 State Prediction 76
3.3.2 Measurement Validation/Association 77
3.3.3 State Estimate Update 79
3.3.4 Belief Matrix Update 80
3.3.5 Local information incorporation 80
3.3.6 Simulation results 83

4 Flight-Mode-Based Conflict Detection **89**

4.1 Conflict detection 90

4.1.1 Nominal conflict detection 90

4.1.2 Probabilistic conflict detection 91

4.2 Numerical examples and validation with ETMS data 95

4.2.1 Numerical examples for multiple-aircraft conflict detection 95

4.2.2 Validation with ETMS data 99

4.2.3 Conflict detection using waypoint information 102

5 Protocol-based Conflict Resolution **107**

5.1 Motivation and problem formulation 110

5.2 Safety conditions for multiple-aircraft conflict resolution 113

5.3 Protocols for multiple-aircraft conflict resolution 117

5.3.1 Derivation of a protocol for multiple-aircraft conflict resolution for an exact conflict 117

5.3.2 Derivation of a protocol for multiple-aircraft conflict resolution for an inexact conflict 121

5.4 Flight-mode-based conflict prevention 126

5.4.1 Grouping 128

5.5 Validation with ETMS data 130

III Safety Verification **131**

6 Approx. Comp. of Reach Sets using Polytopic Approx. **133**

6.1 Motivation and Problem Formulation 135

6.2 Computation of polytopic reachable sets 138

6.2.1 Linear dynamical systems 139

6.2.2 Feedback linearizable nonlinear systems 141

6.2.3 Linear dynamic games 142

6.2.4 Norm-bounded nonlinear systems 143

6.3 Examples 144

6.3.1 Linear dynamical systems 144

6.3.2 Norm-bounded nonlinear systems 144

6.3.3 Conflict resolution between two aircraft 145

7 Conclusions and Future Work **151**

A Proof of Lemma 8 in Section 5.2 **155**

B Validation of PBCR with dynamic aircraft model **157**

C Robustness Analysis of PBCR **161**

Bibliography **167**

List of Tables

2.1 Performance comparison between hybrid estimation algorithms: two mode example 45

2.2 Performance comparison between hybrid estimation algorithms: three mode example 45

List of Figures

1.1 Organization of this thesis. 7

2.1 Structure of the Residual-Mean Interacting Multiple Model (RMIMM) algorithm (for two-mode systems). 14

2.2 Illustration of the transition sequence. 25

2.3 (a) Exponential convergence of error. (b) Convergence of error when modes have same dynamics but different noise characteristics. The triangles denote discrete transition times ($\mu = 0.99$, $M_0 = 0$, and $T = 2sec$). 31

2.4 Structure of the IMM algorithm (for two modes) from [1]. 35

2.5 Mode estimates by (a) IMM and (b) MMAE: 100 trial Monte Carlo simulation results. The asterisks denote mode estimation delay. 46

2.6 Estimation error by (a) IMM and (b) MMAE: 100 trial Monte Carlo simulation results. 47

2.7 Mode estimates by (a) IMM and (b) MMAE: 100 trial Monte Carlo simulation results (mode CV = 0, mode CT = 1). 48

2.8 100 times Monte Carlo simulation results using IMM and RMIMM for the aircraft tracking example. (a) Likelihood of each mode. (b) Mode probability of each mode. 53

2.9 A simulation result using IMM for the aircraft tracking example. (a) Likelihood of each mode. (b) Mode probability and mode estimate of each mode. 54

2.10 A simulation result using RMIMM for the same aircraft tracking example. (a) Likelihood of each mode. (b) Mode probability and mode estimate of each mode. 55

3.1 (a) Clutter in raw radar measurements. (b) Estimated trajectories of aircraft from raw radar measurements [2]. 60

3.2 General Multiple-Target Tracking and Identity Management (MTIM) structure. 63

3.3 Validation Gate. 65

3.4 (a) Trajectory comparisons, (b) Belief matrix evolution. 74

3.5 MTIM Block Diagram. 75

3.6 Validation gates determined by the original residual covariance S and the effective residual covariance S' which accounts for the maneuvering uncertainty of a target. 78

3.7 State estimates (x) and measurements (o) for two-aircraft example. 80

3.8 (a)-(d): Possible joint events in MHT. 81

3.9 Measurement points with clutter (top), aircraft trajectories (center) and accompanying belief matrix plot (bottom). 86

3.10 Measurement points with clutter (top), aircraft trajectories (center) and plot of belief information for Aircraft A and B (bottom). 87

4.1 Trajectory prediction error ellipses. (a) Both aircraft are in the constant velocity mode. (b) An aircraft is in constant velocity mode and the other is in the coordinated turn mode. (c) Both aircraft are in the coordinated turn mode, motivated by the single mode computations in [3]. 91

4.2 Transformed conflict geometry [3]. 92

4.3 200 Monte Carlo simulation results using nominal and probabilistic conflict detection algorithms. (a) Both aircraft are in the constant velocity mode. (b) An aircraft is in constant velocity mode and the other is in the coordinated turn mode. (c) Both aircraft are in the coordinated turn mode. 96

4.4 Blunder detection scenario for two aircraft. 98

4.5 Simulations for multiple-aircraft conflict detection. (a) Structured airspace scenario (four aircraft). (b) Free Flight scenario (ten aircraft). 98

4.6 Performance analysis of the flight-mode-based conflict detection algorithm with ETMS data (altitude $31,000ft$). (a) Successful conflict detection probability. (b) False alarm probability. 100

4.7 System operating characteristic curve (Successful conflict detection probability vs. False alarm probability). 101

4.8 Multiple-aircraft conflict detection around waypoints *Mina* and *Clovis*. . . . 102

4.9 (a) Trajectory prediction error ellipses when an aircraft passes a waypoint within the look-ahead time. (b) Geometry of the turning maneuver around a waypoint. 103

4.10 Performance comparison between the conflict detection algorithms using and without using waypoint information. (a) Conflict scenario around waypoint *Clovis*. (b) Successful conflict detection probabilities of the conflict detection algorithms using and without using waypoint information. 104

5.1 An illustration of the exact conflict resolution maneuver for one of the N-aircraft and parameters involved. 111

5.2 Airspace partition. 118

5.3 Protocol-based conflict resolution for three aircraft (nm), exact conflict case. 122

5.4 Protocol-based conflict resolution for ten aircraft (nm), exact conflict case. 123

5.5 Inexact conflict. 124

5.6 Inexact conflict (General case) (nm). 127

5.7 Grouping using conflict prevention algorithm. 128

5.8 Protocol-based conflict resolution with flight-mode-based conflict prevention. 129

5.9 Simulations with ETMS data using FMBCD/PBCR: (a) Structured airspace scenario. (b) Free flight scenario. 130

6.1 Backward reachable set for two aircraft collision avoidance problem from [4]. Courtesy of I. Mitchell. 137

6.2 The forward reachable set of a four dimensional linear dynamical system (projection onto x_3 and x_4). 145

6.3 The forward reachable set of a norm-bounded nonlinear system. 146

6.4 Relative configuration of two aircraft showing the protected zone. 146

6.5 Comparison between over-approximate and exact (Figure 6.1) backward reachable sets (unsafe sets) of conflict resolution between two aircraft: view from (a) Azimuth 80^o and elevation 10^o (b) Azimuth 10^o and elevation -10^o. . . 148

6.6 Conflict scenario: Aircraft 2 reaches the boundary of the unsafe zone of aircraft 1 with a given initial relative angle ψ_r. 149

6.7 Conflict resolution simulation with relative initial states ($x_r = 10, y_r = -20, \psi_r = 115^o$). Aircraft 1 tries to avoid a conflict with aircraft 2 with the optimal strategy. 149

B.1 Validation with a dynamic aircraft model. 159

C.1 Robustness analysis. 163

Chapter 1

Introduction

The Air Traffic Control (ATC) system is responsible for safe air traffic operations of both commercial and general aviation within the nation's airspace. Even though the ATC has managed air traffic with a strong safety record over the past several decades, the system does suffer from the occasional serious accident. Based on projections in the 1990's of the United States and world economic growth, total passenger traffic between the United States and the rest of the world is expected to grow from 132 million in 1999 to 466.8 million in 2025, an increase of more than 250%. Passenger traffic is expected to be strongest in the Latin American and Pacific markets, growing by almost 350% over the forecast period. Passenger traffic is projected to grow 180% in Atlantic markets and 140% in transborder Canadian markets [5]. After September 11, 2001, the air traffic growth rate declined, yet it is now expected to return to the rate of the expected growth forecasted before 9/11 after 2004 [6].

Due to this continued growth of air traffic, airborne delays and ground holds have become common today and will increase rapidly in the near future unless there are changes to the equipment and structure of the current ATC. For example, air traffic control equipment such as the plan view displays used by controllers to look at radar tracks and flight information are based on displays designed in the early 1970s and microprocessors developed in the 1980s. Even if the introduction of new equipment for air traffic control alleviates the problems caused by failing equipment, more automation and structural changes in the ATC system will be required [7]. The pressure for changes in the ATC system originates from

three primary sources: the need for improved safety and efficiency; the availability of new technology; and the desire to support the controllers. Even given the current low accident rate in commercial and general aviation, there remains a need to strive for even greater safety levels.

The current airspace has a rigid, centralized structure, and aircraft are required to follow predefined airways or instructions given by controllers. Due to recent advances in navigation and data communication technologies such as the Global Positioning System (GPS), and a new data link between aircraft and between aircraft and controllers known as Automatic Dependent Surveillance-Broadcast (ADS-B), it may be plausible in the near future for aircraft to fly their own trajectories instead of predefined paths in the National Airspace System (NAS) [8, 9, 7, 10]. The idea of decentralizing some of the ATC functionality to individual aircraft is a part of the *Free Flight* concept, which has been studied by various research groups in the last several years [11, 12, 3, 13, 14, 15, 16]. In Free Flight, each aircraft could optimally choose its own trajectory to minimize fuel consumption and time delay. This requires clear and unambiguous methods for maintaining safe separation between aircraft. Therefore, conflict detection and resolution (CD&R) emerges as a critical issue for the implementation of Free Flight.

In this thesis, we consider the air traffic surveillance and control problems of multiple-aircraft tracking in clutter, and conflict detection and resolution, and propose tools to aid ATC, and thus potentially improve the efficiency as well as safety of the current system.

In air traffic control, accurate tracking (state estimation) of aircraft is important because all traffic advisories are based on the aircraft's current state estimates. However, even tracking a single maneuvering aircraft is not a trivial problem. Not only are the continuous state measurements noisy, the aircraft may also be switching flight modes; either changing from straight flight to turning mode, or vice versa. In reality, precision tracking of the continuous state estimate when the aircraft makes an unanticipated change of flight mode is a difficult task. The challenge in aircraft tracking thus lies in the ability to provide high quality state estimates of the aircraft in such situations. The current algorithms in use for ATC tracking are based on constant gain Kalman filters, known as $\alpha - \beta$ or $\alpha - \beta - \gamma$ filters [17, 18]. However, a single Kalman filter tuned to a flight mode such as a straight flight mode cannot provide good estimates when an aircraft changes its flight mode from a straight flight mode to a coordinated turn mode. The trajectory of an aircraft is composed of straight lines

and circular arcs that represent the flight modes of the aircraft and its flight mode changes are dependent on the pilot's (or autopilot's) input. However, this input is unknown to the surveillance system in target tracking problems. The way a pilot controls an aircraft is generally restricted to a choice of a control input from a finite set of possible control inputs, including inputs such as left turn, right turn, speed up and speed down. The control input selected by the pilot determines the flight mode of the aircraft. Thus, the dynamics of an aircraft can be easily modelled as a hybrid system which has a finite set of discrete states corresponding to the pilot's input, that is, the flight mode of the aircraft; and each discrete state has its own continuous dynamics that describes the behavior of the aircraft for a given pilot's input. The surveillance system does not know the pilot's intention, that is, when the pilot changes the control input or what control input the pilot chooses. Thus, the unknown pilot's input (maneuver command or flight mode change command) can be modelled as a random process which selects a control input from a finite set of possible control inputs at any time. Therefore, we model the dynamics of an aircraft for target tracking as a stochastic linear hybrid system. A stochastic linear hybrid system has discrete states and each discrete state has its own continuous dynamics. A transition between the discrete states is governed by a finite Markov chain which models the unknown pilot's input, that is, flight mode changes of an aircraft. Therefore, a hybrid estimation algorithm for such stochastic linear hybrid systems could provide more accurate state estimates than a single Kalman filter. It also gives additional information in the form of discrete state estimates (flight mode estimates), which could be useful for other applications such as maneuvering-target tracking and conflict detection.

In order to derive a new hybrid estimation algorithm which gives accurate hybrid state estimates (both the continuous and discrete state estimates), we first derive observability conditions for discrete-time stochastic linear hybrid systems and analyze the performance of hybrid estimation algorithms for such observable hybrid systems. Based on the analysis results, we propose a hybrid state estimation algorithm called the *Residual-Mean Interacting Multiple Model* (RMIMM) algorithm [19], which is a modified version of the Interacting Multiple Model (IMM) algorithm [20]. The RMIMM could provide more accurate state estimates than the IMM. The algorithms and tools for air traffic surveillance and control developed in this thesis are based on the RMIMM algorithm.

Firstly, we consider air traffic surveillance problems, that is, tracking multiple aircraft in clutter and managing their identities at the same time. The current ATC system is based on

a surveillance system that uses data from radar measurements to track aircraft. In the early versions of this system, human operators manually followed the blips on the video displays, but the increasing number of aircraft necessitated the development of automated tracking algorithms. In spite of a substantial improvement in technology, the radar system is still vulnerable to several problems, such as extraneous measurements from clouds, birds and other objects, as well as "phantom" blips [17, 21], all made worse with larger numbers of aircraft. Another issue that poses danger is the growing number of general aviation aircraft. These aircraft do not transmit their identities unless their transponders are switched on, and even then, the transponders can be fraught with problems [22]. Since air traffic controllers are instructed not to issue orders to aircraft unless they are certain of their identity [23], it becomes essential that controllers have access to reliable track data with identities, so that they can maintain safety. Thus, we propose a *Multiple-Target Tracking and Identity Management* (MTIM) algorithm, which can keep track of multiple-aircraft in clutter and of their identities at the same time. The applications of multiple-target tracking are: land, sea, air and space surveillance for military uses; and collision avoidance, navigation, and image processing for civilian uses. There are several issues that need to be tackled while solving this problem. For tracking a maneuvering aircraft, we use the RMIMM algorithm to get accurate state estimates of aircraft. The next level of difficulty is in simply tracking multiple targets, given their measurements. This problem has been addressed by several data association algorithms, which associate measurement data with targets [17]. In addition, [24] developed an identity-management algorithm for sensor networks, but did not use the knowledge of system dynamics to improve their belief in the identity of the different targets. In this thesis, we propose a combination of the modified approximate Joint Probabilistic Data Association (MAJPDA) algorithm and the Identity Management (IM) algorithm for sensor networks [24], and utilize target attribute information from local sensors to maintain the target's identity correctly, when there are extraneous measurements or "phantom" blips in the measurements. We refer to this algorithm as the Multiple-Target Tracking and Identity Management (MTIM) algorithm.

Secondly, we consider conflict detection problems and propose a new conflict detection algorithm, which would be relevant in a decentralized air traffic control environment. To the best of our knowledge, all existing conflict detection algorithms are based on the continuous state information of the aircraft (see Kuchar and Yang [11] for a comprehensive survey). The performance of these algorithms depends strongly on the accuracy of state estimates.

Since we develop a hybrid estimation algorithm which provides accurate state estimates, we propose a hybrid probabilistic conflict detection algorithm which uses both the continuous and discrete state estimates of aircraft, as computed by the RMIMM algorithm. This conflict detection algorithm can be applicable to both ground and airborne control scenarios in ATC – to ground control using radar information, and airborne control using information from ADS-B data [25]. We validate the proposed conflict detection algorithm with real air traffic data called Enhanced Traffic Management System (ETMS) data in the Oakland Center airspace.

Thirdly, we consider multiple-aircraft conflict resolution problems. Many previous algorithms [11] either consider two aircraft only, or are based on solving optimization schemes which are not guaranteed to converge in real time. It is easy to show, for example [26], that the successive application of pairwise resolution cannot guarantee safety for a multiple-aircraft conflict. In this thesis, we consider 2D conflict resolution; aircraft are assumed to fly at the same altitude with varying velocities. An aircraft's position, velocity, and heading are assumed to be available to all other aircraft which are involved in the conflict; this assumption can be justified with the proposed availability of GPS [9] and a new datalink ADS-B [25]. We propose a new algorithm denoted as the *Protocol-Based Conflict Resolution* (PBCR) algorithm for multiple-aircraft conflict resolution. While other rule-based algorithms require as many rules as possible conflict cases, the proposed algorithm is simple and easily understandable since the protocol is obtained from a closed-form analytic solution. Thus, it can be implemented in airborne systems for real-time conflict resolution, as well as in ATC ground systems. Most of all, though our method may not be optimal in the sense that some conflicts can be resolved with smaller deviations from the desired trajectory than the solution produced by our protocol, the proposed algorithm is *provably-safe* within the limits of the model used. This is the main difference from many other currently available multiple aircraft conflict resolution methods. We show that the solution is robust to uncertainties in the aircraft's position, velocity, and heading, and that the method remains safe when sharp corners on the resolution trajectories are replaced by circular arcs. Though a kinematic model is used to design the protocol, a dynamic model is used for validation and to simulate real situations. Finally, we validate our conflict detection and resolution algorithms using real air traffic data (ETMS data).

Finally, we consider safety verification problems for air traffic control. It has been shown that reachability analysis for continuous and hybrid systems is important for the automatic

verification of safety properties and for the synthesis of safe controllers for these systems [27, 28]. Convergent approximations of reachable sets for such systems can be computed by solving the exact Hamilton-Jacobi partial differential equation (PDE). Numerical methods have been devised to do this [29, 4] which work well in up to three continuous variable dimensions, yet these methods are not practical for solving high dimensional problems. Therefore, we propose to use polytopic approximation for computing reachable sets for real time computation. To do this, we use the method proposed by Varaiya [30] to compute reachable sets for linear time invariant systems using dynamic optimization. Inspired by Kurzhanski and Varaiya [31, 32, 33, 34], we then extend it to more general dynamical systems such as feedback linearizable nonlinear systems, linear dynamic games, and norm-bounded nonlinear systems.

The contributions of this thesis are thus summarized as follows: First, we derive observability conditions for discrete-time stochastic linear hybrid systems [35] and analyze the performance of hybrid estimation algorithms for those observable stochastic linear hybrid systems [36]. Using the performance analysis results, we develop a new hybrid estimation algorithm called the **Residual-Mean Interacting Multiple Model (RMIMM)** algorithm based on a multiple-model-based state estimation algorithm called the Interacting Multiple Model (IMM) algorithm, which performs better than the IMM [19]. Next, we develop a **Multiple-Target Tracking and Identity Management (MTIM)** algorithm [37] which can keep track of multiple targets (aircraft) in clutter and also maintain a record of their identities at the same time, so that it can not only improve controllers' situational awareness but can also significantly reduce their workload. We propose a **Flight-Mode-Based Conflict Detection (FMBCD)** algorithm for conflict detection [19]. This algorithm uses not only continuous state estimates but also flight mode estimates in conflict detection, and can therefore detect conflicts with good accuracy, especially in congested airspace around airports and waypoints. We develop a **Protocol-Based Conflict Resolution (PBCR)** algorithm for multiple-aircraft conflict resolution [38, 39]. The resulting algorithm is simple to understand and implement, and is provably-safe to within the limits of the model used. Finally, for safety verification and safe controller synthesis, we use polytopic approximations, based on optimal control theory, to compute over-approximate reachable sets for linear systems as well as feedback linearizable nonlinear systems, linear dynamic games, and norm-bounded nonlinear systems [40, 41, 42].

A detailed literature review for each topic can be found in the corresponding chapter.

1.1 Overview

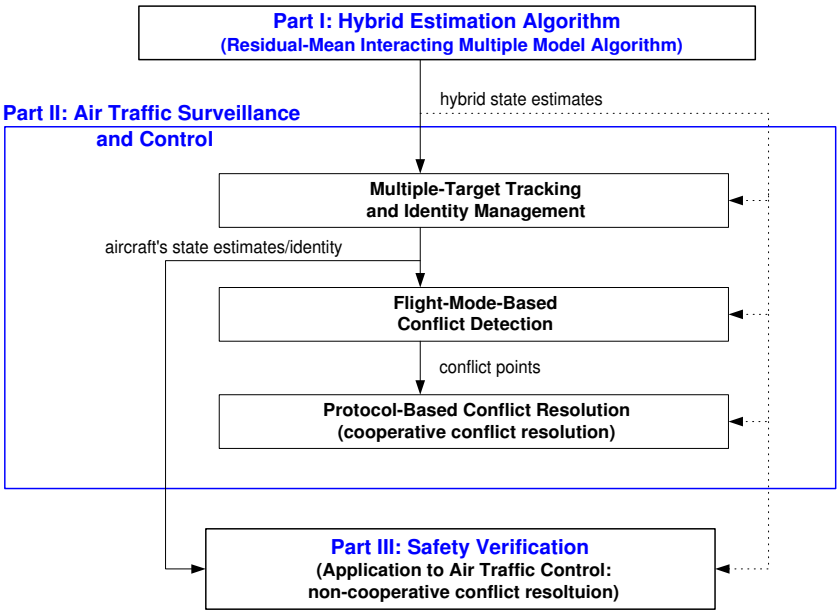


Figure 1.1: Organization of this thesis.

This thesis is composed of three main parts as shown in Figure 1.1: hybrid estimation, air traffic surveillance and control, and safety verification. In the hybrid estimation part, we derive observability conditions for discrete-time stochastic hybrid systems and analyze the performance of hybrid estimation algorithms for those observable discrete-time stochastic hybrid systems. Based on the analysis results, a new hybrid estimation algorithm which is a modified version of the Interacting Multiple Model (IMM) algorithm [20] called the Residual-Mean Interacting Multiple Model (RMIMM) is developed, which can provide more accurate estimates of both the continuous and discrete states than the IMM. We will use this hybrid estimation algorithm for multiple-target tracking for air traffic surveillance and conflict detection and resolution for air traffic control.

The air traffic surveillance and control part is composed of three chapters: Multiple-Target Tracking and identity management (MTIM), Flight-Mode-Based Conflict Detection (FMBCD), and Protocol-Based Conflict Resolution (PBCR). Chapter 3 presents the MTIM algorithm which can keep track of multiple targets (in this thesis, aircraft) in clutter using the RMIMM as a state estimator, and also maintain belief of their identities at the same

time. Thus, both the accurate continuous and discrete state estimates of aircraft, as well as their identities, are available for conflict detection and resolution in air traffic control. In Chapter 4, we develop a probabilistic conflict detection algorithm using the aircraft's continuous state estimates as well as discrete state (flight mode) estimates computed by the RMIMM. Chapter 5 presents a protocol-based conflict detection algorithm which is based on a closed-form analytic solution and thus can be implemented for both on-board real-time applications and ground control applications.

In the safety verification part, we approach the safety verification problems of dynamical systems through reachability analysis and develop an over-approximate computation method of reachable sets of the given systems using polytopic approximations. Since the evolution of the faces of the polytope is based on an analytic equation, our method can be implemented in real-time applications. Our method always guarantees an over-approximation of the exact reachable set, and thus if the system satisfies the over-approximate polytopic constraints computed by our algorithm, it is guaranteed to be safe.

Part I

Hybrid Estimation

Chapter 2

Hybrid Estimation Algorithm

For air traffic surveillance and control applications such as multiple-target tracking, and conflict detection and resolution, we need a state estimation algorithm which can estimate the current states of an aircraft accurately. However, state estimation of a maneuvering aircraft is difficult when the aircraft changes its flight mode unexpectedly. This is because a single linear (or nonlinear) dynamic model with constant system parameters cannot accurately represent the behavior of an aircraft over all of its flight regime. Since the performance of a Kalman filter (or extended Kalman filter, respectively) is strongly dependent on the accuracy of the model used, it might not provide accurate state estimates for the aircraft when the actual dynamics of the aircraft deviates from the valid region of the model used. For example, if a single linear (or nonlinear) continuous model is used for aircraft tracking, the process noise covariance in the model has to be large in order to account for model inaccuracy. This large process noise covariance leads to poor state estimates. Hybrid models with multiple modes (discrete states) that represent the flight regimes of an aircraft could represent the dynamics of the aircraft more accurately than one continuous model, and thus each continuous model could have a small process noise covariance that would give accurate state estimates. The flight mode changes of an aircraft depend on the pilot's input, and in the tracking problems we will consider, this input is usually unknown to the surveillance system. This unknown pilot's input makes the flight mode changes of an aircraft nondeterministic because we cannot tell *a priori* when the pilot changes the control input and what control input the pilot takes. Thus, the unknown pilot's input (flight mode change) can be

modelled as a random process. In this thesis, we model the dynamics of an aircraft as a discrete-time stochastic linear hybrid system whose discrete states (or modes) correspond to the flight modes of the aircraft. The flight mode logic of the aircraft is represented by the discrete-state dynamics governed by a finite Markov chain. Each discrete state has its own discrete-time continuous linear dynamics that describes the motion of the aircraft in the corresponding flight mode. In this way, a discrete-time stochastic linear hybrid system may be used to represent the behavior of an aircraft over all of its flight regime. In this chapter, we develop a new hybrid estimation algorithm, called the Residual-Mean Interacting Multiple Model (RMIMM) algorithm (as shown in Figure 2.1), which provides accurate discrete and continuous state estimates of discrete-time stochastic linear hybrid systems.

Before deriving the RMIMM algorithm, we address the following problem about hybrid estimation: Under what conditions can the hybrid state of the given discrete-time stochastic linear hybrid system be reconstructed uniquely from an output data sequence? In order to answer this question, we derive conditions for the observability of discrete-time stochastic linear hybrid systems by exploiting the information obtained from system noise characteristics. We then study performance analysis of hybrid estimation algorithms, and derive conditions under which the state estimation error of hybrid estimation algorithms converges exponentially to zero.

The observability of continuous systems [43] or discrete systems [44, 45], leading to the full state reconstruction from the output signal is a well-known problem and has been studied intensively. However, the observability of hybrid systems has only recently been investigated. For deterministic linear hybrid systems, Alessandri and Coletta [46] propose a Luenberger observer design method, which guarantees that the estimation error converges if the discrete state evolution is known. Balluchi and Benvenuti [47] suggest a methodology for designing a location observer for discrete state estimation as well as a Luenberger observer for continuous state estimation and also provide a condition under which the estimation error converges exponentially. Vidal *et al.* [48] derive observability conditions for continuous-time linear hybrid systems, which are rank conditions similar in form to those for the observability of continuous-time linear systems. Bemporad *et al.* [49] propose the concept of incremental observability of linear hybrid systems. The algorithm requires the solution of a mixed-integer linear program. Baram and Kailath [50] propose the concept of estimability as a criterion to gauge stochastic linear systems. For stochastic linear hybrid systems, Costa

and do Val [51] present a new observability concept for discrete-time Markov jump linear systems with finite Markov states with the observability condition that the solution to the coupled algebraic Riccati equation associated to the quadratic control problem is a stabilizing solution. Estimation algorithms for discrete-time Markov jump linear systems have been widely used for various applications such as multiple-target tracking [52] and speech recognition [53]. Vidal *et al.* [54] propose the notion of *indistinguishability*: a hybrid system is indistinguishable if two different initial conditions give the same output sequence. The observability conditions for jump linear systems are based on rank tests, similar to those for deterministic linear hybrid systems. Since output sequences of stochastic systems could be different from the same initial condition, a new definition of indistinguishability is proposed, and based on this definition, we derive more general conditions for the observability of discrete-time stochastic linear hybrid system than those proposed in [54], thus enabling us to treat a larger subclass of these systems.

We then consider the performance of hybrid estimation algorithms for such observable discrete-time stochastic linear hybrid systems. The problem of being able to estimate both the discrete and continuous states of a hybrid system given only the continuous output sequence is a difficult one, and while algorithms [55, 56] exist for this purpose, little has been proved on the limitations of these algorithms, or even the dependence of their performance on system parameters (even though the performance of hybrid estimators has been studied for several decades). Magill [57] provides sufficient conditions for the convergence of the adaptive weights in a multiple model adaptive estimation algorithm of a specific class of systems, in which a constant parameter vector is unknown and there is a single output over all the modes. Lainiotis *et al.* [58, 59, 60] extend the results in [57] to multiple outputs and derive the recursive form of the optimal adaptive estimator as well as its exact error covariance. Hawkes *et al.* [61] examine the asymptotic behavior of the adaptive weights in hybrid estimation algorithms, which in turn determines the performance of the hybrid estimators. The authors show, using the Kullback information function [62], that the weight corresponding to the true model converges almost surely to unity and that the other weights converge to zero. Many other approaches to the performance analysis of adaptive estimation can be found in the references cited in [61]. Baram *et al.* [63] provide conditions under which, for a set of systems driven by stationary white Gaussian inputs and no discrete transitions, the mode probability of the true model converges to unity, that is, the probability that the estimated model is the true model converges asymptotically to unity. Baram [64] shows

that for the hybrid system in [63], the uniqueness of the prediction error covariance matrix is a sufficient condition for the true model to be estimated asymptotically. However, the system in [63, 64] is a set of stochastic, stationary Gaussian models which do not interact with each other. Thus, the conditions in [63, 64] are more relevant to the observability of stochastic linear hybrid systems [35]. Caputi [65] derives a necessary condition for the performance of hybrid estimation algorithms through the analysis of steady state residuals and shows that the performance of the hybrid estimator depends on the DC gains of the linear systems. This condition is only valid for a specific class of hybrid systems in which the continuous dynamics for all the modes is the same, but the inputs are distinct and consist of a constant bias vector and zero-mean white Gaussian random noise. The research summarized above analyzes hybrid estimation in several special classes of systems, yet general analytical techniques for the performance of hybrid estimation algorithms have not been investigated in detail. Maybeck [55] gives qualitative reasons for the performance of hybrid estimators but adds that no rigorous general proofs are available for the (asymptotic convergence) properties of the hypothesis conditional probabilities.

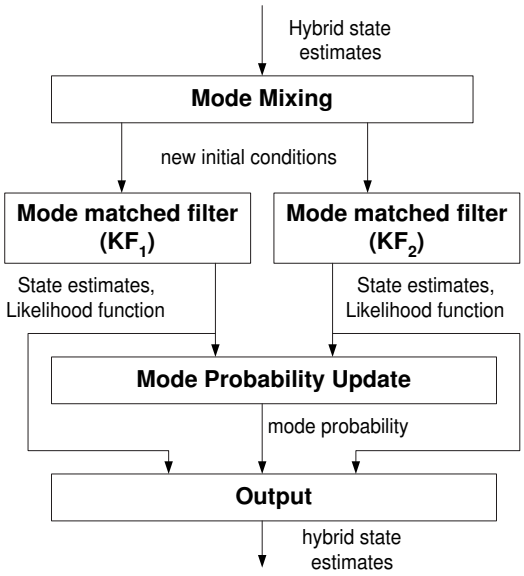


Figure 2.1: Structure of the Residual-Mean Interacting Multiple Model (RMIMM) algorithm (for two-mode systems).

In this chapter, we first derive observability conditions for stochastic linear hybrid systems. We then analyze the properties of hybrid estimation algorithms and derive conditions under

which the computed hybrid estimates converge exponentially to the exact hybrid states. The results of this analysis give some insight into which mode transitions are more detectable than others and also into how to improve the performance of hybrid estimators. (We say that a mode transition is *more detectable* than another if the time taken for the mode estimate to converge to the true mode is less for the former transition than for the latter).

From performance analysis, we show that the mean of the residual computed by an individual Kalman filter in the hybrid estimation algorithm is not zero. Using this fact, we derive the RMIMM algorithm which provides more accurate mode estimates and thus more accurate continuous state estimates than previous algorithms. The RMIMM algorithm has four main blocks as shown in Figure 2.1. The *Mode Mixing* block computes new initial conditions for each mode matched filter (Kalman filter in this thesis) using a weighted sum of the state estimates from the previous time step. The optimal hybrid estimator which minimizes the mean-square estimation error has to keep track of all the mode histories up to the current time, and the number of such histories grows exponentially with time. The optimal estimator is therefore impossible to implement in practice. The RMIMM algorithm is a suboptimal algorithm which, at each time step, keeps information from N^2 mode histories (where N is the number of modes) in just N mode histories by mixing the N mode histories of the previous time step into inputs to the estimators that produce the N mode histories of the current time step. Its complexity involves just N discrete histories, but its performance is close to that of more complex algorithms that keep N^2 mode histories [56]. Then each Kalman filter computes its own state estimates and likelihood function that denotes how likely the model used in the Kalman filter is the correct one. The *Mode Probability Update* block computes the mode probability of each mode using the likelihood functions computed by the individual Kalman filter. The *Output* block computes both the continuous and mode (discrete state) estimates. The mode estimate is defined to be the mode which has the maximum mode probability.

This chapter is organized as follows: Section 2.1 presents observability conditions for stochastic linear hybrid systems. In Section 2.2, we derive conditions under which the state estimation error of hybrid estimation algorithms converges exponentially. Section 2.3 presents the Residual-Mean Interacting Multiple Model algorithm.

2.1 Properties of Discrete-Time Stochastic Linear Hybrid Systems

In this section, we extend the concepts of indistinguishability, observability of the hybrid initial state, and discrete transition times as defined in [54] and derive an observability condition for discrete-time stochastic linear hybrid systems using the knowledge of noise covariances.

2.1.1 Observability of discrete-time stochastic linear hybrid systems

We consider a discrete-time stochastic linear hybrid system

$$H_d : \begin{cases} x(k+1) &= A(q(k))x(k) + B(q(k))u(k) + w_k(q_k) \\ z(k) &= C(q(k))x(k) + v_k(q(k)) \\ q(k+1) &= \delta(q(k), \gamma(k)) \end{cases}, \; k \in \{0, 1, \cdots\} \tag{2.1}$$

where k is a non-negative integer ($k \in \mathbb{N}$); $x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^l$ and $z(k) \in \mathbb{R}^p$ are the continuous state, continuous control input, and output variables respectively; $q(k) \in \{1, 2, \cdots, N\}$ is the discrete state, $\gamma(k) \in \{\gamma^1, \cdots, \gamma^m\}$ is a discrete control input, and $\delta(\cdot, \cdot)$ is a deterministic discrete transition relation which governs the discrete state evolution. We assume the event time at which a discrete transition occurs is unknown. The system parameters $A(q(k)) \in \mathbb{R}^{n \times n}$, $B(q(k)) \in \mathbb{R}^{n \times l}$, and $C(q(k)) \in \mathbb{R}^{p \times n}$ for $q(k) \in \{1, 2, \cdots, N\}$ are real matrices. We assume that the initial state $x(k_0)$ is a white Gaussian random variable with an unknown mean and a known covariance $E[x(k_0)x(k_0)^T] = \pi_0$, and that the process noise $w_k(q(k))$ and the measurement noise $v_k(q(k))$ are uncorrelated, zero-mean white Gaussian sequences with the covariance matrices $E[w_k(q(k))w_k(q(k))^T] = Q(q(k))$ and $E[v_k(q(k))v_k(q(k))^T] = R(q(k))$ respectively. These random sequences are assumed to be uncorrelated with the initial state, i.e., $E[x(k_0)w_k(q(k))^T] = E[x(k_0)v_k(q(k))^T] = 0$. Here, system parameters ($A(q(k))$, $B(q(k))$ and $C(q(k))$), noise covariances ($Q(q(k))$ and $R(q(k))$), and the covariance of the initial continuous state (π_0) are known, while the mean of the initial continuous state and the discrete transition times are unknown. Since the state evolution of a hybrid system has continuous trajectories as well as discrete jumps, we define a hybrid time trajectory:

Definition 1. *(Hybrid time trajectory) A hybrid time trajectory is a sequence of intervals* $[k_0, k_1-1][k_1, k_2-1]\cdots[k_i, k_{i+1}-1]\cdots$ *where* k_i $(i \geq 1)$ *is the time at which the* i*-th discrete state transition occurs.*

Before deriving the observability conditions, we review the definition of observability for discrete-time stochastic linear hybrid systems [54]:

Definition 2. *(Observability of discrete-time stochastic linear hybrid systems) A discrete-time stochastic linear hybrid system* H_d *is observable on* $[k_0, k_0 + K]$ *if the hybrid state* $(q(k), x(k))$ *for* $k \in [k_0, k_0 + K]$ *is uniquely determined (in the case of* $x(k)$*, meaning it converges to a unique probability distribution in the sense of the minimum mean-square error), from the output sequence* $\mathcal{Z}_K = [z(k_0)^T \cdots z(k_0 + K)^T]^T$*, where* $K \in \mathbb{N}$*.*

For stochastic systems, different realizations of the system could have different means and covariances, but the distribution of a set of realizations of the system converges to a unique solution in the minimum mean-square error sense as the number of realizations goes to infinity. Vidal et al. [54] developed rank tests for the observability of stochastic jump linear systems using the notion of indistinguishability. Since we know the noise covariances as well as the system dynamics for a stochastic system, we use this additional knowledge to obtain a more general condition than that in [54]. Since the output sequences of stochastic systems could be different from the same initial condition, we extend the notion of indistinguishability in [54] as follows:

Definition 3. *(Indistinguishability of discrete-time stochastic linear hybrid systems) A discrete-time stochastic linear hybrid system* H_d *is indistinguishable on* $[k_0, k_0 + K]$ *if there exist output sequences* $\mathcal{Z}_K$ *and* $\mathcal{Z}'_K$ *on* $k \in [k_0, k_0+K]$ *starting from any two different hybrid states* $(q(k_0), x(k_0))$ *and* $(q(k_0)', x(k_0)')$*, which are identically distributed.*

Without loss of generality, we consider stochastic linear hybrid systems which do not have a continuous (deterministic) control input, to derive observability conditions for discrete-time stochastic linear hybrid systems. The presence of a known control input would not affect observability of the system. We derive observability conditions for discrete-time stochastic linear systems in two steps: observability of the initial state and observability of discrete transition times.

2.1.1.1 Observability of the initial state

Using a procedure similar to that in [54], we derive the conditions under which the hybrid initial state $(q(k_0), x(k_0))$ can be uniquely determined from the output sequence on $[k_0, k_1 - 1]$ $(k_1 - 1 \leq k_0 + K)$, i.e., before the first discrete transition occurs. We define $t_i = k_{i+1} - k_i$ $(i \geq 0)$ as the sojourn time, which denotes how long the system stays in a discrete state after the i-th discrete transition. Based on **Definition 2** and **Definition 3**, we get the following lemma:

Lemma 1. *The hybrid initial state of a discrete-time stochastic linear hybrid system H_d is observable if and only if it is distinguishable.*

In order to check if the hybrid initial state is distinguishable, we need to compare the distributions of the output sequences. In the case of a linear system with Gaussian noise, this task is greatly simplified, since the distributions are uniquely determined by their means and covariances. A difference in the mean between two output sequences reduces to observability in the corresponding deterministic case, in which the noise terms in the equations are removed, and we can test this in several ways; in particular, [54] presents elegant rank conditions for this purpose, henceforth referred to as the Vidal-Chiuso-Soatto (VCS) conditions.

Lemma 2. *The hybrid initial state of a discrete-time linear stochastic hybrid system is observable if the corresponding deterministic system is observable, i.e, the VCS conditions are satisfied.*

If the VCS conditions are not satisfied, the hybrid system may still be observable, by **Definition 2**; we can compute the covariances (higher moments) of the output sequence $\mathcal{Z}_{t_0}$ on $[k_0, k_1 - 1]$. The output sequence starting from the hybrid initial state $(q(k_0), x(k_0))$ on $[k_0, k_1 - 1]$ is

$$\mathcal{Z}_{t_0}(q(k_0)) = \mathcal{O}_{t_0}(q(k_0))x(k_0) + \mathcal{T}_{t_0}(q(k_0))W_{t_0}(q(k_0)) + V_{t_0}(q(k_0)) \tag{2.2}$$

where

$$\begin{aligned}
\mathcal{O}_{t_0}(q(k_0)) &= [C(q(k_0))^T (C(q(k_0))A(q(k_0)))^T \cdots ((C(q(k_0))A(q(k_0)))^{k_1-1})^T]^T \\
\mathcal{T}_{t_0}(q(k_0)) &= \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ C(q(k_0)) & 0 & 0 & \cdots & 0 \\ C(q(k_0))A(q(k_0)) & C(q(k_0)) & 0 & \cdots & 0 \\ \vdots & & & & \\ C(q(k_0))A(q(k_0))^{k_1-k_0-2} & C(q(k_0))A(q(k_0))^{k_1-k_0-3} & \cdots & C(q(k_0)) & 0 \end{pmatrix}
\end{aligned}$$

$$\begin{aligned}
W_{t_0}(q(k_0)) &= [w_{k_0}(q(k_0))^T w_{k_0+1}(q(k_0))^T \cdots w_{k_1-1}(q(k_0))^T]^T \\
V_{t_0}(q(k_0)) &= [v_{k_0}(q(k_0))^T v_{k_0+1}(q(k_0))^T \cdots v_{k_1-1}(q(k_0))^T]^T
\end{aligned}$$

$\mathcal{O}_{t_i}(q(k_i)) \in \mathbb{R}^{pt_i \times n}$ is the extended observability matrix for a linear system $(A(q(k_i)), C(q(k_i)))$ [54] and $\mathcal{T}_{t_0}(q(k_0))$ is a Toeplitz matrix.

If $rank[\mathcal{O}_{t_0}(q(k_0))] = n$, i.e., the linear system $(A(q(k_0)), C(q(k_0)))$ is observable and $t_0 \geq n$, then the minimum mean-square error solution (which we denote by $\hat{x}_{k_0}(q(k_0))$) to (2.2) can be determined uniquely.

$$\begin{aligned}
\hat{x}_{k_0}(q(k_0)) &= \mathcal{O}^{\dagger}_{t_0}(q(k_0))\mathcal{Z}_{t_0}(q(k_0)) \\
&= x(k_0) + \mathcal{O}^{\dagger}_{t_0}(q(k_0))\mathcal{T}_{t_0}(q(k_0))W_{t_0}(q(k_0)) + \mathcal{O}^{\dagger}_{t_0}(q(k_0))V_{t_0}(q(k_0))
\end{aligned} \tag{2.3}$$

where $\mathcal{O}^{\dagger}_{t_0}(q(k_0)) = (\mathcal{O}^T_{t_0}(q(k_0))\mathcal{O}_{t_0}(q(k_0)))^{-1}\mathcal{O}^T_{t_0}(q(k_0))$. The last two terms represent the estimation error due to the process and the measurement noise.

Similarly, the output from another initial state $(q'(k_0), x'(k_0))$ before the first discrete transition is

$$\mathcal{Z}_{t_0}(q'(k_0)) = \mathcal{O}_{t_0}(q'(k_0))x'(k_0) + \mathcal{T}_{t_0}(q'(k_0))W_{t_0}(q'(k_0)) + V_{t_0}(q'(k_0)) \tag{2.4}$$

From **Lemma 1**, in order that the initial state of a discrete-time stochastic linear hybrid system be observable, it should be distinguishable, i.e., if **Lemma 2** is not satisfied, the covariances of $\mathcal{Z}_{t_0}(q(k_0))$,

$$\begin{aligned}
E[\mathcal{Z}_{t_0}(q(k_0))\mathcal{Z}_{t_0}(q(k_0))^T] &= \mathcal{O}_{t_0}(q(k_0))\pi_0\mathcal{O}_{t_0}(q(k_0))^T \\
&\quad + \mathcal{T}_{t_0}(q(k_0))Q(q(k_0))\mathcal{T}^T_{t_0}(q(k_0)) + R(q(k_0))
\end{aligned} \tag{2.5}$$

and the covariance of $\mathcal{Z}_{t_0}(q(k_0)')$,

$$\begin{aligned} E[\mathcal{Z}_{t_0}(q'(k_0))\mathcal{Z}_{t_0}(q'(k_0))^T] &= \mathcal{O}_{t_0}(q'(k_0))\pi_0\mathcal{O}_{t_0}(q'(k_0))^T \\ &\quad +\mathcal{T}_{t_0}(q'(k_0))Q(q'(k_0))\mathcal{T}_{t_0}^T(q'(k_0)) + R(q'(k_0)) \end{aligned} \tag{2.6}$$

should be different for any distinct $(q(k_0), x(k_0))$ and $(q'(k_0), x'(k_0))$. Then, the discrete initial state can be uniquely determined from the covariance of the output sequence, and the continuous initial state can also be uniquely determined using (2.3). Since the condition $t_0 \geq n$ for observability of the linear system $(A(q(k_0)), C(q(k_0)))$ is conservative, we relax this condition by introducing $\tau_{q(k)}$, the minimum integer which satisfies $rank[\mathcal{O}_{\tau_{q(k)}}(q(k))] = n$ $(\forall q(k) \in \{1, 2, \cdots, N\})$, and $\bar{\tau} = \max_{q(k)} \tau_{q(k)}$ (similar to the joint observability index used in [54]). Then, we have the following condition for the observability of the hybrid initial state:

Lemma 3. *(Observability of the hybrid initial state) If $(A(q(k)), C(q(k)))$ are observable for each $q(k) \in \{1, \cdots, N\}$ and $t_0 \geq \bar{\tau}$, the hybrid initial state $(q(k_0), x(k_0))$ is observable if and only if either the hybrid initial state of the corresponding deterministic system is observable, or*

$$\begin{aligned} & \mathcal{O}_{\bar{\tau}}(q(k_0))\pi_0\mathcal{O}_{\bar{\tau}}(q(k_0))^T + \mathcal{T}_{\bar{\tau}}(q(k_0))Q(q(k_0))\mathcal{T}_{\bar{\tau}}^T(q(k_0)) + R(q(k_0)) \\ \neq\ & \mathcal{O}_{\bar{\tau}}(q'(k_0))\pi_0\mathcal{O}_{\bar{\tau}}(q'(k_0))^T + \mathcal{T}_{\bar{\tau}}(q'(k_0))Q(q'(k_0))\mathcal{T}_{\bar{\tau}}^T(q'(k_0)) + R(q'(k_0)) \end{aligned}$$

for all $q(k_0) \neq q'(k_0) \in \{1, \cdots, N\}$.

We show through a following simple example how a noise free unobservable discrete-time linear hybrid system may be rendered observable, if each discrete state is endowed with different measurement noise covariances.

Example: Consider a discrete-time linear hybrid system with two discrete states

$$q_1 : \begin{cases} x(k+1) = x(k) \\ z(k) = c_1 x(k) + v_1 \end{cases} \qquad q_2 : \begin{cases} x(k+1) = x(k) \\ z(k) = c_2 x(k) + v_2 \end{cases}$$

where $c_1 \neq c_2 \neq 0$. The covariance of the initial state $E[x_0 x_0^T] = \pi_0$; v_1 and v_2 are uncorrelated, zero-mean white Gaussian sequences with covariances $E[v_1 v_1^T] = \sigma_1 \neq 0$, and $E[v_2 v_2^T] = \sigma_2 \neq 0$ respectively.

If there is no random noise, i.e., $v_1 = v_2 = 0$, the hybrid system is unobservable: it is easy to see that two different initial states (q_1, x_0) and $(q_2, \frac{c_1}{c_2}x_0)$ generate the same output sequences [48]. However, if there is random noise, (q_1, x_0) and $(q_2, \frac{c_1}{c_2}x_0)$ generate distinct output sequences. If the random noise v_1 and v_2 have different covariances, then we can uniquely determine the initial state. If we consider the case in which the actual initial state is (q_1, x_0), the output and its covariance are

$$z = c_1 x_0 + v_1, \qquad E[zz^T] = \pi_0 c_1 c_1^T + \sigma_1 \tag{2.7}$$

Next, if we consider the case in which the actual initial state is $(q_2, \frac{c_1}{c_2}x_0)$, the output and its covariance are

$$z = c_2(\tfrac{c_1}{c_2}x_0) + v_2, \qquad E[zz^T] = \pi_0 c_1 c_1^T + \sigma_2 \tag{2.8}$$

Thus, if $\sigma_1 \neq \sigma_2$, we can determine from which discrete state the output comes because the output covariances in (2.7) and (2.8) are different and since $c_1 \neq 0$ and $c_2 \neq 0$, the initial state can be determined uniquely. For instance, if the output comes from q_1, then the estimate of the continuous initial state is $\hat{x}_0 = x_0 + \frac{v_1}{c_1}$.

2.1.1.2 Observability of the discrete transition times

Lemma 3 gives the condition for the hybrid initial state to be observable, over a time interval up to, but not including the first transition. In this section, we focus without loss of generality on deriving the conditions under which the first discrete transition time k_1 can be uniquely determined from the output sequence $\mathcal{Z}_K$ on $[k_0, k_0 + K]$; similarly, the times of the ensuing transitions $k_i (i \in \{2, \dots\})$ can be computed. We define observability of the first discrete transition time as follows:

Definition 4. *(Observability of the first discrete transition time) The first discrete transition time of a discrete-time stochastic linear hybrid system H_d is observable on $[k_0, k_0 + K]$ if it can be determined uniquely from the output sequence $\mathcal{Z}_K = [z(k_0)^T \cdots z(k_0 + K)^T]^T$.*

If there is a discrete transition at time $k_1 \in [k_0, k_0 + K]$, the output at time k_1 and its

covariance are

$$\begin{aligned} z(k_1) &= C(q(k_1))A(q(k_0))^{k_1-k_0}x(k_0) \\ &\quad +C(q(k_1))\mathcal{F}_{t_0}(q(k_0))W_{t_0}(q(k_0)) + v_{k_1}(q(k_1)) \\ E[z(k_1)z(k_1)^T] &= C(q(k_1))A(q(k_0))^{k_1-k_0}\pi_0(A(q(k_0))^{k_1-k_0})^T C(q(k_1))^T \\ &\quad +C(q(k_1))\mathcal{F}_{t_0}(q(k_0))Q(q(k_0))\mathcal{F}_{t_0}(q(k_0))^T C(q(k_1))^T \\ &\quad +R(q(k_1)) \end{aligned} \tag{2.9}$$

where $\mathcal{F}_{t_0}(q(k_0)) := [A(q(k_0))^{k_1-k_0-1}\, A(q(k_0))^{k_1-k_0-2} \cdots I]$. If there is not a state transition at time k_1, the measurement at time k_1 and its covariance are

$$\begin{aligned} z(k_1) &= C(q(k_0))A(q(k_0))^{k_1-k_0}x(k_0) \\ &\quad +C(q(k_0))\mathcal{F}_{t_0}(q(k_0))W_{t_0}(q(k_0)) + v_{k_1}(q(k_0)) \\ E[z(k_1)z(k_1)^T] &= C(q(k_0))A(q(k_0))^{k_1-k_0}\pi_0(A(q(k_0))^{k_1-k_0})^T C(q(k_0))^T \\ &\quad +C(q(k_0))\mathcal{F}_{t_0}(q(k_0))Q(q(k_0))\mathcal{F}_{t_0}(q(k_0))^T C(q(k_0))^T \\ &\quad +R(q(k_0)) \end{aligned} \tag{2.10}$$

In order that the transition at time k_1 be observable, either means (deterministic observability) or the covariances of $z(k_1)$'s in (2.9) and (2.10) should be different. Thus, the observability condition of the first discrete transition time is:

Lemma 4. *(Observability of the first discrete transition time) The first discrete transition time is observable if and only if either the first discrete transition time of the corresponding deterministic hybrid system is observable, or*

$$\begin{aligned} & C(q(k_1))A(q(k_0))^{k_1-k_0}\pi_0(A(q(k_0))^{k_1-k_0})^T C(q(k_1))^T \\ & +C(q(k_1))\mathcal{F}_{\kappa_0}(q(k_0))Q(q(k_0))\mathcal{F}_{\kappa_0}(q(k_0))^T C(q(k_1))^T + R(q(k_1)) \\ \neq\ & C(q(k_0))A(q(k_0))^{k_1-k_0}\pi_0(A(q(k_0))^{k_1-k_0})^T C(q(k_0))^T \\ & +C(q(k_0))\mathcal{F}_{\kappa_0}(q(k_0))Q(q(k_0))\mathcal{F}_{\kappa_0}(q(k_0))^T C(q(k_0))^T + R(q(k_0)) \end{aligned}$$

for all $q(k) \neq q'(k) \in \{1, \cdots, N\}$.

Therefore, from **Lemma 3** and **Lemma 4**, the hybrid initial state and the first discrete transition time can be uniquely determined. The remaining state trajectories can be determined by repeating the procedure. For k_i $(i \geq 1)$, the $\hat{x}(k_i)$ will be given from the initial state estimate. Thus, we have the following observability condition:

Theorem 1. *A discrete-time stochastic linear hybrid system H_d with Gaussian noise is observable if and only if it satisfies both* ***Lemma 3*** *and* ***Lemma 4****.*

The observability conditions in **Theorem 1** are intrinsic to the system, i.e., the observability conditions can be tested without using measured output data since they depend on only the system parameters and noise characteristics. This test needs the operations of multiplication and addition of matrices (which are system parameters and noise covariances): the computation is straightforward with computational complexity depending on data size.

2.1.2 Exponential convergence condition for hybrid estimation algorithms

Based on conditions for the observability of stochastic linear hybrid systems, we would like to design estimators for those observable systems, and also quantify values of system parameters that would guarantee performance (exponential convergence, in our case). We extend the design methods proposed by Balluchi et al. [47] for hybrid systems with continuous-time, continuous state dynamics to encompass discrete-time stochastic hybrid systems.

A hybrid estimator finds estimates $\hat{q}$ and $\hat{x}$ for the current discrete state q and the continuous state x respectively. In this section, we first describe the structure of the hybrid estimator, and then analyze the continuous component of the estimator in detail to obtain bounds on the time between discrete transitions of state which would guarantee exponential convergence of our hybrid estimator. Throughout this section, all norms, unless specified otherwise, are 2-norms.

Definition 5. *(Exponential convergence of a hybrid estimator) Given a hybrid system H_d with N discrete modes, we say that a hybrid estimator is exponentially convergent if its discrete state estimate $\hat{q}$ exhibits correct identification of the discrete-state transition sequence of the original system after a finite number of steps; the continuous state estimate at any instant has a unique mean and convergent covariance in the sense of the minimum mean-square error; and the mean of the estimation error, $\bar{\zeta} = E[\hat{x} - x]$ converges exponentially to the set $\|\bar{\zeta}\| \leq M_0$ with a rate of convergence μ, where M_0 is the given steady-state error bound, and $|\mu| < 1$. In other words, the estimator is convergent if, for any switching time k_i,*

$$\hat{q}(k) = q(k), \quad \forall k > K, K \in \mathbb{N}^+ \tag{2.11}$$

$$\|\bar{\zeta}(k)\| \leq \mu^{(k-k_i)}\|\bar{\zeta}(k_i)\| + M_0, \quad \forall k > k_i \tag{2.12}$$

We design the hybrid estimator as a combination of a discrete observer to detect the discrete state switches, and an estimator to estimate the continuous dynamics, as proposed in [47]. In the rest of this section, we assume that we have a discrete observer that correctly identifies the discrete state, either immediately after a switch takes place, or with a known maximum time delay δ after a discrete transition. A discrete observer could be constructed using a bank of N estimators as a residual generator [52, 47] – even in this case, we could further increase the probability of correct discrete-state identification by enforcing a discrete state (or mode) estimation delay δ on the discrete observer. This would be possible only if the system were observable in the sense of a stochastic hybrid system, as explained earlier.

In this section, we design a least-square estimator in the form of N Kalman filters for the continuous state estimate. Although the underlying system in [47] is continuous-time and deterministic, the design methodology of [47] adapts well to discrete-time stochastic hybrid systems, as we show here.

We consider a hybrid system of the form described in (2.1). For the sake of simplicity of notation, we replace $A(q_k)$ and $C(q_k)$ with A_l and C_l, where $l \in \{1 \dots N\}$. We can then write the equations for the least-square estimator of a linear stochastic system as

$$\hat{x}(k+1) = (A_l - K_{k,l}C_l)\hat{x}(k) + K_{k,l}z(k),\ k \geq 0 \tag{2.13}$$

where l is the estimated discrete state, and $K_{k,l}$ is the optimal Kalman filter gain for mode l, given by $K_{k,l} = A_l P(k) C_l^T (R_l + C_l P(k) C_l^T)^{-1}$ and $P(k)$ satisfies the discrete Riccati recursion,

$$\begin{array}{lcl} P(k+1) & = & A_l P(k) A_l^T + Q_l - K_{k,l}(R_l + C_l P(k) C_l^T) K_{k,l}^T \\ P(0) & \triangleq & \pi_0 \end{array}$$

The Discrete Algebraic Riccati Equation (DARE) has a stabilizing solution that is unique if and only if $\{A_l, C_l\}$ is detectable and $\{A_l, Q_l^{1/2}\}$ is controllable on the unit circle. Any such solution is positive definite [66]. If these conditions are satisfied for every discrete state $l \in \{1 \dots N\}$, we can design a bank of N steady-state, exponentially convergent Kalman filters to estimate the continuous state of the system. We can then show that, for a given discrete state l, correctly identified,

$$\begin{array}{lcl} \hat{x}(k+1) & = & (A_l - K_l C_l)\hat{x}(k) + K_l z(k) \\ \hat{\zeta}(k+1) & = & (A_l - K_l C_l)\hat{\zeta}(k) \end{array} \tag{2.14}$$

where K_l is a steady-state Kalman filter gain for mode l. Clearly, $\hat{\zeta}$ is exponentially convergent if

$$(A_l - K_l C_l) \text{ is stable} \tag{2.15}$$

We now follow the methodology of [47] to determine the evolution of the estimation error across the discrete transition sequence. Let us consider two consequent discrete transitions of H_d, occurring at times k_i and k_{i+1}. Suppose the transition at time k_{i+1} was from discrete state m to l, and was detected at time k'_{i+1} such that $k'_{i+1} - k_{i+1} \leq \delta$. Similarly, $k'_i - k_i \leq \delta$. This is illustrated in Figure 2.2. We are interested in the region $k \in \{k'_i, k'_i + 1, \ldots, k'_{i+1}\}$.

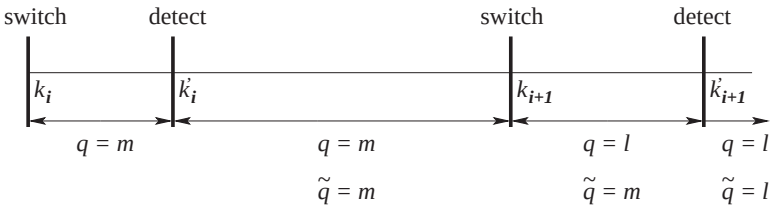


Figure 2.2: Illustration of the transition sequence.

Since we assume that by time-step k'_i the discrete state has been identified correctly, for the exponential convergence of the estimation error on k'_i to k'_{i+1}, we require that:

1. The error converges exponentially between k'_i and k_{i+1};
2. The error divergence between k_{i+1} and k'_{i+1} due to wrong discrete state estimation does not upset the exponential convergence of the error on k'_i to k'_{i+1}.

Following the methodology of [47], dividing the time interval between k'_i and k'_{i+1} into two regions, we get the error dynamics of the form ($i \in \{0, 1, \cdots\}$)

$$\begin{aligned} \bar{\zeta}(k+1) &= (A_m - K_m C_m)\bar{\zeta}(k), \quad k \in \{k'_i, \ldots, k_{i+1} - 1\} \\ \bar{\zeta}(k+1) &= (A_m - K_m C_m)\bar{\zeta}(k) + [(A_m - A_l) \\ &- K_m(C_m - C_l)]\bar{x}(k), \quad k \in \{k_{i+1}, \ldots, k'_{i+1} - 1\} \end{aligned} \tag{2.16}$$

where $\bar{x} = E[x]$. The second term in (2.16) arises because a Kalman filter designed for the discrete state m is being used to estimate the dynamics of the discrete state l. Combining

(2.16), we express the error dynamics by $\bar{\zeta}(k+1) = (A_m - K_m C_m)\bar{\zeta}(k) + y(k),\ k \in \{k'_i, \ldots, k'_{i+1} - 1\}$ where $(i \in \{0, 1, \cdots\})$

$$y(k) = \begin{cases} 0, & \text{for } k \in \{k'_i, \ldots, k_{i+1} - 1\} \\ ((A_m - A_l) - K_m(C_m - C_l))\bar{x}_k, & \text{for } k \in \{k_{i+1}, \ldots, k'_{i+1} - 1\} \end{cases} \tag{2.17}$$

From this, we get:

$$\left\|\bar{\zeta}(k+1)\right\| \leq \left\|(A_m - K_m C_m)^{k+1-k'_i}\bar{\zeta}(k'_i)\right\| + \left\|\sum_{l=0}^{k-k'_i}(A_m - K_m C_m)^{k-k'_i-l} y(k'_i + l)\right\| \tag{2.18}$$

where $k \in \{k_{i+1}, \ldots, k'_{i+1} - 1\}$.

Lemma 5. *Given a matrix $A \in \mathbb{R}^{n\times n}$ with all distinct eigenvalues,*

$$\|A^t\| \leq k(A)\alpha^t(A),\ \forall t \geq 0 \tag{2.19}$$

where $\alpha(A)$ is the maximal absolute value of the eigenvalues of A, and $k(A) = \|\Gamma\|\|\Gamma^{-1}\|$, the condition number of A under the inverse, where $\Gamma^{-1}A\Gamma = J$, the Jordan canonical form.

Further simplification of (2.18) using **Lemma 5** gives us

$$\begin{aligned} \left\|\bar{\zeta}(k+1)\right\| \leq\ & k(A_m - C_m)[\alpha(A_m - K_m C_m)]^{k+1-k'_i}\left\|\bar{\zeta}(k'_i)\right\| \\ & + k(A_m - C_m)\max\|y(k)\|\,(k - k_{i+1}) \end{aligned} \tag{2.20}$$

where $k \in \{k_{i+1}, \ldots, k'_{i+1}\}$. Since $k'_{i+1} - k_{i+1} \leq \delta$, if

$$\|y(k)\|_\infty \leq U = \max\|(A_m - A_l) - K_m(C_m - C_l)\|_1 X \tag{2.21}$$

such that $X \geq \|x\|_\infty$, $X > 0$, we can write

$$\left\|\bar{\zeta}(k+1)\right\| \leq k(A_m - C_m)[\alpha(A_m - K_m C_m)]^{k+1-k'_i}\left\|\bar{\zeta}(k'_i)\right\| + \sqrt{n}U\delta k(A_m - C_m) \tag{2.22}$$

Lemma 6. *Consider a hybrid system with a single discrete state, in which the discrete-time evolution of the continuous state variable is given by $x(k+1) = \eta x(k),\ |\eta| < 1$. Suppose the state x is subject to resets $x(t_s) = a\eta x(t_s - 1) + b$, occurring at switching times $\{t_s\}$,*

with $a \geq 1$ and $b \geq 0$. Then the evolution of x can be described by

$$x(k) = \eta^{k-t_{s-1}} x(t_{s-1}),\ k \in \{t_{s-1}, \ldots, t_s - 1\},\ s \in \{2, 3, \cdots\} \tag{2.23}$$

$$x(t_s) = a\eta^{t_s - t_{s-1}} x(t_{s-1}) + b \tag{2.24}$$

Let us also assume there exists a lower bound β on the time between resets, i.e., $t_s - t_{s-1} \geq \beta \geq 1$, for all $s > 1$. Then, if $x_{t_0} > 0$ and $\mu = \eta^{(\frac{\log_\eta a}{\beta}+1)}$ such that $|\mu| < 1$, $x(k)$ converges exponentially to the set $[0, \frac{b}{1-\eta^\beta}]$ with a rate of convergence greater than or equal to μ.

Using (2.15), (2.21) and (2.22) with **Lemma 5** and **Lemma 6**, we arrive at the following theorem:

Theorem 2. *Consider a stochastic linear hybrid system of the form in (2.1), a steady-state error bound M_0 and rate of convergence μ, $|\mu| < 1$, $|\alpha(A_m - K_m C_m)| \leq |\mu|$ for all $m \in \{1, \cdots, N\}$, where $\alpha(A)$ is the maximal absolute value of the eigenvalues of A. Let $k(A) = \|\Gamma\|\|\Gamma^{-1}\|$, the condition number of A under the inverse, where $\Gamma^{-1} A \Gamma = J$, the Jordan canonical form. Suppose the following five conditions are satisfied:*

1. *The system is observable under **Definition 2**;*
2. *$\{A_m, C_m\}$ couples are observable, $\{A_m, Q_m^{1/2}\}$ couples are controllable, and $(A_m - K_m C_m)$ is stable with all distinct eigenvalues, for all $m \in \{1, \cdots, N\}$;*
3. *There exists $X > 0$, $\|x(k)\|_\infty \leq X$, $k = 1, 2, \ldots$ such that*

$$\|y(k)\|_\infty \leq U = \max \|(A_m - A_l) - K_m(C_m - C_l)\|_1 X$$

4. *The maximum discrete state estimation delay, δ satisfies the relation*

$$\delta \leq \frac{M_0}{\sqrt{n} U \max_m [k(A_m - K_m C_m)]} \tag{2.25}$$

5. *The minimum time between switching events, β, known as the* minimum sojourn time, *satisfies the conditions*

$$\begin{aligned} \beta &> \bar{\beta} + \delta, \quad \text{where} \\ \bar{\beta} &> \max\left[\frac{1}{|\log \mu|} \log\left|\left(1 - \frac{\sqrt{n} U \delta \max_m [k(A_m - K_m C_m)]}{M_0}\right)\right|, \max_m \frac{\log[k(A_m - K_m C_m)]}{|\log[\alpha(A_m - K_m C_m)]|}\right] \end{aligned}$$

Then, a hybrid estimator can be designed that converges to within the steady-state bound M_0 with a rate of convergence greater than or equal to μ.

Corollary 1. *If Conditions (1)-(4) of* **Theorem 2** *are satisfied, then, given a steady-state error bound M_0 and a rate of convergence μ, an estimator can be designed that converges exponentially to M_0 with a rate of at least μ if the time between switching events is at least $\beta = \bar{\beta} + \delta$, where*

$$\bar{\beta} = \max \left[\frac{1}{|\log \mu|} \log \left| \left(1 - \frac{\sqrt{n} U \delta k (A_m - K_m C_m)}{M_0} \right) \right|, \max \frac{\log[k(A_m - K_m C_m)]}{|\log[\alpha(A_m - K_m C_m)]|} \right]$$

Theorem 2 and **Corollary 1** provide conditions on switching times to guarantee exponential convergence of hybrid estimators for stochastic linear hybrid systems. That is, for exponential convergence of hybrid estimators for a stochastic linear hybrid system in (2.1), the system must remain in a given discrete state long enough (i.e., it must satisfy the inequality in Condition 5 in **Theorem 2**) so that the estimation error divergence due to the discrete estimation delay cannot destroy the exponential convergence when the discrete estimation is correct.

2.1.3 Aircraft model for tracking

We test the exponential convergence condition for hybrid estimation algorithms in **Theorem 2** using aircraft tracking examples. For this purpose, in this section, we first derive aircraft models for aircraft tracking in the plane. These models will be used throughout this dissertation unless otherwise stated. Since an aircraft trajectory is composed of straight lines and circular arcs, we model the dynamics of an aircraft as a discrete-time stochastic hybrid system with two discrete modes: a constant velocity (CV) mode in which the aircraft flies with constant velocity and constant heading so that its trajectory is a straight line, and a coordinated turn (CT) mode in which the aircraft flies with constant yaw rate so that its trajectory is a circular arc. The CV mode could correspond to different constant velocities, i.e., two different constant velocities could constitute the same CV mode. Similarly, different yaw rates could correspond to the same CT mode. In other words, the CV mode is parameterized with the aircraft's velocity and the CT mode with the aircraft's yaw rate. In this way, two discrete modes can describe all possible values of the velocity and yaw rate of aircraft.

We use the coordinated turn model with one CT mode parameterized by a yaw rate instead of multiple CT modes with different constant yaw rates, since a coordinated turn model composed of a finite set of CT modes with known constant yaw rates could not estimate the actual yaw rate and flight mode accurately, if the aircraft's actual yaw rate is not matched to one of the values of those used in the models. This is because the yaw rate estimate comes from the finite set of predetermined yaw rates. To solve this problem, the yaw rate is included as a state component, to be estimated. Thus, a new model has an additional equation for the yaw rate ω:

$$\omega(k+1) = \omega(k) + w_\omega(k) \tag{2.26}$$

where $w_\omega(k)$ is zero-mean white Gaussian noise. This model is nonlinear, so an extended Kalman filter must be used for state estimation. However, the accuracy of yaw rate estimates is sensitive to the design parameters used for state estimation. The estimate of yaw rate oscillates severely around the true rate at the start of a maneuver. This model has been used for aircraft tracking widely [1, 67, 18], however, because it has produced good position and velocity estimates.

In order to obtain accurate yaw rate estimates, we use the *Wiener-sequence acceleration model* [68] as the aircraft model for the coordinated turn mode. It assumes that the acceleration increment is an independent, zero-mean, white Gaussian noise process, i.e., $a(k) - a(k-1) = w(k)$ (where a is an acceleration and w is a zero-mean, white Gaussian noise). Standard Kalman filters are used for state estimation because this model is linear. A radar is assumed to be used for measurements and thus we use only aircraft's position information. However, Automatic Dependent Surveillance-Broadcast (ADS-B) is assumed to be used for measurements, both the aircraft's position and velocity information are used to estimate the aircraft's current states.

If the state of an aircraft is defined as $x = [x\, \dot{x}\, \ddot{x}\, y\, \dot{y}\, \ddot{y}]^T$, the aircraft model for constant velocity (CV) mode is

$$x(k) = \begin{bmatrix} 1 & T & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} x(k-1) + \begin{bmatrix} T^2/2 & 0 \\ T & 0 \\ 0 & 0 \\ 0 & T^2/2 \\ 0 & T \\ 0 & 0 \end{bmatrix} w_{cv}(k) \tag{2.27}$$

$$y(k) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} x(k) + v_{cv}(k) \tag{2.28}$$

For the coordinated turn (CT) mode, the aircraft model (Wiener-sequence acceleration model) is

$$x(k) = \begin{bmatrix} 1 & T & T^2/2 & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & T^2/2 \\ 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} x(k-1) + \begin{bmatrix} T^2/2 & 0 \\ T & 0 \\ 1 & 0 \\ 0 & T^2/2 \\ 0 & T \\ 0 & 1 \end{bmatrix} w_{ct}(k) \tag{2.29}$$

$$y(k) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} x(k) + v_{ct}(k) \tag{2.30}$$

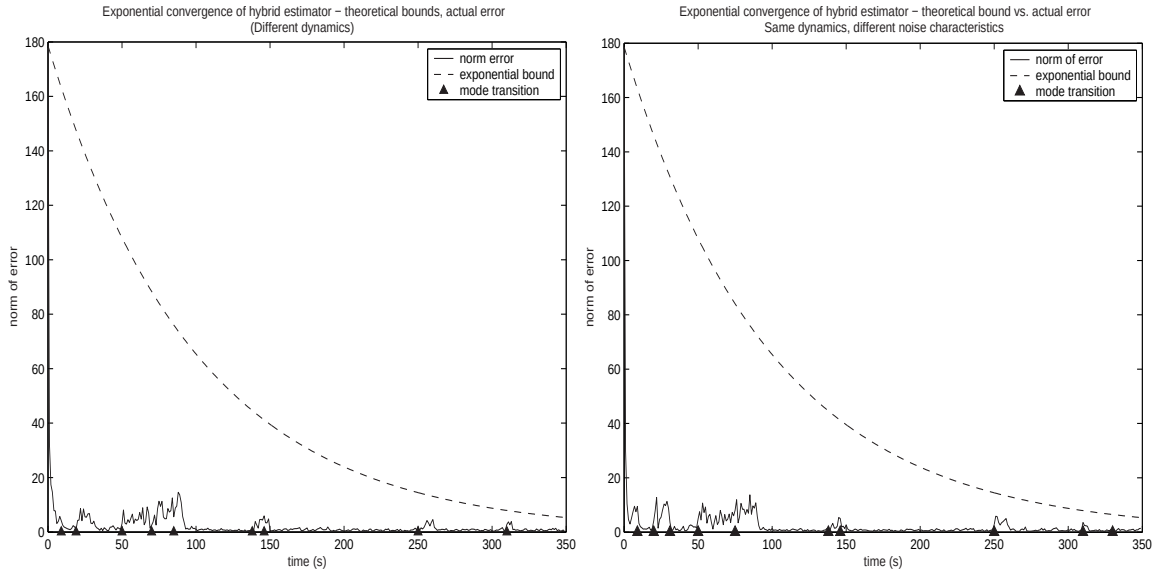
where T is the sampling interval and w_{cv}, w_{ct}, v_{cv}, and v_{ct} are zero-mean, uncorrelated, white Gaussian process noise and measurement noise for CV mode and CT mode, respectively. The yaw rate estimate $\hat{\omega}$ is computed with a state estimate from (2.29) as follows:

$$\hat{\omega}(k) = sign(\hat{\dot{x}}(k)\hat{\ddot{y}}(k) - \hat{\ddot{x}}(k)\hat{\dot{y}}(k))\sqrt{\frac{\hat{\ddot{x}}(k)^2 + \hat{\ddot{y}}(k)^2}{\hat{\dot{x}}(k)^2 + \hat{\dot{y}}(k)^2}} \tag{2.31}$$

To get a smooth estimate of ω, we design a first-order, low-pass filter $G(s) = \frac{1}{11.9s+1}$. Therefore, for aircraft tracking, we use a stochastic linear hybrid system model with two modes which are parameterized with the velocity and yaw rate of the aircraft.

Example: Exponential convergence of hybrid estimation algorithms

We apply the hybrid estimator design criteria in **Theorem 2** derived in Section 2.1.2 to the design of an estimator for the switched, linearized trajectory of an aircraft. We consider two discrete states, both coordinated turns, but with different angular velocities, one with a turn rate of 2° per second, and the other with a turn rate of 5° per second, which represent aircraft trajectories composed of slow turns and sharp turns. Here we use a simpler aircraft



(a) (b)

Figure 2.3: (a) Exponential convergence of error. (b) Convergence of error when modes have same dynamics but different noise characteristics. The triangles denote discrete transition times ($\mu = 0.99$, $M_0 = 0$, and $T = 2sec$).

model for a coordinated turn mode than that in Section 2.1.3, given by

$$\begin{aligned} x_k &= \begin{bmatrix} 1 & \frac{\sin\omega T}{\omega} & 0 & -\frac{1-\cos\omega T}{\omega} \\ 0 & \cos\omega T & 0 & -\sin\omega T \\ 0 & \frac{1-\cos\omega T}{\omega} & 1 & \frac{\sin\omega T}{\omega} \\ 1 & \sin\omega T & 0 & \cos\omega T \end{bmatrix} x_{k-1} + \begin{bmatrix} \frac{T^2}{2} & 0 \\ T & 0 \\ 0 & \frac{T^2}{2} \\ 0 & T \end{bmatrix} u_{k-1} + w_k \\ y_k &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x_k + v_k \end{aligned} \tag{2.32}$$

where $x = [\ x_1 \quad \dot{x}_1 \quad x_2 \quad \dot{x}_2\ ]^T$ with x_1 and x_2 the position coordinates, $u = [\ u_1 \quad u_2\ ]^T$ with u_1 and u_2 the velocity components, ω is the turn rate, T is the sampling interval, w is the process noise, and v is the sensor noise. The yaw rate is fixed in the coordinated turn model in (2.32), yet (2.29) can represent infinitely many yaw rates. We choose an operating velocity of 150 knots. We find that for an instantaneous discrete decision time, the time between discrete transitions should be at least 8 seconds to guarantee exponential convergence with a rate of 0.99. The comparison of the bounds is shown in Figure 2-(a). We

also note that by **Lemma 5**, the norm of the mean error does not have to be monotonic, but if the conditions explained above are satisfied, it will be bounded by an exponential of rate μ. This is also seen in the example. From **Theorem 1**, identical dynamics with different noise characteristics in each discrete state can still make the system observable in the stochastic hybrid context. We demonstrate this by designing an exponentially convergent hybrid estimator for a switched aircraft trajectory - the two discrete states correspond to 2° per second turns with different process noise covariances. This is shown in Figure 2-(b).

2.2 Performance analysis of hybrid estimation algorithms

Discrete-time stochastic hybrid systems with the Markov discrete state dynamics have been extensively used for many applications, such as multiple-target tracking problems [56] and speech recognition problems [53]. Since we have established conditions for the observability of the stochastic linear hybrid systems in Section 2.1, in this section, we analyze the performance of hybrid estimators for a class of observable discrete-time stochastic linear hybrid systems in which the discrete transitions are governed by finite-state Markov chains.

2.2.1 Discrete-time stochastic linear hybrid systems with the Markov discrete state dynamics

For the sake of notational simplicity, in this section, we replace $A(q(k)), B(q(k)), C(q(k))$, $w_k(q(k))$, and $v_k(q(k))$ in (2.1) with $A_i, B_i, C_i, w_i(k)$, and $v_i(k)$ respectively. We consider a discrete-time stochastic linear hybrid system with the Markov discrete state dynamics [20] defined as:

$$H : \begin{cases} x(k+1) &= A_i x(k) + B_i u(k) + w_i(k) \\ z(k) &= C_i x(k) + v_i(k) \end{cases}, \ k \in \mathbb{N} \tag{2.33}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^l$ and $z \in \mathbb{R}^p$ are the continuous state, control input, and output variables respectively. The index $i \in \{1, 2, \cdots, N\}$ represents the discrete state whose evolution is governed by the finite state Markov chain

$$\mu(k+1) = \Pi\mu(k) \tag{2.34}$$

where $\Pi = \{\pi_{ij}\} \in \mathbb{R}^{N\times N}$ is the *mode transition matrix* whose elements represent mode transition probabilities, and $\mu(k) \in \mathbb{R}^N$ is the *mode probability* at time k. We denote the process noise covariance and the measurement noise covariance as $E[w_i(k)w_i(k)^T] = Q_i \in \mathbb{R}^{n\times n}$ and $E[v_i(k)v_i(k)^T] = R_i \in \mathbb{R}^{p\times p}$ respectively. We define $Z(k) = \{z(0), \cdots, z(k)\}$ as the measurement sequence up to time k.

For aircraft tracking, we model the dynamics of an aircraft as a stochastic linear hybrid system whose discrete state denotes the aircraft's flight mode, since the flight mode changes of an aircraft depend on the pilot's input which is unknown to the surveillance system. This unknown pilot's input could be modelled as a random process, and we model the discrete state dynamics as a finite Markov chain. Thus, we use the discrete-time stochastic linear hybrid system in (2.33)-(2.34) as an aircraft model for tracking throughout this dissertation.

2.2.2 Multiple Model Adaptive Estimation (MMAE) algorithm

In this section, we consider a generic hybrid estimation algorithm for the discrete-time stochastic linear hybrid system (2.33)-(2.34). Following the Bayesian estimation derivation in [55], the state estimate of hybrid estimation is the conditional mean:

$$\hat{x}(k+1) = E[x(k+1)|Z(k+1)] = \int_{-\infty}^{\infty} x(k+1)p(x(k+1)|Z(k+1))dx(k+1) \tag{2.35}$$

where $p(\cdot|\cdot)$ is the conditional probability density function, given by:

$$p(x(k+1)|Z(k+1)) = \frac{p(x(k+1), Z(k+1))}{p(Z(k+1))} = \frac{\sum_{i=1}^{N} p(x(k+1), Z(k+1), m_i(k+1))}{p(Z(k+1))} \tag{2.36}$$

Thus, the state estimate (2.35) is

$$\begin{aligned} & \hat{x}(k+1) \\ = & \int_{-\infty}^{\infty} x(k+1) \sum_{i=1}^{N} p(x(k+1)|Z(k+1), m_i(k+1))p(m_i(k+1)|Z(k+1))dx(k+1) \\ = & \sum_{i=1}^{N} \hat{x}_i(k+1)p(m_i(k+1)|Z(k+1)) \end{aligned} \tag{2.37}$$

where $\hat{x}_i(k+1) = \int_{\infty}^{\infty} x(k+1)p(x(k+1)|Z(k+1), m_i(k+1))dx(k+1)$ is the mode-conditioned state estimate of $x(k+1)$ given $m_i(k+1)$. $\hat{x}_i(k+1)$ is computed by the state estimator matched to mode i. Therefore, the state estimate (2.37) is a weighted sum

of N mode-conditioned state estimates produced by each Kalman filter with the weight $p(m_i(k+1)|Z(k+1))$. The weight can be expressed by

$$\begin{aligned} p(m_i(k+1)|Z(k+1)) &= \frac{p(z(k+1)|m_i(k+1),Z(k))p(m_i(k+1)|Z(k))}{p(z(k+1)|Z(k))} \\ &= \frac{\Lambda_i(k+1)p(m_i(k+1)|Z(k))}{p(z(k+1)|Z(k))} \end{aligned} \tag{2.38}$$

where $\Lambda_i(k+1) := \mathcal{N}(r_i(k+1); 0, S_i(k+1))$ is the likelihood function of mode i (a measure of how likely the model used in Kalman filter i is the correct one), $r_i(k+1) = z(k+1) - C_i\hat{x}_i(k+1)$ is the residual produced by the Kalman filter i, $S_i(k+1) \in \mathbb{R}^{p\times p}$ is the corresponding residual covariance, and $\mathcal{N}(a; b, c)$ is the probability at a of a normal distribution with mean b and covariance c. $p(m_i(k+1)|Z(k))$ is the mode probability estimate at time $k+1$. If the mode transitions are governed by a finite Markov chain, the mode probability estimate is

$$\begin{aligned} p(m_i(k+1)|Z(k)) &= \sum_{l=1}^{N} p(m_i(k+1)|m_l(k))p(m_l(k)|Z(k)) \\ &= \sum_{l=1}^{N} \pi_{il} p(m_l(k)|Z(k)) \end{aligned} \tag{2.39}$$

Thus, the weight (mode probability) (2.38) is

$$p(m_i(k+1)|Z(k+1)) = \frac{1}{c(k+1)}\Lambda_i(k+1)\sum_{l=1}^{N}\pi_{il}p(m_l(k)|Z(k)) =: \mu_i(k+1) \tag{2.40}$$

where $c(k+1)$ is a normalization constant. The mode estimate at time k is chosen to be the mode which has the maximum mode probability at that time. The mode probability depends not only on the finite Markov chain but also on the likelihood produced by each Kalman filter. The state estimate (2.37) is

$$\hat{x}(k+1) = \sum_{i=1}^{N}\hat{x}_i(k+1)\big[\frac{1}{c(k+1)}\Lambda_i(k+1)\sum_{l=1}^{N}\pi_{il}p(m_l(k)|Z(k))\big] \tag{2.41}$$

Equations (2.40)-(2.41) are referred to as the Multiple Model Adaptive Estimation (MMAE) algorithm [55]. In MMAE, all individual Kalman filters run independently at every time step (which is different from the Interacting Multiple Model (IMM) algorithm that will be described next). Equation (2.41) shows that the state estimate depends on the likelihood function, the performance of the hybrid estimator thus greatly depends on the behavior of the likelihood function.

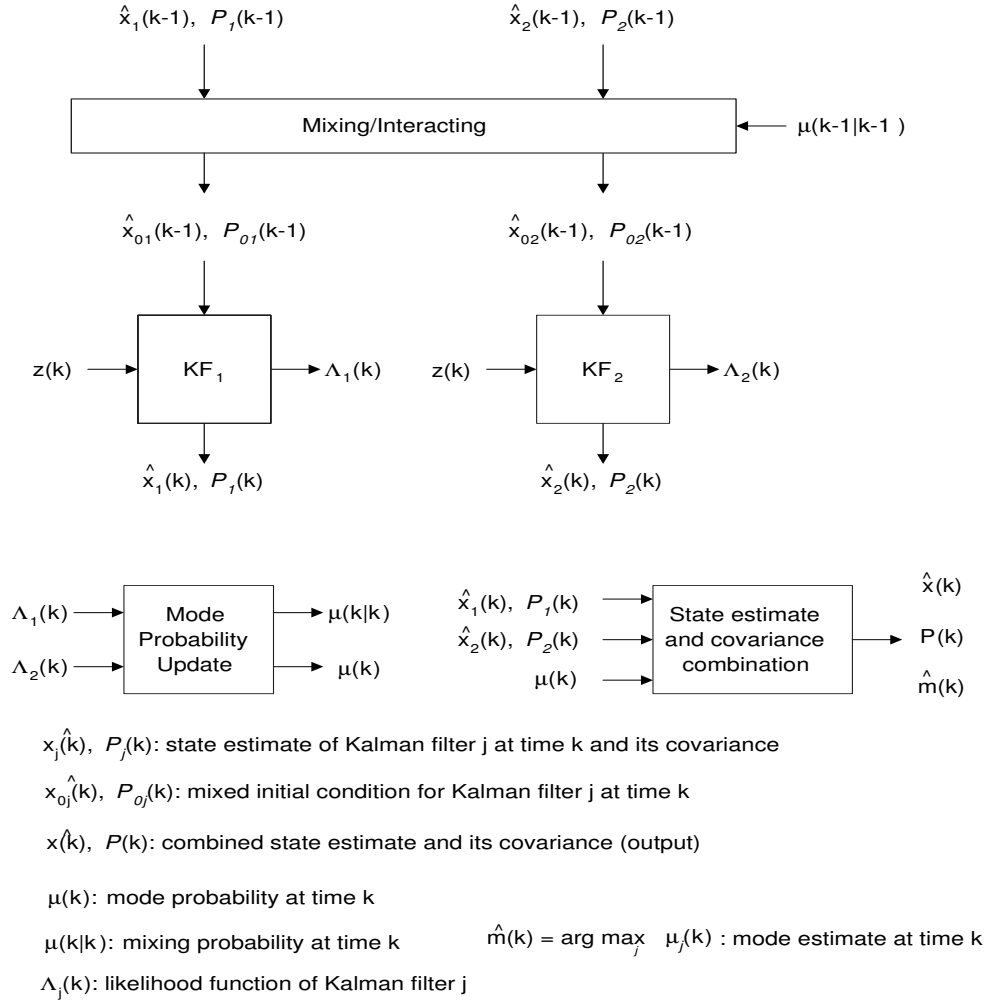


Figure 2.4: Structure of the IMM algorithm (for two modes) from [1].

The Interacting Multiple Model (IMM) algorithm [56] (on which RMIMM is based) has the same structure as MMAE except that it has a *Mixing/Interacting* step at the beginning of the estimation process, which computes new initial conditions for mode-matched Kalman filters at each time step. Detailed analysis of IMM will be discussed in a later section.

2.2.3 Sojourn time analysis for stochastic linear hybrid systems

In Section 2.1.2, we derived a condition under which hybrid estimation algorithms converge exponentially to the true hybrid state, given the maximum mode estimation delay and the

minimum sojourn time ($\beta = \min_{i=1}^{N} \beta_i$ where β_i is the minimum sojourn time at mode i). Intuitively, until the correct mode is detected after a transition, the estimator is mismatched and its mean-square error diverges (for the maximum mode estimation delay δ in the worst case) and therefore it needs enough time ($\bar{\beta}$: the time between mode changes required for the convergence of the Kalman filters) in the correct mode after the correct mode detected for the mean-square error to decrease before the next transition occurs. Therefore, $\beta \geq \bar{\beta} + \delta$ is the required minimum sojourn time of the system in order to guarantee exponential convergence of the hybrid estimator. Since the sojourn time of the stochastic linear hybrid system is governed by a finite Markov chain, we cannot compute the minimum sojourn time exactly, rather with some probability. The probability of the system being in mode j at time $k + \beta_i$ given that it is in mode i at time k (i.e. no mode transition in $(k, k + \beta_i)$) is:

$$P(m_j(k + \beta_i)|m_i(k)) = \pi_{ii}^{\beta_i - 1}\pi_{ji} \tag{2.42}$$

where $m_i(k)$ denotes the event that the mode at time k is mode i. Thus, the probability that the sojourn time at mode i is greater than or equal to β_i is

$$P(t \geq \beta_i) = \sum_{t=\beta_i}^{\infty} \pi_{ii}^{t-1}(1 - \pi_{ii}) = \pi_{ii}^{\beta_i - 1} \tag{2.43}$$

Proposition 1. *Let $\beta = \min_i \beta_i$ ($i \in \{1, \cdots, N\}$) be the minimum sojourn time for the stochastic linear hybrid system (2.33)-(2.34) and $j = \arg\min_i \beta_i$. If $\beta \geq \bar{\beta} + \delta$, then hybrid estimation algorithms for the stochastic linear hybrid system (2.33)-(2.34) in Section 2.2.2 can be guaranteed to converge exponentially to the true hybrid state with probability $\pi_{jj}^{\beta - 1}$.*

Proof. The proof follows from (2.43) and **Theorem 2**. □

However, the probability in (2.43) becomes small rapidly as the required minimum sojourn time β becomes large. In this case, the hypothesis of **Proposition 1** holds with very small probability and we can obtain very little useful information about the performance of hybrid estimation algorithms through sojourn time analysis alone. This is because the sojourn time analysis based on a finite Markov chain takes into account only the discrete dynamics of the hybrid system. Therefore, we need to incorporate knowledge from the continuous dynamics to get accurate performance analysis results of hybrid estimation algorithms.

2.2.4 Steady-state and transient analysis for hybrid estimation

In this section, we first analyze the performance of hybrid estimation (either MMAE or IMM) by analyzing steady-state mean residuals. Since steady-state analysis gives only necessary conditions on the performance of hybrid estimation, we then analyze the transient behavior of mode probabilities, which are functions of the likelihoods and therefore of the residuals.

Motivated by Caputi [65], we first derive the steady-state mean residual for each mode i of the hybrid system (2.33)-(2.34). We define the following quantities:

$$\begin{aligned}
\Delta A_i &:= A_T - A_i \\
\Delta B_i &:= B_T - B_i \\
\Delta C_i &:= C_T - C_i \\
\Delta u_{i_{ss}} &:= u_{T_{ss}} - u_{i_{ss}} := \lim_{k\to\infty} u_T(k) - \lim_{k\to\infty} u_i(k) \\
\hat{x}_{i_{ss}} &:= \lim_{k\to\infty} \hat{x}_i(k) \\
\bar{e}_{i_{ss}} &:= \lim_{k\to\infty} E[e_i(k)] = \lim_{k\to\infty} E[(x(k) - \hat{x}_i(k))]
\end{aligned} \tag{2.44}$$

where the subscript $T \in \{1, \cdots, N\}$ represents the true mode. Following a procedure similar to the one adopted by Hanlon et al. [69], the steady-state mean residual for mode i is

$$\begin{aligned}
\bar{r}_{i_{ss}} &= C_T A_T \bar{e}_{i_{ss}} + (C_T \Delta A_i + \Delta C_i A_T - \Delta C_i \Delta A_i)\hat{x}_{i_{ss}} + C_T B_T \Delta u_{i_{ss}} \\
&\quad + (C_T \Delta B_i + \Delta C_i B_T - \Delta C_i \Delta B_i) u_{i_{ss}} \\
&= \{C_T A_T [I - (I - K_i C_T) A_T]^{-1} [(I - K_i C_T)\Delta A_i - K_i \Delta C_i A_i] \\
&\quad + (C_T \Delta A_i + \Delta C_i A_T - \Delta C_i \Delta A_i)\}\hat{x}_{i_{ss}} \\
&\quad + \{C_T A_T [I - (I - K_i C_T) A_T]^{-1} (I - K_i C_T) B_T + C_T B_T\}\Delta u_{i_{ss}} \\
&\quad + \{C_T A_T [I - (I - K_i C_T) A_T]^{-1} [(I - K_i C_T)\Delta B_i - K_i \Delta C_i B_i] \\
&\quad + (C_T \Delta B_i + \Delta C_i B_T - \Delta C_i \Delta B_i)\} u_{i_{ss}}
\end{aligned} \tag{2.45}$$

where K_i is the steady-state Kalman filter gain for mode i. We assume that the matrix inverse in (2.45) exists. If mode i is the correct mode ($i = T$), then $\bar{r}_{i_{ss}} = 0$. If $\bar{r}_{j_{ss}} \neq 0$ ($\forall j \neq i$), then the correct mode can be detected. However, even if mode i is not the correct mode ($i \neq T$), the steady-state mean residual for mode i is zero if the following is true: $(I - K_i C_T)\Delta A_i - K_i \Delta C_i A_i = 0 \bigwedge (C_T \Delta A_i + \Delta C_i A_T - \Delta C_i \Delta A_i) = 0 \bigwedge (I - K_i C_T)\Delta B_i - K_i \Delta C_i B_i = 0 \bigwedge (C_T \Delta B_i + \Delta C_i B_T - \Delta C_i \Delta B_i) = 0 \bigwedge \Delta u_{i_{ss}} = 0$. This means that if at least

two models are identical and the corresponding control inputs are the same, then the steady-state residuals of both the corresponding modes are zero. In this case, a hybrid estimation algorithm will not work, as its performance depends on the differences between residuals, which in turn arise from model differences and input differences. In the above condition, the first four equalities come from model differences and the last equality comes from input differences. From this condition, we know that the model differences and/or the input differences should be large enough for hybrid estimation to work; this supports Maybeck's heuristic observation that the performance of MMAE depends on a significant difference between the residual characteristics [55]. We now study the transient characteristics of the mode probability (a function of the residuals) in detail to estimate the mode estimation delay, which is a good measure of the performance of hybrid estimation.

We now consider the transient mean behavior of a hybrid estimator, and analyze its performance in the sense of exponential convergence as defined in **Definition 5**. A steady-state Kalman filter is assumed to be used as the state estimator for each mode. For the sake of notational simplicity, we define a predicted mode probability of mode i at time k as

$$\mu_i^-(k) \;:=\; \textstyle\sum_{l=1}^{N} \pi_{il}\mu_l(k-1) \tag{2.46}$$

Using $\mu_i(k) := p(m_i(k)|Z(k))$ in (2.40), the condition for correct mode detection at time k is: $\forall i \neq T$

$$\begin{aligned} & \mu_T(k) > \mu_i(k) \\ \iff\; & \tfrac{1}{c(k)}\Lambda_T(k)\mu_T^-(k) > \tfrac{1}{c(k)}\Lambda_i(k)\mu_i^-(k) \\ \iff\; & \Lambda_T(k) > \Lambda_i(k)\tfrac{\mu_i^-(k)}{\mu_T^-(k)} \end{aligned} \tag{2.47}$$

Since $\Lambda_i(k) = \mathcal{N}(\bar{r}_i(k); 0, S_i(k)) = (2\pi)^{-n/2}|S_i|^{-1}\exp[-\frac{1}{2}\bar{r}_i(k)^T S_i^{-1}\bar{r}_i(k)]$ $(S_i = S_i^T > 0)$ where $\bar{r}$ is the mean residual, (2.47) becomes

$$0 \leq \bar{r}_T(k)^T S_T^{-1}\bar{r}_T(k) < \bar{r}_i(k)^T S_i^{-1}\bar{r}_i(k) + 2\ln\left(\frac{|S_i|}{|S_T|}\right) + 2\ln\left(\frac{\mu_T^-(k)}{\mu_i^-(k)}\right) \tag{2.48}$$

To detect the correct mode exactly for any $k \in \mathbb{N}$, (2.48) must hold for all $k \in \mathbb{N}$ $(\forall i \neq T)$. If there is a time delay (δ_T) for correct mode detection when a mode transition into mode T occurs at time k_l $(l \in \mathbb{N}^+)$, (2.48) holds for $k \in [k_l + \delta_T, k_{l+1})$. For the existence of a $\bar{r}_T(k)$ satisfying (2.48), the right-hand-side in (2.48) must be greater than or equal to zero.

The following condition holds for a positive definite matrix S_i^{-1} [70]:

$$\lambda_{min}(S_i^{-1})\bar{r}_i(k)^T\bar{r}_i(k) \leq \bar{r}_i(k)^T S_i^{-1}\bar{r}_i(k) \leq \lambda_{max}(S_i^{-1})\bar{r}_i(k)^T\bar{r}_i(k) \tag{2.49}$$

where $\lambda_{min}(S_i^{-1})$ and $\lambda_{max}(S_i^{-1})$ are the minimum and the maximum eigenvalues of S_i^{-1} respectively. Thus, we have the following condition:

Proposition 2. *The correct mode can be detected in δ_T time steps after a mode transition at time k_l if there exists $\delta_T \in \mathbb{N}^+$ such that for $k \in [k_l + \delta_T, k_{l+1})$ ($l \in \mathbb{N}^+$, $\forall i \neq T$), Condition 1 holds and either Condition 2 or Condition 3 is true.*

1. $\bar{r}_i(k)^T S_i^{-1}\bar{r}_i(k) + 2\ln\left(\frac{|S_i|}{|S_T|}\right) + 2\ln\left(\frac{\bar{\mu_T}(k)}{\bar{\mu_i}(k)}\right) > 0.$
2. $\bar{r}_T(k)^T S_T^{-1}\bar{r}_T(k) < \bar{r}_i(k)^T S_i^{-1}\bar{r}_i(k) + 2\ln\left(\frac{|S_i|}{|S_T|}\right) + 2\ln\left(\frac{\bar{\mu_T}(k)}{\bar{\mu_i}(k)}\right).$
3. $\|\bar{r}_T(k)\|^2 < \frac{\lambda_{min}(S_i^{-1})}{\lambda_{max}(S_T^{-1})}\|\bar{r}_i(k)\|^2 + \frac{2}{\lambda_{max}(S_T^{-1})}\left[\ln\left(\frac{|S_i|}{|S_T|}\right) + \ln\left(\frac{\bar{\mu_T}(k)}{\bar{\mu_i}(k)}\right)\right].$

Condition 3 in **Proposition 2** is a sufficient condition on, and might be a very conservative test for, the correct mode detection. However, Condition 3 gives valuable insight into the performance of hybrid estimation. Fast mode detection is dependent not only on the magnitudes of the residuals produced by each Kalman filter but also on the residual covariances. If $\frac{\lambda_{min}(S_i^{-1})}{\lambda_{max}(S_T^{-1})}$ is small and/or $\frac{|S_i|}{|S_T|}$ is small, it is difficult for Condition 3 to hold and thus to detect the correct mode. Therefore, by checking the eigenvalues of S_i^{-1} and the determinant of its inverse, we can tell which mode transitions are more detectable than the others. This is similar to the idea of the observability grammian as a measure of which states are more observable than others [43]. If we consider the steady-state mean of the residual, Condition 3 becomes

$$\|\bar{r}_{T_{ss}}\|^2 < \frac{\lambda_{min}(S_i^{-1})}{\lambda_{max}(S_T^{-1})}\|\bar{r}_{i_{ss}}\|^2 + \frac{2}{\lambda_{max}(S_T^{-1})}\left[\ln\left(\frac{|S_i|}{|S_T|}\right) + \ln\left(\frac{\bar{\mu_{T_{ss}}}}{\bar{\mu_{i_{ss}}}}\right)\right],\ \forall i \neq T \tag{2.50}$$

Therefore, if the asymptotic behavior of the residuals satisfies (2.50) and the minimum sojourn time is long enough for the residual to converge to its steady-state value, then MMAE is guaranteed to estimate hybrid states correctly.

We then derive the mode estimation delay δ_i using Condition 3 in **Proposition 2**. The mean residual of the correct filter at time $k_l + \delta_i$ ($l \in \mathbb{N}^+$), when $i = T$, are:

$$\bar{r}_T(k_l + \delta_T) \quad = \quad C_T A_T [(I - K_T C_T) A_T]^{\delta_T - 1} \bar{e}_T(k_l) \tag{2.51}$$

The mean residual and the mean estimation error of the incorrect filter i ($i \neq T$) at time k are

$$\begin{aligned}
\bar{r}_i(k) &= C_T A_T \bar{e}_i(k-1) + [C_T \Delta A_i + \Delta C_i A_T - \Delta C_i \Delta A_i] \hat{x}_i(k-1) \\
&\quad + [C_T \Delta B_i + \Delta C_i B_T - \Delta C_i \Delta B_i] u(k-1) \\
&= \left[\begin{array}{ccc} C_T A_T & C_T \Delta A_i + \Delta C_i A_T - \Delta C_i \Delta A_i & C_T \Delta B_i + \Delta C_i B_T - \Delta C_i \Delta B_i \end{array} \right] \\
&\quad \cdot \left[\begin{array}{ccc} \bar{e}_i(k-1) & \hat{x}_i(k-1) & u(k-1) \end{array} \right]^T \\
\bar{e}_i(k) &= (I - K_i C_T) A_T \bar{e}_i(k-1) + [(I - K_i C_T) \Delta A_i - K_i \Delta C_i A_i] \hat{x}_i(k-1) \\
&\quad + [(I - K_i C_T) \Delta B_i - K_i \Delta C_i B_i] u(k-1) \\
&= \left[\begin{array}{ccc} (I - K_i C_T) A_T & (I - K_i C_T) \Delta A_i - K_i \Delta C_i A_i & (I - K_i C_T) \Delta B_i - K_i \Delta C_i B_i \end{array} \right] \\
&\quad \cdot \left[\begin{array}{ccc} \bar{e}_i(k-1) & \hat{x}_i(k-1) & u(k-1) \end{array} \right]^T
\end{aligned} \tag{2.52}$$

For the sake of notational simplicity, we define

$$\begin{aligned}
F_T &:= (I - K_T C_T) A_T \\
F_i &:= (I - K_i C_T) A_T \\
H_i^x &:= C_T \Delta A_i + \Delta C_i A_T - \Delta C_i \Delta A_i \\
H_i^u &:= C_T \Delta B_i + \Delta C_i B_T - \Delta C_i \Delta B_i \\
G_i^x &:= (I - K_i C_T) \Delta A_i - K_i \Delta C_i A_i \\
G_i^u &:= (I - K_i C_T) \Delta B_i - K_i \Delta C_i B_i \\
L_i &:= \left[\begin{array}{ccc} C_T A_T & H_i^x & H_i^u \end{array} \right]
\end{aligned} \tag{2.53}$$

The norm of the mean residual of the correct filter at time $k_l + \delta_T$ is

$$\|\bar{r}_T(k_l + \delta_T)\| = \|C_T A_T F_T^{\delta_T - 1} \bar{e}_T(k_l)\| \leq \bar{\sigma}(C_T A_T) \bar{\sigma}(F_T)^{\delta_T - 1} \|\bar{e}_T(k_l)\| \tag{2.54}$$

where $\bar{\sigma}(\cdot)$ denotes the maximum singular value. Similarly, from (2.52), the norm of the

mean residual of the incorrect filter at time $k_l + \delta_T$ is

$$\|\bar{r}_i(k_l+\delta_T)\| \geq \underline{\sigma}(L_i)\|\bar{e}_i(k_l+\delta_T-1)\| \geq \underline{\sigma}(L_i)\underline{\sigma}\left(\begin{bmatrix} F_i & G_i^x & G_i^u \end{bmatrix}\right)^{\delta_T-1}\|\bar{e}_i(k_l)\| \quad (2.55)$$

where $\underline{\sigma}(\cdot)$ denotes the minimum singular value. We define

$$\begin{array}{rcl} \alpha & := & \frac{\lambda_{min}(S_i^{-1})}{\lambda_{max}(S_T^{-1})} \\ \beta(k) & := & \frac{2}{\lambda_{max}(S_T^{-1})}\left[\ln\left(\frac{|S_i|}{|S_T|}\right)+\ln\left(\frac{\mu_T^-(k)}{\mu_i^-(k)}\right)\right] \end{array} \quad (2.56)$$

Using (2.54), (2.55), and (2.56) in Condition 3 of **Proposition 2**, we obtain the following condition:

$$\begin{array}{rcl} \bar{\sigma}(C_TA_T)^2\bar{\sigma}(F_T)^{2(\delta_T-1)}\|\bar{e}_T(k_l)\|^2 & < & \alpha\underline{\sigma}(L_i)^2\underline{\sigma}\left(\begin{bmatrix} F_i & G_i^x & G_i^u \end{bmatrix}\right)^{2(\delta_T-1)}\|\bar{e}_i(k_l)\|^2 \\ & & +\beta(k+\delta_T) \end{array} \quad (2.57)$$

Even though we cannot find the mode estimation delay δ_T explicitly from (2.57), (2.57) will be used to derive a condition for instantaneous mode estimation. To find δ_T explicitly, we try a different approach. From (2.52), the mean residual of the incorrect filter at time $k_l + \delta_T$ can be written as:

$$\begin{array}{rcl} \bar{r}_i(k_l+\delta_T) & = & C_TA_TF_i^{\delta_T-1}\bar{e}_i(k_l)+H_i^x\hat{x}_i(k_l+\delta_T-1) \\ & & +C_TA_T[F_i^{\delta_T-2}G_i^x\hat{x}_i(k_l)+\cdots+G_i^x\hat{x}_i(k_l+\delta_T-2)]+H_i^u u(k_l+\delta_T-1) \\ & & +C_TA_T[F_i^{\delta_T-2}G_i^u u(k_l)+\cdots+G_i^u u(k_l+\delta_T-2)] \end{array} \quad (2.58)$$

We define

$$\begin{array}{rcl} b_i(k_l+\delta_T-1) & := & H_i^x\hat{x}_i(k_l+\delta_T-1)+C_TA_T[F_i^{\delta_T-2}G_i^x\hat{x}_i(k_l)+\cdots+G_i^x\hat{x}_i(k_l+\delta_T-2)] \\ & & +H_i^u u(k_l+\delta_T-1)+C_TA_T[F_i^{\delta_T-2}G_i^u u(k_l)+\cdots+G_i^u u(k_l+\delta_T-2)] \end{array} \quad (2.59)$$

Then, Condition 3 of **Proposition 2** becomes

$$\bar{\sigma}(C_TA_T)^2\bar{\sigma}(F_T)^{2(\delta_T-1)}\|\bar{e}_T(k_l)\|^2 < \alpha\underline{\sigma}(C_TA_T)^2\underline{\sigma}(F_i)^{2(\delta_i-1)}\|\bar{e}_i(k_l)\|^2+J_i(k_l+\delta_T) \quad (2.60)$$

where $J_i(k_l+\delta_T) := \alpha\|b_i(k_l+\delta_T-2)\|^2+\beta(k_l+\delta_T)$.
$\beta(k_l+\delta_T) = \frac{2}{\lambda_{max}(S_T^{-1})}\left[\ln\left(\frac{|S_i|}{|S_T|}\right)+\ln\left(\frac{\mu_T^-(k_l+\delta_T)}{\mu_i^-(k_l+\delta_T)}\right)\right]$ might be negative, yet its magnitude is

usually not large because it is in a logarithmic scale. Thus, $J_i(k_l+\delta_T) \geq 0$ is easily satisfied. If $J_i(k_l+\delta_T) \geq 0$, we can have the following condition:

$$\delta_T > 1 + \left\{ 2\ln\left[\frac{\underline{\sigma}(F_i)}{\bar{\sigma}(F_T)}\right] \right\}^{-1} \left\{ -\ln\alpha + 2\ln\left[\frac{\bar{\sigma}(C_T A_T)}{\underline{\sigma}(C_T A_T)}\right] + 2\ln\left[\frac{\|\bar{e}_T(k_l)\|}{\|\bar{e}_i(k_l)\|}\right] \right\} \tag{2.61}$$

Proposition 3. *The correct mode can be detected δ_T time steps after a mode transition if Condition 1 of **Proposition 2** holds and there exists $\delta_T \in \mathbb{N}^+$, $\delta_T < k_{l+1} - k_l$, $l \in \mathbb{N}^+$, $\forall i \neq T$, such that either of the following conditions is true.*

1. $\bar{\sigma}(C_T A_T)^2 \bar{\sigma}(F_T)^{2(\delta_T-1)} \|\bar{e}_T(k_l)\|^2 < \alpha \underline{\sigma}(L_i)^2 \underline{\sigma}\left(\begin{bmatrix} F_i & G_i^x & G_i^u \end{bmatrix}\right)^{2(\delta_T-1)} \|\bar{e}_i(k_l)\|^2 + \beta(k+\delta_T)$
2. $\delta_T > 1 + \left\{ 2\ln\left[\frac{\underline{\sigma}(F_i)}{\bar{\sigma}(F_T)}\right] \right\}^{-1} \left\{ -\ln\alpha + 2\ln\left[\frac{\bar{\sigma}(C_T A_T)}{\underline{\sigma}(C_T A_T)}\right] + 2\ln\left[\frac{\|\bar{e}_T(k_l)\|}{\|\bar{e}_i(k_l)\|}\right] \right\}$, *when* $J_i(k_l+\delta_T) \geq 0$.

For a system with only two discrete modes, we can further simplify (2.60) and also relax the condition that $J_i(k_l+\delta_T) \geq 0$. To do this, we first consider (2.47). For a system with two modes, for any time $k \in [k_l, k_{l+1})$, if T is the true mode in the interval $[k_l, k_{l+1})$,

$$\frac{\mu_i(k+1)}{\mu_T(k+1)} = \frac{\Lambda_i(k+1)}{\Lambda_T(k+1)} \left(\frac{\pi_{ii}\mu_i(k) + (1-\pi_{TT})\mu_T(k)}{(1-\pi_{ii})\mu_i(k) + \pi_{TT}\mu_T(k)} \right) \tag{2.62}$$

Let $\Omega(k) = \frac{\mu_i(k)}{\mu_T(k)}$. Then, for a diagonally dominant transition matrix Π,

$$\Omega(k+1) \;=\; \frac{\Lambda_i(k+1)}{\Lambda_T(k+1)} \left(\frac{\pi_{ii}\Omega(k) + (1-\pi_{TT})}{(1-\pi_{ii})\Omega(k) + \pi_{TT}} \right) \approx \frac{\Lambda_i(k+1)}{\Lambda_T(k+1)} \left(\frac{\pi_{ii}\Omega(k)}{(1-\pi_{ii})\Omega(k) + \pi_{TT}} \right) \tag{2.63}$$

We assume that the estimator converges before the transition takes place, which gives us

$$\mu_i(k_l - 1) \;\approx\; 1 \tag{2.64}$$

$$\mu_T(k_l - 1) \;\approx\; 0 \tag{2.65}$$

$$\Omega(k_l) \;=\; \frac{\Lambda_i(k_l)}{\Lambda_T(k_l)} \left(\frac{\pi_{ii}}{1-\pi_{ii}} \right) \tag{2.66}$$

After the transition at time k_l and before detection,

$$\begin{array}{rccccc} & 1 & \leq & \Omega(k) & \leq & \infty \\ \Longrightarrow & \frac{\Lambda_i(k+1)}{\Lambda_T(k+1)} \left(\frac{\pi_{ii}}{(1-\pi_{ii})+\pi_{TT}} \right) & \leq & \Omega(k+1) & \leq & \frac{\Lambda_i(k+1)}{\Lambda_T(k+1)} \left(\frac{\pi_{ii}}{1-\pi_{ii}} \right) \end{array} \tag{2.67}$$

Detection occurs at the smallest δ_T when $\Omega(k_l+\delta_T)<1$. Thus we try to find the smallest δ_T such that

$$\begin{array}{ll} & \frac{\Lambda_i(k_l+\delta_T)}{\Lambda_T(k_l+\delta_T)}\left(\frac{\pi_{ii}}{1-\pi_{ii}}\right)<1 \\ \Longrightarrow & \bar{r}_T(k_l+\delta_T)^T S_T^{-1}\bar{r}_T(k_l+\delta_T)<\bar{r}_i(k_l+\delta_T)^T S_i^{-1}\bar{r}_i(k_l+\delta_T)+2\ln\left(\frac{|S_i|}{|S_T|}\right) \\ & -2\ln\left(\frac{\pi_{ii}}{1-\pi_{ii}}\right) \end{array} \tag{2.68}$$

Using (2.54), (2.55), and (2.56), we obtain the following condition:

Proposition 4. *For a hybrid system with two discrete modes, the correct mode can be detected δ_T time steps after a mode transition if there exists $\delta_T\in\mathbb{N}^+$, $\delta_T<k_{l+1}-k_l$, $l\in\mathbb{N}^+$, $i\neq T$, such that*

$$\begin{array}{rcl} \bar{\sigma}(C_TA_T)^2\bar{\sigma}(F_T)^{2(\delta_T-1)}\|\bar{e}_T(k_l)\|^2 & < & \alpha\underline{\sigma}(L_i)^2\underline{\sigma}\left(\begin{bmatrix} F_i & G_i^x & G_i^u \end{bmatrix}\right)^{2(\delta_T-1)}\|\bar{e}_i(k_l)\|^2 \\ & & +\frac{2}{\lambda_{max}(S_T^{-1})}\left[\ln\left(\frac{|S_i|}{|S_T|}\right)-\ln\left(\frac{\pi_{ii}}{1-\pi_{ii}}\right)\right] \end{array} \tag{2.69}$$

Proposition 4 implies that if (2.69) is satisfied, then the mode probability of the correct mode is greater than those of the other modes, δ_T time steps after a mode transition at time k_l. Thus, the correct mode is detected δ_T time steps after a mode transition.

Now, we consider the conditions under which the mode change detection is instantaneous. Consider the case of a system with two modes. Assuming that the time between discrete transitions is sufficient to allow the Kalman filters to converge, we can assume that the mode probabilities before the transition have converged. Following a procedure similar to (2.62)-(2.69), we obtain

$$\begin{array}{rcl} \bar{\sigma}(C_TA_T)^2\|\bar{e}_T(k_l-1)\|^2 & < & \alpha\underline{\sigma}(L_i)^2\underline{\sigma}\left(\begin{bmatrix} F_i & G_i^x & G_i^u \end{bmatrix}\right)^2\|\bar{e}_i(k_l-1)\|^2 \\ & & +\frac{2}{\lambda_{max}(S_T^{-1})}\left[\ln\left(\frac{|S_i|}{|S_T|}\right)-\ln\left(\frac{\pi_{ii}}{1-\pi_{ii}}\right)\right] \end{array} \tag{2.70}$$

We can extend this to the instantaneous detection of a transition in a system with n modes.

Proposition 5. *The correct mode is detected instantaneously if the following condition holds ($\forall k_l,\ l\in\mathbb{N}^+$):*

$$\begin{array}{rcl} \bar{\sigma}(C_TA_T)^2\|\bar{e}_T(k_l-1)\|^2 & < & \alpha\underline{\sigma}(L_i)^2\underline{\sigma}\left(\begin{bmatrix} F_i & G_i^x & G_i^u \end{bmatrix}\right)^2\|\bar{e}_i(k_l-1)\|^2 \\ & & +\frac{2}{\lambda_{max}(S_T^{-1})}\left[\ln\left(\frac{|S_i|}{|S_T|}\right)+\ln\left(\frac{\pi_{jT}}{\pi_{jj}}\right)_{min}\right] \end{array} \tag{2.71}$$

where $\left(\frac{\pi_{jT}}{\pi_{jj}}\right)_{min}$ *is the smallest ratio of off-diagonal to diagonal elements in any row of the transition matrix.*

Along with **Theorem 2**, **Propositions 2-5** present the conditions under which, in the event of a mode detection delay δ, the sojourn time is long enough for the error convergence during the period of correct detection $(\beta-\delta)$ to balance the divergence of the error during the mode mismatch. From the present results, we now have a way of determining δ. Therefore, combining the two results, we can evaluate the performance of a given hybrid estimator and also find the minimum sojourn time required in each mode to guarantee exponential convergence of the mean-square error.

2.2.5 Performance comparison between hybrid estimation algorithms

In this section, we discuss the performance of hybrid estimation and also compare the performance of the MMAE and the IMM algorithms. We focus on the mode estimation delay since usually, the smaller the mode estimation delay, the smaller the estimation error. By investigating (2.61), we can explain the performance of hybrid estimation algorithms qualitatively. For the mode estimation delay to be small, the following must be small if $J_i(k_l+\delta_T)\geq 0$:

$$\left\{\log\left(\frac{\lambda_{max}(S_T^{-1})}{\lambda_{min}(S_i^{-1})}\right)+2\log\left[\frac{\bar{\sigma}(C_TA_T)}{\underline{\sigma}(C_TA_T)}\right]+2\log\left[\frac{\|\bar{e}_T(k_l)\|}{\|\bar{e}_i(k_l)\|}\right]\right\},\ \ (\forall T\in\{1,\cdots,N\},\ \forall i\neq T) \tag{2.72}$$

where mode T is the correct mode after the mode transition at time k_l ($l\in\mathbb{N}^+$). Firstly, $\frac{\lambda_{max}(S_T^{-1})}{\lambda_{min}(S_i^{-1})}$ must be small. Here, the pre-computed residual covariance $S_i=C_iP_i^{ss}C_i+R_i$, and P_i^{ss} (the steady-state error covariance matrix computed by Kalman filter i) satisfies the algebraic Riccati equation. Therefore, $\frac{\lambda_{max}(S_T^{-1})}{\lambda_{min}(S_i^{-1})}$ depends only on the system parameters A_i, C_i, Q_i, R_i and A_T, C_T, Q_T, R_T. Thus, by checking the residual covariance matrices for each Kalman filter (which can be done without any measurements), we can tell which mode transition is more detectable than the others. In addition, since Q_i and R_i are design parameters for the Kalman filter i, and Q_T and R_T are design parameters for the Kalman filter T, we can make $\frac{\lambda_{max}(S_T^{-1})}{\lambda_{min}(S_i^{-1})}$ small by adjusting these parameters (also known as Kalman filter tuning) and thus reduce the mode estimation delay. Secondly, if the condition number of C_TA_T is close to 1, the second term becomes small. Thus, we also say which mode

is more easily estimated than the others by checking the condition number of $C_T A_T$ for all T. Thirdly, $\frac{\|\bar{e}_T(k_l)\|}{\|\bar{e}_i(k_l)\|}$ must be small, i.e., the mean state estimation errors produced by mode-mismatched Kalman filters should be small (and close to the error produced by the correct Kalman filter).

The mixing step in IMM was originally devised to reduce the complexity of the algorithm, yet it also keeps the estimation errors produced by mismatched Kalman filters small. The IMM algorithm readjusts, through the mixing step at each time instant, the initial conditions for each Kalman filter, and shifts them closer to the (correct) estimate computed by IMM at the previous time step. Therefore, the means of the state estimation errors produced by the incorrect Kalman filters are close to that of the correct Kalman filter. Thus, the mode estimation delay of IMM is smaller than that of MMAE (which does not have this mixing mechanism). The smaller time delay translates to better estimation performance of IMM compared to MMAE. Maybeck [55] proposes two ad hoc methods to improve adaptability of MMAE: enforcing a lower bound on the mode probabilities and adding pseudonoise to the the Kalman filter models. IMM does both inherently. We illustrate this through examples in the next section.

2.2.5.1 Examples

MMAE and IMM Performance vs. Parameters (Monte Carlo Simulation, 100 trials)														
Case	Algorithm	Mode Parameters ($b_1 = b_2 = 1$)				Instant Detection Condition		Mode detection delay				$\frac{\lambda_{max}(S_T^{-1})}{\lambda_{min}(S_i^{-1})}$	
								Predicted		Observed			
		a1	a2	c1	c2	$2 \to 1$	$1 \to 2$	δ_1^*	δ_2^*	δ_1	δ_2	T=1	T=2
1	MMAE	0.95	0.25	1	0.8	$\surd$	$\times$	0	2	0	2	0.48	2.09
	IMM					$\surd$	$\times$	0	2	0	2		
2	MMAE	0.85	0.85	0.8	0.2	$\times$	$\times$	6	2	4	0	1.29	0.77
	IMM					$\times$	$\surd$	2	0	0	0		
3	MMAE	0.95	0.85	1	0.4	$\times$	$\times$	10	2	5	1	1.13	0.89
	IMM					$\surd$	$\surd$	0	0	0	0		

Table 2.1: Performance comparison between hybrid estimation algorithms: two mode example

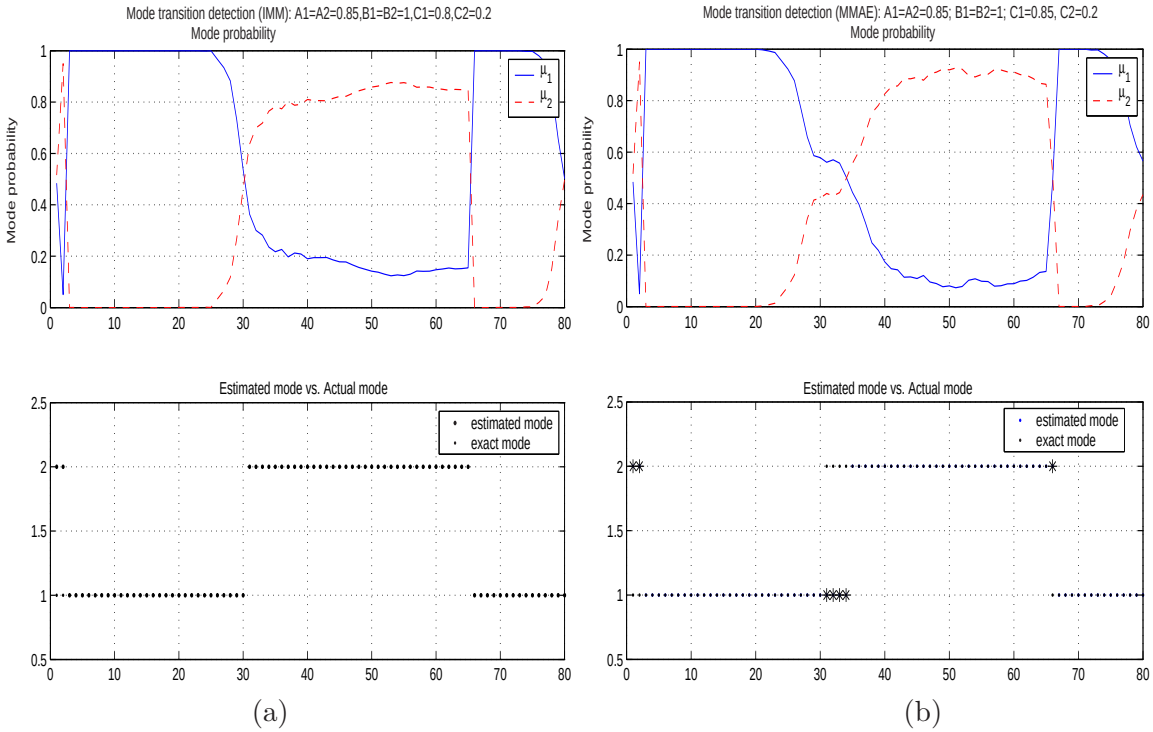
MMAE and IMM Performance vs. Parameters (Monte Carlo Simulation, 100 trials)																	
Case	Algorithm	Mode Parameters ($b_1 = b_2 = b3 = 1$)						Mode detection delay						max $\frac{\lambda_{max}(S_T^{-1})}{\lambda_{min}(S_i^{-1})}$		
								Predicted			Observed					
		a1	a2	a3	c1	c2	c3	δ_1^*	δ_2^*	δ_3^*	δ_1	δ_2	δ_3	T=1	T=2	T=3
1	MMAE	1.2	0.25	0.95	0.8	1.0	0.8	2	0	7	1	0	5	1.2	0.80	1.26
	IMM							1	0	5	1	0	4			

Table 2.2: Performance comparison between hybrid estimation algorithms: three mode example

We first consider mode detection in a simple, one-dimensional system such as the one in [20]. The dynamics is of the form

$$\begin{aligned} x(k) &= a_i x(k-1) + b_i u(k) + w_i(k) \\ y(k) &= c_i x(k) + v_i(k) \\ u(k) &= 5cos(\frac{2\pi t}{100}) \end{aligned}$$

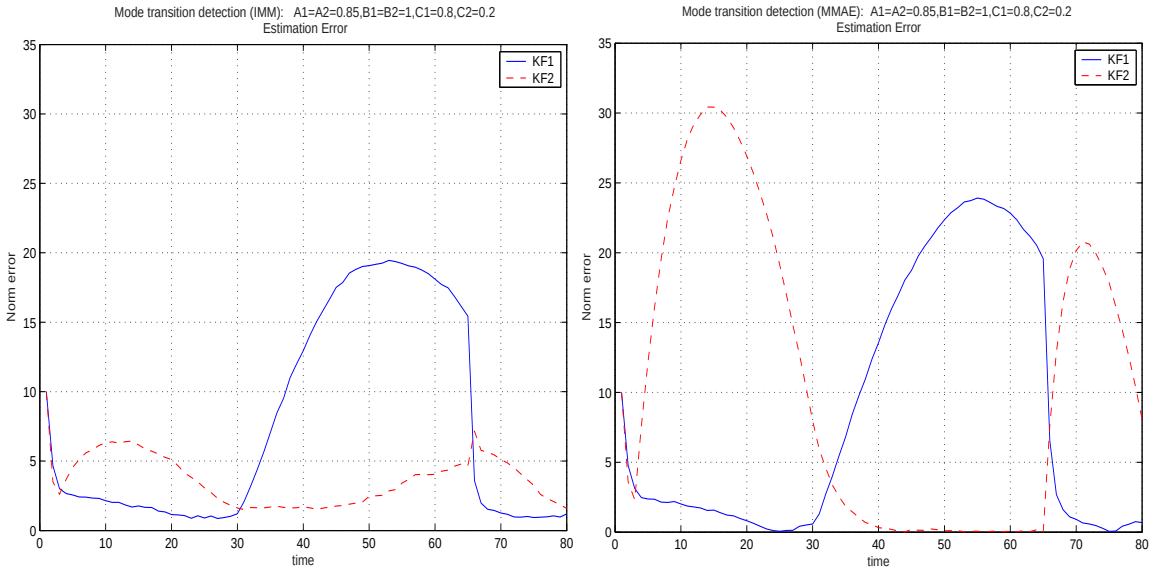
where the state variables and model parameters are scalar, there are 2 discrete modes,



(a) (b)

Figure 2.5: Mode estimates by (a) IMM and (b) MMAE: 100 trial Monte Carlo simulation results. The asterisks denote mode estimation delay.

and the input is deterministic and sinusoidal. We estimate the hybrid state sequence from the output sequence using both the MMAE and the IMM algorithms. We first check for instantaneous mode detection at a switch using (2.70). We then compute the maximum mode detection delay (or the minimum sojourn time needed to guarantee correct mode detection) using (2.61) and (2.69). We perform this experiment for various values of the model parameters and compare our predictions with the simulations. Clearly, IMM performs

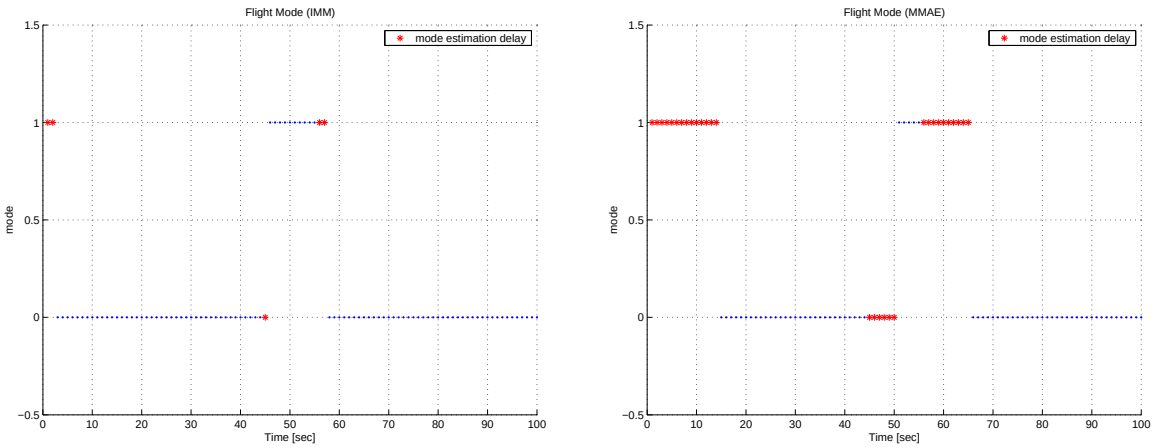


(a) (b)

Figure 2.6: Estimation error by (a) IMM and (b) MMAE: 100 trial Monte Carlo simulation results.

better than MMAE, especially in cases 2 and 3. Also, since we only compute a conservative estimate of the mode detection delay, it is quite possible that the observed delay is less than the computed bound (as in cases 2 and 3). Figure 2.5 shows the mode probabilities and estimates for case 2. The reason for the difference in the performance of the MMAE and IMM algorithms is clear when we consider the estimation errors in Figure 2.6. At the mode transition times, the errors of the matched and mismatched filters of IMM are almost equal, thus keeping (2.72), and therefore the mode detection delay, small. In our example, (2.72) simply reduces to $\frac{\lambda_{max}(S_T^{-1})}{\lambda_{min}(S_i^{-1})}$ in the case of IMM estimation. Comparing these values for the different transitions with the values of the delay in Table 2.1, it is seen that as predicted, the smaller the value of (2.72), the smaller the mode detection delay. The biggest advantage of this result is that given a system and its error bounds, this gives us a way to determine *a priori* transitions to which modes are the most detectable. We also try a three-mode example for the same system. The results are as expected, and are shown in Table 2.2.

We now consider an aircraft tracking example and recall the aircraft model derived in Section 2.1.3 which has two discrete states, the constant velocity (CV) mode and the coordinated turn (CT) mode. This represents flight trajectories composed of straight lines and circular arcs. For brevity, we only include the two mode example but the performance analysis



(a) (b)

Figure 2.7: Mode estimates by (a) IMM and (b) MMAE: 100 trial Monte Carlo simulation results (mode CV = 0, mode CT = 1).

conditions can be applied to multiple-mode cases. In this example, we use a simpler version of the aircraft model in Section 2.1.3 for simple analysis. The dynamics of both modes is given by

$$x(k+1) = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} T^2/2 & 0 \\ T & 0 \\ 0 & T^2/2 \\ 0 & T \end{bmatrix} u_i(k) + w_i(k) \tag{2.73}$$

$$y(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x(k) + v_i(k), \qquad (i \in \{CV, CT\}) \tag{2.74}$$

where $x = [\ x_1 \quad \dot{x}_1 \quad x_2 \quad \dot{x}_2\ ]^T$ where x_1 and x_2 are the position coordinates, $u = [\ u_1 \quad u_2\ ]^T$ where u_1 and u_2 are the acceleration components. The control input has a different constant value for each mode:

$$u_{CV} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \text{ for CV mode, } \quad u_{CT} = \begin{bmatrix} 1.5 \\ 1.5 \end{bmatrix} \text{ for CT mode} \tag{2.75}$$

T is the sampling interval, w_i is the process noise, and v_i is the sensor noise. We choose an operating velocity of 150 knots. We design a flight trajectory such that the mode change

from CV mode to CT mode occurs at time=45 seconds and the mode change CT mode to CV mode occurs at time=56 seconds. Using **Proposition 3** for IMM, we find that the mode estimation delay for the mode switching from mode CV to mode CT is $\delta_{ct} = 1$ and the mode estimation delay for the mode switching from mode CT to mode CV is $\delta_{cv} = 2$. Thus, we expect the mode switching from mode CV to CT to be more detectable than the mode switching from mode CT to CV. Similarly, we obtain $\delta_{ct} = 7$ and $\delta_{cv} = 12$ for MMAE. Figure 2.7-(a) shows that IMM detects the mode change from mode CV to mode CT after a 1 time step delay and the mode change from mode CT to mode CV after a 2 time step delay. Similarly, Figure 2.7-(b) shows that MMAE detects the mode change from mode CV to mode CT after a 6 time step delay and the mode change from mode CV to mode CT after a 10 time step delay. The performance analysis predicts that IMM performs better than MMAE and the simulation results support this.

2.3 Residual-Mean Interacting Multiple Model Algorithm (RMIMM)

In this section, using the results of the performance analysis of hybrid estimation algorithms, we propose a modified IMM algorithm called the Residual-Mean Interacting Multiple Model (RMIMM), which has a likelihood function that uses the properties of the mean of the residual produced by each Kalman filter. Using the idea proposed in Hanlon and Maybeck [69] that the residual produced by each Kalman filter in the IMM algorithm should have a non-zero mean if the Kalman filter is not the correct one, we design a likelihood function that gives clearer and sharper differences between the correct mode and the other modes, so that the number of false mode estimates decreases when compared to the standard IMM algorithm. This property is useful since the standard IMM algorithm may frequently produce incorrect mode estimates as the number of modes in the model increases. In this section, we consider an autonomous discrete-time stochastic linear system (2.33)-(2.34), i.e., a system without control input ($u(k) \equiv 0$).

First, we investigate the IMM algorithm discussed briefly in Section 2.2.2 (as shown in Figure 2.4) in detail. The components of IMM are defined as follows:

1. **Mixing probability:** This is the probability that the system is in mode i at time $k-1$, given that it is in mode j at time k ($i, j \in \{1, \cdots, N\}$):

$$\mu_{ij}(k-1|k-1) = \frac{1}{c_j}\pi_{ji}\mu_i(k-1) \tag{2.76}$$

where c_j is a normalization constant, and where $\mu_i(k)$ is the mode probability of mode i at time k, i.e., a measure of how probable it is that the system is in mode i at time k. The initial condition $\mu_i(0)$ is assumed given, and is usually obtained from properties of the system.

2. **New initial states and covariances:** The input to each Kalman filter is adjusted by weighting the output of each Kalman filter with the mixing probability as the weight:

$$\begin{aligned}
\hat{x}_{0j}(k-1) &= \sum_i \hat{x}_i(k-1)\mu_{ij}(k-1|k-1) \\
P_{0j}(k-1) &= \sum_i \{P_i(k-1) + [\hat{x}_i(k-1) - \hat{x}_{0j}(k-1)][\hat{x}_i(k-1) \\
&\quad - \hat{x}_{0j}(k-1)]^T\}\mu_{ij}(k-1|k-1)
\end{aligned}$$

where $\hat{x}_i(k-1)$ and $P_i(k-1)$ are the state estimate and its covariance produced by Kalman filter i after the measurement update at time $k-1$.

3. **Kalman Filter:** N Kalman filters run in parallel (multiple-model-based (hybrid) estimation).

4. **Mode likelihood functions:** The likelihood function of mode j is a measure of how likely it is that the model used in Kalman filter j is the correct one; it is computed with the residual and its covariance produced by Kalman filter j:

$$\Lambda_j(k) = \mathcal{N}(r_j(k); 0, S_j(k)) \tag{2.77}$$

5. **Mode probabilities:** The probability of mode j is a measure of how probable it is that the system is in mode j:

$$\mu_j(k) = \frac{1}{c}\Lambda_j(k)\sum_i \pi_{ji}\mu_i(k-1) \tag{2.78}$$

where c is a normalization constant. The probability of each mode is updated using the likelihood function.

6. **Combination (output of the IMM algorithm):** The state estimate is a weighted sum of the estimates from N Kalman filters and the mode estimate is the mode which has the highest mode probability:

$$\begin{array}{rcl} \hat{x}(k) & = & \sum_j \hat{x}_j(k)\mu_j(k) \\ P(k) & = & \sum_j \{P_j(k) + [\hat{x}_j(k) - \hat{x}(k)][\hat{x}_j(k) - \hat{x}(k)]^T\}\mu_j(k) \\ \hat{m}(k) & = & \arg\max_j \mu_j(k) \end{array} \tag{2.79}$$

where $\hat{m}(k)$ is the mode estimate at time k.

As can be seen from the IMM algorithm and the performance analysis of hybrid estimation algorithms in Section 2.2, the mode probability in (2.78) strongly depends on the likelihood function Λ_j. Thus, if the likelihoods of the modes are close to each other, the mode estimate may be inaccurate. Inaccurate mode estimates could produce poor state estimates, degrading the tracking accuracy. Because we are interested in using this for conflict detection, we propose a method which reduces false mode estimation by increasing the difference between the likelihood of the correct mode and the likelihoods of the other modes, using the fact that if the Kalman filter corresponding to mode j is the correct one, then the residual in (2.77) should be a white Gaussian process with a zero mean. Otherwise, its mean should not be zero.

If we consider a discrete-time stochastic linear hybrid system (2.33) without the control input ($u(k) \equiv 0$), from (2.52), the mean of the residual is

$$\bar{r}_j(k) = C_T A_T \bar{e}_j(k-1) + (C_T \Delta A_j + \Delta C_j A_T - \Delta C_j \Delta A_j)\hat{x}_j(k-1) \tag{2.80}$$

and the mean of estimation error is

$$\begin{array}{rcl} \bar{e}_j(k) & = & (I - K_j(k)C_T)A_T\bar{e}_j(k-1) \\ & & +((I - K_j(k)C_T)\Delta A_j - K_j(k)\Delta C_j A_j)\hat{x}_k(k-1) \end{array} \tag{2.81}$$

where $K_j(k)$ is the Kalman filter gain for Kalman filter j. This is a recursive equation with respect to the state estimation error. Thus, the mean of the residual is computed from (2.80) and (2.81).

To the best of our knowledge, all multiple-model-based estimation and learning algorithms including various IMM algorithms use a likelihood function whose mean is zero to determine the current mode in which the system lies [71, 52, 72]. We propose RMIMM, which uses the mean of the residual to increase the difference between the likelihood of the correct mode and those of the other modes, thereby decreasing the number of false mode estimates. Since in the IMM framework there is no true model for sure, i.e., with probability 1, we propose a new definition of the mean of the residual: a weighted sum of the mean of the residual computed by each Kalman filter with the mode probability estimate in (2.78) as the weight. Similarly, a new definition of the mean of the state estimation error is proposed as a weighted sum of the mean of the state estimation error corresponding to Kalman filter j with the same weight.

$$\begin{aligned} \bar{r}_j(k) &:= \textstyle\sum_{j=1}^{N}\{C_T A_T \bar{e}_j(k-1) + (C_T \Delta A_j + \Delta C_j A_T - \Delta C_j \Delta A_j)\hat{x}_j(k-1)\}\mu_j^-(k) \\ \bar{e}_j(k) &:= \textstyle\sum_{j=1}^{N}\{(I - K_j(k)C_T)A_T \bar{e}_j(k-1) \\ &\quad +((I - K_j(k)C_T)\Delta A_j - K_j(k)\Delta C_j A_j)\hat{x}_k(k-1)\}\mu_j^-(k) \end{aligned} \tag{2.82}$$

If the mode probability of mode j is large, the mean of the residual becomes small (i.e., close to zero) because the other mode probabilities $\mu_i(k)$ for $\forall i \neq j$ are small (the residual has a zero mean if the Kalman filter is the correct one). Since the proposed mean of the residual is small if the mode probability of the corresponding Kalman filter is small, and large if the mode probability of the corresponding Kalman filter is large, we can use the mean of the residual in (2.82) to make the likelihood of the correct mode more distinct from those of the other modes. Therefore, using the mean of the residual provided by each Kalman filter, we propose a new likelihood function:

$$\Lambda_j^{new}(k) = \begin{cases} \frac{N_j(k)\Lambda_j(k)}{\sum_{i=1}^{N} N_i(k)\Lambda_i(k)} & \text{if } \bar{r}_j(k) \neq 0 \\ \Lambda_j(k) & \text{otherwise} \end{cases} \tag{2.83}$$

where

$$N_i(k) = \begin{cases} \|\bar{r}_i(k)\|^{-1} & \text{if } \bar{r}_i(k) \neq 0 \\ 1 & \text{otherwise} \end{cases}$$

Proposition 6. *The difference between the new likelihood function (2.83) for the correct mode and those for the incorrect mode, is greater than the corresponding difference using the previous likelihood function from (2.77).*

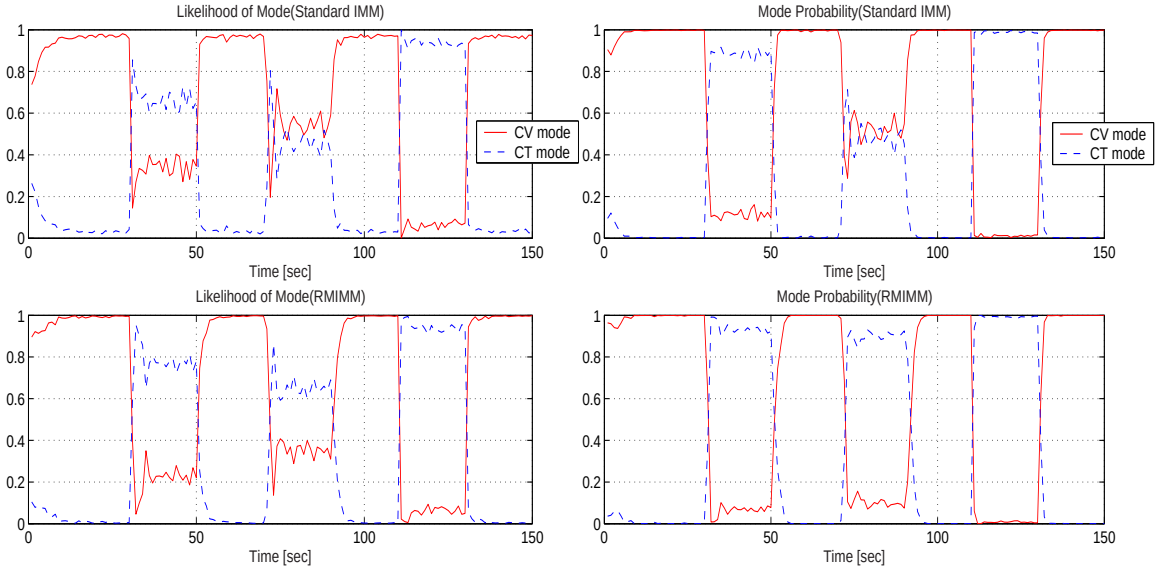
Proof. If the model in Kalman filter j is incorrect, the mean of residual is not zero and the likelihood of mode j from the new likelihood function in (2.83) is less than that of the standard likelihood function in (2.77). If the model in Kalman filter j is correct, the likelihood of mode j from the new likelihood function is the same as that of the standard likelihood function in (2.77). Thus, the differences between the likelihood of the correct mode and those of incorrect modes are greater and the result follows. □

2.3.1 Example: Tracking a single aircraft

We demonstrate the performance of the proposed RMIMM algorithm through an aircraft tracking example, using the aircraft model described in (2.73)-(2.29) in Section 2.1.3. The following Markov discrete state (mode) transition matrix defined in (2.34) is used

$$\Pi = \begin{bmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{bmatrix} \tag{2.84}$$

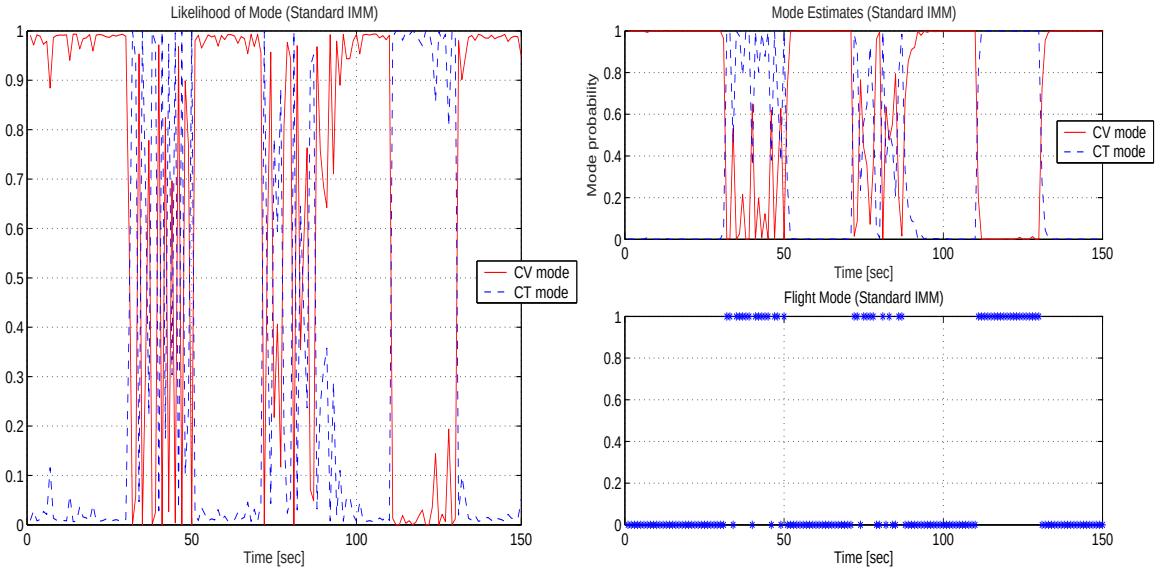
Here, the first column and the first row correspond to the CV mode and the second column



(a) (b)

Figure 2.8: 100 times Monte Carlo simulation results using IMM and RMIMM for the aircraft tracking example. (a) Likelihood of each mode. (b) Mode probability of each mode.

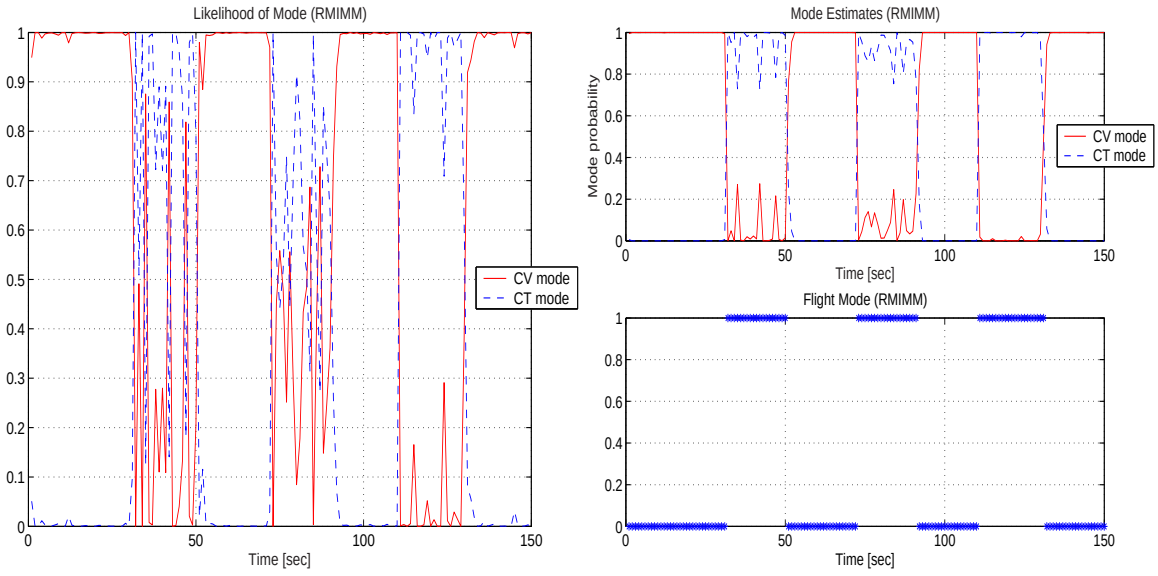
and the second row correspond to the CT mode. For example, π_{12} represents the mode transition probability from the CT mode to the CV mode. The discrete transition matrix is a system parameter which represents the discrete dynamics of the system. However, we could use this as a design parameter for hybrid estimation. The large diagonal values in Π produce a large peak estimation error at the start of a maneuver but a small estimation error when an aircraft stays in a certain mode. This can be expected from (2.78). Large diagonal elements mean that the probabilities that an aircraft stays in a certain mode are high and thus mode transition probabilities are low. Thus, hybrid estimation algorithms using this discrete transition matrix tend to predict the aircraft will stay in the current mode with a high probability. The mode transition probability matrix in (2.84) has been chosen after many simulations. We design a test flight trajectory with constant aircraft



(a) (b)

Figure 2.9: A simulation result using IMM for the aircraft tracking example. (a) Likelihood of each mode. (b) Mode probability and mode estimate of each mode.

speed $v = 480$ knots (kn), composed of seven segments: straight flight from 0 to 30 seconds, a coordinated turn with $\omega = 3^o/sec$ from 31 to 50 seconds, straight flight from 51 to 70 seconds, a coordinated turn with $\omega = 1.5^o/sec$ from 71 to 90 seconds, straight flight from 91 to 110 seconds, a coordinated turn with $\omega = -4.5^o/sec$ from 111 to 130 seconds, and straight flight from 131 to 150 seconds. 100 trial Monte Carlo simulation results in Figure 2.8 show that RMIMM gives more distinct likelihoods of modes than those of the standard



(a) (b)

Figure 2.10: A simulation result using RMIMM for the same aircraft tracking example. (a) Likelihood of each mode. (b) Mode probability and mode estimate of each mode.

IMM algorithm. In addition, a single simulation result in Figure 2.9 and Figure 2.10 shows how RMIMM reduces the number of false mode estimates. The RMS estimation errors of position and velocity using RMIMM are $15m$ and $2.1m/sec$. The RMS estimation errors of position and velocity using IMM are $18m$ and $2.3m/sec$. The RMS estimation errors of RMIMM are slightly better than those of IMM, yet both algorithms give smaller RMS errors than those of the raw measurements. Thus, the main advantage of RMIMM – that it gives better mode estimates than those of IMM – is demonstrated.

Part II

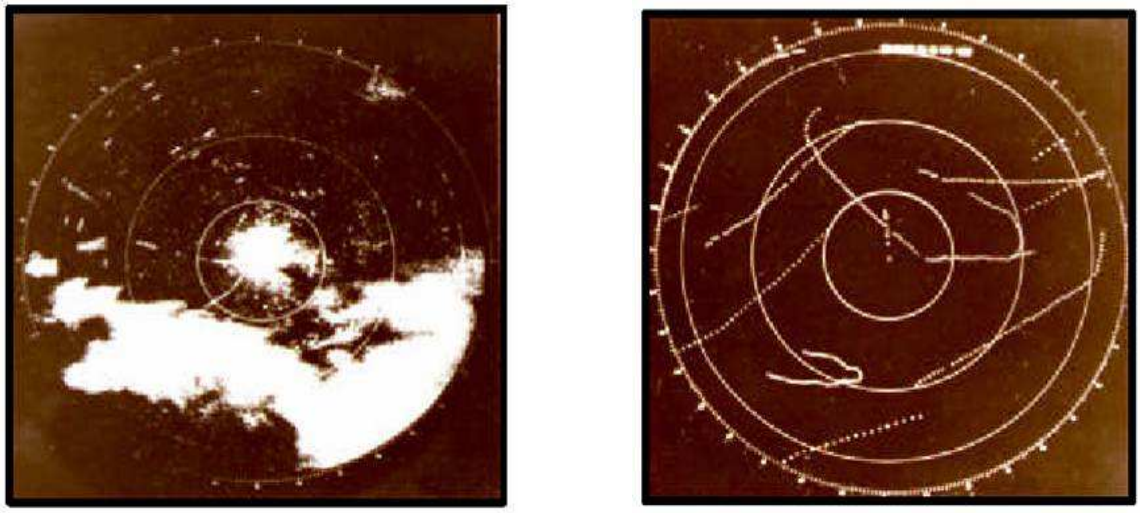
Air Traffic Surveillance and Control

Chapter 3

Multiple-Target Tracking and Identity Management

In the previous chapter, we studied hybrid estimation algorithms for discrete-time stochastic linear hybrid systems and have developed a new hybrid estimation algorithm, called the Residual-Mean Interacting Multiple Model (RMIMM) algorithm. Now, we would like to solve problems of tracking multiple aircraft and managing their identities in a noisy environment for Air Traffic Control (ATC), using the RMIMM algorithm as a state estimator. Figure 3.1-(a) shows such noisy measurements, called *clutter*, from a radar surveillance system for ATC. Clutter is defined as measurements originated from non-targets, such as nearby objects, weather, and electromagnetic interference, that are generally random in number, location and intensity. From the noisy measurements shown in Figure 3.1-(a), we cannot tell how many aircraft are in the surveillance region and what the trajectories of aircraft look like. Thus, the goal of multiple-aircraft tracking and identity management in a noisy environment is to keep track of the trajectories of aircraft and their identities simultaneously from the noisy measurements. Figure 3.1-(b) shows the estimated trajectories of aircraft from the raw radar measurements.

In order to develop an algorithm for conflict detection and resolution for ATC, we need a state estimation algorithm which can track the trajectories of all the aircraft in the surveillance region of the sensors used (radars in the case of the current ATC) as shown in Figure 1.1. This is because conflict resolution is based on conflicts *detected* by a conflict



(a) (b)

Figure 3.1: (a) Clutter in raw radar measurements. (b) Estimated trajectories of aircraft from raw radar measurements [2].

detection algorithm, and conflict detection is in turn based on estimates of an aircraft's current state such as position and velocity. Thus, for conflict detection and resolution, it is crucial to develop an accurate state estimation algorithm which can track multiple aircraft from potentially noisy measurements. In order to resolve conflicts, the current ATC requires the individual identities of all aircraft. This is to enable a ground controller to issue instructions for conflict resolution. For aircraft identification, most aircraft are equipped with transponders which automatically transmit their own identities. However, some general aviation aircraft are not equipped with transponders and it is reported that a transponder itself may frequently malfunction [21]. When a controller detects a conflict but does not know the identity of the aircraft involved in the conflict, he/she tries to identify the aircraft through voice communications. This adds more load on the already congested voice communication channels. The current aircraft identification procedures not only take time but also increase the ground controller's work load and distract his/her attention from other aircraft in his/her region of responsibility. Therefore, we consider identity management problems along with multiple-target (aircraft in this dissertation) tracking problems.

The multiple-target tracking problem deals with correctly tracking several targets given noisy sensor measurements at every instant, while the identity management problem tries to associate target identities with the state estimates available at every instant in a sensor

network. Although closely related, the two problems have so far only been studied independently. We foresee the application of such algorithms to tracking in sensor networks [24], or to Air Traffic Control [1]. The chief emphasis of this work is in the development of an advisory tool for Air Traffic Controllers, although with fairly simple modifications, it could be applied to networks of sensors.

Most multiple-target data association and tracking algorithms proposed until now do not attempt to use local attribute information about targets to improve their performance. While they emphasize the need to track several targets simultaneously, they do not address the necessity to distinguish between the different targets, and indeed, often lead to target-swapping while tracking. In practice, given a network of sensors (or a radar system), in addition to the continuous state measurements, we frequently also receive local sensor information about identities, which can be exploited to reduce target-swapping during tracking. In the case of radar systems, these may be derived from either the physical attributes of an aircraft or from establishment of communication with one of the aircraft.

In reality, the multiple-target tracking and identity management (MTIM) problem is complicated by several problems in the quality of available information about the targets. Firstly, the surveillance system makes measurement errors assumed to be Gaussian and may miss measurements entirely. In certain environments, the surveillance system may also measure extraneous signals, known as clutter. The behavior of the targets also presents complexity to the problem: many targets may be interacting in a small spatial region, and these interactions increase the entropy (a measure of uncertainty) of the system. These issues motivate the extension of the MTIM algorithm to cluttered environments and solving the problems that arise in these larger, more complex systems.

For multiple-target tracking in clutter, we have to decide which measurement is associated to which target and which measurements are clutter. This problem has been addressed by several data association algorithms, which associate measurement data with targets [17]. One such algorithm is the Joint Probabilistic Data Association (JPDA) algorithm in which a target kinematic information (position and velocity) is used for associating measurements with targets. However, JPDA is computationally expensive, and might not work for tracking many aircraft in clutter [73]. For this purpose, many modified versions of JPDA have been proposed [74, 75, 73]. The approximate JPDA proposed in [73] is useful for tracking many aircraft, but unfortunately does not give a stochastic association matrix whose elements

represent measurement-target association probabilities, thus losing some of the physical constraints imposed by the system. Assignment algorithms have also been used to overcome the computational complexity of data association for multiple-target tracking problems [76, 77]. These assignment algorithms minimize the sum of all probabilistic distances between measurements and expected target positions. This means that the assignment algorithms select the measurement that is closest to the predicted measurement without considering measurement-target correlation like nearest neighbor data association. Therefore, they lose the advantage of the JPDA algorithm which considers all possible correlation between measurements and targets.

In this chapter, for data association, we develop a modified approximate JPDA (MAJPDA) which is more computationally efficient than JPDA and provides a stochastic association matrix as JPDA. In order to overcome the defects of the previously proposed assignment algorithms, we use data association probabilities computed by the MAJPDA algorithm as weighting coefficients for the assignment problem and use the *extended Munkres algorithm* [78, 79] in order to maximize the overall data association probability. In this way, we can consider measurement-data correlation as JPDA does. The Munkres algorithm is an assignment algorithm which computes an optimal assignment of N workers to N jobs, assuming that numerical ratings are given for each worker's performance on each job [78]; an extended Munkres algorithm works for the case in which the numbers of men and jobs are different. We also develop a method of generating local information on the system when such information is not explicitly available. For this, we use a Multiple Hypothesis Testing (MHT) algorithm [20, 80] to correct the identities of the targets when targets are close and thus their identities are mixed. By logically combining these algorithms, we develop the MTIM algorithm which can keep track of multiple-aircraft in clutter and their identities.

3.1 Multiple-Target Tracking and Identity Management algorithm

In this section, we consider the problem of associating time series of measurements to the tracks and identities of one or more targets in the presence of clutter. The MTIM algorithm approaches this problem using the structure shown in Figure 3.2 at each time step. This section details the algorithms used to implement each of the three stages. Because these

algorithms are independent of each other, their inputs and outputs do not interface exactly. Other stages exist which process the outputs of any of these stages into a form that can be input into another primary stages.

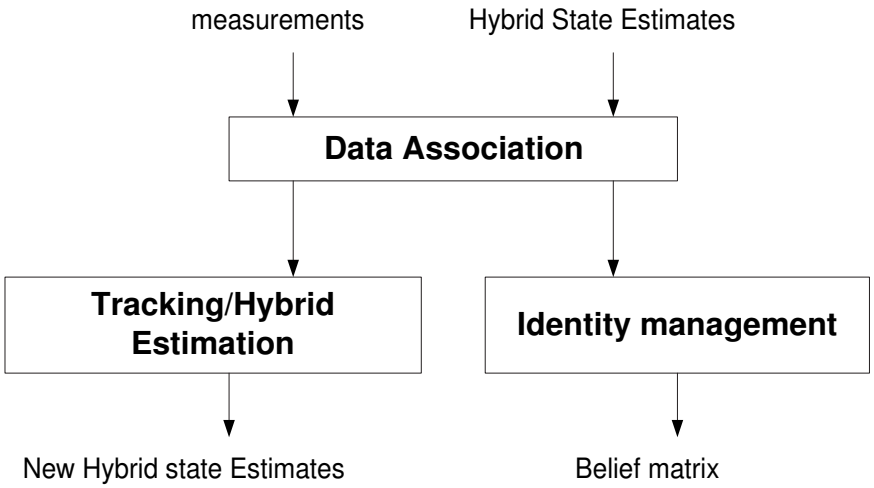


Figure 3.2: General Multiple-Target Tracking and Identity Management (MTIM) structure.

The first stage is *Data Association*, which consists of matching incoming measurements to the targets. Given state estimates of T targets from the previous time step and M measurements from the current time step, the Data Association block is used to generate an $M \times T$ matrix of association probabilities. Entries in this matrix represent the probability of a given measurement having originated from a given target.

The *Tracking/Hybrid State Estimation* block of MTIM performs the tracking of T targets in parallel. The tracking algorithm for each target takes as input the hybrid state estimate from the previous time $k-1$ and a single measurement from the current time k. The measurement input comes from the Data Association block. The hybrid state estimate comprises position and velocity estimates, their covariances, and a flight mode estimate. The output of the Tracking/Hybrid State Estimation block is the hybrid state estimate at time k.

The *Identity Management* block takes as input the belief matrix from time $k-1$ and the $T \times T$ association probability matrix. Entries in the belief matrix represent the probability that a given target (column) has a corresponding given identity (row). For example, the

belief matrix is initialized to the identity matrix, since it is assumed that all targets initially uniquely identified. The Identity Management stage outputs the belief matrix at time k. The following sections discuss each structural block and the algorithms used to implement the stages.

3.1.1 Data association algorithms

This section gives an overview of data association algorithms [17]. Consider the problems of associating measurements and tracking one or more targets in the presence of random noise (clutter). Denote $z(k)$ as a measurement at time k and $\hat{z}(k+1|k)$ as a predicted measurement at time $k+1$ using information up to time k. Assume $p(z(k+1)|Z^k) = \mathcal{N}(z(k+1); \hat{z}(k+1|k), S(k+1))$, where $\mathcal{N}(x; a, b)$ denotes a Gaussian random variable x with mean a and covariance b. The *validation gate* is defined as:

$$\begin{aligned} \tilde{V}_{k+1}(\gamma) &:= \{z(k+1)|[z(k+1) - \hat{z}(k+1|k)]^T S(k+1)^{-1}[z(k+1) - \hat{z}(k+1|k)] \leq \gamma^2\} \\ &= \{z(k+1)|r^T(k+1)S(k+1)^{-1}r(k+1) \leq \gamma^2\} \end{aligned} \tag{3.1}$$

where $r(k+1) = z(k+1) - \hat{z}(k+1|k)$ is the residual, $S(k+1)$ is its covariance, and γ is a design parameter which determines the size of the validation gate. Measurements that lie inside the gate are considered valid. The set of validated measurements at time k is denoted as:

$$Z(k) := \{z_i(k)\}_{i=1}^{m_k} \tag{3.2}$$

where m_k is a random variable. The measurement sequence up to time k is defined as:

$$Z^k := \{Z(j)\}_{j=1}^{k} \tag{3.3}$$

Thus, the problem of associating each validated measurement with an appropriate target or identifying it as clutter and discarding it is known as data association, measurement association, or data correlation. The set of validated measurements consists of both the potentially correct and incorrect measurements.

Nearest-Neighbor Standard Filter (NNSF)

Consider tracking a target in clutter. The Nearest-Neighbor Standard Filter (NNSF) selects the validated measurement closest to the predicted measurement and uses it for state

estimation, where the distance measure is given by

$$d(z(k))^2 = r^T(k)S(k)^{-1}r(k) \tag{3.4}$$

The distance defined in (3.4) is a weighted distance with the inverse of the measurement prediction uncertainty represented by $S(k)$ (residual covariance) as a weight. For example, Figure 3.3 shows that measurements z_1 and z_3 are validated for Target 1 whose predicted measurement is denoted as $\hat{z}_1$, measurements z_2, z_3 and z_5 for Target 2, and z_4 is not validated for either target. The NNSF selects measurement z_3 as the correct measurement for both targets because z_3 is the closest measurement to both the predicted measurements for Target 1 and 2. However, this is physically impossible because only one measurement can originate from one target. This incorrect measurement-target association comes from the fact that the residual covariance matrix S computed by a state estimator (Kalman filter in this dissertation) could not account for the possibility of incorrect measurement association. Thus, the performance of the NNSF might be poor in such cases.

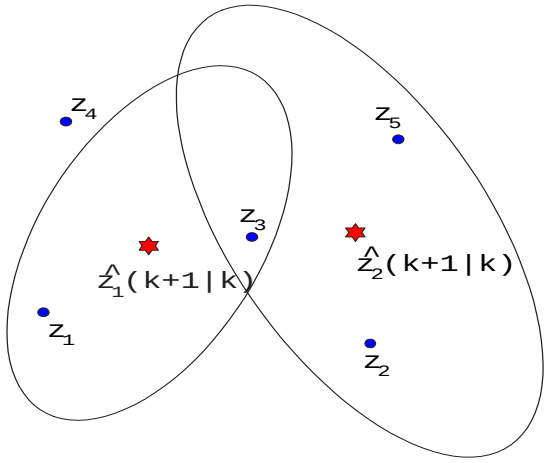


Figure 3.3: Validation Gate.

Probabilistic Data Association (PDA)

Consider problems of tracking a single target in clutter. Among the possibly many validated measurements, only one is target-originated, and the other measurements are assumed to

be clutter and modelled as independent identically distributed (IID) random variables with a uniform spatial distribution.

The basic assumption of Probabilistic Data Association (PDA) is:

$$p(x(k)|Z^{k-1}) = \mathcal{N}(x(k); \hat{x}(k|k-1), P(k|k-1)) \tag{3.5}$$

i.e., the state is assumed to be Gaussian according to the latest estimate $\hat{x}(k|k-1)$ and covariance $P(k|k-1)$. Define the events

$$\begin{array}{rcl} \theta_i(k) & := & \{z_i(k) \text{ is the true measurement}\} \\ \theta_0(k) & := & \{\text{none of the measurements at time } k \text{ is true}\} \end{array} \tag{3.6}$$

with probability $\beta_i(k) := p(\theta_i(k)|Z^k)$, $i \in \{0, 1, \cdots, m_k\}$, called the *association probability*, and $\sum_{i=0}^{m_k} \beta_i(k) = 1$. Then, the conditional mean state estimate at time k is:

$$\hat{x}(k|k) = E[x(k)|Z^k] = \textstyle\sum_{i=0}^{m_k} E[x(k)|\theta_i(k), Z^k] p(\theta_i(k)|Z^k) = \sum_{i=0}^{m_k} \hat{x}_i(k|k)\beta_i(k) \tag{3.7}$$

The association probability $\beta_i(k)$ is:

$$\begin{array}{rcll} \beta_i(k) & = & p(\theta_i(k)|Z(k), m_k, Z^{k-1}) & \\ & = & \frac{1}{c} p(Z(k)|\theta_i(k), m_k, Z^{k-1}) p(\theta_i(k)|m_k, Z^{k-1}) & \\ & = & \left\{ \begin{array}{ll} V_k^{-m_k+1} P_G^{-1} \mathcal{N}(r_i(k); 0, S_i(k)) p(\theta_i(k)|m_k, Z^{k-1}), & \text{for } i = 1, \cdots, m_k \\ V_k^{-m_k} p(\theta_i(k)|m_k, Z^{k-1}), & \text{for } i = 0 \end{array} \right. & \end{array} \tag{3.8}$$

where V_k is the volume of the validation gate and $P_G := \{z(k) \in \tilde{V}_k(\gamma)\}$ is the *gate probability* that the correct measurement lies in the validation gate.

Since PDA is based on the assumption that there is one target and only one measurement is correct and all others are clutter, its performance could be poor when there are multiple targets. In the next section, we consider multiple target tracking problems.

Joint Probabilistic Data Association (JPDA)

The Joint Probabilistic Data Association (JPDA) algorithm is an extension of PDA to the situation in which there is a known number of targets in clutter. If there are several

targets in the same neighborhood, measurements from one target can persistently fall in the validation gate of a neighboring target. Since PDA could not account for this situation, its performance could degrade significantly in such a situation.

The key to the JPDA algorithm is the evaluation of the conditional probabilities of the following joint events:

$$\Theta = \bigcap_{j=1}^{m_k} \theta_{jt}, \quad j = 1, \cdots, m_k; \;\; t = 0, 1, \cdots, T \tag{3.9}$$

where $\theta_{jt} := \{\text{measurement } j \text{ originated from target t}\}$. For example, for the case in Figure 3.3, one possible joint event is an event that z_2 originates from Target 1, z_1 originates from Target 2, and z_3, z_4, and z_5 are clutter. There are combinatorial many possible joint events (in the case in Figure 3.3, there are $\frac{5!}{2!} = 60$ joint events) and thus computing the probabilities of all possible joint events would be computationally intractable as the number of targets and measurements increase. We can reduce the number of joint events using validation gates. For example, the above joint event is not feasible because z_1 could not originate from Target 2 since it does not lie in the validation gate of Target 2. A joint event association matrix can be represented by the matrix

$$\hat{\Omega} = [\hat{\omega}_{jt}(\Theta)], \;\; \text{where } \hat{\omega}_{jt}(\Theta) \begin{cases} 1 & \text{if } \theta_{jt} \subset \Theta \\ 0 & \text{otherwise} \end{cases} \tag{3.10}$$

where $\hat{\Omega}$ is a $(m_k \times (T+1))$ matrix. $\hat{\omega}_{jt} = 1$ represents an event that measurement j originates from target t. The first column in $\hat{\Omega}$ corresponds to the event $t = 0$ which represents the *no target* event. That is, $\hat{\omega}_{j0}$ represents an event that measurement j is clutter. A feasible joint event is one which satisfies:

1. A measurement can have only one target $(j = 1, \cdots, m_k)$.
   $\tau_j(\Theta) := \sum_{t=0}^{T} \hat{\omega}_{jt}(\theta) = 1 \leftarrow$ measurement association indicator

2. No more than one measurement can originate from a target $(t = 1, \cdots, T)$.
   $\delta_t(\Theta) := \sum_{j=1}^{m_k} \hat{\omega}_{jt}(\Theta) \leq 1 \leftarrow$ target detection indicator

For example, a feasible joint event for the case in Figure 3.3 is

$$\hat{\Omega} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{3.11}$$

which represents an event that z_1 originates from Target 1, z_2 originates from Target 2, and z_3, z_4 and z_5 are clutter. Measurements not associated with any target are assumed to be uniformly distributed in space. We define the following notation:

$$\begin{array}{rcl} \phi(\Theta) & := & \sum_{j=1}^{m_k}[1-\tau_j(\Theta)], \ \leftarrow \text{ number of unassociated measurements in event } \Theta \\ P_D & : & \text{target detection probability} \end{array} \tag{3.12}$$

Then, the association probability is

$$\begin{array}{rcl} \beta_{jt} & = & \sum_{\Theta} p(\Theta|Z^k)\hat{\omega}_{jt}(\Theta) \\ p(\Theta|Z^k) & = & \frac{1}{c}\frac{\phi!}{V^\phi}\prod_{j=1}^{m_k}[\mathcal{N}_{t_j}(z_j(k))]^{\tau_j}\prod_{t=1}^{T}(P_D^t)^{\delta_t}(1-P_D^t)^{1-\delta_t} \end{array} \tag{3.13}$$

where $\mathcal{N}_{t_j}(z_j(k)) := \mathcal{N}(z_j(k); \hat{z}^{t_j}(k|k-1), S^{t_j}(k))$ and $\hat{z}^{t_j}(k|k-1)$ denotes the predicted measurement for target t_j with an associated residual covariance S^{t_j}. A more detailed explanation of PDA and JPDA is provided in [81].

3.1.2 Identity management algorithm

Identity management denotes the assignment of labels to targets and the evolution of the labels over time. The interaction of multiple targets could make the problem complex. Indeed, a common approach to this problem is to maintain probabilities of all possible label-target assignments at each time step in the system. The computational complexity of such an algorithm grows exponentially in time.

A scalable distributed algorithm for computing and maintaining multiple-target identity information has been developed using Identity-Mass Flow which overcomes the exponential computational complexity problem [24]. This algorithm maintains identity information

over time given information about the interaction between N targets. This information is stored in an $N \times N$ *identity belief matrix* $B(k)(= [B_{ij}(k)])$, where k is the current time step. The matrix is doubly stochastic; that is, $\sum_{i=1}^{N} B_{ij}(k) = 1$, for $j \in \{1, \ldots, N\}$ and $\sum_{j=1}^{N} B_{ij}(k) = 1$, for $i \in \{1, \ldots, N\}$. The evolution of this belief matrix is governed by a *mixing matrix* $M(k)(= [M_{ij}(k)])$, which stores interaction information for a single time step. $M_{ij}(k)$ represents the probability that target i at time $k-1$ has become target j at time k. The belief matrix is updated according to the equation [24]

$$B(k) = B(k-1)M(k). \tag{3.14}$$

In certain applications, identity information about a target could be obtained from sensors which can measure its physical attributes, such as the shape and noise characteristics. For example, in applications of ad-hoc sensor networks, vision sensors (acoustic sensors) can measure the shape (noise characteristics) of a target and from this measurement, its identity can be inferred [72]. This information about the identity of a target obtained from sensors, is called *local information* in this dissertation. Identity management can also utilize this target attribute information, if available from local sensors, to maintain the identity of a target correctly.

The identity management algorithm in [24] assumes that target position estimates are given, but in practice, it is difficult to obtain the target position state estimates accurately enough for the identity management algorithm to work in the multiple-target tracking environment. For example, in Air Traffic Control (ATC) applications, tracking multiple maneuvering aircraft in clutter is not trivial, as shown in Section 3.1.1. Data association for target tracking and identity management are closely related since both algorithms compute and update the relation between tracks and targets.

3.1.3 Incorporation of Local Information

In this section, we consider a problem of how local information, if available, could be used to reduce the uncertainty of the belief matrix. For this, we use the entropy of a probability vector $f \in [0,1]^n$, defined as $H(f) := \sum_{i=1}^{n} -f_i \ln f_i$, as an uncertainty measure of the probability vector f. Using this, we define the average entropy of the belief matrix $B(k)$ of the N targets as $\bar{H}(B(k)) := \frac{1}{N} \sum_{j=1}^{N} H(b_j(k))$, where $B(k) = \begin{bmatrix} b_1(k) & \cdots & b_N(k) \end{bmatrix}$. The

average entropy of the belief matrix is used as a measure of statistical uncertainty of the probability density of the identity of the N targets. In the belief matrix, since the columns represent the probabilities of identity belief for each target, the probability distribution of belief for each target is given by the corresponding column. Using this definition, we can prove Lemma 1 in [24].

Lemma 7. *Let $\bar{H}(B(k))$ be the average entropy over all the columns of the belief matrix $B(k)$. Then, $\bar{H}(B(k)) \geq \bar{H}(B(k-1))$, if $B(k) = M(k)B(k-1)$; that is, mixing does not decrease the average entropy.*

Proof. From the definition of average entropy of the system,

$$\begin{aligned} \bar{H}(B(k)) &= \tfrac{1}{N}\textstyle\sum_{j=1}^{N} H(b_j(k)) \\ &= \tfrac{1}{N}\textstyle\sum_{j=1}^{N} H([M(k)B(k-1)]_j) \\ &= \tfrac{1}{N}\textstyle\sum_{j=1}^{N} H([\sum_{i=1}^{N!} \alpha_i \Phi_i B(k-1)]_j) \\ &= \tfrac{1}{N}\textstyle\sum_{j=1}^{N} H(\sum_{i=1}^{N!} \alpha_i [\Phi_i B(k-1)]_j) \\ &\geq \tfrac{1}{N}\textstyle\sum_{j=1}^{N} \sum_{i=1}^{N!} \alpha_i H([\Phi_i B(k-1)]_j) \end{aligned} \tag{3.15}$$

where Φ_i is a permutation matrix. But premultiplying by a permutation matrix simply permutes the rows, so the set of values in the column does not change.

$$\implies H([\Phi_i B(k-1)]_j) = H(b_j(k-1)) \tag{3.16}$$

Therefore, we get

$$\begin{aligned} \bar{H}(B(k)) &\geq \tfrac{1}{N}\textstyle\sum_{j=1}^{N} \sum_{i=1}^{N!} \alpha_i H(b_j(k-1)) \\ &= \tfrac{1}{N}\textstyle\sum_{j=1}^{N} H(b_j(k-1)), \ \text{since} \textstyle\sum_{i=1}^{N!} \alpha_i = 1 \\ &= \bar{H}(B(k-1)) \end{aligned} \tag{3.17}$$

□

Corollary 2. *Since $\bar{H}(B(k)) = \frac{1}{n}\sum_{j=1}^{N} H(b_j(k))$(sum over columns) $= \frac{1}{n}\sum_{j=1}^{N} H(b_j(k))$ (sum over rows), the same proof of no decrease of entropy holds for mixing of the form $B(k) = B(k-1)M(k)$.*

In the identity management algorithm, we assume that local information arrives in the form of a column vector whose elements represent the probabilities of identity belief of a

target. We replace the column corresponding to the target in the belief matrix with local information, and scale the rest of the belief matrix to make it doubly-stochastic, using Sinkhorn scaling [82]. We now investigate in detail when the local information should be used to reduce uncertainties of the beliefs of targets' identities.

1. **Identity-type local information:** This is local information which gives with certainty the identity of one of the targets. In the implementation, this corresponds to local information in the form of a column unit-vector. We could not find a case in which identity-type local information increases the entropy of the system through many simulations. The entropy of identity-type local information is zero since it is in the form of a unit vector. Thus, from these observations, we believe that it decreases the entropy of the system.

   Conjecture 1. *Identity-type local information always reduces the entropy of the system.*

   Therefore, identity-type local information should be incorporated to reduce the uncertainty of the belief matrix.

2. **General forms of local information:** In general, local information is in the form of a stochastic column vector (elements sum to 1). In this case, clearly, the effect on the entropy depends on the elements of the column and need not necessarily decrease the entropy. Consider, for example, the belief matrix $\begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix}$. The average entropy of this matrix is 0.5004. If local information arrives at column 2 in the form $\begin{bmatrix} 0.3 \\ 0.7 \end{bmatrix}$, then the corresponding doubly stochastic matrix after Sinkhorn scaling is $\begin{bmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{bmatrix}$. The average entropy of the updated matrix is 0.6109, that is, the average entropy increases when we incorporate information of this form. Therefore, local information in the form of a stochastic vector should be incorporated only if it decreases the average entropy of the belief matrix.

The above statements have important implications in the incorporation of local information. Sinkhorn iteration is an approximate algorithm to obtain a doubly-stochastic matrix, and there may be situations in which the incorporation of local information would increase

our uncertainty in belief. However, we are justified in always incorporating identity-type information. Since it is computationally quite simple to compute the average entropy, we only incorporate general local information if the doubly-stochastic matrix after the Sinkhorn scaling has a smaller average entropy than before the incorporation of the local information.

3.2 Multiple-Target Tracking and Identity Management in an environment without clutter

Let us assume, for the moment, that we are operating in an environment without clutter, meaning that each measurement corresponds to a target, and there are exactly as many targets (T) as measurements (T). Then, the target detection probability $P_D = 1$, the number of unassociated measurements $\phi(\Theta) = 0$, and the measurement association indicator $\tau_j(\Theta) = 1$ defined in Section 3.1.1. The association probability in (3.13) becomes

$$\beta_{jt}(k) = \sum_{\Theta}[\prod_{j=1}^{T} \mathcal{N}[z_j(k)]\hat{\omega}_{jt}(\Theta)] \tag{3.18}$$

We apply the Sinkhorn iteration to the association matrix $A(k) := [\beta_{jt}(k)]$ to make it a doubly-stochastic matrix $A'(k)$, and we use $A'(k)$ as the mixing matrix $M(k)$ in the identity management algorithm. Then, the evolution of the belief matrix is governed by (3.14). Thus, the MTIM algorithm for an environment without clutter is as follows:

Algorithm 1. *Algorithm for Multiple-target Tracking and Identity Management in an environment without clutter*
For target t ($t \in \{1, \cdots, T\}$),

- *Step 1: mixing/interaction: initialize $\hat{x}_{0i}(k-1|k-1)$ and $P_{01}(k-1|k-1)$*
- *Step 2: Kalman Filter i (for mode i, $i \in \{1, \cdots, N\}$)*
  1. *State propagation/prediction*

$$\begin{aligned} \hat{x}_i(k|k-1) &= A_i\hat{x}_{0i}(k-1|k-1) \\ P_i(k|k-1) &= A_iP_{0i}(k-1|k-1)A_i^T + Q_i \\ S_i(k) &= C_iP_i(k|k-1)C_i^T + R_i \end{aligned} \tag{3.19}$$

2. *Measurement Validation*

$$r_{ij}(k)^T S_i(k)^{-1} r_{ij}(k) < \gamma^2 \tag{3.20}$$

where $r_{ij}(k) = z_j(k) - C_i\hat{x}_i(k|k-1)$ *(*$j \in \{1, \cdots, m_k^i\}$*) and* m_k^i *is the number of validated measurements for target* i *at time* k*.*

3. *Measurement Update*

   (a) *Compute an association matrix using (3.18) and a mixing matrix by making the association matrix a doubly stochastic matrix using Sinkhorn iteration.*

   (b) *Update the belief matrix using (3.14).*

   (c) *If local information arrives and it decreases the entropy of the belief matrix, then update the column corresponding to the local information, and scale the rest of the matrix (using Sinkhorn iteration) to make it doubly-stochastic.*

   (d) *Update the continuous state estimate and its covariance*

$$\begin{aligned} \hat{x}_i(k|k) &= \hat{x}_i(k|k-1) + K_i(k)\textstyle\sum_{l=1}^{m_k^i} \beta_{il}(k) r_{il}(k) \\ P_i(k|k) &= [I - K_i(k)C_i]P_i(k-1|k-1) + K_i(k)[\textstyle\sum_{l=1}^{m_k^i} \beta_{il}(k) r_{il}(k) r_{il}(k)^T \\ &\quad -(\textstyle\sum_{l=1}^{m_k^i} \beta_{il}(k) r_{il}(k)) \times (\textstyle\sum_{l=1}^{m_k^i} \beta_{il}(k) r_{il}(k))^T] K_i(k)^T \end{aligned} \tag{3.21}$$

where a Kalman filter gain $K_i(k) = P_i(k|k-1)C_i^T[C_iP(k|k-1)C_i^T + R_i]^{-1}$*.*

- *Step 3: Compute mode likelihood functions*

$$\Lambda_i(k) = \mathcal{N}(r_i(k); 0, S_i(k)) \tag{3.22}$$

  where $r_i(k) := \sum_{l=1}^{m_k^i} \beta_{il}(k) r_{il}(k)$*.*

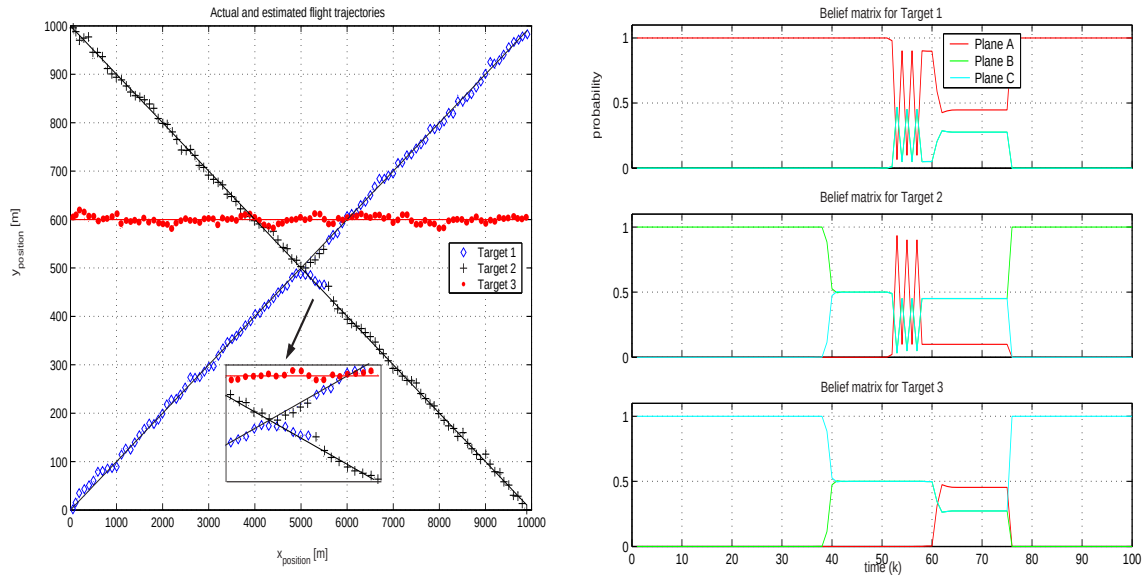
- *Step 4: Compute mode probabilities:* $\mu_i(k)$*.*

- *Step 5: Compute outputs:* $\hat{x}(k|k), P(k|k), \hat{m}(k)$ *using (2.79) and* $B(k)$*.* ∎

3.2.1 Simulation results

We consider a three-aircraft scenario, in which given the (noisy) measurements of the position coordinates of three different aircraft (in random order) at every instant of time, we

track the aircraft not only in their state estimates, but also with respect to their identities. This is representative of a scenario in which we might receive local information about one of the aircraft, based on, perhaps, its physical attributes such as the shape and noise characteristics of the aircraft. The simulation results are shown in Figures 3.4. From the figure, it is clear that the proposed algorithm is effective in maintaining both track and identity, and is also capable of incorporating any local information that may become available in a manner consistent with the system behavior.



(a) (b)

Figure 3.4: (a) Trajectory comparisons, (b) Belief matrix evolution.

3.3 Multiple-Target Tracking and Identity Management in clutter

In this section, we extend the MTIM algorithm developed in the previous section to its application in a cluttered environment. We first present the structure of the MTIM algorithm. Next, each step is detailed in its specific use in the MTIM algorithm for a cluttered environment. Figure 3.5 shows the evolution of hybrid state estimates and identity at a single time k. The first stage is *State Prediction*, which generates an estimate of the state at time k based only on the outputs of MTIM at time $k-1$. This stage is performed in parallel

for each of the N targets. The next stage is *Measurement Validation/Association*. There are at least N (the number of targets) measurements input to the MTIM algorithm at time k. The measurements are validated in validation gates defined by the predicted states of the targets. The JPDA algorithm can be used to choose N measurements and generate an $N \times N$ mixing matrix $M(k)$. However, in order to deal with many targets in clutter with good accuracy, we develop a modified approximate JPDA (MAJPDA) to generate the mixing matrix. The approximate JPDA on which MAJPDA is based outperforms JPDA for large numbers of measurements, such as in a cluttered environment [73]. The extra modification is to make the $N \times N$ subset of the association probability matrix $\beta(k)$ doubly stochastic, resulting in a mixing matrix. The element at the i^{th} row and the j^{th} column of the mixing matrix $M(k)$, output by MAJPDA, contains the probability that measurement i, of the chosen N measurements, is associated with target j.

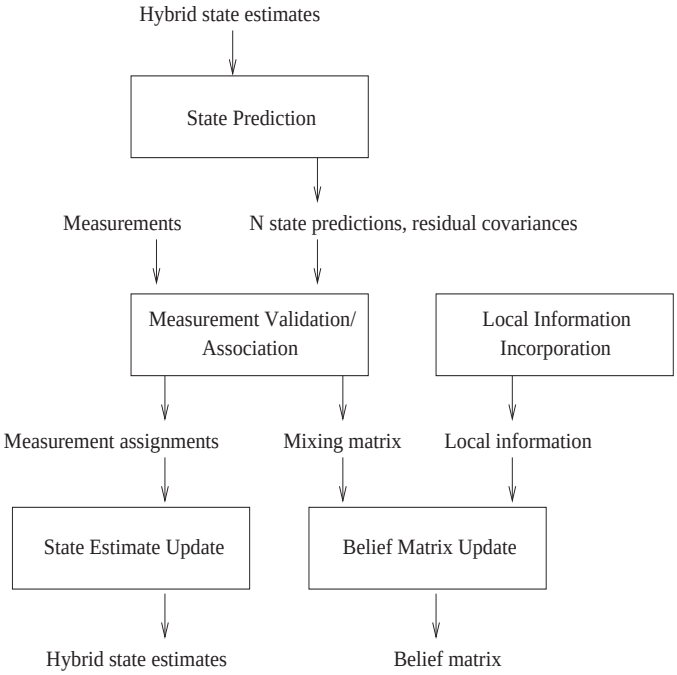


Figure 3.5: MTIM Block Diagram.

The outputs from this stage are the mixing matrix and the one-to-one assignment between the N chosen measurements and the N targets. The assignment between the measurements and targets are then input to the *State Estimate Update* block. The RMIMM algorithm

is used to update continuous state, covariance, and mode estimates for each of the N targets. These estimates are part of the MTIM algorithm output for time k and are input to the State Prediction block at time $k+1$. The other output from the Measurement Validation/Association block, the mixing matrix, is input to the *Belief Matrix Update* block. This block updates the belief matrix from time $k-1$ to reflect the interaction described by the mixing matrix. The *Local Information Incorporation* block generates target identity information and incorporates this information into the belief matrix only if it reduces uncertainty in the belief matrix [37]. The belief matrix and the hybrid state estimate are the outputs generated by the MTIM algorithm at each time step. The following sections detail each block and present the logic behind various algorithmic choices.

3.3.1 State Prediction

State prediction is carried out for each of the N targets. The details that follow refer to the procedure used for a single target. The stage takes as input the continuous state estimates $\hat{x}_i(k-1|k-1)$, covariances $P_i(k-1|k-1)$, and mode probabilities $\mu_i(k-1)$, where i refers to the mode of the target. The output of the block is a prediction of the state and its covariance at time k without information from time k. First, the mixing stage of RMIMM is used to combine the state estimates from the different modes, resulting in new initial states $\hat{x}_{0i}(k-1|k-1)$ and covariances $P_{0i}(k-1|k-1)$. These are input to a set of Kalman filters, one for each mode, without measurement inputs. The outputs of the Kalman filters are state predictions

$$\hat{x}_i(k|k-1) = A_i\hat{x}_{0i}(k-1|k-1), \tag{3.23}$$

covariances

$$P_i(k|k-1) = A_iP_{0i}(k-1|k-1)A_i^T + Q_i, \tag{3.24}$$

and residual covariances

$$S_i(k) = C_iP_i(k|k-1)C_i^T + R_i, \tag{3.25}$$

The mode estimate $\hat{m}(k-1)$ from the previous time step is used to obtain a single continuous state prediction $\hat{x}(k|k-1)$ and a single residual covariance $S(k)$. Because the predicted state is assumed to have a Gaussian distribution, the state prediction is the mean (center) of the validation gate of the target, while the residual covariance is the covariance of the

validation gate. $S(k)$ is also supposed to be used to determine the size of the validation gate, according to (3.1). However, when a target changes modes (starts a maneuver), the Kalman filter overestimates its confidence in its state estimate, which results in a smaller $S(k)$ than is appropriate. Often, the measurement of the maneuvering target does not fall inside its validation gate; as a result, the size of the validation gate must be increased. This increase is obtained by increasing the state covariance S with an additional term that compensates for the additional uncertainty about the maneuvering target. This additional term is related to the state velocity estimate $\hat{v}$ according to the expression

$$S_{extra} = \tau^2 \hat{v}\hat{v}^T + \nu^2 \hat{v}_\perp \hat{v_\perp}^T, \tag{3.26}$$

where $\hat{v}_\perp$ is obtained by rotating $\hat{v}$ by 90^o in the counterclockwise direction. The effective residual covariance S' is then equal to

$$S' = S + S_{extra}. \tag{3.27}$$

Since S_{extra} is positive definite, the region covered by the validation gate created from S' is larger than that created by S, as shown in Figure 3.6. In this figure, the smaller ellipse is the validation gate as determined by S, while the larger ellipse is that determined by S'. The extended validation gate is longer in the cross-track direction to account for the likelihood of targets maneuvering to either side of their expected track. The constants τ and ν are chosen empirically to ensure that maneuvers are extremely unlikely to lead to measurements outside validation gates; the cross-track term ν is chosen to be larger than the along-track term τ. The additional term S_{extra} is related to velocity because errors in track due to a maneuver will be directly related to the velocity of the target. Thus, the outputs from the first block are state prediction $\hat{x}_t(k|k-1)$, residual covariance $S_t(k)$, and effective residual covariance $S'_t(k)$ for target t. There are N sets of outputs, one set for each target. The effective residual covariance S' is used for measurement validation only.

3.3.2 Measurement Validation/Association

We first define the residual $r_j^t(k) = z_j(k) - C^t\hat{x}^t(k|k-1)$ for target t and measurement j. The measurement at position $z_j(k)$ falls inside the validation gate if (3.1) is satisfied, with $S(k)$ replaced with the effective residual covariance $S'(k)$. Given the set of validated

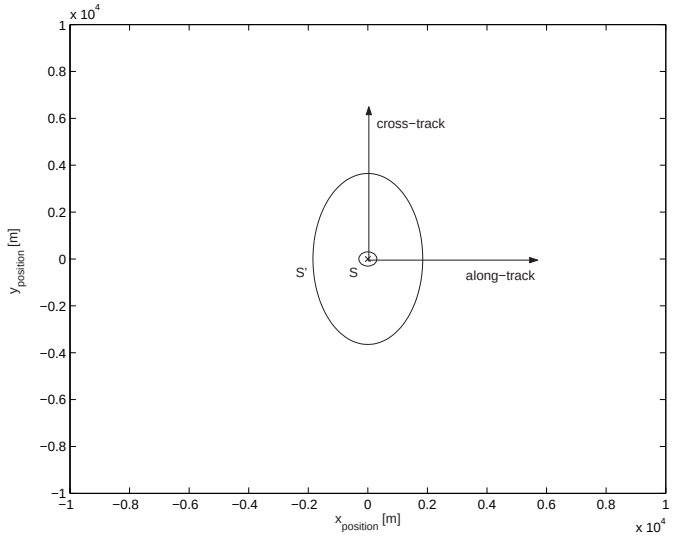


Figure 3.6: Validation gates determined by the original residual covariance S and the effective residual covariance S' which accounts for the maneuvering uncertainty of a target.

measurements, MAJPDA is used to obtain the association probability matrix $\beta(k)$ and the mixing matrix $M(k)$.

The approximate JPDA (AJPDA) algorithm is a computationally abbreviated version of JPDA [73]. Denote the Gaussian probability density function of the residual $\mathcal{N}_{t_j}(z_j(k))$ as $G_{jt}(k)$. Thus, $G_{jt}(k)$ is proportional to the Gaussian likelihood function that represents the closeness between target t and measurement j. We let

$$P_{st}(k) := \sum_{j}(G_{jt}(k)) \tag{3.28}$$

and

$$P_{rj}(k) := \sum_{t}(G_{jt}(k)) \tag{3.29}$$

Then the association probability is defined as [73]

$$\beta_{jt}(k) = \frac{G_{jt}(k)}{P_{st}(k) + P_{rj}(k) - G_{jt}(k) + B}. \tag{3.30}$$

Thus, (3.30) puts more weight on the target which does not fall into the validation gates of any other targets as JPDA does. B is a bias term set to 0 in most cases, including

for this paper. However, the data association matrix computed by AJPDA might not be stochastic, which JPDA produces. Thus, the accuracy of AJPDA might not be good enough for certain situations. To correct this and improve the performance of data association, we propose a modified approximate JPDA (MAJPDA) which uses the Sinkhorn algorithm [82, 37] to make the data association matrix $\beta(k)$ doubly stochastic. Thus, MAJPDA keeps essential characteristics of JPDA and thus, outperforms AJPDA with far less computational complexity than JPDA for tracking many targets in clutter.

Because there are more measurements than targets in a cluttered environment, there is a need to choose a subset of the full data association matrix as the mixing matrix, which should be a square matrix [24]. The MAJPDA algorithm entails both the determination of the association probability matrix and the doubly stochastic, square mixing matrix. In the no-clutter case, the mixing matrix is nothing more than the doubly stochastic form of the marginal association probability matrix β. However, for a cluttered environment, if there are K measurements, then β has K rows and $N \leq K$ columns. The mixing matrix $M(k)$ must still have N rows and columns. To choose N of the K rows, we use the extended Munkres algorithm [79]. The Munkres algorithm is an assignment algorithm which chooses the set of N numbers with maximum sum from all sets of N numbers taken from an N by N matrix such that the numbers cover every row and every column [78]. Bourgeois and Lassalle extend this algorithm to rectangular matrices [79]. Thus, for a $K \times N$ matrix, with $K \geq N$, the extended Munkres algorithm picks N from the matrix with maximum sum so that these numbers cover N distinct rows and all of the N columns. This extension lends itself to processing the association probability matrix output by MAJPDA. The N numbers chosen from $\beta(k)$ by the extended Munkres algorithm constitute $z_j(k)$, which are the N measurements assigned to the N targets to maximize the sum of association probabilities. The assignment of measurements to targets is a one-to-one correspondence between measurement j and target t; that is, j is a function of t and vice-versa. The N rows of $\beta(k)$ representing these measurements form an $N \times N$ matrix. The doubly stochastic form of this matrix is the mixing matrix $M(k)$. The mixing matrix and measurement assignments are then passed to the Belief Matrix Update and State Estimate Update blocks, respectively.

3.3.3 State Estimate Update

Given a measurement for target t, $z_j(k)$, RMIMM propagates the continuous state $\hat{x}^t(k-1|k-1)$, its covariance $P^t(k-1|k-1)$, and mode probabilities $\mu^t(k-1)$ to time k. The

outputs include $\hat{x}^t(k|k)$, $P^t(k|k)$, and $\hat{m}^t(k)$. These outputs are then used as input to the MTIM algorithm at time $k+1$.

3.3.4 Belief Matrix Update

The mixing matrix $M(k)$ from the Measurement Validation/Association block is input to the Belief Matrix Update block. The evolution of belief matrix $B(k)$ is governed by (3.14). The entry $B_{ij}(k)$ represents the probability that target j can be identified as label i. The belief matrix at time k is held independent of the hybrid state estimation and is only used in the Belief Matrix Update block at time $k+1$. The belief matrix can also be updated by local information, which is parsed in the Local Information Incorporation block.

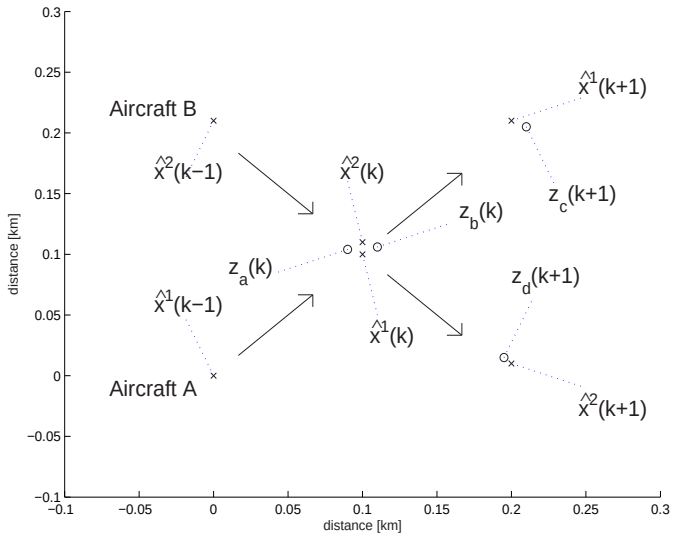


Figure 3.7: State estimates (x) and measurements (o) for two-aircraft example.

3.3.5 Local information incorporation

Local information is useful only if the uncertainty of the belief matrix is reduced, where uncertainty is measured as entropy defined in Section 3.1.3. In Section 3.1.3, it has been shown that local information that identifies one or more targets with absolute certainty can always be incorporated, since such information will never increase entropy. In this

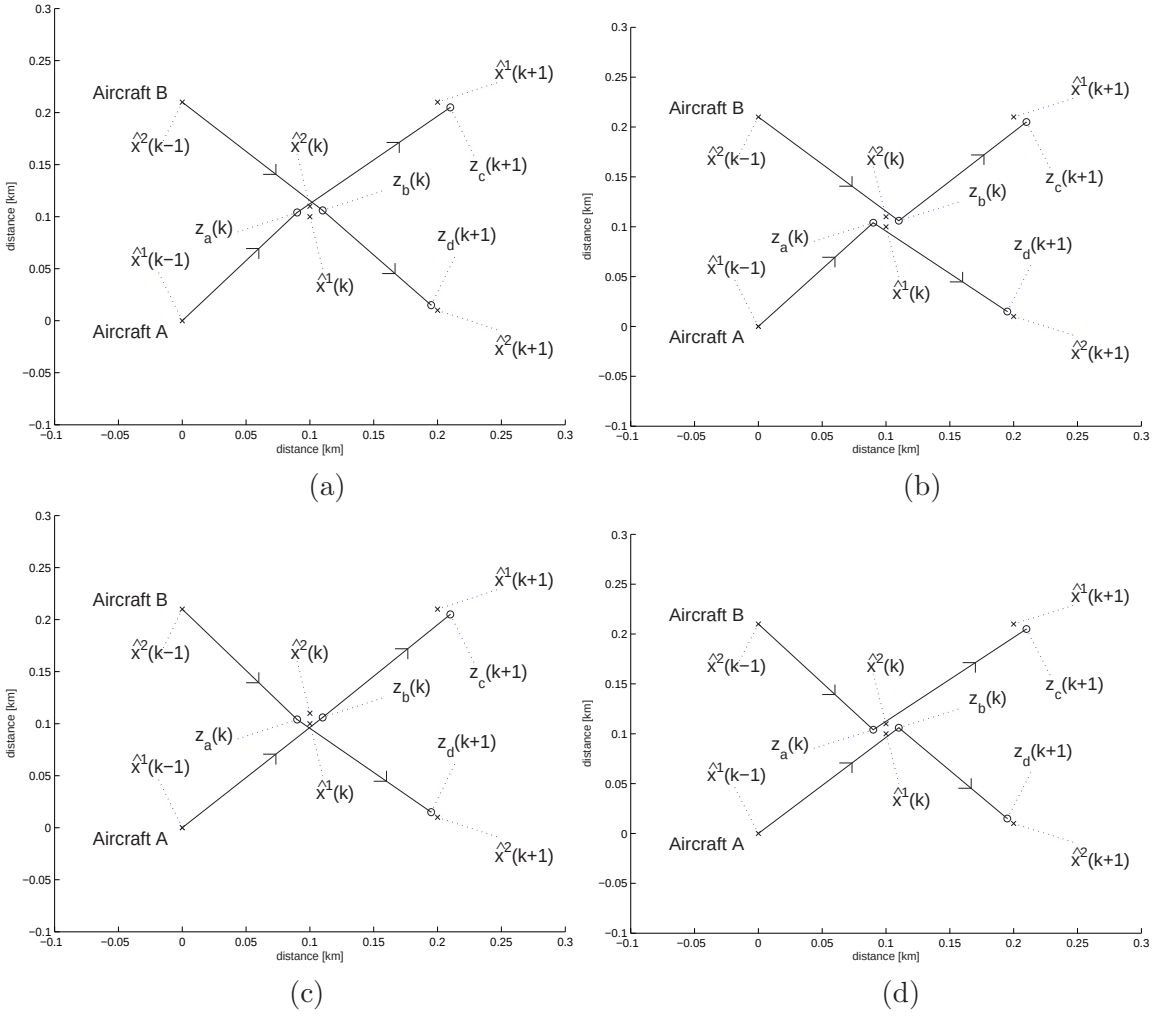


Figure 3.8: (a)-(d): Possible joint events in MHT.

section, an additional set of possible local information is presented. This set is automatically generated whenever targets interact and entropy increases significantly.

Without using extra sensors to get attribute information about the targets to correct target identities, we propose to use the Multiple Hypothesis Testing (MHT) algorithm to get local (attribute) information about interacting targets. The reason for using MHT is that it covers all possible target identity hypotheses. MHT is used only when the minimum diagonal element of the mixing matrix is below a threshold, which we treat as a design parameter. The local information comes from applying the MHT algorithm on track estimates for two time steps. This information is useful in the situation portrayed in Figure 3.7. In this figure,

two aircraft cross perpendicularly at time k. Their estimated positions are marked with x's, while the radar measurements are marked with o's. The expression $\hat{x}^t(l)$ denotes the state estimate for target t at time l. The measurements $z_a(k)$, $z_b(k)$, $z_c(k+1)$, and $z_d(k+1)$ are indexed by letters to reflect that of possibly many choices, two measurements were chosen by MAJPDA at each time step to correspond to the two targets. Aircraft A starts at the bottom left at time $k-1$ and moves to the top right at time $k+1$, while Aircraft B starts at the top left and moves to the bottom right. The assumption is that Aircraft A(B) is Target 1(2) with absolute certainty at time $k-1$. That is, the belief matrix $B(k-1)$ is the identity $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. At time k, the two targets are close together and almost equally likely to be associated with each of two measurements. Assume the mixing matrix $M(k)$, and thus the belief matrix $B(k) = B(k-1)M(k)$, is $\begin{bmatrix} 0.51 & 0.49 \\ 0.49 & 0.51 \end{bmatrix}$ at time k. At time $k+1$, the aircraft have diverged, and the validation gates of the two aircraft no longer intersect. Thus, the mixing matrix $M(k+1)$ from MAJPDA is the identity, and the belief matrix $B(k+1) = B(k)M(k+1)$ remains at $\begin{bmatrix} 0.51 & 0.49 \\ 0.49 & 0.51 \end{bmatrix}$.

The MAJPDA algorithm cannot differentiate between the two measurements at time k; as a result, uncertainty in the belief matrix is essentially maximum. This uncertainty remains even after the aircraft separate. However, from analyzing the dynamics of the two aircraft, a belief matrix with lower entropy can be determined. If the aircraft are assumed to turn at $3^o/$ sec, a common turn rate for commercial jets, neither aircraft can execute a 90^o turn in one time step. The swapping of aircraft-target association is physically impossible. Indeed, the only possible outcome is that Aircraft A(B) remains associated with Target 1(2). This should yield a belief matrix equal to the identity matrix, which is minimum entropy.

The MHT algorithm is utilized to obtain a lower entropy belief matrix than MAJPDA and standard Belief Matrix Updates can achieve. This algorithm is discussed in detail in [17, 80]. Given initial conditions $\hat{x}^1(k-1)$ and $\hat{x}^2(k-1)$, as well as measurements $z_a(k)$, $z_b(k)$, $z_c(k+1)$, and $z_d(k+1)$, there are four possible target-measurement matchings that can occur; these are illustrated in Figure 3.8. Figure 3.8-(a) refers to the outcome chosen by MAJPDA, since Target 1 is assumed to have gone through measurements $z_a(k)$ and $z_c(k+1)$. Each plot in Figure 3.8 is a joint event made up of four events represented by the line segments in the plot. The likelihood of the joint event that each target chooses its pair

of measurements is the product of these individual events. The result is four likelihoods for the four joint events portrayed in the plots of Figure 3.8.

To determine belief, one is only interested in whether Target 1 reaches the expected position of Aircraft A at time $k+1$ or not. Thus, the sum of the likelihoods from Figure 3.8-(a) and (c) is the likelihood that Target 1(2) remains identified as Aircraft A(B); let this quantity be denoted L_1. The sum of the likelihoods from Figure 3.8-(b) and (d) is the likelihood that the targets swap identities; let this quantity be denoted L_{-1}. Because a 90^o turn in one time step is not allowed in the dynamic models of the aircraft, $L_{-1} = 0$.

The doubly stochastic version of the matrix $\begin{bmatrix} L_1 & L_{-1} \\ L_{-1} & L_1 \end{bmatrix}$ represents the mixing matrix for the two aircraft between time steps $k-1$ and $k+1$. This matrix, a two-step mixing matrix, is denoted as $\Gamma(k+1)$. For the example presented, $\Gamma(k+1)$ is the identity. Thus, the belief matrix determined by MHT at time $k+1$ is $B'(k+1) = B(k-1)\Gamma(k+1) = \begin{bmatrix} L_1 & L_{-1} \\ L_{-1} & L_1 \end{bmatrix}$. The resulting belief matrix $B'(k+1)$ is the identity, which has lower entropy than the $B(k+1)$ from the standard MTIM model. The local information can thus be incorporated through the Belief Matrix Update block of MTIM.

Because there is no guarantee that automated MHT local information will improve the entropy of the belief matrix, it is only incorporated only if this local information decreases entropy. This automated local information and identity local information are both handled by the Incorporate Local Information block of the MTIM algorithm. The next section applies the MTIM algorithm to scenarios involving multiple aircraft in a cluttered environment.

3.3.6 Simulation results

Two examples are presented below to demonstrate the efficacy of the MTIM algorithm in clutter. Both examples are scenarios where multiple aircraft are interacting in a cluttered environment. In both examples, several system parameters are set to the same values. First, measurement points are made available every 5 seconds. Measurement covariance R is $\begin{bmatrix} (100)^2 & 0 \\ 0 & (100)^2 \end{bmatrix}$, which means the standard deviation of position error is $100m$ in both

dimensions. Process noise is set to be $\begin{bmatrix} 0.001 & 0 \\ 0 & 0.001 \end{bmatrix}$ for straight flight and $\begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$ for turning mode. The above constants are realistic values for aircraft in clutter and are taken from [83]. Clutter is uniformly distributed in space and Poisson distributed in number; the density of clutter points is $0.5 * 10^{-6}$ clutter points per square meter, or 0.5 points per square kilometer. The validation gate parameter γ is set to 9.2, which would correspond to a 3σ confidence level if residual covariance S were used. The effective residual covariance S' that is actually used is determined with system constants τ and ν set to 3 and 6, respectively. The threshold for initiating MHT is set to 0.75. In both examples, the target 1 is initially identified as Aircraft A, 2 as B, and so on.

The first example is an extreme (acrobatic) scenario where four aircraft fly at each other directly and maneuver; this example is useful in understanding the capabilities of the MTIM algorithm. Figure 3.9 (top) shows a shot of the radar screen including the entire flight data, but without the trajectories explicitly indicated. This gives us an idea of the clutter density, as well as how unclear the system is, especially when the aircraft come close to each other. Figure 3.9(center) displays the actual and estimated positions of four aircraft following symmetric paths that first converge, then maneuver around a common point, and finally diverge. The dashed lines with dots as markers are the noisy measurements from the targets. The solid lines with markers as shown in the legend are the estimated positions found by MTIM. The fainter dots interspersed throughout the plot are clutter points. Aircraft A, B, C, and D fly with constant velocity of 200 m/s (or 390 knots). All turns are executed at 3^o/sec. Target tracking is accurate except for overshoot when aircraft start turning. Indeed, the dashed lines depicting the noisy measurements are not clearly visible because the solid lines depicting estimated target positions match them almost exactly. Figure 3.9(bottom) displays the evolution of the belief matrix in graphical form. The plots, from top to bottom, show the probability that any aircraft is identified with targets 1 through 4, respectively. From this figure, it is clear that the belief matrix is unchanged while the aircraft are distant from and not interacting with each other. When paths cross, the belief matrix is changed significantly only if the measurements for both targets happen to nearly coincide. For example, at time 30, targets 1 and 2 nearly coincide, leading to the belief that both targets 1 and 2 are nearly 0.5 Aircraft A and 0.5 Aircraft B. However, the automated MHT local information generated by this interaction restores the belief matrix to nearly identity at the following time step. At time 30, targets 3 and 4 also interact with equally

drastic loss of identity between Aircraft C and D. Again, the local information restores the belief matrix at the following time step. At time 32, targets 1 and 3 interact, with similar jump in belief matrix entropy followed by belief matrix restoration from local information. Targets 2 and 4 also interact in the same fashion at time 32. The scenario depicted in Figure 3.9 establishes the efficacy of the MTIM algorithm in clutter.

The next example is designed to demonstrate the effectiveness of the MTIM algorithm in a more realistic free flight scenario. There are six aircraft flying in both straight and turning modes. We show the measurements obtained with clutter in Figure 3.10(top). The trajectory plot in Figure 3.10(center) is set up the same way as that in Figure 3.9(center). This plot includes realistic accident scenarios. For example, the intersection of Aircraft A and B at coordinates (20,10) depict a blunder by Aircraft B into Aircraft A's path. Like the four aircraft scenario, tracking is successful for the six aircraft scenario except for overshoot at the start of a maneuver. The dashed lines for the target measurements are visible at these overshoot points. Identity management also performs well; part of the belief matrix is shown in Figure 3.10(bottom). Only the belief for the first two targets is shown. One can see the interaction between Aircraft A and B at time 21, which leads to an increase in the entropy in the belief matrix followed by restoration of the belief matrix through automated local information incorporation. Aircraft A and F have a similar interaction at time 37; only the changes in the belief matrix for target A are shown in Figure 3.10(bottom). The interaction between Aircraft B and C at time 27 is of note because the interaction is mild. The belief matrix is not changed enough to trigger the automated local information, so the belief of Target 2 is not restored. Because the interaction is mild, one can confidently label target 2 as Aircraft B. The identity management portion of MTIM performs successfully. Indeed, MTIM performs successfully for both examples shown.

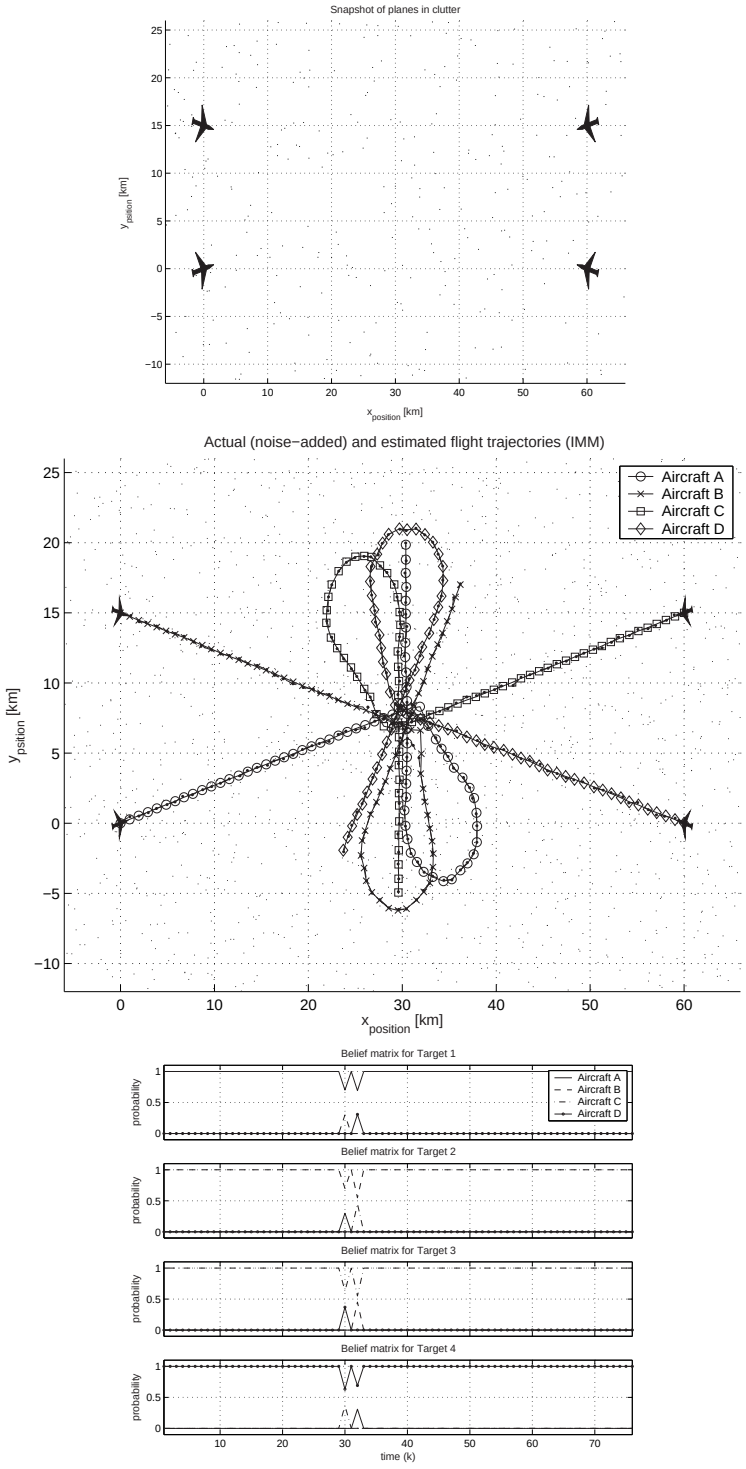


Figure 3.9: Measurement points with clutter (top), aircraft trajectories (center) and accompanying belief matrix plot (bottom).

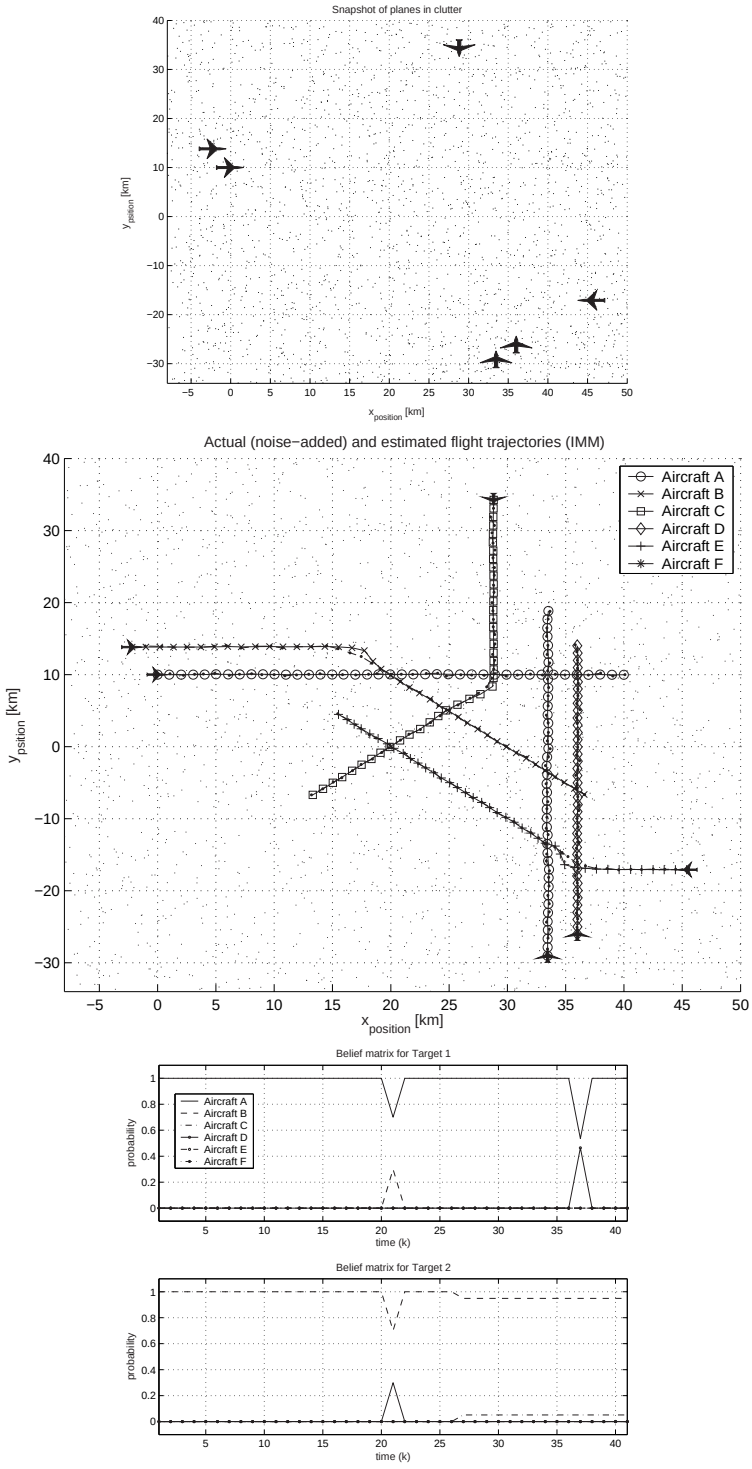


Figure 3.10: Measurement points with clutter (top), aircraft trajectories (center) and plot of belief information for Aircraft A and B (bottom).

Chapter 4

Flight-Mode-Based Conflict Detection

In the previous two chapters, we have derived hybrid state estimates computed using the Residual-Mean Interacting Multiple Model (RMIMM) algorithm, and used them in the Multiple-Target Tracking and Identity Management (MTIM) algorithm, as shown in Figure 1.1. In this chapter, we consider 2D aircraft conflict detection problems in which the aircraft are assumed to fly at the same altitude. To the best of our knowledge, all existing conflict detection algorithms are based on the continuous state information of the aircraft (see Kuchar and Yang [11] for a comprehensive survey). For example, Yang *et al.* [15] propose a conflict alerting logic based on sensor and trajectory uncertainties, with conflict probability based on Monte Carlo simulation; Paielli *et al.* [3] and Prandini *et al.* [84] propose analytic algorithms for computing the probability of conflict. The performance of these conflict detection algorithms depends strongly on the accuracy of state estimates. Here, we propose conflict detection algorithms using the aircraft's current continuous state estimates as well as flight mode (discrete state) estimates computed by the RMIMM algorithm. This algorithm is applicable to both ground and airborne control scenarios in ATC – to ground control using radar information, and to airborne control using information from Automatic Dependent Surveillance-Broadcast (ADS-B) data [85].

For conflict detection, we use the aircraft model derived in Section 2.1.3 which has two flight modes: constant velocity and coordinated turn. ADS-B information is assumed to be used

for measurements, and thus both aircraft's position and velocity information are available for state estimation. However, as shown in the previous chapter, there is no restriction in using only the aircraft's position information to estimate the aircraft's current states with RMIMM.

In this chapter, we develop a probabilistic conflict detection algorithm using an analytic equation, which can compute conflict probabilities efficiently. This analytic equation is based on the algorithm in [3] (which is based only on the continuous state) which we extend to incorporate flight mode estimates. We use RMIMM for state and mode estimation. Using Enhanced Traffic Management System (ETMS) data, we show that even in cases in which the turn mode is a small portion of the whole flight trajectory, the accuracy of this hybrid conflict detection algorithm is better than that of continuous schemes, especially in the airspace around waypoints and airports where several airways converge. In addition, the flight mode estimate could also be used for *blunder detection*, that is, it could be used to allow an aircraft (or a ground controller) to detect a conflict early enough to take (or issue) a safe resolution maneuver if a neighboring aircraft (blunderer) starts a maneuver which might cause a conflict with itself (or another aircraft).

4.1 Conflict detection

In this section, we use RMIMM to estimate current states and flight mode with the aircraft models derived in Section 2.1.3. Then, we predict conflicts within a look-ahead time ($T_{LA} < \infty$) using these estimates. A conflict is defined by an event in which the relative separation of two aircraft becomes less than a predefined safety distance (R). We set $R := 5$ nautical miles (nm) in this paper. For conflict detection, we project each aircraft's current state into the future in two ways: a nominal state projection which constructs a single trajectory based on the current state vector without prediction uncertainties, and a probabilistic state projection in which uncertainties in the model are used to construct a set of possible future trajectories, each weighted by its probability of occurring. However, the nominal state projection method is used for comparison only.

4.1.1 Nominal conflict detection

We perform conflict detection by projecting the aircraft's current velocity estimate, obtained by RMIMM, into the future. We project along a single trajectory which is a straight line if

the current flight mode estimate is CV mode or a circular arc with an estimated yaw rate if the estimate is CT mode. To validate the nominal conflict detection algorithm, we use the prediction uncertainty model in [3]. The trajectory prediction error is modelled as normally distributed with a zero mean, and a covariance with eigenvectors in the along-track and cross-track directions, as shown in Figure 4.1. The along-track error is modelled as a 15 knots (kn) standard deviation speed uncertainty. The cross-track error grows from its initial uncertainty of $15m$ at the aircraft's current position (which is the position estimation error bound of RMIMM) to a steady-state error with a standard deviation of $1nm$. The growth rate is $\frac{v}{57}nm/min$ (where v is an aircraft speed). This growth rate is equivalent to a lateral deviation error with a standard deviation of 1^o [15]. Thus, the prediction error covariance matrix in the body-fixed frame aligned with the aircraft heading is a diagonal matrix.

4.1.2 Probabilistic conflict detection

For probabilistic conflict detection, we need to compute the probabilities of all possible conflicts. In [3], the trajectory prediction errors are modelled as Gaussian random variables with zero mean and time-varying covariances. Through coordinate transformation, an analytical solution for conflict probability is derived based on the assumption that aircraft fly straight with constant velocity. Since our conflict detection algorithm allows the aircraft to be in CT mode, we suggest a modified conflict probability computation from that in [3] when aircraft are in CT mode.

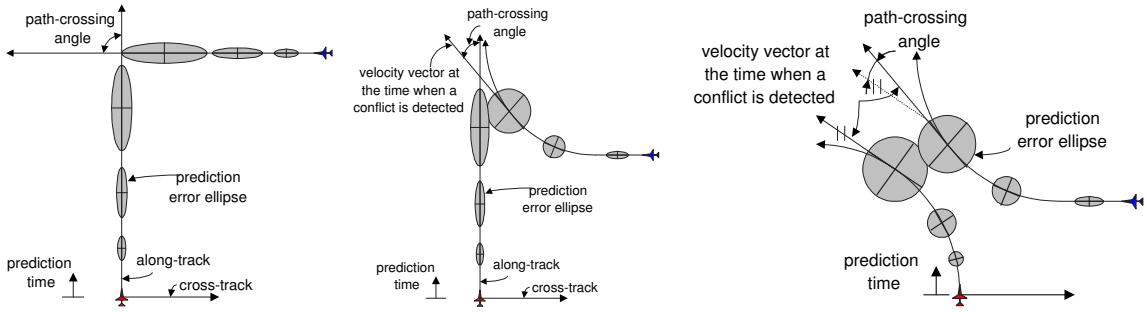


Figure 4.1: Trajectory prediction error ellipses. (a) Both aircraft are in the constant velocity mode. (b) An aircraft is in constant velocity mode and the other is in the coordinated turn mode. (c) Both aircraft are in the coordinated turn mode, motivated by the single mode computations in [3].

First, we summarize the conflict detection algorithm in [3] (in which the authors consider only the case in which both aircraft are in CV mode). The trajectory prediction error is modelled as a zero-mean normal distribution, with diagonal covariance matrix S in the body-fixed frame aligned with the aircraft heading. If a rotation matrix R_{rot} transforms the heading-aligned body-fixed frame to the reference frame, the transformed covariance matrix is

$$Q := R_{rot} S R_{rot}^T \tag{4.1}$$

Two error covariances for an aircraft pair can be combined into a single equivalent covariance, which is assigned to one of the aircraft (which we call the "stochastic aircraft"). The other aircraft, called the "reference aircraft", is assumed to have no position uncertainty. Then, the combined prediction error covariance is

$$M := Q_s + Q_r - Q_{sr} \tag{4.2}$$

where Q_s and Q_r are the covariance of the stochastic aircraft and the covariance of the reference aircraft, respectively. Q_{sr} is the cross-correlation between the two aircraft. Through

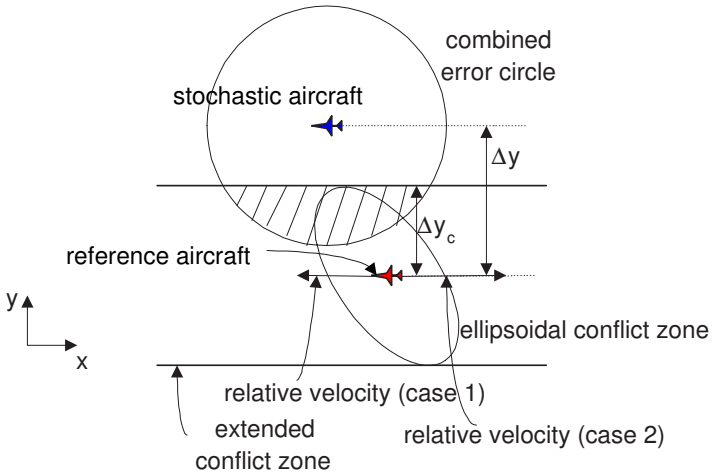


Figure 4.2: Transformed conflict geometry [3].

coordinate transformation, we make the combined covariance ellipse centered on the stochastic aircraft a unit circle and the relative velocity vector aligned in the x-direction as shown in Figure 4.2. The elliptical conflict zone is centered on the reference aircraft. If the

relative heading angle of two aircraft is constant during the conflict, the total conflict probability is the volume of the probability density function over the shaded area. Since the combined error covariance ellipse is a unit circle, the two dimensional Gaussian probability density function decouples into the product of two identical one-dimensional functions: $p(x, y) = p(x)p(y)$ where $p(x) = \exp(-x^2/2)/\sqrt{2\pi}$. Thus, the conflict probability P_c is

$$\begin{aligned} P_c &= \int_{-\Delta y+\Delta y_c}^{-\Delta y-\Delta y_c} \int_{\infty}^{\infty} p(x,y)dxdy \\ &= \int_{-\Delta y+\Delta y_c}^{-\Delta y-\Delta y_c} p(y)dy \int_{\infty}^{\infty} p(x)dx \\ &= \int_{-\Delta y+\Delta y_c}^{-\Delta y-\Delta y_c} p(y)dy \end{aligned} \tag{4.3}$$

Equation (4.3) is valid only in the case in which both aircraft are in CV mode (Figure 4.1-(a)). Therefore, for the conflict cases in Figure 4.1-(b) and Figure 4.1-(c), we change the prediction model (i.e., prediction error covariance S). If an aircraft is in CT mode, both the along-track error and the cross-track error are modelled as a 15 kn standard deviation speed uncertainty because the aircraft's acceleration is composed of tangential (along-track) acceleration and centripetal (cross-track) acceleration. The relative heading is assumed to be constant during the conflict. We use the relative heading angle at time t_{md}, when the predicted minimum distance between aircraft is achieved, as the pass-crossing angle. The latter assumption is coarse, but since typical civilian aircraft yaw rates are not large ($\approx 3^o/sec$) and are of short duration, the computation error may be small. So, it would be reasonable to use the analytical solution in (4.3) instead of a numerical one, which could be computationally intensive. However, to reduce the error in conflict probability computation, we introduce a time interval around the time t_{md} ($[t_{md} - \delta, t_{md}, t_{md} + \delta]$, $\delta > 0$), compute the conflict probability distribution within this interval, and choose the maximum value as the conflict probability.

Two main points about this algorithm warrant additional conditions for conflict detection. First, the conflict probability computation algorithm in [3] is based on the conflict geometry, and uses the extended conflict zone as shown in Figure 4.2 to compute the conflict probability. Thus, the conflict probability when the relative velocity of the reference aircraft is pointing towards the stochastic aircraft (case 1) is the same as that of the case in which the relative velocity of the reference aircraft is pointing away from the stochastic aircraft (case 2). Second, the proposed conflict detection algorithms can be applied to both on-board applications and ground control applications. If the conflict detection algorithms

are applied in ground control applications, they may handle large numbers of aircraft at one time. Thus, to eliminate the above error and to reduce the number of aircraft pairs for conflict detection computation, we propose the following conditions.

(Conditions for Conflict Detection Computation)
P_{th} is defined as a conflict probability threshold and used as a design parameter. $\vec{r}_{ij}$ is defined as a relative position vector from aircraft i to aircraft j and $\vec{v}_i$ as a velocity vector of aircraft i.

1. (a) The conflict detection algorithm is executed every τ $(0 < \tau < \infty)$ seconds.
   (b) If mode changes are observed within the time interval τ, the conflict detection algorithm is executed instantly for the corresponding aircraft pairs.
   (c) For aircraft pairs with conflicts with $P_c \geq P_{th}$, the conflict detection is executed every τ_c $(0 < \tau_c < \tau)$ seconds.

2. (a) The conflict detection algorithm is executed for aircraft pairs satisfying $(\forall i, j)$
      i. CASE I: Two aircraft fly generally towards each other:
      $[(\vec{r}_{ij}(k)\cdot\vec{v}_i(k) \geq 0)\wedge(\vec{r}_{ij}(k)\cdot\vec{v}_j(k) \leq 0)]\wedge[sign(\vec{r}_{ij}(k)\times\vec{v}_i(k)) \neq sign(\vec{r}_{ji}(k)\times\vec{v}_j(k))]$
      ii. CASE II: An aircraft follows another aircraft:
      $[(\vec{r}_{ij}(k)\cdot\vec{v}_i(k) \geq 0)\wedge(\vec{r}_{ij}(k)\cdot\vec{v}_j(k) \geq 0)\wedge(\|\vec{v}_i(k)\| > \|\vec{v}_j(k)\|)]$
      $\wedge[(\vec{r}_{ij}(k)\cdot\vec{v}_i(k) \leq 0)\wedge(\vec{r}_{ij}(k)\cdot\vec{v}_j(k) \leq 0)\wedge(\|\vec{v}_i(k)\| < \|\vec{v}_j(k)\|)]$
   (b) The conflict probability is computed only when the above Condition 2-(a) is satisfied and the predicted minimum distance within the look-ahead time is less than the test safety distance ($\bar{R} := \alpha R$, $\alpha \geq 1$).

We set $\tau = 20sec$ and $\tau_c = 5sec$ for simulation. Condition 1-(b) corresponds to event-driven simulation, i.e., the simulation (or computation) is triggered not by a time instant but by an event (mode change in this case) [86, 87]. Thus, we can reduce computational load significantly. However, for safety, even if there is no mode change, we perform conflict detection every τ seconds because measurements contain noise and thus the conflict detection may have prediction errors. We also pay more attention to aircraft pairs which have conflict probabilities over the threshold for safety. Condition 2 tests whether an aircraft

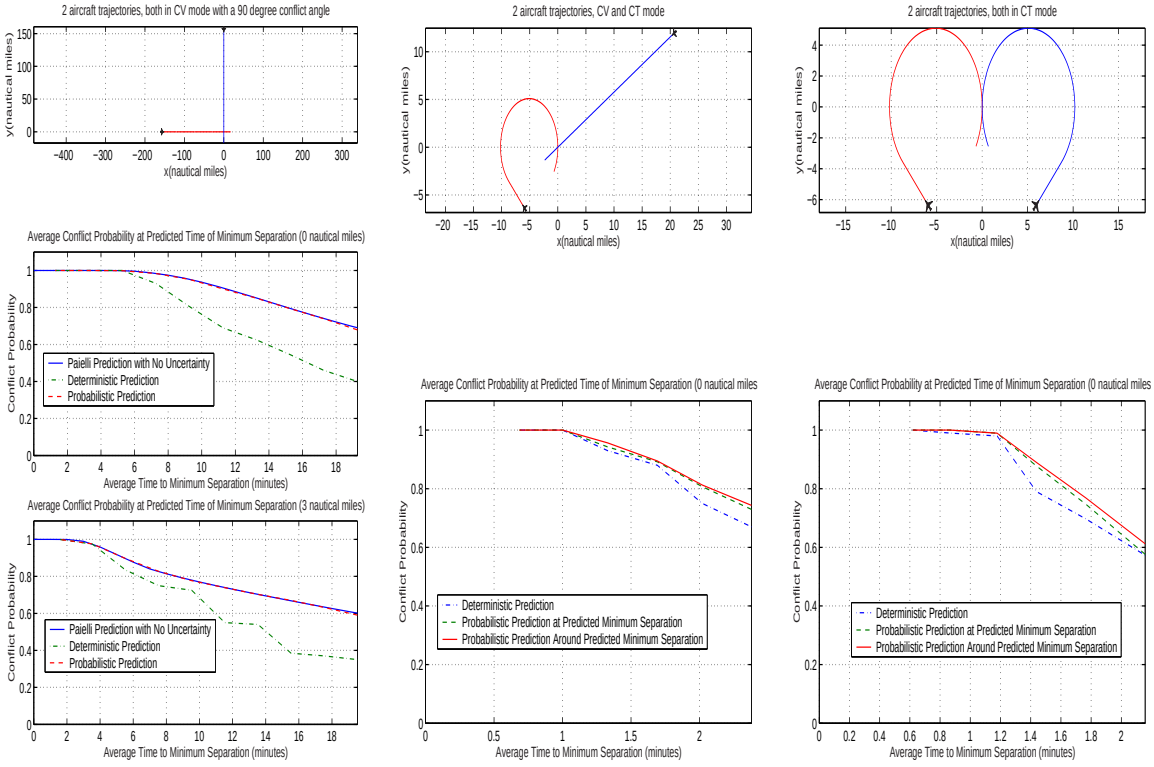
pair may have a conflict based on the current aircraft's state and mode estimates. We can then filter out many aircraft pairs before applying the conflict detection algorithms. Since the predicted minimum distance is not exact due to prediction uncertainties and current state estimation errors, we compute conflict probability when it is less than the "test safety distance" ($\bar{R} \geq R$) so that we do not miss possible conflicts. Here, we set $\alpha = 1.5$.

4.2 Numerical examples and validation with ETMS data

In this section, we apply the flight-mode-based conflict detection algorithm to various multiple-aircraft conflict detection scenarios and validate it with Enhanced Traffic Management System (ETMS) data. The Federal Aviation Administration's (FAA's) ETMS database contains all flight plan information for flights in the National Airspace System (NAS). The flight information for an airborne aircraft in the NAS in the ETMS database contains: time, flight number, aircraft type, latitude, longitude, speed, altitude, heading, and flight plan information [88]. Data are collected from the entire population of flights in the NAS with filed flight plans. The FAA uses these data to monitor the effectiveness of its National Route Program, in which the user community is offered flexible, cost-effective routing options as an alternative to published ATC preferred routes. The ratio of flights not subject to ATC preferred routes to total flights is considered to be a measure of system flexibility, and is included as an index in the FAA's Air Traffic Services Performance Measures. In order to improve conflict detection performance around waypoints, a conflict detection algorithm using waypoint information is derived and tested with ETMS data.

4.2.1 Numerical examples for multiple-aircraft conflict detection

We apply the proposed conflict detection algorithms in Section 4.1 to the scenarios of Figure 4.1, through many simulations. The RMIMM algorithm is used to estimate the aircraft's current state and flight mode. We assume the aircraft speed is constant at $v = 480kn$ for all simulations unless otherwise stated. The predefined safety distance is $5nm$. For prediction, the along-track and the cross-track RMS errors grow linearly at a rate of $0.25nm/min$ and $\frac{v}{57}nm/min$ from the initial uncertainty $15m$. Yet, the cross-track RMS error saturates when it reaches $1nm$ [3]. The RMS errors of position and velocity measurements are assumed



(a) (b) (c)

Figure 4.3: 200 Monte Carlo simulation results using nominal and probabilistic conflict detection algorithms. (a) Both aircraft are in the constant velocity mode. (b) An aircraft is in constant velocity mode and the other is in the coordinated turn mode. (c) Both aircraft are in the coordinated turn mode.

to be $50m$ and $3m/sec$ respectively. Wind-error cross-correlation between aircraft (Q_{sr}) in (4.2) is assumed to be zero.

In Figure 4.3, the deterministic conflict detection probability refers to the probability of successful conflict detection. In Figure 4.3-(a), both aircraft are in CV mode. The top graph shows the conflict geometry with the path-crossing angle at 90^o. The middle and bottom plots show the average conflict probability as a function of the average time required to get to the predicted minimum separation point, where the minimum separation is 0 (collision) and $3nm$ respectively. The conflict probability of the probabilistic prediction is close to that of Paielli and Erzerberger's algorithm [3]. Thus, we conclude that RMIMM used in the proposed probabilistic conflict detection algorithm produces good state estimates since the conflict probability of Paielli and Erzerberger's algorithm assumes that there is no

uncertainty in the current state.

In Figure 4.3-(b), only one aircraft is in CV mode, with a heading angle of 210^o. The other aircraft begins in CV mode for a little bit and then switches to CT mode for most of its trajectory. The turn rate of the aircraft in CT mode is $1.5^o/sec$. There is a collision at the origin. The bottom plot shows the computed conflict probability versus the average time to minimum separation. The computation starts after the second aircraft switches to CT mode so that both aircraft predict that there is some conflict in the future. At 6 regularly spaced points on the trajectory, we project the current state estimate to find a predicted minimum separation point and compute the conflict probability using the deterministic and probabilistic method. The dotted line represents the conflict probability at the predicted minimum separation. The continuous line comes from the maximum conflict probability in an interval with half width 5 seconds (i.e., $\delta = 5$) around the time of predicted minimum separation. The difference is small, but we can reduce the conflict probability computation error due to the assumption that the heading angle of aircraft is constant during the conflict. In Figure 4.3-(c), both aircraft are in CT mode and the plots are analogous to those in Figure 4.3-(b). Figure 4.3 shows that probabilistic conflict detection is more accurate than the deterministic one, especially when the look-ahead time is long.

We can interpret the conflict scenarios in Figure 4.1-(b) and in Figure 4.1-(c) as blunder detection, because the two aircraft are close to each other and the conflict time is very short compared to the look-ahead time. If an aircraft can detect that another aircraft (blunderer) starts a sudden maneuver that may cause a conflict within a short time, the aircraft can take a safe resolution maneuver. Blunder detection could be useful for conflict detection in congested airspace such as around waypoints and airports, and for parallel landings [89]. This is one of the advantages of using flight mode information for conflict detection over existing conflict detection methods. Figure 4.4 shows a blunder detection scenario for two aircraft. Initially two aircraft fly in parallel. Then one aircraft (blunderer) changes its heading. The trajectory interval in which the aircraft violate the $5nm$ safety zone is shown by the thick black lines. On the right figure, we plot the conflict probability as a function of the time of trajectory. The circles here correspond to the same time instants as the circles in the left figure. Thus, we can see how the conflict probability changes at points along the trajectory. This result was obtained with 200 simulations. One of the problems we observed was that even though the aircraft are no longer in conflict and heading away from

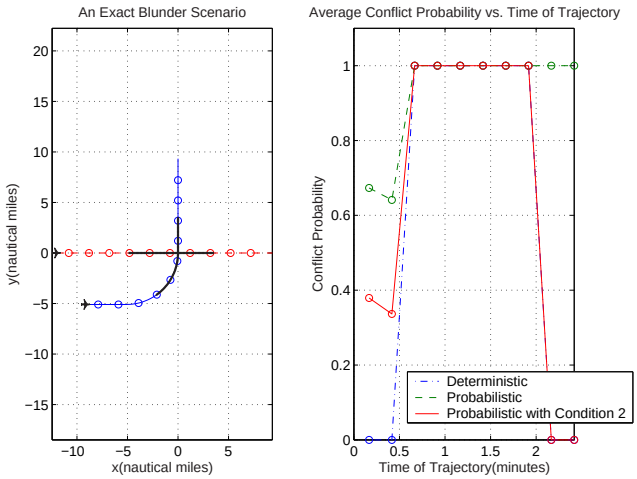
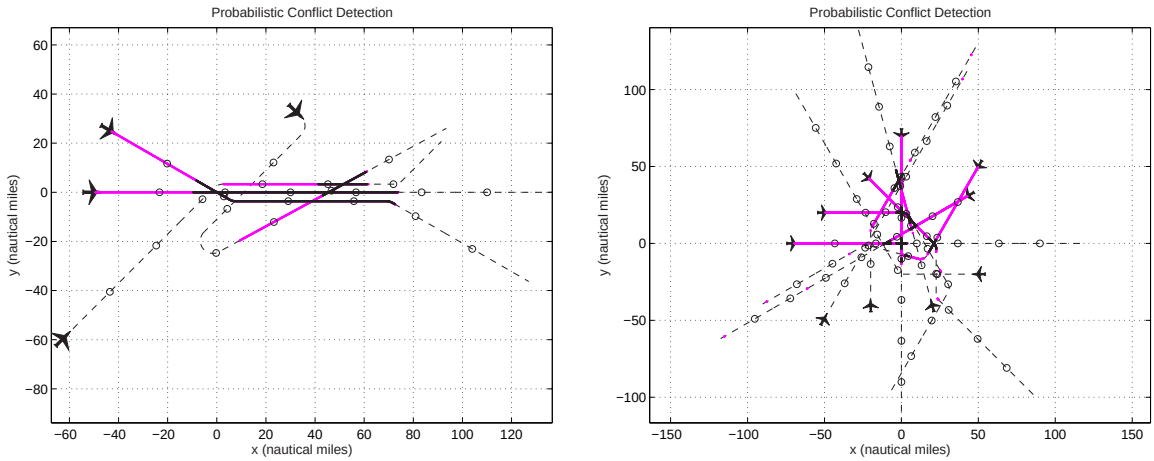


Figure 4.4: Blunder detection scenario for two aircraft.

each other at the end of the trajectory, the conflict probability algorithm in [3] still returns a conflict probability of one. Thus, we use Condition 2-(b) to eliminate this phenomenon and the continuous line in Figure 4.4 shows this correction. Figure 4.5 shows multiple-aircraft



(a) (b)

Figure 4.5: Simulations for multiple-aircraft conflict detection. (a) Structured airspace scenario (four aircraft). (b) Free Flight scenario (ten aircraft).

conflict detection scenarios. We set the look-ahead time to 20 minutes and the conflict probability threshold (P_{th}) to 0.7. The conditions for conflict detection computation are applied to reduce the number of aircraft pairs for conflict detection. We perform conflict

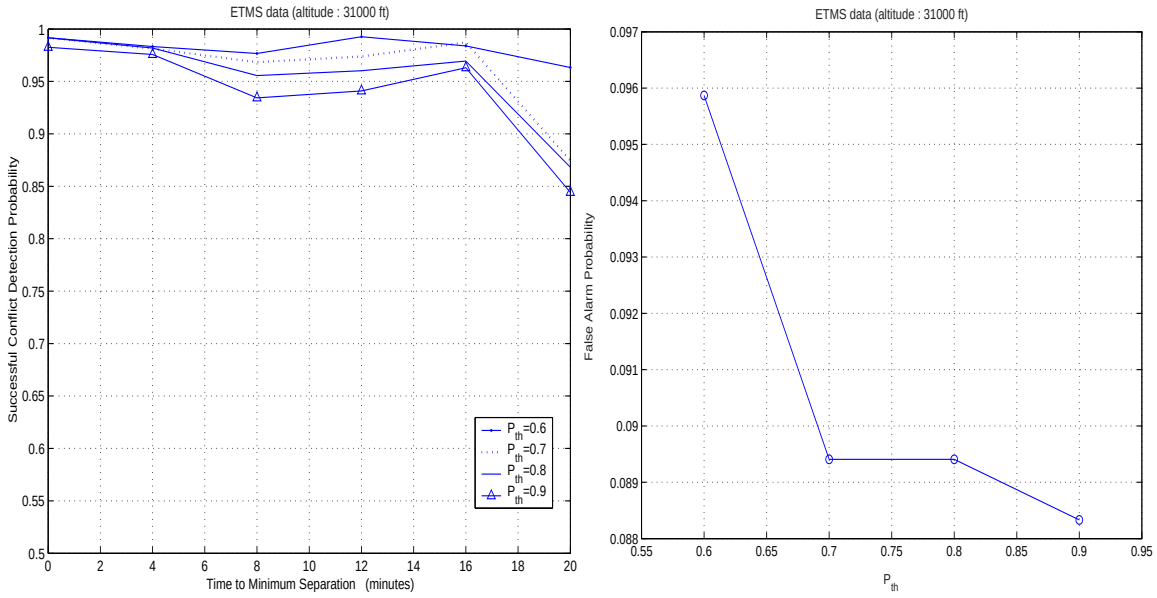
detection every 20 seconds ($\tau = 20sec$) even if there is no flight mode change. For the aircraft pairs where conflict probability is greater than or equal to $P_{th} = 0.7$, conflict detection is performed every 5 seconds ($\tau_c = 5sec$).

There are four aircraft in the multiple-aircraft conflict scenario for structured airspace in Figure 4.5-(a). The circles correspond to positions that are 100 seconds away from each other. All the aircraft fly at $8nm/min$ and have turn rates of $1.5^o/sec$. The grey regions indicate where the aircraft detect conflict. The black regions are where the aircraft are actually in conflict. The Free Flight scenario in Figure 4.5-(b) has the same parameters except that there are ten aircraft. Probabilistic methods are used for both scenarios. Here the circles are spaced 200 seconds apart for clarity. In these simulations, the conflict detection algorithm is implemented in a centralized way, so that it performs aircraft tracking and conflict detection for all the aircraft.

Since computational efficiency is important for real-time air traffic control, we compute the average time required for both the state estimation and the conflict probability computation. The average computation times for two-aircraft tracking using IMM and RMIMM are 0.01 seconds and 0.013 seconds respectively. The average computation time for conflict probability for two-aircraft conflict is 0.0041 seconds. These values are obtained using MATLAB on a 500 MHz Pentium III PC.

4.2.2 Validation with ETMS data

In this section, we validate the flight-mode-based conflict detection algorithm with ETMS data. We use ETMS data for the Oakland center from track time $921735971sec$ to $921756011sec$ which starts on January 1, 1970 ending approximately 5 hours 34 minutes later. Since currently air traffic is stratified in altitude and we consider 2D conflict detection problems in this paper, ETMS data for a fixed altitude is used for simulations. We choose $31,000ft$ since it is among the busiest altitudes in air traffic. Figure 4.6-(a) and (b) show two of the performance measures of the proposed conflict detection algorithm – the successful conflict detection probability (the probability that the conflict detection algorithm correctly detects conflict), and the false alarm probability (the probability that the conflict detection algorithm detects a conflict when in reality there is none). From Figure 4.6-(a) and (b), we see that the successful conflict detection probability tends to increase as the look-ahead time



(a) (b)

Figure 4.6: Performance analysis of the flight-mode-based conflict detection algorithm with ETMS data (altitude $31,000ft$). (a) Successful conflict detection probability. (b) False alarm probability.

decreases. This means that most conflicts can be detected within a certain look-ahead time. As the conflict probability threshold increases, the successful conflict detection probability decreases, and the false alarm probability also decreases. Thus, we need to make a trade-off between the successful conflict detection probability and the false alarm probability by varying the conflict probability threshold (P_{th}). A system operating characteristic (SOC) curve shown in Figure 4.7 clearly shows how to choose a good conflict probability threshold. Based on the SOC curve in Figure 4.7, we justify our choice $P_{th} = 0.7$, since it gives a high successful conflict detection probability and yet a small false alarm probability.

The proposed conflict detection algorithm is applied to various scenarios based on ETMS data. Figure 4.8 shows one such scenario. In this simulation, air routes around *Mina* and *Clovis* are considered. These two airspace fixes are the waypoints which have the most complex air traffic in Oakland center's airspace. Several air routes from other centers converge at *Mina*, and the converged air route leads to the Oakland airport. Three different air routes intersect at *Clovis*. In Figure 4.8, four aircraft are approaching *Mina*. AC1 and AC2 are going to have a conflict with each other near *Mina*. The conflict detection

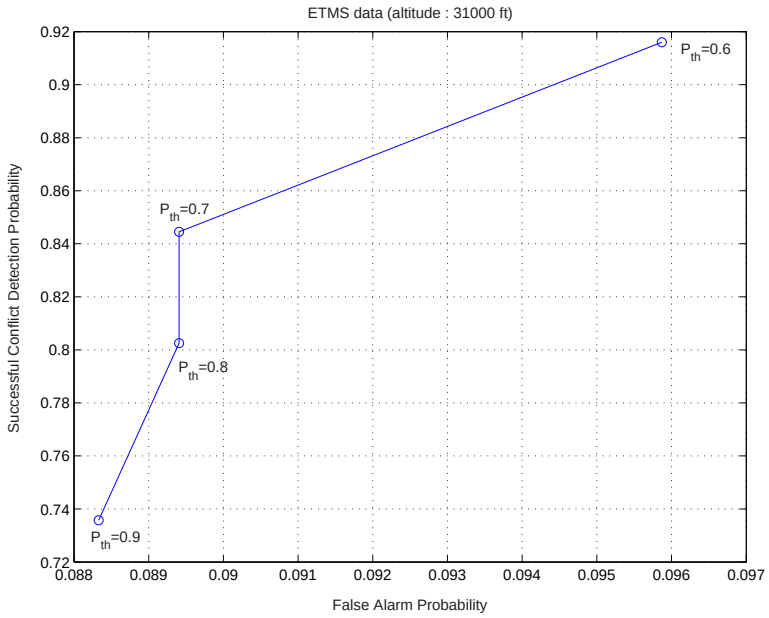


Figure 4.7: System operating characteristic curve (Successful conflict detection probability vs. False alarm probability).

algorithm detects the conflict and it also detects another conflict between AC3 and AC4 around *Coaldale*. A conflict between AC5 and AC6 is denoted as $\times$ in Figure 4.8. In Figure 4.8, the dotted lines represent the trajectories of the aircraft, the thick grey lines indicate where the aircraft detect conflict, and the thick black lines denote where the aircraft are actually in conflict. However, the conflict detection algorithm does not detect the conflict around *Clovis* until AC6 changes its heading around *Clovis* because it does not consider the heading change of AC6 around *Clovis* . The conflict detection algorithm does detect the conflict only after AC6 passes *Clovis* and changes its heading toward AC5. This delayed conflict detection is due to the fact that the conflict detection algorithm does not use waypoint information; in this case around *Clovis*. The proposed conflict detection algorithm uses only aircraft's current state and mode estimate for conflict detection based on the assumption that all aircraft keep current speed and heading. However, this may not be true when the look-ahead time is long or aircraft are close to their waypoints. Therefore, a conflict detection algorithm using waypoint information would be useful for conflict detection in airspace around waypoints, or for midterm conflict detection, in which the look-ahead time is long. In the next section, we develop an extended conflict detection algorithm using waypoint information.

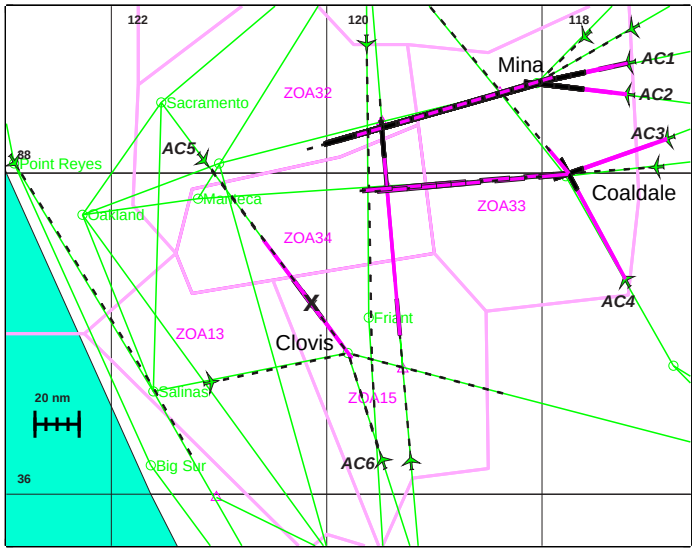


Figure 4.8: Multiple-aircraft conflict detection around waypoints *Mina* and *Clovis*.

4.2.3 Conflict detection using waypoint information

We now consider waypoint information in order to compute conflict probabilities. We consider the case shown in Figure 4.9-(a), in which an aircraft passes through a waypoint within the look-ahead time (T_{LA}). In this case, the projected trajectory is composed of three segments: $\overline{w_0q_1}, \overline{q_1q_2}$, and $\overline{q_2w_2}$. The trajectory prediction errors are modelled as Gaussian random variables with zero means and time-varying covariances, as in Section 4.1.2. Therefore, the trajectory prediction uncertainty ellipses for three segments of the projected trajectory should be computed in order to compute conflict probabilities.

If an aircraft is supposed to be on a straight path, i.e., $\overline{w_0q_1}$ or $\overline{q_2w_2}$, the along-track error is modelled as a $15kn$ standard deviation speed uncertainty. The cross-track error grows from the initial uncertainty to a steady-state error with a standard deviation of $1nm$ if the initial uncertainty is less than $1nm$; if the initial uncertainty is greater than or equal to $1nm$, the standard deviation of the cross-track error is the same as the initial uncertainty. If an aircraft is supposed to be on a circular arc, i.e., $\overline{q_1q_2}$, both the along-track error and the cross-track error are modelled as a $15kn$ standard deviation speed uncertainty so that it accommodates the uncertainty about a turning maneuver around a waypoint.

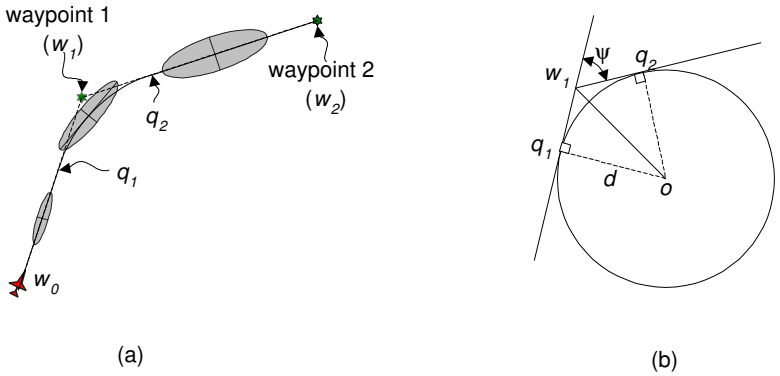


Figure 4.9: (a) Trajectory prediction error ellipses when an aircraft passes a waypoint within the look-ahead time. (b) Geometry of the turning maneuver around a waypoint.

Now, we consider the question of how to find three segments of the projected trajectory, i.e., how to find q_1 and q_2. From Figure 4.9-(b), we can compute the distance between q_1 and w_1, $\|\overline{q_1 w_1}\|$:

$$\|\overline{q_1 w_1}\| = \frac{v}{\omega_0} \tan \frac{\psi}{2} \tag{4.4}$$

where $\omega_0 = 3^o/sec$ is a nominal angular velocity. The two triangles in Figure 4.9-(b) are symmetric, and so $\|\overline{q_1 w_1}\| = \|\overline{w_1 q_2}\|$. Since we know where w_1 is, we can compute the positions of q_1 and q_2. With this knowledge, the arrival time at point q_1, t_1 and at point q_2, t_2 can be computed using the aircraft's current velocity estimate. Therefore, the magnitudes of the semi-major (along-track) axis ($a\,[nm]$) and the semi-minor (cross-track) axis ($b\,[nm]$) of the trajectory prediction error ellipse at conflict time t_c (seconds) are:

1. Case I: $0 \leq t_c \leq t_1$

$$\begin{aligned} a_1(t_c) &= 0.042 t_c + a_0 \\ b_1(t_c) &= \min[\tfrac{v t_c}{3420} + b_0, 1] \end{aligned} \tag{4.5}$$

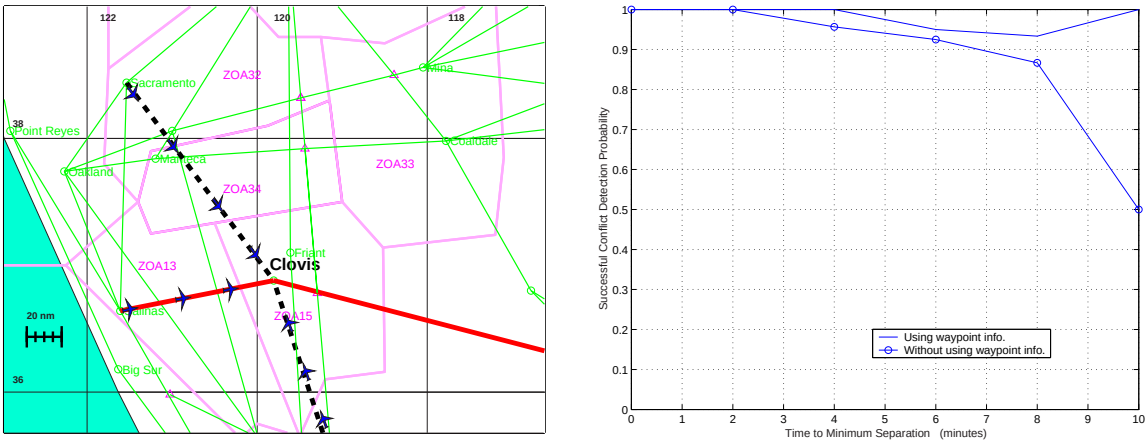
2. Case II: $t_1 < t_c \leq t_2$

$$\begin{aligned} a_2(t_c) &= 0.042(t_c - t_1) + a_1(t_1) \\ b_2(t_c) &= 0.042(t_c - t_1) + b_1(t_1) \end{aligned} \tag{4.6}$$

3. Case III: $t_2 < t_c \leq T_{LA}$

$$\begin{aligned} a_3(t_c) &= 0.042(t_c - t_2) + a_2(t_2) \\ b_3(t_c) &= \max[b_2(t_2), 1] \end{aligned} \tag{4.7}$$

where a_0 and b_0 are initial uncertainties ($15m$). With these trajectory prediction error ellipses, conflict probabilities are computed in the same way as in Section 4.1.2. This conflict detection algorithm can also be easily extended to the case in which an aircraft trajectory has multiple waypoints.



(a) (b)

Figure 4.10: Performance comparison between the conflict detection algorithms using and without using waypoint information. (a) Conflict scenario around waypoint *Clovis*. (b) Successful conflict detection probabilities of the conflict detection algorithms using and without using waypoint information.

Figure 4.10-(a) shows three trajectories around *Clovis*, and the aircraft have *Clovis* as the common waypoint. Flight plans passing *Clovis* are extracted from ETMS data and are used for the simulation in Figure 4.10. The dotted lines and solid lines represent different flight plans. If the heading change of each aircraft at *Clovis* is not considered, the conflict among the three aircraft along the dotted line cannot be detected before the heading changes. However, the conflict detection algorithm using waypoint information can detect the conflict before the heading changes at *Clovis*. Figure 4.10-(b) shows the comparison of conflict detection results using and not using waypoint information. As the time to the minimum separation increases, the successful conflict detection probability of the conflict detection algorithm without using waypoint information drops rapidly, yet that of the conflict detection algorithm using waypoint information remains close to one. Thus,

this simulation shows that using waypoint information could improve the performance of the conflict detection algorithm, especially around waypoints.

Chapter 5

Protocol-based Conflict Resolution

In the previous chapter, we developed a probabilistic conflict detection algorithm, which uses the current continuous state estimates of an aircraft as well as its flight mode estimates computed by the Residual-Mean Interacting Multiple Model (RMIMM) algorithm. In this chapter, we consider multiple-aircraft conflict resolution problems and propose a protocol-based conflict resolution algorithm for resolving multiple-aircraft conflicts.

Several recent papers have focused on solving conflict resolution problems [11]. The studies of conflict resolution, which is the scope of this chapter, may be categorized into three different cases according to the methods by which a solution is obtained : 1) optimization 2) rule-based and 3) force field methods.

First, optimized conflict resolution algorithms produce a resolution maneuver, which minimizes a cost function such as deviation from the original trajectory, flight time, fuel consumption, or energy [90, 91, 92, 93, 94, 95, 96]. Menon *et al.* [12] formulate conflict resolution as a multi-participant optimal control problem. Using parameter optimization and state-constrained trajectory optimization, they compute a conflict resolution trajectory for two different cost functions: deviation from the original trajectory and a linear combination of total flight time and fuel consumption. Their method results in 3D optimal multiple-aircraft conflict resolution. In general, the optimization process is computationally intensive and is difficult to implement in real-time. Frazzoli *et al.* [97] use randomized searches to choose a particular cross pattern from all possibilities, and apply convex optimization to obtain a

minimum energy resolution maneuver for this cross pattern. Hu *et al.* [98] categorize types of multiple-aircraft resolutions using braid theory, and select a type by a randomized algorithm. Resolution maneuvers are determined by solving a convex optimization problem to minimize the total length of trajectories (or energy) for the selected type. Due to the properties of convex optimization, the algorithms are numerically efficient. However, randomized algorithms could produce different solutions to the same conflict problems. Bilimoria [99] proposes a method for pairwise geometric optimization which minimizes trajectory deviation. Conflict resolution maneuvers are obtained from a closed-form analytic solution, and thus can be applied in real-time to two-aircraft conflicts. Mao *et al.* [100] propose a decentralized conflict avoidance strategy based on successive pairwise resolution and numerical optimization. Durand *et al.* [101] propose that predefined maneuvers be used to construct a multiple-aircraft conflict resolution maneuver. Their method uses genetic algorithms to generate each resolution maneuver successively. Alliot *et al.* [26] solve multiple-aircraft conflicts by generating resolution maneuvers for each aircraft sequentially with a token allocation strategy; the A^* algorithm is then used to select the optimal maneuver (minimum length trajectory) from a set of predefined maneuvers. In order to resolve multiple-aircraft conflicts, pairwise resolution algorithms should be executed successively, but it is fairly easy to come up with situations in which the successive application of pairwise resolution does not guarantee safety for multiple-aircraft conflicts [26].

The second class of conflict resolution problems may be referred to as rule-based conflict resolution, i.e., a conflict avoidance maneuver is determined according to predescribed rules [102, 103, 104, 105]. Duong *et al.*[105] propose Extended Flight Rules, which extend the current Visual Flight Rules by assigning a priority to each aircraft involved in a conflict. The rules may be simple to understand and easily implementable, but they do not properly account for unexpected events. For example, a conflict situation may be classified as several predefined cases at the same time. It results from uncertainties in the aircraft's position and heading. The ambiguity in which rule to choose may lead to an unsafe resolution. The method may also require many rules to completely cover all possible conflicts.

Another class of conflict resolution techniques uses force field methods. Aircraft are assumed to fly in the force field generated by a potential function; the forces induced by the potential function form a resolution maneuver [106, 107, 108, 109, 110, 111]. Duong and Zeghal [110] present a technique in which a force field "generated" by an intruding aircraft produces a

conflict avoidance action and a force from the flight plan generates an attracting action. Zeghal and Ferber [111] propose a distributed conflict resolution algorithm based on a symmetrical force field method. They give a resolution maneuver using a relatively simple equation, yet the resolution maneuver may have several discontinuities which are not flyable. Most importantly, safety cannot be proven about such multiple-aircraft maneuvers.

We present a new conflict resolution algorithm, which we call Protocol-Based Conflict Resolution (PBCR), for multiple-aircraft conflicts. The PBCR algorithm is simple and easily understandable since the protocol is obtained from a closed-form analytic solution. Therefore, it can be implemented in airborne systems for real-time conflict resolution, as well as in ATC ground systems. Most importantly, the proposed algorithm always guarantees safe conflict resolution within the limits of the model used. This is the main difference from many other currently available multiple aircraft resolution solutions.

Since aircraft fly in a vertically stratified airspace in the current Air Traffic Control (ATC), aircraft are assumed to cruise at the same altitude with varying velocities. The position, velocity, and heading of an aircraft are assumed to be available to all aircraft which are involved in the conflict; this assumption can be justified with the proposed availability of the Global Positioning System (GPS) [9] and Automatic Dependent Surveillance-Broadcast (ADS-B) [25].

For the derivation of the protocol, the multiple-aircraft conflict is categorized into two cases: 1) *exact* and 2) *inexact conflict*. First, we derive a closed form analytic solution describing the resolution maneuver for the exact conflict case, which represents the situation in which all aircraft would come into a conflict at a single point in time and space. This result is then generalized to cover the inexact conflict case, in which conflict points of multiple aircraft do not coincide in time and space. A finite partition of the airspace around the conflict is constructed in real time according to the minimum relative angle between aircraft. With the results from the exact and inexact conflict cases, the protocol for resolving the worst-case conflict within each partition is derived. Heading change from the original path is the primary control input and is used exclusively when the conflict is exact. Velocity change is also used as a control input when an exact conflict assumption fails.

For implementing PBCR, we have to determine which aircraft involved in the conflict should follow the protocol to resolve a conflict. We could assign all aircraft within a certain region

around an anticipated conflict point to be part of a resolution group, however, this may be not efficient since it could cause large deviation from the aircraft's nominal path. For a more efficient set of maneuvers, a flight-mode-based conflict prevention algorithm is presented. It uses the conflict detection algorithm developed in the previous chapter. The conflict prevention algorithm runs in real time, and detects all conflicts when an aircraft changes its heading and/or speed, that is, changes its flight-mode. By applying the conflict prevention algorithm, we find the smallest group of aircraft which will not cause another conflict.

PBCR also takes into account aircraft flight plans, by considering both a resolution and return path.

Computational complexity of the conflict prevention algorithm is polynomial in the number of aircraft, and thus the PBCR with the conflict prevention algorithm can be implemented with real time. Finally, we combine the Flight-Mode-Based Conflict Detection (FMBCD) algorithm developed in the previous chapter and the Protocol-Based Conflict Resolution algorithm and validate it with actual air traffic data.

5.1 Motivation and problem formulation

Multiple-aircraft conflict resolution is motivated by the fact that pairwise conflict resolution is not guaranteed to resolve multiple-aircraft conflicts. Several recent papers have focused on the multiple-aircraft conflict resolution case. However, their approaches are based on optimization procedures and randomized algorithms [97, 98, 12], and thus could produce different solutions to identical problems, which may cause confusion in real applications. Since optimization procedures are computationally intensive, they are undesirable for real-time airborne applications. Thus, we propose a protocol-based conflict resolution for multiple-aircraft conflicts to overcome the above problems. While the majority of conflicts occurring in the current airspace are pairwise conflicts, multiple-aircraft conflict resolution methods are important for two reasons. First, a method that evolves gracefully from treatment of two aircraft to treatment of multiple aircraft conflicts would be efficient in resolving even today's small number of multiple aircraft problems; second, as the airspace and air traffic system evolve to a stage in which aircraft are more often flying user preferred routes, one would expect more multiple aircraft conflicts.

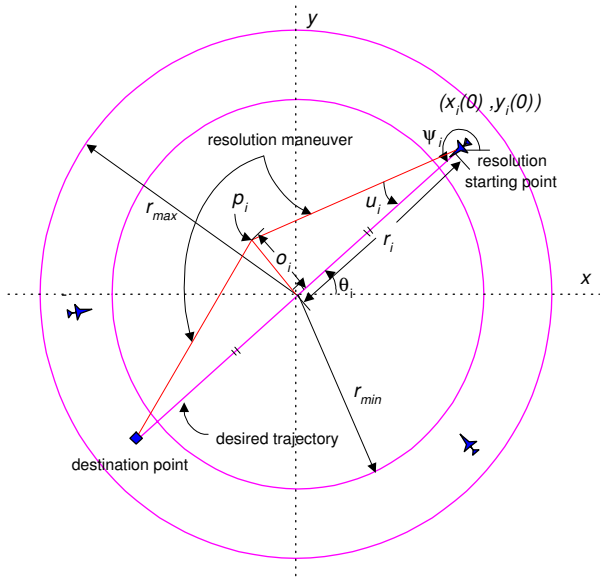


Figure 5.1: An illustration of the exact conflict resolution maneuver for one of the N-aircraft and parameters involved.

We consider the problem of conflict resolution in the horizontal plane, using only information about each aircraft's current position, velocity, and heading. A *time horizon* of $[0, T_f]$ is assumed, where 0 is an initial time at which all aircraft involved in a conflict initiate their conflict resolution maneuver and T_f is a maneuver completion time for all aircraft. We define a *conflict* to be the event in which, the distance between any pair of aircraft, $d(t)$, is less than a predefined safety distance, R, at some $t \in [0, T_f] \Leftrightarrow \exists t \in [0, T_f]$ such that $d(t) < R$ (R is assumed to be 5 nautical miles (nm) in this paper). Then, *safety* means there are no conflicts over $[0, T_f]$, i.e., the minimum distance between any pair of aircraft is greater than or equal to R. Each aircraft's velocity is bounded by a minimum allowable velocity, v_{min}, and a maximum allowable velocity, v_{max}: $v \in [v_{min}, v_{max}]$.

We consider the following two cases:

Definition 6 (Exact conflict). *All aircraft involved in a conflict come into conflict at a single point in space and time.*

Definition 7 (Inexact conflict). *At least one of the aircraft involved comes into conflict at a different point in space and time than the others.*

In this section, we present the model and formulate the collision avoidance problem for the *exact conflict case*; the more general *inexact conflict case* will be detailed in Section 5.3.2, using analysis and results from this and following sections.

Initially, we assume that each aircraft has a desired, or nominal, trajectory, which is a straight path of constant heading. Conflict detection is achieved by the FMBCD algorithm with a Look-ahead time, which we assume in this chapter to be greater than 20 minutes, though our method is general enough to work for any finite time horizon. When a conflict is detected, all aircraft involved in the conflict prepare to initiate a conflict resolution maneuver. The maneuver is assumed to start at ($\textit{conflict-time} - T$), where *conflict-time* is the time at which the aircraft would have reached the conflict point, and T is a fixed time, assumed to be 20 minutes. In our notation, we thus initiate the time horizon $[0, T_f]$ so that ($\textit{conflict-time} - T$) $= 0$. The aircraft involved in the conflict are assumed to be flying at constant velocity during a resolution maneuver. The velocities are bounded between known values v_{min} and v_{max}.

We initially partition the airspace around the conflict point into two concentric circular discs of radii r_{min} and r_{max}, as shown in Figure 5.1. These radii are designed so that $r_{min} = v_{min}T$ and $r_{max} = v_{max}T$, which ensures that the aircraft lie in the annulus between the two radii at the initiation of the conflict resolution maneuver.

For aircraft i, we denote the starting position as $(x_i(0), y_i(0))$ such that

$$\begin{bmatrix} x_i(0) \\ y_i(0) \end{bmatrix} = \begin{bmatrix} r_i \cos\theta_i \\ r_i \sin\theta_i \end{bmatrix}, \quad \text{where } i = 1, 2, \cdots, N \tag{5.1}$$

where $r_i = v_i T$ is the radial position of aircraft i, θ_i is the angular position of aircraft i, and v_i is the velocity of aircraft i as shown in Figure 5.1.

For aircraft i, the destination point is computed as the point at which the aircraft must "rejoin" its original trajectory after completing the conflict resolution maneuver. This point is assumed to be an equal distance along the desired trajectory from the conflict point, as the conflict point is from $(x_i(0), y_i(0))$. We then relax this assumption and use waypoint

information to find the destination point. We use a kinematic model for each aircraft to design the protocol:

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} = \begin{bmatrix} v_i \cos\psi_i \\ v_i \sin\psi_i \end{bmatrix}, \quad \text{where } i = 1, 2, \cdots, N \tag{5.2}$$

In order to derive a real-time solution, we assume in our model that the aircraft can change heading instantaneously, and that the control input is a piecewise constant heading change ($\Delta\psi_i$) which we denote as u_i. However, we show (in Appendix C) that the protocol is still valid when a sharp corner in the resulting (x_i, y_i) path is replaced with a smooth circular arc. Also, we assume synchronous maneuvers, in which all aircraft change their heading at the same time, yet we will show that our protocol is robust with respect to asynchronous maneuvers, in which all aircraft may not necessarily change their heading at the same time. Detailed robustness analysis is given in Appendix C. We assume u_i to be the same for all aircraft. The resolution maneuver for each aircraft is set to be an isosceles triangular path composed of two straight segments of constant heading and of constant velocity, the apex of which is the new waypoint, p_i. The position of the waypoint p_i, and the vertical deviation o_i, is determined up to sign by the heading change u_i. It can easily be shown that under this maneuver, all aircraft arrive at their waypoints p_i, and their destination points, at the same time. Thus the problem of maneuver protocol design reduces to computing the heading change u_i to ensure safety. Based on this symmetric resolution maneuver, we propose an asymmetric resolution maneuver, in which the resolution trajectory and the returning trajectory are considered separately and thus may not be symmetric, in order to account for flight plan information.

Finally, in this design, we are assuming that all aircraft share information about the other aircraft's position, velocity, and heading. This information sharing can be achieved by communication through Automatic Dependent Surveillance-Broadcast (ADS-B). In [38], we analyzed the effect on the conflict resolution protocol of differing information horizons.

5.2 Safety conditions for multiple-aircraft conflict resolution

In this section, we derive the safety conditions for multiple-aircraft conflict resolution for an exact conflict which is unrealistic yet illustrative. First, a necessary and sufficient condition

for safety of multiple-aircraft conflict resolution is derived. Since the safety condition for the proposed conflict resolution is based on the minimum distance between aircraft, the geometry of the conflict resolution maneuver is symmetric, and all aircraft change their heading at the same time, the safety condition can be derived over the first half and the second half of the conflict resolution path separately. Here, the safety condition is based on the assumption of an exact collision; we show in Section 5.3.2 that the same safety condition is also used for an inexact conflict resolution.

To ensure there is no conflict during the resolution maneuver, the minimum distance between aircraft should be greater than or equal to the predefined safety distance, R. For notational simplicity, we define for aircraft i and j: $\phi_i := \psi_i + u_i$, $a := v_i \cos\phi_i - v_j \cos\phi_j$, $b := r_i \cos\theta_i - r_j \cos\theta_j$, $c := v_i \sin\phi_i - v_j \sin\phi_j$, $e := r_i \sin\theta_i - r_j \sin\theta_j$.

Then, the distance squared between aircraft i and aircraft j on the first half path is:

$$\begin{aligned} S_{ij}^1(t) &= (x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2 \\ &= [(v_i \cos\phi_i - v_j \cos\phi_j)t + (r_i \cos\theta_i - r_j \cos\theta_j)]^2 \\ &\quad +[(v_i \sin\phi_i - v_j \sin\phi_j)t + (r_i \sin\theta_i - r_j \sin\theta_j)]^2 \\ &= (at+b)^2 + (ct+e)^2 \end{aligned} \tag{5.3}$$

where $0 \le t \le \frac{T_f}{2}$, and $i, j = 1, 2, \cdots, N$, $i \neq j$.

Similarly, we define: $\chi_i := \psi_i - u_i$, $\lambda := v_i \cos\chi_i - v_j \cos\chi_j$, $\mu := r_i \sin\chi_i - r_j \sin\chi_j$, $\xi := \frac{1}{2}T_f a + b$, $\sigma := \frac{1}{2}T_f c + e$.

Then, the distance squared between aircraft i and aircraft j on the second half path is:

$$\begin{aligned} S_{ij}^2(\tau) &= [(v_i \cos\chi_i - v_j \cos\chi_j)\tau + (v_i \cos\phi_i - v_j \cos\phi_j)\tfrac{T_f}{2} \\ &\quad +(r_i \cos\theta_i - r_j \cos\theta_j)]^2 + [(v_i \sin\chi_i - v_j \sin\chi_j)\tau \\ &\quad +(v_i \sin\phi_i - v_j \sin\phi_j)\tfrac{T_f}{2} + (r_i \sin\theta_i - r_j \sin\theta_j)]^2 \\ &= (\lambda\tau + \xi)^2 + (\mu\tau + \sigma)^2 \end{aligned} \tag{5.4}$$

where $0 < \tau \le \frac{T_f}{2}$, and $i, j = 1, 2, \cdots, N$, $i \neq j$.

In our problem setting, the worst case occurs when the minimum distance among aircraft is achieved. If the minimum distance between aircraft is always greater than or equal to the predefined safety distance, then the whole resolution maneuver is safe.

We define $(S_{ij}^k)_{min}(t)$ to be the minimum distance between aircraft i and aircraft j over the first half path if $k = 1$ and over the second half path if $k = 2$. In order to make multiple-aircraft conflict resolution safe, the following inequality is satisfied:

$$(S_{ij}^k)_{min}(t) \geq R^2 \text{ for } t \in \begin{cases} [0, \frac{T_f}{2}] & \text{if } k = 1 \\ (\frac{T_f}{2}, T_f] & \text{if } k = 2 \end{cases} \tag{5.5}$$

where $ij \in \{12, 23, 13, \cdots, (N-1)N\}$, which leads to $N(N-1)$ safety conditions. For example, if we consider a three-aircraft conflict case ($N = 3$), then there are six conditions for safe resolution. Condition 1 (C_1) is $(S_{12}^1)_{min}(t) \geq R^2$ and Condition 2 (C_2) is $(S_{12}^2)_{min}(t) \geq R^2$. Condition 3 ($C_3$) and Condition 4 ($C_4$) for aircraft 2 and aircraft 3, and Condition 5 (C_5) and Condition 6 (C_6) for aircraft 1 and aircraft 3 can be obtained in the same way. We claim that the inequality in (5.5) for $ij \in \{12, 23, 13, \cdots, (N-1)N\}$ comprise necessary and sufficient conditions for safety of multiple-aircraft conflict resolution for an exact conflict.

Lemma 8. *(Safety conditions for multiple-aircraft conflict resolution) There are no conflicts over $t \in [0, T_f]$ for an exact conflict case if and only if $(C_1 \wedge C_2) \wedge (C_3 \wedge C_4) \wedge \cdots \wedge (C_{N(N-1)-1} \wedge C_{N(N-1)})$.*

Proof. See Appendix A. □

We compute the minimum distance over the first half path and over the second half path sequentially. From the geometry of the conflict resolution maneuver, the distance between aircraft along each straight path is a differentiable function of t, thus the derivative of the distance between aircraft with respect to time is always well defined and finite in these regions. The minimum distance can be obtained by setting the derivative of the distance squared between aircraft with respect to time to zero:

$$\frac{dS_{ij}^1(t)}{dt} = 0 \Rightarrow t_{min} = -\frac{ab + ce}{a^2 + c^2} \tag{5.6}$$

After algebraic manipulation, it may be shown that (5.6) reduces to a very simple relation between heading change and the time at which the minimum distance occurs: $t_{min} =$

$T\cos u_i$. Substituting t_{min} into (5.3), the safety condition between aircraft i and aircraft j over the first half path becomes

$$S_{ij}^1(t_{min}) = (a^2+c^2)(b^2+e^2-R^2)-(ab+ce)^2 \geq 0 \tag{5.7}$$

(5.7) can be rewritten as:

$$b^2+e^2-\frac{(ab+ce)^2}{a^2+c^2} \geq R^2 \tag{5.8}$$

where

$$\begin{aligned}
b^2+e^2 &= (r_i\cos\theta_i - r_j\cos\theta_j)^2 + (r_i\sin\theta_i - r_j\sin\theta_j)^2 \\
&= r_i^2+r_j^2-2r_ir_j\cos(\theta_i-\theta_j) \\
a^2+c^2 &= (v_i\cos\phi_i - v_j\cos\phi_j)^2 + (v_i\sin\phi_i - v_j\sin\phi_j)^2 \\
&= \frac{1}{T^2}[r_i^2+r_j^2-2r_ir_j\cos(\theta_i-\theta_j)] \\
ab+ce &= (v_i\cos\phi_i - v_j\cos\phi_j)(r_i\cos\theta_i - r_j\cos\theta_j) + (v_i\sin\phi_i - v_j\sin\phi_j)(r_i\sin\theta_i - r_j\sin\theta_j) \\
&= -\frac{\cos u}{T}[r_i^2+r_j^2-2r_ir_j\cos(\theta_i-\theta_j)]
\end{aligned}$$

Since the heading changes of all aircraft are assumed to be the same $u := u_i = u_j$ for all $i,j = 1,2,\cdots,N$, the safety condition in (5.8) can be simplified as

$$\sin^2 u \geq \frac{R^2}{r_i^2+r_j^2-2r_ir_j\cos(\theta_i-\theta_j)} = \frac{R^2}{S_{ij}^1(0)} \tag{5.9}$$

where $i,j = 1,2,\cdots,N$, $i \neq j$.

Now, we consider the second half path. Similar to the first half path, the minimum distance occurs at

$$\frac{dS_{ij}^2(t)}{dt} = 0 \tag{5.10}$$

The same procedure as previous is taken, and the safety condition between aircraft i and aircraft j over the second half path is identical to the condition in (5.9) for the first half path. This is expected from the geometry of the resolution maneuvers into which the conflict resolution problem is cast because heading changes of all aircraft are the same and aircraft change their headings at the same time.

From (5.9), we see that the magnitude of the heading change required for conflict resolution is inversely proportional to the aircraft's initial distance to a conflict point and to the

minimum relative angular distance among aircraft. In other words, the magnitude of the safe heading change is inversely proportional to the distance squared between aircraft at the resolution starting time. This result coincides with the intuition that if aircraft are far apart from each other, then a small heading change is enough to resolve a conflict but if aircraft are close, then a bigger heading change is needed for safe separation. Inequality (5.9) represents a *closed-form analytic solution* for the heading change required of all aircraft to resolve the conflict. The minimum heading change, u_{min}, can be obtained from (5.9) for a given initial configuration and known parameters. Since both $\pm u$ satisfy (5.9), we resolve this turn ambiguity for exact conflicts by restricting the control input to be such that $0^o < u < 90^o$. Thus, inequality (5.9) (with turn ambiguity resolved) is simple and deterministic; and can be used to design a protocol for multiple-aircraft conflict resolution which can be implemented in both airborne systems and ground ATC systems, and also easily implemented for real-time applications because its computing time is negligible. No other currently available multiple-aircraft conflict resolution algorithm has all of these properties. The design of such a protocol for exact and inexact conflicts is the subject of the next two sections.

5.3 Protocols for multiple-aircraft conflict resolution

In this section, we derive protocols for multiple-aircraft conflict resolution using the safety condition of inequality (5.9). Based on the unrealistic yet geometrically simple exact conflict case, we derive a protocol for the general inexact conflict case.

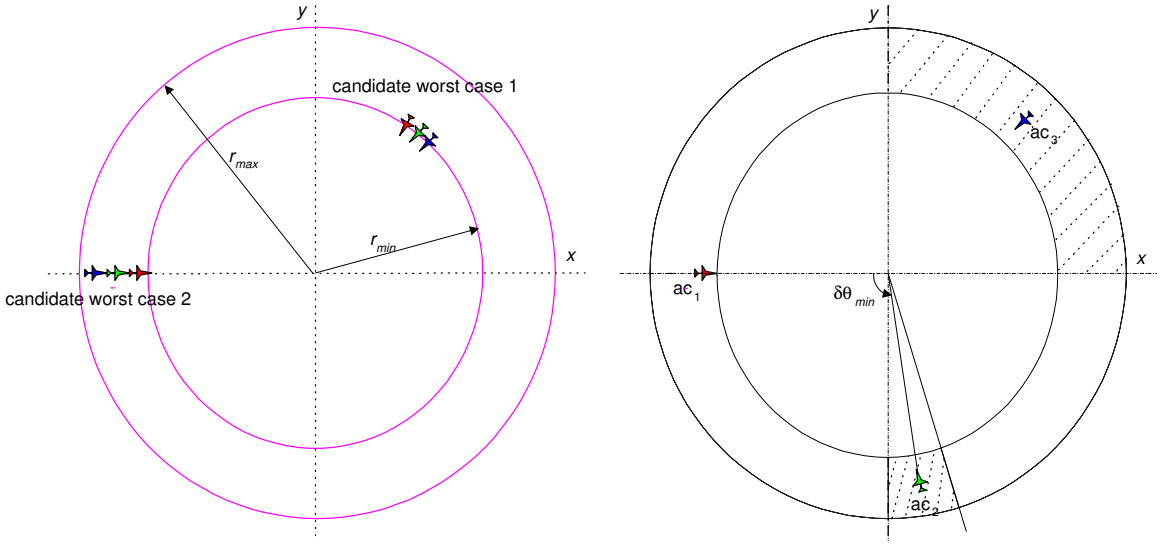
5.3.1 Derivation of a protocol for multiple-aircraft conflict resolution for an exact conflict

We construct a finite partition of the airspace around the conflict, and using our analytic solution in (5.9), we derive a protocol for resolving the worst-case conflict within each partition, where we define the worst-case conflict to be that configuration in which the required heading change to resolve the conflict is maximized. Thus, for each partition, the worst-case conflict is computed and the safe heading change which causes the minimum deviation from the desired trajectories of each aircraft is computed for this worst case. From (5.9), the minimum heading change which resolves a conflict is achieved when the

denominator in the right-hand side is minimum, i.e. when $r_i = r_j = r_{min}$ and $|\theta_i - \theta_j| = \delta\theta_{min}$:

$$u_{min} = \sin^{-1}\left(\frac{R}{r_{min}\sqrt{2(1-\cos\delta\theta_{min})}}\right) \tag{5.11}$$

In this chapter, we use parameters from the data of a B747 which cruises with Mach 0.8 at $40,000ft$ altitude [112]. We set $r_{min} = 130nm$ and $r_{max} = 170nm$ based on the data of a B747, though these are general and may be altered. All initial configurations are assumed to be safe.



(a) **Two possible worst cases.**

(b) **Possible conflict configuration in Case 1 :** $90^o \leq \delta\theta_{min} \leq 120^o$**.**

Figure 5.2: Airspace partition.

For clarity, we derive the protocol below for a three-aircraft conflict, however, as we shall show, our method is general to N-aircraft conflicts. For a three-aircraft conflict, there are two possible worst cases as shown in Figure 5.2-(a). One is the case in which all three aircraft are initially at r_{min} and separated by R. The other is the case in which all three aircraft are initially located on a line and separated by R. However, these two cases become unsafe instantaneously, i.e., this is a tactical conflict resolution problem which cannot be resolved by our (or any) algorithm. Therefore, we impose a condition that the initial minimum distance between aircraft is not less than $3R$ so that we rule out such cases. In fact, the initial minimum distance between aircraft can be any value greater than R, the reason we

choose $3R$ is to avoid large heading changes ($\geq 20^o$) while at the same time making the restriction not overly conservative. Then, the worst case occurs when the initial distance between any two aircraft is $3R$. Therefore, a protocol for the worst case is:

$$u \geq 0.3400 = 19.4798^o \tag{5.12}$$

This is an interesting result: it indicates that a triangular deviation maneuver as shown in Figure 5.1, with $u \geq 19.5^o$ for all aircraft, will resolve any conflict for any number of aircraft, as long as at the start of the maneuver, the radial positions of all aircraft are between $130nm$ and $170nm$, and the initial minimum distance between aircraft is not less than $3R$. However, this result is overly conservative as it was computed using the worst case data. Thus, consider the following partition: divide the airspace in the angular direction around the conflict point in order to alleviate conservativeness. Consider, for example, six different cases corresponding to the minimum relative angle between any pair of aircraft:

- Case 1: $90^o \leq \delta\theta_{min} \leq 120^o$
- Case 2: $60^o \leq \delta\theta_{min} < 90^o$
- Case 3: $45^o \leq \delta\theta_{min} < 60^o$
- Case 4: $30^o \leq \delta\theta_{min} < 45^o$
- Case 5: $10^o \leq \delta\theta_{min} < 30^o$
- Case 6: $0^o \leq \delta\theta_{min} < 10^o$

where the minimum relative angle between aircraft is defined: $\delta\theta_{min} = \min\{|\theta_i - \theta_j||i, j = 1, 2, \cdots, N, \text{and}\, i \neq j\}$ as shown in Figure 5.2-(b) for a three-aircraft conflict. For each of these cases, we compute the worst-case conflict and the safe heading change which causes the minimum deviation from the desired trajectories of each aircraft.

Consider, for example, Case 1, in which without loss of generality, the relative angle between aircraft 1 and aircraft 2 is assumed to be minimum. Place aircraft 1 on the negative x-axis as shown in Figure 5.2-(b), such that the conflict occurs at the origin. To satisfy the hypothesis of Case 1, aircraft 2 should be located inside the shaded region, $270^o \leq \theta_2 \leq 300^o$ (since $\delta\theta_{min}$ is 90^o) and aircraft 3 could be located somewhere inside the shaded region, $0^o \leq \theta_3 \leq 90^o$, as shown in Figure 5.2-(b). Here, the shaded region which contains aircraft 3 is the largest possible one. Indeed, if aircraft 2 lies on the negative y-axis, then aircraft 3 could lie on the positive x-axis, and the hypothesis of Case 1 would still be satisfied.

The minimum safe heading change can be obtained by substituting $r_{min} = 130nm$ and $\delta\theta_{min} = 90^o$ into equation (5.11). This procedure can be similarly applied to the other five cases (where the protocol for Case 6 is the same as equation (5.12) derived earlier), leading to the following protocol:

Multiple-aircraft protocol in the case of exact conflict

- Case 1: $90^o \leq \delta\theta_{min} \leq 120^o \Rightarrow u \geq 0.0272 = 1.5584^o$
- Case 2: $60^o \leq \delta\theta_{min} < 90^o \Rightarrow u \geq 0.0385 = 2.2048^o$
- Case 3: $45^o \leq \delta\theta_{min} < 60^o \Rightarrow u \geq 0.0503 = 2.8805^o$
- Case 4: $30^o \leq \delta\theta_{min} < 45^o \Rightarrow u \geq 0.0744 = 4.2628^o$
- Case 5: $10^o \leq \delta\theta_{min} < 30^o \Rightarrow u \geq 0.2225 = 12.7471^o$
- Case 6: $0^o \leq \delta\theta_{min} < 10^o \Rightarrow u \geq 0.3400 = 19.4798^o$

If we consider an N-aircraft conflict in which the minimum relative angular position among N aircraft is, for example, 50^o, then heading change $u = 2.8805^o$ guarantees safety of the N-aircraft conflict resolution. In general, we consider the case in which each aircraft has the authority to choose its own airspace partition. Since the finer the airspace partition, the smaller the deviation, it is reasonable that each aircraft may select its own airspace partition according to its on-board computing ability and aerodynamic performance. Then, the resolution procedure is as follows:

Algorithm 2. (Conflict resolution procedure in the case of exact conflict)
We assume that each aircraft has access to the other aircraft's position, velocity, and heading, and thus detects a conflict:

1. *Each aircraft chooses a partition, computes its own heading change according to this partition, and broadcasts this heading change u_i to all other aircraft;*
2. *Among all heading changes broadcast including its own each aircraft chooses the smallest (i.e. $u = \min_i |u_i|$, for i= 1, 2, $\cdots$,N.), and follows this protocol.* ∎

Since each aircraft has the authority to choose its own airspace partition and to compute its own resolution maneuver from **Algorithm 2**, our method is a *decentralized* conflict resolution method.

Proposition 7. ***Algorithm 2*** *guarantees safety for a multiple-aircraft exact conflict.*

Proof. Since each aircraft's heading change u_i satisfies (5.9), from **Lemma 8**, Step 1 of **Algorithm 2** is safe. Since all heading changes u_i chosen in Step 2 of **Algorithm 2** are safe, the smallest u guarantees safety. **Algorithm 2** guarantees safety of N-aircraft conflicts because **Lemma 8** and the safety condition in (5.9) do not depend on the number of aircraft involved in the conflict. □

5.3.1.1 Simulation results for an exact conflict

Simulation and validation are performed with a dynamic aircraft model under linear control using the data from Bryson [112] of a B747 which cruises with Mach 0.8 at $40,000ft$ altitude, presented in Appendix B. If the resolution procedure designed above is applied to an arbitrary initial configuration (which has a conflict point at the origin), the conflict-free resolution maneuvers are obtained as shown in Figure 5.3 for a three-aircraft conflict (for all six cases in the example above) and Figure 5.4 for a ten-aircraft conflict. The ten-aircraft scenario converging to a conflict point is motivated from Bilimoria [99], Frazzoli *et al.* [97], and Hoekstra *et al* [113]. Since the computing time for **Algorithm 2** is negligible because the protocol is based on a simple geometric inequality in (5.9), it can be used for real-time airborne applications.

5.3.2 Derivation of a protocol for multiple-aircraft conflict resolution for an inexact conflict

The exact conflict case is unrealistic, yet the methodology of the previous section may be extended to the inexact case. In this section, we relax the exact assumption to consider the case in which the conflict points of a multiple aircraft conflict do not coincide. For N-aircraft conflicts, there are at most $\frac{N(N-1)}{2}$ original conflict points. We generalize our protocol so that only the first segment of a resolution maneuver contributes to safe separation (the case in which $k = 1$ in (5.5)). The second segment of the resolution maneuver is used in order to return to the nominal trajectory. For example, if we consider the three-aircraft conflict, Condition 1 (C_1) is $(S^1_{12})_{min}(t) \geq R^2$, Condition 2 ($C_2$) is $(S^1_{23})_{min}(t) \geq R^2$ and Condition 3 (C_3) is $(S^1_{13})_{min}(t) \geq R^2$ where $t \in [0, \frac{T_f}{2}]$.

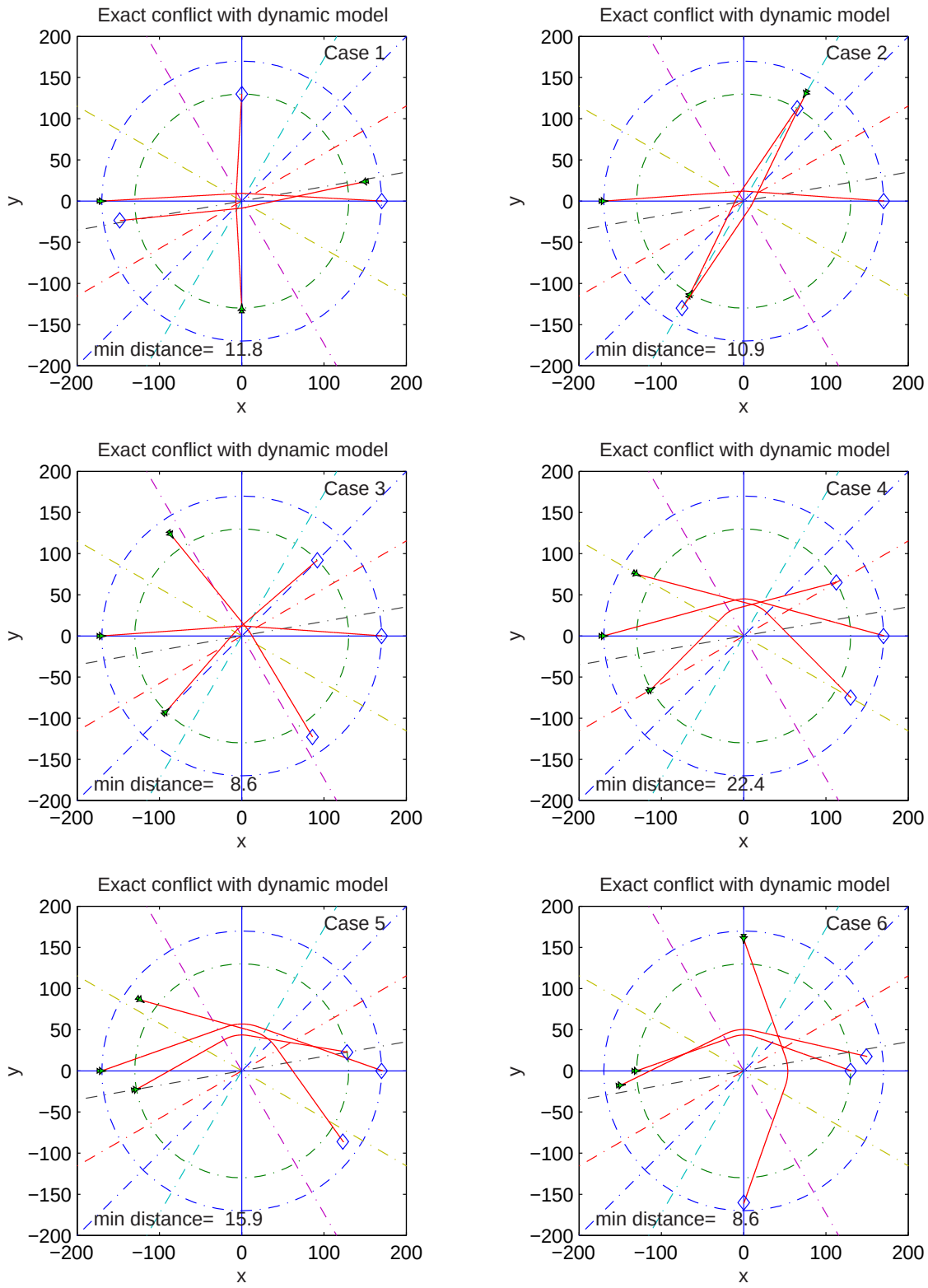


Figure 5.3: Protocol-based conflict resolution for three aircraft (nm), exact conflict case.

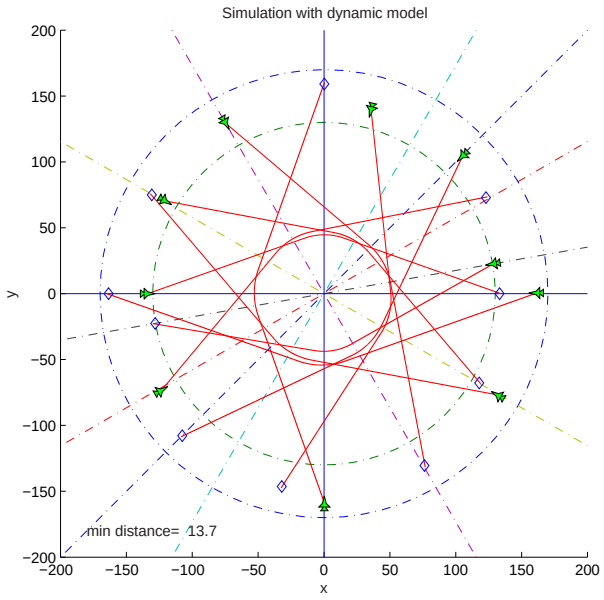


Figure 5.4: Protocol-based conflict resolution for ten aircraft (nm), exact conflict case.

If the point at which the last conflict among all conflicts occurs is chosen as the origin of our reference frame, and thus the midpoint of each aircraft's nominal trajectory segment for resolution purposes, then the other possible conflict points are on the first segments of each aircraft's nominal trajectory. In this case, only the first segment has to be considered in order to derive a safety condition. The three-aircraft example is illustrated in Figure 5.5.

Our strategy for solving this general conflict will be to first transform the problem to that of an exact conflict, and then to use the solution derived in Section 5.3.1 for an exact conflict to resolve the transformed problem. We then map the resolution maneuver back to the coordinates of the general conflict, and the general resolution results. The proof of safety follows from the fact that the exact resolution solves the transformed problem, and that the mapping between the original and transformed problem is unique.

Thus, we introduce new variables w_i, defined as the heading difference between the true heading of aircraft i and the origin of our reference frame, as shown in Figure 5.5 for aircraft 3. Here, we define the counter-clockwise direction as positive, thus $\text{sign}(w_3) = -1$, where the sign function is defined as:

$$\text{sign}(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases}$$

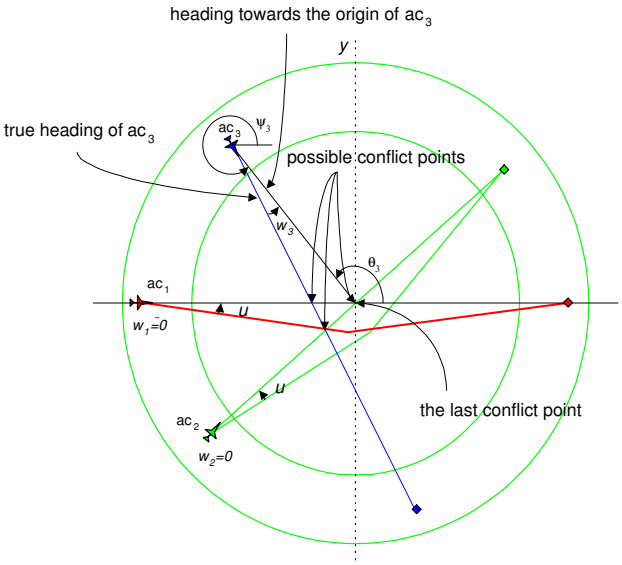


Figure 5.5: Inexact conflict.

Now suppose that aircraft i were not involved in the conflict at the origin; if we adjust the velocity of aircraft i such that $v_i = \frac{r_i}{T}$ and we change its heading by w_i, then it would be involved in a conflict at the origin. If we performed this transformation for all aircraft not involved in conflict at the origin (at most $N-2$ aircraft), then the inexact conflict would become an exact conflict, and the required heading change u for safe resolution of this exact conflict can be obtained from **Algorithm 2**. For clarity, we denote this u as u_{exact} in this section.

Now, define for aircraft i and aircraft j: $\phi_i := \psi_i - u$, $\bar{a} := v_i \cos\phi_i - v_j \cos\psi_j$, $\bar{b} := r_i \cos\theta_i - r_j \cos\theta_j$, $\bar{c} := v_i \sin\phi_i - v_j \sin\psi_j$, $\bar{e} := r_i \sin\theta_i - r_j \sin\theta_j$, where $i, j = 1, 2, \cdots, N$. Then, the distance squared between aircraft i and aircraft j is:

$$\begin{aligned} S_{ij}^1(t) &= (x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2 \\ &= [(v_i \cos\phi_i - v_j \cos\psi_j)t + (r_i \cos\theta_i - r_j \cos\theta_j)]^2 \\ &\quad + [(v_i \sin\phi_i - v_j \sin\psi_j)t + (r_i \sin\theta_i - r_j \sin\theta_j)]^2 \\ &= (\bar{a}t + \bar{b})^2 + (\bar{c}t + \bar{e})^2 \end{aligned} \tag{5.13}$$

The safety condition between aircraft i and aircraft j becomes

$$\bar{b}^2 + \bar{e}^2 - \frac{(\bar{a}\bar{b} + \bar{c}\bar{e})^2}{\bar{a}^2 + \bar{c}^2} \geq R^2 \tag{5.14}$$

We define $w_{max} = \max_i |w_i| \cdot \text{sign}(w_i)$, $i = 1, 2, \cdots, N$: thus w_{max} has the maximum magnitude among w_i and preserves sign. For example, in the case of Figure 5.5, $w_1 = 0$, $w_2 = 0$, and $w_3 = |w_3|\text{sign}(w_3)$, thus $w_{max} = -w_3$. If we assumed that $u_{exact} = w_{max}$ and follow the same procedure as in Section 5.3.1, the safety condition for a general inexact conflict would be:

$$\sin^2 u_{exact} \geq \frac{R^2}{r_1^2 + r_3^2 - 2r_1 r_3 \cos(\theta_1 - \theta_3)} \tag{5.15}$$

which is identical to the safety condition for an exact conflict in (5.9). However, the assumption that $u_{exact} = w_{max}$ may not necessarily be true. Thus, we consider the two cases $u_{exact} > |w_{max}|$ and $u_{exact} \leq |w_{max}|$: if $u_{exact} > |w_{max}|$, a safe heading change for an inexact conflict must be u_{exact} because $|w_{max}|$ would not satisfy the safety condition in (5.15); if $u_{exact} \leq |w_{max}|$, then both u_{exact} and w_{max} satisfy (5.15). However, since the sign of the heading change of all aircraft should be the same according to our protocol, then u must be set to w_{max}. Therefore, a protocol u for a general inexact conflict is:

$$u = \begin{cases} u_{exact}\,\text{sign}(w_{max}) & \text{if } u_{exact} > |w_{max}| \\ w_{max} & \text{if } u_{exact} \leq |w_{max}| \end{cases}$$

Since this protocol u is computed in the transformed frame, aircraft i must actually change its heading by $u - w_i$ $(i = 1, 2, \cdots, N)$. The protocol for a general inexact conflict is as follows:

Algorithm 3. (Multiple-aircraft protocol in the case of inexact conflict)
New Protocol *(heading change + velocity change)*
For $i = 1, 2, \cdots, N$*:*

1. *Select the last conflict point among possible conflict points as the center of conflict resolution;*
2. *If aircraft i is not involved in the conflict at the origin, adjust its velocity such that* $v_i = \frac{r_i}{T}$ *(← new velocity);*
3. *Aircraft i computes its own heading change* u_{exact} *from* ***Algorithm 1****, and computes* w_i*,* w_{max}*, and* u*;*
4. *Aircraft i changes its heading by* $u - w_i$*.* ■

Algorithm 3 may be implemented in a decentralized manner, since each aircraft recognizes a conflict situation and computes its own resolution maneuver autonomously.

Proposition 8. ***Algorithm 3*** *guarantees safety for the multiple-aircraft inexact conflict.*

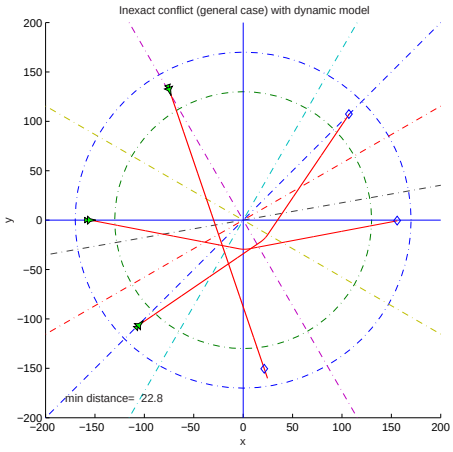
Proof. Step 1 $\sim$ Step 3 transform an inexact conflict situation into an exact conflict, and the heading changes in Steps 3 and 4 satisfy the safety condition in (5.9). Therefore, from **Lemma 8**, the multiple-aircraft protocol in the case of a general inexact conflict guarantees safety. □

5.3.2.1 Simulation results for an inexact conflict

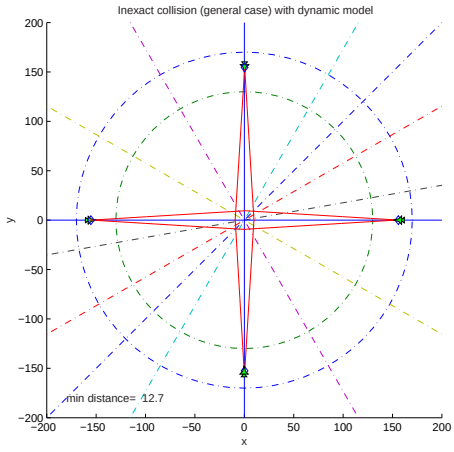
If the protocol for an inexact conflict is applied to an arbitrary initial configuration which satisfies the given assumptions, the conflict-free resolution maneuvers are obtained as shown in Figure 5.6. A symmetric encounter pattern for four-aircraft conflicts [113] and a scenario of two converging aircraft streams in Figure 5.6-(b) and (d), and random encounter patterns for ten-aircraft conflicts in Figure 5.6-(e) are motivated from Frazzoli *et al.* [97]. As in Section 5.1, r_{min} and r_{max} are the radii of two concentric circular discs around the conflict. From the data of a B747 with Mach 8 at $40,000ft$, we used $r_{min} = 130nm$ and $r_{max} = 170nm$ throughout this chapter. However, the simulation result with $r_{min} = 30nm$ and $r_{max} = 50nm$ in the airspace partition in Figure 5.6-(f) shows that our protocol is valid with respect to various airspace partitions with different r_{min} and r_{max} (correspondingly different T), which is detailed in random simulation results in Appendix B. Appendix B further presents details of validation with the dynamic aircraft model and random simulation results with different parameters such as look-ahead time, initial minimum distance between aircraft, and the number of airspace partitions.

5.4 Flight-mode-based conflict prevention

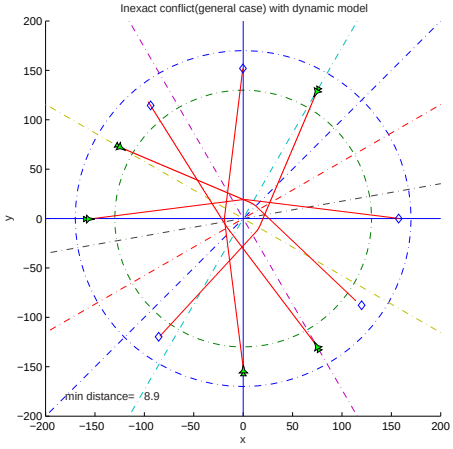
In this section, we propose a flight-mode-based conflict prevention algorithm in order to minimize the number of aircraft in a resolution group, which could result in a small deviation from the aircraft's nominal trajectory. With the use of the conflict prevention algorithm, we can also relax the assumption that an aircraft's trajectory is a straight line, so that a new protocol could resolve conflict around waypoints.



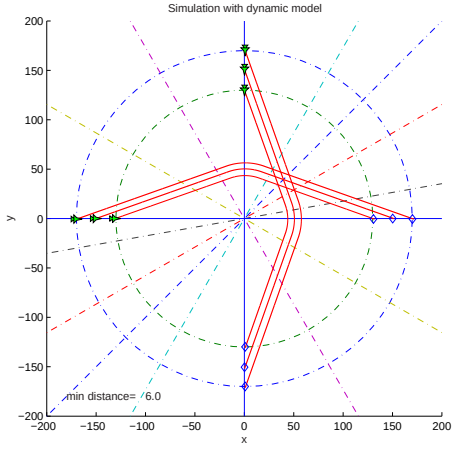
(a) **For three aircraft.**



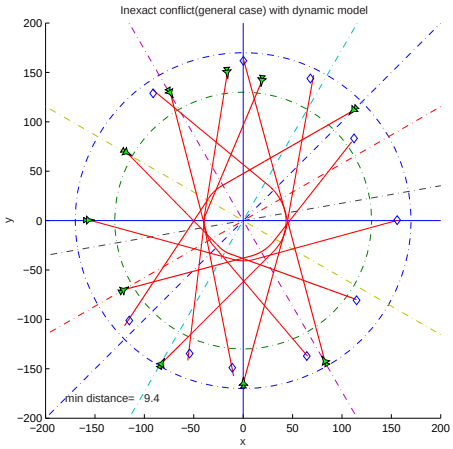
(b) **For four aircraft.**



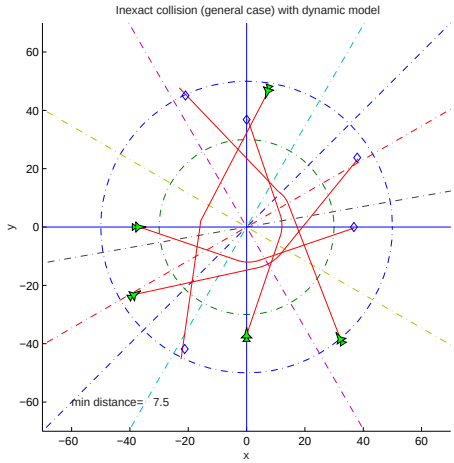
(c) **For five aircraft.**



(d) **For six aircraft.**



(e) **For ten aircraft.**



(f) **For five aircraft with** $r_{min} = 30nm$ **and** $r_{max} = 50nm$**.**

Figure 5.6: Inexact conflict (General case) (nm).

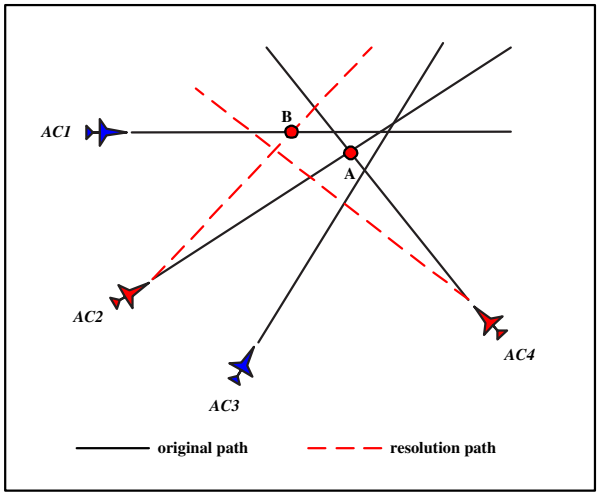


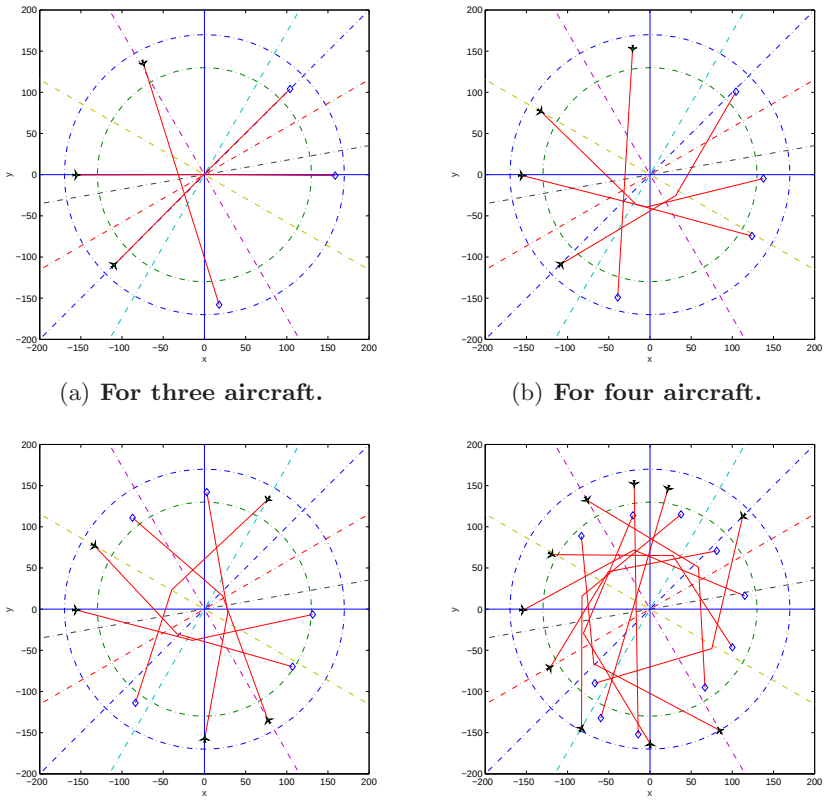
Figure 5.7: Grouping using conflict prevention algorithm.

5.4.1 Grouping

To implement PBCR, we have to determine which aircraft should take part in a resolution maneuver. One option is to include in the resolution group all aircraft within a certain range around a conflict point, yet this is inefficient since some of the aircraft may not need to take the resolution maneuver. From **Algorithm 3**, a safe heading change required for conflict resolution increases as θ_{min} decreases. If many aircraft are included in the resolution, it is more likely that θ_{min} would be small. This means that perturbation from the aircraft's original path will be large. Thus, it is desirable to keep a resolution group as small as possible. In this section, we present an algorithm to determine which aircraft should be involved in the resolution maneuver.

Figure 5.7 shows the concept of grouping using the conflict prevention algorithm. The conflict prevention algorithm is an event-driven algorithm , i.e., it is executed whenever an aircraft changes its flight mode to see if there will be any other conflicts due to the flight mode change. In Figure 5.7, solid and dotted lines represent original and resolution paths of the aircraft, respectively. Suppose that $AC2$ and $AC4$ are going to have a conflict at point A, and $AC1$ and $AC3$ have no conflict with any other aircraft. First, $AC2$ and $AC4$ are involved in a resolution group, and their maneuvers are determined according to the protocol in **Algorithm 3**. However, a resolution maneuver (dotted line) of $AC2$ causes

a new conflict with $AC1$ along its resolution trajectory, and it is detected by the conflict prevention algorithm. Using this information, the new resolution group should contain $AC1$, $AC2$ and $AC4$. After making a new resolution group, there are no more conflicts. Thus, the conflict of the four aircraft can be safely resolved by a resolution maneuver involving only three aircraft.



(a) **For three aircraft.** (b) **For four aircraft.**

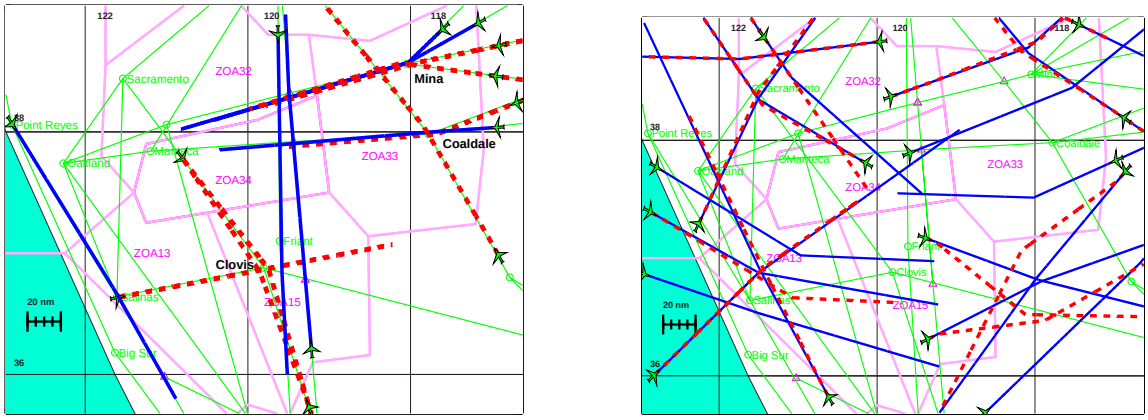
(c) **For five aircraft.** (d) **For ten aircraft.**

Figure 5.8: Protocol-based conflict resolution with flight-mode-based conflict prevention.

Figure 5.8 shows that the size of the resolution group can be reduced by applying the conflict prevention algorithm. In the three aircraft case, only two aircraft take a resolution maneuver to resolve the conflict and thus, the perturbation from the desired paths is smaller than the case in which all three aircraft take a resolution maneuver. In the four aircraft case, one aircraft is excluded from the resolution group and the other three aircraft are involved in the resolution maneuver. In the case of ten aircraft, two aircraft do not take a resolution maneuver and the resolution with only eight aircraft can avoid the conflicts.

5.5 Validation with ETMS data

In this section, we combine the PBCR algorithm with the Flight-Mode-Based Conflict Detection (FMBCD) algorithm (FMBCD/PBCR) and validate it with the scenarios based on Enhanced Traffic Management System (ETMS) data. The FMBCD algorithm is used for both conflict detection and conflict prevention. We set a look-ahead time (T_{LA}) of $20min$ for both conflict detection and conflict prevention and a conflict resolution time (T) of $10min$. The FMBCD/PBCR algorithm is applied to various air traffic scenarios based on ETMS data and flight plans from ETMS data are used to generate nominal trajectories of aircraft.



(a) (b)

Figure 5.9: Simulations with ETMS data using FMBCD/PBCR: (a) Structured airspace scenario. (b) Free flight scenario.

Figure 5.9-(a) shows one of such scenarios. Solid lines represent the nominal trajectories and dotted lines denote resolution trajectories. FMBCD detects all conflicts around waypoints such as *Mina*, *Coaldale*, and *Clovis*. The conflicts around *Mina* and *Coaldale* are resolved by only two aircraft taking a resolution maneuver, yet all three aircraft join conflict resolution to resolve a conflict near *Clovis*. Figure 5.9-(b) shows a "free flight" scenario result in which each aircraft follows its own optimal trajectory. All conflicts are detected early enough by the FMBCD algorithm for the resolution maneuvers. The PBCR algorithm resolves all conflicts safely, and it is noted that not all aircraft around conflict points join the conflict resolution due to the conflict prevention algorithm.

Part III

Safety Verification

Chapter 6

Approximate Computation of Reachable Sets using Polytopic Approximation

In the previous chapter, we considered cooperative resolution for multiple aircraft conflicts in which all aircraft involved in a conflict coordinate their maneuvers to resolve it. We have proposed algorithms which can solve these problems efficiently with guarantees of safety if the maneuvers are followed. However, safety of these algorithms may not be guaranteed if at least one of the aircraft involved in a conflict resolution does not follow the resolution maneuver. Thus, in this chapter, we consider non-cooperative conflict resolution between two aircraft, which we model as a dynamic game. In other words, we consider the worst-case conflict scenario between two aircraft in which an aircraft (evader) tries to avoid a conflict for any maneuver of the other aircraft (pursuer). By solving this problem, safety for all possible conflict cases within the limits of the model used is guaranteed. We solve this problem through reachable set computation.

Reachability analysis for continuous and hybrid systems is important for the automatic verification of safety properties and for the synthesis of safe controllers for these systems [27, 28]. Unsafe configurations of a dynamic system can be represented by a set of states (unsafe set) in the state space. By computing a reachable set, that is, a set of states from which the system could reach the unsafe set, we can show whether the dynamic system

has an unsafe configuration. The exact reachable set boundary is known to be the zero level set of the viscosity solution [114] of a Hamilton-Jacobi type of partial differential equation (PDE). To the best of our knowledge, Leitmann was the first to recognize the relationship between Bellman functions and the boundaries of the reachable sets [115]. In [116] it was shown how to approximate boundaries of reachable sets with an arbitrary accuracy using smooth functions. In the sequence of papers by Khrustalev [117, 118], locally Lipschitz functions are used to describe arbitrarily accurate under- and over-approximations of reachable sets. The numerical solutions which provide convergent approximations of reachable sets for dynamic systems have a computational complexity which is exponential in the continuous variable dimension [29, 4]. Therefore, approximate methods for reachable set computation have been proposed.

Tiwari and Khanna [119] and Alur *et al.* [120] propose predicate abstraction for reachable set computation: this method can be used to extract equivalent finite state models from complex, infinite state models, and then use them to find approximate reachable sets of the original systems. In [121] Hwang *et al.* have used an augmented form of predicate abstraction to compute reachable sets for a simple biological cell network. However, since the accuracy of reachability analysis using predicate abstraction greatly depends on the choice of polynomials for abstraction, it is important to have information about a given system *a priori* (from analysis and simulations) to get good results in the reachability analysis. Chutinan and Krogh [122, 123] present a method to approximate the flows of autonomous systems with convex polyhedra. An experimental system called **d/dt** [124, 125, 120, 126] has been developed to approximate reachable sets for linear dynamical systems using griddy orthogonal polyhedra. In these methods, optimization or an analytic relation (for linear time invariant systems) are used to push the faces of polyhedra at each time step in order to over-approximate reachable sets; it is difficult to compute the control input from the boundary of the over-approximative set, which is guaranteed to keep the system on the boundary or outside the set. Varaiya [30] designs a polytopic approximation for linear time invariant systems using optimal control. Kostousova [127] develops two-sided approximations of reachable sets for linear dynamic systems using parallelotopes. Kurzhanski and Varaiya [31, 32, 128] propose an ellipsoidal approximation for forward and backward reachable sets (a computational tool VeriSHIFT [129] has been developed based on their ideas) and in [33, 34], they define various types of reachable sets for linear time-varying systems with bounded perturbations using both open and closed-loop input laws. In

[34], they propose ellipsoidal over-approximations of reachable sets for linear systems under uncertainty via solutions of a particular type of differential equation. The authors clearly state in the conclusions of both papers that the computational methods of reachable sets and their approximations are topics for future research. These statements were a motivation for the results presented in this chapter.

In this chapter, to overcome computational complexity of convergent numerical algorithms, we use a polytopic approximation method [30] based on optimal control methods (e.g., [130, 131]), and polytopic level set functions, to efficiently compute approximations of forward and backward reachable sets for linear dynamic systems. We then extend this method to more general dynamical systems such as feedback linearizable nonlinear systems, linear dynamic games, and norm-bounded nonlinear systems. Finally, as an example, we solve a two-aircraft, non-cooperative conflict avoidance problem in real time.

6.1 Motivation and Problem Formulation

Consider a dynamical system,

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t), d(t)), \\ & \quad x(0) \in X_0 \quad (\text{or } x(t_f) \in \mathcal{Y}_0), \;\; t \in [0, t_f] \end{aligned} \tag{6.1}$$

where $0 \leq t_f < \infty$, $x \in \mathbb{R}^n$, $u \in U \subset \mathbb{R}^m$ is the control input, $d \in D \subset \mathbb{R}^p$ is the disturbance input, $X_0 = \{x : l(x) \leq 0\}$ is an initial set of states, and $\mathcal{Y}_0 = \{x : y(x) \leq 0\}$ is a target set of states. We assume f to be Lipschitz. The spaces of admissible control input trajectories and disturbance input trajectories are denoted as the spaces of piecewise continuous functions $\mathcal{U} = \{u(\cdot) \in PC^0 | u(t) \in U, 0 \leq t \leq t_f\}$ and $\mathcal{D} = \{d(\cdot) \in PC^0 | d(t) \in D, 0 \leq t \leq t_f\}$ respectively. The forward and the backward reachable sets of the system (6.1) are defined as follows.

Definition 8. *The forward reachable set $\mathcal{X}(\tau)$ at time τ $(0 < \tau \leq t_f)$, of the system (6.1) from the initial set X_0, is the set of all states $x(\tau)$ that are reachable from some $x(0) \in X_0$ with a control input $u(t) \in \mathcal{U}$ $(0 \leq t \leq \tau)$ for every disturbance $d(t) \in \mathcal{D}$ $(0 \leq t \leq \tau)$, and the system satisfying (6.1).*

Definition 9. *The backward reachable set $\mathcal{Y}(\tau)$ at time τ $(0 < \tau \leq t_f)$, of the system (6.1) from the target set $\mathcal{Y}_0$, is the set of all states $x(\tau)$ from which there exists a control input $u(t) \in \mathcal{U}$ $(\tau \leq t \leq t_f)$ such that for all disturbances $d(t) \in \mathcal{D}$ $(\tau \leq t \leq t_f)$, and the system satisfying (6.1), $x(t_f) \in \mathcal{Y}_0$.*

It has been shown that a forward reachable set computation can be formulated as a dynamic optimization problem [32]. The forward reachable set of the dynamical system (6.1) at time τ $(0 < \tau \leq t_f)$ is shown to be [32]:

$$\mathcal{X}(\tau) = \{x : v(x, \tau) \leq 0\} \tag{6.2}$$

where $v(x, \tau)$ is a (viscosity) solution of the Hamilton-Jacobi-Isaacs (HJI) partial differential equation,

$$D_t v(x,t) + \max_{u \in U} \min_{d \in D} \{< D_x v(x,t), f(x,u,d) >\} = 0 \tag{6.3}$$

with $v(x, 0) = l(x)$ and $< p, q >= p^T q$ the inner product in $\mathbb{R}^n$. Then, the forward reachable set of the dynamical system (6.1) is the zero sublevel set of the solution to the HJI equation in (6.3).

Similarly, the backward reachable set of the dynamical system (6.1) at time τ $(0 < \tau \leq t_f)$ is a zero sublevel set of the solution to the HJI equation [32],

$$D_t v(x,t) + \min_{u \in U} \max_{d \in D} \{< D_x v(x,t), f(x,u,d) >\} = 0 \tag{6.4}$$

with $v(x, t_f) = y(x)$.

In [29, 4], a numerical tool for computing convergent approximations for backwards reachable sets is designed and presented. This method is based on the level set method for computing PDE solutions [132]. The computational complexity of this tool is exponential in the number of continuous variables dimensions: it has been shown to work well in up to three continuous variables dimensions, yet for larger problems computation time is currently prohibitive. Numerical convergence has been demonstrated on several examples; we will use a "benchmark" three-dimensional example from [4] in this dissertation.

Consider planar kinematic models of two aircraft, labelled 1 and 2 as shown in Figure 6.4. Let $(x_r, y_r, \psi_r) \in \mathbb{R}^2 \times [-\pi, \pi)$ represent the relative position and orientation of aircraft

2 with respect to aircraft 1. Given the absolute positions and orientations of the two aircraft, denoted as x_i, y_i, ψ_i for $i = 1, 2$, the relative coordinates are defined as: $x_r = \cos\psi_1(x_2 - x_1) + \sin\psi_1(y_2 - y_1), y_r = -\sin\psi_1(x_2 - x_1) + \cos\psi_1(y_2 - y_1), \psi_r = \psi_2 - \psi_1$. The relative kinematics are thus given by:

$$\begin{aligned} \dot{x}_r &= -\sigma_1 + \sigma_2 \cos\psi_r + \omega_1 y_r \\ \dot{y}_r &= \sigma_2 \sin\psi_r - \omega_1 x_r \\ \dot{\psi}_r &= \omega_2 - \omega_1 \end{aligned} \tag{6.5}$$

where σ_i is the linear velocity of aircraft i and ω_i is its angular velocity. Safety is encoded

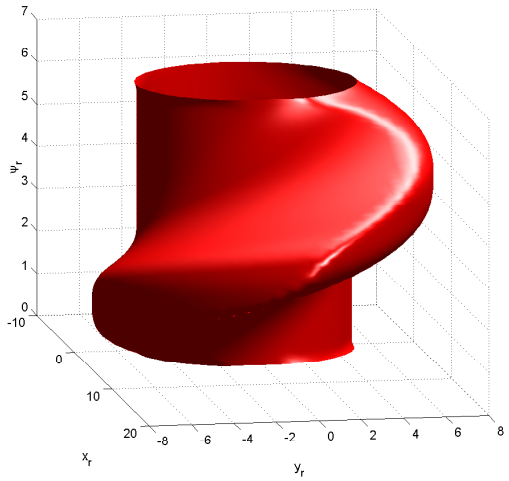


Figure 6.1: Backward reachable set for two aircraft collision avoidance problem from [4]. Courtesy of I. Mitchell.

as a 5 nautical mile radius cylinder "protected zone" centered at the origin of the relative frame. In this chapter, following the notation in **Definition 9** (which is different from that in [4]), we define the angular velocity of aircraft 2 (ω_2) as the control input that steers the system (6.5) into the target set and the angular velocity of aircraft 1 (ω_1) as the disturbance input that keeps the system (6.5) outside of the target set. Posing this problem as a game, we label aircraft 1 as "evader" and aircraft 2 as "pursuer", and we compute the set of states (x_r, y_r, ψ_r) for which for all possible disturbance inputs, ω_1 action of the evader, there is a control input, ω_2 action of the pursuer, such that the system state enters the protected zone, which we consider the target set of the game. For values $\sigma_1 = \sigma_2 = 5$ and

$\omega_i \in [-1, 1]$, the problem has been solved numerically, and the results are shown in Figure 6.1 (visualization tool courtesy of Professor Ronald Fedkiw at Stanford). This computation took approximately 15 minutes to run on a Sun UltraSparc II, in which 50 grid nodes in each dimension were used. This example may also be solved analytically [133], and it may be verified using this that the average error in computation is less than one tenth of a grid cell, with maximum error always less than one grid cell.

In the following section, we extend Varaiya's method [30] to treat this kind of system and in Section 6.3.3, we compare the above computation with the resulting approximation.

6.2 Computation of polytopic reachable sets

We first define the over-approximate reachable set [32] (here we specialize to the case of (6.1) in which there are no disturbances). Assume that $x_*(0) \in X_0$ and $u_*(t) \in \mathcal{U}$ for all $t \geq 0$ such that $x_*(\tau) \in \mathcal{X}(\tau)$ $(0 \leq t \leq \tau)$. Then, an over-approximate solution to the solution of the HJI equation in (6.3) is defined as a function $v^+(x,t)$ satisfying [32]:

$$\begin{aligned} \frac{dv^+(x,t)}{dt}|_{x=x_*(t),u=u_*(t),\dot{x}=f(x,u)} &= D_t v^+(x_*,t) + < D_x v^+(x_*,t), f(x_*,u_*) > \\ &\leq D_t v^+(x_*,t) + \max_{u\in U}\{< D_x v^+(x_*,t), f(x_*,u) >\} \\ &\leq \mu(t) \end{aligned} \tag{6.6}$$

where $v^+(x_*,t)$ is a piecewise continuous function, and $\mu(t)$ is a positive-definite, integrable function. Thus,

$$\frac{dv^+(x,t)}{dt}|_{x=x_*(t),u=u_*(t),\dot{x}=f(x,u)} \leq \mu(t) \tag{6.7}$$

By integrating (6.7) from 0 to τ, we obtain the over-approximate reachable set of the dynamical system (6.1) at time τ as:

$$V^+(\tau) = \{x | v^+(x,\tau) \leq \int_0^\tau \mu(t)dt + \max_{x(0)\in X_0} v^+(x(0),0)\} \tag{6.8}$$

Next, we review the polytopic over-approximation of reachable sets for linear dynamical systems and derive computational methods for polytopic over-approximate reachable sets for feedback linearizable nonlinear systems, linear dynamic games, and norm-bounded nonlinear systems.

6.2.1 Linear dynamical systems

In this section, we review the polytopic over-approximation of reachable sets for linear systems from [30]. Consider a time-varying linear dynamical system

$$\dot{x}(t) = A(t)x(t) + B(t)u(t), \quad x(0) \in X_0,\ u(t) \in U \tag{6.9}$$

where the initial set X_0 and the admissible control input set U are assumed to be convex polytopes which have N and N_u faces respectively. We assume the initial set X_0 is a polytope, but in general the number of faces of the initial set is a design parameter since X_0 may be a convex compact set and thus the more the number of faces of X_0 the better the over-approximate reachable sets.

A convex polytope $\mathcal{P}$ with K faces can be represented in two ways; it can be represented as the bounded intersection of K half spaces,

$$\mathcal{P} = \bigcap_{i=1}^{K} \{x | h_i^T x \leq \gamma_i\} \tag{6.10}$$

where h_i is a normal vector to the i^{th} face of the polytope $\mathcal{P}$. A convex polytope can also be represented as the convex hull of its vertices: if a convex polytope $\mathcal{P}$ has m vertices $\{v^1, \cdots, v^m\}$, then

$$\mathcal{P} = \{x | x = \sum_{i=1}^{m} \alpha_i v^i,\ \alpha_i \geq 0,\ \sum_{i=1}^{m} \alpha_i = 1\} \tag{6.11}$$

Define a set of affine functions as

$$v_i^+(x,t) = h_i^T(t)x, \quad i \in \{1, 2, \cdots, N\} \tag{6.12}$$

These affine functions are used to represent a convex polytope as shown in (6.10). In order to find a polytopic over-approximate reachable set, solve for $v_i^+(x,t)$ in (6.12) that satisfies (6.6). Then, (6.6) becomes

$$\begin{array}{rl} & D_t v_i^+(x,t) + \max_{u\in U}\{< D_x v_i^+(x,t), f(x,u) >\} \\ = & < \dot{h}_i(t), x(t) > + \max_{u\in U}\{< h_i(t), A(t)x(t) + B(t)u(t) >\} \\ = & < \dot{h}_i(t), x(t) > + < A(t)^T h_i(t), x(t) > + \max_{u\in U}\{< h_i(t), B(t)u(t) >\} \\ \leq & \mu(t) \end{array} \tag{6.13}$$

From optimal control theory [134, 131], the adjoint equation for linear systems when the input set does not depend on x is $\dot{\lambda}(t) = -A(t)^T\lambda(t)$. Since $\lambda(t) = h_i(t)$ in our case, then (for $i \in \{1, 2, \cdots, N\}$)

$$< \dot{h}_i(t), x(t) > + < A(t)^T h_i(t), x(t) >= 0, \tag{6.14}$$

This represents the evolution of the normal vector of the i^{th} face. Let $h_i(0),\ i \in \{1, 2, \cdots, N\}$ be the normal vectors of the faces of the initial set X_0. Then, the solution to (6.14) is

$$h_i(t) = \Phi(t, 0)h_i(0), \quad i \in \{1, 2, \cdots, N\} \tag{6.15}$$

where $\Phi(t, 0)$ is the state transition matrix satisfying $\dot{\Phi} = -A(t)^T\Phi,\ \Phi(0, 0) = I$. If the system dynamics in (6.9) is time invariant, then $\Phi(t, 0) = e^{-A^T t}$ and (6.15) becomes

$$h_i(t) = e^{-A^T t}h_i(0), \quad i \in \{1, 2, \cdots, N\} \tag{6.16}$$

Thus, for a linear time invariant system, the evolution of normal vectors can be determined analytically. We denote $\{u^1, \cdots, u^{m_u}\}$ as the vertices of the input set U. Since U is a convex polytope, the following must hold: (for $j \in \{1, \cdots, m_u\}$)

$$\max_{u \in U} < h_i(t), B(t)u(t) >= \max_j < h_i(t), B(t)u^j >\leq \mu(t) \tag{6.17}$$

that is, the maximum is achieved at a vertex of U [30]. Furthermore, if the system dynamics in (6.9) is time invariant, (6.17) is simplified to

$$\max_j < h_i(t), Bu^j >= \max_j < e^{-A^T t}h_i(0), Bu^j >\leq \mu(t) \tag{6.18}$$

for $j \in \{1, \cdots, m_u\}$. We choose $\mu(t) = \max_j < h_i(t), B(t)u^j >$ and note that $\mu(t)$ is always positive for a properly chosen input set U (e.g., chosen such that $0 \in U$). Then, the affine function $v_i^+(x, t)$ in (6.12) is a supporting hyperplane of the exact reachable set [30]. A polytopic over-approximate forward reachable set $V^+(t)$ for the dynamical system (6.9) is the intersection of half spaces as follows:

$$V^+(t) = \bigcap_{i=1}^{N}\{x : v_i^+(x, t) \leq \int_0^t \max_j < h_i(s), B(s)u^j > ds + \max_{x(0) \in X_o} v_i^+(x(0), 0)\} \tag{6.19}$$

$V^+(t)$ is a convex polytope which contains the exact reachable set at time t since each $v_i^+(x,t)$ in (6.19) is a supporting hyperplane of the exact reachable set. If the system dynamics is linear time invariant, $V^+(t)$ becomes

$$V^+(t) = \bigcap_{i=1}^{N} \{x : v_i^+(x,t) \leq \int_0^t \max_j < e^{-A^T s} h_i(0), Bu^j > ds + \max_{x(0)\in X_o} v_i^+(x(0),0)\} \quad (6.20)$$

6.2.2 Feedback linearizable nonlinear systems

In this section, we consider a class of nonlinear systems [135], in which $u(t)$ is a feedback control:

$$\dot{x}(t) = f(x) + g(x)u(t) \quad (6.21)$$

$$u(t) = a(x) + b(x)v(t) \quad (6.22)$$

We assume that there exists a diffeomorphism T: such that $z = T(x)$, which transforms, with the control input $u(t)$, the nonlinear system (6.21) into an equivalent linear system. Then, we can compute an over-approximate forward reachable set for the nonlinear system (6.21) as follows:

- Step 1: Transform the nonlinear system (6.21) to an equivalent linear system, $\dot{z}(t) = A(t)z(t) + B(t)v(t)$ with appropriate $u(t)$ and T.
- Step 2: Compute a polytopic over-approximate forward reachable set $V^+(t)$ of the linear system following the procedure in Section 6.2.1.
- Step 3: Using the inverse state transformation $x = T^{-1}(z)$, we obtain the over-approximate forward reachable set for the original nonlinear system (6.21) from $V^+(t)$.

Since there is no approximation during the transformation and the transformation is a diffeomorphism on a given domain of interest, the forward reachable set obtained in Step 3 is guaranteed to be an over-approximate forward reachable set of the nonlinear system (6.21).

6.2.3 Linear dynamic games

Now, we consider the linear dynamic game:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) + C(t)d(t), \quad x(0) \in X_0,\ u(t) \in U,\ d(t) \in D \tag{6.23}$$

where the initial set X_0, the admissible control input set U, and the disturbance input set D are assumed to be convex polytopes which have N, N_u, and N_d faces respectively. Then, the HJI equation in (6.3) for a forward reachable set computation becomes [33, 34],

$$D_t v(x,t) + \max_{u\in U}\min_{d\in D}\{< D_x v(x,t), A(t)x(t) + B(t)u(t) + C(t)d(t) >\} = 0 \tag{6.24}$$

To find an over-approximate solution to (6.24), we look for a set of affine functions $v_i^+(x,t)$ in (6.12) satisfying the following inequality,

$$\begin{aligned}
& D_t v_i^+(x,t) + \max_{u\in U}\min_{d\in D}\{< D_x v_i^+(x,t), A(t)x(t) + B(t)u(t) + C(t)d(t) >\} \\
= \ & < \dot{h}_i(t), x(t) > + \max_{u\in U}\min_{d\in D}\{< h_i(t), A(t)x(t) + B(t)u(t) + C(t)d(t) >\} \\
= \ & < \dot{h}_i(t), x(t) > + < A(t)^T h_i(t), x(t) > + \max_{u\in U}\{< h_i(t), B(t)u(t) >\} \\
& + \min_{d\in D}\{< h_i(t), C(t)d(t) >\} \\
= \ & \max_{u\in U}\{< h_i(t), B(t)u(t) >\} + \min_{d\in D}\{< h_i(t), C(t)d(t) >\} \\
\leq \ & \mu(t)
\end{aligned} \tag{6.25}$$

Since $< h_i(t), x(t) > + < A(t)^T h_i(t), x(t) >= 0$, the evolution of normal vectors is the same as (6.14). We denote $\{u^1, \cdots, u^{m_u}\}$ and $\{d^1, \cdots, d^{m_d}\}$ as the vertices of U and D respectively. Since (6.25) is linear with respect to u and d, the maximum and the minimum in (6.25) are achieved at vertices of U and D as follows:

$$\max_j < h_i(t), B(t)u^j > + \min_k < h_i(t), C(t)d^k > \leq \mu(t) \tag{6.26}$$

for ($j \in \{1, \cdots, m_u\}$, $k \in \{1, \cdots, m_d\}$).
We choose $\mu(t) = \max_j < h_i(t), B(t)u^j > + \min_k < h_i(t), C(t)d^k >$ and then a polytopic over-approximate reachable set $V^+(t)$ for a linear dynamic game (6.23) is

$$\begin{aligned}
V^+(t) = \ & \bigcap_{i=1}^{N} \ \{x : v_i^+(x,t) \leq \int_0^t [\max_j < h_i(s), B(s)u^j > + \min_k < h_i(s), B(s)d^k >] ds \\
& + \max_{x(0)\in X_0} v_i^+(x(0), 0)\}
\end{aligned} \tag{6.27}$$

6.2.4 Norm-bounded nonlinear systems

In this section, we consider a norm-bounded nonlinear system,

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) + \phi(x,t), \qquad x(0) \in X_0,\ u(t) \in U,\ \|\phi(x,t)\| \le \beta(t) \tag{6.28}$$

where the initial set X_0 and the admissible control input set U are assumed to be convex polytopes which have N and N_u faces respectively and $\|\cdot\|$ represents the Euclidean norm. $\beta(\cdot)$ is a positive-definite function. Then, the HJB equation in (6.3) becomes

$$D_t v(x,t) + \max_{u\in U}\{< D_x v(x,t), A(t)x(t) + B(t)u(t) + \phi(x,t) >\} = 0 \tag{6.29}$$

To compute an over-approximate solution to the HJB equation in (6.29), we find the affine functions $v_i^+(x,t)$ in (6.12) satisfying the following inequality,

$$\begin{aligned}
& D_t v_i^+(x,t) + \max_{u\in U}\{< D_x v_i^+(x,t), A(t)x(t) + B(t)u(t) + \phi(x,t) >\} \\
= \ & < \dot{h}_i(t), x(t) > + \max_{u\in U}\{< h_i(t), A(t)x(t) + B(t)u(t) + \phi(x,t) >\} \\
= \ & < \dot{h}_i(t), x(t) > + < A(t)^T h_i(t), x(t) > + \max_{u\in U}\{< h_i(t), B(t)u(t) >\} \\
& + < h_i(t), \phi(x,t) > \\
\le \ & \max_{u\in U}\{< h_i(t), B(t)u(t) >\} + \tfrac{1}{2}(\|h_i(t)\|^2 + \|\phi(x,t)\|^2) \\
\le \ & \max_j\{< h_i(t), B(t)u^j >\} + \tfrac{1}{2}(\|h_i(t)\|^2 + \beta(t)^2) \\
\le \ & \mu(t)
\end{aligned} \tag{6.30}$$

If we choose $\mu(t)$ such that

$$\mu(t) = \max_j < h_i(t), B(t)u^j > + \frac{1}{2}(\|h_i(t)\|^2 + \beta(t)^2) \tag{6.31}$$

then, a polytopic over-approximate reachable set $V^+(t)$ for the norm-bounded dynamical system (6.28) is

$$\begin{aligned}
V^+(t) = \ \bigcap_{i=1}^N \ & \{x : v_i^+(x,t) \le \int_0^t [\max_j < h_i(s), B(s)u^j > + \tfrac{1}{2}(\|h_i(s)\|^2 + \beta(s)^2)]ds \\
& + \max_{x(0)\in X_0} v_i^+(x(0),0)\}
\end{aligned} \tag{6.32}$$

If $\phi(x,t)$ belongs to a polytope with vertices $\{\phi^1, \cdots, \phi^{m_\phi}\}$, a polytopic over-approximate reachable set $V^+(t)$ becomes

$$\begin{aligned}
V^+(t) = \ \bigcap_{i=1}^N \ & \{x : v_i^+(x,t) \le \int_0^t [\max_j < h_i(s), B(s)u^j > + \max_k\{< h_i(s), \phi^k >\}]ds \\
& + \max_{x(0)\in X_0} v_i^+(x(0),0)\}
\end{aligned} \tag{6.33}$$

6.3 Examples

Here, we consider three examples: a linear system, a norm-bounded nonlinear system, and we conclude with the example which motivated this study, a nonlinear, feedback linearizable, dynamic game.

6.3.1 Linear dynamical systems

In this section, we consider a linear dynamical system $\dot{x} = Ax + Bu$, $x(0) \in X_0$ where the control input $u(t)$ can vary inside a convex polytope U and the initial set X_0 is also a convex polytope. The system parameters are defined as [125]

$$A = \begin{bmatrix} 0.0 & 1.0 & 0.0 & 0.0 \\ -8.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.1 \\ 0.0 & 0.0 & -4.0 & 0.0 \end{bmatrix}, \quad B = 1$$

$$X_0 = [0.0, 2.0] \times [-1.0, 1.0] \times [0.0, 2.0] \times [-1.0, 1.0]$$
$$U = [-0.5, 0.5] \times [-0.005, 0.005] \times [-0.5, 0.5] \times [-0.005, 0.005]$$

Figure 6.2 shows the evolution of the projection on x_3 and x_4 over time. This result is similar to that in [125], yet computation time with the method shown in Section 6.2.1 is 1.17 seconds (which includes plotting the result shown in Figure 6.2) using MATLAB on a 700MHz Pentium III PC. For comparison, the algorithm proposed in [125] takes 18 seconds using the same parameters.

6.3.2 Norm-bounded nonlinear systems

We consider a norm-bounded nonlinear system

$$\dot{x} = A(t)x + B(t)u(t) + \phi(x,t),\ x(0) \in X_0,\ u(t) \in U \tag{6.34}$$

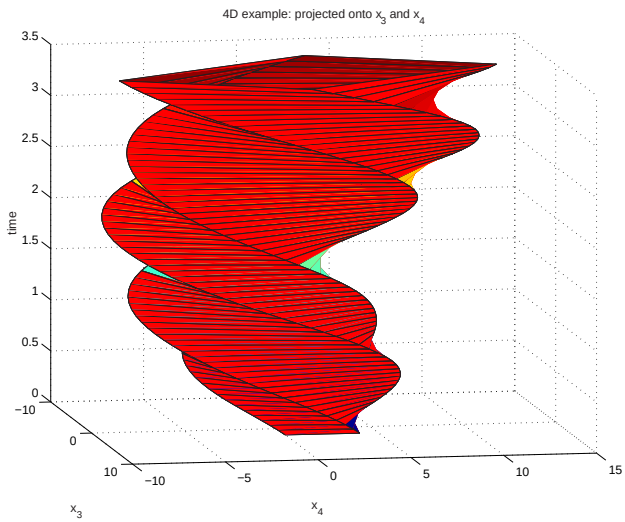


Figure 6.2: The forward reachable set of a four dimensional linear dynamical system (projection onto x_3 and x_4).

where the initial set X_0 and the control input set U are convex polytopes. The nonlinear function $\phi(x,t)$ is assumed to be norm-bounded i.e., $\|\phi(x,t)\| \leq \frac{1}{3}t$ where $t > 0$. The system parameters are defined as follows:

$$A = \begin{bmatrix} -0.5 & 4.0 \\ -3.0 & -0.5 \end{bmatrix}, \quad B = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

$$X_0 = [4,5] \times [4,5], \; U = [-0.1, 0.1]$$

The evolution of the forward reachable set over time is shown in Figure 6.3 and its computation time is 0.87 seconds (including plotting the result) using MATLAB on the same PC.

6.3.3 Conflict resolution between two aircraft

Last, we consider the two aircraft collision avoidance problem, as an example of feedback linearizable nonlinear systems and linear dynamic games. Figure 6.4 shows the relative configuration between two aircraft showing the protected zone. Aircraft 1 tries to avoid a conflict with aircraft 2 no matter how aircraft 2 behaves within the limit of its capability.

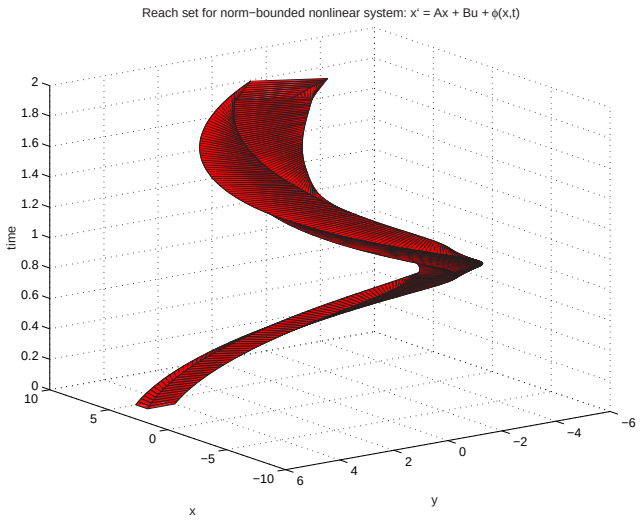


Figure 6.3: The forward reachable set of a norm-bounded nonlinear system.

Thus, we want to compute a backward reachable set (unsafe set) from the target set (protected zone) which represents the states from which the two aircraft would eventually have a conflict no matter how aircraft 1 tries to avoid it [4].

The planar kinematic model for each aircraft i $(i = 1, 2)$ in (6.5) is:

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\psi}_i \end{bmatrix} = \begin{bmatrix} \sigma_i \cos\psi_i \\ \sigma_i \sin\psi_i \\ \omega_i \end{bmatrix} \tag{6.35}$$

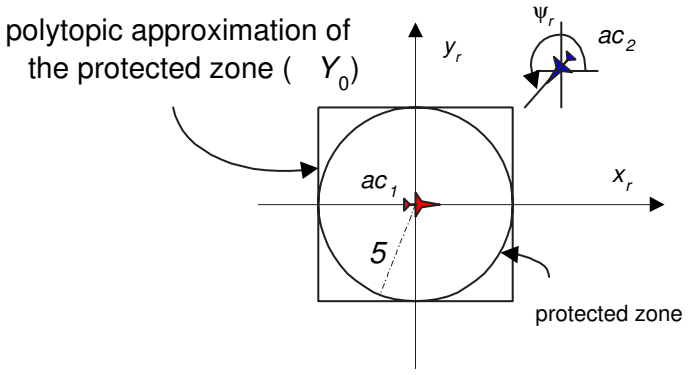


Figure 6.4: Relative configuration of two aircraft showing the protected zone.

Using dynamic extension [135] with σ_i as a new state variable, we can obtain a new nonlinear model which is feedback linearizable [136],

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\psi}_i \\ \dot{\sigma}_i \end{bmatrix} = \begin{bmatrix} \sigma_i \cos\psi_i \\ \sigma_i \sin\psi_i \\ \omega_i \\ a_i \end{bmatrix} \tag{6.36}$$

where a_i is the acceleration of aircraft i and is a new control input. Thus, the new state and input variables are $\xi_i := [x_i\, y_i\, \psi_i\, \sigma_i]^T$ and $\eta_i := [a_i\, \omega_i]^T$ respectively. We introduce a change of the state variables

$$z_i = T(\xi_i) \quad \text{such that} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix}_i = \begin{bmatrix} x_i \\ y_i \\ \sigma_i \cos(\psi_i) \\ \sigma_i \sin(\psi_i) \end{bmatrix} \tag{6.37}$$

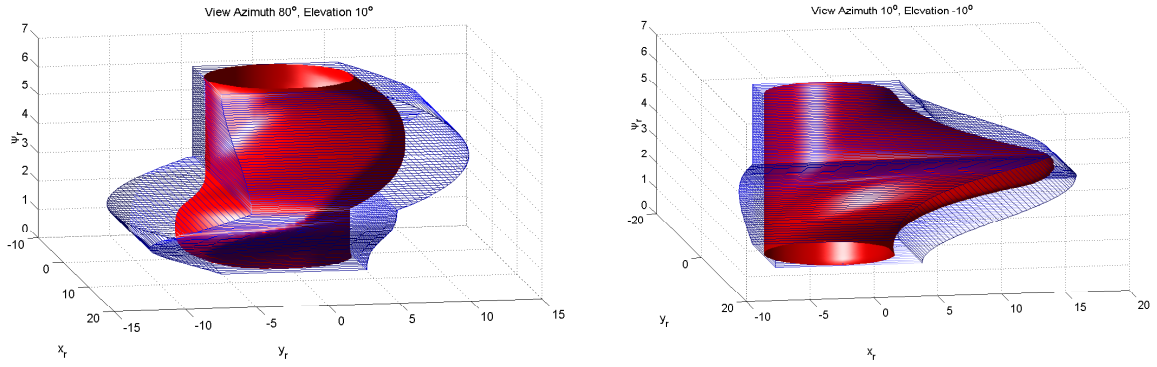
and a change of the input variables

$$\eta_i = M(\xi_i)u_i \quad \text{such that} \quad M(\xi_i) = \begin{bmatrix} \cos(\psi_i) & \sin(\psi_i) \\ -\sin(\psi_i)/\sigma_i & cos(\psi_i)/\sigma_i \end{bmatrix} \tag{6.38}$$

where u_i is the control input for the linearized model, and T and M are diffeomorphisms everywhere except at $\sigma_i = 0$. Then, the feedback linearized model of the nonlinear kinematic aircraft model in (6.36) obtained through the transformations in (6.37) and (6.38) is [136]:

$$\begin{aligned} \dot{z}_i &= \tfrac{\partial T}{\partial \xi}_i \dot{\xi}_i \Rightarrow \dot{z}_i = Az_i + Bu_i \\ \text{where} \qquad A &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \; B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned} \tag{6.39}$$

The relative kinematic aircraft model between two aircraft can be obtained by introducing new states $\xi_r = \xi_2 - \xi_1$ in the original nonlinear state space and $z_r = z_2 - z_1$ in the linearized



(a) (b)

Figure 6.5: Comparison between over-approximate and exact (Figure 6.1) backward reachable sets (unsafe sets) of conflict resolution between two aircraft: view from (a) Azimuth 80^o and elevation 10^o (b) Azimuth 10^o and elevation -10^o.

state space. Thus, a linearized relative kinematic aircraft model is

$$\dot{z}_r = Az_r + Bu_2 - Bu_1, \quad u_2 \in U,\, u_1 \in D, \tag{6.40}$$

where the admissible control input set U and the disturbance input set D are polytopes. This is a linear dynamic game since aircraft 1 (u_1) tries to keep aircraft 2 from entering into its protected zone (target set) to prevent a conflict, but aircraft 2 (u_2) tries to enter the protected zone of aircraft 1. For a backward reachable set computation for the linear dynamic game (6.40), the HJI equation in (6.24) becomes [33, 34],

$$D_t v(x,t) + \min_{u_2 \in U} \max_{u_1 \in D} \{< D_x v(x,t), A(t)x(t) + B(t)u_2(t)) - B(t)u_1(t) >\} = 0 \tag{6.41}$$

A target set (protected zone) is assumed to be $\mathcal{Y}_0 = [-5,5] \times [-5,5] \times [-\pi,\pi]$ as shown in Figure 6.4 in which we have performed the computation in four dimensions (6.40) and projected the result onto the relative coordinate in three-dimensional space. A polytopic over-approximate backward reachable set is first computed in the linearized space and then the over-approximate backward reachable set in the original state space is obtained through the transformations in (6.37) and (6.38). The over-approximate backward reachable set for conflict resolution with heading changes only, using the target set $\mathcal{Y}_0$, normalized aircraft speeds $\sigma_1 = \sigma_2 = 5$, angular velocities $|\omega_1| \leq 1$ and $|\omega_2| \leq 1$ is compared with the exact solution in [29] in Figure 6.5. The backward reachable set obtained by using the polytopic

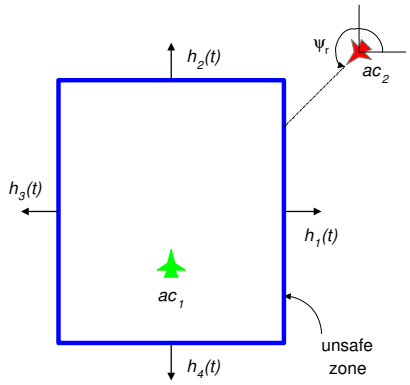


Figure 6.6: Conflict scenario: Aircraft 2 reaches the boundary of the unsafe zone of aircraft 1 with a given initial relative angle ψ_r.

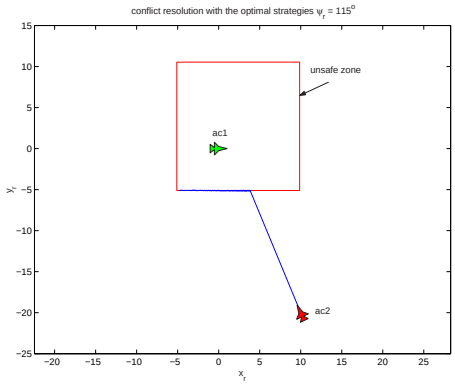


Figure 6.7: Conflict resolution simulation with relative initial states ($x_r = 10, y_r = -20, \psi_r = 115^o$). Aircraft 1 tries to avoid a conflict with aircraft 2 with the optimal strategy.

approximation is an over-approximation of the exact reachable set and its computation time is about 1.0 seconds (including plotting the result as shown in Figure 6.5) using MATLAB on the same PC, yet the exact solution [4] takes approximately 15 minutes on a Sun UltraSparc II with 50 grid nodes in each dimension. Figure 6.6 shows a conflict scenario in which aircraft 2 tries to enter the unsafe zone. When aircraft 2 reaches the boundary of the unsafe zone as shown in Figure 6.6, the optimal control input for aircraft 1 can be easily obtained as follows:

$$\begin{aligned} u_1^*(t) &= \arg\max_{u_1 \in D}\{< D_x v(x,t), -B(t)u_1(t)) >\} \\ &= \arg\max_j < e^{-A^T t} h_1(0), -Bu_1^j > \end{aligned} \tag{6.42}$$

Figure 6.7 shows a simulation for conflict resolution between the two aircraft with the initial

condition ($x_r = 10$, $y_r = -20, \psi_r = 115^o$). Since both aircraft behave optimally, the relative position of aircraft 2 moves along the boundary of the unsafe set. As expected, chattering occurs along the boundary. To avoid such a phenomenon, we could introduce a buffer zone around the boundary so that the control inputs change smoothly as aircraft 2 approaches the boundary.

Chapter 7

Conclusions and Future Work

The contributions of this dissertation are the development of a set of algorithms and tools for air traffic surveillance and control, with provable properties such as safety, convergence, and computational efficiency. For air traffic surveillance and control, accurate state estimates of aircraft are required because traffic advisories are highly dependent on them. The flight mode changes of an aircraft depend on the pilot's input, which is typically unknown to the surveillance system. In this dissertation, the flight mode logic of the aircraft is represented by discrete state dynamics governed by a finite Markov chain. Each discrete state has its own discrete-time continuous linear dynamics that describe the motion of the aircraft in the corresponding flight mode. In this way, a discrete-time stochastic linear hybrid system could be used to accurately represent the behavior of an aircraft over all of its flight regime. To estimate the aircraft state using such a model, hybrid estimation algorithms are needed.

In order to derive a hybrid estimation algorithm, we have constructed observability conditions for discrete-time stochastic linear hybrid systems and analyzed the performance of hybrid estimation algorithms for such hybrid systems. Based on the results of this analysis, we have developed a new hybrid estimation algorithm called the Residual-Mean Interacting Multiple Model (RMIMM) algorithm. RMIMM uses the mean of the residual computed by an individual Kalman filter so that it reduces mode estimation confusion and thus provides accurate state estimates. Using the RMIMM algorithm for state estimation, we have developed a Multiple-Target Tracking and Identity Management algorithm, which can keep track of multiple aircraft in clutter and maintain their identities at the same time. We have also

developed a Flight-Mode-Based Conflict Detection algorithm which uses not only continuous state estimates but also flight mode (discrete state) estimates computed by the RMIMM algorithm. In order to resolve conflicts detected, we have proposed a Protocol-Based Conflict Resolution algorithm for multiple-aircraft conflict resolution. The proposed conflict resolution algorithm is based on a closed-form analytic solution so that its computational time is negligible, and it is shown to be provably-safe within the limits of the model used. These algorithms may be used either as supporting tools to reduce the controllers' work load and to increase their performance, or as automation tools for some functions that are currently performed manually. For safety verification and synthesis of safe controllers, we have proposed an over-approximate method for reachable set computation using polytopic approximations, based on optimal control theory. The proposed method efficiently computes over-approximate reachable sets for linear systems, feedback linearizable nonlinear systems, linear dynamic games, and norm-bounded nonlinear systems. We have successfully solved a two-aircraft conflict avoidance problem in real time.

Future Work

The algorithms that we have developed in this dissertation have properties of computational efficiency, safety, and convergence, and could be applied to many applications related to the control of multiple-vehicle systems, such as the control of a group of mobile robots, as well as to ad hoc sensor networks.

Since hybrid estimation is quite general, its applications span many possible arenas. I am interested in extending the application of hybrid estimation algorithms to fault detection and isolation, and fault-tolerant estimation. Various faults of a dynamic system can be modelled by a hybrid system whose discrete states denote different error states, and the continuous dynamics in each discrete state describes the physical behavior of the system corresponding to the fault occurred. A hybrid estimation algorithm can detect and isolate a fault through its discrete state estimate. Since the continuous dynamics in the corresponding mode describes the behavior of the faulty system correctly, the hybrid estimation algorithm could provide accurate continuous state estimates even after a fault occurs in the system. As the number of faulty modes increases to account for many possible faults, the number of discrete states in a hybrid system should increase. And thus, the number of mode-matched Kalman filters in a hybrid estimation algorithm should increase accordingly. This not only increases the computational complexity of hybrid estimation but also could deteriorate the

performance of hybrid estimation due to mode confusion. Since the proposed RMIMM algorithm decreases mode estimation confusion using the mean of residual computed by mode-matched Kalman filters, we predict that the RMIMM algorithm would work better for such cases than any other existing multiple-model-based state estimation algorithm. If we can reduce the number of discrete state trajectories for hybrid estimation, we might further increase the performance of hybrid estimation algorithms. For this, I have an interest in the Viterbi algorithm [137] which computes the maximum likelihood discrete state trajectory. Using the Viterbi algorithm, we keep only a subset ($M < N$, where N is the number of discrete states in the hybrid system and M denotes the number of discrete state trajectories) of discrete state trajectories which have the M greatest likelihoods.

I am also interested in extending hybrid estimation to information inference (or system identification) for discrete-time stochastic linear hybrid systems. In practice, system identification methods for hybrid systems are quite ad hoc, consisting of highly simplified models of continuous dynamics and no systematic way of designing discrete dynamics. I have worked on this problem recently and have developed an algorithm that finds the maximum-likelihood values of parameters of the continuous dynamics in each discrete state, and also the transition probabilities between the discrete states [138]. I would like to extend this method to the fields of modelling of biological systems and the statistical analysis of time-series data.

The current Multiple-Target Tracking and Identity Management (MTIM) algorithm is centralized in the sense that it uses information about all aircraft. I would like to extend the MTIM algorithm to the distributed case. That is, an individual aircraft could track multiple targets and manage their identities within its surveillance region, and aircraft could communicate with each other only when necessary to get global information. The distributed algorithm could decrease computational complexity significantly and would be robust with respect to sensor or processor failure. This distributed version of the MTIM algorithm could be used for real-time onboard applications for control of multiple-vehicle systems (including air traffic control). In the ad hoc sensor network applications, each sensor has its own (possibly small) surveillance region and the global surveillance region is comprised of these individual surveillance regions of each sensor. In order to obtain global surveillance information, individual sensors have to communicate with each other. Then the problems to be addressed are *when* and *what* information has to be communicated and how to get global

information from local information. Therefore, the decentralized algorithm is essential for these applications.

For safety verification, I would like to extend the over-approximate reachable set computation to hybrid systems and other complex dynamic models. The current algorithm can compute reachable sets for two-player dynamic games. I am interested in extending the algorithm to multiple-player dynamic games for multiple-aircraft conflict avoidance. Since we use polytopic functions for approximation of reachable sets, we can compute approximate reachable sets computationally efficiently. In order to further increase computational efficiency, I would like to incorporate algorithms for computational geometry [139] which could efficiently handle polytopic objects.

Appendix A

Proof of Lemma 8 (Safety conditions for multiple-aircraft conflict resolution) in Section 5.2

Proof. Denote by P_1 the statement: there are no conflicts over $t \in [0, T_f]$; and by P_2 the statement $(C_1 \wedge C_2) \wedge (C_3 \wedge C_4) \wedge \cdots \wedge (C_{N(N-1)-1} \wedge C_{N(N-1)})$.

(1) Show (*if*) is true first, i.e., $P_2 \Rightarrow P_1$.
Assume P_2 is not true. This means at least, one of C_i $(i = 1, \cdots, N(N-1))$ is false. Without loss of generality, we suppose C_3 is false, i.e., $(S_{23}^1)_{min}(t) < R^2$. Then, there exists $t \in [0, \frac{T_f}{2}]$ such that $(d_{23}^1)_{min}(t) = d(t) < R$ at some $t \in [0, \frac{T_f}{2}]$. From the definition of a conflict, there is a conflict over $[0, \frac{T_f}{2}]$, thus, over $[0, T_f]$. This is a contradiction to P_1. Hence, $P_2 \Rightarrow P_1$ is true.

(2) Show (*only if*) is true, i.e., $P_1 \Rightarrow P_2$.
Assume there is at least one conflict over $[0, T_f]$. Without loss of generality, we suppose a conflict occurs between aircraft 1 and aircraft 3. This means that $d_{13}(t) < R$ at some $t \in [0, T_f]$. We consider two cases. First, we assume a conflict occurs at some $t \in [0, \frac{T_f}{2}]$. Then, $d_{13}(t) < R$ is true at some $t \in [0, \frac{T_f}{2}]$. Since $(d_{13}^1)_{min}(t) \leq d_{13}(t)$ at all $t \in [0, \frac{T_f}{2}]$, $(d_{13}^1)_{min}(t) < R$ is true at all $t \in [0, \frac{T_f}{2}]$. This is a contradiction to P_2. Hence, $P_1 \Rightarrow P_2$. Second, we suppose a conflict occurs at $t \in (\frac{T_f}{2}, T_f]$. Then, $d_{13}(t) < R$ occurs at some

$t \in (\frac{T_f}{2}, T_f]$. Since $(d_{13}^2)_{min}(t) \leq d_{13}(t)$ over $(\frac{T_f}{2}, T_f]$, $(d_{13}^2)_{min}(t) < R$ is true at all $t \in [0, \frac{T_f}{2}]$. This is a contradiction to P_2. Hence, $P_1 \Rightarrow P_2$.

Therefore, $P_1 \Leftrightarrow P_2$. $\square$

Appendix B

Validation of Protocol-Based Conflict Resolution with dynamic aircraft model

The protocols for multiple-aircraft conflict resolution have been derived using a kinematic aircraft model. However, validation with a dynamic aircraft model is necessary to show applicability of the protocols to real air traffic control problems. The lateral motion of an aircraft is controlled by the aileron δa (rad), and the rudder δr (rad). The important variables of an aircraft's lateral motion are: sideslip velocity v (ft/sec), yaw rate r (rad/sec), roll rate p (rad/sec), roll angle ϕ (rad), and heading angle ψ (rad).

A linear lateral dynamic model of a B747 which cruises with $M = 0.8$ at $40,000 ft$ altitude is used for validation [112]:

$$\begin{aligned} \dot{x} &= Ax + B\delta \\ y &= Cx \end{aligned} \tag{B.1}$$

where $x := [v\ r\ p\ \phi\ \psi]^T$ is the state vector, $\delta := [\delta_a\, \delta_r]^T$ is the control input, and $y := [v\, \psi]$ is the output, and the system parameters A, B, and C are given by

$$
\begin{aligned}
A &= \begin{bmatrix} -0.0558 & -7.74 & 0 & 0.322 & 0 \\ 0.0773 & -0.115 & -0.0318 & 0 & 0 \\ -0.394 & 0.388 & -0.465 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \\
B &= \begin{bmatrix} 0 & 0.0564 \\ 0.00775 & -0.475 \\ 0.143 & 0.153 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \\
C &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
\tag{B.2}
$$

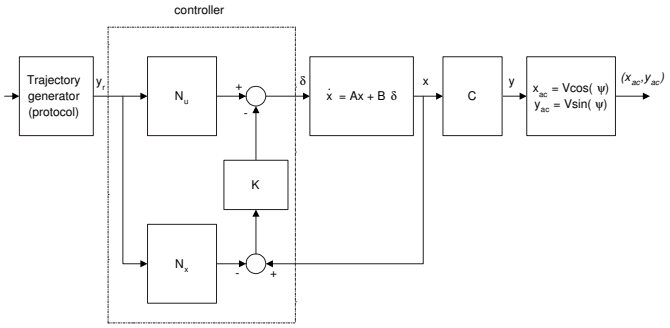
In Figure B.1-(a), the trajectory generator produces a conflict resolution trajectory according to the corresponding protocol for multiple-aircraft conflict resolution and the controller keeps the aircraft on the conflict resolution trajectory. The control law is:

$$\delta = -Kx + (N_u + KN_x)y_r \tag{B.3}$$

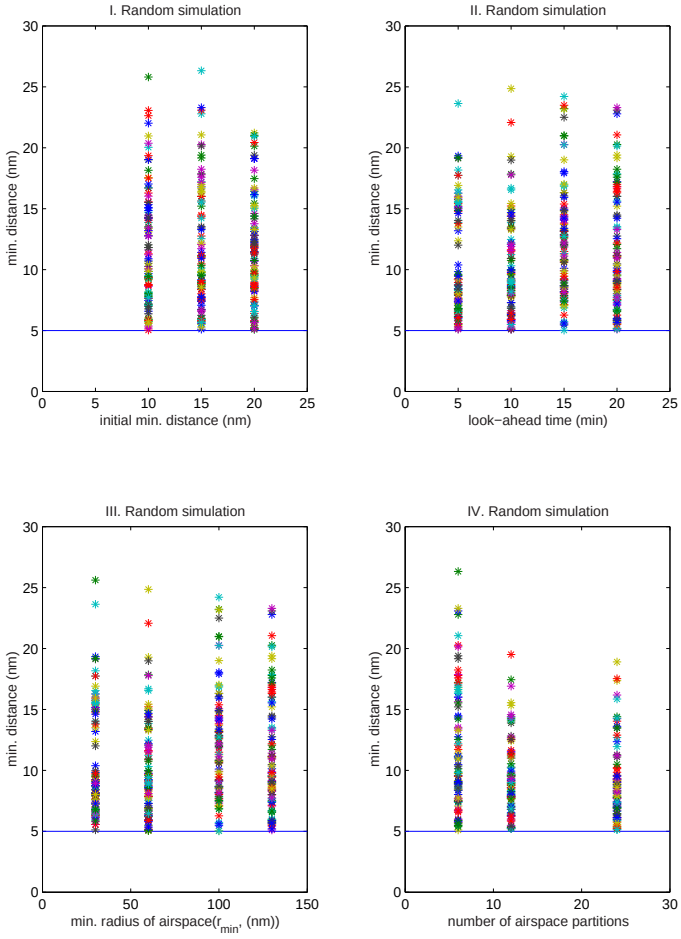
$$\text{where, } \begin{bmatrix} N_x \\ N_u \end{bmatrix} = \begin{bmatrix} A & B \\ C & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix}$$

The feedforward control in (B.3) makes the aircraft follow the conflict resolution trajectory and the feedback control K regulates the error dynamics. The feedback control gain K is designed using linear optimal control (LQR). All validation results in Section 4 and Section 5 have been obtained with the closed-loop dynamic model in Figure B.1-(a).

We validate our protocol for randomly generated five-aircraft conflict situations with different parameters and a dynamic aircraft model. For each parameter, simulation has been done for 200 different conflict configurations. Figure B.1-(b:(I)) shows random simulation results with different initial minimum distance such as $2R$, $3R$, and $4R$, and Figure B.1-(b:(II)) shows random simulation results with different look-ahead time such as $5min$, $10min$, $15min$, and $20min$. Figure B.1-(b:(III)) shows random simulation results with different minimum radius of airspace partition (r_{min}) such as $30nm$, $60nm$, $100nm$, and $130nm$



(a) **Block diagram of closed-loop lateral dynamics with a LQR controller for B747.**



(b) **Random simulation with different parameters and a dynamic aircraft model.**

Figure B.1: Validation with a dynamic aircraft model.

and Figure B.1-(b:(IV)) shows random simulation results with different number of airspace partitions such as 6, 12, and 24 partitions. The simulation results demonstrate that our protocol resolves all conflicts safely.

Appendix C

Robustness Analysis of Protocol-Based Conflict Resolution

Demonstrating robustness of the protocol for multiple-aircraft conflict resolution is important; the protocol must guarantee safety against uncertainty in the position, velocity, and heading of the aircraft. Here, we mean by robustness that the small perturbations in the system parameters causes small perturbations in the solution for conflict resolution. In the following, we show the protocol is robust with respect to uncertainty in sector boundary, to maneuver shape, and to asynchronous maneuvers for both exact and inexact maneuvers.

Robustness analysis with respect to sector boundaries

Consider Figure C.1-(a). We express sectors of allowable initial conditions for aircraft as $(r_{min} \leq r \leq r_{max}) \wedge (\theta_{min} \leq \theta \leq \theta_{max})$, shown as a shaded region in Figure C.1-(a). The uncertainties that are important for the protocol are those in r_{min} and $\delta\theta_{min}$, as seen from (5.11). Uncertainty in r_{min} is considered first. If r_{min} has $\pm 5\%$ error, a new heading change u_i can be obtained from (5.11) for both the exact conflict case and the inexact conflict case. For an exact conflict, u_i should be:

$$u_i \geq \sin^{-1}\left(\frac{R}{0.95 r_{min}\sqrt{2(1-\cos\delta\theta_{min})}}\right) \tag{C.1}$$

If Case 1 is considered, $u_i \geq 0.02864$ which differs from the original heading change by 5.29%. But since the heading change itself is small, the difference is only 0.0825^o. For Case 5, $u_i \geq 0.2344$ which differs from the original heading change by 5.35% and by 0.6818^o. From this analysis, we know that the protocols for multiple-aircraft conflict resolution are robust to uncertainty in r_{min}. To make conflict resolution in **Algorithm 2** and **Algorithm 3** safe with respect to uncertainty in r_{min}, a new r_{min} which takes into account this uncertainty should be used when each aircraft computes its own heading change u_i.

Now, consider uncertainty in $\delta\theta_{min}$. Since the protocol is derived from the worst case in each category, the control input, u_i, is the minimum heading change which can resolve conflicts for a given category. If the aircraft position is uncertain due to sensor errors, it can be represented by a region, usually a circle or an ellipse. If this region crosses the sector boundary as shown in Figure C.1-(a), the larger u_i must be chosen for safety. For example, in Figure C.1-(a), if there is no uncertainty in the position of aircraft 2, then this configuration corresponds to Case 1. Thus, the protocol for an exact conflict case is $u_i = 0.0272$. However, if there is uncertainty in the position of aircraft 2, then the protocol for an exact conflict case must be that for Case 2, i.e., $u_i = 0.0385$ rather than $u_i = 0.0272$. Similarly, the larger u_i must be chosen for safety for an inexact conflict case. Then, the protocols for multiple-aircraft conflict resolution are robust to uncertainty in $\delta\theta_{min}$.

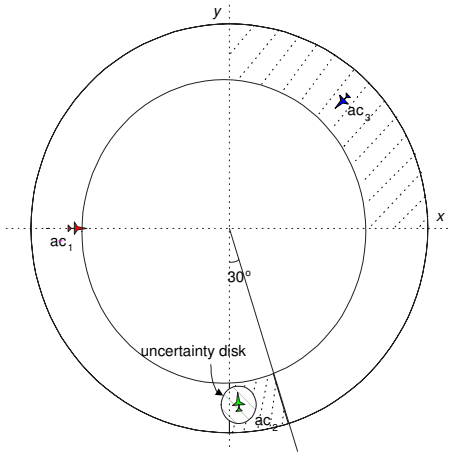
Robustness analysis with respect to circular arc turns

We assumed instantaneous heading changes when deriving the protocol for multiple-aircraft conflict resolution, impossible in a real situation. Thus, we consider robustness with respect to replacing corners by circular arc turns. Again, without loss of generality, we consider three-aircraft conflicts. In Figure C.1-(b), all three aircraft follow the conflict resolution paths: consider aircraft 1 and aircraft 2. Aircraft 1 and aircraft 2 reach q_1 and q_2 respectively at the same time. On the circular arcs, the two aircraft move with the same angular velocity, ω. This can be easily shown. The distance l_1 in Figure C.1-(b) is

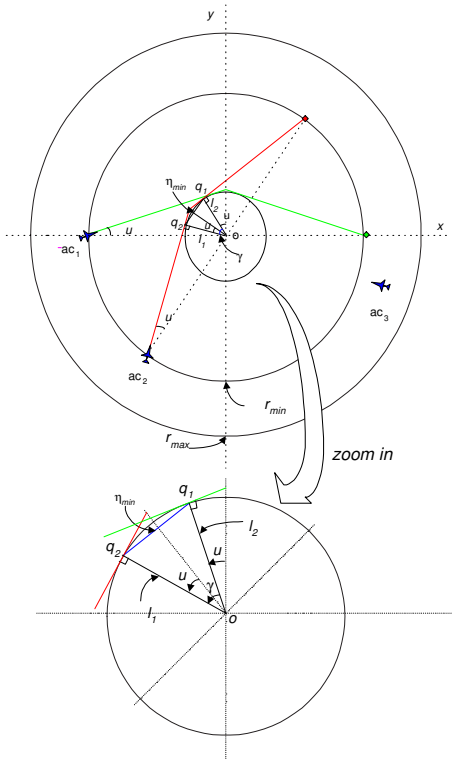
$$l_1 = v_1 T \cos u_1 \tag{C.2}$$

Thus, the angular velocity of aircraft 1 is:

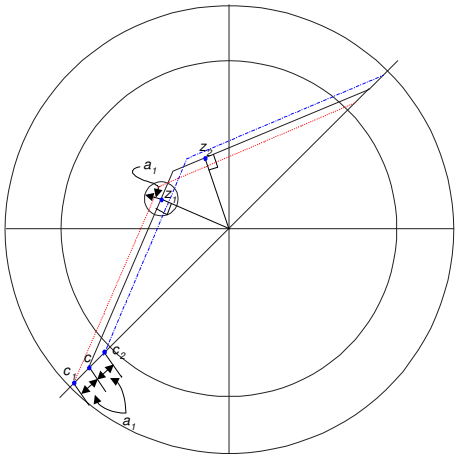
$$\omega_1 = \frac{v_1}{l_1} = \frac{v_1}{v_1 T \sin u_1} = \frac{1}{T \sin u_1} =: \omega \tag{C.3}$$



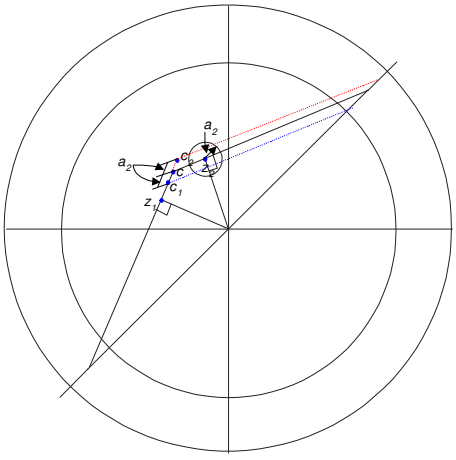
(a) **Airspace sectors and parameter uncertainty for Case 1.**



(b) **Robust analysis to smoothing corners.**



(c) **Asynchronous maneuver at a starting point.**



(d) **Asynchronous maneuver at a new waypoint.**

Figure C.1: Robustness analysis.

From (C.3), the angular velocity of aircraft 1 depends on T and u_1 only. Since $u_1 = u_2 = u_3$, the angular velocities of all three aircraft are the same.

To show robustness of the protocol, we need only to show now that the maneuvers are safe when the aircraft are on the circular arc paths. Of course, the worst case occurs when the distance between aircraft on the circular paths is minimum. From the geometry and the fact that all three aircraft have the same angular velocity during the resolution maneuver, the worst case is achieved between a point q_1 and a point q_2 when aircraft 1 and aircraft 2 are both on r_{min} and the relative angle between aircraft 1 and aircraft 2 is $\delta\theta_{min}$. The minimum distance between aircraft on the circular arcs from Figure C.1-(b) is:

$$\eta_{min}^2 = 2r_{min}^2(1 - \cos\gamma) \tag{C.4}$$

where γ is $\delta\theta_{min}$. Therefore, the minimum distance on the circular arcs is:

$$\eta_{min} = r_{min}\sqrt{2(1 - \cos\delta\theta_{min})} \tag{C.5}$$

If $\eta_{min} \geq R$ for all possible conflict cases over the circular arcs, the resolution maneuvers are safe, i.e. the protocol for three-aircraft conflict resolution is robust to smoothing corners. If (C.5) is applied to all six cases, the results are:

- Case 1: $90^o \leq \delta\theta_{min} \leq 120^o \Rightarrow \eta_{min} = 5.0004 > R \Rightarrow$ **safe**
- Case 2: $60^o \leq \delta\theta_{min} < 90^o \Rightarrow \eta_{min} = 5.0038 > R \Rightarrow$ **safe**
- Case 3: $45^o \leq \delta\theta_{min} < 60^o \Rightarrow \eta_{min} = 5.0026 > R \Rightarrow$ **safe**
- Case 4: $30^o \leq \delta\theta_{min} < 45^o \Rightarrow \eta_{min} = 5.0032 > R \Rightarrow$ **safe**
- Case 5: $10^o \leq \delta\theta_{min} < 30^o \Rightarrow \eta_{min} = 5.0005 > R \Rightarrow$ **safe**
- Case 6: $0^o \leq \delta\theta_{min} < 10^o \Rightarrow \eta_{min} = 5.0002 > R \Rightarrow$ **safe**

This shows that the protocol for an exact conflict is robust to replacing corners by circular arc turns. Since the protocol for an inexact conflict is based on the protocol for an exact conflict, robustness of the protocol for an inexact conflict to replacing corners by circular arc turns can be easily shown by following the same procedure for an exact conflict. Since angular velocities of all aircraft are the same, the minimum distance between aircraft remains same after replacing corners by circular arcs. Thus, the protocol for an inexact conflict perturbation case is also robust to replacing corners by circular arcs.

Robustness analysis with respect to asynchronous maneuvers

We assumed synchronous maneuvers, in which all aircraft in a resolution group change their heading at the same time, when deriving the protocol for multiple-aircraft conflict resolution. Here, we show that our protocol is robust with respect to asynchronous maneuvers.

First, consider the case in which aircraft initiate their resolution maneuver earlier or later than the exact resolution starting time, as shown in Figure C.1-(c). In Figure C.1-(c), c represents an exact resolution starting point. If an aircraft initiates its resolution maneuver earlier or later than the exact resolution starting point, c_1 and c_2 represent a real resolution starting point respectively and a_1 is the distance between c and c_1 (or c_2). If the aircraft's velocity is v, then the time discrepancy τ_1 between the exact resolution starting time and a real resolution starting time at the resolution starting point is defined as $\tau_1 = \frac{a_1}{v}$. From (5.6), the minimum distance among aircraft occurs at z_1 on the first segment and at z_2 on the second segment. Due to asynchronous maneuvers, the minimum distance can be changed by a_1 at z_1 and z_2. We use an uncertainty disk with radius a_1 at the z_1 and z_2 to accommodate this change in the minimum distance [140]. Then, for safety, a new predefined safety distance R_{safe} should be

$$R_{safe} = R + v\tau_1 \tag{C.6}$$

We consider the case in which aircraft change their heading before or after reaching a new waypoint c as shown Figure C.1-(d). In Figure C.1-(d), a_2 is the distance between c and c_1 (or c_2) and the time discrepancy τ_2 at the new waypoint is defined as $\tau_2 = \frac{a_2}{v}$. If the same procedure as previous is taken, a new predefined distance should be

$$R_{safe} = R + v\tau_2 \tag{C.7}$$

Therefore, in order to make our protocol safe against asynchronous maneuvers both at a resolution starting point and at a new waypoint, from (C.6) and (C.7), a new predefined safety distance should be

$$R_{safe} = R + v(\tau_1 + \tau_2) \tag{C.8}$$

Then, a new protocol for exact and inexact conflicts can be obtained from (5.9) with the new safety distance in (C.8). For example, if we consider a B747 cruising with M=8 at $40,000ft$

altitude and the maximum allowable time discrepancy both at a resolution starting point and at a new waypoint is $\tau_1 = 10sec$ and $\tau_2 = 10sec$ respectively, then a new safety distance is $R_{safe} = 7.53nm$.

Bibliography

[1] X.R. Li and Y. Bar-Shalom. Design of an Interacting Multiple Model algorithm for air traffic control tracking. *IEEE Transactions on Control Systems Technology*, 1(3):186–194, September 1993.

[2] R.R. LaFrey. ATC Surveillance: A Retrospective Look. In *NASA GRC CNS Conference*, April 2002.

[3] R.A. Paielli and H. Erzberger. Conflict probability estimation for free flight. *Journal of Guidance, Control, and Dynamics*, 20(3):588–596, 1997.

[4] I. Mitchell, A. M. Bayen, and C. J. Tomlin. Computing reachable sets for continuous dynamic games using level set methods. *IEEE Transactions on Automatic Control*, December 2001. Submitted.

[5] U.S. Federal Aviation Administration and Office of Policy and Plans. FAA Aerospace Forecasts, Fiscal Years 2000-2011. http://www.apo.data.faa.gov, 2000.

[6] EUROCONTROL Traffic Statistics and Forecasts Panel. Air Traffic Statistics and Forecasts (STATFOR) : Medium-Term Forecast of Annual Number of IFR Flights (2002-2009). Volume 1. Technical report, European Organization for the Safety of Air Navigation, 2002.

[7] T.S. Perry. In search of the future of air traffic control. *IEEE Spectrum*, 34(8):18–35, 1997.

[8] M.S. Nolan. *Fundamentals of Air Traffic Control*. Brooks/Cole, Wadsworth, 3rd edition, 1998.

[9] R. Gazit. *Aircraft Surveillance and Collision Avoidance Using GPS*. PhD thesis, Stanford University, Stanford, CA, August 1996.

[10] S. Kahne and I. Frolow. Air traffic management: Evolution with technology. *IEEE Control Systems Magazine*, 16(4):12–21, 1996.

[11] J.K. Kuchar and L.C. Yang. A review of conflict detection and resolution modeling methods. *IEEE Transactions on Intelligent Transportation Systems*, 1(4):179–189, December 2000.

[12] P.K. Menon and G.D. Sweriduk. Optimal strategies for free-flight air traffic resolution. *Journal of Guidance, Control, and Dynamics*, 22(2), March-April 1999.

[13] C. Tomlin. *Hybrid control of Air Traffic Management Systems*. PhD thesis, University of California, Berkeley, 1998.

[14] C. Tomlin, G.J. Pappas, and S. Sastry. Conflict resolution for air traffic management: A study in multiagent hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):509–521, April 1998.

[15] L.C. Yang and J.K. Kuchar. Prototype conflict alerting system for free flight. *Journal of Guidance, Control, and Dynamics*, 20(4):768–773, July-August 1997.

[16] P. Faure and V. Duong. On the applicability of free-flight mode in European airspace. In *Proceedings of the 2nd USA/EUROPE Air Traffic Management R&D Seminar*, Orlando, FL, December 1998.

[17] Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association*. Academic Press, 1988.

[18] H.A.P. Blom, R.A. Hogendoorn, and B.A. van Doorn. Design of a multisensor tracking system for advanced air traffic control. In Y. Bar-Shalom, editor, *Multitarget-Multisensor Tracking: Application and Advances*, volume 2, pages 31–63. Artech House, 1990.

[19] I. Hwang, J. Hwang, and C. Tomlin. Flight-mode-based aircraft conflict detection using a Residual-Mean Interacting Multiple Model algorithm. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, AIAA 2003-5340, Austin, TX, 2003. AIAA 2003-5340.

[20] H.A.P. Blom and Y. Bar-Sharlom. The interacting multiple model algorithm for systems with Markovian switching coefficients. *IEEE Transactions on Automatic Control*, 33(8):780–783, August 1988.

[21] S. S. Krause. *Avoiding Mid-Air Collisions*. TAB Books, 1995.

[22] N. J. Talotta. A field study for transponder performance in general aviation aircraft. *U.S. Department of Transportation Federal Aviation Administration Report DOT/FAA/CT-97/7*, December 1997.

[23] M. S. Nolan. *Fundamentals of Air Traffic Control*. Wadsworth Publishing Company, 2 edition, 1994.

[24] J. Shin, L.J. Guibas, and F. Zhao. A distributed algorithm for managing multi-target identities in wireless ad-hoc sensor networks. In F. Zhao and L. Guibas, editors, *Proceedings of the* 2^{nd} *Information Processing in Sensor Networks*, LNCS 2634, pages 223–238, Palo Alto, CA, April 2003.

[25] RTCA Special Committee. Minimum aviation system performance standards for automatic dependent surveillance broadcast (ADS-B). Technical report, January 1998.

[26] J.M. Alliot, N. Durand, and G. Granger. Faces: a free flight autonomous and coordinated embarked solver. In *Proceedings of the 2nd USA/EUROPE Air Traffic Management R&D Seminar*, Orlando, FL, December 1998.

[27] J. Lygeros, C. Tomlin, and S. Sastry. Controllers for reachability specifications for hybrid systems. *Automatica*, 35(3):349–370, 1999.

[28] C. J. Tomlin, J. Lygeros, and S. Sastry. A game theoretic approach to controller design for hybrid systems. *Proceedings of the IEEE*, 88(7):949–970, July 2000.

[29] I. Mitchell, A. Bayen, and C. Tomlin. Validating a Hamilton-Jacobi approximation to hybrid system reachable sets. In M. D. DiBenedetto and A. Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control.*, LNCS 2034, pages 418–431, Rome, Italy, 2001. Springer-Verlag.

[30] P. Varaiya. Reach set computation using optimal control. In *Proceedings of the KIT Workshop on Verification of Hybrid Systems*, pages 377–383, Grenoble, France, 1998.

[31] A. Kurzhanski and I. Valyi. *Ellipsoidal Calculus for Estimation and Control*. Birkhauser, 1997.

[32] A. Kurzhanski and P. Varaiya. Dynamic optimization for reachability problems. *Journal of Optimization Theory and Applications*, 108(2):227–251, 2001.

[33] A. Kurzhanski and P. Varaiya. On reachability under uncertainty. *SIAM Journal on Control and Optimization*, 41(1):181–216, 2002.

[34] A. Kurzhanski and P. Varaiya. Reachability analysis for uncertain systems-the ellipsoidal technique. *Dynamics of Continuous, Discrete & Impulsive Systems Series B: Applications & Algorithms*, 9(3):347–367, 2002.

[35] I. Hwang, H. Balakrishnan, and C. Tomlin. Observability criteria and estimator design for stochastic linear hybrid systems. In *Proceedings of the IEE European Control Conference*, Cambridge, UK, September 2003.

[36] I. Hwang, H. Balakrishnan, and C. Tomlin. Performance analysis of hybrid estimation algorithms. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, Maui, Hawaii, December 2003.

[37] I. Hwang, H. Balakrishnan, K. Roy, J. Shin, L. Guibas, and C. Tomlin. Multiple-target Tracking and Identity Management algorithm for Air Traffic Control. In *Proceedings of the* 2^{nd} *IEEE International Conference Sensors*, Toronto, Canada, October 2003.

[38] I. Hwang and C. Tomlin. Protocol-based conflict resoluiton for finite information horizon. In *Proceedings of the AACC American Control Conference*, pages 748–753, Anchorage, AK, May 2002.

[39] I. Hwang and C. Tomlin. Protocol-based conflict resolution for air traffic control. Technical Report SUDAAR-762, Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305, 2002.

[40] I. Hwang, D. M. Stipanović, and C. J. Tomlin. Applications of polytopic approximations of reachable sets to linear dynamic games and a class of nonlinear systems. In *Proceedings of the AACC American Control Conference*, Denver, CO, June 2003.

[41] D.M. Stipanović, I. Hwang, and C. Tomlin. Computation of an over-approximation of the backward reachable set using subsystem level set functions. In *Proceedings of the IEE European Control Conference*, Cambridge, UK, September 2003.

[42] D.M. Stipanović, I. Hwang, and C. Tomlin. Computation of an over-approximation of the backward reachable set using subsystem level set functions. *Dynamics of Continuous, Discrete, and Impulsive Systems*, December 2003.

[43] T. Kailath. *Linear Systems.* Prentice Hall, 1980.

[44] P.J. Ramadge. Observability of discrete event-systems. In *Proceedings of the* 25^{th} *IEEE Conference on Decision and Control*, pages 1108–1112, Athens, Greece, 1986.

[45] C.M. Özveren and A.S. Willsky. Observability of discrete event dynamic systems. *IEEE Transactions on Automatic Control*, 35:797–806, 1990.

[46] A. Alessandri and P. Coletta. Design of Luenberger observers for a class of hybrid linear systems. In M. D. DiBenedetto and A. Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control. LNCS*, volume 2034, pages 7–18. Springer-Verlag, 2001.

[47] A. Balluchi, L. Benvenuti, M.D. Di Benedetto, and A.L. Sangiovanni-Vincentelli. Design of observers for hybrid systems. In C. Tomlin and M.R. Greenstreet, editors, *Hybrid Systems: Computation and Control. LNCS*, volume 2289, pages 76–89. Springer-Verlag, 2002.

[48] R. Vidal, A. Chiuso, S. Soatto, and S. Sastry. Observability of linear hybrid systems. In *Hybrid Systems: Computation and Control. LNCS*, volume 2623, Prague, Czech Republic, April 2003. Springer-Verlag.

[49] A. Bemporad, G. Ferrari, and M. Morari. Observability and controllability of piecewise affine and hybrid systems. *IEEE Transactions on Automatic Control*, 45(10):1864–1876, 2000.

[50] Y. Baram and T. Kailath. Estimability and regulability of linear systems. *IEEE Transactions on Automatic Control*, 33(12):1116–1121, 1988.

[51] E.F. Costa and J.B.R. do Val. On the detectability and observability of discrete-time markov jump linear systems. In *Proceedings of the* 39^{th} *IEEE Conference on Decision and Control*, Sydney, Australia, December 2000.

[52] Y. Bar-Shalom and X.R. Li. *Estimation and Tracking: Principles, Techniques, and Software*. Artech House, Boston, 1993.

[53] L.R. Rabiner and B.H. Juang. An introduction to Hidden Markov Models. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 3(1):4–16, 1986.

[54] R. Vidal, A. Chiuso, and S. Soatto. Observability and identifiability of jump linear systems. In *Proceedings of the* 41^{th} *IEEE Conference on Decision and Control*, Las Vegas, NV, December 2002.

[55] P.S. Maybeck. *Stochastic Models, Estimation, and Control*, volume 2. New York: Academic Press, 1982.

[56] Y. Bar-Shalom, X.R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, 2001.

[57] D.T. Magill. Optimal adaptive estimation of sampled stochastic processes. *IEEE Transactions on Automatic Control*, 10(4):434–439, October 1965.

[58] D.G. Lainiotis and F.L. Sims. Performance measures for adaptive Kalman estimators. *IEEE Transactions on Automatic Control*, 15(2):249–250, April 1970.

[59] F.L. Sims, D.G. Lainiotis, and D.T. Magill. Recursive algorithm for the calculation of the adaptive Kalman filter weighting coefficients. *IEEE Transactions on Automatic Control*, 14(2):215–218, April 1969.

[60] D.G. Lainiotis. Optimal adaptive estimation: Structure and parameter adaptation. *IEEE Transactions on Automatic Control*, 16(2):160–170, April 1971.

[61] R.M. Hawkes and J.B. Moore. Performance bounds for adaptive estimation. *Proceedings of the IEEE*, 64(8):1143–1150, August 1976.

[62] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 1991.

[63] Y. Baram and N.R. Sandell. Consistent estimation on finite parameter sets with application to linear system identification. *IEEE Transactions on Automatic Control*, 23(3):451–454, June 1978.

[64] Y. Baram. A sufficient condition for consistent discrimination between stationary Gaussian models. *IEEE Transactions on Automatic Control*, 23(5):958–960, October 1978.

[65] M.J. Caputi. A necessary condition for effective performace of the multiple model adaptive estimator. *IEEE Transactions on Aerospace and Electronic Systems*, 31(3):1132–1138, July 1995.

[66] T. Kailath, A.H. Sayed, and B. Hassibi. *Linear Estimation*. Prentice Hall, New Jersey, 2000.

[67] P. Vacher, I. Barret, and M. Gauvrit. Design of a tracking algorithm for an advanced ATC system. In Y. Bar-Shalom, editor, *Multitarget-Multisensor Tracking: Application and Advances*, volume 2, pages 1–29. Artech House, 1990.

[68] X.R. Li and V.P. Jilkov. A survey of maneuvering target tracking: Dynamic models. In O.E. Drummond, editor, *Proceedings of SPIE: In Signal and Data Processing of Small Targets*, volume 4048, pages 212–235, 2000.

[69] P.D. Hanlon and P.S. Maybeck. Characterization of Kalman filter residuals in the presence of mismodeling. In *Proceedings of the* 37^{th} *IEEE Conference on Decision and Control*, pages 1254–1259, Tampa, FL, December 1998.

[70] R.A. Horn. *Matrix Analysis*. Cambridge University Press, 1985.

[71] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan. Interacting Multiple Model Methods in target tracking: A survey. *IEEE Transactions on Aerospace and Electronic Systems*, 34(1):103–123, January 1998.

[72] D.D. Sworder and J.E. Boyd. *Estimation Problems in Hybrid Systems*. Cambridge University Press, 1999.

[73] R.J. Fitzgerald. Development of practical PDA logic for multitarget tracking by microprocessor. In Y. Bar-Shalom, editor, *Multitarget-Multisensor Tracking: Advanced Applications*, volume 1, chapter 1, pages 1–23. Artech House, 1990.

[74] J.A. Roecker. A class of near optimal JPDA algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 30(2):504–510, April 1994.

[75] B. Zhou and N.K. Bose. Multitarget tracking in clutter: Fast algorithm for data association. *IEEE Transactions on Aerospace and Electronic Systems*, 29(2):352–362, April 1993.

[76] M. Hadzagic, H. Michalska, and A. Jouan. IMM-JVC and IMM-JPDA for closely maneuvering targets. In *Conference Record of the Thirty-Fifth Asilomar Conference on Signals, Systems and Computers*, volume 2, pages 1278 –1282, November 2001.

[77] R.L. Popp, K.R. Pattipati, and Y. Bar-Shalom. Dynamically adaptable m-best 2- D assignment algorithm and multilevel parallelization. *IEEE Transactions on Aerospace and Electronic Systems*, 35(4):1145 –1160, October 1999.

[78] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society of Industrial and Applied Mathematics*, 5(1):32–38, March 1957.

[79] F. Bourgeois and J.C. Lassalle. An extension of the Munkres algorithm for the assignment problem to rectangular matrices. *Communications of the Association for Computing Machinery*, 14(12):802–806, December 1971.

[80] S.S. Blackman. *Multiple-Target Tracking with Radar Applications*. Artech House, 1986.

[81] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe. Sonar tracking of multiple targets using Joint Probabilistic Data Association. *IEEE Journal of Oceanic Engineering*, OE-8(3):173–184, July 1983.

[82] R. Sinkhorn. Diagonal equivalence to matrices with prescribed row and column sums. *American Mathematical Monthly*, 74:402–405, 1967.

[83] M. de Feo, A. Graziano, R. Migliolo, and A. Farina. IMMJPDA versus MHT and Kalman Filter with NN correlation: Performance comparison. *IEEE Proceedings – Radar, Sonar and Navigation*, 144(2):49–56, April 1997.

[84] M. Prandini, J. Hu, J. Lygeros, and S. Sastry. A probabilistic approach to aircraft conflict detection. *IEEE Transactions on Intelligent Transportation Systems*, 1(4):199–219, December 2000.

[85] RTCA Special Committee. Minimum aviation system performance standards for automatic dependent surveillance-broadcast (ADS-B). Technical report, January 1998.

[86] J. Basch, L.J. Guibas, and J. Hershberger. Data structures for mobile data. In *Proceedings of the* 8^{th} ACM-SIAM *Symposium on Discrete Algorithms*, pages 747–756, 1997.

[87] L.J. Guibas. Kinetic data structures: A state of the art report. In *Proceedings of the* 3^{rd} *Workshop on Algorithmic Foundations of Robotics*, 1998.

[88] A. Bayen, S. Santhanam, I. Mitchell, and C. Tomlin. A differential game formulation of alert levels in ETMS data for high altitude traffic. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, AIAA 2003-5341, Austin, TX, August 2003.

[89] R. Teo and C. Tomlin. Computing danger zones for provably safe closely spaced parallel approaches. *Journal of Guidance, Control, and Dynamics*, 26(3):434–443, May-June 2003.

[90] M Prandini, J. Lygeros, A. Nilim, and S. Sastry. A probabilistic framework for aircraft conflict detection. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, AIAA 99 4144, Portland, OR, August 1999.

[91] J. Krozel, M. Peters, and G. Hunter. Conflict detection and resolution for future air transportation management. CR-97-205944, NASA Tech. Rep., April 1, 1997.

[92] R. Lachner. Collision avoidance as a differential game: Real-time approximation of optimal strategies using higher derivatives of the value function. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pages 2308–2313, Orlando, FL, October 12-15, 1997.

[93] R. Irvine. Gears conflict resolution algorithm. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, pages 786–796, AIAA 98-4236, Boston, MA, August 10-12, 1998.

[94] K. Harper, S. Mulgund, S. Guarino, A. Mehta, and G. Zacharias. Air traffic controller agent model for free flight. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, pages 288–301, AIAA 99-3987, Portland, OR, August 9-11, 1999.

[95] I. Burdun and O. Parfentyev. AI knowledge model for self-organizing conflict prevention/resolution in close free-flight air space. In *Proceedings of the IEEE Aerospace Conference*, pages 409–428, Snowmass, CO, March 1999.

[96] A. Bicchi and L. Pallottino. On optimal cooperative conflict resolution for air traffic management systems. *IEEE Transactions on Intelligent Transportation Systems*, 1(4):221–232, 2000.

[97] E. Frazzoli, Z.H. Mao, J.H. Oh, and E. Feron. Resolution of conflicts involving many aircraft via semidefinite programming. *Journal of Guidance, Control, and Dynamics*, 24(1):79–86, February 2001.

[98] J. Hu, M. Prandini, and S. Sastry. Optimal maneuver for multiple aircraft conflict resolution: a braid point of view. In *Proceedings of the IEEE Conference on Decision and Control*, pages 4164–4169, Sydney, Australia, December 2000.

[99] K.D. Bilimoria. A geometric optimization approach to aircraft conflict resolution. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, AIAA 2000-4265, Denver, CO, August 2000.

[100] Z. Mao, E. Feron, and K. Bilimoria. Stability of intersecting aircraft flows under decentralized conflict avoidance rules. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, AIAA 2000-4271, Denver, CO, August 2000.

[101] N. Durand, J. Alliot, and O. Chansou. An optimizing conflict solver for ATC. *Air Traffic Quarterly*, 3(3):139–161, 1995.

[102] R. Schild. *Rule Optimization for Airborne Aircraft Separation.* PhD thesis, T.U, Wien, 1998.

[103] K. Bilimoria, B. Sridhar, and G. Chatterji. Effects of conflict resolution maneuvers and traffic density of free flight. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, AIAA 96-3767, San Diego, CA, 1996.

[104] B. Carpenter and J. Kuchar. Probability-based collision alerting logic for closely-spaced parallel approach. In *Proceedings of the AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, January 1997.

[105] V. Duong, E. Hoffman, L. Floc'hic, and J.P. Nicolaon. Extended flight rules to apply to the resolution of encounters in autonomous airborne separation. Technical report, EUROCONTROL Experimental Center, September 1996.

[106] J. Kosecka, C. Tomlin, G. Pappas, and S. Sastry. Generation of conflict resolution maneuver for air traffic management. In *Proceedings of the International Conference on Robotics and Intelligent Systems*, Grenoble, France, September 1997.

[107] J. Hoekstra, R. van Gent, and R. Ruigrok. Conceptual design of free flight with airborne separation assurance. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, pages 807–817, AIAA 98-4239, Boston, MA, August 1998.

[108] M. Eby and W. Kelly. Free flight separation assurance using distributed algorithms. In *Proceedings of the IEEE Aerospace Conference*, pages 429–441, Snowmass, CO, March 1999.

[109] K. Zeghal. A review of different approaches based on force fields for airborne conflict resolution. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, pages 818–827, AIAA 98-4240, Boston, MA, August 1998.

[110] V. Duong and K. Zeghal. Conflict resolution advisory for autonomous airborne separation in low-density airspace. In *Proceedings of the IEEE Conference on Decision and Control*, San Diego, CA, December 1997.

[111] K. Zeghal and J. Ferber. A reactive approach for distributed air traffic control. In *Proceedings of the International Conference on Artificial Intelligence and Expert Systems*, pages 381–390, Avignon, France, 1993.

[112] A.E. Bryson. *Control of Spacecraft and Aircraft*. Princeton University Press, 1994.

[113] J. Hoekstra, R. Ruigrok, R. van Gent, J. Visser, B. Gijsbers, M. Valenti, W. Heesbeen, B. Hilburn, J. Groeneweg, and F. Bussink. Overview of nlr free flight project 1997-1999. Technical Report NLR-CR-2000-227, National Aerospace Laboratory (NLR), May 2000.

[114] M. G. Crandall, L. C. Evans, and P.-L. Lions. Some properties of viscosity solutions of Hamilton-Jacobi equations. *Transactions of American Mathematical Society*, 282(2):487–502, 1984.

[115] G. Leitmann. *An Introduction to Optimal Control*. Mc-Graw-Hill, New York, NY, 1967.

[116] R. Vinter. A characterization of the reachable set for nonlinear control systems. *SIAM Journal on Control and Optimization*, 18(6):599–610, 1980.

[117] M. M. Khrustalev. Exact description of reachable sets and global optimality conditions. *Avtomatika i Telemekhanika*, (5):62–70, 1988.

[118] M. M. Khrustalev. Exact description of reachable sets and global optimality conditions ii: conditions of global optimality. *Avtomatika i Telemekhanika*, (7):70–80, 1988.

[119] A. Tiwari and G. Khanna. Series of abstractions for hybrid automata. In C. Tomlin and M. Greenstreet, editors, *5th International Workshop, Hybrid Systems: Computation and Control*, volume LNCS 2289, pages 465–478, Stanford, CA, 2002.

[120] R. Alur, T. Dang, and F. Ivančić. Reachability analysis of hybrid sytems via predicate abstraction. In C. Tomlin and M. Greenstreet, editors, *the 5th International Workshop, Hybrid Systems: Computation and Control*, volume LNCS 2289, pages 35–48, Stanford, CA, 2002. Springer.

[121] I. Hwang, H. Balakrishnan, R. Ghosh, and C. Tomlin. Reachability analysis of delta-notch lateral inhibition using predicate abstraction. In S. Sahni, V.K. Prasanna, and U. Shukla, editors, *High Performance Computing-HiPC2002.*, LNCS 2552, pages 715–724, Stanford, CA, 2002. Springer-Verlag.

[122] A. Chutinan and B. H. Krogh. Computing polyhedral approximations to flow pipes for dynamic systems. In *Proceedings of the* 37^{th} *IEEE Conference on Decision and Control*, pages 2089–2094, 1998.

[123] A. Chutinan and B. H. Krogh. Verification of polyhedral-invariant hybrid automata using polygonal flow pipe approximations. In F. Vaandrager and J. H. van Schuppen, editors, *Hybrid Systems: Computation and Control.*, LNCS 1569, pages 76–90, Berg en Dal, The Netherlands, 1999. Springer-Verlag.

[124] T. Dang. *Vérification et synthèse des systèmes hybrides*. PhD thesis, Institut National Polytechnique de Grenoble (Verimag), France, 2000.

[125] E. Asarin, O. Bournez, T. Dang, and O. Maler. Approximate reachability analysis of piecewise-linear dynamical systems. In B. Krogh and N. Lynch, editors, *Hybrid Systems: Computation and Control.*, LNCS 1790, pages 20–31, Pittsburgh, PA, 2000. Springer-Verlag.

[126] T. Dang and O. Maler. Reachability analysis via face lifting. In T.A. Henzinger and S. Sastry, editors, *Hybrid Systems: Computation and Control.*, LNCS 1386, pages 96–109, Berkeley, CA, 1998. Springer-Verlag.

[127] E.K. Kostousova. State estimation for dynamic systems via parallelotopes. *Optimization Methods and Software*, 9:269–306, 1998.

[128] A. Kurzhanski and P. Varaiya. Ellipsoidal techniques for reachability analysis. In B. Krogh and N. Lynch, editors, *Hybrid Systems: Computation and Control.*, LNCS 1790, pages 202–214, Pittsburgh, PA, 2000. Springer-Verlag.

[129] O. Botchkarev and S. Tripakis. Verification of hybrid systems with linear differential inclusions using ellipsoidal approximations. In B. Krogh and N. Lynch, editors, *Hybrid Systems: Computation and Control.*, LNCS 1790, pages 73–88, Pittsburgh, PA, 2000. Springer Verlag.

[130] V. F. Krotov. *Global Methods in Optimal Control Theory.* Marcel Dekker, Inc, New York, NY, 1996.

[131] T.L. Vincent and J. Grantham. *Nonlinear and Optimal Control Systems.* John Wiley and Sons, New York, NY, 1997.

[132] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[133] I. Mitchell. Games of two identical vehicles. Technical report, SUDAAR 740, Stanford University Department of Aeronautics and Astronautics, July 2001.

[134] A. Bryson and Y.C. Ho. *Applied Optimal Control.* Tayler and Francis, 1975.

[135] S. Sastry. *Nonlinear Systems: Analysis, Stability, and Control.* Springer-Verlag, New York, NY, 1999.

[136] D.M. Stipanović, G.İnalhan, R. Teo, and C. Tomlin. Decentralized overlapping control of a formation of unmanned aerial vehicles. In *Proceedings of the* 41^{th} *IEEE Conference on Decision and Control*, pages 2829–2835, Las Vegas, NV, December 2002.

[137] G. D. Forney Jr. The Viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, March 1973.

[138] H. Balakrishnan, I. Hwang, J. Jang, and C. Tomlin. Inference methods for autonomous stochastic linear hybrid systems. In *Hybrid Systems: Computation and Control, Lecture Notes in Computer Science*, Philadelphia, PA, March 2004. Springer Verlag. Submitted.

[139] T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to Algorithms.* The MIT Press, Cambridge, Massachusetts, 1990.

[140] J. How. Lecture notes for robust control analysis and synthesis. Stanford University, 1997-98.