

**DATA-DRIVEN DECISION SUPPORTING TOOLS FOR
AIRCRAFT CONFLICT RESOLUTION AND
CONFORMANCE MONITORING**

by

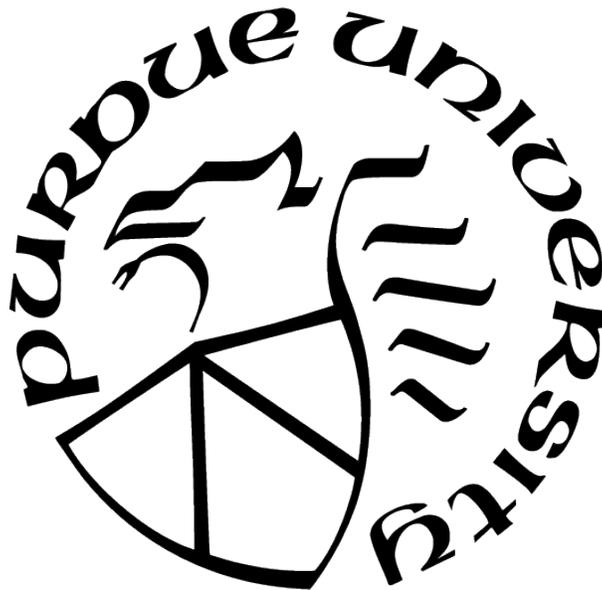
Kwangyeon Kim

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy



School of Aeronautics and Astronautics

West Lafayette, Indiana

August 2021

**THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL**

Dr. Inseok Hwang, Chair

School of Aeronautics and Astronautics

Dr. Martin Corless

School of Aeronautics and Astronautics

Dr. Jianghai Hu

School of Electrical and Computer Engineering

Dr. Dengfeng Sun

School of Aeronautics and Astronautics

Approved by:

Dr. Gregory A. Blaisdell

To my family

ACKNOWLEDGMENTS

First and foremost, I am grateful to my advisor, Professor Inseok Hwang, for his guidance and support for my PhD study. From his insight, enthusiasm, and persistence for research, I have learned a lot about what aspects an excellent researcher, engineer, and advisor should have. Thanks to his patience and motivation, I could complete my PhD.

I also would like to thank my committee members, Professor Martin Corless, Professor Jianghai Hu, and Professor Dengfeng Sun, for their valuable comments and encouragement, as well as their excellent teaching.

I sincerely appreciate my current and past lab members. I'd like to express special gratitude to Dr. Raj Deshmukh and Hong-Cheol Choi for their help in completing this thesis.

Last but not least, I would like to thank my parents, Jinhwan Kim and Misook Kim, for their endless love, patience, and support.

TABLE OF CONTENTS

LIST OF TABLES	6
LIST OF FIGURES	7
ABSTRACT	9
1 INTRODUCTION	11
1.1 Background and motivation	11
1.2 Objectives and contributions	17
1.3 Outline of dissertation	19
2 DATA-DRIVEN GENERATION OF CONFLICT RESOLUTION MANEUVERS IN EN ROUTE AIRSPACE	20
2.1 Intent-based Detection and Characterization of Aircraft Maneuvers in En Route Airspace	20
2.2 Data-Driven Conflict Resolution Generator based on Supervised Learning . .	43
3 DATA-DRIVEN CONFORMANCE MONITORING IN TERMINAL AIRSPACE	58
3.1 Hybrid Data-driven and Physics-based Trajectory Prediction	58
3.2 Conformance Monitoring with Stochastic Conformal Prediction	63
4 CONCLUSION	73
4.1 Concluding remarks	73
4.2 Future Work	74
REFERENCES	75
VITA	81

LIST OF TABLES

2.1	Maneuver Types	22
2.2	Intent Sets	27
2.3	Maneuver Models	27
2.4	Characteristics of Maneuvers	34
2.5	Feature vector	46
2.6	RPs for the horizontal RTs	51
2.7	RPs for the vertical RTs	51
2.8	RPs for the speed RTs	51
2.9	Illustrative case: RT classification	56
2.10	Illustrative case: Advisory list of RTs and corresponding RPs	56

LIST OF FIGURES

2.1	Proposed framework for the identification of aircraft maneuvers	21
2.2	Maneuvers in the horizontal dimension	23
2.3	Maneuvers in the vertical dimension	24
2.4	Maneuvers in the speed dimension	25
2.5	Intent inference	32
2.6	Maneuver models (horizontal)	34
2.7	PS and RO	35
2.8	Maneuver models (vertical)	35
2.9	SA-D and TA-D	36
2.10	Maneuver models (speed)	36
2.11	C-CS and TC-CS	36
2.12	Overall performance results	37
2.13	Illustrative case: Flight plan and track	38
2.14	Illustrative case: Results of the proposed algorithm	39
2.15	Results of the proposed algorithm for the maneuver types in the horizontal dimension	40
2.16	Results of the proposed algorithm for the maneuver types in the vertical dimension	41
2.17	Results of the proposed algorithm for the maneuver types in the speed dimension	42
2.18	Proposed framework: Data-Driven Resolution Generator (D2RG)	43
2.19	Learning of D2RG	44
2.20	Conflict situations	47
2.21	RT: Hierarchical classification	48
2.22	Learning of RPs	49
2.23	RP-action: Path stretch	50
2.24	Application of D2RG to an unseen conflict situation	52
2.25	Minimum distance between two conflicting aircraft	54
2.26	Illustrative case: Trajectories of two conflicting aircraft in the horizontal dimension	55
2.27	Illustrative case: Trajectories with resolution maneuvers	57

2.28	Illustrative case: Separation between two conflicting aircraft	57
3.1	Framework for hybrid data-driven and physics-based trajectory and conformity prediction	58
3.2	Framework for hybrid data-driven and physics-based trajectory prediction	59
3.3	Structure of Long Short-Term Memory (LSTM)	60
3.4	Arrival trajectory prediction	63
3.5	Departure trajectory prediction	63
3.6	Comparison of the computation time: Monte Carlo simulation and the proposed approximation	66
3.7	Conformance monitoring at the current time: Direct-to anomaly (departure)	68
3.8	Conformance monitoring at the current time of departure trajectories with Path stretch anomaly (departure)	68
3.9	Conformance monitoring at the current time: Direct-to anomaly (arrival)	69
3.10	Conformance monitoring at the current time: Path stretch anomaly (arrival)	69
3.11	Conformance monitoring at the future time: normal trajectory.	72
3.12	Conformance monitoring at the future time: abnormal trajectory no. 1	72
3.13	Conformance monitoring at the future time: abnormal trajectory no. 2	72

ABSTRACT

In air traffic management, the primary goal is the safety and efficiency of airspace operations under the responsibility of air traffic controllers (ATCs). With the growing demand of air traffic, it becomes critical to develop advanced techniques to support the decisions made by ATCs, which include control and monitoring of air traffic. To reduce the workload on ATCs in both control and monitoring, this thesis focuses on the development of decision supporting tools for (i) aircraft conflict resolution in en-route airspace and (ii) conformance monitoring in terminal airspace.

The first part of this thesis focuses on the development of a data-driven conflict resolution tool which can aid the decision-making process of ATCs for air traffic conflict resolution. The decision-making process can be viewed as a system that takes conflict situations as input and generates corresponding conflict resolution methods as output. That is, each conflict can be represented as a tuple of (Conflict Situation, Resolution Methods). To construct a conflict data in this form from air traffic surveillance data, we first need to label each conflict situation, or identify resolution methods (outputs) used for the conflict situation. The key idea is that any complex maneuvers can be modeled as a sequence of simple or primitive motions, called intents. Using the domain knowledge obtained from flight data and the intent inference algorithm, a framework for the detection and characterization of aircraft resolution maneuvers is proposed to identify resolution types and resolution parameters. Based on the knowledge extracted from the constructed conflict data with the features representing conflict situations (or inputs), a classification model is designed which determines the resolution type for every two-aircraft conflict in the airspace. In addition to predicting the resolution type, the proposed conflict resolution algorithm can also suggest appropriate resolution parameters for the guaranteed safe separation. The combination of the resolution type prediction model and resolution parameter suggestion model can sufficiently and safely resolve any two aircraft conflict.

The second part of the thesis is for the development of a conformance monitoring methodology for the current and future time, to help enhance the situational awareness of ATCs. To predict the future states of an aircraft, a trajectory prediction framework is developed

by combining a data-driven prediction model, which generates expected states of an aircraft learned from flight data, and a physics-based prediction method, which incorporates the current motion of an aircraft. Since the estimated or predicted states of an aircraft are stochastic, a stochastic version of anomaly detection and prediction algorithm for sequentially updated aircraft trajectories is developed using a smooth approximation for numerical integration.

All the proposed methods are demonstrated with real flight data to show their potentials as decision supporting tools that can help reduce the workload on air traffic controllers and enhance the safety of air traffic operations.

1. INTRODUCTION

In air traffic management (ATM), the primary goal is the safety and efficiency of airspace operations managed by air traffic controllers (ATCs). With the growing demand of air traffic [1], it becomes critical to develop advanced techniques to support the decisions made by ATCs, which include control and monitoring of air traffic:

1. Control: Although airspace is efficiently divided into operational classes and flight plans are scheduled to assure separation between airborne aircraft, ATCs intentionally instruct deviations of aircraft from their flight plans for the tasks such as aircraft conflict resolution, avoidance of weather cells, or sequencing and scheduling [2].
2. Monitoring: Any deviations of an aircraft from its flight plan are monitored by ATCs, or if ATCs instruct a deviation intentionally for the purpose of ATM, ATCs should monitor whether the aircraft follows the instructed path or not. Since any deviation against the intent of aircraft (following flight plan or the instructions of ATCs) could compromise the safety and efficiency of the air traffic operations, it is critical to monitor the aircraft's behavior by understanding their current status and predicting the futures states under the ATCs' control.

In order to reduce the workload of ATCs in both control and monitoring by helping their decision-making process, several tools have been developed. In this thesis, we focus on specific kinds of such tools, that is, decision supporting tools for (i) aircraft conflict resolution in en-route airspace (control) and (ii) conformance monitoring in terminal airspace (monitoring).

1.1 Background and motivation

In this section, a literature review of the aforementioned tools is presented along with the motivation that drives the research presented in this thesis.

1.1.1 Conflict resolution in en-route airspace

In the current practice, aircraft conflicts are resolved by ATCs by comparing the flight plans and the predicted flight trajectories. There have been several automation tools developed for conflict resolution such as Center-TRACON Automation System (CTAS) [3], [4], Traffic Alert and Collision Avoidance System (TCAS) [5], [6], and Autoresolver [7], [8].

In the research efforts, the conflict resolution has been posed as an optimal control problem. In [9], [10], the uncertainty due to wind is considered by using a stochastic optimal control. In [11], the wind uncertainty is modeled as stochastic variables and the conflict intensity is computed by employing the probability transformation method. The conflict resolution has also been posed as an optimization problem. The cost function to be minimized is set as the deviation from a flight plan with the constraint of the safe separation between aircraft. The formulated optimization problems have been solved by using various methods, such as Metaheuristic [12], [13], Integer Linear Programming [14], [15], and Constraint Satisfaction Program [16], [17].

The methods mentioned above can fall into flight dynamics (or physics)-based methods [18] which generate a static protocol. Updating such static protocols would be challenging because it requires a new model that can address the uncertainties due to weather and/or pilot's response. Also, for the increased air traffic demand or newly introduced concept for airspace operations, scaling up the protocols can be challenging.

In this regard, data-driven approaches have been proposed to address these issues. By learning from flight data, the data-driven approaches can handle the uncertainties which are inherently embedded in flight data. Also, by learning from data that embed the human's decision making process, the learned data-driven models can increase the rate of acceptance of the resolution methods by human, i.e., ATCs. The advanced data mining and machine learning techniques along with large-scale aviation data enable the application of data-driven approaches to conflict resolution. In [19], the data-driven models for supporting the decision making of ATCs in a sector have been developed using the data collected from a Human-In-The-Loop simulator. A resolution advisory in the horizontal dimension is generated by a model learned by the data based on Convolutional Neural Networks. In [20], a method has

been proposed based on Reinforcement Learning that can assign a new trajectory change point in the horizontal dimension to resolve aircraft conflict.

1.1.2 Conformance monitoring in terminal airspace

ATCs perform the task of conformance monitoring, i.e., the detection of deviations, by comparing the observed states (such as position and speed) of an aircraft (e.g., from surveillance radar) with its expected states based on the intent of the aircraft (following a flight plan or deviating from the flight plan due to the instructions of ATCs). There have been research interests in the topic of conformance monitoring.

In [21], an extensive literature survey on the existing decision support tools for conformance monitoring has been performed. The tools include Precision Runway Monitor (PRM) [22], Host Computer System (HCS) [23], and its recent versions, En Route Automation Modernization (ERAM) and Standard Terminal Automation Replacement System (STARTS), and User Request Evaluation Tool (URET) [24], all of which have been deployed in the United States. Conformance monitoring is performed by these tools based on the comparison between the observed and expected states of an aircraft to detect any excessive deviation, or non-conformance. A model-based fault detection technique has been proposed [25] for the investigation of the issues in these tools, based on which new techniques can be guided in their development. Among the several identified challenges, the uncertainties in trajectory deviations have been focused in the following methods: in [26], [27], hybrid estimation methods have been proposed to estimate the continuous states such as position and speed of an aircraft and the discrete states such as heading hold mode using a stochastic hybrid system model; in [28], [29], probabilistic approaches have been developed which computes the conformance probability; and in [30], another probabilistic approach has been proposed using adaptive time-series which can optimally represent the aircraft's time-varying deviation under uncertainty.

All the above methods can fall into physics-based approaches, in which a model describing the normal behaviors of a system is developed based on the system's dynamics or governing physics, in order to detect events that do not conform the model. Recently, aviation

data become rich due to the advances in sensing and data collection technologies, which enables the data-mining and modern machine learning techniques to be employed as a tool for analyzing air traffic operations. With respect to the conformance monitoring problem, data-driven approaches have been extensively investigated in aviation domain by identifying flight data that does not conform to normal data, called *anomaly*, which could lead to the degraded safety and efficiency of air traffic operations. Due to the inherent properties of the aviation data, that is, (i) the states of aircraft in air traffic operations (i.e., flights) keep changing along time and thus any collected datasets are recorded in the form of sequential or time-series data; and (ii) since the information about whether a flight is normal or abnormal is typically unavailable, most of the aviation data are unlabeled, which requires unsupervised learning techniques, the anomaly detection in aviation domain is typically tackled by unsupervised learning approaches for detecting anomalies in time-series data, which can be broadly categorized as follows [31]:

- Distance-based methods are characterized by using the notion of distance between two data points. A well-known method in this category is the k-Nearest Neighbors (kNN), which computes an anomaly score of a data point by computing the distance to its k-Nearest Neighbors, and if it is larger than some threshold, the data point is called anomaly [32]. Another method is clustering, which groups similar data instances together based on the distance or similarity. In [33], a clustering-based method is proposed to detect anomalies in Flight Operations Quality Assurance (FOQA) dataset during take-off and approach operations. The identified clusters detect various types of anomalies, such as energy excess/deficiency and abnormal pitch angle and flap settings. For the airport surface operations, a hierarchical clustering method [34] is used to group taxi paths in the spatio-temporal space. The detected anomalies are interpreted as the paths unplanned/unexpected by the controllers, which therefore implies a safety threat.
- Statistical methods are characterized by the use of the probability density estimated from the data. The assumption for anomaly detection is that normal and

abnormal data would reside in higher and lower probability regions, respectively. For the methods based on regression model, a model is fitted to the training data and the test data is fed into the trained model to compute the difference between the actual value and the value predicted from the model, called residual. If the residual is higher than some threshold, the test data is called anomaly. In [35][36], a Vector Auto-Regressive (VAR) model is used to represent each flight in a FOQA dataset and a residual is computed by applying the model learned from one flight to another flight. Another approach is based on Gaussian Mixture Models (GMM) which assumes that the data points are generated from the mixture of Gaussian distributions (or components) with different weights along the components. The GMM has been applied to instantaneous detection of anomalies during a specific flight phase [37].

- Domain-based methods try to find a domain (or its boundary) that separates normal and abnormal data. One-class Support Vector Machine (OCSVM) is a widely used method in this category, based on the assumption that the training data well represent normal data so that the learned domain would well define the normal region. Thus, if a test instance falls outside of the domain, then it is called anomaly. The separating boundary is defined in a feature space, which is obtained by applying kernel, or mapping, to the data in its original space. Based on OCSVM, the Multiple Kernel Anomaly Detection (MKAD) algorithm [38] is developed to find operationally significant anomalies from heterogeneous (both continuous and discrete) variables in data. The MKAD algorithm successfully detects important anomalies such as high airspeed, flights under gusty winds, go-around, and unusual approaches with high energy and under turbulence. OCSVM is also applied to general aviation [39] with energy features such as the specific total/potential/kinetic energies with their rates.
- Reconstruction-based methods transform/project the data in the input space into a lower dimensional space, which is then reconstructed by projecting into the original input space. It is assumed that anomalies will not be effectively

reconstructed using the training data, which mostly consists of normal instances. One of the most widely used method is Autoencoders, in which the input sequence is projected into a smaller number of neurons and then reconstructed as the output sequence whose number of elements is the same as the input sequence. In [40][41][42], the reconstruction error (the difference between input and output trajectories) is used as an anomaly score. From the distribution of the anomaly scores, higher scores correspond to weather impact, while lower ones correspond to usual intervention by the controllers, such as separation and sequencing.

- Temporal logic-based methods learn temporal logic expressions from the data. The results of the methods mentioned above are typically represented as hyperplanes in high-dimensional feature spaces to separate normal and abnormal data. This may lead to a higher accuracy but also to the lack of interpretability by domain experts. Due to the unsupervised nature of aviation data, the feedback from domain experts is crucial to improve the performance of unsupervised anomaly detection methods. In [43][44], a temporal logic-based anomaly detection algorithm (TempAD) is proposed to identify anomalous aircraft trajectories in terminal airspace. The algorithm is tested with air traffic surveillance data and is able to identify anomalies such as go-around, excessive total energy, and above or below the glideslope. In [45], TempAD is extended to an incremental learning version, which can keep adjusting the changes in air traffic operations on a daily basis. With the identified anomalies, a supervised learning method for precursor detection [46] is also proposed by identifying events that precede the occurrence of anomalies.

To enhance the situational awareness of ATCs, we propose a new conformance monitoring algorithm that computes the conformity score of the current state of an aircraft, as well as those of its future states. In predicting the future states of an aircraft, we develop a framework that combines a data-driven approach and a physics-based method, which generates a series of predicted track points that are stochastic. We then present a conformance monitoring algorithm for the current and future time based on a stochastic conformal prediction

method which can sequentially compute the conformity scores of such stochastic predicted track points.

1.2 Objectives and contributions

1.2.1 Conflict resolution in en-route airspace

In this thesis, we develop a framework for the generation of conflict resolution maneuvers in en-route airspace by learning from aviation datasets that contain the decision-making process of ATC. A decision-making process of ATC for conflict resolution can be viewed as a system that takes conflict situations as input and generates corresponding conflict resolution methods as outputs. That is, each conflict can be represented as a tuple of {Conflict Situation, Resolution Methods}. To construct a conflict data in this form from air traffic surveillance data, we first need to *label* each conflict situation, or identify resolution methods (outputs) used for the conflict situation. The key idea is that any complex resolution maneuvers can be represented as a sequence of simple motions, called *intents* in this paper. Using the domain knowledge obtained from flight data and the intent inference algorithm [47], we propose a framework for detection and characterization of aircraft resolution maneuvers to identify *resolution types* (e.g., directly heading to a downstream waypoint by skipping some next waypoints) and *resolution parameters* (e.g., how many next waypoints are skipped). Based on the knowledge extracted from the constructed conflict data with the *features* representing conflict situations (or inputs), we then design a classification model which determines the resolution type for every two-aircraft conflict in the airspace. In addition to predicting the resolution type, the proposed conflict resolution algorithm will also suggest resolution parameters for the guaranteed safety. The combination of the resolution type prediction model and resolution parameter suggestion model can safely resolve any two-aircraft conflict, and constitute the proposed data-driven resolution generator (D2RG) model. For a conflict situation previously unseen to the model, the learned model can predict the resolution type and suggest the corresponding resolution parameters for the guaranteed safety.

1.2.2 Conformance monitoring in terminal airspace

In this thesis, we develop a framework for conformance monitoring for the current and future time using a predicted aircraft trajectory. For the trajectory prediction, if a physics-based trajectory prediction method is used, since no measurements would be available for the future time-steps, the future states of an aircraft can be obtained by simply propagating the aircraft’s dynamics into the future without the correction by the measurements [48] or by utilizing the intent information, such as following its flight plan or deviating due to the instructions given by ATCs, if such intent can be inferred [47][49]. If there exist a large enough number of trajectories representing such intents in dataset, we can learn a data-driven trajectory prediction model that can represent sequential behaviors of an aircraft, e.g., RNN [50] or LSTM [51]. Using the output of the data-driven model as a pseudo-measurement, a physics-based estimation method, such as Kalman filter or its variants, can incorporate the aircraft’s current dynamics (e.g., flying with a constant velocity or performing a coordinated turn) with the expected states in the future, that is, the pseudo-measurements. With the predicted trajectory of an aircraft under monitoring, we compute the conformity score of the aircraft using a stochastic conformal prediction method. The conformal prediction [52] is a technique to provide a conformity score for a new data instance by computing how different the new one is with respect to the existing data instances, which is called Non-Conformity Measure (NCM). In our application, since the predicted trajectory is sequentially updated along time (i.e., the number of track points grows) and it is not *complete* in the sense that it does not reach its final point yet, the NCM is required to be applicable for such sequentially updated, incomplete trajectories. To address this issue, the authors of [53] used the directed Hausdorff Distance (DHD) [54] which can effectively capture the degree of how an incomplete trajectory resembles another complete trajectory. This method can handle deterministic trajectories only, however, the predicted track points of the aircraft under monitoring are stochastic. In this regard, we extend the deterministic version in [53] into a stochastic version using an approximation method and numerical integration for efficient computation. The proposed method, called Hybrid Data-driven and Physics-based Trajectory and Conformity

Prediction, is demonstrated with surveillance data from the repository of real historical air traffic surveillance datasets.

1.3 Outline of dissertation

This thesis is organized as follows: Chapter 2 presents a framework for detection and characterization of aircraft resolution maneuvers, followed by the development and test of the proposed data-driven resolution generator (D2RG). In Chapter 3, the framework for trajectory and conformity prediction using hybrid data-driven and physics-based approaches is described and demonstrated. Final concluding remarks and potential future research directions are presented in Chapter 4.

2. DATA-DRIVEN GENERATION OF CONFLICT RESOLUTION MANEUVERS IN EN ROUTE AIRSPACE

This chapter is organized as follows: in Section 2.1, we describe the framework for detection and characterization of aircraft resolution maneuvers, to construct conflict data. In Section 2.2, the Data-Driven Resolution Generator (D2RG) framework using the conflict data is presented. Each section consists of the details of each framework and the results of test and validation with flight data.

2.1 Intent-based Detection and Characterization of Aircraft Maneuvers in En Route Airspace

In this section, we propose a unified framework for the detection and characterization of aircraft’s resolution maneuvers from flight data which consists of the following tasks: (i) the time when an aircraft starts to deviate from its flight plan by taking a maneuver is first detected, (ii) the maneuver type taken by the aircraft is then identified, and then (iii) the maneuver is characterized based on how the aircraft performs the maneuver. The goal for developing the framework is to represent conflict situations in an appropriate form for data-mining, especially to identify the labels, or the resolution maneuvers, for a given conflict situation, as discussed in Sec. 2.2.

The framework shown in Figure 2.1 is proposed based on the underlying idea that any complex maneuvers of an aircraft can be represented as a sequence of simple motions, or called *intents* in this thesis), such as heading hold and heading change in the horizontal plane, from which we construct the maneuver models. To identify a sequence of intents of an aircraft from flight data, we first use a hybrid estimation method [55] in which the aircraft’s motion is modeled as a stochastic linear hybrid system with the continuous states (such as position and speed) and discrete states (or flight modes, such as heading hold and heading change). The estimated continuous and discrete states are then incorporated with flight plan to infer the aircraft’s intent by extending the intent inference algorithm [47]. The identified sequence of intents is then compared with the maneuver models to identify and characterize the maneuver taken by the aircraft.

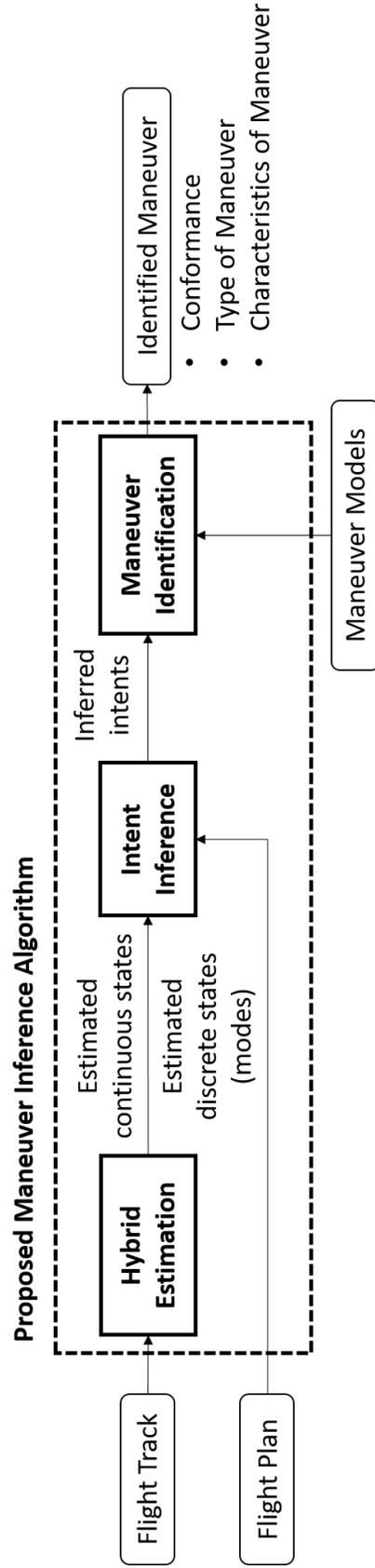


Figure 2.1. Proposed framework for the identification of aircraft maneuvers

2.1.1 Aircraft maneuver modeling

The operation in en route airspace is defined as the flight segment starting from the departure fix (DF) to the arrival fix (AF) [56] with the three phases: in the *climb* phase, an aircraft climbs from the DF to the Top of Climb (TOC); the aircraft then performs the *cruise* phase from the TOC to the Top of Descent (TOD); once reaching the TOD, the aircraft descends to the arrival fix (AF) in the *descent* phase. The aircraft performs the en route operation based on the flight plan that contains a series of waypoints in the horizontal plane, cruise altitude, and planned airspeed values. In this thesis, we define any deviations from the flight plan as *maneuvers*. Based on the domain knowledge [7], [8] and flight data analysis, we identify that air traffic controllers or automated algorithms use a finite number of maneuver types as shown in Table 2.1. Note that Table 2.1 contains the most commonly used ones in air traffic control and any other maneuver types that are not included in Table 2.1, such as the maneuver taken by unmanned aircraft, can be similarly modeled to be included in Table 2.1, if necessary. In this section, we construct the aircraft’s maneuver models by representing a maneuver as a sequence of intents in the horizontal, vertical, and speed dimensions, respectively.

Table 2.1. Maneuver Types

Dimension	Maneuver Types
Horizontal (\mathcal{M}_H)	Direct To (DT)
	Path Stretch (PS)
	Route Offset (RO)
Vertical (\mathcal{M}_V)	Temporary Altitude, Climb (TA-C)
	Step Altitude, Climb (SA-C)
	Step Altitude, Descent (SA-D)
	Temporary Altitude, Descent (TA-D)
Speed (\mathcal{M}_S)	Temporary Change in Cruise Speed (TC-CS)
	Change in Cruise Speed (C-CS)
	Change in Descent Speed (C-DS)

Horizontal Maneuvers

A flight plan in the horizontal dimension is given as a series of waypoints $\{WP_i\}_{i=0}^{N_{WP}}$ where $WP_i = (\xi_i, \eta_i)$ is the horizontal position (ξ_i is the longitude and η_i is the latitude) and N_{WP} is the number of waypoints. We define *current waypoint*, WP_{i_c} , as the waypoint that an aircraft is currently flying to and all the waypoints after WP_{i_c} are called *downstream waypoints*, WP_{i_d} where $i_d > i_c$. Once an aircraft reaches the current waypoint, the current waypoint's index i_c is updated as the next index, $i_c \leftarrow i_c + 1$. If an aircraft deviates from the flight plan by the command of air traffic controllers, there are three types of the horizontal maneuvers in the current operations, as shown in Figure 2.2.

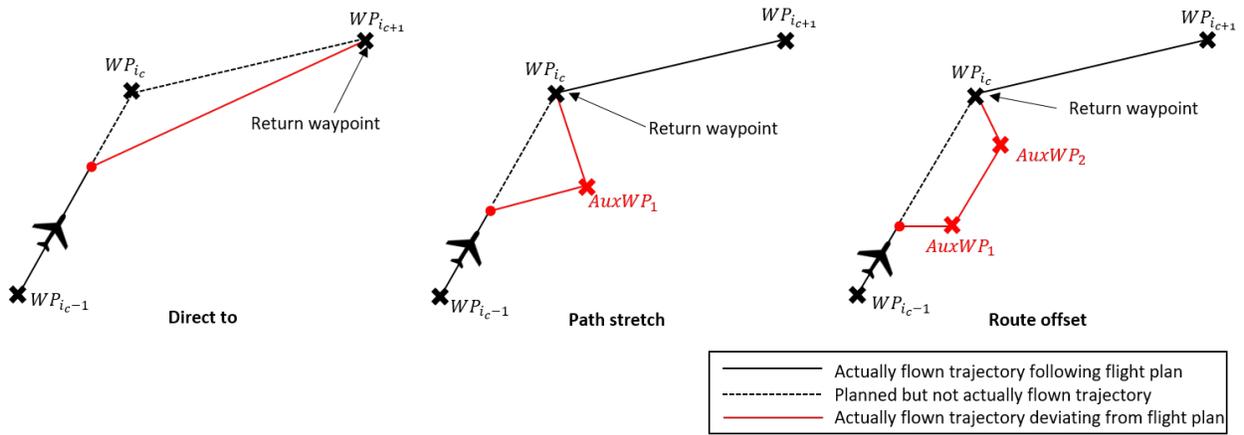


Figure 2.2. Maneuvers in the horizontal dimension

- *Direct To (DT)*: The current waypoint is skipped and then one of the downstream waypoints is selected to take a shortcut. Since the aircraft returns to its flight plan by reaching the selected downstream waypoint, it is called *return waypoint*.
- *Path Stretch (PS)*: An aircraft starts to deviate from the flight plan by changing its heading to a waypoint that was not included in the flight plan, or called *auxiliary waypoint*, $AuxWP_1$. Once reaching $AuxWP_1$, the aircraft is heading to one of the downstream waypoints (or return waypoint).
- *Route Offset (RO)*: With this maneuver type, an aircraft passes two auxiliary waypoints, $AuxWP_1$ and $AuxWP_2$, and then returns to one of the downstream

waypoints (or return waypoint). The two auxiliary waypoints form a line that is parallel to the planned path.

Vertical Maneuvers

A flight plan in the vertical dimension is given as a cruise altitude, h_c . If an aircraft is required to take a maneuver in the vertical dimension, there are four types of the vertical maneuvers in the current operations, as shown in Figure 2.3.

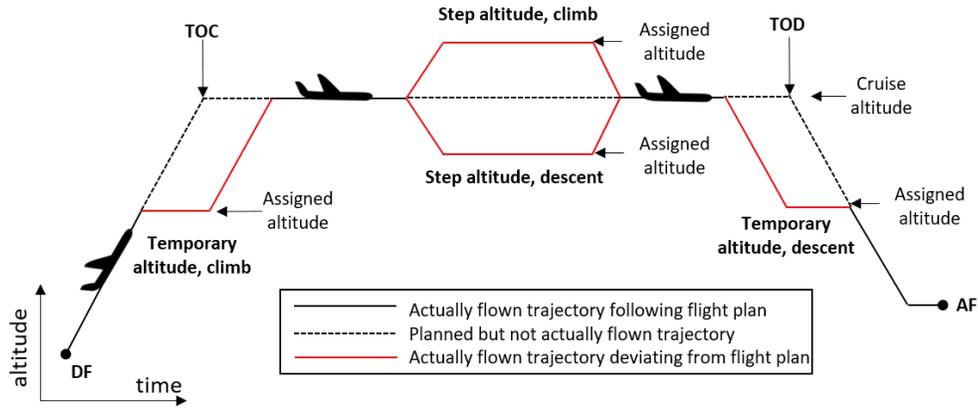


Figure 2.3. Maneuvers in the vertical dimension

- *Temporary Altitude, Climb (TA-C)*. An aircraft in the climb phase levels off at an altitude below h_c , called *assigned altitude*, for some period of time and then climbs to h_c .
- *Step Altitude, Climb or Descent (SA-C or SA-D)*. An aircraft in the cruise phase climbs or descends to an assigned altitude, stays there for some period of time, and then returns to h_c .
- *Temporary altitude, descent (TA-D)*. An aircraft in the cruise phase starts to descend before reaching TOD to an assigned altitude that is below h_c and above the altitude of AF. After staying there for some period of time, it then descends to the altitude of AF.

Speed Maneuvers

A flight plan in the speed dimension is given as planned airspeed values: climb airspeed (v_{CL}), cruise airspeed (v_{CR}), and descent airspeed (v_{DE}). If an aircraft is required to take a maneuver in the speed dimension, there are three types of the speed maneuvers in the current operations, as shown in Figure 2.4.

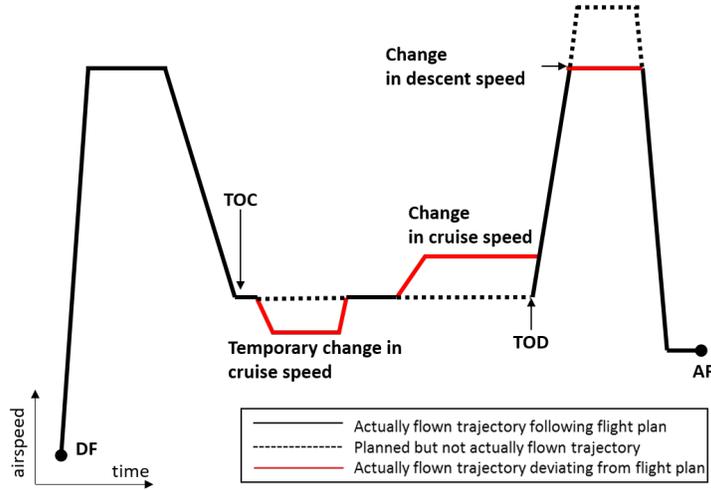


Figure 2.4. Maneuvers in the speed dimension

- *Temporary change in cruise speed (TC-CS)*. An aircraft in the cruise phase changes its airspeed to an assigned airspeed for some period of time.
- *Change in cruise speed (C-CS)*. An aircraft in the cruise phase changes its airspeed to an assigned airspeed and maintains it until reaching the TOD.
- *Change in descent speed (C-DS)*. An aircraft's planned airspeed in the descent phase is changed to an assigned airspeed.

Maneuver Models

As discussed above, the maneuvers can be modeled as a sequence of elementary motions or *intents*, such as *go to the current waypoint*, WP_{i_c} . The set of intents that constitute the maneuver types in Table 2.1 are presented in Table 2.2. The maneuvers are then modeled as

a sequence of the intents as shown in Table 2.3 where the flight segment that deviates from the flight plan is denoted within the brackets ($[\cdot]$).

2.1.2 Hybrid Estimation

To identify the sequence of intents of an aircraft, we use a hybrid estimation algorithm. The aircraft's motion is modeled as a stochastic linear hybrid system (SLHS) as follows: let $x(k)$ and $q(k)$ denote the continuous and discrete states at time step k , respectively. The SLHS model for an aircraft is represented as

$$x(k+1) = A_{q(k)}x(k) + E_{q(k)}w_{q(k)}(k) \quad (2.1)$$

$$z(k) = Cx(k) + v(k) \quad (2.2)$$

where x is the continuous state, $q \in Q$ is the discrete state (Q is a set of discrete states), and z is the measurement. The process noise w_q and the measurement noise v are assumed to be white Gaussian noise with zero mean and the covariances Q_q and R , respectively.

Horizontal Dynamics

The continuous state of an aircraft in the horizontal dimension is defined as

$$x = [\xi \quad \dot{\xi} \quad \ddot{\xi} \quad \eta \quad \dot{\eta} \quad \ddot{\eta}]^T \quad (2.3)$$

where ξ and η represent the longitude and latitude, respectively. The discrete state $q \in Q = \{1, 2\}$ is defined as $q = 1$ for heading hold mode and $q = 2$ for heading change mode. The measurement is defined as $z = [\xi \quad \eta]^T = Cx$ where

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (2.4)$$

With the sampling time T_s , the system matrices A_q and E_q are defined for each discrete mode $q \in Q = \{1, 2\}$ as follows:

Horizontal (M_H)	Vertical (M_V)	Speed (M_S)
H_i : go to WP_i ($i \in \{1, \dots, N_{WP}\}$)	V_{CL} : climb	S_{CL} : hold v_{CL}
$H_{t,c}$: turn at WP_{i_c}	V_{CR} : cruise at h_c	S_{CR} : hold v_{CR}
H_{a_1} : go to $AuxWP_1$	V_{DE} : descend	S_{DE} : hold v_{DE}
H_{a_2} : go to $AuxWP_2$	V_a : hold adjusted altitude	S_a : hold adjusted speed
$H_{t,nc}$: turn not at WP_{i_c}		$S_{c,1}$: change speed before TOD
		$S_{c,2}$: change speed after TOD

Table 2.2. Intent Sets

Maneuver	Sequence of Intents
Direct to (DT)	$\dots - H_{i_c} - [H_{t,nc} - H_{i_d}] - H_{t,c} - \dots$
Path stretch (PS)	$\dots - H_{i_c} - [H_{t,nc} - H_{a_1} - H_{t,nc} - H_{i_d}] - H_{t,c} - \dots$
Route offset (RO)	$\dots - H_{i_c} - [H_{t,nc} - H_{a_1} - H_{t,nc} - H_{a_2} - H_{t,nc} - H_{i_d}] - H_{t,c} - \dots$
Temporary altitude, climb (TA-C)	$V_{CL} - [V_a] - V_{CL} - \dots$
Step altitude, climb (SA-C)	$\dots - V_{CR} - [V_{CL} - V_a - V_{DE}] - V_{CR} - \dots$
Step altitude, descent (SA-D)	$\dots - V_{CR} - [V_{DE} - V_a - V_{CL}] - V_{CR} - \dots$
Temporary altitude, descent (TA-D)	$\dots - V_{CR} - [V_{DE} - V_a] - V_{DE}$
Temporary change in cruise speed (TC-CS)	$\dots - S_{CR} - [S_{c,1} - S_a - S_{c,1}] - S_{CR} - \dots$
Change in cruise speed (C-CS)	$\dots - S_{CR} - [S_{c,1} - S_a] - S_{c,2} - \dots$
Change in descent speed (C-DS)	$\dots - S_{c,2} - [S_a]$

Table 2.3. Maneuver Models

Heading hold mode ($q = 1$)

$$A_1 = \begin{bmatrix} 1 & T_s & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T_s & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad E_1 = \begin{bmatrix} \frac{T_s^2}{2} & 0 \\ T_s & 0 \\ 0 & 0 \\ 0 & \frac{T_s^2}{2} \\ 0 & T_s \\ 0 & 0 \end{bmatrix} \quad (2.5)$$

Heading change mode ($q = 2$)

$$A_2 = \begin{bmatrix} 1 & T_s & \frac{T_s^2}{2} & 0 & 0 & 0 \\ 0 & 1 & T_s & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T_s & \frac{T_s^2}{2} \\ 0 & 0 & 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad E_2 = \begin{bmatrix} \frac{T_s^2}{2} & 0 \\ T_s & 0 \\ 1 & 0 \\ 0 & \frac{T_s^2}{2} \\ 0 & T_s \\ 0 & 1 \end{bmatrix} \quad (2.6)$$

Vertical and Speed Dynamics

The continuous states in the vertical and speed dimensions are defined as $x = [h \quad \dot{h}]^T$ and $x = [v \quad \dot{v}]^T$, respectively, where \dot{h} is the altitude rate and \dot{v} is the airspeed rate. The discrete state $q \in Q = \{1, 2\}$ is defined as $q = 1$ for altitude/speed hold mode and $q = 2$ for altitude/speed change mode, respectively. The measurements are defined as $z = h = Cx$ and $z = v = Cx$, respectively, where

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad (2.7)$$

The system matrices A_q and E_q are defined for each discrete mode $q \in Q = \{1, 2\}$ as follows:

Altitude/Speed hold mode ($q = 1$)

$$A_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad E_1 = \begin{bmatrix} T_s \\ 0 \end{bmatrix} \quad (2.8)$$

Altitude/Speed change mode ($q = 2$)

$$A_2 = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix}, \quad E_2 = \begin{bmatrix} T_s \\ 1 \end{bmatrix} \quad (2.9)$$

Hybrid Estimation

The hybrid estimation algorithm uses a bank of Kalman filters matched with a discrete state (or mode), each of which estimates the continuous state $\hat{x}_i(k)$ and its corresponding error covariance $P_i(k)$ conditioned on mode $i \in Q$, at time step k . With the probability of mode i being correct at time k , denoted as $\alpha_i(k)$, the continuous and discrete states are estimated as:

$$\hat{x}(k) = \sum_{i \in Q} \hat{x}_i(k) \alpha_i(k) \quad (2.10)$$

$$P(k) = \sum_{i \in Q} \{P_i(k) + [\hat{x}_i(k) - \hat{x}(k)][\hat{x}_i(k) - \hat{x}(k)]^T\} \alpha_i(k) \quad (2.11)$$

$$\hat{q}(k) = \arg \max_{i \in Q} \alpha_i(k) \quad (2.12)$$

where $\hat{x}(k)$ is the estimated continuous state with its error covariance $P(k)$ and $\hat{q}(k)$ is the estimated discrete state.

Note that the mode probability $\alpha_j(k)$ is computed as

$$\alpha_j(k) = \frac{1}{c(k)} \Lambda_j(k) \sum_{i \in Q} \pi_{ij} \alpha_i(k-1) \quad (2.13)$$

where π_{ij} is the mode transition probability from mode i at time step $k - 1$ to mode j at time step k and $c(k)$ is a normalizing constant. The likelihood of mode j , $\Lambda_j(k)$, is given as

$$\Lambda_j(k) := \mathcal{N}(r_j(k); 0, S_j(k)) \quad (2.14)$$

where $r_j(k)$ is the residual, or the difference between the actual and estimated measurements, obtained by Kalman filter j with the corresponding covariance $S_j(k)$ and $\mathcal{N}(a; b, c)$ is the probability at a of a normal distribution with mean b and covariance c .

Since the discrete state estimates play a critical role in the inference of the aircraft's intents as discussed in Section 2.1.3, it is desirable to reduce false estimate of the discrete states. In this regard, we propose to use an algorithm called Residual-Mean Interacting Multiple Model (RMIMM) [55]: if $\alpha_j(k)$ is large, i.e., mode j is highly likely to be the correct mode, then the corresponding residual mean $\bar{r}_j(k) := \mathbf{E}[r_j(k)]$ has a small value. To increase the difference of the likelihoods between the correct mode and the other modes, the inverse of the residual mean is used as a weight, that is,

$$\Lambda_j^{new}(k) = \begin{cases} \frac{N_j(k)\Lambda_j(k)}{\sum_{i=1}^N N_i(k)\Lambda_i(k)} & \text{if } \bar{r}_j(k) \neq 0 \\ \Lambda_j(k) & \text{otherwise} \end{cases} \quad (2.15)$$

where $N_i(k) = \|\bar{r}_i(k)\|^{-1}$ if $\|\bar{r}_i(k)\| \neq 0$; $N_i(k) = 1$, otherwise. The more distinct mode probabilities are demonstrated in Figure 2.14.

2.1.3 Intent Inference

With the estimated continuous and discrete states of an aircraft along with the flight plan, we present how the aircraft's intents can be inferred in this section.

Horizontal Intents

From Table 2.2, we can observe that the horizontal intents of *go to a waypoint* correspond to the heading hold mode, $q = 1$, and the horizontal intents of *turn* to the heading change mode ($q = 2$).

If the aircraft's discrete state at time step k in the horizontal dimension is estimated as $\hat{q}(k) = 2$, the distance between the current waypoint and the aircraft's current estimated position is computed: WP_{i_c} , $d(k) = \left\| \left(\hat{\xi}(k), \hat{\eta}(k) \right) - WP_{i_c} \right\|$: if $d(k) \leq \delta$ for a distance threshold δ which is a design parameter, then we infer the horizontal intent as *turn at the current waypoint*; otherwise, *turn not at the current waypoint*.

If the aircraft's discrete state at time step k in the horizontal dimension is estimated as $\hat{q}(k) = 1$, we need to identify the corresponding waypoint, either the known one (included in the flight plan) or the unknown one (auxiliary waypoint). First, the *intent likelihood* $\lambda_i(k)$ [47] for the known waypoint is computed as

$$\lambda_i(k) = \mathcal{N} \left(\psi_i(k); \hat{\psi}(k), \sigma_\psi^2 \right) \quad \text{for } i \in \{i_c, i_c + 1, \dots, N_{WP}\} \quad (2.16)$$

where $\psi_i(k)$ is the heading angle of a unit vector $\hat{e}_i(k)$ from the aircraft's current estimated position to the corresponding waypoint which represents intent H_i for $i \in \{i_c, \dots, N_{WP}\}$, $\hat{\psi}(k)$ is the aircraft's current estimated heading which represents the aircraft's current intent represented as the heading of a unit vector, $\hat{e}_\psi(k)$, as shown in Figure 2.5, and the standard deviation σ_ψ is a design parameter.

From Eq. (2.16), we can observe that an aircraft is flying to a known waypoint, WP_I for some $I \in \{i_c, \dots, N_{WP}\}$ for time steps $k - 1$ and k , then the corresponding intent likelihood will have the maximum value, i.e.,

$$\lambda_I(k) \simeq \lambda_I(k - 1) \simeq \frac{1}{\sqrt{2\pi}\sigma_\psi} \quad (2.17)$$

Hence, either of $\lambda_i(k) \not\approx \lambda_i(k - 1)$ or $\lambda_i(k) \ll \frac{1}{\sqrt{2\pi}\sigma_\psi}$ holds for all $i \in \{i_c, \dots, N_{WP}\}$, the aircraft's horizontal intent is inferred as *go to AuxWP₁*. Furthermore, if the aircraft's previous sequence contains *go to AuxWP₁* and *turn not at the current waypoint* and the

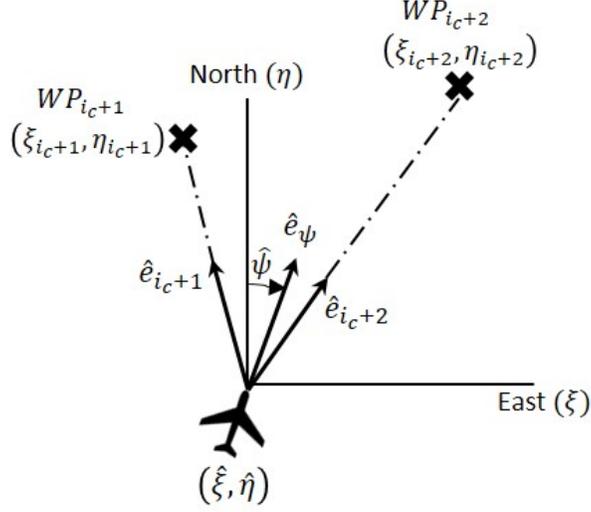


Figure 2.5. Intent inference

heading is parallel to the original planned path, the aircraft's horizontal intent is inferred as *go to AuxWP₂*.

If Eq. (2.17) holds, then the intent of the maximum likelihood is likely to represent the current motion of an aircraft. However, the intent inferred this way could be false if two known waypoints are closely collinear with the aircraft's position [47]. In this regard, the intent likelihood is modified by considering the likelihood in terms of time-to-go (TTG). With the given waypoints and the cruise ground speed, $v_{CR,g}$, the TTG to WP_i for $i \in \{i_c + 1, \dots, N_{WP}\}$ is obtained as

$$\text{TTG}_i(k) = \frac{1}{v_{CR,g}} \left[\left\| (\xi_{i_c}, \eta_{i_c}) - (\hat{\xi}(k), \hat{\eta}(k)) \right\| + \sum_{n=i_c+1}^{N_{WP}} \left\| (\xi_n, \eta_n) - (\xi_{n-1}, \eta_{n-1}) \right\| \right] \quad (2.18)$$

and for $i = i_c$, the TTG is obtained similarly, without the last term. The temporal likelihood is then given as

$$\tau_i(k) = \mathcal{N}(\text{TTG}_i(k); 0, \sigma_\tau^2) \quad (2.19)$$

where σ_τ is a design parameter. The aircraft's intent is finally inferred as $\hat{I}_H(k) = H_{\hat{i}(k)}$ by solving

$$\hat{i}(k) = \arg \max_{i \in \{i_c, \dots, N_{WP}\}} \lambda_i(k) \tau_i(k) \quad (2.20)$$

Vertical Intents

The vertical intents of *cruise at h_c* and *hold an adjusted altitude* correspond to the altitude hold mode ($q = 1$) and the vertical intents of *climb* and *descend* correspond to the altitude change mode ($q = 2$).

If $\hat{q}(k) = 1$, the aircraft's vertical intent is inferred as *cruise at h_c* if the estimated altitude of the aircraft is the same as the cruise altitude, h_c ; otherwise, *hold an adjusted altitude*.

If $\hat{q}(k) = 2$, the aircraft's vertical intent is inferred as *climb* if $\hat{h}(k) > 0$; otherwise, *descend*.

Speed Intents

The speed intents of *hold* correspond to the speed hold mode ($q = 1$), and the speed intents of *change* correspond to the speed change mode ($q = 2$).

If $\hat{q}(k) = 1$, we use the airspeed estimate, $\hat{v}(k)$: if $\hat{v}(k) = v_{CL}$, then $\hat{I}_S(k) = S_{CL}$; if $\hat{v}(k) = v_{CR}$, then $\hat{I}_S(k) = S_{CR}$; if $\hat{v}(k) = v_{DE}$, then $\hat{I}_S(k) = S_{DE}$; otherwise, $\hat{I}_S(k) = S_a$.

If $\hat{q}(k) = 2$ and $k \leq k_{TOD}$, then the aircraft's speed intent is inferred as *change speed before TOD*; otherwise, *change speed after TOD*. The time step for TOD, k_{TOD} , can be computed by using the estimated states in the vertical dimension as the point where the vertical intent changes from *cruise at h_c* to *descent* within 20 minutes or 200 nautical miles from the AF [8].

2.1.4 Maneuver Identification

We use the inferred intents to identify the maneuver, in which the time of starting deviation, the type of maneuver, and the characteristics of the maneuver in Table 2.4.

Identification of Horizontal Maneuvers

The horizontal maneuvers are modeled as a sequence of the inferred horizontal intents based on the maneuver model in Table 2.3, as shown in Figure 2.6.

Maneuver	Characteristics
DT	return WP
PS	AuxWP1, return WP
RO	AuxWP1, AuxWP2, returnWP
TA-C	adjusted altitude, duration
SA-C	adjusted altitude, duration
SA-D	adjusted altitude, duration
TA-D	adjusted altitude
TC-CS	adjusted cruise airspeed, duration
C-CS	adjusted cruise airspeed
C-DS	adjusted descent airspeed

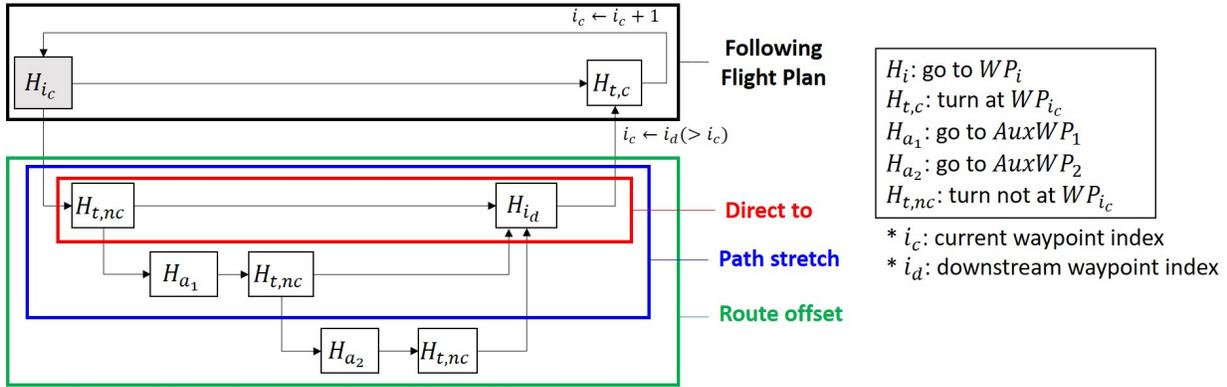


Figure 2.6. Maneuver models (horizontal)

If an aircraft is heading to the current waypoint and takes a turn before reaching the current waypoint, the aircraft is either (i) taking a shortcut to a downstream waypoint or (ii) changing its heading to an unknown waypoint. For the case of (i), the aircraft’s maneuver is determined as *Direct to*. For the case of (ii), the aircraft’s maneuver can be either *Path stretch* or *Route offset*, as shown in (a) in Figure 2.7. The maneuver can be uniquely determined after the aircraft’s intent is inferred as *turn not at the current waypoint*: if the aircraft’s following intent is inferred as *go to a downstream waypoint*, then the maneuver is inferred as *Path stretch*; otherwise, *Route offset*, as shown in (c) in Figure 2.7. The auxiliary waypoints for *Path stretch* and *Route offset* can be determined as the aircraft’s estimated position where the intent is inferred as *turn not at the current waypoint*, as shown in (b) in Figure 2.7.

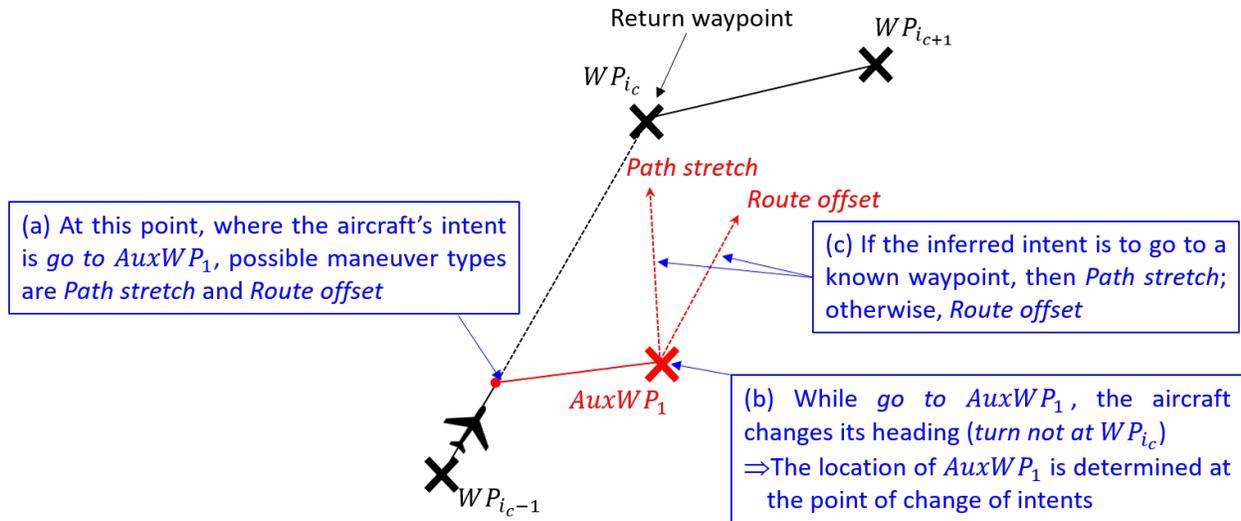


Figure 2.7. PS and RO

Identification of Vertical Maneuvers

The vertical maneuvers are modeled as a sequence of the inferred vertical intents based on the maneuver model in Table 2.3, as shown in Figure 2.8.

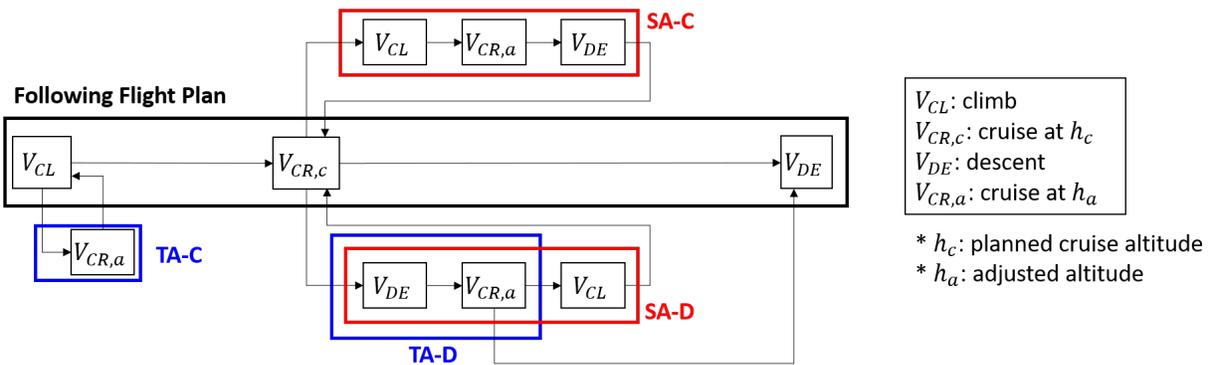


Figure 2.8. Maneuver models (vertical)

If an aircraft's intent was *cruise at h_c* and it changes to *descend*, the aircraft's maneuver can be either *SA-D* or *TA-D* as shown in Figure 2.9. The maneuver type can be uniquely determined once the intent *hold an adjusted altitude* is completed: if the following intent is inferred as *climb*, then the maneuver type is inferred as *SA-D*; if *descend*, then *TA-D*.

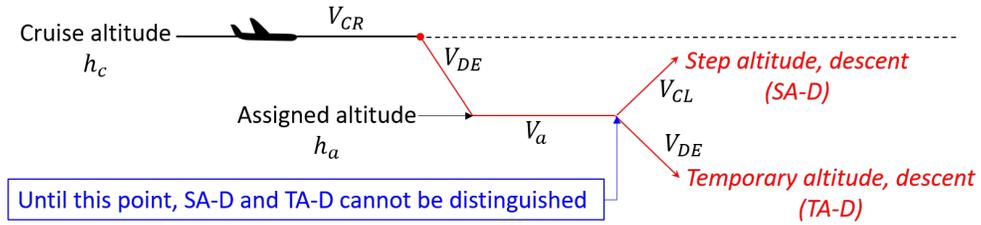


Figure 2.9. SA-D and TA-D

Identification of Speed Maneuvers

The speed maneuvers are modeled as a sequence of the inferred speed intents based on the maneuver model in Table 2.3, as shown in Figure 2.10.

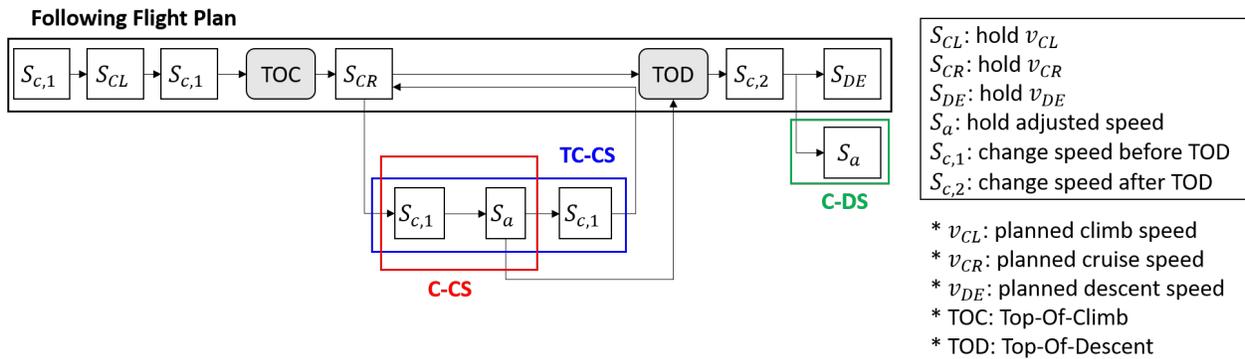


Figure 2.10. Maneuver models (speed)

Similar to *SA-D* and *TA-D* in the vertical dimension, *TC-CS* and *C-CS* can be distinguished once the intent *hold an adjusted speed* is completed, as shown in Figure 2.11.

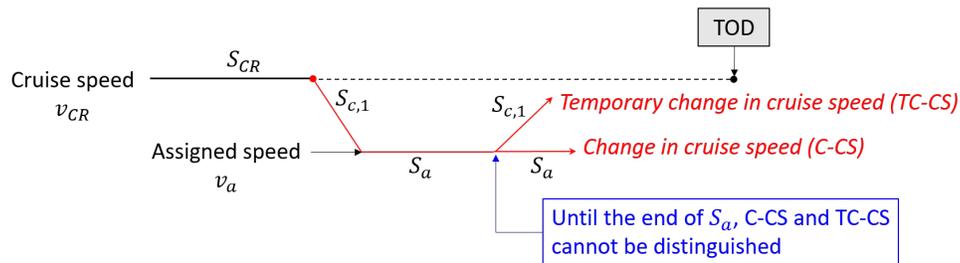


Figure 2.11. C-CS and TC-CS

2.1.5 Demonstration with Surveillance Data

The proposed framework is tested and demonstrated with the surveillance data. We use the flight data generated by Airspace Concept Evaluation System (ACES) [57], which is built for the simulation and evaluation of advanced air traffic control concepts [8]. In ACES, the aircraft’s maneuvers are generated by its internal model called *autoresolver* [7], [8]. The simulated data contains the ground truth about the maneuver’s type and characteristics, so that we can readily validate the proposed method. The proposed method is tested with ACES data that contains 9,286 flights where the total of 3,042 maneuvers are recorded. The type and characteristics of maneuvers identified by the proposed method are compared with the ones recorded in ACES data. As shown in Figure 2.12, the proposed method correctly identifies all of 3,042 maneuvers.

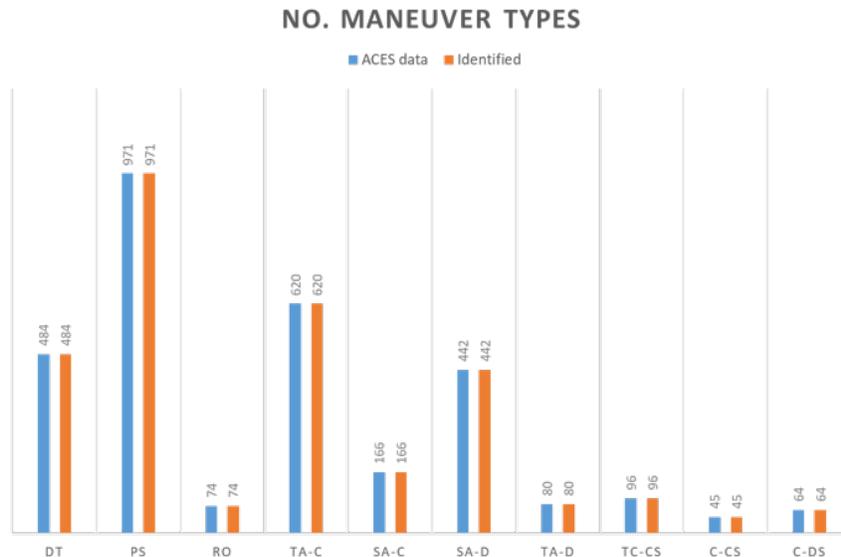


Figure 2.12. Overall performance results

As an illustrative example to show how the proposed algorithm works, consider an aircraft’s flight plan and flight track shown in Figure 2.13.

1. *Hybrid estimation*: By taking the flight track as input, we obtain the continuous state estimates, the mode probabilities, and the discrete state (or mode) estimates, as shown in Figure 2.14 (a).

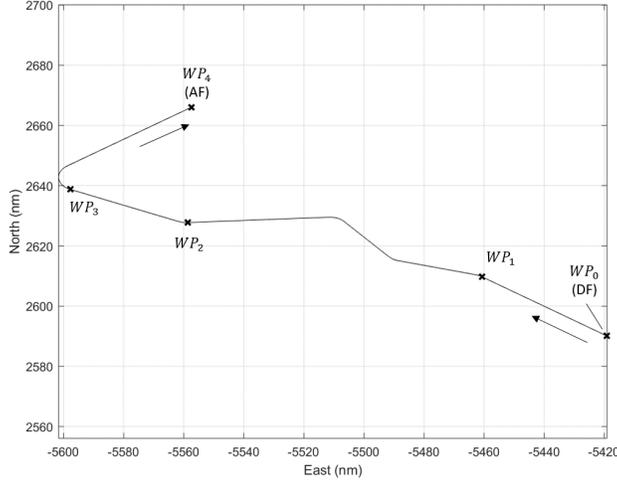
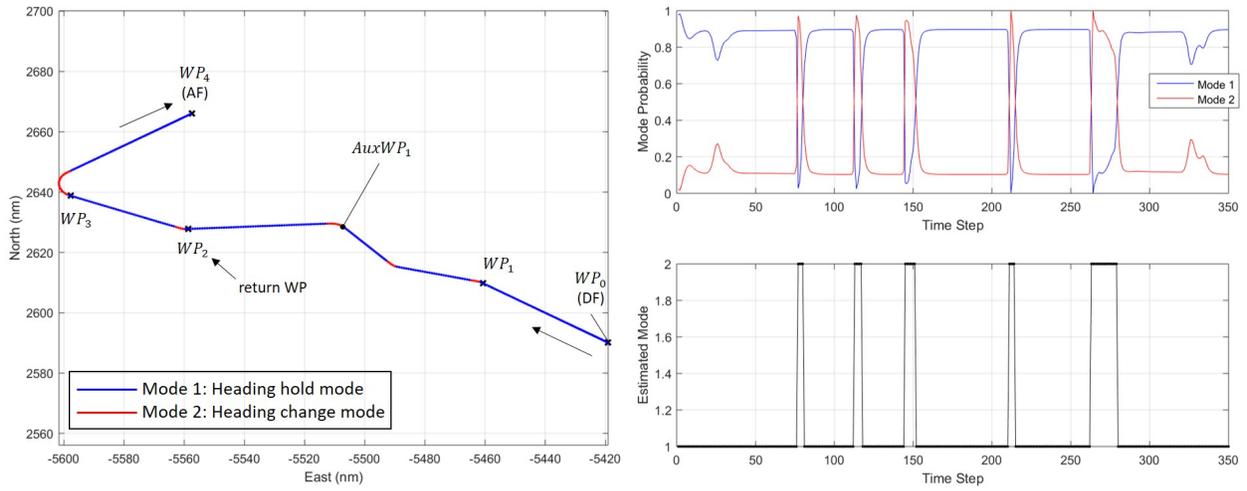


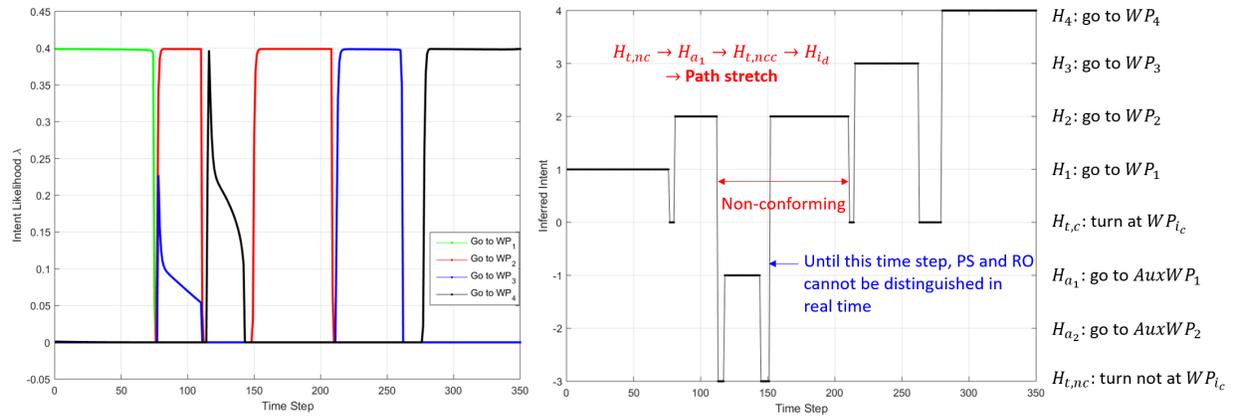
Figure 2.13. Illustrative case: Flight plan and track

2. *Intent inference:* By taking the estimated hybrid states along with the flight plan as input, we obtain the intent likelihoods of the waypoints in the flight plan as shown in the left of Figure 2.14 (b). Whenever the aircraft is heading to one of the waypoints in the flight plan, the corresponding likelihood has the constant, maximum value. For the time-steps between 114 and 143, however, the likelihoods for the waypoints in the flight plan are neither constant nor maximum, thereby the intents during this period being inferred as *go to AuxWP₁*.
3. *Maneuver identification:* From the intent likelihoods, the intents are inferred as shown in the right of Figure 2.14 (b), from which *Path stretch* is identified as the maneuver type with its characteristics: the auxiliary waypoint and the return waypoint in the left of Figure 2.14 (a).

In what follows, the results of applying the proposed algorithm to the rest of all the maneuver types are presented, as shown in Figures 2.15, 2.16, and 2.17 for the horizontal, vertical, and speed maneuver types, respectively.

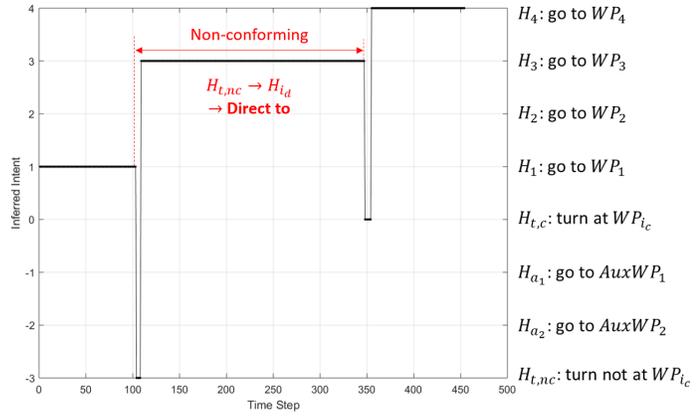
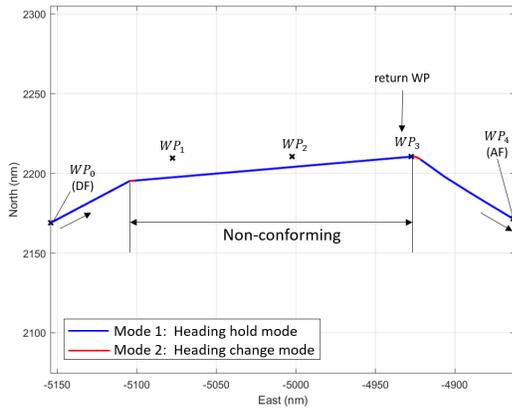


(a) Hybrid estimation

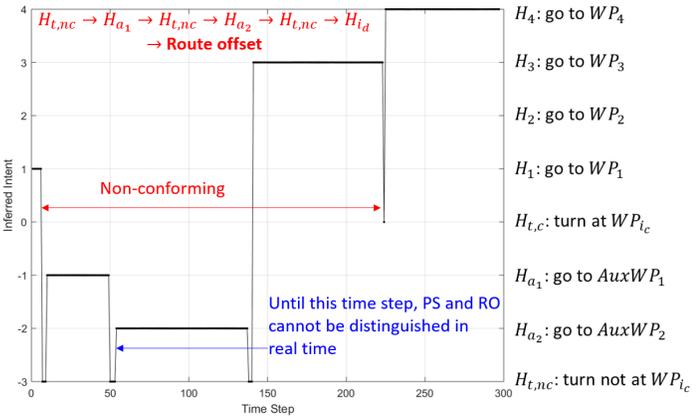
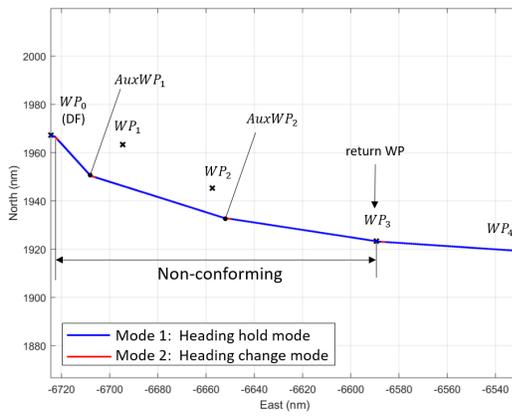


(b) Intent inference (left) and maneuver identification (right)

Figure 2.14. Illustrative case: Results of the proposed algorithm

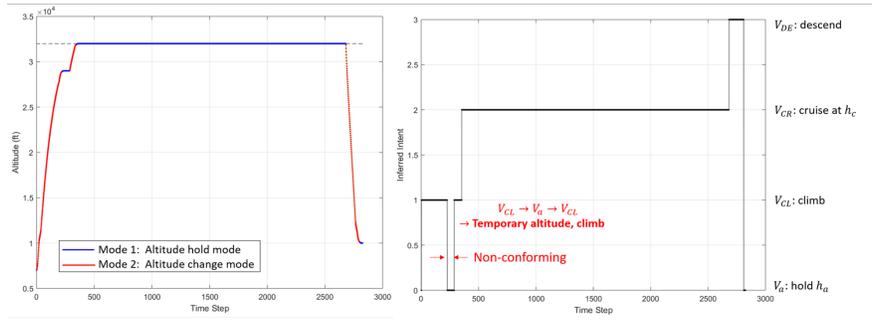


(a) Direct to

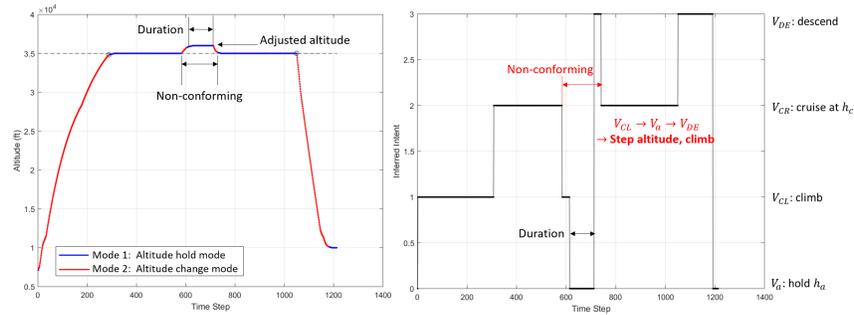


(b) Route offset

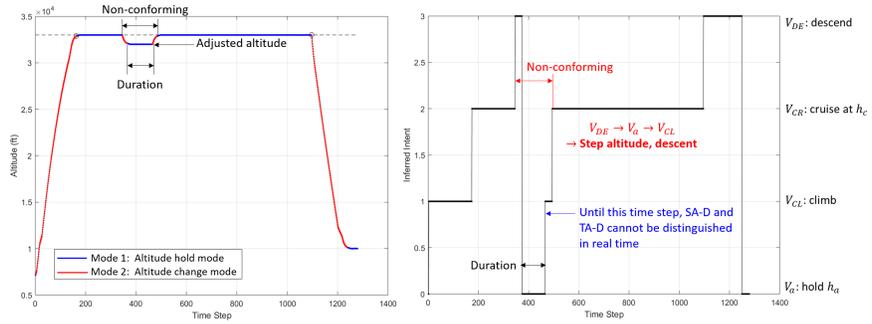
Figure 2.15. Results of the proposed algorithm for the maneuver types in the horizontal dimension



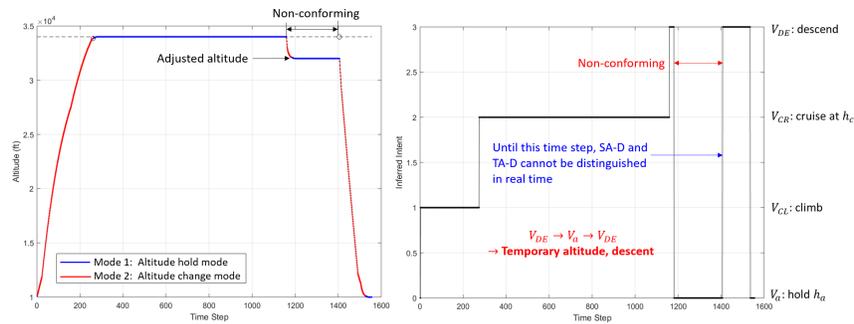
(a) Temporary altitude, climb



(b) Step altitude, climb

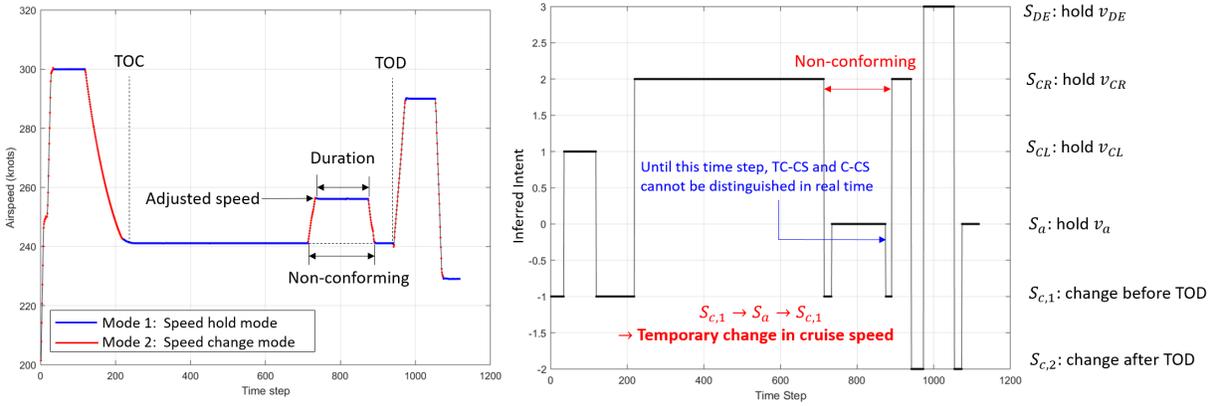


(c) Step altitude, descent

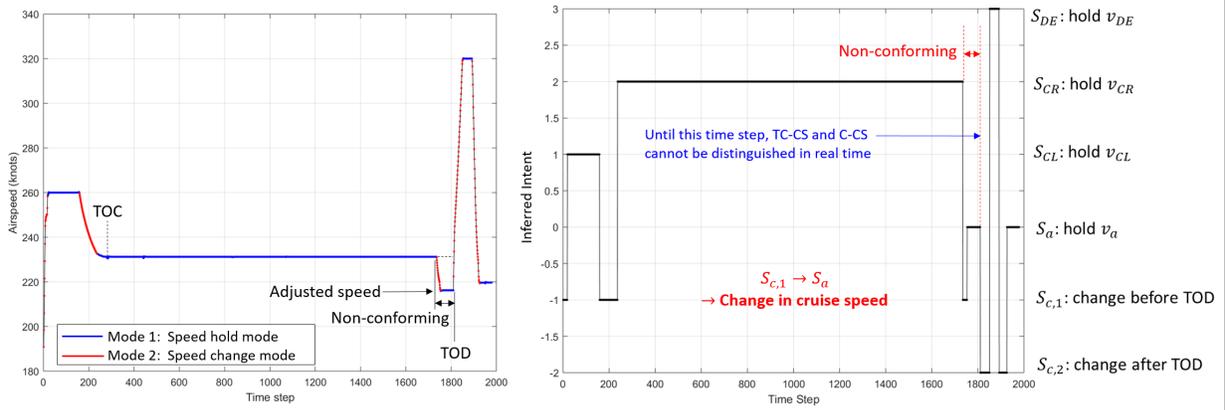


(d) Temporary altitude, descent

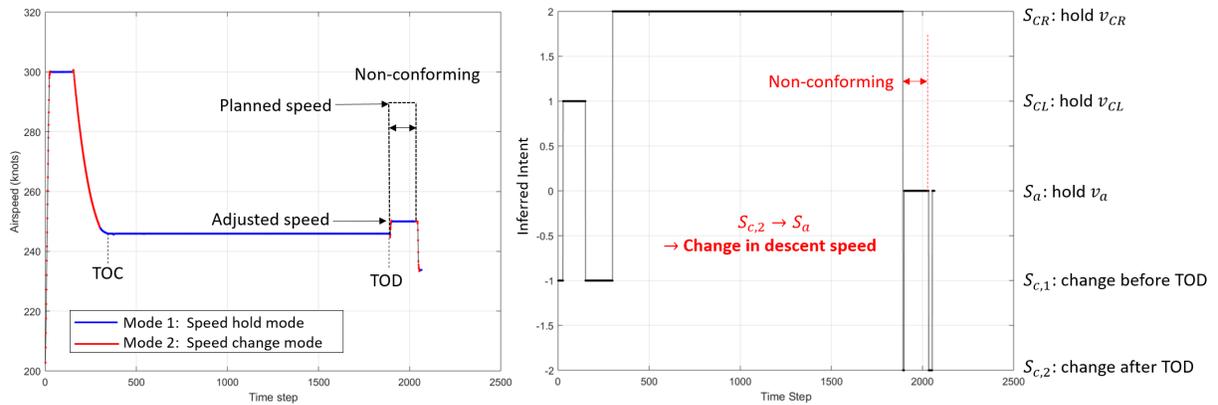
Figure 2.16. Results of the proposed algorithm for the maneuver types in the vertical dimension



(a) Temporary change in cruise speed



(b) Change in cruise speed



(c) Change in descent speed

Figure 2.17. Results of the proposed algorithm for the maneuver types in the speed dimension

2.2 Data-Driven Conflict Resolution Generator based on Supervised Learning

From the current standard by Federal Aviation Administration (FAA) [56], a conflict between two aircraft is defined as the violation of the separation standard, which is in en-route airspace 5 nm in the horizontal and 2,000 ft (above 29,000 ft) or 1,000 ft (below 29,000 ft) in the vertical. A conflict is resolved when the separation between two aircraft satisfies the standard by taking a maneuver, called *resolution*.

In this section, an algorithm for generating a resolution by learning from flight data, called *Data-driven Resolution Generator (D2RG)*, whose framework is shown in Figure 2.18 which consists of (i) learning a D2RG model from flight data and (ii) applying the learned D2RG model to a conflict situation.

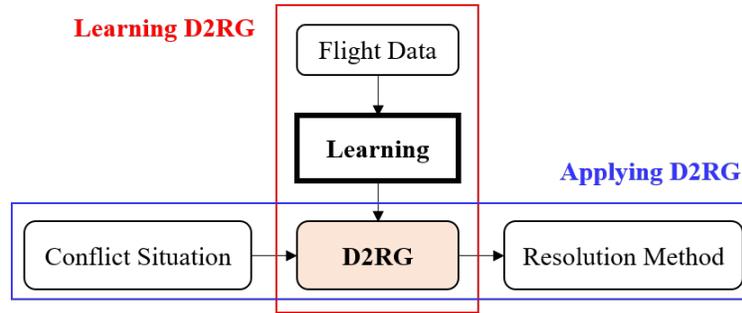


Figure 2.18. Proposed framework: Data-Driven Resolution Generator (D2RG)

From the discussion in Section 2.1, a resolution maneuver is defined by its type and the corresponding characteristics, which are called *resolution type (RT)* and *resolution parameter (RP)*, respectively, in this section. In this regard, the learning of a D2RG model from flight data is performed as follows:

- Step 1: *Construction of dataset*. From flight data, a conflict dataset is constructed (Section 2.2.1).
- Step 2: *Hierarchical classification of RTs*. Models for generating RTs are learned from the conflict dataset by using supervised learning techniques (Section 2.2.2).

- Step 3: *Parameter learning for RPs*. For each RT, a model for generating RP is learned from the conflict dataset by using supervised learning techniques with the guaranteed separation assurance (Section 2.2.3).

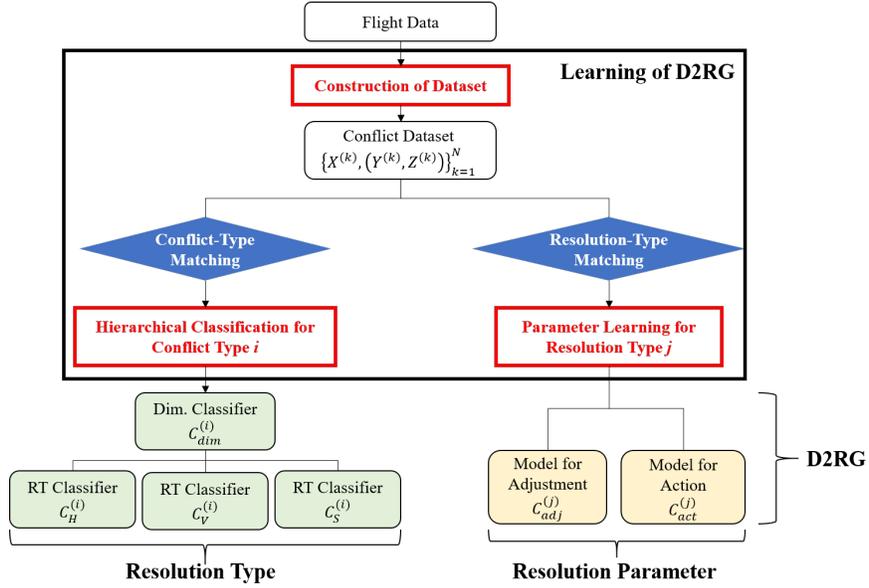


Figure 2.19. Learning of D2RG

The application of the learned D2RG model to a conflict situation is presented in Section 2.2.4

2.2.1 Construction of Dataset

Considering that a conflict situation and the corresponding resolution method can be viewed as an input-output relation, we construct a conflict dataset as a labeled data with input (or a feature vector) and output (or a label), which naturally leads to the use of supervised learning techniques. For N number of conflict situations in flight data, the conflict dataset is denoted as, for

$$\{F^{(k)}, L^{(k)}\}_{k=1}^N \quad (2.21)$$

where, for k -th conflict situation, $F^{(k)}$ is a feature vector (conflict situation) and $L^{(k)}$ is a label (resolution method).

Features

To construct a feature vector that represent a conflict situation, we utilize the domain knowledge about how conflicts are resolved by air traffic controllers [58] and the *autore-solver* [7], [8] which is an embedded tool for conflict resolution in the ACES. The features in the constructed feature vector can be categorized into four types of information that represent a conflict situation, as shown in Figure 2.20 and Table 2.5:

- *Primary conflict*: The distance, relative speed and heading between two aircraft involved in a given conflict situation are used.
- *Flight plan*: The flight plans of the two conflict aircraft are used.
- *(Potential) secondary conflict*: A secondary conflict happens when either of the two conflicting aircraft that takes a resolution maneuver interrupts the other aircraft's path, which should be avoided when generating a resolution. The information related to the neighboring aircraft around the conflicting aircraft is used.
- *Airspace structure*: Air traffic controller's decisions are affected by the structure of an airspace, and hence the related information is used.

Labels

A resolution method, or its type (RT) and the corresponding parameter (RP), can be identified by using the method presented in Section 2.1. For a given conflict situation k , let the identified RT be denoted as $Z^{(k)}$ and its corresponding dimension (the horizontal, vertical, or speed) as $Y^{(k)}$, and then the label in Eq. (2.21) is constructed as $L^{(k)} = (Y^{(k)}, Z^{(k)})$.

By preprocessing (such as feature scaling), the constructed feature vector, $F^{(k)} = X^{(k)}$, is combined with the label so that we now construct a conflict dataset $\{X^{(k)}, (Y^{(k)}, Z^{(k)})\}_{k=1}^N$.

Table 2.5. Feature vector

Type	No.	Description
Primary conflict	1	Distance between two conflicting aircraft in the horizontal dimension
	2	Distance between two conflicting aircraft in the vertical dimension
	3	Relative bearing between two conflicting aircraft
	4, 5	Respective altitudes of two conflicting aircraft
	6, 7	Respective vertical velocities of two conflicting aircraft
	8	Relative vertical speed between two conflicting aircraft
	9, 10	Respective ground speeds of two conflicting aircraft
	11	Time between resolution-issued time and time of predicted conflict
Predicted conflict	12, 13	Respective times to reach the arrival fix for two conflicting aircraft
	14, 15	Ratios of time of resolution issued to total travel time for two conflicting aircraft
	16, 17	Ratios of time of predicted conflict to total travel time for two conflicting aircraft
Secondary conflict	18	Number of neighboring aircraft around two conflicting aircraft
Airspace structure	19	Distance to the nearest boundary of airspace for two conflicting aircraft
Flight plan	20	Waypoints shared by two conflicting aircraft
	21	Feasibility of <i>Direct-to</i>

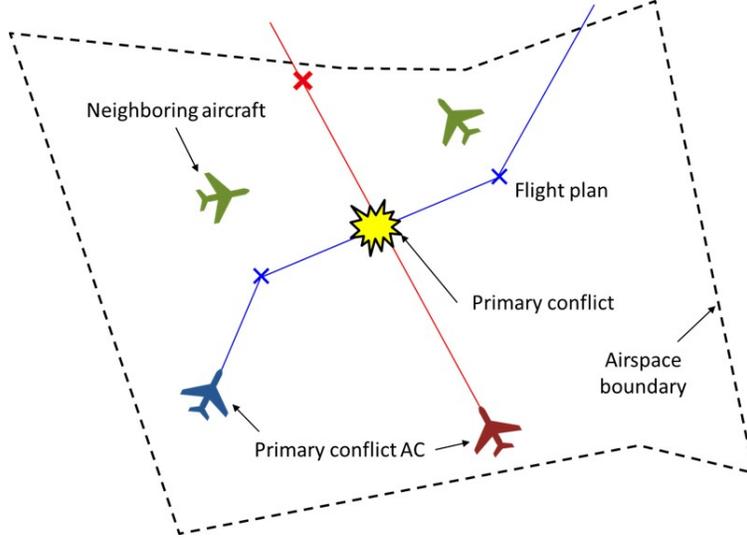


Figure 2.20. Conflict situations

2.2.2 Resolution Type: Hierarchical Classification

By analyzing the flight data, we find that the importance of the features (input) and the distribution of the labels (output) are dependent on the flight phases (climb, cruise, and descent) of two conflicting aircraft at the time of the predicted conflict, or called *conflict type*. In this regard, we first group the conflict situations based on the conflict type, which consists of nine cases, $\mathcal{M}_{CT} = \{CL/CL, CL/CR, CL/DE, CR/CL, CR/CR, CR/DE, DE/CL, DE/CR, DE/DE\}$ where *CL* is climb, *CR* is cruise, *DE* is descent, and the first and second elements in each type correspond to the flight phases of a maneuvering aircraft and a non-maneuvering aircraft, respectively.

With the conflict type-matched conflict data, we propose a framework based on the hierarchical classification-based supervised learning, as shown in Figure 2.21.

Through flight data analysis and the literature review [59], we find that there exists the flexibility in ATC's conflict resolution by choosing any of the maneuvering dimension among the horizontal, vertical, and speed and this flexibility is preferred by ATCs [8]. In this regard, we propose the hierarchical framework where the likelihood of a maneuvering dimension is computed by the upper-level classifier $C_{dim}^{(i)}$ with the dataset $\{X^{(k)}, Y^{(k)}\}_{k=1}^N$ and for each

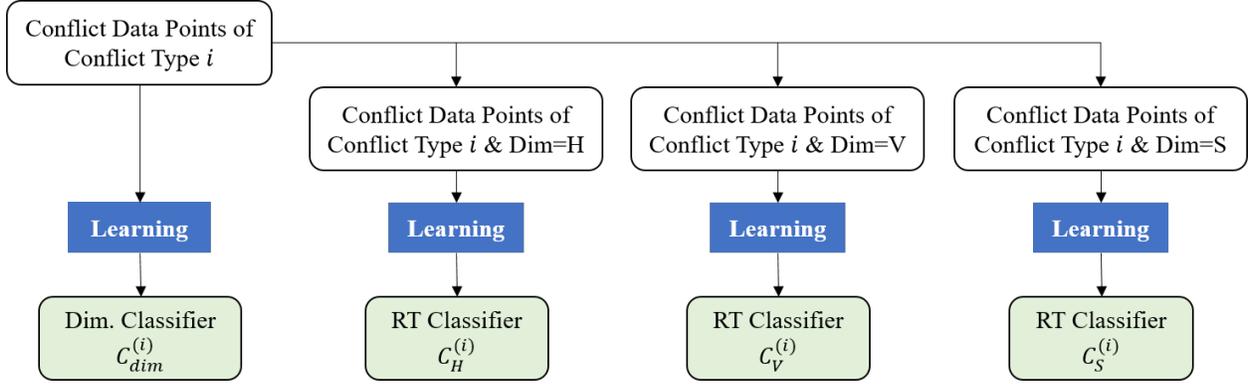


Figure 2.21. RT: Hierarchical classification

maneuvering dimension, the likelihoods of RTs are then provided by the lower-level classifier $\{X^{(k)}, Z^{(k)}\}_{k=1}^N$, and hence we train or learn 4 classifiers for each conflict type.

Any technique for multi-class classification can be used for learning, but considering the highly nonlinear nature of the conflict data and the importance of the features varying along the conflict types, we propose to use the feature-weighted support vector machine (FWSVM) [60], [61]. The important of each feature, or feature weight, is embedded in learning the classifier, i.e., the feature weights and the classifier (SVM) are simultaneously learned.

2.2.3 Resolution Parameter Learning

A resolution trajectory can be viewed as a two-step procedure: (i) the deviation from the planned path by adjusting the heading, altitude, or speed and then (ii) maintaining the deviated path until the conflict is safely resolved, followed by returning to the original planned path. In this regard, for a given RT, a data-driven model for the first step, called *adjustment*, $C_{adj}^{(j)}$, is learned to determine the amount of deviations and a data-driven model for the second step, called *action*, $C_{act}^{(j)}$, is learned to determine a decision whether holding the deviated path or returning to the original planned path, as shown in Figure 2.22.

We illustrate with *Path stretch* as an example how to learn the two models.

Adjustment: In the learning of the adjustment parameter, we apply a data-driven method with a check for the guaranteed safety in an iterative manner so that the resultant deviation

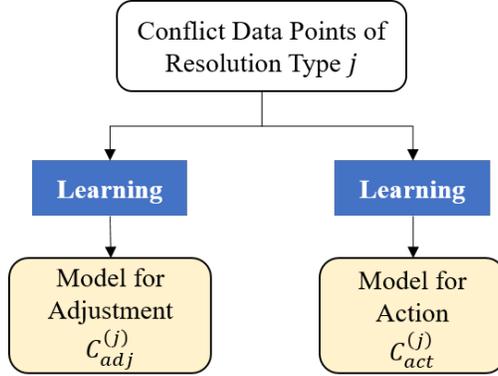


Figure 2.22. Learning of RPs

of the maneuvering aircraft makes it pass the outside of the non-maneuvering aircraft’s protected zone. We assume that the non-maneuvering aircraft does not change its path during the maneuvering aircraft’s resolution process. If the non-maneuvering aircraft deviates from its path during the process, which leads to a secondary conflict, we regard this as a new conflict and subsequently resolve it.

A data-driven method is used to learn the most likely value of the adjustment from the data, for which a dataset (features and labels) is constructed for each RT, as follows:

- Features: Along with the features in Table 2.5, we use (i) the maneuvering aircraft’s distance to the non-maneuvering aircraft’s protected zone when the resolution is issued and (ii) the remaining distance to the downstream waypoint that is closest to the maneuvering aircraft.
- Label: The label is given as a deviation angle, ϕ , which is from -60° to $+60^\circ$ with the increment of 15° , determined through analysis of the flight data.

For the guaranteed safety, the deviation angle, ϕ , should result in a trajectory that passes outside of the protected zone of the non-maneuvering aircraft.

Action: In the learning of the action parameter, the label is given as *Action-hold* for holding the deviated path and *Action-return* for returning to the planned path once the conflict is safely resolved. At each time-step, we sequentially apply a classification model for the decision between hold and return. An example is shown for *Path stretch* in Figure 2.23.

After the adjustment (or the deviation angle) is determined by the adjustment model, we first check if *Action-return* guarantees the safety (no conflict) or not: if not safe, *Action-hold* becomes the action taken at the time-step; otherwise (both *Action-hold* and *Action-return* are safe), a classification model is used to determine an action. The classification model is learned with the dataset as follows:

- Features: We use (i) the maneuvering aircraft’s distance to the non-maneuvering aircraft’s protected zone when the resolution is issued and (ii) the remaining distance to the downstream waypoint that is closest to the maneuvering aircraft.
- Label: *Action-hold* represents the action of maintaining the deviated heading and *Action-return* represents the action of changing the heading to the nearest downstream waypoint.

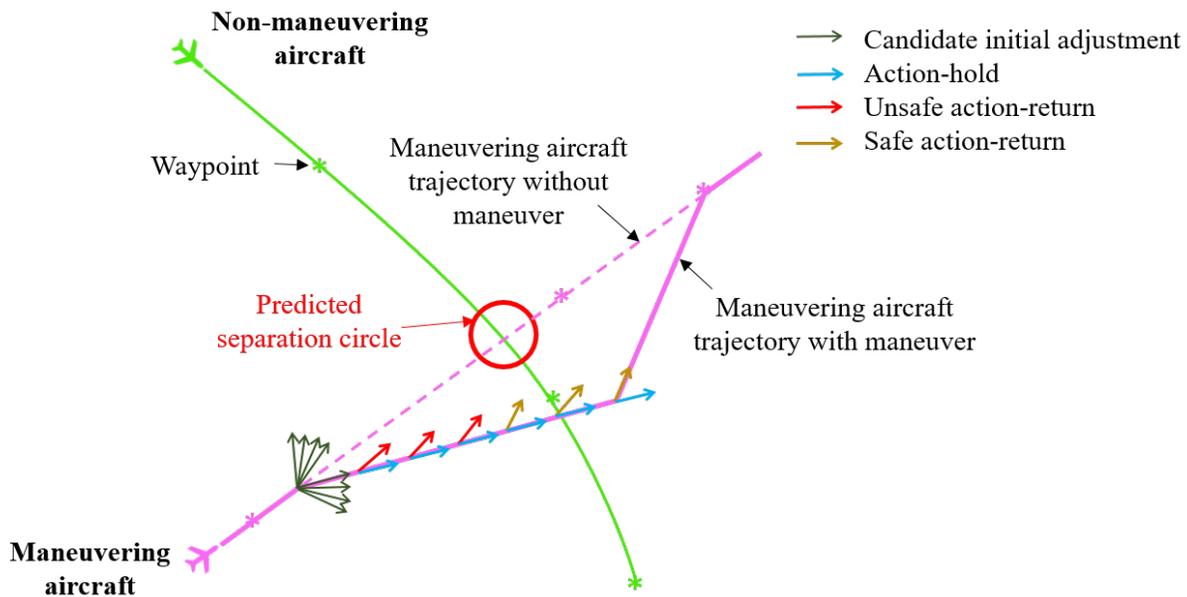


Figure 2.23. RP-action: Path stretch

For the other RTs, the adjustment parameter and the action for hold and return are summarized in Tables 2.6, 2.7 and 2.8.

Table 2.6. RPs for the horizontal RTs

Resolution type	Adjustment parameter	Action - hold	Action - return
<i>DT</i>	Downstream waypoint -		Return to feasible waypoint
<i>PS</i>	Deviation angle	Hold deviated path	Return to downstream waypoint
<i>RO</i>	Deviation angle	Hold deviated path, Hold parallel route	Return to downstream waypoint

Table 2.7. RPs for the vertical RTs

Resolution type	Adjustment parameter	Action - hold	Action - return
<i>TA-C</i>	Level off altitude	Hold altitude	Climb to cruise altitude
<i>SA-C/D</i>	Adjustment in altitude	Hold altitude	Descend/climb to cruise altitude
<i>TA-D</i>	Descent altitude	Hold altitude -	

Table 2.8. RPs for the speed RTs

Resolution type	Adjustment parameter	Action - hold	Action - return
<i>TC-S</i>	Speed deviation	Hold deviated speed	Return to cruise speed
<i>C-CS, C-DS</i>	Speed deviation	Hold deviated speed -	

2.2.4 Applying D2RG to Unseen Conflict Situations

The learned classifiers for the RTs and the learned models for the RPs constitute the proposed D2RG framework for an unseen conflict situation, as shown in Figure 2.24.

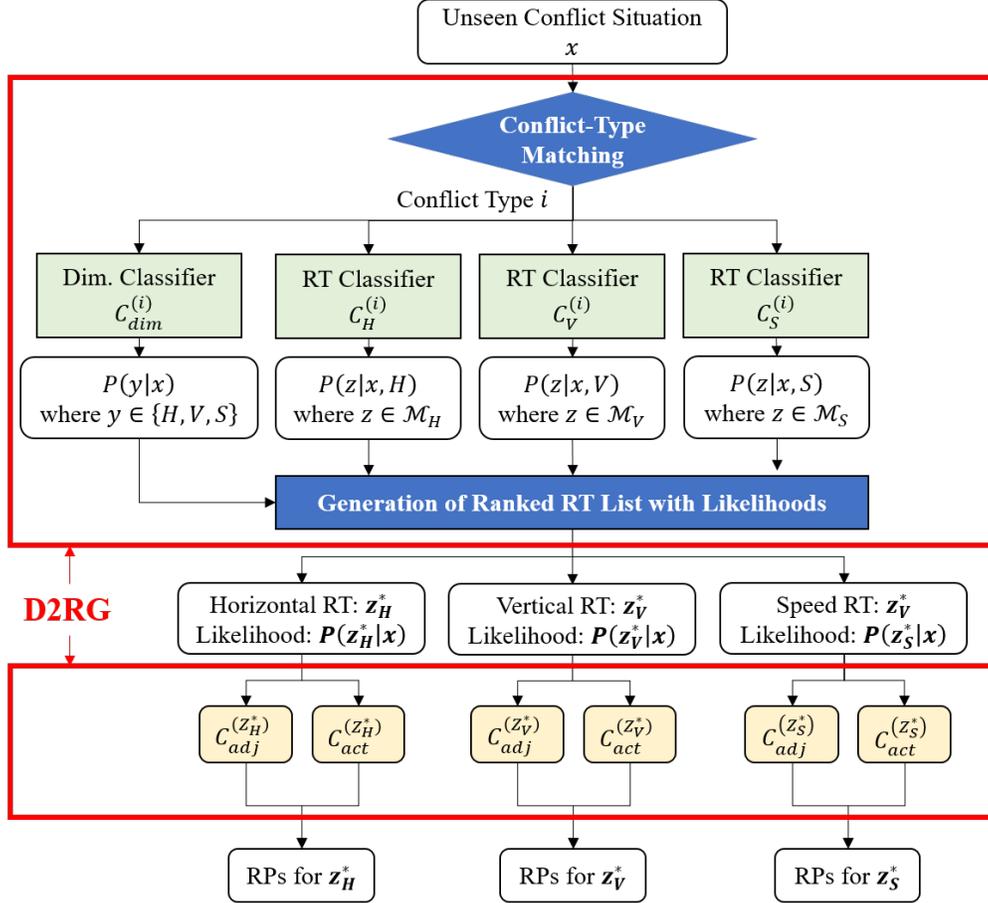


Figure 2.24. Application of D2RG to an unseen conflict situation

Suppose that a feature vector is given as x for an unseen conflict situation. The advisory list of most likely RTs for each dimension for the given x is obtained as follows:

1. The feature vector x is fed into *Conflict-Type Matching* to identify its conflict type $i \in \mathcal{M}_{CT}$.
2. For i , the classifier for the maneuvering-dimension, $C_{dim}^{(i)}$, yields the likelihood of each maneuvering dimension $y \in \{H, V, S\}$ for x , i.e., $P(y|x)$.

3. Within each y , the RT classifier $C_y^{(i)}$ computes the likelihood of RT $z \in \mathcal{M}_y$ within y for x , i.e., $P(z|x, y)$.
4. The likelihood of RT $z \in \mathcal{M}_y$ for x can then be computed as:

$$P(z|x) = P(y|x)P(z|x, y) \quad (2.22)$$

The most likely RT z_y^* for each y ,

$$z_y^* = \arg \max_{z \in \mathcal{M}_y} P(z|x) \quad \text{for } y \in \{H, V, S\} \quad (2.23)$$

and its likelihood $P(z_y^*)$ form the RT advisory list.

For z_y^* in each y , the adjustment parameters and the action parameters are computed from $C_{adj}^{(z_y^*)}$ and $C_{act}^{(z_y^*)}$, respectively, in a way that the safety is guaranteed, as follows:

1. Adjustment: If $C_{adj}^{(z_y^*)}$ suggests an adjustment parameter that is not safe (i.e., passing through the non-maneuvering aircraft's protected zone), the next most likely adjustment parameter is examined, and this process is repeated until the safety is guaranteed.
2. Action: Once the adjustment parameter is determined, $C_{act}^{(z_y^*)}$ determines if *Action-return* is likely, as well as safe at each time-step. If *Action-return* is neither likely nor safe, the model determines *Action-hold* until the next time-step. This procedure is repeated until reaching the time-step at which *Action-return* is both likely and safe.

In summary, the proposed D2RG framework can learn the knowledge about how to resolve aircraft's conflicts embedded in flight data (both in RT and RP), as well as can guarantee the safety (in RP).

2.2.5 Results and Discussion

The proposed framework is tested and demonstrated with the ACES data, which is simulated for 38 hours and 26 minutes, over 21 United States continental Air Route Traffic

Control Centers (ARTCCs). The total number of flights is 54,300, with the total of 14,599 two aircraft conflicts involved.

To determine the accuracy of the RT classification, we randomly split the conflict data into 70% for learning and 30% for test and the random split of the data is performed 10 times. For the conflict situation in the test set, the proposed D2RG is said to *correctly* classify the RT if the true RT recorded in the ACES data is included in the output, i.e., the advisory list of RTs. The prediction accuracy is measured as the ratio of the correctly classified conflict situations to the number of conflict situations in the test set, which results in 84.12% with a standard deviation of 3.02%. That is, the learned D2RG model can correctly imitate the decisions embedded in the conflict data in the determination of the RTs.

We then check the safety of the RTs in the generated advisory list by comparing the minimum distance between two conflicting aircraft by using the RPs generated by the proposed D2RG with the FAA’s separation standard. In Figure 2.25, the minimum distance between two conflicting aircraft without a resolution maneuver (in red) and with a resolution maneuver for the correctly classified cases (blue) and the misclassified cases (magenta). We can observe that all the conflict situations are successfully resolved, even for the cases where the RTs are misclassified, and hence the proposed D2RG can guaranteed the safety.

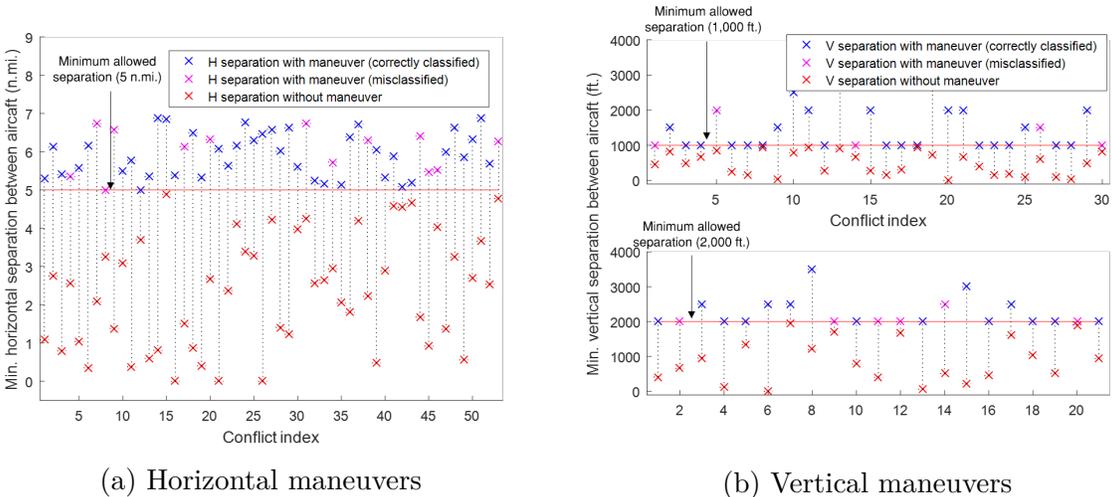


Figure 2.25. Minimum distance between two conflicting aircraft

As an illustrative example to show how the proposed D2RG works, we present a conflict situation where two conflicting aircraft are in their cruise phases and at the same altitude, as shown in Figure 2.26. The conflict is predicted with the minimum distance of 2.18 nm. In the ACES data, the recorded RT is *Path stretch*.

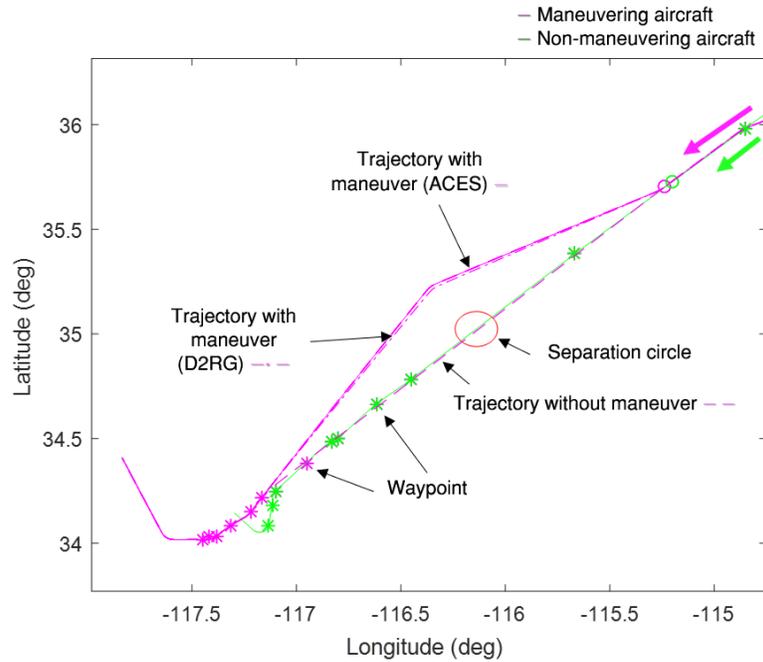


Figure 2.26. Illustrative case: Trajectories of two conflicting aircraft in the horizontal dimension

The RT classifiers result in Table 2.9, from which the advisory list of RTs is constructed as shown in Table 2.10 where PS, SA-C/D, and TC-CS are suggested for the horizontal, vertical, and speed dimensions, and hence the conflict situation is correctly classified.

For each RT in the advisory list, the adjustment and action parameters are obtained by the corresponding models where the adjust parameter values obtained are presented in Table 2.10. The resultant flight trajectories by employing the obtained resolution methods are shown in Figs. 2.26, 2.27 (a), and 2.27 (b) for the horizontal, vertical, and speed dimensions, respectively.

As shown in Figure 2.28, all the resolution methods obtained by the proposed D2RG can safely resolve the conflict situation by guaranteeing that the minimum distance between the

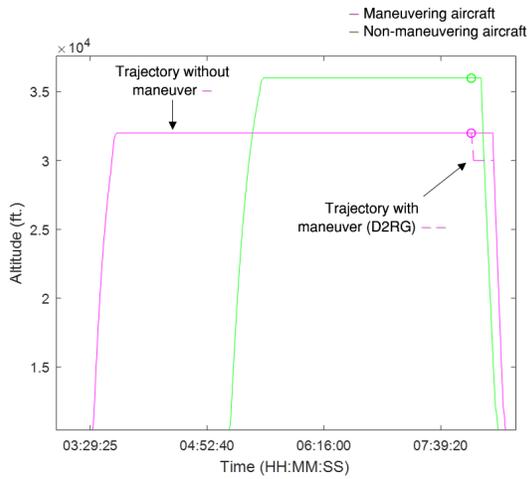
Table 2.9. Illustrative case: RT classification

Upper-level label and $P(y X_{CL/CL})$		Lower-level label and $P(y_k X_{CL/CL}, y)$		Likelihood $P(y_k X_{CL/CL})$
Horizontal	56.64%	<i>PS</i>	69.16%	36.41%
		<i>RO</i>	30.84%	17.46%
		<i>DT</i>	0%	0%
Vertical	32.87%	<i>TA-D</i>	84.22%	27.68%
		<i>SA-C/D</i>	15.78%	5.18%
		<i>TA-C</i>	0%	0%
Speed	10.49%	<i>C-CS</i>	100.00%	10.49%
		<i>TC-CS</i>	0%	0%

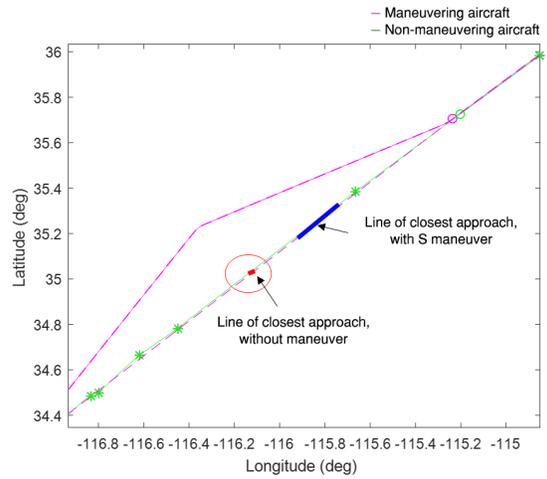
Table 2.10. Illustrative case: Advisory list of RTs and corresponding RPs

Maneuver in each dimension	Likelihood	Adjustment
<i>PS</i> (H)	36.41%	+15°
<i>TA-D</i> (V)	27.68%	-2,000 feet
<i>C-CS</i> (S)	10.49%	+10 knots

two conflicting aircraft never violates the separation standard in the horizontal and vertical dimension simultaneously.

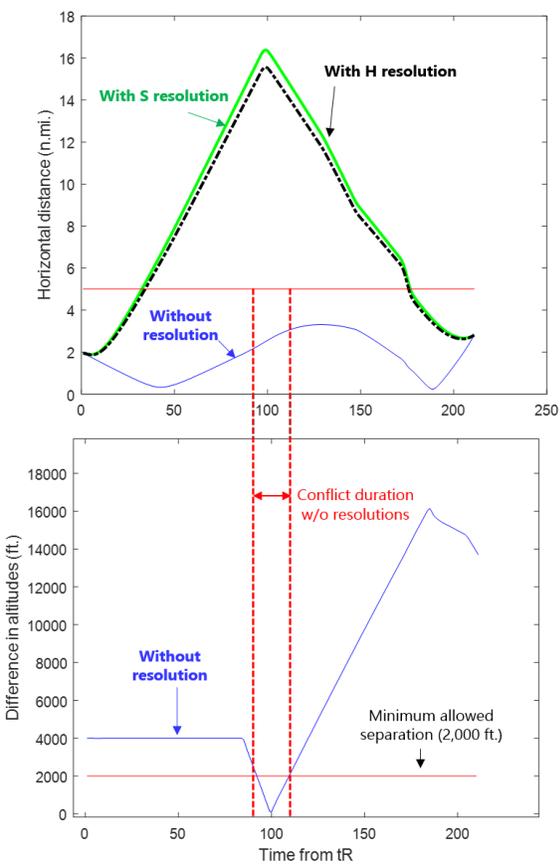


(a) SA-D

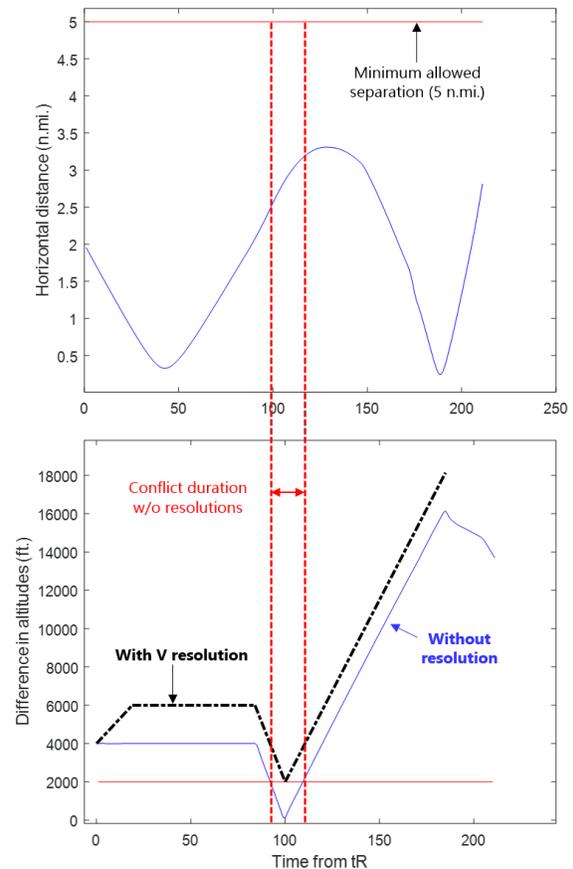


(b) TC-CS

Figure 2.27. Illustrative case: Trajectories with resolution maneuvers



(a) Horizontal separation



(b) Vertical separation

Figure 2.28. Illustrative case: Separation between two conflicting aircraft

3. DATA-DRIVEN CONFORMANCE MONITORING IN TERMINAL AIRSPACE

In this chapter, we present a framework for conformance monitoring for a flight trajectory at the current and future time, by (i) predicting its future track points using a trajectory prediction algorithm and (ii) computing the conformity scores of the future track points using a stochastic conformal prediction method, as shown in Figure 3.1.

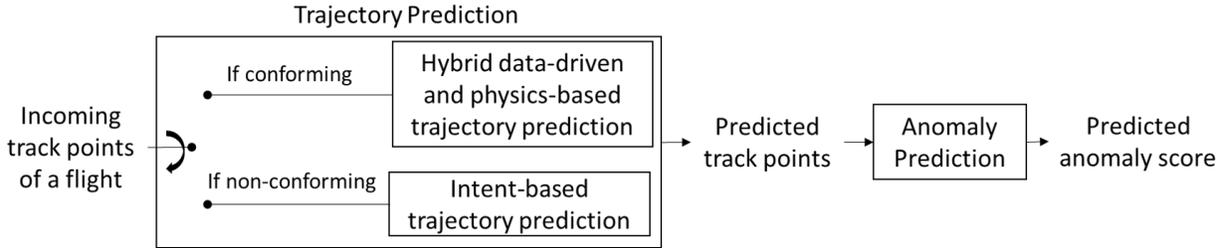


Figure 3.1. Framework for hybrid data-driven and physics-based trajectory and conformity prediction

3.1 Hybrid Data-driven and Physics-based Trajectory Prediction

In this section, we present a framework for hybrid data-driven and physics-based trajectory prediction. Suppose that there exist a large enough number of flight trajectories recorded in a dataset that represent a known intent, e.g., following a flight plan, and, therefore, we can learn a data-driven trajectory prediction model that takes the track points up to the current time-step t as input and generates a predicted track point at time-step $t + 1$, called *data-driven prediction* (which is discussed in Section 3.1.1), whose probability density function (pdf) is given as a Gaussian distribution $\mathcal{N}(\mu_{t+1}, \Sigma_{t+1})$. By using the data-driven prediction as a pseudo-measurement at $t + 1$, we can employ an estimation algorithm, such as Kalman filter or its variants, to predict the one-step ahead future track point at $t + 1$, whose pdf is given as $\mathcal{N}(\hat{z}_{t+1}, R_{t+1})$, as shown in Figure 3.2.

Note that a data-driven prediction model for time-series data typically takes a point input and generates a point output, and a physics-based prediction method takes the pdfs of the estimate at t and the measurement at $t + 1$. Hence, in the proposed framework, the

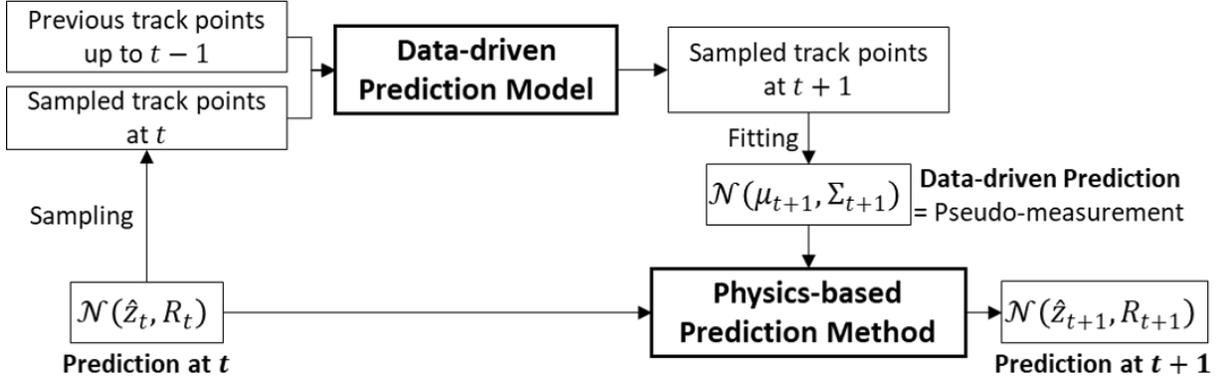


Figure 3.2. Framework for hybrid data-driven and physics-based trajectory prediction

pdf of the prediction at t is sampled as a sampled track point at t and then combine it with the previous track points up to $t - 1$, which is then provided as an input to the data-driven model. Similarly, since the output of the data-driven model is given as a point, we collect a number of sampled track points at $t + 1$ to obtain a pdf of the pseudo-measurement at $t + 1$, by fitting the points into an assumed pdf, which is then given as an input to the physics-based method. In this thesis, we propose to use a hybrid estimation algorithm presented in Section 2.1.2 to accurately model the aircraft’s behaviors. In the following, we describe the development of the data-driven trajectory prediction model, which is then followed by the demonstration of the proposed framework with air traffic surveillance data from the repository of real historical datasets.

3.1.1 Data-driven trajectory prediction

Given a set of recorded trajectories for a given operational condition (e.g., following a specific flight plan), a data-driven trajectory prediction model is learned based on the Recurrent Neural Network (RNN) [50], which is a kind of neural networks widely used for time-series data due to its recurrent structure that can well capture temporal dependency in the time-series data as well as spatial patterns, which leads to better performance in the prediction of time-series data. Suppose we have an input time-series $X = \{x_t\}_{t=0}^T$ and an output time-series $Y = \{y_t\}_{t=0}^T$ where $x_t \in \mathbb{R}^{n_x}$ and $y_t \in \mathbb{R}^{n_y}$. To compute the predicted output \hat{y}_t , RNN captures the temporal dependency by using the current input element x_t

along with the past input elements x_0, \dots, x_{t-1} . This past information up to $t - 1$ can be accounted for in generating \hat{y}_t by introducing a mechanism called *hidden state*, $a_{t-1} \in \mathbb{R}^{n_a}$ which contains all the past information. The hidden state at t , a_t is then updated with a_{t-1} and x_t , which is used in predicting the output y_t

$$\begin{aligned} a_t &= g_a(w_{aa}a_{t-1} + w_{ax}x_t + b_a) \\ y_t &= g_y(w_{ya}a_t + b_y) \end{aligned} \tag{3.1}$$

where g_a and g_y are activation functions, such as hyper-tangent or sigmoid, and w 's and b 's are the weight and bias parameters respectively which are computed by minimizing the sum of the differences between the (true) output y_t and the predicted output \hat{y}_t for $t = 0, \dots, T$.

This standard RNN is, however, limited in learning long-term dependency in time-series data due to the gradient vanishing/exploding issue when the parameters w 's and b 's are computed during the learning process. To address this issue, several variants of RNN has been proposed, such as Long Short-Term Memory (LSTM) [51]. In LSTM, additional mechanism is introduced, called *memory cell*, c_t , as shown in Figure 3.3.

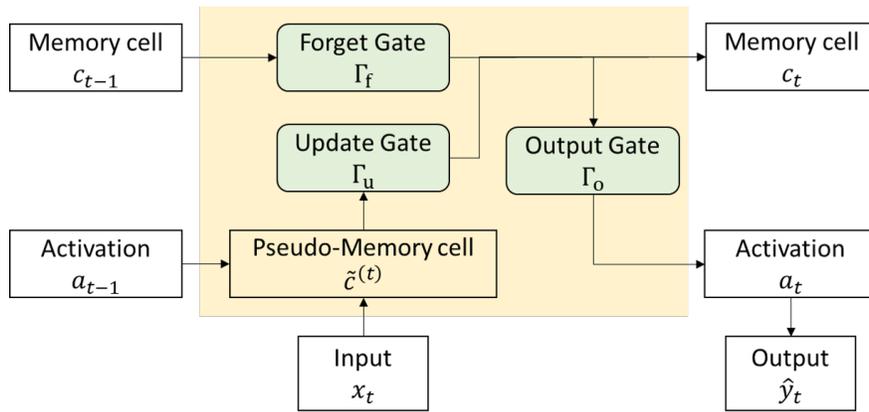


Figure 3.3. Structure of Long Short-Term Memory (LSTM)

The memory cell flows over time, during which the *gates* control how much to forget the previous memory cell c_{t-1} and update a candidate of the current memory cell \tilde{c}_t

$$\tilde{c}_t = g_{\tilde{c}}(w_{\tilde{c}a}a_{t-1} + w_{\tilde{c}x}x_t + b_{\tilde{c}}) \tag{3.2}$$

which is equivalent to a_t in the standard RNN, by adjusting the forget gate $\Gamma_f \in [0, 1]$ and update gate $\Gamma_u \in [0, 1]$, i.e.,

$$c_t = \Gamma_f * c_{t-1} + \Gamma_u * \tilde{c}_t \quad (3.3)$$

where $*$ is the element-wise multiplication, and the gates are given as, for $(\cdot) \in \{f, u\}$, $\Gamma_{(\cdot)} = g_{(\cdot)}(w_{(\cdot)a}a_{t-1} + w_{(\cdot)x}x_t + b_{(\cdot)})$. The current hidden state a_t is then updated by using the current memory c_t , an activation function g_a , and the output gate $\Gamma_o \in [0, 1]$,

$$a_t = \Gamma_o g_a(c_t) \quad (3.4)$$

where $\Gamma_o = g_o(w_{oa}a_{t-1} + w_{ox}x_t + b_o)$ with an activation function g_o . Finally, the predicted output \hat{y}_t is computed from the current hidden state a_t , the same as the way in the standard RNN.

Suppose that we have a set of trajectories for a given operational condition, $\{Z^{(i)}\}_{i=1}^N$ where N is the number of trajectories and $Z^{(i)} = \{z_t^{(i)}\}_{t=0}^{T'}$ is the i -th trajectory where $z_t^{(i)}$ is a track point of the i -th trajectory at time-step t and T' is the final time-step. Since the proposed framework in Figure 3.2 requires a one-step ahead prediction from the data-driven prediction model, the input and output time-series are constructed as $X = \{\{z_t^{(i)}\}_{t=0}^{T'-1}\}_{i=1}^N$ and $Y = \{\{z_t^{(i)}\}_{t=1}^{T'}\}_{i=1}^N$ with one-step shift. The parameters of LSTM are computed by minimizing the difference between the (true) output and the predicted output from LSTM.

Since LSTM, or any existing neural networks, takes a point input and generates a point output as noted above, the uncertainty of the data-driven prediction is not directly available (which corresponds to the covariance Σ_{t+1} in Figure 3.1). To obtain the uncertainty of LSTM, we use a technique called *Monte Carlo dropout (MCDO)* [62], which is a widely used scheme for learning the uncertainty of a neural network. With dropout, a connection in LSTM is dropped, or disconnected, with a probability, called dropout rate, which is a design parameter. By collecting the output of LSTM with MCDO for a number of input samples and Monte Carlo runs, a Gaussian pdf is fitted to the collected output samples, thus resulting in the pdf of the data-driven prediction, $\mathcal{N}(\mu, \Sigma)$.

3.1.2 Demonstration of the proposed trajectory prediction algorithm

In this section, we demonstrate the proposed algorithm with real Automatic Dependent Surveillance-Broadcast (ADS-B) data. With ADS-B technology, an aircraft determines its position through satellite navigation and broadcasts it to a ground station or other aircraft in the proximity. The collected ADS-B used in this research was recorded from January to June in 2020, around the two major airports in the Republic of Korea, Incheon International Airport (ICN) and Gimpo International Airport (GMP).

For the illustration purpose, the proposed framework is applied to one arrival trajectory and one departure trajectory, as shown in Figure 3.4 and Figure 3.5, respectively. We compare the proposed hybrid data-driven and physics-based trajectory prediction method (in blue in each figure) with two baseline methods, the data-driven only (in red) and the physics-based only (in green). The performance of each method is measured by the prediction error, i.e., the difference between the predicted trajectory and the recorded trajectory (in black). In each figure, the left shows the trajectories and the right represents the prediction error. The prediction is performed at the last/initial part of arrival/departure trajectories for the prediction horizon of 20 time-steps where the time interval is 5 seconds.

It is shown that the proposed method outperforms the baseline methods by incorporating the aircraft's current dynamics (physics-based) with the expected states in the future from the learned LSTM (data-driven). In the next section, we use the proposed method to predict the future track points of an aircraft under monitoring for predicting the conformity scores of each future track point.

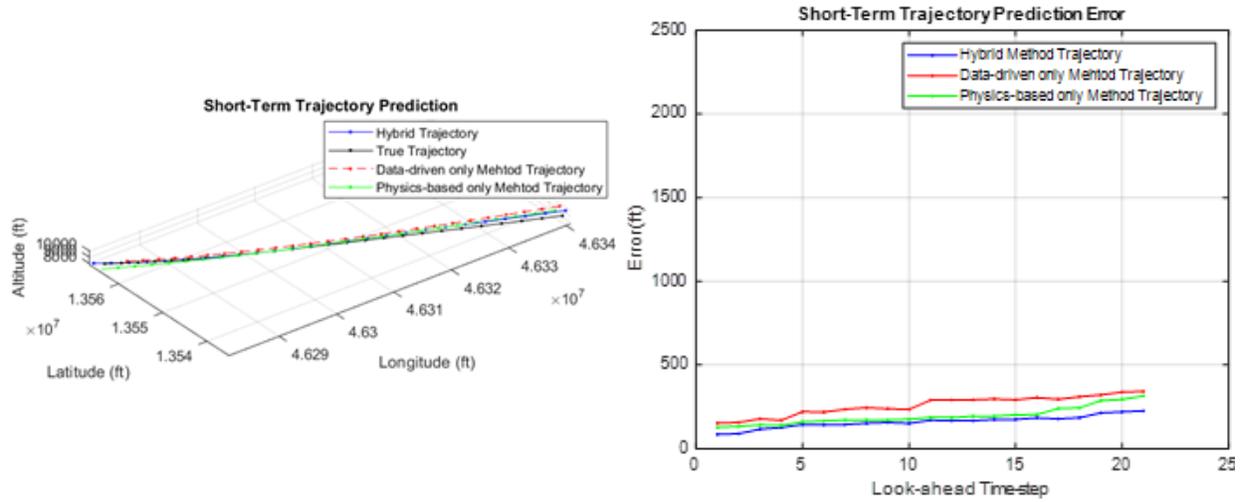


Figure 3.4. Arrival trajectory prediction

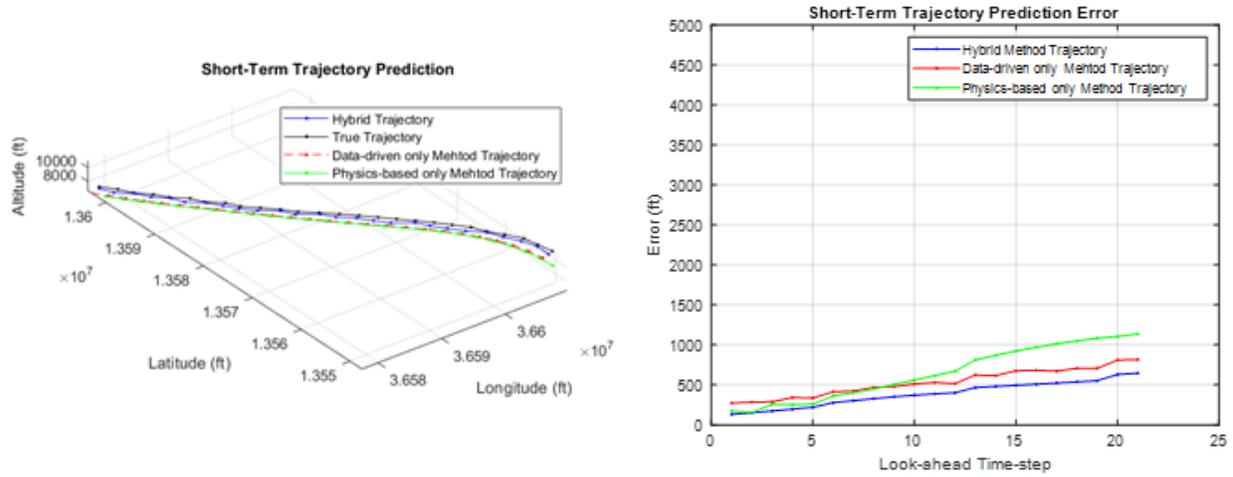


Figure 3.5. Departure trajectory prediction

3.2 Conformance Monitoring with Stochastic Conformal Prediction

In this section, a method for computing the conformity scores of the future track points is presented. Suppose that we have a set of similar trajectories, $\{Z^{(i)}\}_{i=1}^N$ where $Z^{(i)} = \{z_t^{(i)}\}_{t=0}^T$ is an aircraft trajectory, z_t is a track point at time-step t , and T is the final time-step. For the

track points of a new, incoming trajectory observed up to time-step T_p , this new $(N + 1)$ -th trajectory is constructed as

$$Z^{(N+1)} = \{z_t^{(N+1)}\}_{t=0}^{T_p} \cup \{\hat{z}_t^{(N+1)}\}_{t=T_p+1}^{T_p+T_h} \quad (3.5)$$

with the track points observed up to T_p and the ones predicted for the future horizon of T_h that are obtained by the method described in Section 3.1. We use conformal prediction [63] in order to compute the conformity score of $Z^{(N+1)}$ with respect to $\{Z^{(i)}\}_{i=1}^N$, which is given as

$$p^{(N+1)} = \frac{|\alpha^{(i)} \geq \alpha^{(N+1)}, i = 1, \dots, N + 1|}{N + 1} \quad (3.6)$$

where $\alpha^{(i)}$ for $i = 1, \dots, N + 1$ is a non-conformity measure (NCM) that represents how different $Z^{(i)}$ is from $\{Z^{(j)}\}_{j=1, j \neq i}^{N+1}$. The smaller $p^{(N+1)}$, the more non-conforming $Z^{(N+1)}$ to $\{Z^{(j)}\}_{j=1}^N$, and hence $p^{(N+1)}$ can be used as an *conformity score*, i.e., a measure of how conforming the new trajectory is with respect to the set of similar trajectories. To compute the NCM, $\alpha^{(i)}$'s, we use the directed Hausdorff distance (DHD) [53][64], which measures how the shape of a set of points A resembles some part of the shape of another set of points B , defined as

$$\vec{\delta}_H(A, B) = \max_{a \in A} \left\{ \min_{b \in B} \{d(a, b)\} \right\} \quad (3.7)$$

where $d(a, b)$ is a distance between two points a and b , by some metric, e.g., the Euclidean distance. Note that the DHD is not symmetric, i.e., $\vec{\delta}_H(A, B) \neq \vec{\delta}_H(B, A)$ in general.

Suppose that $A = \{a_0, a_1\}$ is a new trajectory where a_0 is deterministic and $a_1 \sim \mathcal{N}(\bar{a}_1, \Sigma_1)$ and $B = \{b_0, b_1, b_2\}$ is a trajectory from the set of trajectories where all the track points b_0, b_1, b_2 are deterministic. Due to the stochastic track point a_1 , the DHD, $\delta_H(A, B)$, as well as the NCMs, α 's, also become stochastic variables. To handle this, we compute the expectation of the DHD, which is given as

$$\mathbb{E} \left(\vec{\delta}_H(A, B) \right) = \int_{a_1} \vec{\delta}_H(A(a_0, a_1), B(b_0, b_1, b_2)) f(a_1) da_1 \quad (3.8)$$

where $f(a_1)$ is the pdf of a_1 . However, due to the minimum and maximum in the DHD, the numerical integration is not trivial to perform. In this regard, we propose to use an approximated form of the DHD that can be numerically integrated. Among the several approximation methods for the minimum/maximum function (called smooth minimum/maximum), such as the LogSumExp (LSE; also called RealSoftMax), p -norm, and the generalized mean [65], we choose the LSE which is widely used for machine learning [66] and given as

$$\max\{x_1, \dots, x_n\} \approx \frac{1}{\rho} \log (\exp (\rho x_1) + \dots + \exp (\rho x_n)) \quad (3.9)$$

where ρ is a large enough positive constant. The minimum can be similarly approximated by replacing ρ with $-\rho$.

By applying the LSE technique twice, the DHD can be approximated as

$$\begin{aligned} \vec{\delta}_H(A, B) &= \max_{a \in A} \left\{ \min_{b \in B} \{d(a, b)\} \right\} \\ &\approx \frac{1}{\rho_M} \log \left\{ \frac{1}{|A|} [\exp \{\rho_M d_m(a_0)\} + \exp \{\rho_M d_m(a_1)\}] \right\} \end{aligned} \quad (3.10)$$

where, for $a \in A$,

$$d_m(a) = -\frac{1}{\rho_m} \log \left\{ \frac{1}{|B|} [\exp \{-\rho_m d(a, b_0)\} + \exp \{-\rho_m d(a, b_1)\} + \exp \{-\rho_m d(a, b_2)\}] \right\} \quad (3.11)$$

and ρ_m and ρ_M are large enough positive constants. By plugging Eq. (3.10) and Eq. (3.11) into Eq. (3.8), we can compute the expectation of the DHD, $\mathbb{E}(\vec{\delta}_H(A, B))$, if trajectory A contains one stochastic track point and trajectory B is entirely deterministic, and similarly for $\mathbb{E}(\vec{\delta}_H(B, A))$ (the case where there are multiple stochastic track points is discussed in Section 3.2.2). Note that if both trajectories A and B are deterministic, the expectation of the DHD is simply equal to Eq. (3.7).

Once computing the expectations of the DHDs for all the pairs from $\{Z^{(i)}\}_{i=1}^{N+1}$, the NCM of the i -th trajectory, $\alpha^{(i)}$ for $i = 1, \dots, N + 1$, is then obtained as

$$\alpha^{(i)} = \sum_{j=1}^k \mathbb{E} \left[\vec{\delta}_H \left(Z^{(i)}, NN \left(\left\{ Z^{(l)} \right\}_{l=1, l \neq i}^{N+1} \setminus Z^{(i)}, j \right) \right) \right] \quad (3.12)$$

where $NN \left(\left\{ Z^{(l)} \right\}_{l=1, l \neq i}^{N+1} \setminus Z^{(i)}, j \right)$ is the j -th nearest neighbor to $Z^{(i)}$ according to the expectation of the DHD and k is a design parameter. With the computed NCMs, the conformity score of the new trajectory, $p^{(N+1)}$, is computed using Eq. (3.6).

For the validation of the proposed method, we compare it with Monte Carlo simulation to measure the performance in terms of computation time and accuracy. The computation time is checked along the number of trajectories, N , and the number of track points in each trajectory, T . As shown in Figure 3.6, the proposed method is significantly efficient compared to Monte Carlo simulation. For a given N and T , the NCM value from Monte Carlo simulation is 2.1908 (considered as a ground truth), while the proposed method has 2.2039, which has 0.60% error with respect to the ground truth. That is, the proposed method using the smooth approximation can perform the computation effectively (in terms of accuracy) and efficiently (in terms of computation time).

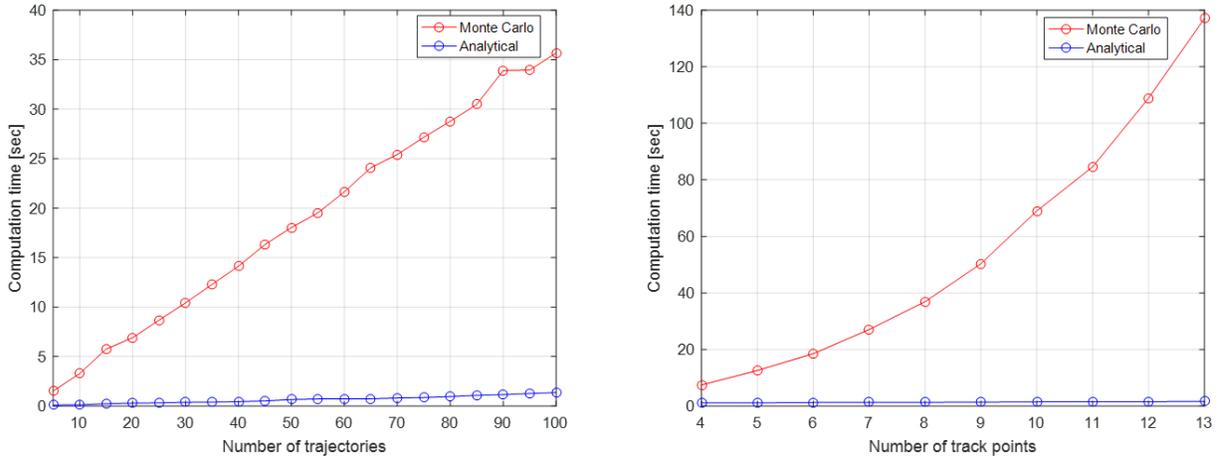


Figure 3.6. Comparison of the computation time: Monte Carlo simulation and the proposed approximation

3.2.1 Conformance monitoring at the current time

In this section, the algorithm described in Section 3.2 is applied to conformance monitoring for the current time. For a new incoming trajectory, the track points observed up to the current time-step are used to detect if it is conforming to the set of similar trajectories. The

states of a trajectory at the current time-step is estimated using the hybrid estimation algorithm in Section 2.1.2 and all the past track points are considered deterministic (therefore the new trajectory contains only one stochastic variable).

With the data used in Section 3.1.2, we present the four abnormal cases of the new trajectory:

1. Departure trajectory that performs Direct-to (deviating from the normal trajectories and never returning to them), shown in Figure 3.7
2. Departure trajectory that performs Path-stretch (deviating from the normal trajectories and returning to them), shown in Figure 3.8
3. Arrival trajectory that performs Direct-to, shown in Figure 3.9
4. Arrival trajectory that performs Path-stretch, shown in Figure 3.10

In each figure, the upper plot shows the set of normal trajectories (in black; the arrow represents the moving direction) and the new trajectory where the red dots and blue ellipses represent the mean and covariance (two standard deviation) of the estimated current states, respectively, in the horizontal plane. The lower plot presents the time history of the conformity score of the new trajectory. The arrows with time in the upper plot represent the initial position (0 sec) and the position where the conformity score first becomes 0.

For the departure trajectories, the Direct-to trajectory in Figure 3.7 is detected as an anomaly within three time-steps, while the Path-stretch trajectory in Figure 3.8 is within seven time-steps. Possible reasons for such detection delay in the Path-stretch include (i) that the Path-stretch is very close to the normal trajectories at the initial time-steps so that its conformity score is high, which means that it is considered as conforming during the period; and (ii) that the set of normal trajectories for the Path-stretch case have a relatively wide variations than the Direct-to case, which implies that the NCMs of the normal trajectories and the new trajectory are comparable each other. Note that, however, although the Path-stretch trajectory returns to the normal trajectories, its conformity score never increases (i.e., it is converged to 0), because the deviated part of the trajectory makes its shape different from the normal trajectories.

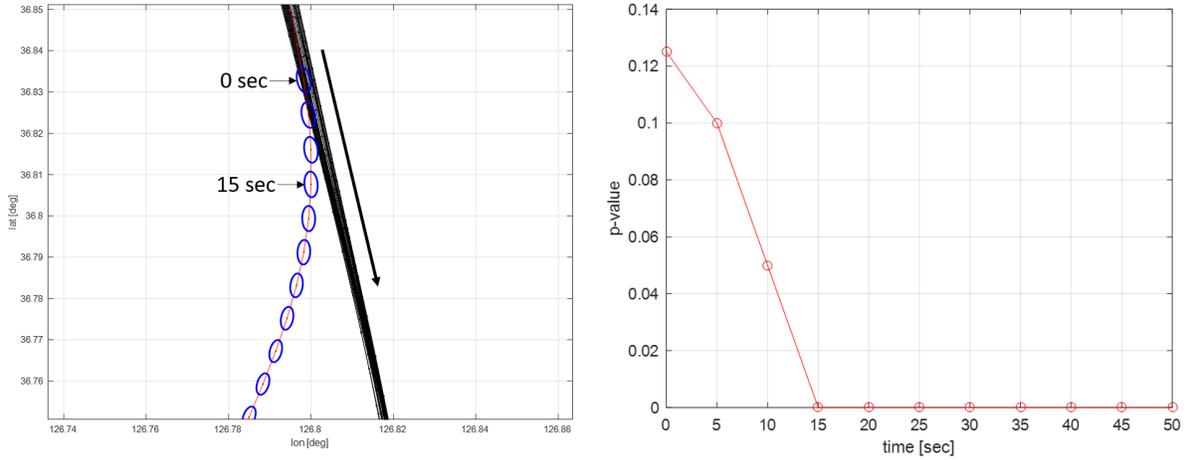


Figure 3.7. Conformance monitoring at the current time: Direct-to anomaly (departure)

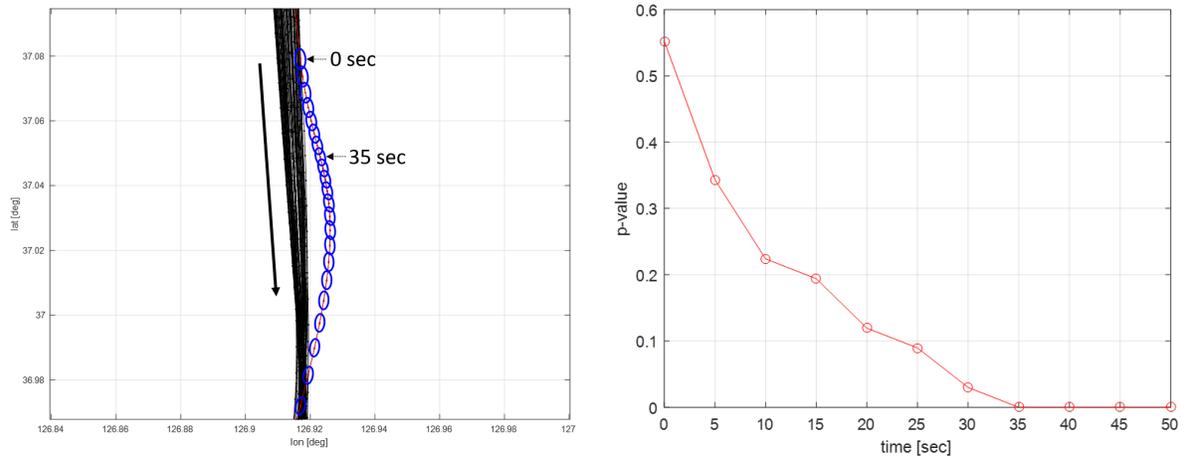


Figure 3.8. Conformance monitoring at the current time of departure trajectories with Path stretch anomaly (departure)

For the arrival trajectories, the Direct-to trajectory in Figure 3.9 is detected as an anomaly within five time-steps, while the Path-stretch trajectory in Figure 3.10 is within four time-steps. For the Direct-to, its conformity score starts from about 0.5 (in the middle of conforming (1) and non-conforming (0)), it quickly decreases to about 0.05 in four time-steps and then converges to 0 in the next time-step. For the Path-stretch, its conformity score starts from very low value of about 0.037, possibly due to its direction before the initial time-step which is different from the normal trajectories. It converges to 0 within four time-

steps, which could be attributed to the wide variations of the normal trajectories, similar to the departure Path-stretch case.

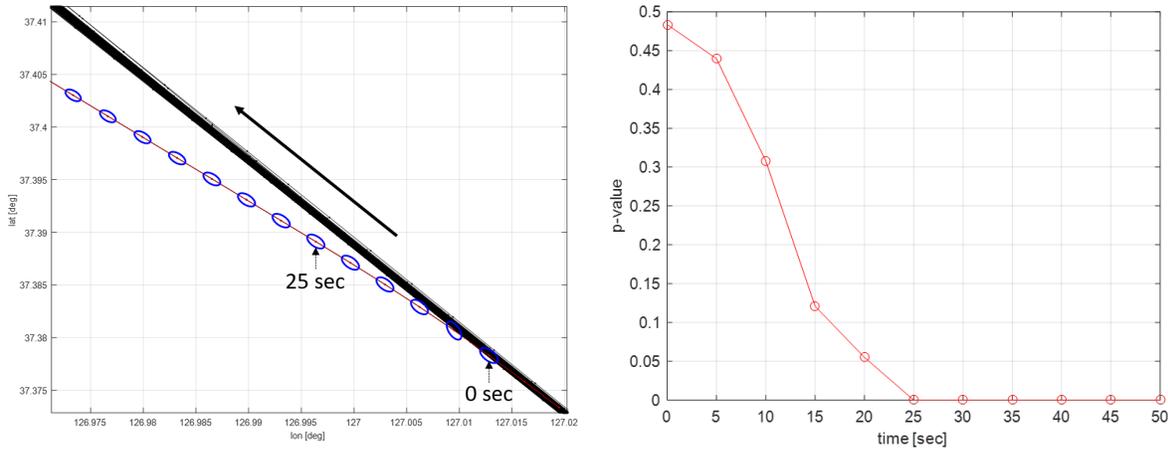


Figure 3.9. Conformance monitoring at the current time: Direct-to anomaly (arrival)

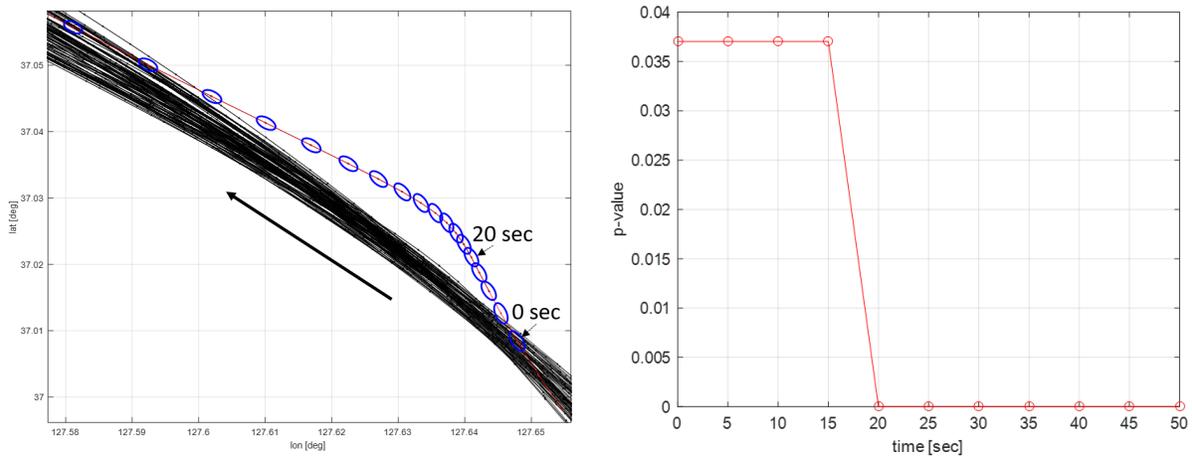


Figure 3.10. Conformance monitoring at the current time: Path stretch anomaly (arrival)

3.2.2 Conformance monitoring at the future time

In this section, the algorithm described in Section 3.2 is applied to conformance monitoring at the future time.

Note that, for the trajectory prediction presented in Section 3.1, it is supposed that there exist a large enough number of trajectories recorded in a dataset corresponding to a

known intent, such as following a flight plan or deviating due to ATC's instructions. If a new incoming trajectory is conforming to any known intents, its future track points can be predicted by the trajectory prediction method presented in Section 3.1. However, if it is non-conforming to any of known intents, the method in Section 3.1 is not applicable. For such cases where any information about the future is not available, the future trajectory can only be predicted by propagating its current states into the future using a dynamics. Note that, in the horizontal plane, the two types of anomaly, Direct-to and Path-stretch, can be distinguished after an aircraft takes turning and then returns to the normal trajectories (Path-stretch) or it reaches a known waypoint with a straight line motion (Direct-to). Direct-to and Path-stretch, however, share a property that the part of the trajectory right after the deviation starts is (almost) straight line. Through data analysis, it is identified that any trajectory of Direct-to or Path-stretch maintains the straight line at least for 25 seconds. In this regard, we propose the following: once a new incoming trajectory is determined that it is not conforming to any known intent (by using the maneuver detection and characterization method in Section 2.1), its current intent is considered as maintaining the heading of the trajectory at the time of detection at least for 25 seconds, based on which the future track points are predicted along this intent [47].

For the either of conforming or non-conforming cases, there are more than one future track points predicted. For two trajectories, $Z^{(i)}$ and $Z^{(j)}$, suppose that $Z^{(i)}$ contains the N_s number of stochastic track points, $z_1^{(i)}, \dots, z_{N_s}^{(i)}$. Then, the expectation of the DHD in Eq. (3.8) for a single stochastic variable becomes

$$\mathbb{E}(\vec{\delta}_H(Z^{(i)}, Z^{(j)})) = \int_{z_{N_s}^{(i)}} \dots \int_{z_1^{(i)}} \vec{\delta}_H(Z^{(i)}, Z^{(j)}) f(z_1^{(i)}) \dots f(z_{N_s}^{(i)}) dz_1^{(i)} \dots dz_{N_s}^{(i)} \quad (3.13)$$

Since the computation time of numerical integration in Eq. (3.13) exponentially increases with the number of the stochastic variables, it cannot be directly applicable in real-time application. In this work, we use the method similar to the one discussed in Section 3.2.1 by replacing the estimate of the current track point with a predicted track point for a future time-step within a prediction horizon. That is, for the time when the prediction is performed, T_p , and the prediction horizon, T_h , the predicted track point at each future

time-step $T_p + i$ for $i = 1, \dots, T_h$ along with the past track points up to T_p are combined to construct a new trajectory. We compare the conformity score computed by the proposed conformance monitoring at the future time (that is, before observing the track point at $T_p + i$ for $i = 1, \dots, T_h$) with the one by conformance monitoring at the current time (that is, once observing the track point at $T_p + i$ for $i = 1, \dots, T_h$). The performance is measured by (i) the accuracy, that is, the difference between the predicted conformity score and the one from the conformance monitoring at the current time, and (ii) detection delay, i.e., the difference between the time-steps when the anomaly is detected by the two methods.

For the illustration, we present three cases of a new trajectory, one normal and two abnormal trajectories, as shown in Figure 3.11 to Figure 3.13. In each figure, the upper plot shows the set of normal trajectories (in grey for the normal case for the visibility and in black for the abnormal cases; the arrow represents the moving direction) and the new trajectory where the blue dots and ellipses represent the mean and covariance of the track points up to the current time-step (represented as 0 sec) and the red dots and ellipses represent those at the future time-steps. The lower plot presents the time history of the conformity score of the new trajectory where the blue one is from the proposed conformance monitoring at the future time (prediction) and the red one is from conformance monitoring at the current time (estimation).

For the normal case in Figure 3.11, the conformity score of the proposed conformance monitoring at the future time does not converge to the one of the conformance monitoring at the current time within the prediction horizon, but both scores are around or above 0.5, which can be regarded as normal.

For the abnormal cases in Figure 3.12 and Figure 3.13, the conformity scores converge to 0 with delay in detection time of zero or one. The trajectory prediction with a straight line and the use of a single stochastic variable work well for the cases shown in this section, but it is required to significantly extend the proposed method to be useful in real-world applications. The corresponding future research directions are discussed in Section 4.2.

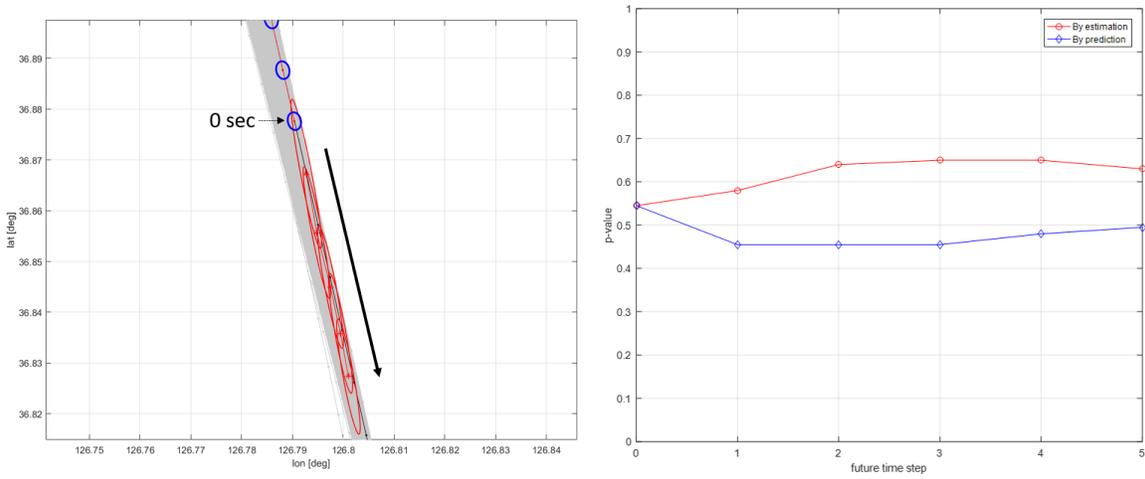


Figure 3.11. Conformance monitoring at the future time: normal trajectory.

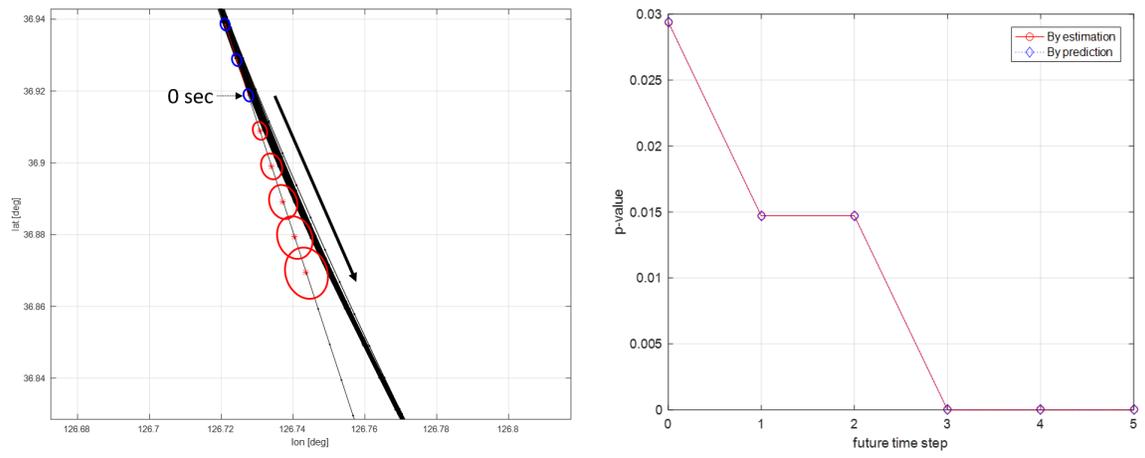


Figure 3.12. Conformance monitoring at the future time: abnormal trajectory no. 1

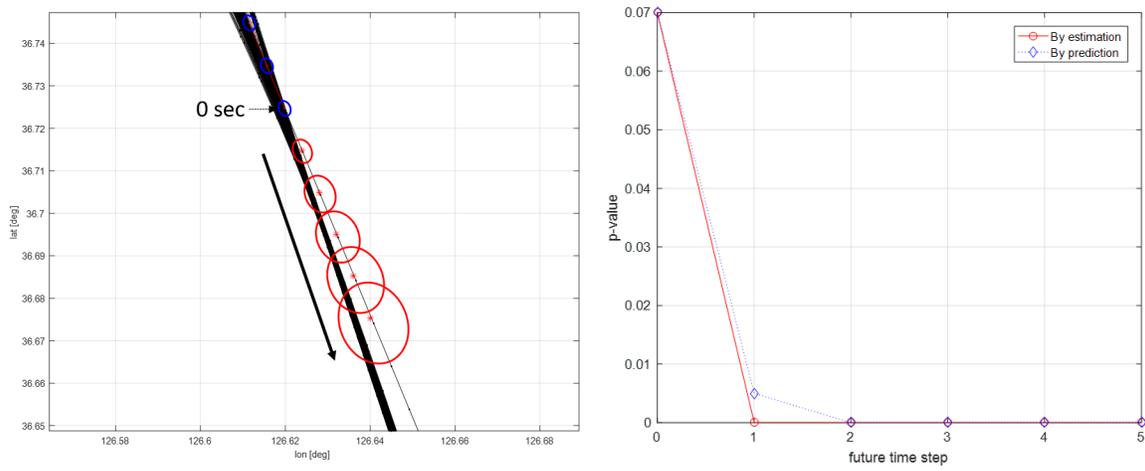


Figure 3.13. Conformance monitoring at the future time: abnormal trajectory no. 2

4. CONCLUSION

Concluding remarks and possible directions of the research presented in this thesis are presented in this chapter.

4.1 Concluding remarks

This thesis focuses on the development of decision supporting tools for air traffic controllers, especially focused on the safety of air traffic management. This effort is presented in two-folds based on the two main tasks of air traffic controllers (ATCs): control and monitoring.

For *control*, a data-driven conflict resolution tool which can aid ATCs in the decision-making process for air traffic conflict resolution has been proposed. To achieve this objective, first, an algorithm has been proposed for the detection and identification of aircraft maneuvers. The maneuver types have been represented as a sequence of simple motions, called intents. Flight plans, flight tracks, and the maneuver models are used to infer the aircraft's intents based on an aircraft's continuous and discrete states by using a hybrid estimation algorithm, and the maneuvers are inferred based on the inferred intents. By testing with flight data, it has been demonstrated that the proposed algorithm can successfully identify the aircraft maneuvers. Second, a data-driven technique has been proposed for the development of conflict resolution tool, which learns from flight data to extract the knowledge about the resolution methods embedded in the data, thereby providing an advisory list of resolution maneuvers that can guarantee the safety. For classification of resolution types, a hierarchical form of supervised learning techniques has been used. For a classified resolution type, the corresponding resolution parameters are also provided in a way that the separation between two conflicting aircraft can be guaranteed.

For *monitoring*, a hybrid data-driven and physics-based trajectory and anomaly prediction framework has been proposed to help enhance the situational awareness of ATCs, by understanding the current status and predicting the future states of aircraft. To predict the future states of an aircraft, a trajectory prediction framework has been developed by combining a data-driven prediction model and a physics-based prediction method. Since the

estimated or predicted states of an aircraft are stochastic, a stochastic version of anomaly detection and prediction algorithm has been developed. The developed methods have been demonstrated with a real surveillance data.

4.2 Future Work

For *control*, to further improve the performance of the conflict resolution generator, learning from large-scale real flight datasets for a specific airspace, such as a sector, would make it more effective in mimicking the ATC response to aircraft conflicts. Also, the supervised learning used in this work tries to learn the air traffic controller's practices recorded in the data, but there could be other performance measures such as time or fuel consumption. The applicability of the algorithm would be improved by investigating the balancing between the exploitation (use the current practice learned from the data) and the exploration (find more efficient solution).

For *monitoring*, the proposed trajectory prediction requires a large enough number of trajectories for a known intent. The known intent in this thesis is only the case where the flight trajectories follow a known flight plan. This can be extended to a smaller set of flight trajectories, such as vectoring patterns, i.e., the trajectory patterns formed by a frequent instructions of ATCs for a specific traffic situation. Also, if the stochastic conformal prediction can be further extended to a version that can account for more than one stochastic variables, the performance of conformance monitoring for the future time would be improved.

REFERENCES

- [1] “Air passenger forecasts - country report (United States),” IATA - Tourism Economics, Tech. Rep., 2016.
- [2] F. A. Regulations, “Part 91-general operating and flight rules,” *Federal Aviation Administration (FAA), USA*, 2017.
- [3] D. R. Isaacson and H. Erzberger, “Design of a conflict detection algorithm for the center/TRACON automation system,” in *Digital Avionics Systems Conference, 1997. 16th DASC., AIAA/IEEE, IEEE*, 1997.
- [4] D. J. Brudnicki, K. S. Lindsay, and A. L. McFarland, “Assessment of field trials, algorithmic performance, and benefits of the user request evaluation tool (uret) conflict probe,” in *16th DASC. AIAA/IEEE Digital Avionics Systems Conference. Reflections to the Future. Proceedings*, Oct. 1997.
- [5] W. C. Brodegard, D. E. Ryan, and P. A. Ryan, *Aircraft traffic alert and collision avoidance device*, US Patent 5,157,615, 1992.
- [6] J. Kuchar and A. C. Drumm, “The traffic alert and collision avoidance system,” *Lincoln Laboratory Journal*, 2007.
- [7] H. Erzberger, “Automated conflict resolution for air traffic control,” in *25th International Congress of the Aeronautical Sciences*, 2006.
- [8] H. Erzberger, T. Lauderdale, and Y.-C. Chu, “Automated conflict resolution, arrival management and weather avoidance for air traffic management,” in *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 2011.
- [9] Y. Matsuno, T. Tsuchiya, J. Wei, I. Hwang, and N. Matayoshi, “Stochastic optimal control for aircraft conflict resolution under wind uncertainty,” *Aerospace Science and Technology*, vol. 43, pp. 77–88, 2015.
- [10] X.-M. Tang, P. Chen, and B. Li, “Optimal air route flight conflict resolution based on receding horizon control,” *Aerospace Science and Technology*, vol. 50, pp. 77–87, 2016.
- [11] E. Hernández-Romero, A. Valenzuela, and D. Rivas, “A probabilistic approach to measure aircraft conflict severity considering wind forecast uncertainty,” *Aerospace Science and Technology*, vol. 86, pp. 401–414, 2019.
- [12] A. Alonso-Ayuso, L. Escudero, and F. Martin-Campo, “Multiobjective optimization for aircraft conflict resolution: A metaheuristic approach,” *European Journal of Operational Research*, vol. 248, pp. 691–702, 2015.

- [13] A. Alonso-Ayuso, L. Escudero, F. Matrin-Campo, and N. Mladenovic, “A vns meta-heuristic for solving the aircraft conflict detection and resolution problem by performing turn changes,” *Journal of Global Optimization*, vol. 63, pp. 583–596, 2015.
- [14] A. Richards and J. How, “Aircraft trajectory planning with collision avoidance using mixed integer linear programming,” in *2002 American Control Conference*, 2002.
- [15] F. Borrelli, D. Subramanian, A. Raghunathan, and L. Biegler, “Milp and nlp techniques for centralized trajectory planning of multiple unmanned air vehicles,” in *2006 American Control Conference*, 2006.
- [16] J. Rong, J. Geng, J. Valasek, and T. R. Ioerger, “Air traffic conflict negotiation and resolution using an onboard multi-agent system,” in *21st Digital Avionics Systems Conference*, IEEE, 2002.
- [17] A. Visinini, W. Glover, J. Lygeros, and J. Maciejowski, “Monte-carlo optimization for conflict resolution in air traffic control,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, 2006.
- [18] J. K. Kuchar and L. C. Yang, “A review of conflict detection and resolution modeling methods,” *IEEE Transactions on intelligent transportation systems*, 2000.
- [19] S. van Rooijen, J. Ellerbroek, C. Borst, and E. van Kampen, “Conformal automation for air traffic control using convolutional neural network,” in *2019 Air Traffic Management Research and Development (ATM RD) Seminar*, 2019.
- [20] P. Tran, D.-T. Pham, S. Goh, S. Alam, and V. Duong, “An interactive conflict solver for learning air traffic conflict resolutions,” *Journal of Aerospace Information Systems*, vol. 17, no. 6, 2020.
- [21] T. Reynolds and R. Hansman, “Investigating conformance monitoring issues in air traffic control using fault detection approaches,” MIT International Center for Air Transportation, Tech. Rep. ICAT-2003-5, 2003.
- [22] E. Shank and K. Hollister, “Precision runway monitor,” *MIT Lincoln Laboratory Journal*, vol. 7, no. 2, 1994.
- [23] “National airspace system en-route configuration management document - computer program functional specifications - introduction to specification series,” FAA, Tech. Rep. NAS-MD-310, 1995.
- [24] D. Love, W. Horn, and L. Fellman, “An evaluation of the use of the uret prototype tool at indianapolis and memphis air route traffic control centers, using benefit matrices,” in *21th Digital Avionics Systems Conference*, 2002.

- [25] T. Reynolds and R. Hansman, “Investigating conformance monitoring issues in air traffic control using fault detection techniques,” *Journal of Aircraft*, vol. 42, no. 5, 2005.
- [26] C.-E. Seah, A. Aligawesa, and I. Hwang, “An algorithm for conformance monitoring in air traffic control,” *AIAA Journal of Guidance, Control, and Dynamics*, vol. 33, no. 2, 2010.
- [27] G. Mann and I. Hwang, “Four-dimensional aircraft taxiway conformance monitoring with constrained stochastic linear hybrid systems,” *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 5, 2012.
- [28] G. Lee and Y. Fukuda, “Conformance monitoring under uncertainty in trajectory,” in *AIAA Guidance, Navigation, and Control Conference*, 2012.
- [29] Q. Zheng and Y. Zhao, “Probabilistic approach to trajectory conformance monitoring,” *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 6, 2012.
- [30] D. Sotiriou, F. Kopsaftopoulos, and S. Fassois, “An adaptive time series probabilistic framework for 4d trajectory conformance monitoring,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 6, 2016.
- [31] L. Basora, X. Olive, and T. Dubot, “Recent advances in anomaly detection methods applied to aviation,” *Aerospace*, vol. 6, no. 11, p. 117, 2019.
- [32] Y. Liu and W. Ding, “A knns based anomaly detection method applied for uav flight data stream,” in *2015 Prognostics and System Health Management Conference (PHM)*, IEEE, 2015, pp. 1–8.
- [33] L. Li, S. Das, R. John Hansman, R. Palacios, and A. N. Srivastava, “Analysis of flight data using clustering techniques for detecting abnormal operations,” *Journal of Aerospace information systems*, vol. 12, no. 9, pp. 587–598, 2015.
- [34] A. M. Churchill and M. Bloem, “Clustering aircraft trajectories on the airport surface,” in *Proceedings of the 13th USA/Europe Air Traffic Management Research and Development Seminar, Chicago, IL, USA*, 2019, pp. 10–13.
- [35] I. Melnyk, B. Matthews, H. Valizadegan, A. Banerjee, and N. Oza, “Vector autoregressive model-based anomaly detection in aviation systems,” *Journal of Aerospace Information Systems*, vol. 13, no. 4, pp. 161–173, 2016.

- [36] I. Melnyk, A. Banerjee, B. Matthews, and N. Oza, “Semi-markov switching vector autoregressive model-based anomaly detection in aviation systems,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1065–1074.
- [37] L. Li, R. J. Hansman, R. Palacios, and R. Welsch, “Anomaly detection via a gaussian mixture model for flight operation and safety monitoring,” *Transportation Research Part C: Emerging Technologies*, vol. 64, pp. 45–57, 2016.
- [38] S. Das, B. L. Matthews, and R. Lawrence, “Fleet level anomaly detection of aviation safety data,” in *2011 IEEE Conference on Prognostics and Health Management*, IEEE, 2011, pp. 1–10.
- [39] T. G. Puranik and D. N. Mavris, “Anomaly detection in general-aviation operations using energy metrics and flight-data records,” *Journal of Aerospace Information Systems*, vol. 15, no. 1, pp. 22–36, 2018.
- [40] X. Olive and L. Basora, “Identifying anomalies in past en-route trajectories with clustering and anomaly detection methods,” in *ATM Seminar 2019*, 2019.
- [41] L. Song, R. Wang, D. Xiao, X. Han, Y. Cai, and C. Shi, “Anomalous trajectory detection using recurrent neural network,” in *International Conference on Advanced Data Mining and Applications*, Springer, 2018, pp. 263–277.
- [42] Y. Ji, L. Wang, W. Wu, H. Shao, and Y. Feng, “A method for lstm-based trajectory modeling and abnormal trajectory detection,” *IEEE Access*, vol. 8, pp. 104 063–104 073, 2020.
- [43] K. Kim and I. Hwang, “Terminal airspace anomaly detection using temporal logic learning,” in *ICRAT 2018 Third International Conference on Research in Air Transportation*, ICRAT Barcelona, Spain, 2018.
- [44] R. Deshmukh and I. Hwang, “Anomaly detection using temporal logic based learning for terminal airspace operations,” in *AIAA Scitech 2019 Forum*, 2019, p. 0682.
- [45] R. Deshmukh and I. Hwang, “Incremental-learning-based unsupervised anomaly detection algorithm for terminal airspace operations,” *Journal of Aerospace Information Systems*, vol. 16, no. 9, pp. 362–384, 2019.
- [46] R. Deshmukh, D. Sun, and I. Hwang, “Data-driven precursor detection algorithm for terminal airspace operations,” in *Proceedings of the 13th USA/Europe Air Traffic Management Research and Development Seminar, Vienna, Austria*, 2019, pp. 17–21.

- [47] J. L. Yepes, I. Hwang, and M. Rotea, “New algorithms for aircraft intent inference and trajectory prediction,” *Journal of guidance, control, and dynamics*, vol. 30, no. 2, pp. 370–382, 2007.
- [48] R. A. Paielli and H. Erzberger, “Conflict probability estimation for free flight,” *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 3, pp. 588–596, 1997.
- [49] W. Liu and I. Hwang, “Probabilistic trajectory prediction and conflict detection for air traffic control,” *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 6, pp. 1779–1789, 2011.
- [50] Z. C. Lipton, J. Berkowitz, and C. Elkan, “A critical review of recurrent neural networks for sequence learning,” *arXiv preprint arXiv:1506.00019*, 2015.
- [51] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [52] H. Papadopoulos, V. Vovk, and A. Gammerman, “Conformal prediction with neural networks,” in *19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007)*, IEEE, 2007, pp. 388–395.
- [53] R. Laxhammar and G. Falkman, “Online learning and sequential anomaly detection in trajectories,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 6, pp. 1158–1173, 2013.
- [54] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, “Comparing images using the hausdorff distance,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 15, no. 9, pp. 850–863, 1993.
- [55] I. Hwang, H. Balakrishnan, and C. Tomlin, “State estimation for hybrid systems: Applications to aircraft tracking,” *IEEE Proceedings Control Theory and Applications*, vol. 153, no. 5, 2006.
- [56] *Instrument procedures handbook*, U.S. Department of Transportation, Federal Aviation Administration (FAA), 2015.
- [57] S. George, G. Satapathy, V. Manikonda, F. Wieland, M. Refal, and R. Dupee, “Build 8 of the airspace concept evaluation system,” in *AIAA Modeling and Simulation Technologies Conference*, 2011.
- [58] E. Ray, “Order JO 7110.65 W,” *US Department of Transportation, Washington, DC*, 2015.

- [59] A. Bonzano, P. Cunningham, and C. Meckiff, “ISAC: A CBR system for decision support in air traffic control,” *Advances in Case-Based Reasoning*, 1996.
- [60] M. H. Nguyen and F. De la Torre, “Optimal feature selection for support vector machines,” *Pattern recognition*, 2010.
- [61] S. Maldonado, R. Weber, and J. Basak, “Simultaneous feature selection and classification using kernel-penalized support vector machines,” *Information Sciences*, 2011.
- [62] Y. Gal and Z. Ghahramani, “Dropout as a bayesian approximation: Representing model uncertainty in deep learning,” in *33rd International Conference on Machine Learning*, 2016, pp. 1050–1059.
- [63] G. Shafer and V. Vovk, “A tutorial on conformal prediction.,” *Journal of Machine Learning Research*, vol. 9, no. 3, 2008.
- [64] R. Laxhammar and G. Falkman, “Conformal prediction for distribution-independent anomaly detection in streaming vessel data,” in *Proceedings of the first international workshop on novel data stream pattern mining techniques*, 2010, pp. 47–55.
- [65] M. Lange, D. Zühlke, O. Holz, T. Villmann, and S.-G. Mittweida, “Applications of lp-norms and their smooth approximations for gradient based learning vector quantization.,” in *ESANN*, 2014, pp. 271–276.
- [66] F. Nielsen and K. Sun, “Guaranteed bounds on the kullback-leibler divergence of univariate mixtures using piecewise log-sum-exp inequalities,” *arXiv:1606.05850*, 2016.

VITA

Kwangyeon Kim received the Bachelor of Science and Master of Science degrees in aerospace engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2006 and 2008, respectively. He then started to pursue a doctorate degree in Aeronautics and Astronautics at Purdue University in 2013. His research focuses on application of machine learning to safety and efficiency in air traffic management.