HYBRID DATA-DRIVEN AND PHYSICS-BASED FLIGHT TRAJECTORY PREDICTION IN TERMINAL AIRSPACE

by

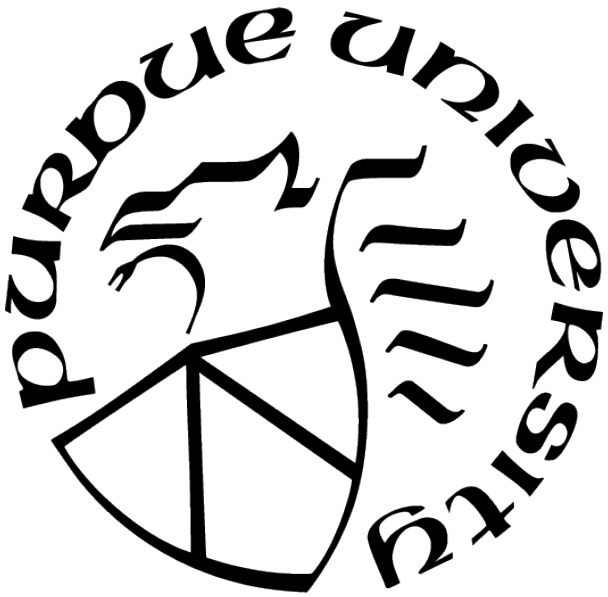
Hansoo Kim

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Master of Science



School of Aeronautics and Astronautics

West Lafayette, Indiana

May 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. Dengfeng Sun

School of Aeronautics and Astronautics

Approved by:

Dr. Gregory A. Blaisdell

This work is dedicated to my parents

ACKNOWLEDGMENTS

First of all, the completion of this work could not have been realized without the assistance of several people around me. I wish to express my utmost gratitude to my advisor, Professor Inseok Hwang, for his great deal of guidance to accomplish this thesis and develop ideas for hybrid trajectory prediction algorithm in Air Traffic Management (ATM). He is one of the smartest people and the most enthusiastic advisor I know. His persistence, insight and enthusiasm have been critical in making the researcher I am. His timely advice and scientific attitude have helped me to a very great extent to accomplish this thesis. I would like to deeply thank you all my committee members - Professor Martin Corless and Professor Dengfeng Sun - for their encouragement and advice. I am also grateful to my colleagues who have supported me during my graduate study. Especially, I want to express my special gratitude to my mentors during all stages of research: Kwangyeon Kim, Hong-cheol Choi, and Chuhao Deng. They provided me with everything. Especially, I deeply indebted to my mentor, Kwangyeon Kim. He has provided me with the tool that I needed to choose the right direction and successfully finish my thesis. I am also thankful to all of my fellow labmates in Flight Dynamics and Control/ Hybrid Systems Lab. They all provided valuable support and advice and helped me a lot. My experience in the lab was greatly enhanced. In addition, thanks are necessary for my friends. They have provided patient advice and valuable help throughout the research process. And their contributions are sincerely appreciated and gratefully acknowledged. Last but not the least, I would like to praise my parents – Mr. Kim and Mrs. Shin for their everlasting love and tolerance. It is because of their continuing encouragement that I could finish my thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

With the growing demand of air traffic, it becomes more important and critical than ever to develop advanced techniques to control and monitor air traffic in terms of safety and efficiency. Especially, trajectory prediction can play a significant role on the improvement of the safety and efficiency because predicted trajectory information is used for air traffic management such as conflict detection and resolution, sequencing and scheduling.

Recently, there have been extensive efforts for the development of trajectory prediction algorithms, which can be categorized into two approaches: (i) in physics-based approaches, a model describing the behaviors of an aircraft is developed based on the aircraft dynamics or governing physics, which is then used in Kalman filtering or its variants, and (ii) in data-driven approaches, collected flight data is used to learn a data-driven model that can be used to predict the future trajectory of an aircraft. Both of the approaches, however, have limitations: Without assistance of dataset, the physics-based approaches use only the aircraft dynamics for its trajectory prediction without correction by the measurement since no measurements will be available in the future; on the other hand, the data-driven methods do not explicitly use the aircraft dynamics at the current time (e.g., an aircraft is performing a coordinated turn).

In this work, we propose a new framework that can overcome these limitations by integrating the two methods, called hybrid data-driven and physics-based trajectory prediction. The proposed algorithm is applied to real air traffic surveillance data to demonstrate its performance. Results show that our new algorithm has a higher trajectory predicting accuracy than the two baseline methods, which could help enhance the safety and efficiency of air traffic operations.

1. INTRODUCTION

In this chapter, a literature survey of trajectory prediction algorithms is presented along with the motivation that propels us to propose a hybrid data-driven and physics-based trajectory prediction algorithm. This is followed contributions and objectives of this thesis.

1.1 Background and motivation

The global Air Traffic Management (ATM) system is one of the most complex and rapidly developing systems, in order to accommodate the growth of the air traffic demand which has doubled in recent 15 years [1]. The goal of air traffic control (ATC) is to control the aircraft's operation for the safety and efficiency of airspace, which consists of en-route, terminal, and surface operations, as shown in Figure 1.1. Among such components, the terminal airspace is the most complex and has the highest traffic density. According to the data published by Boeing [2], 75 percent of the fatal accidents have occurred in terminal airspace, which comprises only 6 percent of total flight time. To improve the efficiency and safety of ATM in terminal airspace, several decision supporting tools have been developed to help ATC's decision making processes. Among such tools, trajectory prediction plays an important role as the predicted trajectory information is used in ATC's tasks for the air traffic management purpose, such as conflict detection, sequencing, and scheduling. The algorithms proposed for trajectory prediction can be broadly categorized into the physics-based method and the data-driven method, as follows.

Physics-based method

The physics-based methods characterize the behavior of an aircraft based on its dynamics to predict the future trajectory by using the tools such as Kalman filtering or its variants. Liu and Hwang [3] proposed a stochastic linear hybrid system (SLHS) with a state-dependent transition model or a Markov transition model to represent the aircraft dynamics with flight modes such as constant velocity (CV) and coordinated turn (CT) modes. The SLHS is combined with flight intents to more accurately predict the aircraft's future states. Hu et

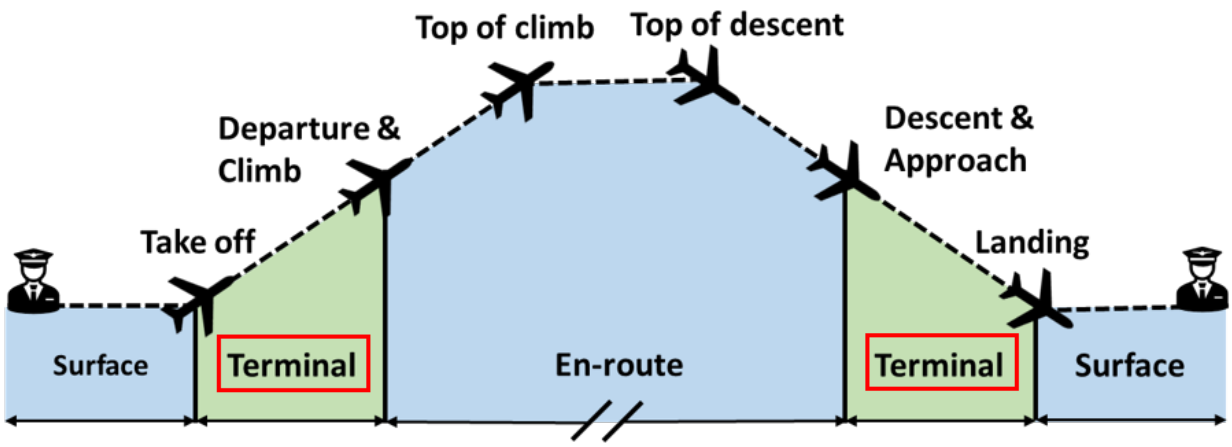


Figure 1.1. National Airspace System (NAS)

al. [4] introduced a trajectory prediction method using a stochastic differential equation to represent the aircraft motion perturbed by wind disturbances and approximated the distribution of the solutions to the stochastic differential equation as the evolution of a Markov chain. Yepes et al. [5] proposed an estimation method for a hybrid system and an intent inference algorithm, both of which were combined to predict future trajectory of aircraft. Zhang et al. [6] proposed an online four-dimensional trajectory prediction method by using a point-mass model with aircraft performance data for trajectory prediction, combined with a conformance monitoring and aircraft intent update process for more accurate trajectory prediction. Baklacioglu et al. [7] proposed an approach to predict the future trajectory based on aerodynamics and propulsive models, especially considering the effects of the compressible drag which occur above the critical Mach number. The models were used to predict the flight trajectory during the descent phase. Wiest et al. [8] proposed a probabilistic trajectory prediction based on Gaussian Mixture Models (GMM). The motion of a vehicle was modeled as a probability distribution inferred from previously observed motion patterns. The motion model was then used to predict the future position conditioned on the motion pattern observed at the current time. Ayhan et al. [9] proposed a stochastic trajectory prediction algorithm. The spatio-temporal characteristics of aircraft trajectories were represented in a set of four-dimensional cubes. For each 4D cube, the weather information of airspace was also considered by using Hidden Markov Model (HMM) for trajectory prediction.

Data-driven method

In the data-driven method, collected data is used to learn a model to predict the trajectory. Collective behavior of a trajectory pattern can be captured from vast amounts of data. In this regard, several data-driven methods using machine learning techniques have been developed. Duca et al. [10] proposed an algorithm based on a K-Nearest Neighbor classifier to predict routes in which the current status was takes as input and the matrix of probabilities of the next status over a grid was returned as output, using the features such as latitude, longitude, heading, speed over ground, and aircraft type. Xiao et al. [11] proposed a vehicle trajectory prediction method using support vector regression (SVR) based on the input from GPS for position information and on-board devices for in-vehicle driving information, such as steering direction. The parameters of SVR were optimally selected using a particle swamr optimization algorithm. Liu and Hansen [12] proposed an approach to predict the actual aircraft 4D trajectories using meteorological data and flight plans. GMMs were used to model aircraft trajectory, which were learned by using an encoder-decoder network. The weather information was represented using convolutional layers.

1.2 Objectives and contributions

Though many trajectory prediction algorithms have been proposed, these methods have limitations in their performance when used alone, and thus the accuracy of these methods could be relatively low, especially in the terminal airspace: if a data-driven model is used alone, the aircraft's current motion cannot be explicitly accounted for because the model is learned sorely from historical data; on the other hand, if a physics-based model is used alone, since no measurements will be available in the future time-steps, the trajectory of an aircraft is predicted by propagating the aircraft dynamics into the future, without correction by the measurements. In this regard, we propose a new trajectory prediction framework that combines the both approaches: a data-driven trajectory prediction model is learned from the historical flight data to represent the aircraft trajectories under a specific operating condition; and then, for incoming track points of an aircraft under monitoring, the data-driven model is used to predict the aircraft's future states, such as position and speed, which

serve as pseudo-measurements for a physics-based prediction method, such as Kalman filter or its variants, thereby incorporating the expected future behaviors (learned from data) with the current status or dynamics (from physics), as shown in Figure 1.2.

The proposed method, called Hybrid Data-driven and Physics-based Trajectory Prediction, is demonstrated with air traffic surveillance data from the repository of real historical flight datasets, comprising of departure and arrival operations at Incheon International Airport (ICN) and Gimpo International Airport (GMP) in South Korea. The demonstration is presented via two case studies: prediction for estimated time of arrival (ETA) and short-term trajectory prediction.

1.3 Organization

The rest of this thesis is organized as follows: In Chapter 2, we describe an architecture and a concept of the proposed hybrid trajectory prediction algorithm and discuss how to utilize the algorithm in detail for predicting aircraft trajectory and present background information of data source. In Chapter 3, through extensive analysis with actual flight data, the performance of the proposed hybrid trajectory prediction algorithm is demonstrated. Finally, in Chapter 4, the summary and possible extensions are discussed.

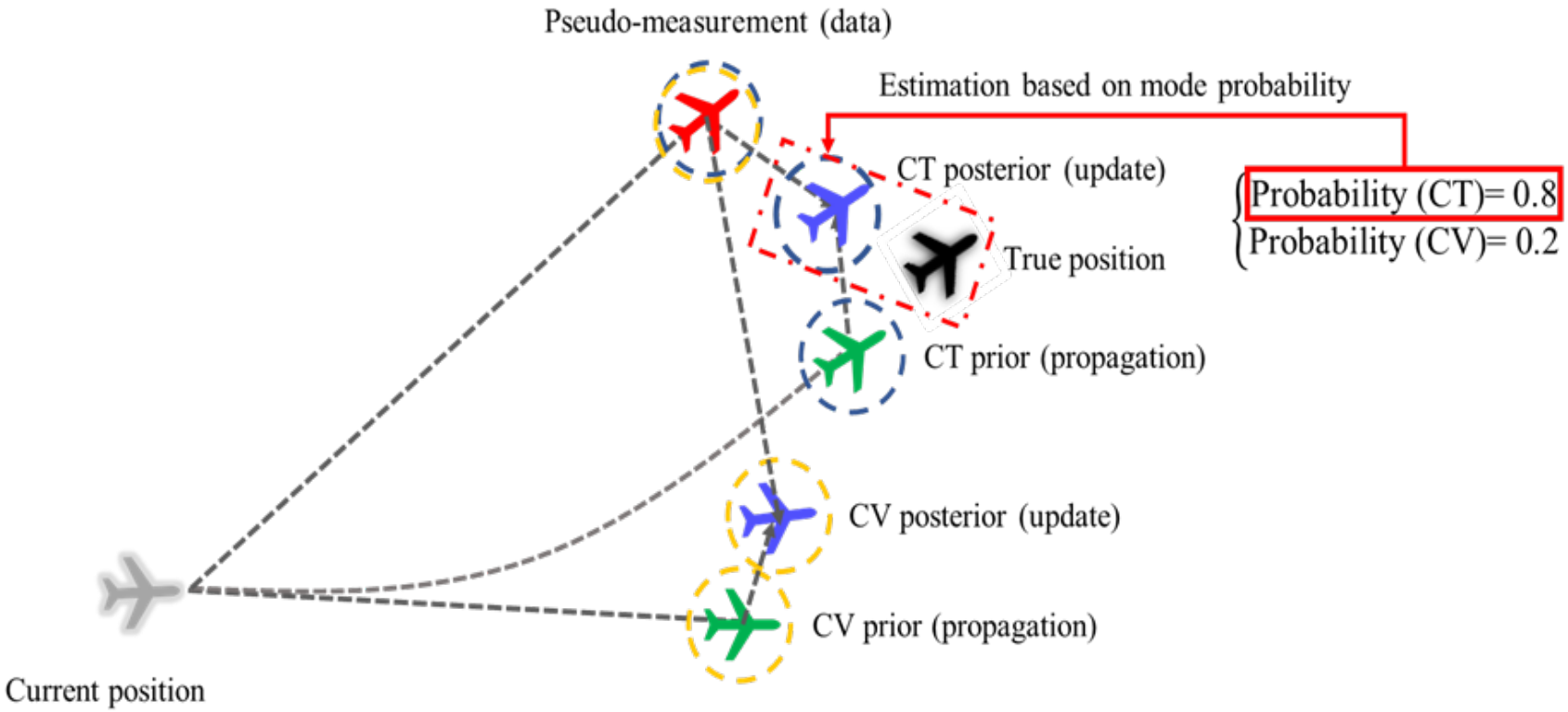


Figure 1.2. Concept of hybrid data-driven and physics-based trajectory prediction

2. PROPOSED TRAJECTORY PREDICTION METHODOLOGY

In this chapter, we present the development of an algorithm for hybrid data-driven and physics-based trajectory prediction method. Figure 2.1 represents the architecture of the proposed algorithm, which consists of data preparation, learning of a data-driven prediction model, and prediction by combining the learned data-driven prediction model and a physics-based prediction model.

- Data preparation: For the flights classified as departure or arrival at GMP or ICN airports, we perform trajectory clustering using a density-based method, called Density-based Spatial Clustering of Applications with Noise (DBSCAN) [13]. By examining each group with similar properties individually, trajectory prediction can be made more efficiently.

- Learning: For each trajectory cluster identified by trajectory clustering, the proposed framework first learns a data-driven trajectory prediction model by using the Long Short-Term Memory (LSTM) [14] which is a special kind of neural network well suited for time-series data due to its recurrent structure that can well represent temporal dependency and spatial patterns, thus demonstrating high predictability for time-series data.

- Prediction: For an aircraft under monitoring, its future states are then predicted by the learned network (data-driven), combined with a Kalman filter-based estimation technique (physics-based). At time-step T', the aircraft's track points up to T', $\{z_t\}_{t=0}^{T'}$ where z_t is an observed track point at time-step t, are used to predict the trajectory for future N time-steps, $\{\hat{z}_t\}_{t=T'+1}^{T'+N}$ where $\hat{z}_t$ is a predicted track point at time-step t.

  - Data-driven Prediction: At $t = T', \ldots, T' + N - 1$, the predicted state $\hat{z}_t$ with its covariance $\hat{R}_t$, denoted as $\left(\hat{z}_t, \hat{R}_t\right)$, is first fed into the

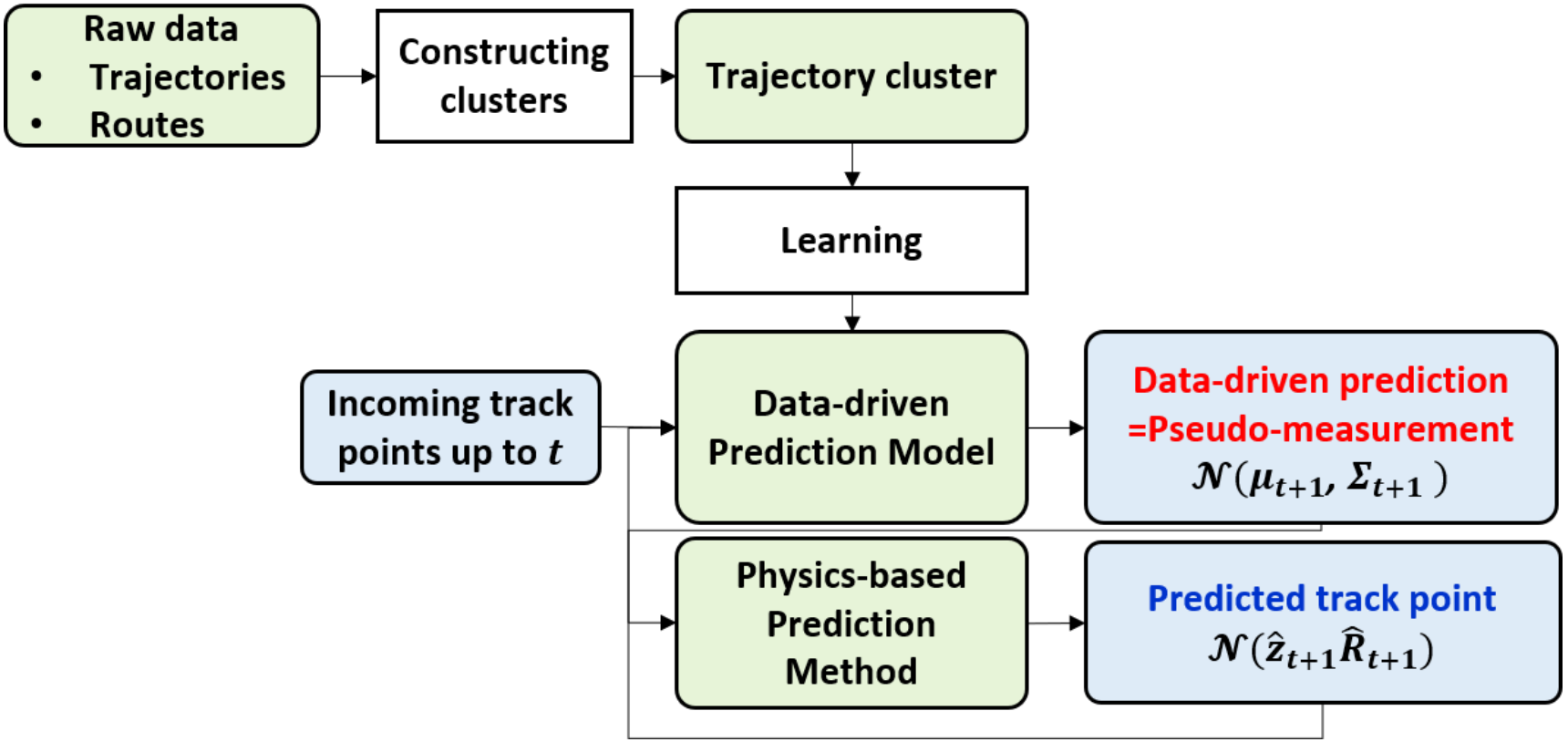


Figure 2.1. Architecture of the proposed hybrid data-driven and physics-based trajectory prediction method

learned data-driven trajectory prediction model to generate a data-driven prediction $(\mu_{t+1},\ \Sigma_{t+1})$ at $t+1$.

– Physics-based Prediction: Given the data-driven prediction $\left(\hat{z}_t, \hat{R}_t\right)$ at t, a physics-based prediction method, such as Kalman filter or its variants, then computes the prediction $(\hat{z}_{t+1}, P_{t+1})$ at $t+1$ by combining physics-based models for propagation and the data-driven prediction as a pseudo-measurement $(\mu_{t+1},\ \Sigma_{t+1})$ for the measurement update (or correction of the propagated prediction by the measurement).

2.1 Data preparation

This section describes the inputs and data pre-processing required for effective application of the proposed algorithm. To understand the air traffic operations in Korea, we have performed a preliminary study of aviation data, which consists of Aeronautical Information Publication (AIP) data [15] (which includes airspaces and routes) and Automatic Dependent Surveillance-Broadcast (ADS-B) data [16] (or recorded flight trajectories), as shown in figure 2.2.

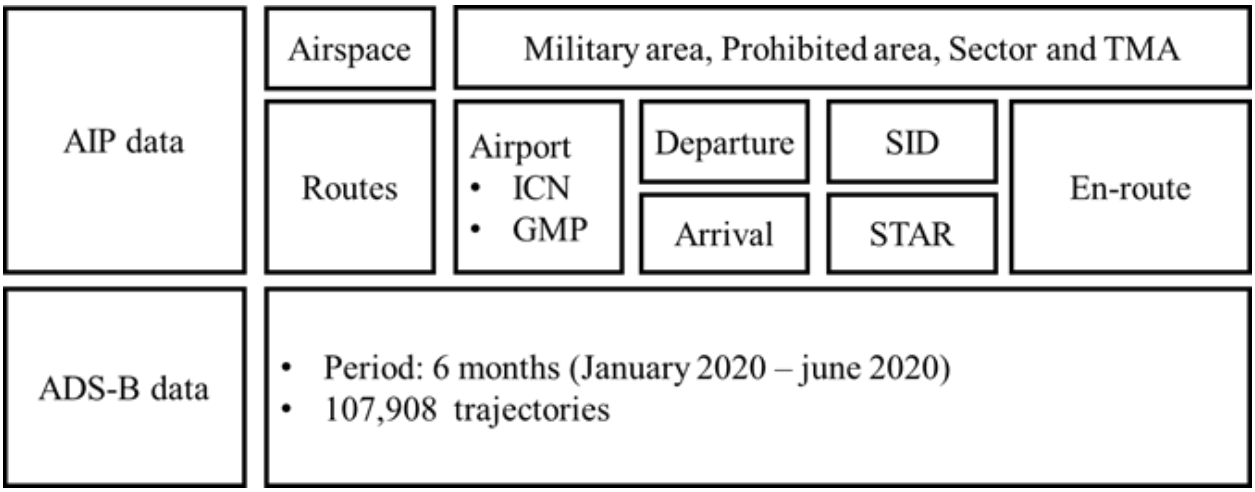


Figure 2.2. Aviation data used for this research

The airspace information including military areas, sectors, terminal maneuvering areas (TMA), prohibited areas is visualized in Figure 2.3,2.4,2.5,2.6.

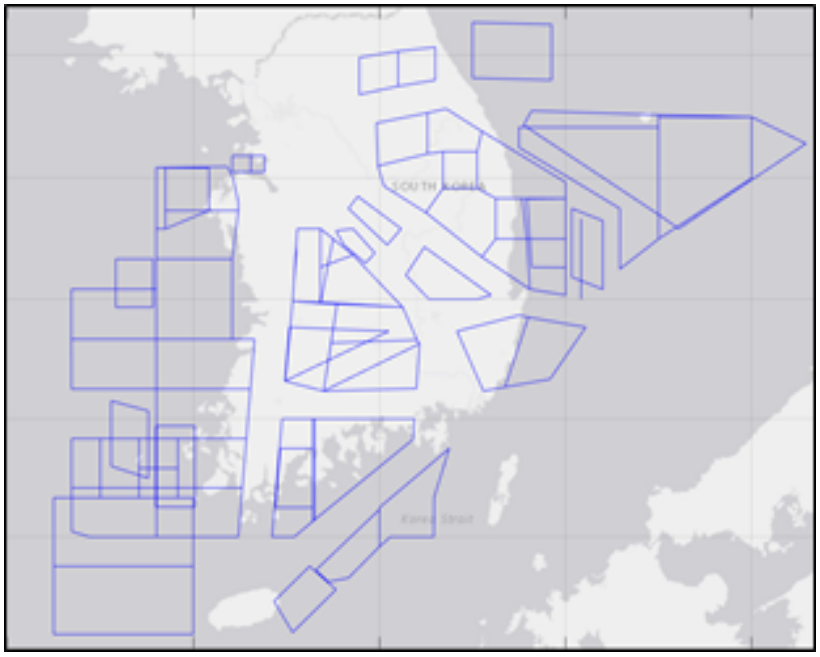


Figure 2.3. Airspace information of military area

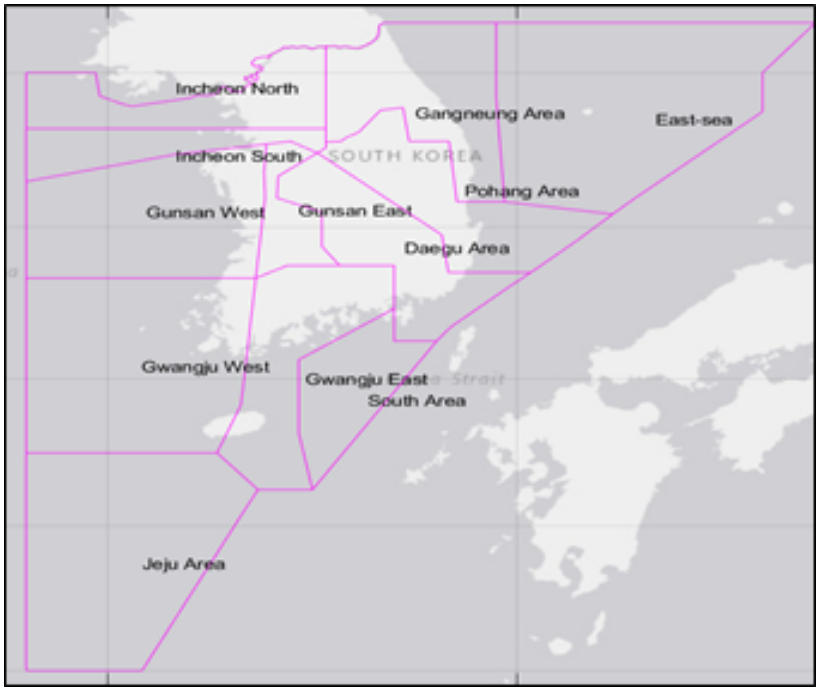


Figure 2.4. Airspace information of sector

With Automatic Dependent Surveillance-Broadcast (ADS-B) data, we demonstrate the proposed hybrid trajectory prediction algorithm. The collected ADS-B was recorded from

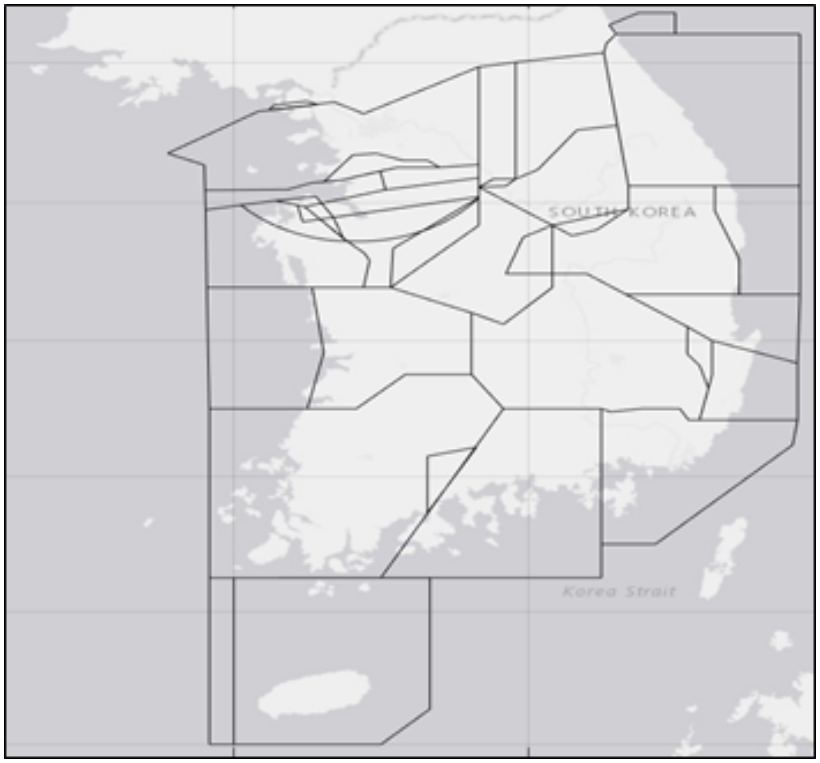


Figure 2.5. Airspace information of TMA

Figure 2.6. Airspace information of prohibited area

January to June in 2020, around the two major airports in South Korea, Incheon International Airport (ICN) and Gimpo International Airport (GMP) as shown in Table 2.1.

Table 2.1. Specification of ADS-B data

	Period	Detection range	Sampling rate	Number of flights
ADS-B	January to June in 2020	200nm	1 second	107,908

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 as shown in Figure 2.7. Each trajectory traveled for around 15 - 20 minutes in terminal airspace.

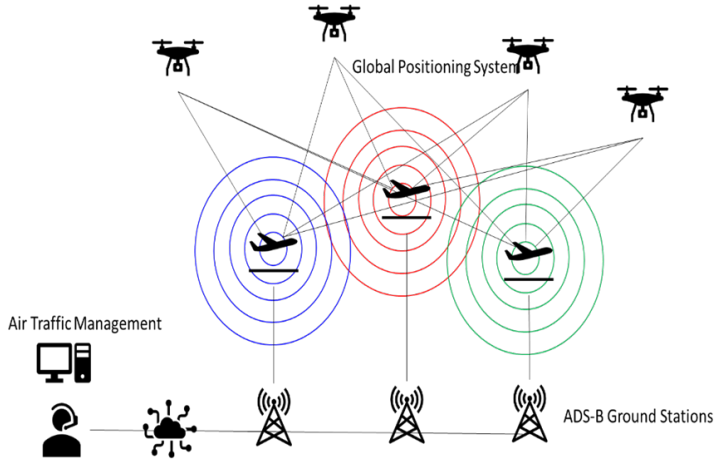


Figure 2.7. An ADS-B system

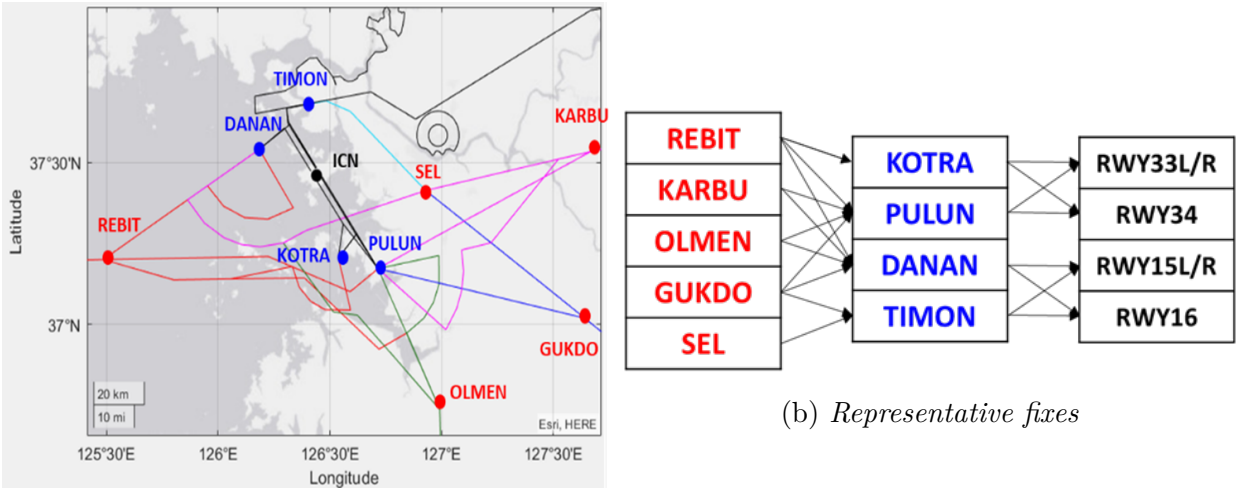
2.1.1 Routes

In the given aviation dataset, a route is represented as a sequence of fixes, e.g., 'RNAV REBIT 1N', as shown in Table 2.2, where the horizontal position (latitude and longitude) of each fix can be found in a separate file, e.g., REBIT is located at 37.20083300 125.48694400. A corresponding altitude, e.g., 15,000 feet for REBIT, together defines the three-dimensional position of a fix. A segment connecting two consecutive fixes (Fix 1 and Fix 2) is characterized by a lateral limit (allowed deviation in cross-track) (nmi) and direction ('Forward' if only a travel from Fix 1 to Fix 2 is allowed; 'Reverse' for the other way around; and 'Both' if no restrictions in travel directions). It is noted that some segment allows Direct-To, which means that taking a shortcut is allowed, e.g., for 'SI931-SI932' segment, an aircraft on this segment can directly go to its last fix, 'DANAN.' We have visualized all the routes on

Table 2.2. An example of a route given in the aviation data ('RNAV REBIT 1N')

Fix1	Fix2	Altitude 1 (ft)	Altitude 2 (ft)	Lateral limit (nmi)	Direction	Additional
REBIT	ALTON	15,000	13,000	1	Forward	-
ALTON	SI931	13,000	9,000	1	Forward	-
SI931	SI932	9,000	9,000	1	Forward	Direct to: DANAN
SI932	SI933	9,000	9,000	1	Forward	Direct to: DANAN
SI933	SI934	9,000	9,000	1	Forward	Direct to: DANAN
SI934	SI935	9,000	9,000	1	Forward	Direct to: DANAN
SI935	DANAN	9,000	5,000	1	Forward	-

the geographical map, as shown in Figure 2.9 for the arrival routes to ICN as an example, (the rest of the routes are given in Appendix) where the prohibited areas are represented as black lines. It is noted that there exists a set of representative fixes: a flight arriving at ICN must pass through either of the fixes in red (REBIT, OLMEN, GUKDO, KARBU, and SEL) and then either of the fixes in blue (DANAN, TIMON, PULUN, and KOTRA). An aircraft reaching either of these blue fixes then follows a straight line in the horizontal plane to land at ICN airport in which there are six runways (15L/R, 16, 33L/R, and 34). Since DANAN and TIMON are located at the Northwest of ICN, an aircraft reaching either of them uses runways 15L/R and 16, and similarly runways 33L/R and 34 are used for the flights passing through KOTRA and PULUN. In Figure 3 (right), the structure of the routes is shown using the representative fixes. The red fixes can be viewed as locations where a flight enters a terminal airspace from en-route airspace, thus starting its arrival, and the blue fixes as locations where a flight begins its (final) approach.



(a) *Arrival routes*

(b) *Representative fixes*

Figure 2.8. Arrival routes to ICN and representative fixes

In Figure 2.9, the arrival routes passing through each of the red fixes along with the approach routes are presented and the number of routes for each pair of the red and blue fixes are shown in Table 2.3, from which the followings are observed:

- A route from each red fix provides (i) at least one path to either of the blue fixes in the Northwest of ICN (DANAN and TIMON), thus to runways 15L/R and

16 and (ii) at least one path to the Southeast blue fixes (KOTRA and TIMON), thus to runways 33L/R and 34. This implies that the no matter what red fixes a flight passes through, it can reach ICN from both the Northwest and Southeast.

- The routes from DANAN (Northwest) and PULUN (Southeast) provide a Direct-To option, which is represented as the arcs in Figure 2.9. Specifically, there are two routes to PULUN from each red fix where one is with Direct-To arcs and the other is no such arcs (thus it is rather 'straight').
- A blue fix KOTRA is used only for the routes from REBIT.
- A blue fix TIMON is used only for the route from SEL and vice versa. This route is close to the prohibited area, which therefore requires a special attention to avoid violating this area.

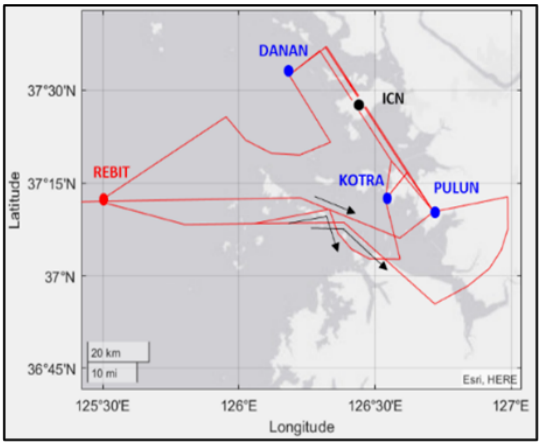
2.1.2 Trajectories

In the given aviation data, the ADS-B data includes the flight trajectories recorded around ICN and GMP during the period of January 2020 – June 2020 (6 months). For each flight trajectory, the recorded variables are four-dimensional information (time, latitude, longitude, and altitude), speed (the horizontal and vertical speeds), and track angle, as shown in Table 2.4.

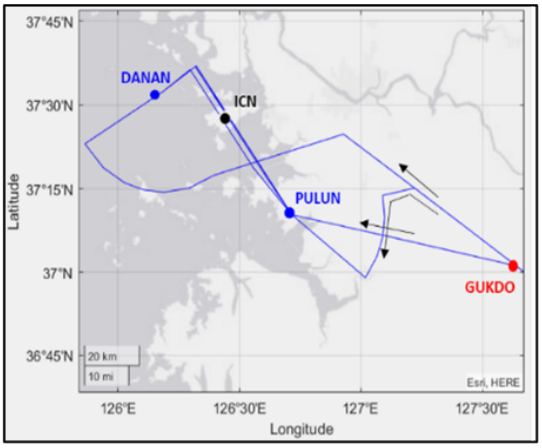
There are 107,908 flights in total. As an example, the total of 1,758 flights recorded on January 10, 2020 are visualized in Figure 2.10 where the red and blue dots represent ICN and GMP, respectively.

Table 2.3. Number of routes for each pair of red and blue fixes

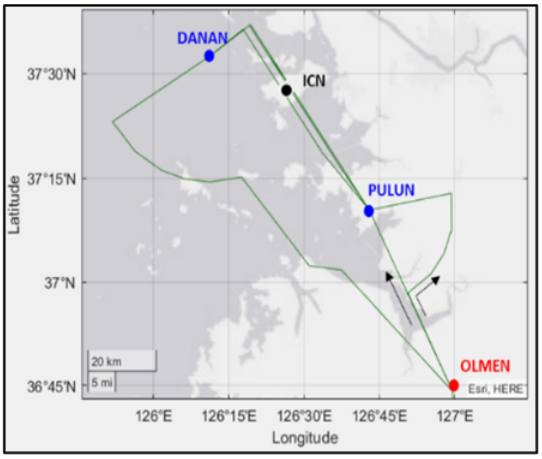
From red fix to blue fix	PULUN	KOTRA	DANAN	TIMON	TOTAL
REBIT	2	1	1	0	4
GUKDO	2	0	1	0	3
OLMEN	2	0	1	0	3
KARBU	2	0	1	0	3
SEL	0	0	0	1	1



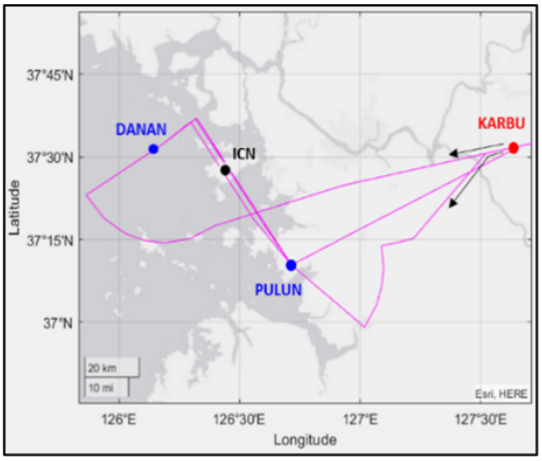
(a) *ICN arrival routes through REBIT*



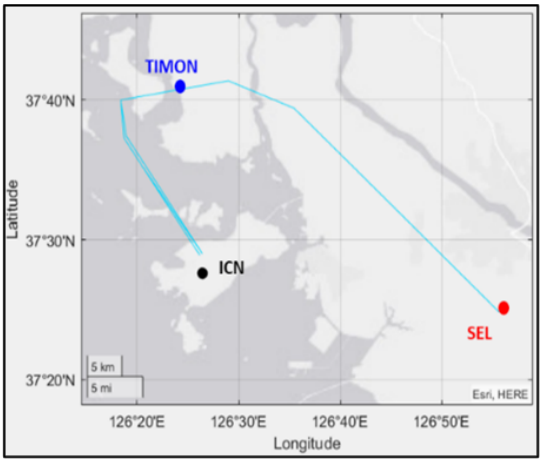
(b) *ICN arrival routes through GUKDO*



(c) *ICN arrival routes through OLMEN*



(d) *ICN arrival routes through KARBU*



(e) *ICN arrival routes through SEL*

Figure 2.9. ICN arrival routes through representative points

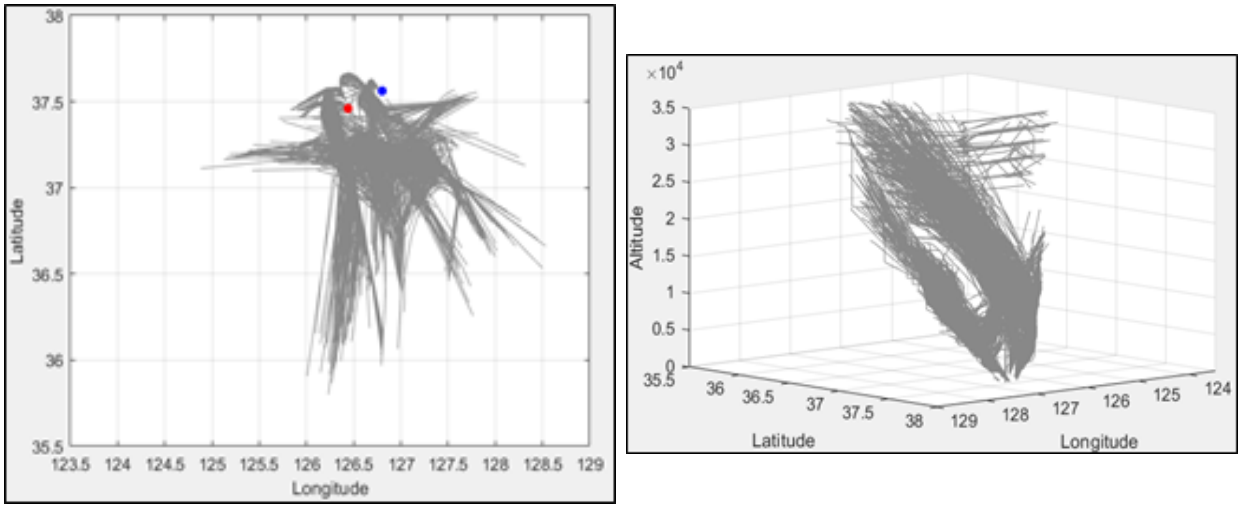


Figure 2.10. All flight trajectories on January 10, 2020 (left: the horizontal plane; right: three-dimensional space)

It is noted that the given data does not provide any information about the flight plan of each flight trajectory, such as (i) whether it is departure or arrival, (ii) which airport is its origin (for departure) or destination (for arriva), and/or (iii) which specific route is used for departure or arrival.

- To identify (i) for a flight trajectory, we have used the difference in altitude between the first and last timestamps, $\delta h = h(T) - h(0)$, where h is altitude, 0 and T are the first and last timestamps, and δh is the difference. We classify a flight trajectory as departure if $\delta h > 0$ and as arrival if $\delta h < 0$. However, there could exist some overflights that pass over the area of our interest, which are neither departure nor arrival. To filter them out, any flights whose minimum

Table 2.4. A sample of ADS-B data (January 7, 2020)

Time	Lat (deg)	Lon (deg)	Alt (ft)	Hspd (knots)	Vspd (fpm)	Trk (deg)
23:46:41	36.9655	127.7680	19,075	418	-1,088	307
23:46:42	36.9667	127.7661	19,050	419	-1,088	308
23:46:44	36.9690	127.7625	19,025	421	-1,152	309
23:47:19	37.0150	127.6976	18,425	426	-960	313
23:47:20	37.0162	127.6959	18,400	426	-960	313
23:47:31	37.0204	127.6901	18,225	425	-1,024	313
23:47:34	37.0348	127.6705	18,175	425	-960	314

altitude is above some threshold are removed where the threshold was set as 8,000 ft by inspection, resulting in classifying overflights. For the rest 107,908 flights, we have identified 54,301 arrival flights and 53,607 departure flights from the altitude difference.

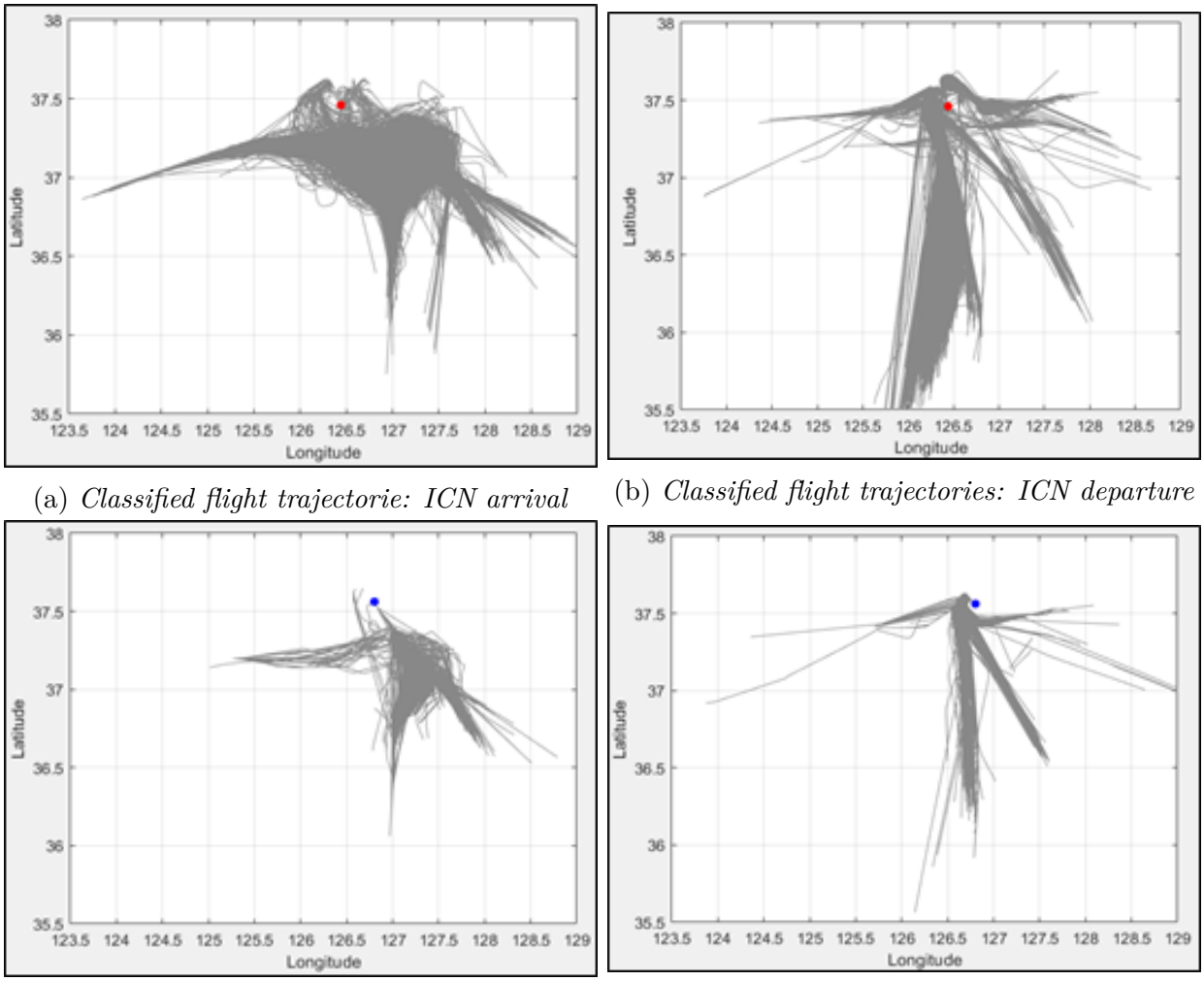
- For (*ii*), the distances between an airport (ICN or GMP) and the first/last horizontal position for departure/arrival are used to identify the origin/destination, resulting in Table 2.5 and Figure 2.11. It is observed that:
  - The numbers of arrival and departure flights are comparable each other for each airport, as well as in total (the number of arrival flights is slightly higher than that of departure).
  - The number of flights from/to ICN is about 1.8 times greater than that of GMP, possibly due to ICN's higher demand and capacity, which therefore leads to more complex operations at ICN for both arrival and departure.
  - For both airports, the arrival flights are seemingly more complex than the departure flights, due to its inherent complexity such as sequencing and spacing.

Table 2.5. Distribution of the number of flights along departure/arrival and corresponding origin/destination

Type	ICN	GMP	Total
Arrival	34,829	19,472	54,301
Departure	34,261	19,346	53,607
Total	69,090	38,818	107,908

2.1.3 Trajectory clustering

Since airspace are highly structured based on the routes (or flight plans), flight trajectories in the ADS-B data can be grouped into a set of clusters (similar trajectories that followed a



(a) *Classified flight trajectorie: ICN arrival*

(b) *Classified flight trajectories: ICN departure*

(c) *Classified flight trajectories: GMP arrival*

(d) *Classified flight trajectories: GMP departure*

Figure 2.11. Classified flight trajectories for the entire recording period (red and blue dots are ICN and GMP, respectively)

specific route). By examining each group with similar properties individually, data analysis and anomaly detection can be made more efficiently and accurately. In this sense, the flight trajectories can be divided into clusters by using a clustering algorithm such as DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [17], which numerically finds clusters from flight trajectories (not using routes information) based on a density-based method. In this regard, for the flights identified as ICN departure or arrival, we performed trajectory clustering by using a density-based method (DBSCAN). As shown in Figure 2.12, 23 different trajectory clusters were identified. It is noted that using a clustering algorithm

is numerically efficient but the routes information is not used in the process, thus requiring a post-analysis to explicitly identify the route used for a trajectory.

It is noted that the objective of data preparation is to group similar trajectories as accurately as possible to effectively reveal hidden behaviors or capture the pattern from trajectories. Therefore, the quality of trajectory clustering would impact the performance of a trajectory prediction.

2.2 Data-driven trajectory prediction method

In this section, we investigate the data-driven method for trajectory prediction, which consists of learning and prediction.

2.2.1 Data-driven trajectory prediction method: Learning

Recurrent Neural Network (RNN)

In the air traffic surveillance data, a flight trajectory is represented as a temporal sequence. To efficiently handle such sequential data, the Recurrent Neural Network (RNN) [18], [19], a variant of Neural Network, has been proposed by introducing a recurrent mechanism to keep a memory of the information in previous time-steps, as shown in Figure 2.13. Suppose that an input time-series $\{x_t\}_{t=0}^{T}$ and an output time-series $\{y_t\}_{t=0}^{T}$ are given. At time-step t, the RNN uses the current input element x_t, as well as the previous elements $x_0, \ldots, x_{t-1}$ to predict the current output element $\hat{y}_t$. The previous elements can be accounted for by using a hidden state, a_t, computed based on x_t and the hidden state at the previous time-step, a_{t-1}, so that it can contain all the previous information (or memory). The output at time-step t is then computed from the hidden state a_t as

$$a_t = g_a\left(w_{aa}a_{t-1} + w_{ax}x_t + b_a\right) \tag{2.1}$$

$$\hat{y}_t = g_y\left(w_{ya}a_t + b_y\right) \tag{2.2}$$

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, to be computed from the data by

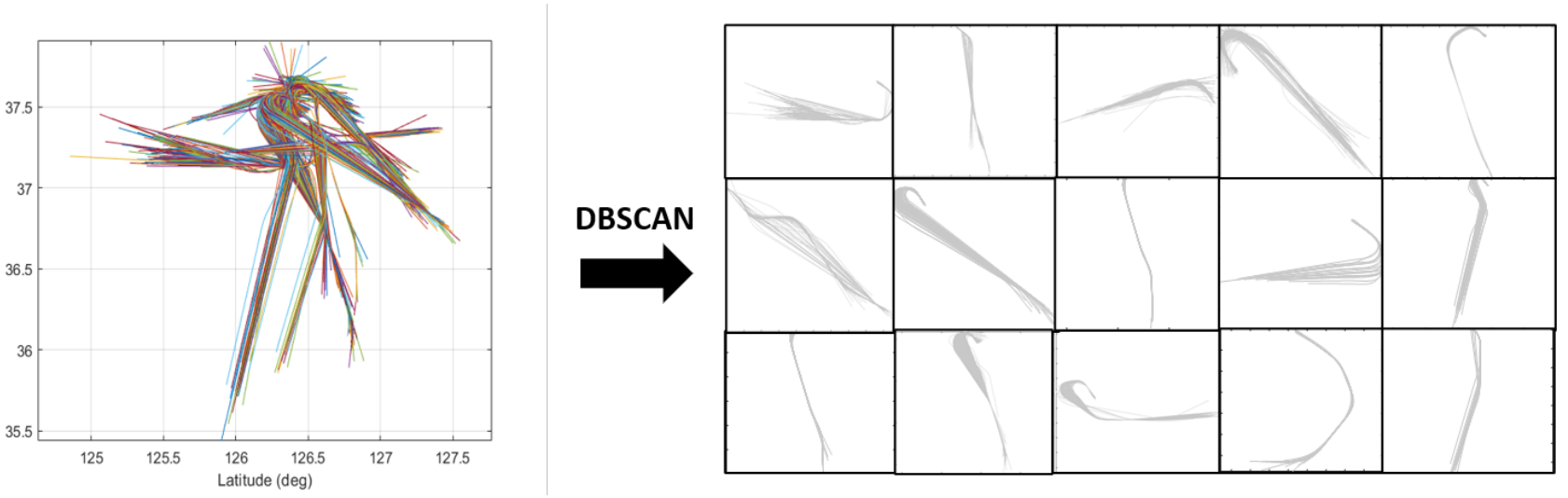


Figure 2.12. Results of using DBSCAN for ICN trajectories

minimizing the difference between the predicted output $\hat{y}_t$ and the (true) output y_t for all $t = 0, \cdots, T$.

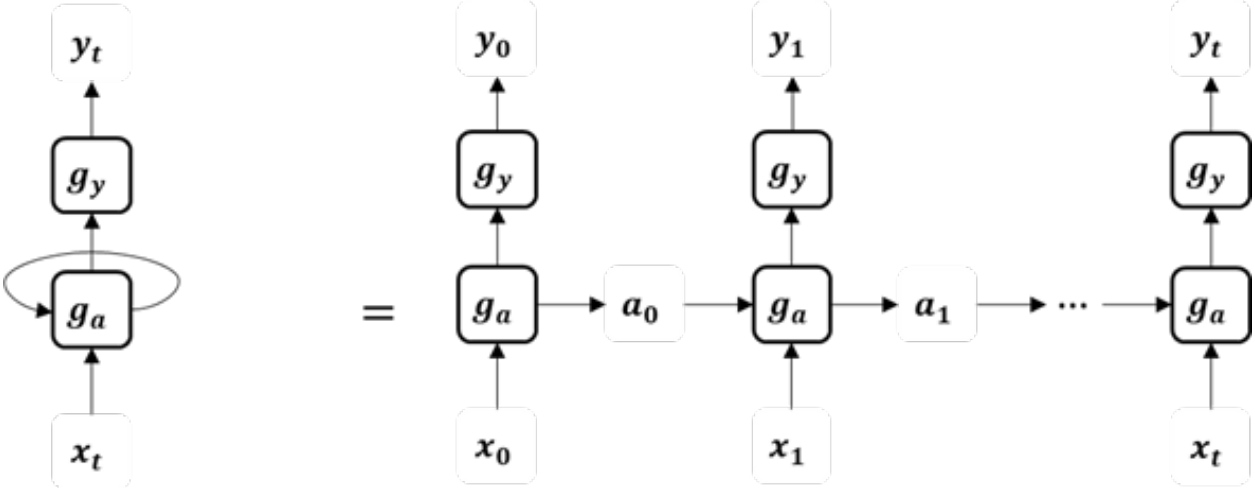


Figure 2.13. Structure of recurrent neural networks (RNN)

Long short-term memory

It has been shown that the standard RNN cannot learn long-term temporal dependency in time-series data due to the problem known as gradient vanishing/exploding which occurs during the learning of the parameters w's and b's [20], [21]. To address this issue, a variant of the RNN, called, Long Short-Term Memory (LSTM), has been proposed [22] by introducing the memory cell (c), as shown in Figure 2.14. In LSTM, the current memory at the current time-step t is obtained by fusing the previous memory c_{t-1} and the current *pseudo* memory cell $\tilde{c}_t$,

$$c_t = \Gamma_f c_{t-1} + \Gamma_u \tilde{c}_t \tag{2.3}$$

where $\Gamma_f \in [0, 1]$ and $\Gamma_u \in [0, 1]$ are the *forget* and *update* gates, respectively, which are given as

$$\Gamma_{(\cdot)} = g_{(\cdot)} \left(w_{(\cdot)a} a_{t-1} + w_{(\cdot)x} x_t + b_{(\cdot)} \right) \text{ for } (\cdot) \in \{u, f\} \tag{2.4}$$

and the pseudo memory cell is given as

$$\widetilde{c}_t = g_{\widetilde{c}}(w_{\widetilde{c}a}a_{t-1} + w_{\widetilde{c}x}x_t + b_a) \tag{2.5}$$

The current hidden state a_t is then updated by using the current memory c_t,

$$a_t = \Gamma_o \; g_a \; (c_t \;) \tag{2.6}$$

where $\Gamma_o = g_o\,(w_{oa}a_{t-1} + w_{ox}x_t + b_o) \in [0,1]$ is the *output* gate and g_a is an activation function. Finally, the output y_t is computed from the current hidden state

$$y_t = g_y\,(w_{ya}a_t + b_y) \tag{2.7}$$

In the proposed framework shown in Figure 2.1, the data-driven prediction model is learned by using the LSTM, as follows:

- A data-driven prediction model is learned from a large amount of data to predict one-step ahead track point given a stream of track points up to the current time. In this regard, the input and output are constructed with one-step shift, as shown in Figure 2.15.
- The parameters in LSTM (w's and b's) are computed to minimize the Mean Square Error (MSE) between the predicted output $\widetilde{z}_t$ and the actual output z_t for $t = 1, \ldots, T$ by using Backward Propagation Through Time (BPTT) algorithm [23].
- We stack multiple layers of LSTMs, which enables more complex representation of time-series data, thus improving the accuracy in time-series prediction.

2.2.2 Data-driven trajectory prediction method: Prediction

Figure 2.16 demonstrates how the learned data-driven trajectory prediction model works. Suppose that a trajectory cluster consists of l trajectories, $\left\{Z^{(i)}\right\}_{i=1}^{l}$ where $Z^{(i)}$ is an i-th

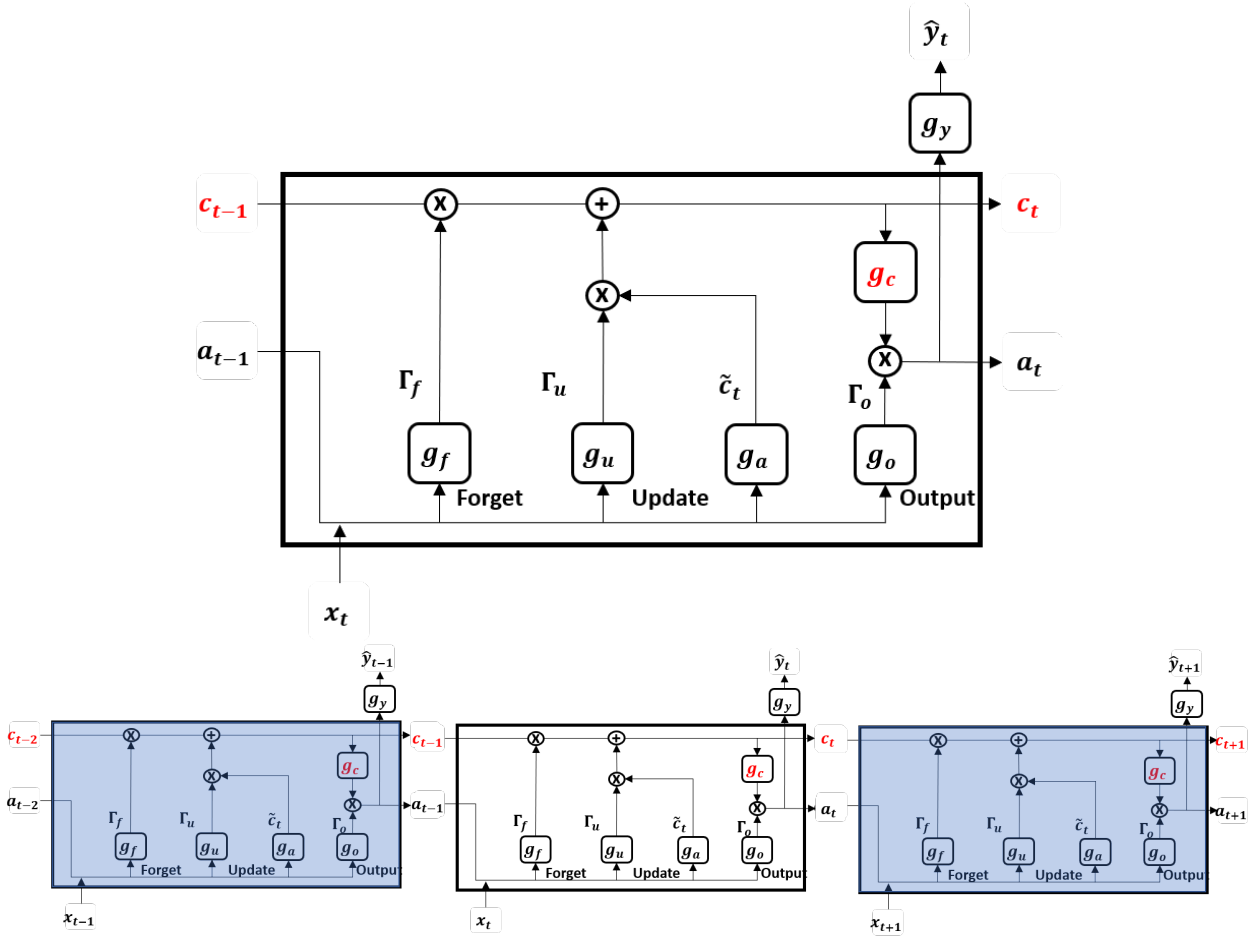


Figure 2.14. Structure of Long Short-Term Memory (LSTM)

trajectory in the cluster. A data-driven trajectory prediction model is learned (i.e., all the parameters w's and b's are computed) from the cluster to represent the aircraft trajectories under a specific operating condition. For incoming track points of an aircraft under monitoring, at time-step T', the data-driven model is used to predict the aircraft's states, such as position and speed, $(\mu_{T'+1},\ \Sigma_{T'+1})$, using the aircraft's track points, $\{z_t\}_{t=0}^{T'}$, observed up to current time-step T'. The data-driven prediction $(\mu_{T'+1},\ \Sigma_{T'+1})$ is then fed into the physics-based prediction model as a pseudo-measurement, and this process is repeated up to future N time-steps.

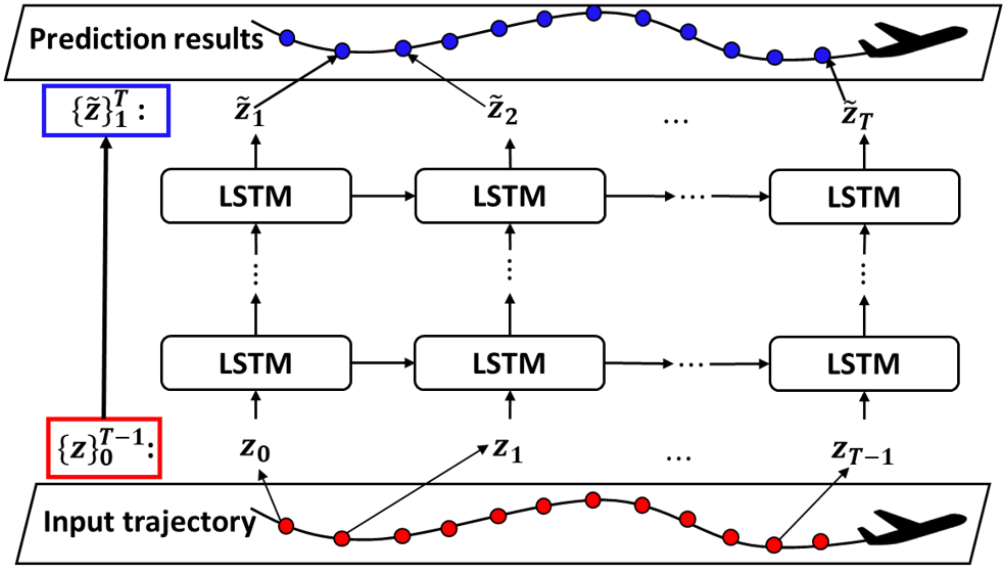


Figure 2.15. The proposed LSTM-based learning framework

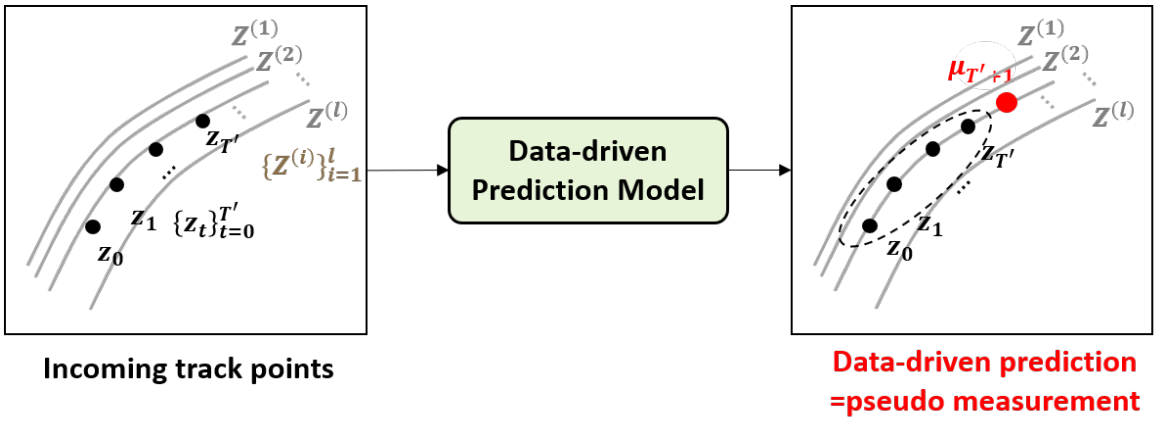


Figure 2.16. Data-driven prediction method

2.3 Physics-based trajectory prediction

The data-driven prediction model can well represent the behavior of aircraft in a specific cluster. Using the data-driven prediction as a pseudo-measurement, we propose to use an estimation algorithm, which consists of (i) the propagation of the aircraft states (such as position and speed) through dynamics, and (ii) the correction of the propagated states by the pseudo-measurement. To accurately account for the aircraft's current status, its dynamics is modeled as the stochastic linear hybrid system (SLHS) [24],[25] in which

- The discrete dynamics describes the transitions in the flight modes, such as Constant Velocity (CV) and Coordinated Turn (CT) modes; and
- The continuous dynamics represents how the aircraft's states evolve along time in each mode.

2.3.1 Continuous Dynamics

For each discrete state or mode $q(t) \in \mathbb{Q} := \{1, \ldots, n_d\}$ where n_d is the total number of modes, the SLHS model is given by:

$$x(t+1) = A_{q(t)}x(t) + E_{q(t)}w_{q(t)}(t) \tag{2.8}$$

$$z(t) = C_{q(t)}x(t) + v_{q(t)}(t) \tag{2.9}$$

where, at a given time-step t, $x(t) \in \mathbb{R}^n$ is the continuous state, $z(t) \in \mathbb{R}^p$ is the measurement, and $w_{q(t)}(t)$ and $v_{q(t)}(t)$ are the process noise and the measurement noise, assumed to be zero-mean white Gaussian noises with the covariances $Q_{q(t)}$ $R_{q(t)}$, respectively. The system matrices $A_{q(t)}$, $E_{q(t)}$ and $C_{q(t)}$ are with proper dimensions for each mode $q(t)$.

2.3.2 Discrete Dynamics

To represent the transition between the modes, several hybrid estimation algorithms have been proposed, including:

- The Interacting multiple model (IMM) [26], [27]: constant mode transition probability matrix;
- The residual-mean interacting multiple model (RM-IMM) [28]–[30]: constant mode transition probability matrix with the use of the residual mean of each mode; and
- The state-dependent-transition hybrid estimation (SDTHE) [24], [31]: mode transitions based on a set of stochastic guard conditions

We use the residual-mean interacting multiple model for the following reasons:

- In SDTHE, a transition between the modes is represented as a linear inequality guard condition using flight plans. Since such information is not readily available for our application, we have extracted flight plan from the data by finding breakpoints to use as the guard conditions. However, even for a single cluster, a single guard condition cannot well represent the transition due to the wide variations in, e.g., a turning point.

- In RM-IMM, the difference between the likelihoods of the correct and other modes is increased by using the residual mean of each mode. This leads to more distinction in mode probabilities, thereby yielding more accurate mode estimate, as well as continuous state estimates.

In RM-IMM, the flight mode transitions is represented by a constant mode transition probability matrix,

$$\Pi = [\pi_{ij}]_{i,j=1,\ldots,n_d} \tag{2.10}$$

where π_{ij} is the constant mode transition probability from mode i to mode j and $\sum_{j=1}^{n_d} \pi_{ij} = 1$ for $i \in Q$.

Suppose $Z^t = z(1), z(2), \ldots, z(t)$ is the measurement sequence up to time-step t. Let $\Pr[\cdot|\cdot]$ denote a conditional probability density function (pdf). Assume that, at time-step t, we have the followings for all the modes $i \in Q$:

- The probabilities of the flight mode $\alpha_i(t) := \Pr[q(t) = i|Z^t]$; and

- a continuous state estimate $\hat{x}_i(t)$ with its covariance $P_i(t)$ conditioned on the measurement Z^t and the event $q(t) = i$.

As shown in Figure 2.17, at time-step $t+1$, a pseudo-measurement $z(t+1)$ is taken as an input from the data-driven trajectory prediction model to update the continuous state estimate and its covariance as:

$$p(x(t+1) = x|\, Z^{t+1}) = \mathcal{N}(x;\ \hat{x}(t+1), P(t+1)) \tag{2.11}$$

where

$$\hat{x}(t+1) = \sum_{j=1}^{n_d} \hat{x}_j(t+1)\,\alpha_j(t+1) \tag{2.12}$$

$$P(t+1) = \sum_{j=1}^{n_d} \left\{ P_j(t+1) + [\hat{x}_j(t+1) - \hat{x}(t+1)]\,[\hat{x}_j(t+1) - \hat{x}(t)]^T\,\alpha_j(t+1) \right\} \tag{2.13}$$

$$\alpha_j(t+1) = \Pr\left(q(t+1) = j | Z^{t+1}\right) \tag{2.14}$$

In what follows, for $j = 1, \ldots, n_d$, the steps for obtaining the mode-conditioned continuous state estimate $\hat{x}_j(t+1)$ and its covariance $P_j(t+1)$ (Step 2 in the below), the (posterior) mode probability $\alpha_j(t+1)$ (Step 3 in the below), and final output $\hat{x}(t+1)$ and $P(t+1)$ are described (Step 4 in the below).

Step 1: Mixing

In the SLHS, the number of mode histories grows exponentially along time. To address this issue, a method called mixing [32] is used by computing the initial conditions of the state $x_{j0}(t)$ and its covariance $P_{j0}(t)$, which are fed into each Kalman filter j, as:

$$\begin{aligned}
\hat{x}_{j0}(t) &:= \mathbb{E}\left[x(t)\,|q(t+1) = j, Z^t\right] = \sum_{i}^{n_d} \hat{x}_i(t)\,\gamma_{ji}(t) \\
P_{j0}(t) &:= \mathbb{E}\left[(x(t) - \hat{x}_{j0}(t))\,(x(t) - \hat{x}_{j0}(t))^T\,|q(t+1) = j, Z^t\right] \\
&= \sum_{i=1}^{n_d} \left\{ P_i(t) + (\hat{x}_i(t) - \hat{x}_{j0}(t))\,(\hat{x}_i(t) - \hat{x}_{j0}(t))^T \right\} \gamma_{ji}(t)
\end{aligned} \tag{2.15}$$

where

$$\gamma_{ji}(t) := \Pr\left[q(t) = i | q(t+1) = j, Z^t\right] = \frac{1}{c_j} \lambda_{ij}(t)\,\alpha_i(t) \tag{2.16}$$

and c_j is a normalizing constant.

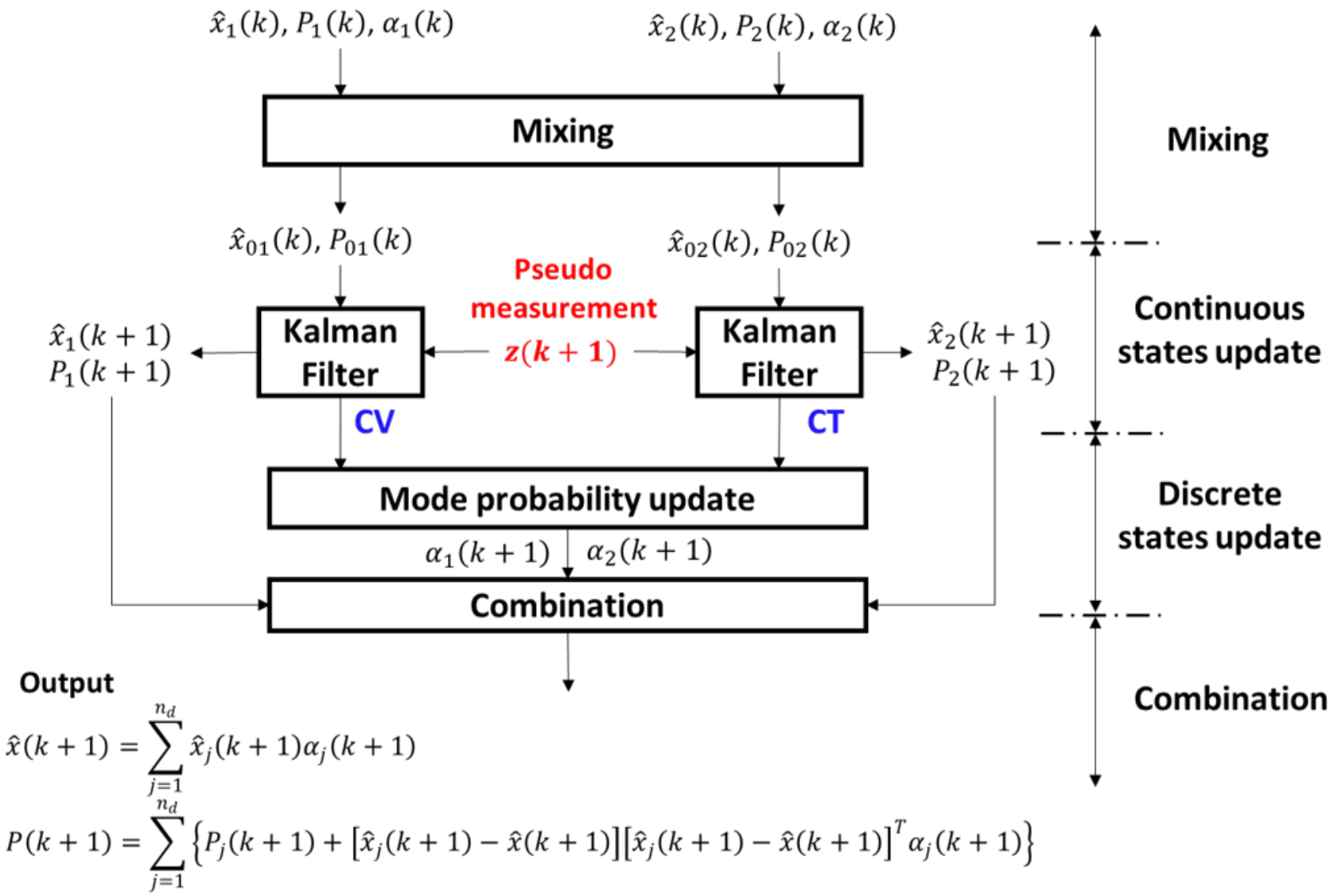


Figure 2.17. structure of hybrid estimation

Step 2: Continuous states update

For every $j \in Q$, Kalman filter j computes the innovation $\nu_j(t+1)$:

$$\nu_j\left(t+1\right) = z\left(t+1\right) - C_j A_j \hat{x}_{jo}(t) \tag{2.17}$$

and then updates the continuous state estimates $\hat{x}_j(t+1)$ and its covariance $P_j\left(t+1\right)$ as:

$$\hat{x}_j\left(t+1\right) = A_j \hat{x}_{j0}\left(t\right) + K_j\left(t+1\right)\nu_j(t+1) \tag{2.18}$$

$$P_j\left(t+1\right) = \left[I - K_j\left(t+1\right)C_j\right] P_j\left(t+1|t\right) \tag{2.19}$$

where

$$P_j\left(t+1|t\right) = A_j P_{j0}\left(t\right) A_j^T + Q_j \tag{2.20}$$

$$S_j\left(t+1\right) = C_j P_j\left(t+1|t\right) C_j^T + R_j \tag{2.21}$$

$$K_j\left(t+1\right) = P_j\left(t+1|t\right) C_j^T S_j^{-1}\left(t+1\right) \tag{2.22}$$

Step 3: Discrete states update

The (posterior) mode probabilities are updated by

$$\alpha_j\left(t+1\right) = \Pr\left[q\left(t+1\right) = j | Z^{t+1}\right] \tag{2.23}$$

$$= \frac{1}{\delta} p\left[z\left(t+1\right) \mid q\left(t+1\right) = j, Z^t\right] \Pr\left[q\left(t+1\right) = j | Z^t\right] \tag{2.24}$$

where the prior mode probability $\alpha_j\left(t+1|t\right)$ and the likelihood function for Kalman filter j are given as

$$\alpha_j\left(t+1|t\right) := \Pr\left[q\left(t+1\right) = j | Z^t\right] = \sum_{i=1}^{n_d} \Lambda_{ij}\left(t\right)\alpha_i\left(t\right) \tag{2.25}$$

$$\Lambda_j(t+1) := \mathcal{N}\left(r_i(t+1); 0, S_i(t+1)\right) \tag{2.26}$$

where $r_i(t+1)$ is the residual produced by Kalman filter i, $S_i(t+1)$ is the corresponding residual covariance, and $\mathcal{N}(a; b, c)$ is the probability at a of a normal distribution with mean b and covariance c.

The underlying idea of RM-IMM is that if the mode i is highly likely to be the correct one, then the residual mean $\bar{r}_i(k+1) := \mathbf{E}[r_i(k+1)]$ would be small. Therefore, by modifying the likelihood of the mode as

$$\Lambda_j(t+1) = \begin{cases} \frac{N_j(t)\Lambda_j(t)}{\sum_{i=1}^{N} N_i(t)\Lambda_i(t)} & \text{if} \quad \bar{r}_j(t) \neq 0 \\ \Lambda_j(t) & \text{otherwise} \end{cases} \tag{2.27}$$

where $N_i(t) = ||\bar{r}_i(t)||^{-1}$ if $||\bar{r}_i(t)|| \neq 0$; $N_i(t) = 1$, otherwise, the difference between the likelihoods of the (highly likely) correct mode and the other modes becomes large, thus giving more distinctive mode probabilities.

Step 4: Combination

The final output is calculated as follows through the combination process:

$$\hat{x}(t+1) = \sum_{j=1}^{n_d} \hat{x}_j(t+1)\,\alpha_j(t+1) \tag{2.28}$$

$$P(t+1) = \sum_{j=1}^{n_d} \left\{ P_j(t+1) + [\hat{x}_j(t+1) - \hat{x}(t+1)]\,[\hat{x}_j(t+1) - \hat{x}(t+1)]^T\,\alpha_j(t+1) \right\} \tag{2.29}$$

2.4 Hybrid data-driven and physics-based trajectory prediction

In this section, we present how to combine the data-driven method and the physics-based method presented in the previous sections. Note that the data-driven prediction takes input and generates input in a deterministic manner, i.e., as a point estimation, μ_{t+1}. The Physics-based prediction method, however, takes as input the measurements in a stochastic form, that is, mean and covariance, or a probability density function (pdf), under the Gaussian

assumption. In the proposed framework, the two methods are integrated as shown in Figure 2.18:

- Physics-based to data-driven: from the output of the physics-based method (given as a pdf), a set of points are sampled, which is fed into the data-driven method as input.
- Data-driven to physics-based: with the output of the data-driven method obtained by a Monte-Carlo method [33],[34] (given as a set of points), we perform a fitting to a Gaussian pdf, which is fed into the physics-based method as input.

The Monte-Carlo method used in the connection from data-driven to physics-based is performed as follows:

- We sample from the output of the physics-based method (the pdf) to feed the samples to the data-driven method.
- For each sample, Monte-Carlo Dropout randomly selects connections in LSTM to be dropped out (or removed) and then the corresponding sampled output is obtained.

Figure 2.19 shows how the integrated framework works for trajectory prediction. Suppose that we have incoming track points of an aircraft, $z_0, \ldots, z_{T'}$, at the current time-step T'. We denote the pseudo-measurement as $\widetilde{z}$ (from the data-driven model) and the prediction as $\hat{z}$ (from the physics-based method). The data-driven prediction model and the physics-based prediction method are denoted as f_d and f_p, respectively. At T', the predicted track points at the future time-steps $T' + k$ for $k = 0, \ldots, N-1$ are computed as follows:

- For $k = 0$
  - The data-driven prediction model generates a pseudo-measurement as

$$\left(\widetilde{z}_{T'+k+1}, \widetilde{R}_{T'+k+1}\right) = f_d\left(\{z_t\}_{t=0}^{T'}\right) \tag{2.30}$$

    where $\{z_t\}_{t=0}^{T'}$ is the sequence of the incoming track points.

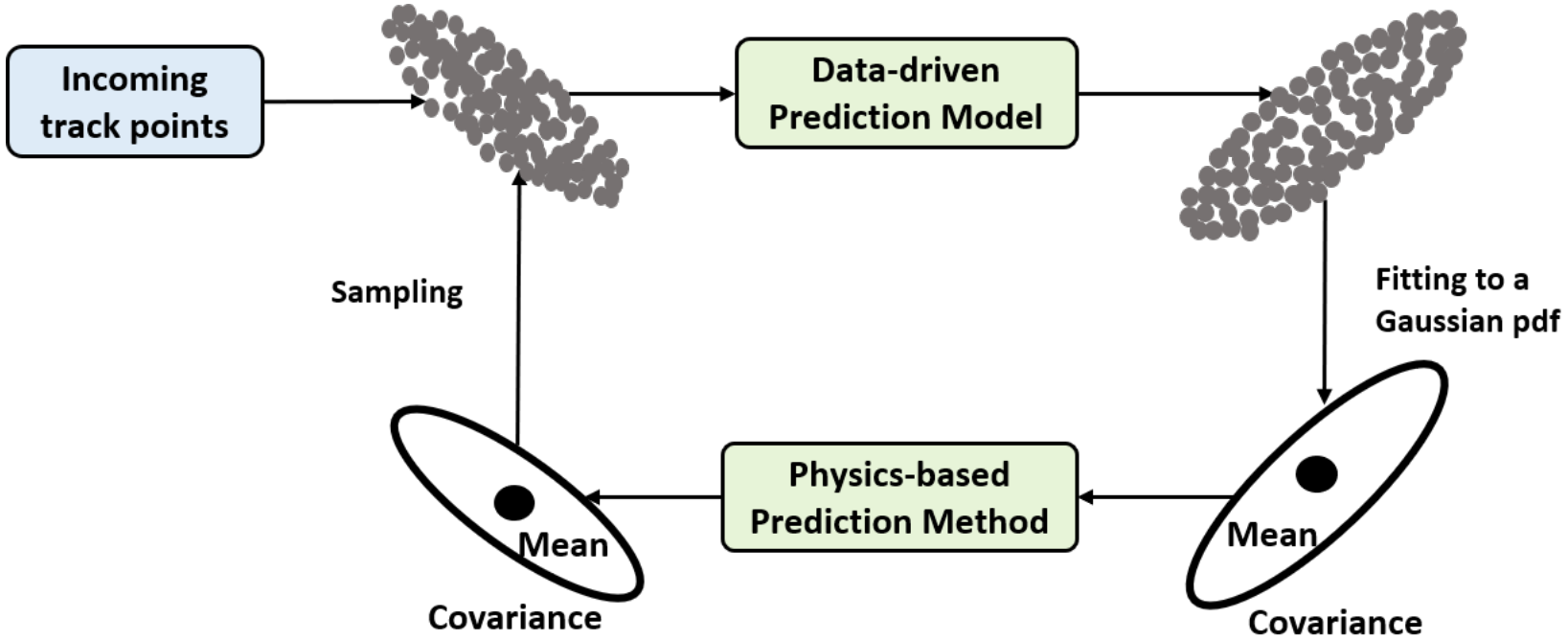


Figure 2.18. Integration of Two Trajectory Prediction Methods

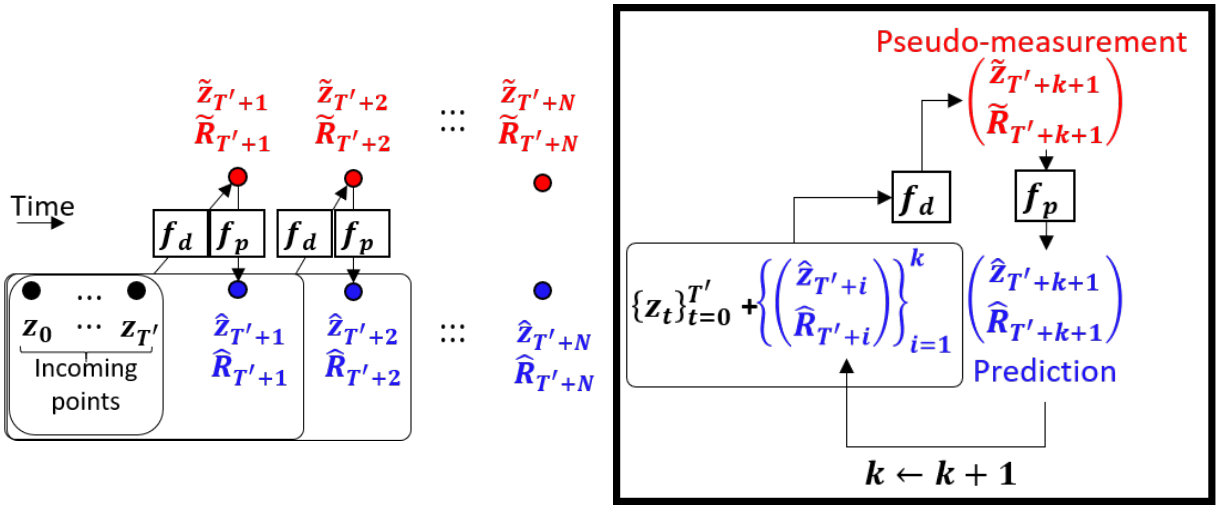


Figure 2.19. Framework of integration

– The physics-based prediction method then computes the prediction by taking the pseudo-measurement

$$\left(\hat{z}_{T'+k+1}, \hat{R}_{T'+k+1}\right) = f_p\left(\tilde{z}_{T'+k+1}, \tilde{R}_{T'+k+1}\right) \tag{2.31}$$

- For $k = 1, \ldots, N-1$, the input to the data-driven prediction model becomes the mixture of the deterministic track points up to T' and the stochastic (predicted) track points from $T'+1$ to $T'+k$,

$$\left(\tilde{z}_{T'+k+1}, \tilde{R}_{T'+k+1}\right) = f_d\left(\{z_t\}_{t=0}^{T'},\ \left\{\left(\hat{z}_{T'+i}, \hat{R}_{T'+i}\right)\right\}_{i=1}^{k}\right) \tag{2.32}$$

The physics-based method is the same as Eq. (2.31).

3. CASE STUDY

In this chapter, the proposed hybrid data-driven and physics-based trajectory prediction is demonstrated with two case studies: short-term trajectory prediction and Estimated Time of Arrival (ETA) prediction.

- Short-term trajectory prediction is to predict a future trajectory over short time duration, ranging from 1 to 10 minutes. Short-term trajectory prediction is important for air traffic operation in the sense that it can improve the situational awareness of ATCs.
- ETA prediction is to calculate the expected arrival time of a trajectory from a starting point (e.g., an entry fix) to a destination (e.g., a runway threshold). ETA prediction can have a benefit to air traffic management, especially in terminal airspace, such as spacing and scheduling.

In what follows, each case study is presented and illustrated with two baseline methods: a physics-based method alone and a data-driven method alone.

3.1 Short-term trajectory prediction

3.1.1 Illustrative examples

For illustration, the proposed framework is applied to two representative trajectories (an arrival trajectory at GMP and a departure trajectory from ICN), as shown in Figures 3.1 and 3.2. The prediction is performed for the prediction horizon of 20 time-steps (a time interval is 5 seconds). Since prediction errors tend to be large around a turning point, we mainly focus on the part of the trajectories where the aircraft take turn. The black line represents the true trajectories, the blue line represents the predicted results of the proposed method, the red line represents the predicted results of the data-driven method alone, and the green line represents the predicted results of the physics-based method alone. The performance of each method is measured by the prediction error, i.e., the difference between the predicted trajectory and the true trajectory.

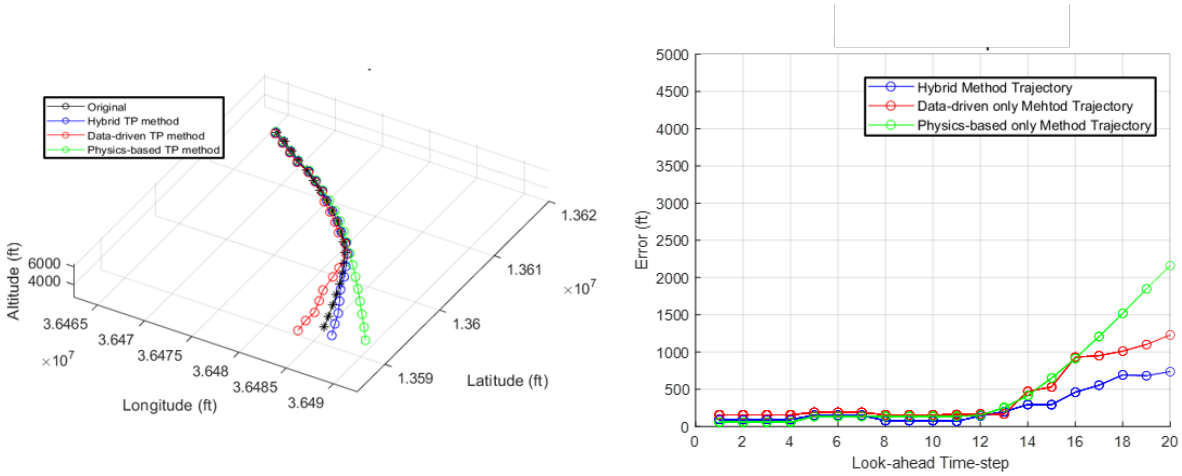


Figure 3.1. Prediction of trajectory for arrival at GMP (left) and prediction error (right)

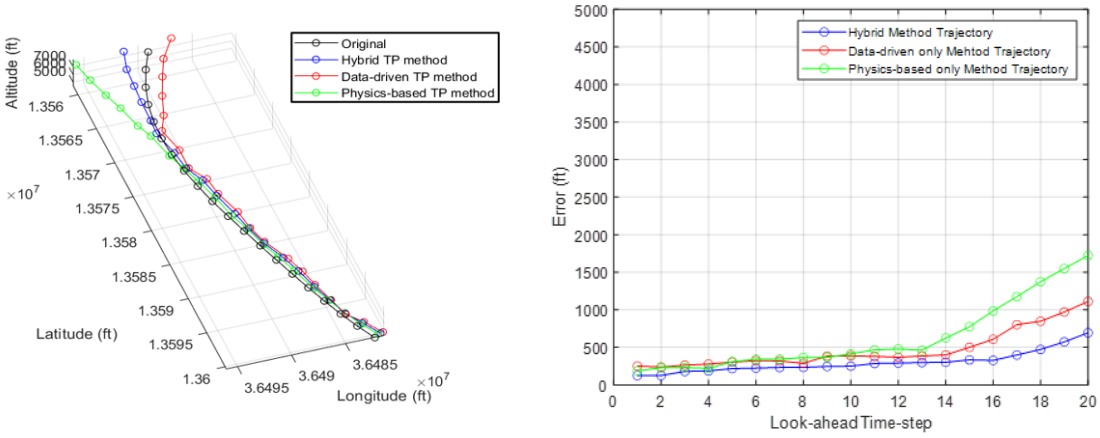


Figure 3.2. Prediction of trajectory for departure from ICN (left) and prediction error (right)

Tables 3.1 and 3.2 compare the performance measures of the proposed method and the two baseline methods for the arrival trajectory at GMP and the departure trajectory from ICN, respectively, where the mean, minimum, and maximum of the error are computed over the prediction horizon. The results show that the proposed method outperforms the baseline methods. For the arrival trajectory at GMP, the mean error of the proposed method is reduced by $41\% = (813.84 - 483.29)/813.84$ compared to the data-driven method and by $58\% = (1148.92 - 483.29)/1148.92$ compared to the physics-based method. For the departure trajectory from ICN, the mean error of the proposed method is reduced by $26\% = (634.21 -$

471.75)/634.21 compared to the data-driven method and by $52\% = (973.02 - 471.75)/973.02$ compared to the physics-based method.

Table 3.1. Performance comparison with baseline methods for trajectory for arrival at GMP

Method	Minimum error	Mean error	Maximum error
Hybrid method	17.09 ft	483.29 ft	800.42 ft
Data-driven method	31.27 ft	813.84 ft	1364.17 ft
Physics-based method	64.64 ft	1148.92 ft	2,235.19 ft

Table 3.2. Performance comparison with baseline methods for trajectory for departure from ICN

Method	Minimum error	Mean error	Maximum error
Hybrid method	29.16 ft	471.75 ft	798.35 ft
Data-driven method	49.33 ft	634.21 ft	1174.32 ft
Physics-based method	83.11 ft	973.02 ft	1,821.42 ft

The followings are observed from Figures 3.1 and 3.2 and Tables 3.1 and 3.2:

- The physics-based method is not able to accurately predict the track points around a turning point since no measurements is available for the future time-steps and thus the correction by measurement is not possible.
- In general, the prediction error tends to increase along the prediction horizon. As shown in Tables 3.1 and 3.2, the maximum error is relatively larger than the minimum error in both cases.

3.1.2 Performance metrics

To measure the performance of the short-term trajectory prediction in detail, we introduce four metrics that are widely used in trajectory prediction [9], [35], [36], as follows:

along-track error, cross-track error, horizontal error, and vertical error, as shown in Figure 3.3. The definitions of the performance metrics are given below:

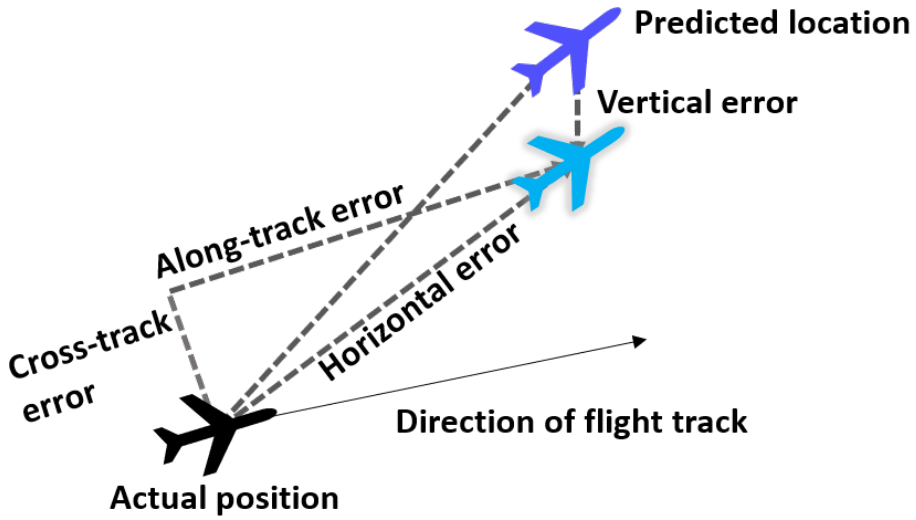


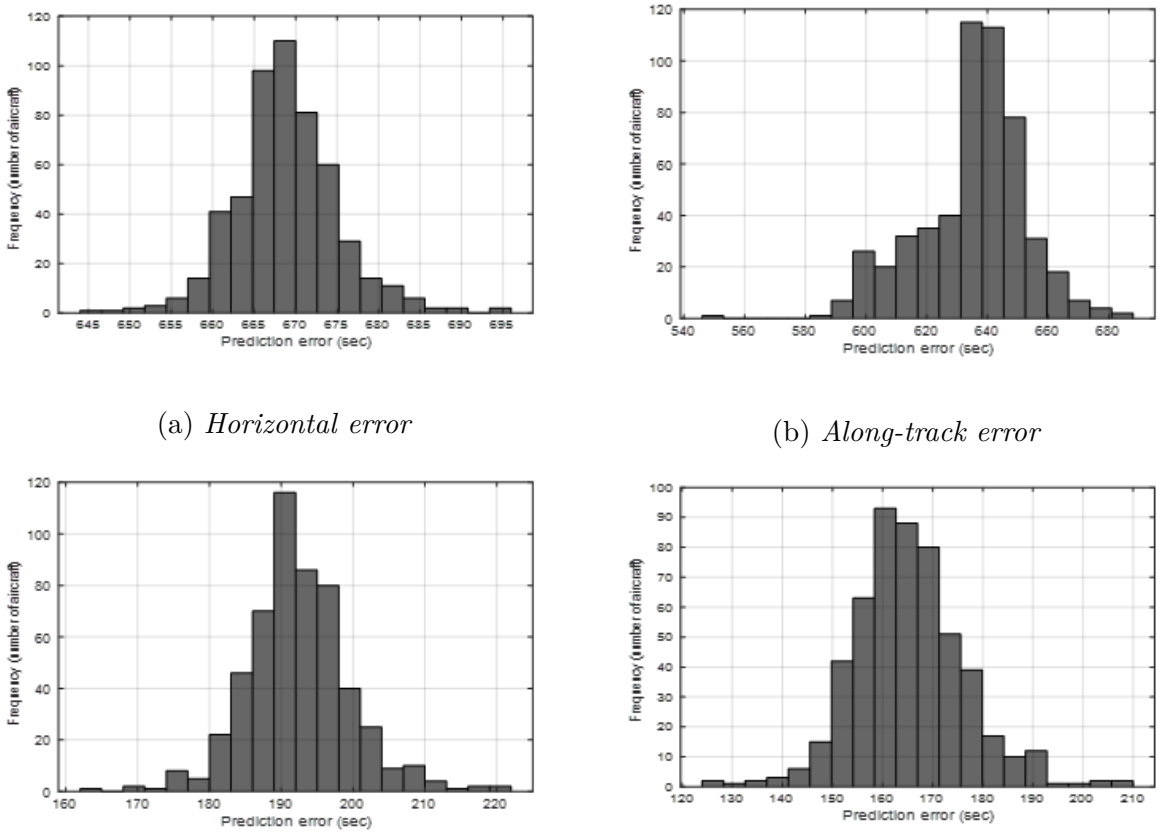
Figure 3.3. Illustration of along-track, cross-track, horizontal and vertical error

- Along-track error: difference between the actual location and expected location projected onto the actual course.
- Cross-track error: difference between the actual location expected location projected onto a vector perpendicular to the actual course.
- Horizontal error: difference between the actual location and expected location in the horizontal dimension.
- Vertical error: difference between the actual location and expected location in the vertical dimension.

For illustration, we present the above metrics for an arrival cluster at GMP in Figure 3.4 and Table 3.3, and for a departure cluster at ICN in Figure 3.5 and Table 3.4, from which the followings are observed:

- The along-track error is larger than the cross-track error and thus the horizontal error is mainly due to the along-track error. This is possibly due to the fact that in the data preparation process, the trajectories are grouped together as a cluster if their spatial patterns (in the cross-track direction) are similar, and therefore the similarity in the temporal patterns (in the along-track direction) is not explicitly considered.

- The horizontal error is larger than vertical error. This is possibly because there are many movements of the aircraft in the horizontal dimension than in the vertical dimension.



(a) *Horizontal error*

(b) *Along-track error*

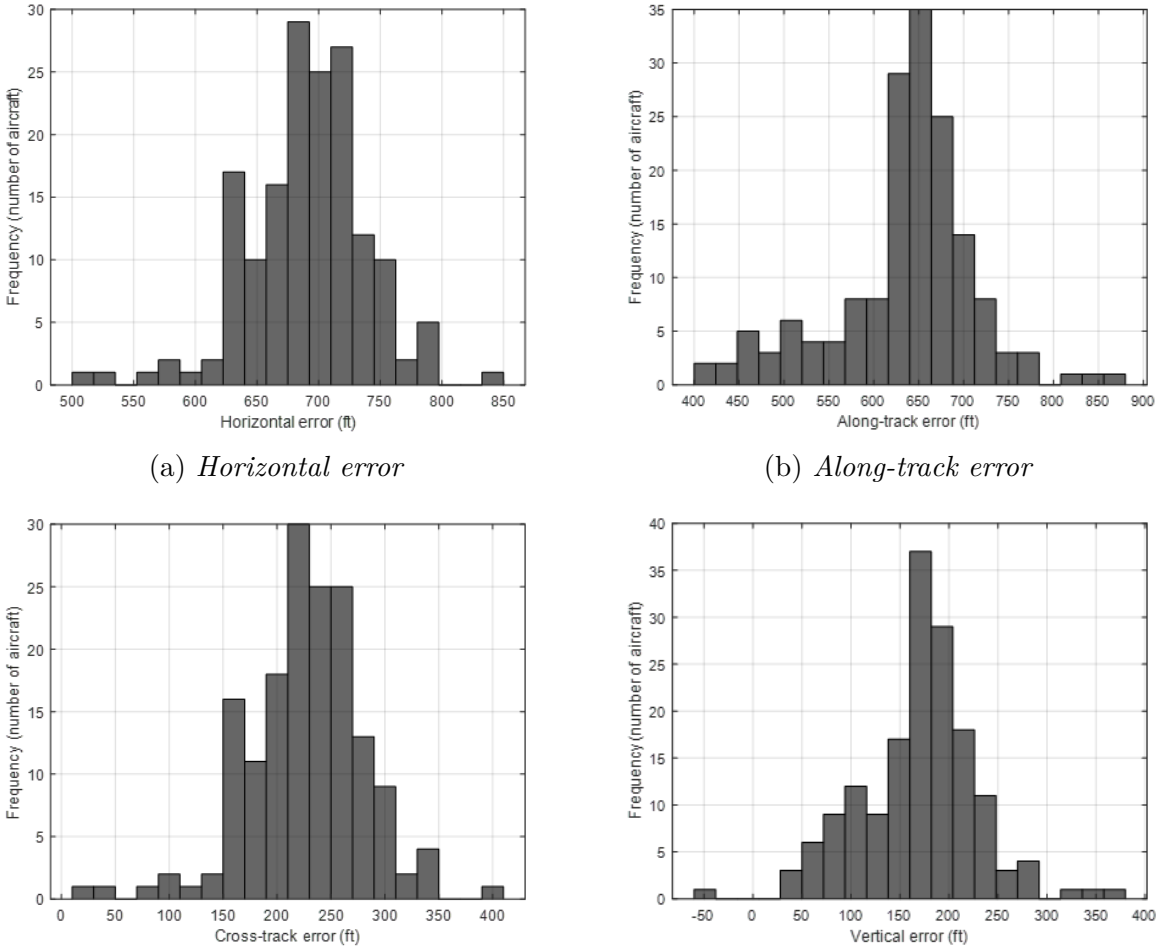
(c) *Cross-track error*

(d) *Vertical error*

Figure 3.4. Histograms of performance metrics for an arrival cluster at GMP

Table 3.3. Performance comparison of performance metrics for an arrival cluster at GMP

Total error	Horizontal error	Along-track error	Cross-track error	Vertical error
688.5 ft	667.5 ft	639.1 ft	192.4 ft	164.1 ft



(a) *Horizontal error* (b) *Along-track error*

(c) *Cross-track error* (d) *Vertical error*

Figure 3.5. Histograms of performance metrics for a departure cluster at ICN

Table 3.4. Performance comparison of performance metrics for a departure cluster at ICN

Total error	Horizontal error	Along-track error	Cross-track error	Vertical error
690.1 ft	667.4 ft	639.0 ft	192.3 ft	167.4 ft

Tables 3.5 and 3.6 show the results for 15 clusters obtained from ICN and 8 clusters from GMP, respectively. It is noted that a typical separation standard in the horizontal in terminal airspace is 3 nm [37],[38]. Since our results show the average prediction error in the horizontal of 694.6 ft ($\approx$ 0.114 nm) and 689.7 ft ($\approx$ 0.113 nm) for ICN and GMP, respectively, which thus could help improve the performance of conflict detection in terminal airspace, thus enhanching the safety.

3.2 Estimated Time of Arrival (ETA) prediction

The performance of ETA prediction is measured by the ETA prediction error, which is defined as, given a starting point and a destination point, the difference between the predicted and true times of arrival at the destination. For the air traffic surveillance data used in this work, the total travel time ranges from 7.5 minutes to about 12 minutes for different clusters. For a given starting point of an aircraft, its trajectory is predicted up to its destination point to compute the predicted ETA, as shown in Figure 3.6.

Figures 3.7 and 3.8 show the histograms of the ETA prediction errors for an arrival cluster of ICN and an arrival cluster at GMP, respectively. The proposed method is compared with a data-driven only method as a baseline method. A physics-based method is not used For ETA prediction because it does not have a correction by measurement, and thus the prediction diverges. For the ICN cluster, as shown in Figure 3.7, the RMS error of the proposed method is reduced by 19.5% (from 37.0 seconds to 29.8 seconds) compared to the baseline method. The maximum error of the proposed method is also significantly reduced: it is 70.1 seconds, which is a 33.7% improvement over the 105.8 seconds from the baseline method. For the GMP cluster, as shown in Figure 3.8, the RMS error of the proposed method is reduced by 19.6% (from 37.8 seconds to 30.4 seconds) compared to the baseline method. The maximum error of the proposed method is also reduced: it is 52.9 seconds, which is a 32.1% improvement over the 77.9 seconds from the baseline method.

For the 15 clusters of ICN and the 8 clusters of GMP presented in Section 3.1.2, Tables 3.7 and 3.8 show the RMS errors of ETA prediction, respectively. In each row, the cluster numbering is presented with its total travel time in the first column; the prediction errors of

Table 3.5. Performance metrics comparison with baseline methods for ICN clusters (all numbers are in ft)

Cluster	Total error	Horizontal error	Along-track error	Cross-track error	Vertical error
1	688.5	667.5	639.1	192.4	164.1
2	713.4	693.4	650.3	238.1	168.2
3	723.5	700.5	669.4	209.4	176.9
4	698.8	677.2	638.3	226.7	170.4
5	750.2	731.2	697.0	223.3	171.8
6	731.6	715.4	687.4	198.1	152.1
7	672.5	656.7	620.6	217.2	146.8
8	789.6	771.8	748.1	186.9	167.8
9	742.6	723.1	681.8	242.7	166.8
10	731.1	714.0	678.3	223.8	156.7
11	708.5	689.3	647.9	232.0	162.9
12	709.9	691.4	653.8	225.9	158.7
13	673.8	652.7	617.0	210.7	166.8
14	694.2	672.4	644.7	191.9	173.3
15	683.3	662.2	627.4	212.4	168.0
Average	714.4	694.6	660.7	214.1	164.7

Table 3.6. Performance metrics comparison with baseline methods for GMP clusters (all numbers are in ft)

Cluster	Total error	Horizontal error	Along-track error	Cross-track error	Vertical error
1	690.1	667.4	639.0	192.3	167.4
2	701.0	681.2	664.7	217.7	176.5
3	704.3	694.7	673.2	218.6	168.3
4	684.2	660.9	641.2	205.8	158.3
5	719.3	702.7	683.2	195.7	165.5
6	693.4	673.2	652.1	204.9	162.5
7	681.7	667.4	646.5	205.7	158.9
8	720.1	703.3	683.1	184.9	179.3
Average	705.8	689.7	671.3	208.6	167.3

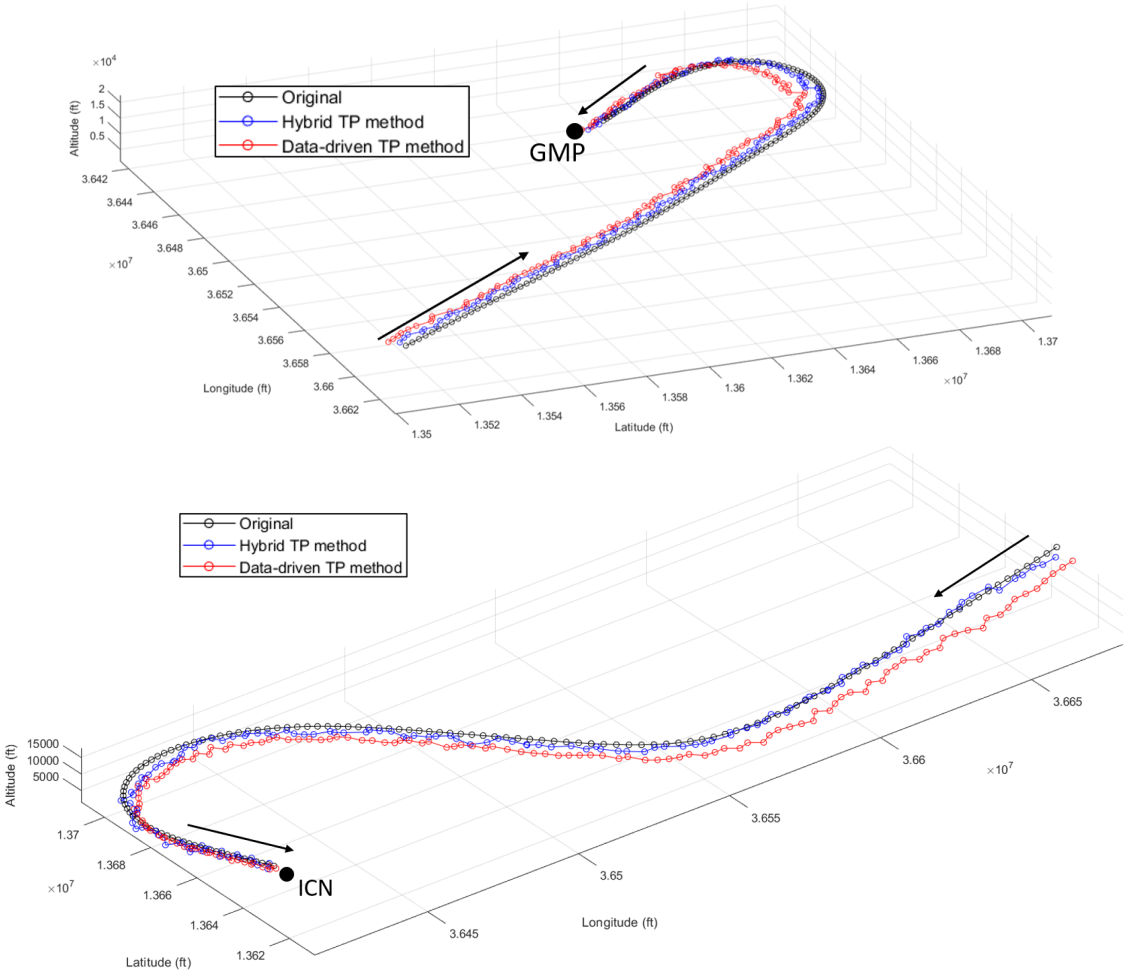
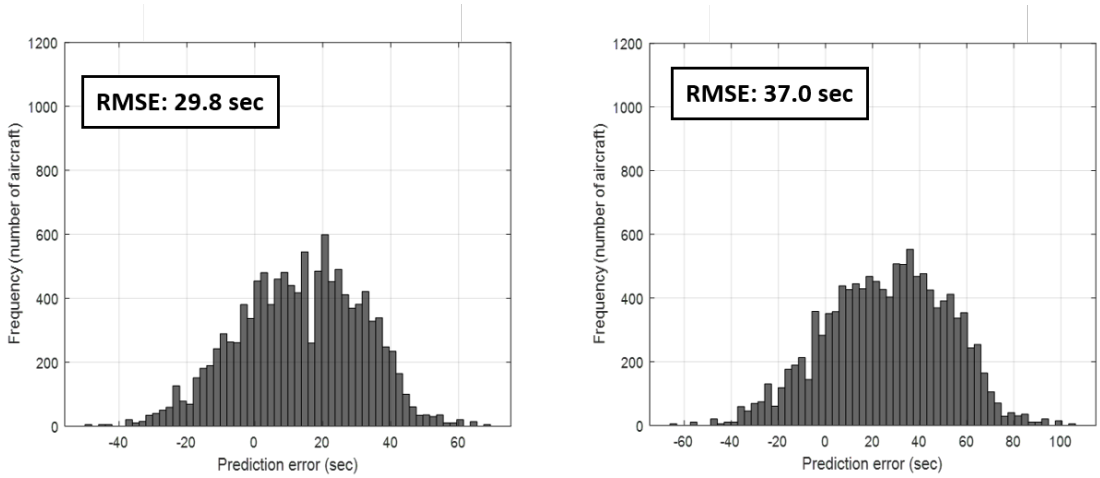


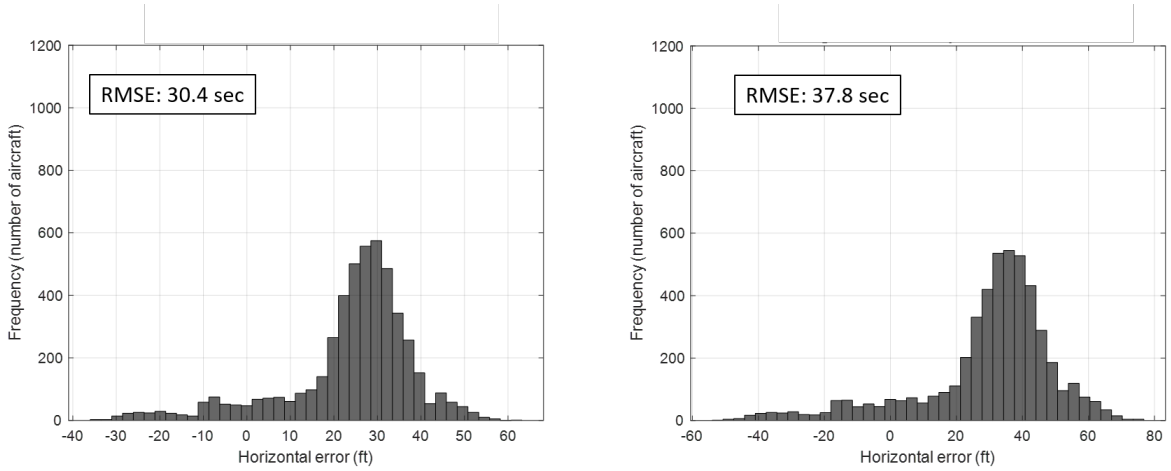
Figure 3.6. Prediction of trajectory for arrival at GMP (left) and arrival at ICN (right)

the proposed method and the baseline method are shown in the second and third columns, respectively, where the percentage in the parenthesis is the ratio of the prediction error with respect to the total travel time; and the reduction in the prediction error from the baseline method to the proposed method is given in the fourth column as a percentage. As the total travel time increase, the prediction error becomes larger. For example, cluster 7 at ICN has the total travel time of 450 seconds with the prediction error of 23.5 seconds by the proposed method; on the other hand, cluster 2 at ICN shows the increased prediction error of 36.1 seconds (53.6% increased) by the proposed method where its travel time is 700 seconds (55.5% increased). In average, the proposed method shows the reduced prediction error compared to the baseline method, by 24.3% for ICN cluster and 20.7% for GMP cluster.



(a) *Proposed method* (b) *Baseline method*

Figure 3.7. Histograms of the ETA prediction errors for an arrival cluster at ICN (RMSE denotes rms error)



(a) *Proposed method* (b) *Baseline method*

Figure 3.8. Histograms of the ETA prediction errors for an arrival cluster at GMP (RMSE denotes rms error)

Note that a typical separation standard between consecutive arrivals at the runway is about 2 minutes [39], the reduced error in the predicted ETA by the proposed method could imply practical benefits in airport operations.

Table 3.7. RMS error in ETAs of proposed method and baseline method for ICN clusters

Cluster number	Proposed method	Baseline method	Reduction
cluster 1 (510 sec)	25.6 sec (5.0 %)	33.8 sec (6.6 %)	28.1
cluster 2 (700 sec)	36.1 sec (5.1 %)	46.2 sec (6.7 %)	27.9
cluster 3 (625 sec)	32.1 sec (5.1 %)	36.6 sec (5.8 %)	14.2
cluster 4 (630 sec)	32.7 sec (5.2 %)	38.1 sec (6.0 %)	16.6
cluster 5 (610 sec)	31.9 sec (5.3 %)	37.2 sec (6.1 %)	16.1
cluster 6 (605 sec)	31.8 sec (5.3 %)	37.0 sec (6.1 %)	20.0
cluster 7 (450 sec)	23.5 sec (5.2 %)	29.8 sec (6.6 %)	26.8
cluster 8 (695 sec)	35.2 sec (5.0 %)	42.1 sec (6.0 %)	19.6
cluster 9 (510 sec)	26.4 sec (5.2 %)	34.0 sec (6.6 %)	25.0
cluster 10 (520 sec)	25.9 sec (4.9 %)	33.5 sec (6.4 %)	25.4
cluster 11 (500 sec)	26.3 sec (5.3 %)	34.7 sec (6.9 %)	35.4
cluster 12 (530 sec)	26.9 sec (5.1 %)	35.7 sec (6.7 %)	32.7
cluster 13 (520 sec)	26.7 sec (5.2 %)	33.3 sec (6.4 %)	20.9
cluster 14 (650 sec)	33.9 sec (5.2 %)	38.5 sec (5.9 %)	13.5
cluster 15 (660 sec)	33.5 sec (5.1 %)	39.1 sec (5.9 %)	16.7
Average	29.8 sec	37.0 sec	24.3 %

Table 3.8. RMS error in ETAs of proposed method and baseline method for GMP clusters

Cluster	Proposed method	Baseline method	Reduction
cluster1 (675 sec)	35.1 sec (5.2 %)	45.9 sec (6.6 %)	23.6 %
cluster2 (685 sec)	34.9 sec (5.1 %)	48.3 sec (7.0 %)	27.8 %
cluster3 (640 sec)	33.8 sec (5.3 %)	36.6 sec (5.8 %)	7.6 %
cluster4 (625 sec)	31.9 sec (5.2 %)	39.4 sec (6.3 %)	19.1 %
cluster5 (570 sec)	29.0 sec (5.0 %)	36.8 sec (6.4 %)	21.2 %
cluster6 (595 sec)	30.4 sec (5.1 %)	37.1 sec (6.2 %)	18.1 %
cluster7 (680 sec)	35.3 sec (5.2 %)	46.0 sec (6.7 %)	23.3 %
cluster8 (635 sec)	31.1 sec (5.0 %)	41.5 sec (6.5 %)	25.1 %
Average	32.6 sec	41.45 sec	20.7%

4. CONCLUSION

In this chapter, summary and potential extensions of the research presented in this thesis are discussed.

4.1 Summary

In this thesis, a trajectory prediction algorithm in air traffic management (ATM), aimed at improving the efficiency and safety of operations in the terminal airspace has been proposed. The proposed trajectory prediction algorithm has been shown to provide air traffic controllers (ATCs), pilots, airlines, and policymakers insights into making air traffic operations safer and more efficient. This was achieved through integrating two trajectory prediction methods: the data-driven method and the physics-based method. The proposed method can make more accurate trajectory predictions by taking advantage of the two methods incorporating the expected future behaviors of the aircraft (from data) with its current status or dynamics (from physics). With Long Short-Term Memory (LSTM) as a data-driven method and Residual-Mean Interacting Multiple Models (RM-IMM) as a physics-based method, the proposed method integrates different types of input and output of each method. For the demonstration of the proposed algorithm, extensive tests were performed using air traffic surveillance data from the repository of real historical flight datasets, comprising of arrival and departure operations at Incheon International Airport (ICN) and Gimpo International Airport (GMP) in South Korea, applied to two cases: estimated time of arrival (ETA) prediction and short-term trajectory prediction. The accurate ETA prediction is important since it can improve the airspace operations, especially in the terminal airspace, such as scheduling and spacing; and the accurate short-term trajectory prediction can improve situational awareness of ATCs by analyzing the current state of the aircraft and the future state of the aircraft under the control of ATCs. The test results have shown that the proposed hybrid trajectory prediction algorithm yields better accuracy than widely used baseline methods, the data-driven method and the physics-based method.

4.2 Future Work

Among potential follow-up research efforts to the work presented in this thesis, the following can be highlighted:

Domain extension: En-route airspace

This thesis has focused on trajectory prediction in terminal airspace. It is expected that en-route airspace presents a completely new environment for the algorithm to learn. The effectiveness of the proposed algorithm applied to the terminal airspace can lead to a possible extension of applying the algorithm to predict flight trajectories in the en-route airspace.

Prior work + trajectory prediction + posterior work

From the literature review, we found that there are three types of works related to trajectory prediction:

- Prior work + trajectory prediction: e.g., trajectory clustering or trajectory modeling
- Trajectory prediction + posterior work: e.g., conflict detection or conformance monitoring
- Trajectory prediction only

As an example for the combination with posterior work, we can consider conformance monitoring [40][41]: the predicted aircraft states obtained from the proposed hybrid trajectory prediction can be used to compute the conformity/anomaly score of an aircraft under monitoring, which measures how confomring/non-conforming the predicted trajectory of the aircraft is with respect to the expected behavior observed from the historical data.

REFERENCES

[1] "Vision 2050," International Air Transport Association (IATA), Tech. Rep., 2011.

[2] B. C. Airplanes, "Statistical summary of commercial jet airplane accidents," *Worldwide Operations*, 2009.

[3] W. Liu, C. E. Seah, and I. Hwang, "Estimation algorithm for stochastic linear hybrid systems with quadratic guard conditions," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, IEEE, 2009, pp. 3946–3951.

[4] J. Hu and M. Prandini, "Aircraft conflict detection: A method for computing the probability of conflict based on markov chain approximation," in *2003 European Control Conference (ECC)*, IEEE, 2003, pp. 2225–2230.

[5] 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.

[6] J. Zhang, J. Liu, R. Hu, and H. Zhu, "Online four dimensional trajectory prediction method based on aircraft intent updating," *Aerospace Science and Technology*, vol. 77, pp. 774–787, 2018.

[7] T. Baklacioglu and M. Cavcar, "Aero-propulsive modelling for climb and descent trajectory prediction of transport aircraft using genetic algorithms," *The Aeronautical Journal*, vol. 118, no. 1199, pp. 65–79, 2014.

[8] J. Wiest, M. Höffken, U. Kreßel, and K. Dietmayer, "Probabilistic trajectory prediction with gaussian mixture models," in *2012 IEEE Intelligent Vehicles Symposium*, IEEE, 2012, pp. 141–146.

[9] S. Ayhan and H. Samet, "Aircraft trajectory prediction made easy with predictive analytics," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 21–30.

[10] A. L. Duca, C. Bacciu, and A. Marchetti, "A k-nearest neighbor classifier for ship route prediction," in *OCEANS 2017-Aberdeen*, IEEE, 2017, pp. 1–6.

[11] Z. Xiao, P. Li, V. Havyarimana, G. M. Hassana, D. Wang, and K. Li, "Goi: A novel design for vehicle positioning and trajectory prediction under urban environments," *IEEE Sensors Journal*, vol. 18, no. 13, pp. 5586–5594, 2018.

[12] Y. Liu and M. Hansen, "Predicting aircraft trajectories: A deep generative convolutional recurrent neural networks approach," *arXiv preprint arXiv:1812.11670*, 2018.

[13] D. Birant and A. Kut, "St-dbscan: An algorithm for clustering spatial–temporal data," *Data & knowledge engineering*, vol. 60, no. 1, pp. 208–221, 2007.

[14] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," 1999.

[15] R. Xiaohui and G. Xianpeng, "Study on a large-number of heterogeneous aeronautical information publication collection mechanism," in *2018 IEEE International Conference of Safety Produce Informatization (IICSPI)*, IEEE, 2018, pp. 454–457.

[16] J. Zhang, L. Wei, and Z. Yanbo, "Study of ads-b data evaluation," *Chinese Journal of Aeronautics*, vol. 24, no. 4, pp. 461–466, 2011.

[17] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise.," in *Kdd*, vol. 96, 1996, pp. 226–231.

[18] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Eleventh annual conference of the international speech communication association*, 2010.

[19] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *arXiv preprint arXiv:1409.2329*, 2014.

[20] A. H. Ribeiro, K. Tiels, L. A. Aguirre, and T. Schön, "Beyond exploding and vanishing gradients: Analysing rnn training using attractors and smoothness," in *International Conference on Artificial Intelligence and Statistics*, PMLR, 2020, pp. 2370–2380.

[21] S. Al-Abri, T. X. Lin, M. Tao, and F. Zhang, "A derivative-free optimization method with application to functions with exploding and vanishing gradients," *IEEE Control Systems Letters*, vol. 5, no. 2, pp. 587–592, 2020.

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[23] Y. Hirose, K. Yamashita, and S. Hijiya, "Back-propagation algorithm which varies the number of hidden units," *Neural networks*, vol. 4, no. 1, pp. 61–66, 1991.

[24] C. E. Seah and I. Hwang, "Stochastic linear hybrid systems: Modeling, estimation, and application in air traffic control," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 3, pp. 563–575, 2009.

[25] 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.

[26] I. Hwang, C. E. Seah, and S. Lee, "A study on stability of the interacting multiple model algorithm," *IEEE Transactions on Automatic Control*, vol. 62, no. 2, pp. 901–906, 2016.

[27] C. E. Seah and I. Hwang, "State estimation for stochastic linear hybrid systems with continuous-state-dependent transitions: An imm approach," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 45, no. 1, pp. 376–392, 2009.

[28] I. Hwang, H. Balakrishnan, and C. Tomlin, "State estimation for hybrid systems: Applications to aircraft tracking," *IEE Proceedings-Control Theory and Applications*, vol. 153, no. 5, pp. 556–566, 2006.

[29] I. Hwang, J. Hwang, and C. Tomlin, "Flight-mode-based aircraft conflict detection using a residual-mean interacting multiple model algorithm," in *AIAA guidance, navigation, and control conference and exhibit*, 2003, p. 5340.

[30] I. Hwang, H. Balakrishnan, K. Roy, J. Shin, L. Guibas, and C. Tomlin, "Multiple-target tracking and identity management," in *SENSORS, 2003 IEEE*, IEEE, vol. 1, 2003, pp. 36–41.

[31] I. Hwang and C. E. Seah, "An estimation algorithm for stochastic linear hybrid systems with continuous-state-dependent mode transitions," in *Proceedings of the 45th IEEE Conference on Decision and Control*, IEEE, 2006, pp. 131–136.

[32] H. A. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with markovian switching coefficients," *IEEE transactions on Automatic Control*, vol. 33, no. 8, pp. 780–783, 1988.

[33] J. Hammersley, *Monte carlo methods*. Springer Science & Business Media, 2013.

[34] N. Metropolis and S. Ulam, "The monte carlo method," *Journal of the American statistical association*, vol. 44, no. 247, pp. 335–341, 1949.

[35] S. Mondoloni, S. Swierstra, and M. Paglione, "Assessing trajectory prediction performance-metrics definition," in *24th Digital Avionics Systems Conference*, IEEE, vol. 1, 2005, pp. 3.C.1-1 –3.C.1-13.

[36] T. Lehouillier, M. I. Nasri, F. Soumis, G. Desaulniers, and J. Omer, "Solving the air conflict resolution problem under uncertainty using an iterative biobjective mixed integer programming approach," *Transportation science*, vol. 51, no. 4, pp. 1242–1258, 2017.

[37] H. Tang, "A tactical separation assurance system for terminal airspace," in *14th AIAA Aviation Technology, Integration, and Operations Conference*, 2014, p. 2021.

[38] E. B. Walsh and R. H. Mayer, "Evaluation of aircraft separations observed in radar data of terminal operations," in *2008 Integrated Communications, Navigation and Surveillance Conference*, IEEE, 2008, pp. 1–11.

[39] J. Tittsworth, S. Lang, E. J. Johnson, S. Barnes, *et al.*, "Federal aviation administration wake turbulence program-recent highlights," 2012.

[40] G. Shafer and V. Vovk, "A tutorial on conformal prediction.," *Journal of Machine Learning Research*, vol. 9, no. 3, 2008.

[41] V. Balasubramanian, S.-S. Ho, and V. Vovk, *Conformal prediction for reliable machine learning: theory, adaptations and applications*. Newnes, 2014.

VITA

Hansoo Kim received the Bachelor of Science in mechanical engineering from Korea Military Academy (KMA), Seoul, Korea, in 2016. He then started to pursue a master's degree in Aeronautics and Astronautics at Purdue University in 2019. His research focuses on application of trajectory prediction in terms of safety and efficiency in air traffic management.