博士論文

Stochastic Optimal Control for Aircraft Conflict Resolution in the Presence of Uncertainty

(不確実環境下における航空機のコンフリクト回避のための確率的最適制御)

by

Yoshinori Matsuno
松野 賀宣

A dissertation submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in
Engineering
at
The University of Tokyo

December 2014

Dissertation Defense Committee:

Associate Professor Takeshi Tsuchiya (土屋 武司), Chair
Department of Aeronautics and Astronautics
The University of Tokyo

Professor Shinji Suzuki (鈴木 真二)
Department of Aeronautics and Astronautics
The University of Tokyo

Professor Koichi Hori (堀 浩一)
Department of Aeronautics and Astronautics
The University of Tokyo

Associate Professor Takehisa Yairi (矢入 健久)
Department of Advanced Interdisciplinary Studies
The University of Tokyo

Professor Seiya Ueno (上野 誠也)
Faculty of Environment and Information Sciences
Yokohama National University

Abstract

In this dissertation, probabilistic aircraft conflict detection and resolution algorithms in the presence of uncertainties are proposed. In order to accommodate the increasing air traffic and alleviate the workload of air traffic controllers, the proposed conflict detection and resolution algorithms can provide the automated advisories and conflict resolution trajectories for the air traffic controllers and pilots and have a potential to ultimately replace the tasks of the air traffic controllers. First, a spatially correlated wind model is used to describe the wind uncertainty, which is the primary uncertainty compared to other possible uncertainties such as navigation errors and pilots' intents. On the basis of stochastic aircraft dynamics containing the uncertainty, a probabilistic conflict detection algorithm using the polynomial chaos expansion method is proposed. The polynomial chaos expansion algorithm can quantify uncertainties in complex nonlinear dynamical systems with high computational efficiency. In addition, a numerical algorithm that incorporates the polynomial chaos expansion algorithm into the pseudospectral method is proposed to solve the conflict resolution problem as the stochastic optimal control problem. The stochastic optimal control method is combined with the proposed conflict detection algorithm to solve the conflict resolution problem under the wind uncertainty. Moreover, a stochastic near-optimal control method is proposed to generate conflict resolution trajectories and maneuvers in real time without actually solving the computationally expensive stochastic optimal control problems. The proposed near-optimal conflict resolution algorithm is based on a surrogate modeling technique called polynomial chaos kriging, which is used to construct the surrogate models of the optimal conflict resolution trajectories from a set of precomputed optimal solutions. The near-optimal conflict resolution trajec-

tories can be accurately generated in real time from the surrogate models with the information of current conditions (e.g., current states). The proposed near-optimal conflict resolution algorithm has the feature of optimal feedback control. When the states on the precomputed optimal trajectory deviate from the actual states due to the uncertainties, the proposed near-optimal feedback control method can accurately generate the near-optimal trajectory starting from the actual states in real time without solving another stochastic optimal control problem to obtain the correct optimal trajectory. Through illustrative aircraft conflict detection and resolution examples, the performance and effectiveness of the proposed conflict detection and resolution algorithms are evaluated and demonstrated.

Acknowledgments

First and foremost, I would like to express my sincere and deepest gratitude to my advisor, Professor Takeshi Tsuchiya for providing invaluable guidance and advice about my research and for giving me an excellent environment for research ever since I joined his research group in 2009. I would also like to express my gratitude to Professor Shinji Suzuki for giving me excellent advice and ideas. It has always been enjoyable not only to conduct the research but to participate in the projects of unmanned aerial vehicles.

I would like to express my gratitude to Professor Seiya Ueno, Professor Koichi Hori and Professor Takehisa Yairi who were on my dissertation defense committee and provided valuable and fruitful comments on my thesis. All their comments and advice helped me to improve my thesis and consider the other research viewpoints.

I owe special thanks to Professor Inseok Hwang for accepting me as a visiting scholar at the Purdue University for nine months from September 2013 to May 2014. During my stay at Purdue, I could accelerate and facilitate my research with valuable and fruitful discussion with him, Dr. Jian Wei and other colleagues at the Flight Dynamics and Control/Hybrid Systems Lab.

I would like to thank Mr. Naoki Matayoshi at the Japan Aerospace Exploration Agency (JAXA) for his meaningful comments and advice about this collaborative research with JAXA. I would also like to express my gratitude to Dr. Ryota Mori at the Electronic Navigation Research Institute (ENRI), Dr. Adriana Andreeva-Mori at JAXA and Mr. Tsuneharu Uemura for giving me their fruitful comments. Their invaluable comments and advice helped me to enhance my knowledge and insight on air traffic management.

I would like to thank Mr. Kenji Karasawa for his support and encouragement. I would also like to thank Dr. Jorg O. Entzinger, Dr. Christopher T. Raabe, Mr. Pramudita S. Palar, Mr. Takahiro Fujikawa and other members of the Suzuki and Tsuchiya Lab for enjoyable conversation and discussion on various topics.

I would like to acknowledge the financial support of the University of Tokyo and the Japan Society for the Promotion of Science who supported my research by a Grant-in-Aid for Research Fellows.

Last but not least, my special thanks go to my family. I would like to thank my parents for their understanding, encouragement and support without which I would never have been able to complete my dissertation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

More than a century has passed since the Wright brothers, Orville and Wilbur Wright, invented and flew the first powered aircraft Wright Flyer in 1903, and it has been a century since the first scheduled commercial airline flight took off with one passenger on January 1, 1914. Starting from one passenger, one aircraft and one route in January 1914, approximately 3.3 billion passengers flew with a fleet of about 25000 aircraft on almost 50000 routes in 2014 [1]. During the first century of commercial flight, aviation has dramatically transformed the world around us (e.g., the economic, social and cultural life) and has become an essential means of transportation. According to the International Civil Aviation Organization (ICAO) [2], an average annual air traffic growth rate is approximately 4.6 % over the past 15 years and expected to be about 4.5 % until 2030, and the air traffic demand is expected to double every 15 years.

The air traffic management (ATM) system is of vital importance for ensuring safety of air transportation. Despite many scientific and technological advances such as the flight management system (FMS), global positioning system (GPS) and automatic dependent surveillance-broadcast (ADS-B), the current ATM system has a centralized (mostly human-operated) architecture. In the current ATM system, air traffic controllers are ultimately responsible for safety, and the achievable capacity of

the ATM system is limited due to the human-operated nature of the ATM system. Since the air traffic has been growing rapidly over the past decades, the current ATM system is under considerable stress. To accommodate the increasing air traffic, ICAO published a new operational concept of global ATM in 2005 [3]. In order to support the new era of air transportation and the second century of commercial flight, the Next Generation Air Transportation System (NextGen) [4], the Single European Sky ATM Research (SESAR) [5] and the Collaborative Actions for Renovation of Air Traffic Systems (CARATS) [6] are currently ongoing programs in the United States, Europe and Japan, respectively. These new ATM programs are aimed at harmonizing air traffic operations, reducing the heavy workload of the air traffic controllers and improving the safety, efficiency, capacity and environmental impact of the current ATM system. Some concepts of these ATM programs have already been implemented, and various research and development activities are still in progress. Unlike the centralized architecture of the current ATM system, the future ATM system has the decentralized architecture so that aircraft (pilots) can take over some ATM tasks from the air traffic controllers on the ground by sharing the information between aircraft. By introducing the automated ATM system to assist and ultimately replace the tasks of the air traffic controllers, it is believed that the performance and efficiency of the ATM system can be improved and the tasks of the air traffic controllers can be simplified and alleviated. Accordingly, the automated tools will allow the air traffic controllers to handle the increasing air traffic demand and possibly enhance the safety level.

The primary concern of the ATM system and the air traffic controllers is to guarantee safety, and one of the major safety critical situations is an aircraft midair conflict when two or more aircraft experience a loss of the minimum allowed separation. In order to ensure safety, there are two important procedures to avoid a midair conflict: conflict detection and conflict resolution. Aircraft trajectories are predicted to identify potential conflicts in the conflict detection phase, and conflict resolution strategies are provided to avoid the predicted potential conflicts in the conflict resolution phase. Therefore, in this study, we focus on the aircraft conflict detection and resolution problem and propose the automated conflict detection and resolution

tools to support the air traffic controllers and provide the pilots with the automated advisories and conflict resolution trajectories to avoid potential conflicts.

1.2 Literature Review

Many approaches have been proposed for conflict detection and resolution, and most of the existing conflict detection and resolution algorithms can be categorized into two classes: deterministic and probabilistic approaches [7]. In the deterministic conflict resolution approaches, an artificial potential field algorithm [8] and game theory [9,10] are used to determine conflict resolution maneuvers. Global optimization approaches such as convex optimization [11] and mixed integer linear programming [12–14] are also employed for deterministic conflict resolution. In addition, model predictive control algorithms [15, 16] are proposed for determining optimal conflict resolution trajectories. Other optimization-based approaches [17–19] and protocol-based approaches [20–22] in a deterministic setting have also been proposed up to now. Many deterministic approaches mentioned above are well suited to real time applications for conflict resolution. In reality, however, aircraft fly in the presence of various uncertainties such as meteorological prediction errors, navigation errors and pilots' intents, and the deterministic approaches cannot provide robust conflict resolution trajectories at all under uncertain environments during the flight. Since aircraft trajectory prediction is inexact due to various uncertainties during the flight and the accuracy of aircraft trajectory prediction is significantly influenced by the uncertainties, it is of vital importance to take into account the effects of uncertainties on conflict detection and resolution. However, it makes the problem more complicated and computationally intensive to consider uncertainties.

In a probabilistic setting, the research efforts have concentrated mainly on conflict detection rather than on conflict resolution due to the complexity of stochastic dynamical models to quantify the effects of possible control inputs. For probabilistic conflict detection in the presence of uncertainties, probabilistic aircraft dynamical models incorporating uncertainties are used to predict future aircraft positions. As

the probabilistic aircraft motion model, the empirical distribution model that simply projects the current position of aircraft into the future is developed [23–27]. The dynamical model using stochastic differential equations is also proposed to incorporate the uncertainties into the aircraft dynamics [28, 29]. In addition, the probabilistic aircraft motion model based on the hybrid systems is developed to combine the continuous aircraft dynamics with the discrete aircraft's intent information such as flight plans [30, 31]. Using the probabilistic aircraft dynamical model, the conflict probability between aircraft is estimated to detect potential conflicts within a look ahead time horizon. Moreover, the probabilistic conflict resolution problem is often formulated as a stochastic optimal control problem to determine the optimal conflict resolution trajectory in the presence of uncertainties during the flight. In the previous works, the probabilistic conflict resolution approaches are commonly based on the time-consuming statistical methods such as Monte Carlo simulation [28], a Markov chain Monte Carlo framework [32] and Bayesian optimal design with the sequential Monte Carlo method [33]. Subliminal controller [34] and reachability analysis [35] with Monte Carlo simulation are also applied for probabilistic conflict resolution. In order to speed up stochastic simulations, computationally expensive sequential Monte Carlo optimization is implemented on graphics processing units (GPUs)[1] in the previous works [36, 37]. Although the computational cost can be drastically reduced by using Monte Carlo methods implemented on GPUs [38, 39], it depends completely on GPU computing. Furthermore, instead of the time-consuming Monte Carlo methods, the stochastic optimal control problem for conflict resolution is also solved by a Markov chain approximation and the Jacobi iteration [40]. However, as with the probabilistic conflict resolution algorithms mentioned above, the problem becomes intractable for the high-dimensional continuous state space and a large number of the discrete states.

[1]The use of GPUs has become popular for scientific computing because of their massively parallel processors, though GPUs are originally developed as the dedicated graphics cards for real time graphics rendering.

1.3 Research Objectives and Scope

Conflict detection and resolution is currently performed at three different levels depending on look ahead time horizons: long, medium and short terms. Long term conflict detection and resolution is usually referred to as air traffic flow management and performed for an entire airspace, over a time horizon of several hours. Long term conflict detection and resolution is performed before take-off and aimed at scheduling of aircraft routes and determining nominal flight plans. Medium term conflict detection and resolution is carried out by the air traffic controllers, over horizons of several tens of minutes. The main purpose of medium term conflict detection and resolution is to modify the preplanned flight plans in order to ensure safety and resolve potential conflicts. Short term conflict detection and resolution is also carried out by the air traffic controllers or on board the aircraft by the FMS, over horizons of seconds to minutes; the short term conflict alert (STCA) system [41] on the ground and the traffic alert and collision avoidance system (TCAS)[2] [43] on board the aircraft belong to this category.

In this dissertation, we propose novel probabilistic conflict detection and resolution algorithms in the presence of uncertainties, and focus on the short and medium terms (10–15 min look ahead time horizon) conflict detection and resolution problem especially for two kinds of conflicts: aircraft–aircraft and aircraft–weather[3] conflicts. The proposed conflict detection and resolution algorithms can provide the automated advisories and conflict resolution trajectories for the air traffic controllers and pilots and have a potential to ultimately replace the tasks of the air traffic controllers. On the basis of a stochastic dynamical model that takes into account uncertainties during flight, the possibility of future conflicts is evaluated. The proposed conflict detection algorithm allows the pilots and air traffic controllers to make a decision in real time on whether potential conflicts will occur within a look ahead time horizon. When the potential conflicts are detected by the conflict detection algorithm, the optimal conflict

[2]Research and development of airborne collision avoidance system X (ACAS X) [42] is currently underway to replace TCAS. ACAS X uses a probabilistic aircraft dynamical model in order to consider uncertainties during flight.

[3]A moving convective weather region that should be avoided is considered.

resolution strategy for an aircraft is determined to avoid other aircraft and a moving convective weather region by the conflict resolution algorithm. In particular, to make suitable for the practical applications of the ATM system, we propose a novel stochastic near-optimal control method to generate conflict resolution trajectories and maneuvers in the presence of uncertainties in real time without actually solving stochastic optimal control problems. The proposed near-optimal conflict resolution algorithm enables the pilots to determine optimal conflict resolution trajectories and maneuvers in real time based on the information of current conditions obtained from onboard equipment such as ADS-B.

As to the uncertainties during the flight, various uncertainties such as the meteorological prediction error, navigation errors and pilots' intents were studied in the previous works [23–28,30–32,34,35]. Among the various uncertainties, the wind prediction error was considered as the primary uncertainty because it has significant influence on the aircraft trajectory compared to other possible uncertainties. In this study, the wind prediction error, especially the spatially correlated wind error [29, 30, 44], is considered because the wind correlation usually has a significant effect on the trajectories of aircraft that are close to each other, and therefore conflict detection and resolution [30]. However, the aircraft dynamics containing the spatially correlated wind prediction error become complex because of the complicated stochastic models, and nonlinear optimal control problems for such complex dynamical systems are challenging to solve and require sophisticated optimization approaches. In this study, we propose a novel probabilistic conflict detection algorithm by employing the polynomial chaos expansion (PCE) method [45–47], which can quantify uncertainties in the complex nonlinear dynamical systems with high computational efficiency. To detect potential conflicts, the conflict probability between aircraft is estimated by the probabilistic conflict detection algorithm based on the PCE method. For the conflict resolution problem, we apply the pseudospectral method [48–50], which can solve deterministic nonlinear optimal control problems effectively. A numerical algorithm incorporating the PCE method into the pseudospectral method is proposed to deal with stochastic elements and solve the challenging stochastic optimal control

problems. The stochastic optimal control method is combined with the probabilistic conflict detection algorithm to guarantee the resolution of potential conflicts in the presence of uncertainties. Moreover, inspired by a spatial statistical approach for synthesizing near-optimal feedback controllers [51, 52], we propose a near-optimal conflict resolution algorithm based on a recently developed surrogate modeling technique called polynomial chaos kriging [53]. The surrogate model, also known as the response surface model and metamodel, can mimic the input-output behavior of an original simulation, and an output can be obtained from a certain input without actually executing the original simulation. Polynomial chaos kriging is a recently developed hybrid algorithm based on two surrogate modeling techniques: PCE [45–47] and kriging [54, 55]. In the previous work [53], the performance of the hybrid polynomial chaos kriging algorithm is demonstrated using analytical benchmark functions, and polynomial chaos kriging generally performs better than PCE or kriging does. In this study, we apply the polynomial chaos kriging method to the practical engineering problem, i.e., the optimal control problem for conflict resolution. By using the polynomial chaos kriging method, the surrogate models of the optimal conflict resolution trajectories are constructed from a set of the precomputed optimal solutions. The near-optimal conflict resolution trajectories in the presence of uncertainties can be accurately estimated in real time from the surrogate models with the information of current conditions (e.g., current states) without actually solving the computationally expensive stochastic optimal control problems. The proposed near-optimal conflict resolution algorithm has the feature of optimal feedback control. When the states on the precomputed optimal trajectory deviate from the actual states due to the uncertainties, our proposed near-optimal feedback control method can accurately generate the near-optimal trajectory starting from the actual states in real time without solving another stochastic optimal control problem to obtain the correct optimal trajectory. In order to consider more general conflict resolution problem, as to the uncertainties, the airspeed measurement error and the uncertainty contained in the moving convective weather region are considered as well as the spatially correlated wind error for the near-optimal conflict resolution problem. Through numerical simulations, we

demonstrate that the proposed probabilistic conflict detection and resolution algorithms can provide the automated advisories and conflict resolution trajectories for the air traffic controllers and pilots in real time.

1.4 Organization of Dissertation

The paper is organized as follows. Chapter 2 presents the stochastic aircraft dynamics and the probabilistic conflict detection algorithm employing the PCE method. In Chapter 3, the stochastic optimal control method for conflict resolution is developed. In Chapter 4, the surrogate modeling technique is introduced to generate near-optimal conflict resolution trajectories in real-time. In Chapter 5, the aircraft conflict detection and resolution problem is formulated and solved. Through numerical simulations, the effectiveness and performance of the probabilistic conflict detection and resolution algorithms are evaluated and demonstrated. The dissertation ends with conclusions and future research directions in Chapter 6.

Chapter 2

Probabilistic Conflict Detection

In this chapter, we introduce the stochastic aircraft dynamics including spatially correlated wind uncertainty, and propose a conflict detection algorithm based on the PCE method.

2.1 Stochastic Aircraft Dynamics

We consider the aircraft midair conflicts in the two-dimensional horizontal plane in which the aircraft coming from different directions merge to the waypoint. Note that though merging operations are considered for illustration, the proposed algorithm can be applied to any phase of flight. The aircraft dynamics are given by the following point mass model with three state variables $\boldsymbol{x} = (x, y, \psi)^T$ and one control variable u:

$$\dot{x} = \frac{v \cos\psi + w_x}{3600} \tag{2.1}$$

$$\dot{y} = \frac{v \sin\psi + w_y}{3600} \tag{2.2}$$

$$\dot{\psi} = u \tag{2.3}$$

where x and y are the Cartesian coordinates in nautical miles (nmi); ψ is the heading angle; v is the true airspeed in knots (kt); w_x and w_y are the stochastic wind velocities in knots (kt) in the x and y directions, respectively; and the constant values in

Eqs. (2.1) and (2.2) are the unit conversion factors. As to the wind uncertainty, the wind model contains the deterministic and stochastic components. The deterministic wind component represents the meteorological predictions, e.g., the periodical meteorological prediction data provided by the Japan Meteorological Agency in Japan and the rapid update cycle data from the National Oceanic and Atmospheric Administration in the United States. The effects of the deterministic wind component on the aircraft trajectory can be known a priori because the meteorological prediction data can be obtained and considered for the aircraft trajectory prediction ahead of time. In this study, the deterministic component representing the meteorological prediction is ignored and set to zero for simplicity, because we are interested in the effects of the stochastic nature of the wind model. The wind model accounts for only the stochastic component, i.e., the wind prediction error representing the uncertainty in the deterministic meteorological prediction. Thus, the wind velocities w_x and w_y are referred to the wind prediction errors. The meteorological predictions and wind errors are slowly changing with time in reality. Thus, the wind error is assumed to be time-invariant, because the time horizon for conflict detection and resolution considered in this study is short (approximately 10 min) and the temporal change in the wind error is small [40].

In order to describe the wind errors realistically, the spatially correlated wind model is considered, because the wind correlation usually has a significant effect on the trajectories of aircraft that are close to each other and therefore conflict detection and resolution. From the correlated wind model [30], which is constructed based on the comparison between the real historical air traffic data and wind forecast data [44], $w_x(x, y)$ and $w_y(x, y)$ are assumed to be Gaussian random processes with zero mean and the following exponential covariance function:

$$C\left((x, y), (x', y')\right) = \sigma_w^2 \exp\left(-\mu_x |x - x'|\right) \exp\left(-\mu_y |y - y'|\right) \tag{2.4}$$

where σ_w is the standard deviation of the wind error and set to 5.35 m/s (= 10.40 kt) [30,56]; and the parameters μ_x and μ_y are set to the same value of $1/182$ nmi^{-1} [30,44].

σ_w is determined based on the comparison between the real historical wind forecast data and aircraft reports collected by the aircraft communications, addressing, and reporting system (ACARS) observations in the previous work [56]. Note that σ_w is estimated in consideration of the errors in the ACARS observations [57], and the ACARS observation errors and wind forecast errors are independent at the time of their observations. As the distance difference increases, the correlation described in Eq. (2.4) decays exponentially. The Gaussian random processes $w_x(x, y)$ and $w_y(x, y)$ are approximated as a linear combination of deterministic functions multiplied by independent random variables using the Karhunen–Loève (KL) expansion [62, 63]:

$$w_x(x, y) = \sum_{i=1}^{N_{\mathrm{KL}}} \left(\sqrt{\lambda_i} g_i(x, y) X_{x,i} \right) \tag{2.5}$$

$$w_y(x, y) = \sum_{i=1}^{N_{\mathrm{KL}}} \left(\sqrt{\lambda_i} g_i(x, y) X_{y,i} \right) \tag{2.6}$$

where $X_{x,i}$ and $X_{y,i}$ $(i = 1, \ldots, N_{\mathrm{KL}})$ are the independent standard Gaussian random variables; N_{KL} is the number of independent random variables for each of $w_x(x, y)$ and $w_y(x, y)$; and λ_i and $g_i(x, y)$ $(i = 1, \ldots, N_{\mathrm{KL}})$ are the eigenvalue and eigenfunction of the following integral equation in descending order of the magnitude of the eigenvalue λ_i, respectively:

$$\lambda_i g_i(x, y) = \int_{\boldsymbol{D}} C\left((x, y), (x', y')\right) g_i(x', y') dx' dy'$$

where x and y are defined over a given domain $\boldsymbol{D}$. Thus, the wind error is represented as the spatially correlated wind error with the finite number of independent random variables by using the KL expansion, and it makes the differential equations in Eqs. (2.1) and (2.2) more manageable. (See Appendix A and [63] for more detailed discussion of KL expansion.)

Various statistical wind models representing the wind errors are proposed and mostly based on Gaussian random processes [30, 44, 58, 59]. The Gaussian process model for the wind uncertainty [30] is widely used in ATM research such as air-

craft trajectory prediction [60, 61] and conflict detection and resolution [30, 33–35]. However, to the best of the author's knowledge, there is no conclusive experimental evidence to support this assumption, and it is obviously required to demonstrate the validity of the wind model in future work. In addition, other wind models with different probability distributions may be also needed for specific weather conditions and terrains. We would like to note that our proposed algorithm can be extended to these cases because it is based on the PCE method that can deal with the stochastic models with various distributions.

A conflict between aircraft is defined as a situation where two or more aircraft experience a loss of the minimum required separation established by ICAO [64]. Computing the distance between each pair of aircraft, we can identify the potential conflicts. To avoid the conflict, the two aircraft need to satisfy the following safety constraint:

$$d_{H\min} \leq L_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \qquad (\forall i, j \in \{1, \ldots, N_a\} : i < j) \tag{2.7}$$

where the subscript i and j denote the ith and jth aircraft; N_a is the total number of aircraft; $d_{H\min}$ is the horizontal separation requirement; and L_{ij} is the horizontal distance between the ith and jth aircraft. Note that the shortest distance between aircraft and the convective weather region is computed for the aircraft–weather conflict. The aircraft positions x and y in Eq. (2.7) become random variables because Eqs. (2.1) and (2.2) contain the stochastic terms w_x and w_y. Since x and y are random variables, the horizontal distance between aircraft L given by Eq. (2.7) also becomes a random variable. Since there is not a simple closed-form expression for L, it can only be computed numerically. In this study, we propose a novel conflict detection algorithm based on the computationally efficient PCE method to calculate L. To detect potential conflicts, L is computed by the PCE algorithm, which is described in more detail in Section 2.2, and therefore the conflict probability between aircraft can be estimated as described in Section 2.3.

2.2 Polynomial Chaos Expansion

Wiener first introduced the homogeneous chaos, also known as the Hermite chaos, and used Hermite polynomials to approximate Gaussian processes in the 1930s [65]. According to the Cameron–Martin theorem [66], the Hermite chaos converges in the L_2 sense [62,67]. While the Hermite chaos is useful for the analysis of the stochastic processes, it was applied to quantify uncertainties in physical applications. In particular, Ghanem and Spanos pioneered to combine the Hermite chaos with a finite element method for solid mechanics applications [63]. The Hermite chaos was extended as the generalized polynomial chaos (gPC) method [45] for the analysis of various types of stochastic processes using the corresponding orthogonal polynomials [68]. The arbitrary polynomial chaos (aPC) method was also developed for arbitrary distributions with arbitrary probability measure [69]. The PCE method have been applied to many engineering problems: solid mechanics [63, 70–73], fluid mechanics [74–78] and multibody dynamics [79, 80] to name but a few.

With the PCE method, the stochastic model response can be expanded as the summation of the orthogonal polynomials of the independent random variables, which can be described as the following equation:

$$Y(\boldsymbol{X}) = \sum_{m=1}^{\infty} C_m \Phi_m(\boldsymbol{X}) \tag{2.8}$$

where $Y(\boldsymbol{X}) \in \mathbb{R}$ is the stochastic model response; $\boldsymbol{X} = (X_1, \ldots, X_{N_X})^T \in \mathbb{R}^{N_X}$ is the independent random variables; $C_m \in \mathbb{R}$ is the unknown expansion coefficient to be estimated; $\Phi_m(\boldsymbol{X}) \in \mathbb{R}$ is the multivariate orthogonal polynomial basis function; and N_X is the number of random variables. $\Phi_m(\boldsymbol{X})$ is obtained from the l_ith order ($l_i \in \mathbb{N}$) univariate polynomial basis function $\phi_i^{(l_i)}(X_i) \in \mathbb{R}$ of each random variable X_i ($i = 1, \ldots, N_X$) by the tensor product rule:

$$\Phi_m(\boldsymbol{X}) = \prod_{i=1}^{N_X} \phi_i^{(l_i)}(X_i) \tag{2.9}$$

Table 2.1: Probability distribution of random variable and corresponding polynomial basis function [45]

	Distributions	Orthogonal polynomials
Continuous	Gaussian	Hermite
	Uniform	Legendre
	Gamma	Laguerre
	Beta	Jacobi
Discrete	Poisson	Charlier
	Binomial	Krawtchouk
	Negative binomial	Meixner
	Hypergeometric	Hahn

Note that a combination of l_i, $(l_1,\ldots,l_{N_X})$, is unique for each subscript m in Eq. (2.9). The normalized orthogonal (orthonormal) polynomial is used by satisfying the following orthonormality condition:

$$\mathrm{E}[\phi_i^{(j)}(X_i)\phi_i^{(k)}(X_i)] = \int \phi_i^{(j)}(X_i)\phi_i^{(k)}(X_i)\rho_i(X_i)\mathrm{d}X_i = \delta_{jk}$$

where $\mathrm{E}[\cdot]$ denotes the expectation operator; δ_{jk} is the Kronecker delta function that takes 1 if $j = k$ and 0 otherwise; and $\rho_i(X_i) \in \mathbb{R}$ is the probability density function corresponding to the ith random variable X_i. Since $\Phi_m(\boldsymbol{X})$ in Eq. (2.9) is the product of univariate orthonormal polynomials $\phi_i^{(l_i)}(X_i)$ $(i = 1,\ldots,N_X)$, it is clear that:

$$\mathrm{E}[\Phi_j(\boldsymbol{X})\Phi_k(\boldsymbol{X})] = \delta_{jk}$$

The best choice of the orthonormal polynomials depends on the type of $\rho_i(X_i)$ to achieve better convergence [45], and some of the probability distributions and corresponding orthogonal polynomials are listed in Table 2.1. For example, Hermite polynomials are used with the Gaussian random variables. (See [45] for more detailed discussion.) In this study, the Gaussian random variables are considered for the wind errors in Eqs. (2.5) and (2.6), and accordingly Hermite polynomials are used for the basis functions. (See Appendix B for more information on the orthogonal polynomials.)

For computational purpose, the infinite series in Eq. (2.8) is truncated in order

to retain a finite number of terms. There are several ways to truncate the PCE in Eq. (2.8) and select the number of polynomials or expansion coefficients, e.g., empirical truncation scheme [85] and hyperbolic truncation scheme [47]. The commonly used simple truncation scheme, which we employ in this study, is that the total degree of polynomials is not larger than $P \in \mathbb{N}$. That is, the maximum total degree of polynomials is P, which is referred to as the Pth order PCE approximation. The Pth order approximation of $Y(\boldsymbol{X})$ is written as the following equation:

$$Y_P(\boldsymbol{X}) = \sum_{m=1}^{M} C_m \Phi_m(\boldsymbol{X}) = \boldsymbol{\Phi}^T(\boldsymbol{X})\boldsymbol{C} \tag{2.10}$$

where $Y_P(\boldsymbol{X})$ is the Pth order approximation of $Y(\boldsymbol{X})$; M is the total number of polynomial basis functions or expansion coefficients; $\boldsymbol{C} = (C_1, \ldots, C_M)^T \in \mathbb{R}^M$ is the vector of the expansion coefficients C_m $(m = 1, \ldots, M)$; and $\boldsymbol{\Phi}(\boldsymbol{X}) = (\Phi_1(\boldsymbol{X}), \ldots, \Phi_M(\boldsymbol{X}))^T \in \mathbb{R}^M$ is the vector of the multivariate orthonormal polynomial basis functions $\Phi_m(\boldsymbol{X})$ $(m = 1, \ldots, M)$. In the truncation scheme, l_i in Eq. (2.9) satisfies the following condition:

$$p_m = \sum_{i=1}^{N_X} l_i \leq P$$

where p_m $(m = 1, \ldots, M)$ is the sum of the order of the univariate polynomial of the ith random variable $\phi_i^{(l_i)}(X_i)$ in Eq. (2.9); and P is the approximation order of $Y(\boldsymbol{X})$ and the maximum total degree of the multivariate polynomial Φ_m. Note that when $m = 1$, $\Phi_1(\boldsymbol{X})$ is set to the zero-order polynomial basis ($p_1 = 0$) that takes 1. (See Appendix B.) In this truncation strategy, since the total degree of the polynomial basis function p_m $(m = 1, \ldots, M)$ is not larger than P, the total number of polynomial basis functions M is determined by the binomial coefficient: $M = \binom{N_X+P}{N_X}$.

The unknown PCE coefficient C_m in Eq. (2.10) can be estimated using either an intrusive or a nonintrusive approach. With respect to implementation, a disadvantage of the intrusive approach using a Galerkin projection [45, 63] is that it can be cumbersome and difficult to implement for complex nonlinear systems [81]. In contrast, the nonintrusive methods are much more convenient to deal with general

nonlinear problems [81]. There are various nonintrusive approaches: the projection approach [74, 75, 82], the stochastic collocation method [46, 81], the regression approach [47, 83–86] and other approaches [87–90]. The nonintrusive approach is straightforward to implement because it uses the sample points of input random variables and repetitive executions of deterministic simulations. Therefore, in this study, the nonintrusive PCE method is implemented to determine C_m. In particular, we employ two different forms of the PCE methods: the stochastic collocation method [46, 81] and the least angle regression (LARS) algorithm [47]. For a small number of input random variables, the stochastic collocation method is employed because it uses the strategically selected sample points, i.e., collocation points, of the random variables and a significantly small number of collocation points with the sparse grid quadrature based on the Smolyak rule [91, 92]. On the other hand, the LARS algorithm is a computationally efficient approach especially for a large number of input random variables (more than 10 variables). Thus, we use either of these two forms of the PCE methods according to the number of input random variables: the stochastic collocation method is used for dealing with the wind errors and the LARS method is employed for the input variables of the surrogate models.

2.2.1 Stochastic Collocation Method

With the stochastic collocation method [46, 81], C_m in Eq. (2.10) can be obtained as follows. Since $\Phi_m(\boldsymbol{X})$ is the orthonormal polynomial, C_m can be determined by the following equation:

$$C_m = \mathrm{E}[Y(\boldsymbol{X})\Phi_m(\boldsymbol{X})] = \int Y(\boldsymbol{X})\Phi_m(\boldsymbol{X})\rho(\boldsymbol{X})\mathrm{d}\boldsymbol{X} \tag{2.11}$$

where $\rho(\boldsymbol{X}) \in \mathbb{R}$ is the joint probability density function:

$$\rho(\boldsymbol{X}) = \prod_{i=1}^{N_X} \rho_i(X_i)$$

Since $\boldsymbol{X} = (X_1, \ldots, X_{N_X})^T$ are the independent random variables, $\rho(\boldsymbol{X})$ is obtained from the univariate probability density function by the tensor product rule. The integral in Eq. (2.11) can be approximated by the Gaussian quadrature. On the basis of the quadrature rule, a set of collocation points and quadrature weights is chosen. The q-point univariate quadrature operator $\mathcal{U}^q$ approximates the polynomial $\phi(X) \in \mathbb{R}$ by using the set of q collocation points $X^{(j)} \in \mathbb{R}$ and associated weights $\alpha^{(j)} \in \mathbb{R}$ $(j = 1, \ldots, q)$:

$$\mathcal{U}^q[\phi(X)] = \sum_{j=1}^{q} \phi(X^{(j)})\alpha^{(j)} \approx \int_{-\infty}^{\infty} \phi(X)\rho(X)\mathrm{d}X$$

As q gets larger, the accuracy of the quadrature can be increased. The collocation points are the roots of the orthogonal polynomial, and the quadrature weight $\alpha^{(j)}$ satisfies the following condition:

$$\sum_{j=1}^{q} \alpha^{(j)} = 1$$

Note that a set of collocation points and quadrature weights depends on the type of $\rho(X)$. (See Appendix B.) The N_X-dimensional quadrature is readily derived from the univariate quadrature by the tensor product rule:

$$\mathcal{T}^{q,N_X} = \mathcal{U}_1^q \otimes \cdots \otimes \mathcal{U}_{N_X}^q$$

where $\mathcal{T}^{q,N_X}$ is the N_X-dimensional tensor grid quadrature operator based on the q-point univariate quadrature; $\mathcal{U}_i^q$ $(i = 1, \ldots, N_X)$ is the q-point univariate quadrature for the ith random variable X_i; and $\otimes$ denotes the tensor product. The total number of collocation points is q^{N_X} $(= \prod_{i=1}^{N_X} q)$ determined by the tensor product rule. A set of collocation points and weights is also determined by the tensor product rule. In general, as the number of random variables N_X gets larger, the tensor grid $\mathcal{T}^{q,N_X}$ suffers from the *curse of dimensionality*. Thus, we employ the sparse grid quadrature based on the Smolyak rule [46,91–93]. The sparse grid with Q collocation points $\boldsymbol{X}^{(j)}$

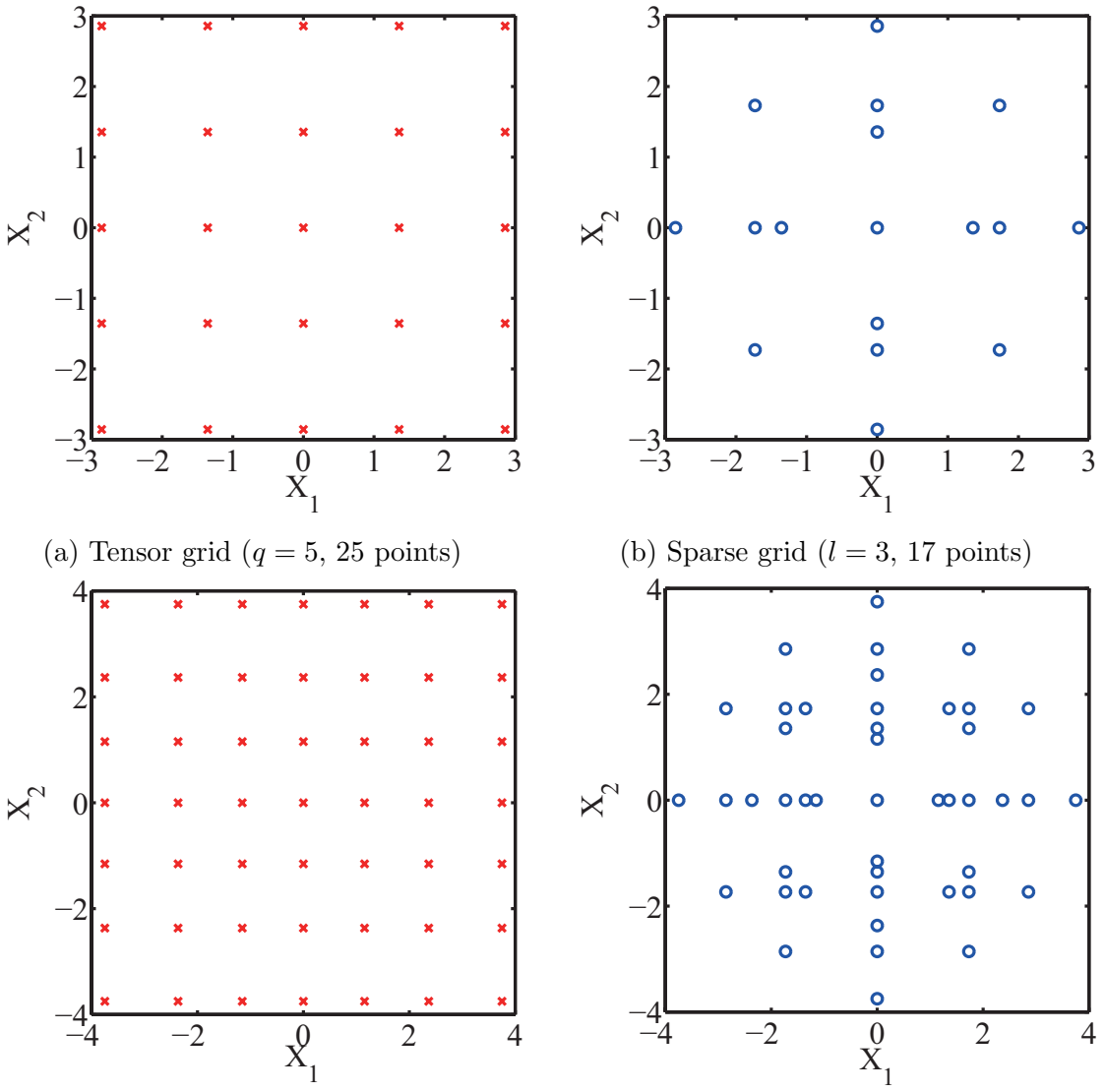
and associated weights $\alpha^{(j)}$ $(j = 1, \ldots, Q)$ consists of a much smaller number of collocation points than that of the tensor grid, and it can reduce the computational cost significantly. To reduce the number of collocation points, the Smolyak approach uses a strategically chosen linear combination of the tensor grid while retaining the accuracy of the quadrature. On the basis of the Smolyak rule, the N_X-dimensional sparse grid quadrature is derived from the univariate quadrature, and the N_X-dimensional sparse grid quadrature operator $\mathcal{S}^{l,N_X}$ is given by either of the following two forms:

$$\mathcal{S}^{l,N_X} = \sum_{N_X+1 \leq |\boldsymbol{k}| \leq N_X+l} (\Delta^{k_1} \otimes \cdots \otimes \Delta^{k_{N_X}}) \tag{2.12}$$

$$\mathcal{S}^{l,N_X} = \sum_{N_X+1 \leq |\boldsymbol{k}| \leq N_X+l} (-1)^{N_X+l-|\boldsymbol{k}|} \begin{pmatrix} N_X - 1 \\ N_X + l - |\boldsymbol{k}| \end{pmatrix} (\mathcal{U}^{k_1} \otimes \cdots \otimes \mathcal{U}^{k_{N_X}}) \tag{2.13}$$

where $l \in \mathbb{N}$ is the accuracy level of the sparse grid; $|\boldsymbol{k}| = \sum_{i=1}^{N_X} k_i$ is the multi-index $(k_i \in \mathbb{N})$; and Δ^{k_i} is given by $\Delta^{k_i} = \mathcal{U}^{k_i} - \mathcal{U}^{k_{i-1}}$ and $\mathcal{U}^{k_0} = 0$. The number of univariate nodes q_i for the ith random variable X_i is commonly set to 2^{k_i-1} $(i = 1, \ldots, N_X)$, and the nested nodes [93] is employed in this study. As l gets larger, the accuracy of the sparse grid quadrature can be increased. The number of collocation points Q in the sparse grid quadrature is uniquely determined by the accuracy level l and the dimension N_X. The original stochastic problem is transformed into the deterministic problem at each collocation point and can be solved by repetitive application of a deterministic solver.

Figure 2-1 shows the collocation points for two-dimensional random variables (X_1, X_2) obtained with the tensor and sparse grids. Figures 2-1a and 2-1b show the two-dimensional collocation points based on the same univariate grids in the tensor and sparse grids, respectively. Figures. 2-1c and 2-1d also show the collocation points based on the same univariate grids. As shown in Fig. 2-1, it is clear that the sparse grids consist of a much smaller number of collocation points than that of the tensor grids. With the sparse grid quadrature based on the Smolyak rule, we can reduce the computational cost compared with the tensor grid quadrature.



(a) Tensor grid ($q = 5$, 25 points)

(b) Sparse grid ($l = 3$, 17 points)

(c) Tensor grid ($q = 7$, 49 points)

(d) Sparse grid ($l = 4$, 45 points)

Figure 2-1: Collocation points with tensor and sparse grids for two-dimensional random variables.

By using the stochastic collocation method based on the sparse grid quadrature rule in Eq. (2.13), the approximation of C_m in Eq. (2.11) can be given by the following equation:

$$\hat{C}_m = \sum_{j=1}^{Q} Y(\boldsymbol{X}^{(j)}) \Phi_m(\boldsymbol{X}^{(j)}) \alpha^{(j)} \tag{2.14}$$

where $\hat{C}_m$ is the approximation of C_m; and $Y(\boldsymbol{X}^{(j)})$ denotes the deterministic model response at the jth collocation point $\boldsymbol{X}^{(j)}$ $(j = 1, \ldots, Q)$. Thus, the approximate stochastic model response $Y_P(\boldsymbol{X})$ is determined by Eqs. (2.10) and (2.14) as the orthonormal polynomials of the random variables $\boldsymbol{X}$. As described in Eqs. (2.10) and (2.14), $Y_P(\boldsymbol{X})$ is the distribution function of $\boldsymbol{X}$ and can be evaluated for any given random inputs.

The procedures to determine the stochastic model response $Y(\boldsymbol{X})$ with the stochastic collocation method are described as follows:

1. Generate a set of Q collocation points of random variables $\boldsymbol{X}^{(j)}$ and associated weights $\alpha^{(j)}$ $(j = 1, \ldots, Q)$ based on the sparse grid quadrature in Eq. (2.13).

2. Calculate the value of the orthonormal polynomial $\Phi_m(\boldsymbol{X}^{(j)})$ at each collocation point $\boldsymbol{X}^{(j)}$ $(j = 1, \ldots, Q,\ m = 1, \ldots, M)$ in Eq. (2.9).

3. Determine the deterministic model response $Y(\boldsymbol{X}^{(j)})$ at each collocation point $\boldsymbol{X}^{(j)}$ $(j = 1, \ldots, Q)$.

4. Compute the PCE coefficient $\hat{C}_m$ $(m = 1, \ldots, M)$ in Eq. (2.14).

5. Determine the approximate stochastic model response $Y_P(\boldsymbol{X})$ in Eq. (2.10).

2.2.2 Least Angle Regression

Another approach to obtain $\boldsymbol{C}$ in Eq. (2.10) is the regression method [47, 83]. With the regression method, $\boldsymbol{C}$ can be computed by minimizing the variance of the residual $Y(\boldsymbol{X}) - \boldsymbol{\Phi}^T(\boldsymbol{X})\boldsymbol{C}$:

$$\hat{\boldsymbol{C}} = \underset{\boldsymbol{C} \in \mathbb{R}^M}{\arg\min}\, \mathrm{E}\left[\left(Y\left(\boldsymbol{X}\right) - \boldsymbol{\Phi}^T(\boldsymbol{X})\boldsymbol{C} \right)^2 \right] \tag{2.15}$$

where $\hat{\boldsymbol{C}}$ is the least squares estimate of $\boldsymbol{C}$; and the mean of the residual is assumed to be zero in the regression method. The solution of Eq. (2.15) can be given by the following equation:

$$\hat{\boldsymbol{C}} = \left(\mathrm{E}\left[\boldsymbol{\Phi}(\boldsymbol{X})\boldsymbol{\Phi}^T(\boldsymbol{X})\right]\right)^{-1}\mathrm{E}\left[\boldsymbol{\Phi}(\boldsymbol{X})Y(\boldsymbol{X})\right] \tag{2.16}$$

In order to compute Eq. (2.16), we generate N_s sample points of random variables $\boldsymbol{X}^{(j)}$ and the corresponding deterministic model responses on the sample points $Y(\boldsymbol{X}^{(j)})$ $(j = 1, \ldots, N_s)$. Using the N_s sample points and associated model responses, Eq. (2.16) leads to the following equation:

$$\hat{\boldsymbol{C}} = \left(\frac{1}{N_s}\sum_{j=1}^{N_s}\boldsymbol{\Phi}(\boldsymbol{X}^{(j)})\boldsymbol{\Phi}^T(\boldsymbol{X}^{(j)})\right)^{-1}\left(\frac{1}{N_s}\sum_{j=1}^{N_s}\boldsymbol{\Phi}(\boldsymbol{X}^{(j)})Y(\boldsymbol{X}^{(j)})\right)$$

which can be rewritten as:

$$\hat{\boldsymbol{C}} = \left(\boldsymbol{F}^T\boldsymbol{F}\right)^{-1}\boldsymbol{F}^T\boldsymbol{Y} \tag{2.17}$$

where $\boldsymbol{F}$ is the matrix of the polynomial basis functions defined as $\boldsymbol{F} = (\boldsymbol{\Phi}(\boldsymbol{X}^{(1)}), \ldots, \boldsymbol{\Phi}(\boldsymbol{X}^{(N_s)}))^T$; and $\boldsymbol{Y}$ is the vector of the model responses defined as $\boldsymbol{Y} = (Y(\boldsymbol{X}^{(1)}), \ldots, Y(\boldsymbol{X}^{(N_s)}))^T$.

When performing the regression approach, N_s should be larger than M, which is the total number of the expansion coefficients $\boldsymbol{C}$ and determined by $M = \binom{N_X+P}{N_X}$ in the PCE truncation scheme. In practice, N_s is generally set to two to three times as much as M [47]. As N_X gets larger, the ordinary least squares regression approach suffers from the *curse of dimensionality*. In addition, with the stochastic collocation method, as N_X increases, the number of collocation points is considerably larger even though the sparse grid is used. Therefore, for a large number of input random variables ($N_X \geq 10$), we employ the LARS method [47,94] to determine $\boldsymbol{C}$ with high computational efficiency.

The LARS method is an efficient model selection algorithm especially for high-

dimensional data. With the LARS algorithm, we can select the polynomial basis functions (predictors) that have the greatest influence on the model response among a large number of candidate predictors. In practice, the model response depends only on a limited number of predictors, and it is not needed to contain all predictors to express the model response [47]. The LARS method can select a limited number of PCE coefficients $\boldsymbol{C}$ and polynomial terms $\boldsymbol{\Phi}(\boldsymbol{X})$ in the approximate stochastic model response $Y_P(\boldsymbol{X})$ compared to the usual number of coefficients or polynomial terms $M = \binom{N_X+P}{N_X}$. Accordingly, the LARS method can provide the sparse PCE approximation.

We generate N_s sample points of random variables $\boldsymbol{X}^{(j)}$ $(j = 1, \ldots, N_s)$ and the corresponding model response vector $\boldsymbol{Y} = \left(Y(\boldsymbol{X}^{(1)}), \ldots, Y(\boldsymbol{X}^{(N_s)})\right)^T$. The procedures to determine the PCE coefficients $\boldsymbol{C}$ and polynomial terms $\boldsymbol{\Phi}(\boldsymbol{X})$ with the original LARS algorithm are described as follows [47, 94]:

1. Initialize the PCE coefficients $\boldsymbol{C}$ and residual to zeros and $\boldsymbol{Y}$, respectively.

2. Find the vector of the polynomials (predictor) $\boldsymbol{\Phi}_j = (\Phi_j(\boldsymbol{X}^{(1)}), \ldots, \Phi_j(\boldsymbol{X}^{(N_s)}))^T$ that has the highest correlation with the current residual among the candidate predictors $\boldsymbol{\Phi}_m$ $(m = 1, \ldots, M)$.

3. Move $C_j^{(1)}$ (associated PCE coefficient of $\boldsymbol{\Phi}_j$) from zero to the least squares coefficient of the current residual on $\boldsymbol{\Phi}_j$, until another predictor $\boldsymbol{\Phi}_k$, which has as much correlation with the current residual as $\boldsymbol{\Phi}_j$ does, can be found.

4. Move jointly $C_j^{(2)}$ and $C_k^{(2)}$ in the direction defined by their least squares coefficients of the current residual on $\boldsymbol{\Phi}_j$ and $\boldsymbol{\Phi}_k$, until another predictor $\boldsymbol{\Phi}_l$, which has as much correlation with the current residual, can be found.

5. Continue this way until $M_{\text{LARS}} = \min(M, N_s - 1)$ predictors can be determined.

$\hat{\boldsymbol{Y}}^{(i)}$ $(i = 1, \ldots, M_{\text{LARS}})$ is the estimation of $\boldsymbol{Y}$ with selected i predictors, i.e., $\hat{\boldsymbol{Y}}^{(1)} = C_j^{(1)}\boldsymbol{\Phi}_j$ in Step 3 and $\hat{\boldsymbol{Y}}^{(2)} = C_j^{(2)}\boldsymbol{\Phi}_j + C_k^{(2)}\boldsymbol{\Phi}_k$ in Step 4. $\hat{\boldsymbol{Y}}^{(i)}$ retains i predictors, and eventually M_{LARS} approximate model responses $\hat{\boldsymbol{Y}}^{(i)}$ $(i = 1, \ldots, M_{\text{LARS}})$ are

constructed. Note that the last step of the LARS algorithm provides the ordinary least squares solution if $N_s \geq M$. In this study, we use the hybrid LARS approach [47,94], which is a variant of the original LARS method. In the hybrid LARS method, the coefficients associated with the retained predictors $\boldsymbol{C}^{(i)} \in \mathbb{R}^i$ $(i = 1, \ldots, M_{\mathrm{LARS}})$ are estimated by the ordinary least squares regression approach in Eq. (2.17). That is, $C_j^{(1)}$ in Step 3 and $(C_j^{(2)}, C_k^{(2)})^T$ in Step 4 are computed by not the LARS approach but the ordinary least squares regression approach (Eq. (2.17)). The hybrid LARS algorithm selects only a set of predictors, whereas the original LARS algorithm provides both the coefficients and predictors.

With the LARS method, M_{LARS} approximate model responses $\hat{\boldsymbol{Y}}^{(i)}$ $(i = 1, \ldots, M_{\mathrm{LARS}})$ are constructed, and the optimal approximate model response among $\hat{\boldsymbol{Y}}^{(i)}$ $(i = 1, \ldots, M_{\mathrm{LARS}})$ can be determined by using the leave-one-out cross validation technique [47]. By using the leave-one-out cross validation approach, the optimal model response that has the smallest leave-one-out error among M_{LARS} model responses can be determined. The leave-one-out error Err_{loo} is estimated as the following equation:

$$Err_{loo} = \frac{1}{N_s} \sum_{j=1}^{N_s} {\Delta^{(j)}}^2$$

where $\Delta^{(j)}$ is the predicted residual defined as the following equation:

$$\Delta^{(j)} = Y(\boldsymbol{X}^{(j)}) - \hat{Y}^{(-j)}(\boldsymbol{X}^{(j)}) \qquad (j = 1, \ldots, N_s)$$

where $Y(\boldsymbol{X}^{(j)})$ is the exact model response at the jth sample point $\boldsymbol{X}^{(j)}$; and $\hat{Y}^{(-j)}(\boldsymbol{X}^{(j)})$ is the approximate model response at $\boldsymbol{X}^{(j)}$, which is built when removing the jth sample point from all N_s sample points. In the case of linearly parametrized regression, $\Delta^{(j)}$ can be computed analytically as the following equation [47]:

$$\Delta^{(j)} = \frac{Y(\boldsymbol{X}^{(j)}) - \hat{Y}(\boldsymbol{X}^{(j)})}{1 - h_j} \qquad (j = 1, \ldots, N_s)$$

where $\hat{Y}(\boldsymbol{X}^{(j)})$ is the approximate model response at $\boldsymbol{X}^{(j)}$, which is built with all N_s sample points; and h_j is the jth diagonal term of the matrix $\boldsymbol{F}(\boldsymbol{F}^T\boldsymbol{F})^{-1}\boldsymbol{F}^T$

($\boldsymbol{F} = \left(\boldsymbol{\Phi}(\boldsymbol{X}^{(1)}), \ldots, \boldsymbol{\Phi}(\boldsymbol{X}^{(N_s)})\right)^T$). Thus, Err_{loo} can be rewritten as the following equation:

$$Err_{loo} = \frac{1}{N_s} \sum_{j=1}^{N_s} \left(\frac{Y(\boldsymbol{X}^{(j)}) - \hat{Y}(\boldsymbol{X}^{(j)})}{1 - h_j} \right)^2 \tag{2.18}$$

By using the leave-one-out cross validation technique, the steps to determine the optimal approximate model response $\hat{\boldsymbol{Y}}^{(i^*)}$ and coefficients $\boldsymbol{C}^{(i^*)}$ are listed as follows:

1. Select the predictors for the model responses $\hat{\boldsymbol{Y}}^{(i)}$ ($i = 1, \ldots, M_{\text{LARS}}$) by using the LARS algorithm.

2. Estimate the coefficients associated with the retained predictors $\boldsymbol{C}^{(i)} \in \mathbb{R}^i$ for each model response $\hat{\boldsymbol{Y}}^{(i)}$ ($i = 1, \ldots, M_{\text{LARS}}$) by using the ordinary least squares regression approach in Eq. (2.17).

3. Compute the leave-one-out error $Err_{loo}^{(i)}$ in Eq. (2.18) for each model response $\hat{\boldsymbol{Y}}^{(i)}$ ($i = 1, \ldots, M_{\text{LARS}}$).

4. Find the optimal model response $\hat{\boldsymbol{Y}}^{(i^*)}$ that has the smallest leave-one-out error $Err_{loo}^{(i^*)}$: $i^* = \arg\min_i Err_{loo}^{(i)}$.

5. Retain the optimal model response $\hat{\boldsymbol{Y}}^{(i^*)}$, coefficients $\boldsymbol{C}^{(i^*)}$ and corresponding predictors.

Using the optimal coefficients $\boldsymbol{C}^{(i^*)}$ and corresponding predictors, we can determine the approximate stochastic model response $\hat{Y}(\boldsymbol{X})$, which is equivalent to $Y_P(\boldsymbol{X})$ in Eq. (2.10). With the LARS algorithm, we can select a limited number of predictors that have the greatest influence on the model response. Thus, the LARS method is a computationally efficient approach to compute the sparse PCE approximation $\hat{Y}(\boldsymbol{X})$ especially for high-dimensional input random variables $\boldsymbol{X}$. By using the LARS algorithm and the leave-one-out cross validation criterion, we can determine the approximate stochastic model response $\hat{Y}(\boldsymbol{X})$.

2.2.3 Post-Processing

Once we compute the PCE coefficients $\boldsymbol{C}$, we can obtain the approximate stochastic model response $Y_P(\boldsymbol{X})$ in Eq. (2.10). There are several post-processing techniques using $\boldsymbol{C}$ and $Y_P(\boldsymbol{X})$ as follows. As described in Eqs. (2.10) and (2.14), $Y_P(\boldsymbol{X})$ is the distribution function of $\boldsymbol{X}$ and can be evaluated for any given random inputs. That is, $Y_P(\boldsymbol{X})$ can be used as the surrogate model, and an output $Y_P(\boldsymbol{X})$ can be obtained from a certain input $\boldsymbol{X}$ without actually executing the original simulation. By using the surrogate model of $Y_P(\boldsymbol{X})$, the probability density function of $Y_P(\boldsymbol{X})$ can be readily estimated, though the actual one is unknown. Since $Y_P(\boldsymbol{X})$ is given by the polynomials of random variables in Eq. (2.10), the probability distribution of $Y_P(\boldsymbol{X})$ can be computed by Monte Carlo simulation of the input random variables $\boldsymbol{X}$.

In addition, the statistical moments of $Y(\boldsymbol{X})$ can be derived analytically from $\boldsymbol{C}$ given by Eq. (2.14). For instance, the expected value and variance of $Y(\boldsymbol{X})$ are described as the following equations:

$$\begin{aligned} \mathrm{E}[Y(\boldsymbol{X})] &= \int \left[\sum_{m=1}^{\infty} C_m \Phi_m(\boldsymbol{X}) \right] \rho(\boldsymbol{X}) \mathrm{d}\boldsymbol{X} \\ &= C_1 \end{aligned} \tag{2.19}$$

$$\begin{aligned} \mathrm{V}[Y(\boldsymbol{X})] &= \mathrm{E}\left[\left(Y(\boldsymbol{X}) - \mathrm{E}[Y(\boldsymbol{X})] \right)^2 \right] \\ &= \int \left[\sum_{m=1}^{\infty} C_m \Phi_m(\boldsymbol{X}) - C_1 \right]^2 \rho(\boldsymbol{X}) \mathrm{d}\boldsymbol{X} \\ &\approx \int \left[\sum_{m=2}^{M} C_m \Phi_m(\boldsymbol{X}) \right]^2 \rho(\boldsymbol{X}) \mathrm{d}\boldsymbol{X} \\ &= \sum_{m=2}^{M} [C_m]^2 \end{aligned} \tag{2.20}$$

where $\mathrm{V}[\cdot]$ denotes the variance operator. It should be noted that $\Phi_1(\boldsymbol{X}) = 1$ because $\Phi_1(\boldsymbol{X})$ is set to the zero-order polynomial basis ($p_1 = 0$). The higher-order moments

such as the skewness and kurtosis as well as the expected value and variance can also be obtained by using $\boldsymbol{C}$ [86].

Moreover, the PCE method can apply to the global sensitivity analysis, which aims at quantifying the relative importance of each input random variable X_i ($i = 1, \ldots, N_X$) on the variance of the model response $Y(\boldsymbol{X})$ [85, 95, 96]. The global sensitivity indices called Sobol' sensitivity indices [97] can be computed analytically from the PCE coefficients $\boldsymbol{C}$ [85, 95, 96]. (See [85, 95, 96] for more detailed discussion of the global sensitivity analysis.)

2.3 Conflict Probability Estimation

To detect potential conflicts between aircraft, we need to compute the distance between each pair of aircraft. Since the wind errors in Eqs. (2.5) and (2.6) are described by random variables, the horizontal distance between aircraft L in Eq. (2.7) is also a random variable. By using the PCE method mentioned in Section 2.2, L can be solved and described as the orthonormal polynomials of the random variables representing the wind errors in Eqs. (2.5) and (2.6). As the post-processing of the PCE method in Eqs. (2.19) and (2.20), the statistical information of L ($\mathrm{E}[L]$ and $\mathrm{V}[L]$) can be computed.

The actual probability density function of L, $\rho(L)$, is unknown; however, the probability distribution of a random variable can be characterized by its moments, and the unknown distribution can be estimated by the moment matching technique. Using the statistical information of L, $\rho(L)$ is approximated by the univariate Gaussian distribution by matching the first two moments: $L \sim \mathcal{N}(\mathrm{E}[L], \mathrm{V}[L])$, where $\mathcal{N}(\mu, \sigma^2)$ denotes the Gaussian distribution with mean μ and variance σ^2. We demonstrate that $\rho(L)$ can be approximated accurately as the Gaussian distribution by the moment matching technique in Section 5.1. Accordingly, on the basis of the safety constraint in Eq. (2.7), the conflict probability between the ith and jth aircraft $\Pr[\mathcal{C}_{ij}]$ can be

given by the following equation:

$$\Pr\left[\mathcal{C}_{ij}\right] = \Pr\left[L_{ij} \leq d_{H\min}\right] = 1 - \int_{d_{H\min}}^{\infty} \rho(L_{ij}) \mathrm{d}L_{ij} \tag{2.21}$$

where $\Pr[\cdot]$ denotes the probability of an event; and $\mathcal{C}_{ij}$ indicates the conflict between the ith and jth aircraft. Since $\rho(L_{ij})$ is approximated as the Gaussian distribution, $\Pr\left[\mathcal{C}_{ij}\right]$ in Eq. (2.21) can be easily computed. It should be noted that the conflict probability between aircraft and the convective weather region can also be estimated by Eq. (2.21).

The steps to estimate the conflict probability between the ith and jth aircraft are described as follows.

1. Compute the stochastic solution of the distance between aircraft L_{ij} by using the PCE method described in Section 2.2.

2. Calculate the statistical information of L_{ij} ($\mathrm{E}(L_{ij})$ and $\mathrm{V}(L_{ij})$) as the post-processing of the PCE method in Eqs. (2.19) and (2.20).

3. Approximate the probability density function of L_{ij}, $\rho(L_{ij})$, as the Gaussian distribution by the moment matching technique.

4. Estimate the conflict probability between aircraft $\Pr\left[\mathcal{C}_{ij}\right]$ in Eq. (2.21).

Using the probabilistic conflict detection algorithm mentioned above, the potential conflicts can be detected. To resolve the potential conflicts, the stochastic optimal control method is proposed to determine the conflict resolution trajectory in the presence of uncertainty in Chapter 3.

Chapter 3

Stochastic Optimal Control for Conflict Resolution

In this chapter, a stochastic optimal control method incorporating the PCE algorithm into the pseudospectral method is developed to solve the conflict resolution problem in the presence of uncertainty.

3.1 Deterministic Optimal Control

Optimal control problems especially in aeronautics and astronautics are most often solved numerically due to the complexity of the problems. Numerical methods for solving continuous-time optimal control problems have been well developed closely paralleled by advancements in aerospace technologies and powerful computational tools. In general, numerical methods to solve continuous-time optimal control problems can be categorized into two classes: indirect and direct methods [98, 99]. In an indirect method, the first-order optimality conditions are derived from the optimal control problem using the calculus of variations and Pontryagin's maximum (or minimum) principle [100]. The optimality conditions form the Hamiltonian boundary value problem (HBVP), which is solved to find the optimal solution [101, 102]. The indirect methods can provide a highly accurate solution and the assurance that the solution satisfies the first-order optimality conditions. However, the indirect meth-

ods require a good initial guess for the costate, a priori knowledge of the activeness of inequality constraints, and the analytical derivation of the HBVP. On the other hand, in a direct method, the continuous-time optimal control problem is transcribed into the nonlinear programming (NLP) problem by discretization. Then, the NLP problem can be solved numerically to satisfy the NLP optimality conditions, i.e., Karush-Kuhn-Tucker (KKT) conditions. Direct methods have the good convergence and robustness properties compared to indirect methods.

There are several direct methods to transcribe the optimal control problem into the NLP problem: direct shooting methods [98,99], direct collocation methods [98,99] and pseudospectral methods [103–106]. In a direct shooting method, only the control variables are discretized and parametrized, and an explicit numerical integration method is applied to satisfy the differential constraints. In a direct collocation method, both the state and control variables are discretized and parametrized, and the differential equations are approximated using the piecewise polynomials at specific points called collocation points. Although the scale of the NLP problem in direct collocation methods becomes large compared to direct shooting methods, the direct collocation methods have an advantage that they can avoid the numerically intensive explicit numerical integration in a direct shooting method.

In a pseudospectral method, which has increased in popularity for solving nonlinear optimal control problems, the state and control variables are discretized and parametrized using global polynomials, and the differential equations are approximated using orthogonal polynomials. Pseudospectral methods have advantages of an exponential convergence rate [107] and a highly accurate costate mapping from KKT multipliers of the NLP problem due to the equivalence between the KKT conditions and the HBVP optimality conditions [48, 49, 108, 109]. Therefore, in this study, we employ the pseudospectral method for solving continuous-time nonlinear optimal control problems.

We consider the following continuous-time deterministic optimal control problem: determine the state variables $\boldsymbol{x}(t) \in \mathbb{R}^{N_x}$, the control variables $\boldsymbol{u}(t) \in \mathbb{R}^{N_u}$, the initial time $t_0 \in \mathbb{R}$ and the terminal time $t_f \in \mathbb{R}$ on the time interval $t \in [t_0, t_f]$ that minimize

the cost functional:

$$J = g_M(\boldsymbol{x}(t_0), t_0, \boldsymbol{x}(t_f), t_f) + \int_{t_0}^{t_f} g_L(\boldsymbol{x}(t), \boldsymbol{u}(t), t)\mathrm{d}t \tag{3.1}$$

subject to the dynamic constraints:

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \boldsymbol{f}_S(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \tag{3.2}$$

the boundary conditions:

$$\boldsymbol{b}_{\min} \leq \boldsymbol{b}(\boldsymbol{x}(t_0), t_0, \boldsymbol{x}(t_f), t_f) \leq \boldsymbol{b}_{\max} \tag{3.3}$$

the path constraints:

$$\boldsymbol{c}_{\min} \leq \boldsymbol{c}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \leq \boldsymbol{c}_{\max} \tag{3.4}$$

where $g_M \in \mathbb{R}$ and $g_L \in \mathbb{R}$ define the Mayer and Lagrange terms in the cost function, respectively; $\boldsymbol{f}_S \in \mathbb{R}^{N_x}$ is the system dynamics; $\boldsymbol{b} \in \mathbb{R}^{N_b}$ expresses the boundary condition functions; $\boldsymbol{c} \in \mathbb{R}^{N_c}$ defines the path constraint functions; N_x is the number of state variables; N_u is the number of control variables; N_b is the number of boundary conditions; and N_c is the number of path constraints. We introduce a new time interval $\tau \in [-1, 1]$, which can be transformed to $t \in [t_0, t_f]$ via the affine transformation:

$$t = \frac{t_f - t_0}{2}\tau + \frac{t_f + t_0}{2}$$

The continuous-time optimal control problem of Eqs. (3.1)–(3.4) is then modified in terms of τ as follows: determine the state variables $\boldsymbol{x}(\tau) \in \mathbb{R}^{N_x}$, the control variables $\boldsymbol{u}(\tau) \in \mathbb{R}^{N_u}$, the initial time t_0 and the terminal time t_f on the new time interval $\tau \in [-1, 1]$ that minimize the cost functional:

$$J = g_M(\boldsymbol{x}(-1), t_0, \boldsymbol{x}(1), t_f) + \frac{t_f - t_0}{2} \int_{-1}^{1} g_L(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \tau; t_0, t_f)\mathrm{d}\tau \tag{3.5}$$

subject to the dynamic constraints:

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}\tau} = \frac{t_f - t_0}{2}\boldsymbol{f}_S(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \tau; t_0, t_f) \tag{3.6}$$

the boundary conditions:

$$\boldsymbol{b}_{\min} \leq \boldsymbol{b}(\boldsymbol{x}(-1), t_0, \boldsymbol{x}(1), t_f) \leq \boldsymbol{b}_{\max} \tag{3.7}$$

the path constraints:

$$\boldsymbol{c}_{\min} \leq \boldsymbol{c}(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \tau; t_0, t_f) \leq \boldsymbol{c}_{\max} \tag{3.8}$$

In the pseudospectral method[1], the continuous optimal control problem (Eqs. (3.5)–(3.8)) is discretized and transcribed into the NLP problem. We firstly select the N_p collocation points $\tau_i \in [-1, 1]$ $(i = 1, \ldots, N_p)$. Note that the end points are not included in the set of the collocation points. The states $\boldsymbol{x}(\tau)$ and controls $\boldsymbol{u}(\tau)$ are approximated and parametrized using $N_p + 1$ Lagrange basis polynomials:

$$\begin{aligned} \boldsymbol{x}(\tau) &\approx \boldsymbol{X}(\tau) = \sum_{i=0}^{N_p} \boldsymbol{X}(\tau_i) L_{x,i}(\tau) \\ \boldsymbol{u}(\tau) &\approx \boldsymbol{U}(\tau) = \sum_{i=1}^{N_p} \boldsymbol{U}(\tau_i) L_{u,i}(\tau) \end{aligned} \tag{3.9}$$

where $\boldsymbol{X}(\tau)$ and $\boldsymbol{U}(\tau)$ denote the polynomial approximations; and $L_{x,i}(\tau)$ and $L_{u,i}(\tau)$ represent the Lagrange interpolating polynomials for the states and controls, respectively. $L_{x,i}(\tau)$ and $L_{u,i}(\tau)$ are defined as the following equations:

$$\begin{aligned} L_{x,i}(\tau) &= \prod_{j=0, j\neq i}^{N_p} \frac{\tau - \tau_j}{\tau_i - \tau_j} \qquad (i = 0, \ldots, N_p) \\ L_{u,i}(\tau) &= \prod_{j=1, j\neq i}^{N_p} \frac{\tau - \tau_j}{\tau_i - \tau_j} \qquad (i = 1, \ldots, N_p) \end{aligned}$$

[1]Note that as an example, we discuss the Gauss pseudospectral method [48, 108, 109] in this section.

It should be noted that $L_{x,i}(\tau_j)$ and $L_{u,i}(\tau_j)$ take 1 if $i = j$ and 0 otherwise. Differentiating Eq. (3.9) with respect to τ, the approximation of Eq. (3.6) can be obtained as the following equation:

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\boldsymbol{x}(\tau) \approx \frac{\mathrm{d}}{\mathrm{d}\tau}\boldsymbol{X}(\tau) = \sum_{i=0}^{N_p} \boldsymbol{X}(\tau_i)\frac{\mathrm{d}}{\mathrm{d}\tau}L_{x,i}(\tau)$$

The derivatives of the Lagrange polynomials at the collocation points can be represented in a differentiation matrix $\boldsymbol{D} \in \mathbb{R}^{N_p \times (N_p+1)}$ with the (k,i)th component D_{ki}:

$$D_{ki} = \frac{\mathrm{d}}{\mathrm{d}\tau}L_{x,i}(\tau_k) = \sum_{l=0}^{N_p} \frac{\prod_{j=0,j\neq i,l}^{N_p}(\tau_k - \tau_j)}{\prod_{j=0,j\neq i}^{N_p}(\tau_i - \tau_j)} \qquad (k = 1,\ldots,N_p,\ i = 0,\ldots,N_p)$$

The dynamic constraints in Eq. (3.6) can be discretized and transcribed into the algebraic constraints using the differentiation matrix $\boldsymbol{D}$:

$$\sum_{i=0}^{N_p} D_{ki}\boldsymbol{X}_i - \frac{t_f - t_0}{2}\boldsymbol{f}_S(\boldsymbol{X}_k, \boldsymbol{U}_k, \tau_k; t_0, t_f) = \boldsymbol{0} \qquad (k = 1,\ldots.N_p) \tag{3.10}$$

where $\boldsymbol{X}_k = \boldsymbol{X}(\tau_k)$ and $\boldsymbol{U}_k = \boldsymbol{U}(\tau_k)$ $(k = 1,\ldots,N_p)$. Note that $\boldsymbol{X}(\tau)$ and $\boldsymbol{U}(\tau)$ are not collocated at the end points, and the terminal state $\boldsymbol{X}_f = \boldsymbol{X}(1)$ is defined via the Gaussian quadrature:

$$\boldsymbol{X}_f = \boldsymbol{X}_0 + \frac{t_f - t_0}{2}\sum_{k=1}^{N_p} w_k \boldsymbol{f}_S(\boldsymbol{X}_k, \boldsymbol{U}_k, \tau_k; t_0, t_f) \tag{3.11}$$

where $\boldsymbol{X}_0 = \boldsymbol{X}(-1)$ is the initial states; and w_k is the Gaussian quadrature weight $(k = 1,\ldots,N_p)$. The cost functional in Eq. (3.5) is also discretized and approximated using the Gaussian quadrature:

$$J = g_M(\boldsymbol{X}_0, t_0, \boldsymbol{X}_f, t_f) + \frac{t_f - t_0}{2}\sum_{k=1}^{N_p} w_k g_L(\boldsymbol{X}_k, \boldsymbol{U}_k, \tau_k; t_0, t_f) \tag{3.12}$$

Furthermore, the discretized form of the boundary conditions in Eq. (3.7) is given by

the following equation:

$$\boldsymbol{b}_{\min} \leq \boldsymbol{b}(\boldsymbol{X}_0, t_0, \boldsymbol{X}_f, t_f) \leq \boldsymbol{b}_{\max} \tag{3.13}$$

The path constraints in Eq. (3.8) can be discretized as the following equation:

$$\boldsymbol{c}_{\min} \leq \boldsymbol{c}(\boldsymbol{X}_k, \boldsymbol{U}_k, \tau_k; t_0, t_f) \leq \boldsymbol{c}_{\max} \qquad (k = 1, \ldots . N_p) \tag{3.14}$$

Thus, the continuous-time optimal control problem (Eqs. (3.5)–(3.8)) can be discretized and transcribed into the nonlinear programming (NLP) problem, which is defined by the discretized cost functional in Eq. (3.12) and the algebraic constraints in Eqs. (3.10), (3.11), (3.13) and (3.14). Then, an NLP solver such as sequential quadratic programming (SQP) is applied to compute the optimal solution. It should be noted that we can deal with the multiple-phase optimal control problem by dividing the problem into the multiple phases, where the dynamics are discretized within each phase and then connected to each other by the phase linkage constraints.

There are several software packages that have been developed to solve the continuous-time optimal control problems: DIDO implements the Legendre pseudospectral method [110]; and the General Pseudospectral Optimization Software (GPOPS) [111] implements the Radau pseudospectral method[2] [49, 50, 114, 115]. In this study, we employ GPOPS [111], which is performed in MATLAB and using SNOPT [116, 117] as the NLP solver. By using GPOPS, the continuous-time optimal control problem is transformed into the NLP problem for the SNOPT NLP solver which finds the optimal solution.

3.2 Stochastic Optimal Control

The following continuous-time stochastic optimal control problem is considered: determine the state variables $\boldsymbol{x}(t)$, the control variables $\boldsymbol{u}(t)$, the initial time t_0 and the

[2]The software originally implemented the Gauss pseudospectral method [48, 108, 109, 112], and GPOPS-II [113] is recently developed.

terminal time t_f on the time interval $t \in [t_0, t_f]$ that minimize the cost functional:

$$J = \mathrm{E}\left[g_M(\boldsymbol{x}(t_0), t_0, \boldsymbol{x}(t_f), t_f, \boldsymbol{X}) + \int_{t_0}^{t_f} g_L(\boldsymbol{x}(t), \boldsymbol{u}(t), t, \boldsymbol{X})\mathrm{d}t\right] \tag{3.15}$$

subject to the dynamic constraints:

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \boldsymbol{f}_S(\boldsymbol{x}(t), \boldsymbol{u}(t), t, \boldsymbol{X}) \tag{3.16}$$

the boundary conditions:

$$\boldsymbol{b}_{\min} \leq \mathrm{E}\left[\boldsymbol{b}(\boldsymbol{x}(t_0), t_0, \boldsymbol{x}(t_f), t_f, \boldsymbol{X})\right] \leq \boldsymbol{b}_{\max} \tag{3.17}$$

and the chance constraints [118–120]:

$$\boldsymbol{\eta}_{\min} \leq \Pr\left[\boldsymbol{c}_{\min} \leq \boldsymbol{c}(\boldsymbol{x}(t), \boldsymbol{u}(t), t, \boldsymbol{X}) \leq \boldsymbol{c}_{\max}\right] \leq \boldsymbol{\eta}_{\max} \tag{3.18}$$

where $\boldsymbol{X}$ is the random variables; and $\boldsymbol{\eta} \in \mathbb{R}^{N_c}$ is the confidence level. In this study, the conflict probability is formulated as the chance constraint in Eq. (3.18), which is proposed in the previous studies [118–120]. It should be noted that the expected values of the cost functional and boundary conditions in Eqs. (3.15) and (3.17) are considered for the stochastic optimal control problem.

To deal with the stochastic elements and solve the stochastic optimal control problem, the PCE method is applied in the previous works [121–125], and the stochastic solution including the statistical information is approximated by the theory of the PCE method. In this study, we also incorporate the PCE algorithm into the pseudospectral method to deal with the stochastic elements. Within the framework of the pseudospectral method described in Section 3.1, the state and control variables are approximated and parametrized using Lagrange basis polynomials, and the cost functional and the constraints are discretized using orthogonal polynomials based on a quadrature rule. Thus, the continuous-time optimal control problem can be discretized and transcribed into the NLP problem, and the optimal solution can be

computed by using the NLP solver. Therefore, by incorporating the PCE algorithm into the pseudospectral method, we can deal with the stochastic elements and solve the stochastic optimal control problem. As mentioned in Section 3.1, we employ GPOPS [111] with the NLP solver SNOPT [116, 117]. Unlike the previous studies [121–125], we consider the chance constrained stochastic optimal control problem of Eqs. (3.15)–(3.18) for conflict resolution. To solve the conflict resolution problem, the stochastic optimal control method is combined with the proposed probabilistic conflict detection algorithm to guarantee the resolution of potential conflicts in the presence of uncertainty. By solving the stochastic optimal control problem for conflict resolution, the optimal conflict resolution trajectory in the presence of uncertainty.

Chapter 4

Stochastic Near-Optimal Control for Conflict Resolution

By using the stochastic optimal control method mentioned in Section 3.2, we can generate the optimal conflict resolution trajectory starting from a given initial condition. However, the states on the precomputed optimal trajectory may differ from the actual states due to the uncertainties during the flight. In this case, another optimal control problem starting from the actual states is necessary to be solved to obtain the correct optimal trajectory. In this section, we propose a near-optimal control method for generating conflict resolution trajectories in real time by constructing surrogate models based on the recently developed polynomial chaos kriging method. Polynomial chaos kriging is a hybrid algorithm based on two surrogate modeling techniques: PCE and kriging. We first introduce kriging, and then present the hybrid polynomial kriging method. After that, we propose the near-optimal conflict resolution algorithm based on polynomial chaos kriging. Constructing the surrogate models of the optimal conflict resolution trajectories from a set of precomputed optimal solutions, the approximate optimal conflict resolution trajectories can be obtained in real time based on the information of the current conditions without actually solving the computationally expensive stochastic optimal control problems.

4.1 Kriging

A surrogate modeling technique has been widely used in engineering applications, especially design optimization problems [54, 126–129]. A surrogate model can approximate the input-output behavior of an original simulation, and an output can be obtained from a certain input in real time without actually executing the original simulation. There are various surrogate modeling methods[1] such as polynomial response surface models [130], radial basis functions [131], multivariate adaptive regression splines [132], support vector regression [133], moving least squares [134–136] and kriging (also known as Gaussian process regression) [54, 55, 137, 138]. Among these surrogate modeling techniques, the kriging approach[2] is a stochastic interpolation method, which can provide exactly the same output values at the observed sample points as the observations. In addition, the kriging model provides the best linear unbiased prediction, i.e., the expected value of the error is zero and the variance of the error is minimized.

The kriging model is assumed that the model response is the realization of the Gaussian process with a priori covariance matrix, and the mathematical form of the kriging model is expressed as the following equation:

$$Y(\boldsymbol{X}) = \mu(\boldsymbol{X}) + Z(\boldsymbol{X}) \tag{4.1}$$

where $\boldsymbol{X} = (X_1, \ldots, X_{N_X})^T \in \mathbb{R}^{N_X}$ is the vector of the input variables; $Y(\boldsymbol{X}) \in \mathbb{R}$ is the model response; $\mu(\boldsymbol{X}) \in \mathbb{R}$ is the regression function, also known as the trend function; $Z(\boldsymbol{X}) \in \mathbb{R}$ is the realization of the Gaussian process; and N_X is the number of the input variables. $\mu(\boldsymbol{X})$ provides a global trend of the design space, and $Z(\boldsymbol{X})$ creates local deviations or residuals so that the kriging model interpolates the sampled observations by quantifying the correlation of nearby sample points. In classical kriging methods, the ordinary kriging method assumes that the trend function $\mu(\boldsymbol{X})$

[1] Surrogate models can also be classified into three classes: parametric, non-parametric and semi-parametric models.

[2] Kriging, named after a South African mining engineer Daniel G. Krige, was originally developed in geostatistics for predicting mineral resources based on sampled sites in the 1950s.

is an unknown constant value to be estimated, and the universal kriging approach assumes that $\mu(\boldsymbol{X})$ is a linear combination of polynomials. Since ordinary kriging is the specific case of universal kriging, we discuss the universal kriging model in this study. In universal kriging, $\mu(\boldsymbol{X})$ is described as the following equation:

$$\mu(\boldsymbol{X}) = \sum_{m=1}^{M} \beta_m f_m(\boldsymbol{X}) = \boldsymbol{f}^T(\boldsymbol{X})\boldsymbol{\beta} \tag{4.2}$$

where $\boldsymbol{f}(\boldsymbol{X}) = (f_1(\boldsymbol{X}), \ldots, f_M(\boldsymbol{X}))^T \in \mathbb{R}^M$ is the vector of trend basis functions $f_m(\boldsymbol{X})$ $(m = 1, \ldots, M)$; $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_M)^T \in \mathbb{R}^M$ is the vector of unknown regression coefficients β_m $(m = 1, \ldots, M)$ to be estimated; and M is the number of the trend basis functions. In order to determine the kriging model in Eqs. (4.1) and (4.2), we consider N_s sample points of the input variables $\boldsymbol{X}^{(j)}$ and corresponding model responses (observations) $Y(\boldsymbol{X}^{(j)})$ $(j = 1, \ldots, N_s)$. $Z(\boldsymbol{X})$ in Eq. (4.1) is the Gaussian process with zero mean and the following covariance matrix:

$$\mathrm{Cov}[\boldsymbol{X}^{(i)}, \boldsymbol{X}^{(j)}] = \sigma^2 R_{ij} = \sigma^2 R(\boldsymbol{X}^{(i)}, \boldsymbol{X}^{(j)}) \qquad (\forall i, j = \{1, \ldots, N_s\}) \tag{4.3}$$

where $\mathrm{Cov}[\cdot]$ denotes the covariance operator; σ^2 is the process variance; $\boldsymbol{X}^{(i)}$ and $\boldsymbol{X}^{(j)}$ are the ith and jth sample points, respectively; and $R_{ij} = R(\boldsymbol{X}^{(i)}, \boldsymbol{X}^{(j)})$ is the spatial correlation function between $\boldsymbol{X}^{(i)}$ and $\boldsymbol{X}^{(j)}$. There are various formulations of correlation functions, and some of the commonly used correlation functions [53, 137] are listed as follows:

the Dirac's delta function:

$$R_{ij} = \prod_{l=1}^{N_X} \delta(X_l^{(i)} - X_l^{(j)})$$

the linear function:

$$R_{ij} = \prod_{l=1}^{N_X} \max\left(0, 1 - \theta_l |X_l^{(i)} - X_l^{(j)}|\right)$$

the exponential function[3]:

$$R_{ij} = \exp\left(-\sum_{l=1}^{N_X} \theta_l |X_l^{(i)} - X_l^{(j)}|^{\lambda_l} \right)$$

the Gaussian function:

$$R_{ij} = \exp\left(-\sum_{l=1}^{N_X} \theta_l |X_l^{(i)} - X_l^{(j)}|^2 \right) \tag{4.4}$$

and the Matérn function:

$$R_{ij} = \prod_{l=1}^{N_X} \frac{1}{2^{\nu-1}\Gamma(\nu)} \left(\sqrt{2\nu}\theta_l |X_l^{(i)} - X_l^{(j)}| \right)^{\nu} \kappa_\nu \left(\sqrt{2\nu}\theta_l |X_l^{(i)} - X_l^{(j)}| \right)$$

where $\delta(\cdot)$ is the Dirac's delta function that takes 1 if $X_l^{(i)} = X_l^{(j)}$ and 0 otherwise; ν is the shape parameter for the Matérn function ($\nu \geq 1/2$, $\nu = 3/2$ and $\nu = 5/2$ are the most popular forms); $\Gamma(\cdot)$ is the Gamma function; $\kappa_\nu(\cdot)$ is the modified Bessel function of the second kind, also known as the Bessel function of the third kind and the Hankel function; $X_l^{(i)}$ is the lth input variable of the ith sample point $\boldsymbol{X}^{(i)} = (X_1^{(i)}, \ldots, X_{N_X}^{(i)})^T$; and $\theta_l \in \mathbb{R}$ and $\lambda_l \in \mathbb{R}$ ($l = 1, \ldots, N_X$) are the unknown correlation parameters to be estimated, also known as the hyperparameter ($\theta_l > 0$ and typically $1 \leq \lambda_l \leq 2$). It should be noted that the accuracy of the kriging model highly depends on the choice of the correlation functions because the influence of the observed values is determined by the correlation functions. In a wide range of physical applications, the correlation function R_{ij} is set to be the Gaussian function in Eq. (4.4), which is employed in this study.

We introduce the matrix of the basis functions $\boldsymbol{F} = (\boldsymbol{f}(\boldsymbol{X}^{(1)}), \ldots, \boldsymbol{f}(\boldsymbol{X}^{(N_s)}))^T$, the vector of the model responses at the sample points $\boldsymbol{Y} = (Y(\boldsymbol{X}^{(1)}), \ldots, Y(\boldsymbol{X}^{(N_s)}))^T$, the correlation matrix $\boldsymbol{R}$[4] with the (i,j)th component $R_{ij} = R(\boldsymbol{X}^{(i)}, \boldsymbol{X}^{(j)})$ and the correlation vector $\boldsymbol{r}(\boldsymbol{X}) = (R(\boldsymbol{X}, \boldsymbol{X}^{(1)}), \ldots, R(\boldsymbol{X}, \boldsymbol{X}^{(N_s)}))^T$ to express the correlation between a new input vector $\boldsymbol{X}$ and each sample point $\boldsymbol{X}^{(j)}$ ($j = 1, \ldots, N_s$). The

[3]Note that the exponential function is equivalent to the Gaussian function when $\lambda_l = 2$.

[4]$\boldsymbol{R}$ is the symmetric matrix ($R_{ij} = R_{ji}$) with ones on the diagonal because of $R_{ii} = 1$.

predicted response of the kriging model $\hat{Y}(\boldsymbol{X})$ for a new input vector $\boldsymbol{X}$ can be given by the following equation:

$$\hat{Y}(\boldsymbol{X}) = \boldsymbol{f}^T(\boldsymbol{X})\hat{\boldsymbol{\beta}} + \boldsymbol{r}^T(\boldsymbol{X})\boldsymbol{R}^{-1}(\boldsymbol{Y} - \boldsymbol{F}\hat{\boldsymbol{\beta}}) \tag{4.5}$$

where $\hat{\boldsymbol{\beta}}$ is the least squares estimate of $\boldsymbol{\beta}$ and can be estimated by the following equation:

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{F}^T\boldsymbol{R}^{-1}\boldsymbol{F})^{-1}\boldsymbol{F}^T\boldsymbol{R}^{-1}\boldsymbol{Y} \tag{4.6}$$

The process variance σ^2 in Eq. (4.3) can be calculated by the following equation:

$$\hat{\sigma}^2 = \frac{(\boldsymbol{Y} - \boldsymbol{F}\hat{\boldsymbol{\beta}})^T\boldsymbol{R}^{-1}(\boldsymbol{Y} - \boldsymbol{F}\hat{\boldsymbol{\beta}})}{N_s}$$

where $\hat{\sigma}^2$ is the estimation of σ^2.

The hyperparameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_{N_X})^T \in \mathbb{R}^{N_X}$ in Eq. (4.4) are needed to be estimated for computing the krging model $\hat{Y}(\boldsymbol{X})$ in Eqs. (4.5) and (4.6). $\boldsymbol{\theta}$ can be obtained by either of the following two methods: maximum likelihood estimation and cross validation [139, 140]. In the maximum likelihood estimation, one of the core assumptions is that the observations are derived from a Gaussian process. In this study, $\boldsymbol{\theta}$ is estimated by using the maximum likelihood estimation and solving the following nonlinear optimization problem:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}\in\mathbb{R}^{N_X}}{\arg\max} -\frac{N_s\ln(\hat{\sigma}^2) + \ln(\det(\boldsymbol{R}))}{2} \tag{4.7}$$

where $\hat{\boldsymbol{\theta}}$ is the maximum likelihood estimate of $\boldsymbol{\theta}$; and $\det(\cdot)$ is the determinant operator of a matrix. In Eq. (4.7), both $\hat{\sigma}^2$ and $\det(\boldsymbol{R})$ are the functions of $\boldsymbol{\theta}$. The optimal kriging model can be determined by solving the N_X-dimensional unconstrained nonlinear optimization problem in Eq. (4.7). The optimization problem in Eq. (4.7) is highly nonlinear and potentially multimodal, i.e., the solution space contains multiple local maxima. Several methods for solving the optimization problem in Eq. (4.7), e.g., the Hooke–Jeeves method [141], a genetic algorithm [128, 138] and the Broyden–

Fletcher–Goldfarb–Shanno (BFGS) algorithm [142], have been proposed. Moreover, there are several software packages that have been developed to solve the optimization problem in Eq. (4.7) and determine the kriging model: DACE [141] is performed in MATLAB; and DiceKriging [142] is written in R. In this study, we use a hybrid optimization approach that combines the genetic algorithm with the SQP method. The genetic algorithm can be used to find the optimal solution globally and the SQP method search for the optimal solution locally. (See Refs. [54,55,137] for more detailed derivation and discussion of kriging.)

4.2 Polynomial Chaos Kriging

Polynomial chaos kriging is a recently developed hybrid algorithm based on two surrogate modeling techniques: PCE and kriging [53]. PCE can provide a global trend of the model response, and kriging (especially a Gaussian process term) can create local deviations to interpolate the sampled observations by quantifying the correlation of nearby sample points. Combining these two methods, the hybrid surrogate modeling technique called polynomial chaos kriging has an advantage of providing both the global behavior of PCE and the local behavior of kriging. In the previous work [53], the performance of the hybrid polynomial chaos kriging algorithm is demonstrated using analytical benchmark functions, and polynomial chaos kriging generally performs better than PCE or kriging does. In this study, we apply the polynomial chaos kriging method to the practical engineering problem, i.e., the optimal control problem for conflict resolution.

On the basis of the mathematical forms of the PCE model in Eq. (2.10) and the kriging model in Eqs. (4.1) and (4.2), the hybrid polynomial chaos kriging model can be described as the following form:

$$Y(\boldsymbol{X}) = \sum_{m=1}^{M} C_m \Phi_m(\boldsymbol{X}) + Z(\boldsymbol{X}) = \boldsymbol{\Phi}^T(\boldsymbol{X})\boldsymbol{C} + Z(\boldsymbol{X}) \tag{4.8}$$

where $Y(\boldsymbol{X}) \in \mathbb{R}$ is the model response; $\boldsymbol{X} = (X_1, \ldots, X_{N_X})^T \in \mathbb{R}^{N_X}$ is the vector of

the input variables; $\boldsymbol{C} = (C_1, \ldots, C_M)^T \in \mathbb{R}^M$ is the vector of the PCE coefficients C_m $(m = 1, \ldots, M)$; $\boldsymbol{\Phi}(\boldsymbol{X}) = (\Phi_1(\boldsymbol{X}), \ldots, \Phi_M(\boldsymbol{X}))^T \in \mathbb{R}^M$ is the vector of the multivariate orthonormal polynomial basis functions $\Phi_m(\boldsymbol{X})$ $(m = 1, \ldots, M)$; $Z(\boldsymbol{X})$ is the Gaussian process with zero mean and the covariance matrix given by Eqs. (4.3) and (4.4); M is the number of the PCE coefficients or polynomials; and N_X is the number of the input variables. As mentioned in Section 4.1, $Z(\boldsymbol{X})$ is characterized by the unknown hyperparameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_{N_X})^T \in \mathbb{R}^{N_X}$ in Eq. (4.4).

As described in Eq. (4.8), the polynomial chaos kriging model uses the PCE terms in Eq. (2.10) as the trend function $\mu(\boldsymbol{X})$ of the kriging model in Eq. (4.1). Therefore, $\boldsymbol{\Phi}(\boldsymbol{X})$ and M can be determined within the PCE framework, and $\boldsymbol{C}$ and $\boldsymbol{\theta}$ can be computed within the kriging framework. Since the stochastic collocation approach of the PCE method is computationally efficient when the number of random variables or input variables is small, we employ the stochastic collocation method for dealing with the wind errors. However, as the number of input variables increases, the number of collocation points is considerably larger even though the sparse grid is used. We consider a large number of input variables for the surrogate models (more than 10 variables), which becomes computationally laborious and intractable with the stochastic collocation method. Therefore, to reduce the computational cost for constructing the surrogate models, we employ the PCE method based on the LARS algorithm. With the LARS method, the sparse PCE terms can be provided. As mentioned in Section 2.2.2, the LARS method is a computationally efficient model selection algorithm for high-dimensional data.

The hybrid polynomial chaos kriging algorithm consists of two steps: in the first step, the set of polynomials $\boldsymbol{\Phi}(\boldsymbol{X})$ and the number of polynomials M can be determined within the PCE framework, i.e. the LARS algorithm; and in the second step, the hyperparameters $\boldsymbol{\theta}$ can be computed by using the maximum likelihood estimation in Eq. (4.7) and the PCE coefficients $\boldsymbol{C}$ can be estimated by Eq. (4.6) within the kriging framework. Although $\boldsymbol{C}$ as well as $\boldsymbol{\Phi}(\boldsymbol{X})$ can be estimated within the PCE framework, $\boldsymbol{C}$ is computed and given by Eq. (4.6) within the kriging framework. The LARS algorithm selects the optimal set of the polynomial basis functions $\boldsymbol{\Phi}(\boldsymbol{X})$.

The procedures to construct the polynomial chaos kriging model $\hat{Y}(\boldsymbol{X})$ are listed as follows:

1. Generate N_s sample points of input variables $\boldsymbol{X}^{(j)}$ and corresponding model responses $Y(\boldsymbol{X}^{(j)})$ $(j = 1, \ldots, N_s)$.

2. Determine the optimal set of the polynomial basis functions $\boldsymbol{\Phi}(\boldsymbol{X})$ and the number of the polynomials M with the PCE method based on the LARS algorithm.

3. Compute the hyperparameters $\boldsymbol{\theta}$ by using the maximum likelihood estimation in Eq. (4.7).

4. Estimate the PCE coefficients $\boldsymbol{C}$ by Eq. (4.6).

5. Determine the polynomial chaos kriging model $\hat{Y}(\boldsymbol{X})$ by Eq. (4.5), which is equivalent to the approximation of Eq. (4.8).

4.3 Stochastic Near-Optimal Control

The surrogate models of the optimal conflict resolution trajectories are constructed based on the polynomial chaos kriging method for generating near-optimal conflict resolution trajectories in real time. The discrete time steps, $t_0 = T_1 < \cdots < T_{N_t+1} = t_f$, are considered, and the surrogate models are built at each time step. The inputs of the surrogate models $\boldsymbol{X}(T_k)$ are the current condition (e.g., current states and parameters for a convective weather region) at the current time step T_k $(k = 1, \ldots, N_t)$. The outputs are the optimal states $\boldsymbol{x}^*_{k+1}(\boldsymbol{X}(T_k))$ at the next time step T_{k+1}, optimal controls $\boldsymbol{u}^*_k(\boldsymbol{X}(T_k))$ at the current time step and optimal terminal times $\boldsymbol{t}^*_{f,k}(\boldsymbol{X}(T_k))$ at the current time step. We construct these surrogate models of the optimal conflict resolution trajectories by using a set of precomputed N_s optimal solutions, which are obtained by solving N_s stochastic optimal control problems starting from the different initial conditions $\boldsymbol{X}^{(j)}(t_0) = \boldsymbol{X}^{(j)}(T_1)$ $(j = 1, \ldots, N_s)$. Once the surrogate models are constructed using the polynomial chaos kriging method, the approximate

optimal conflict resolution trajectories on the time interval between the current time and the terminal time can be obtained as the functions of the current conditions. Thus, the near-optimal solutions can be estimated in real time by using the surrogate models with the information of the current conditions without actually solving the computationally expensive stochastic optimal control problems.

The procedures to generate the near-optimal conflict resolution trajectory by constructing the surrogate models are listed as follows:

1. Generate N_s sample points of the initial condition $\boldsymbol{X}^{(j)}(t_0) = \boldsymbol{X}^{(j)}(T_1)$ $(j = 1, \ldots, N_s)$.

2. Solve N_s stochastic optimal control problems starting from the different initial conditions $\boldsymbol{X}^{(j)}(T_1)$ $(j = 1, \ldots, N_s)$ by using the stochastic optimal control method mentioned in Section 3.2.

3. Determine the optimal states $\boldsymbol{x}^*_{k+1}(\boldsymbol{X}^{(j)}(T_k^{(j)}))$, optimal controls $\boldsymbol{u}^*_k(\boldsymbol{X}^{(j)}(T_k^{(j)}))$ and optimal terminal times $\boldsymbol{t}^*_{f,k}(\boldsymbol{X}^{(j)}(T_k^{(j)}))$ at each sample point $\boldsymbol{X}^{(j)}(T_k^{(j)})$ and time step $T_k^{(j)}$ $(j = 1, \ldots, N_s,\ k = 1, \ldots, N_t)$. (Note that $T_k^{(i)}/T_{N_t+1}^{(i)} = T_k^{(j)}/T_{N_t+1}^{(j)}$ $(\forall i, j \in \{1, \ldots, N_s\},\ k = 1, \ldots, N_t)$.)

4. Construct the surrogate models of the optimal conflict resolution trajectories (inputs: the current condition at the current time step $\boldsymbol{X}(T_k)$; outputs: the optimal states at the next time step $\hat{\boldsymbol{x}}^*_{k+1}(\boldsymbol{X}(T_k))$, optimal controls at the current time step $\hat{\boldsymbol{u}}^*_k(\boldsymbol{X}(T_k))$ and optimal terminal times at the current time step $\hat{\boldsymbol{t}}^*_{f,k}(\boldsymbol{X}(T_k)))$ at each time step T_k $(k = 1, \ldots, N_t)$ from the set of the precomputed optimal solutions in Step 3 by using the polynomial chaos kriging method.

5. Generate the near-optimal states $\hat{\boldsymbol{x}}^*_{k+1}(\boldsymbol{X}(T_k))$, controls $\hat{\boldsymbol{u}}^*_k(\boldsymbol{X}(T_k))$ and terminal times $\hat{\boldsymbol{t}}^*_{f,k}(\boldsymbol{X}(T_k))$ of the conflict resolution trajectories from the surrogate models with any given inputs of the current condition $\boldsymbol{X}(T_k)$ at the current time step T_k.

In Step 5, the near-optimal states, controls and terminal times on the time interval between the current time and the terminal time can be successively generated from the surrogate models with the inputs of the current condition at the current time step, and these near-optimal solutions can be updated at the next time step based on the information of the condition at the next time step. With respect to implementation, though Steps 2 and 4 could be computationally expensive, N_s stochastic optimal control problems can be solved in parallel in Step 2 and the surrogate models can also be constructed in parallel in Step 4. While Steps 1–4 can be computed offline, Step 5 can be applied online by using the surrogate models already constructed by Steps 1–4. After constructing the surrogate models, each aircraft can determine its own conflict resolution trajectory based on the information obtained from onboard equipment such as ADS-B[5].

[5]It is better for each aircraft to have the same surrogate models in order to obtain the same output with a certain input, and the surrogate models are built on the ground (under centralized ATM architecture). Once the surrogate models are constructed and transferred to aircraft, aircraft can determine their own conflict resolution trajectories from the surrogate models based on the information of current conditions obtained from onboard equipment (under decentralized ATM architecture).

Chapter 5

Numerical Simulations

In this chapter, numerical simulations of the two-dimensional conflict detection and resolution problem are conducted to demonstrate the performance and effectiveness of the proposed conflict detection and resolution algorithms. First, the conflict detection problem is solved by the probabilistic conflict detection algorithm based on the PCE method. After that, the conflict resolution problem is solved by the stochastic optimal control method. Lastly, the near-optimal control problem for conflict resolution is formulated and solved. The simulations are performed on a computer with a 3.20 GHz Intel Xeon E3-1225 v2 processor and 32 GB RAM.

5.1 Probabilistic Conflict Detection

5.1.1 Problem Description

As shown in Fig. 5-1, we consider the two-dimensional conflict scenario between two aircraft, labeled 1 and 2. A conflict is defined by the minimum separation requirement $d_{H\min}$ established by ICAO [64], which is set to 5 nmi in en route airspace. As shown in Fig. 5-1, we consider two aircraft flying toward the merging point $(x, y) = (0, 0\text{ nmi})$ without any maneuvers or control inputs. Both aircraft fly level at the same altitude and the same constant airspeed v of 400 kt. The heading angle ψ_i $(i = 1, 2)$ $(-\pi/2 \leq \psi_i \leq \pi/2)$ is randomly set to a constant value: $\psi_1 = 0.49$ and $\psi_2 = -0.34$. In

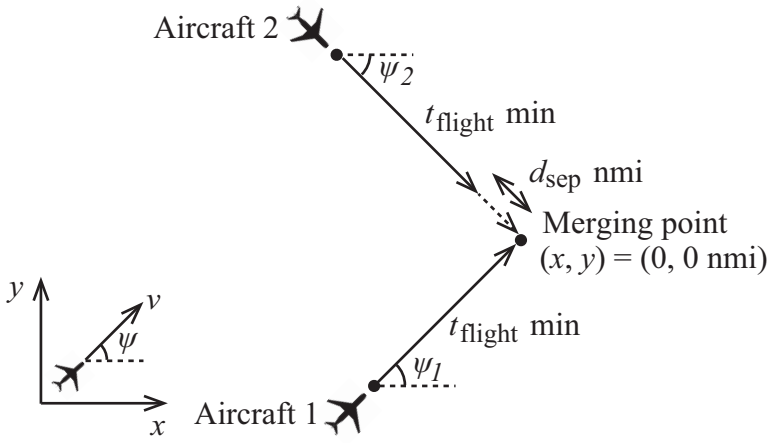


Figure 5-1: Conflict scenario for conflict detection problem.

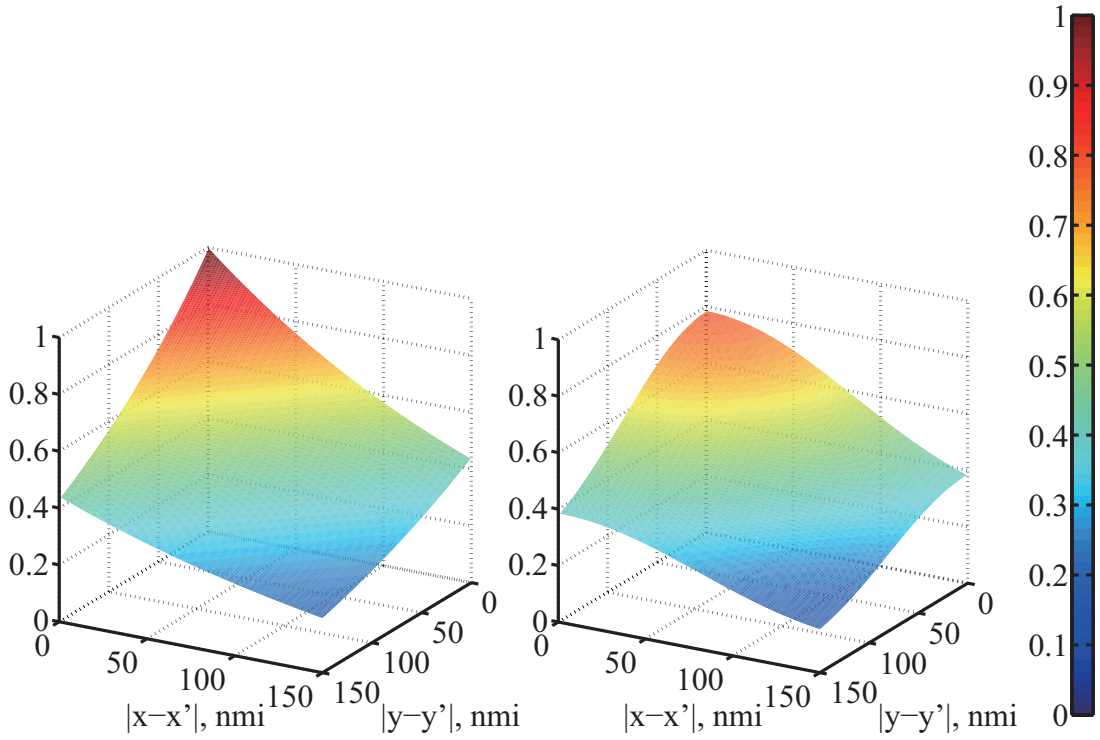
addition, as shown in Fig. 5-1, t_{flight} is the nominal flight time to the merging point for aircraft 1 in the absence of the wind field and set to 10 min. In the absence of the wind field, aircraft 1 reaches the merging point after t_{flight} min flight, and the nominal separation between aircraft 1 and 2 after t_{flight} min flight d_{sep} ($0 \leq d_{\text{sep}} \leq 5$ nmi) is randomly selected: $d_{\text{sep}} = 3.72$ nmi, as shown in Fig. 5-1. The initial positions are determined geometrically in the absence of the wind field.

As described in Section 2.1, the spatially correlated wind model is considered by using the KL expansion. In Eqs. (2.5) and (2.6), the number of independent random variables N_{KL} is set to three, and accordingly the total number of random variables is six. To compute the KL expansion, x and y are defined over the domain D: $|x| \leq 150$ nmi and $|y| \leq 150$ nmi. Figure 5-2 shows the comparison between the exact covariance function $C((x, y), (x', y'))$ in Eq. (2.4) and the covariance function $\tilde{C}((x, y), (x', y'))$ obtained with the KL expansion, where $\tilde{C}((x, y), (x', y'))$ is given by the following equation[1]:

$$\tilde{C}((x, y), (x', y')) = \sum_{i=1}^{N_{\text{KL}}} \lambda_i g_i(x, y) g_i(x', y')$$

The root mean square (RMS) error between the exact covariance function and the covariance function obtained with the KL expansion is 0.0334, which is small enough

[1]See Appendix A and [63].



(a) Exact function (b) KL expansion

Figure 5-2: Comparison between exact covariance function and covariance function obtained with KL expansion.

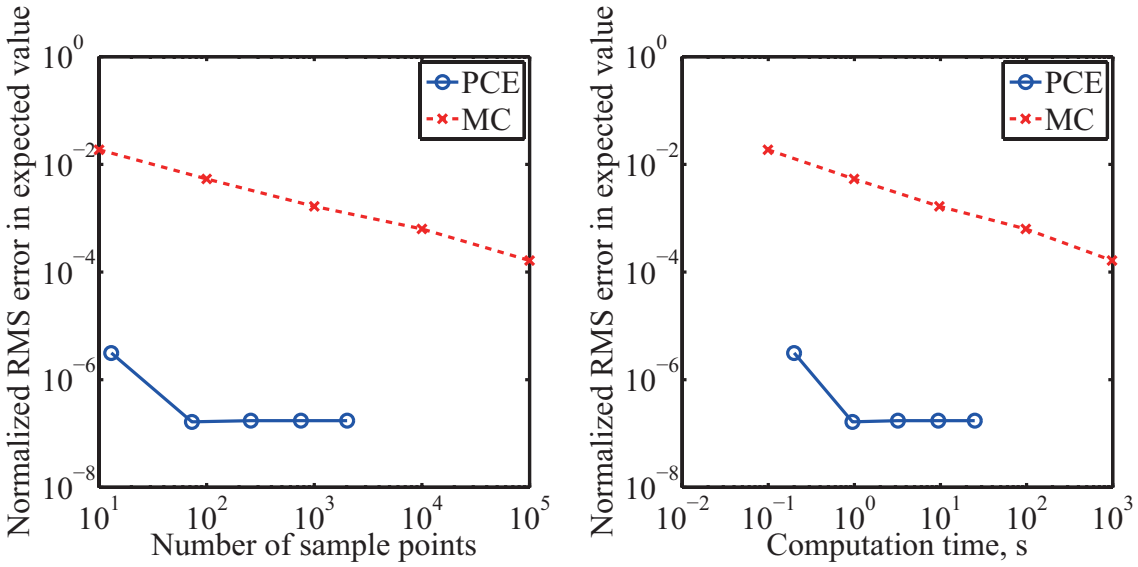
to suggest that the covariance function obtained with the KL expansion has a good approximation. In general, as N_{KL} gets larger, the covariance function can be obtained more accurately with the KL expansion. However, when the total number of random variables increases, the computational cost is considerably higher. Thus, in this study, N_{KL} is set to three, and the total number of random variables is six.

To demonstrate the effectiveness and performance of the conflict detection algorithm, the statistical information of the distance between aircraft L ($\mathrm{E}[L]$ and $\mathrm{V}[L]$) is computed on the time interval $t \in [0, t_{\mathrm{flight}}]$ by the PCE algorithm with the stochastic collocation method and MC method, and the results are compared with each other. In addition, we also demonstrate that the probability distribution of L can be approximated as a Gaussian distribution.

5.1.2 Simulation Results

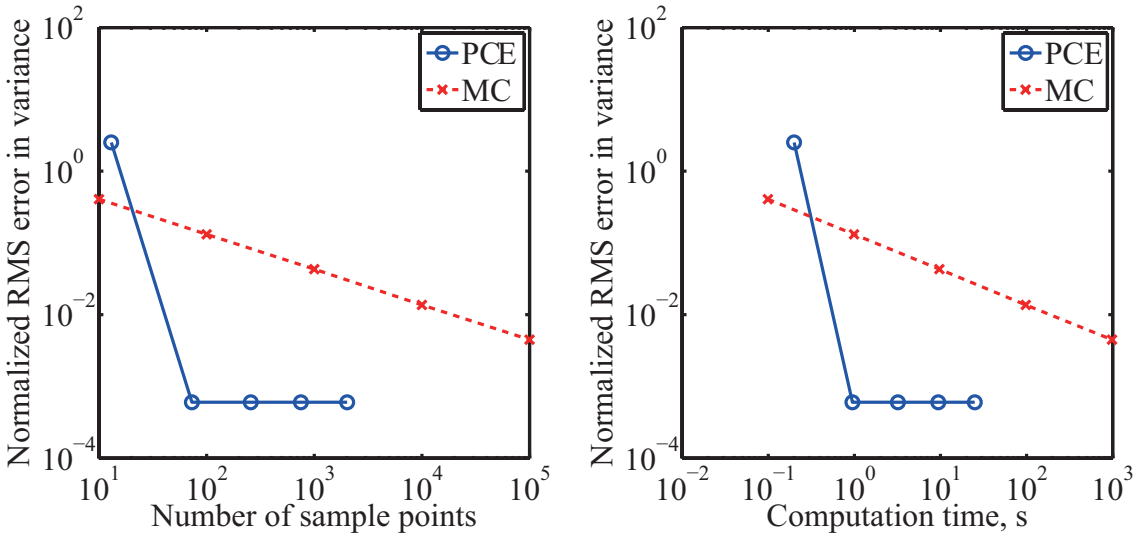
The conflict detection problem mentioned in Section 5.1.1 is solved by the probabilistic conflict detection algorithm, and the statistical information of the distance between aircraft L (E[L] and V[L]) is computed. Figures 5-3 and 5-4 show the normalized RMS errors in E[L] and V[L] (at time instant $t = t_{\text{flight}} = 10$ min), respectively. The normalized RMS errors are computed by the PCE algorithm with the stochastic collocation method and MC method with different numbers of sample points. The circle marker on the blue solid line is the normalized RMS error with the PCE algorithm. For the PCE algorithm, the approximation order P is set to three, and the accuracy level l of the sparse grid is set to two to six. As l gets larger, the number of collocation points increases: 13, 73, 257, 749 and 2021 (l = 2–6). The cross marker on the red dashed line indicates the normalized RMS error obtained by conducting 100 runs of the MC simulation. The number of sample points is set to 10, 100, 1000, 10000, and 100000. To compute the normalized RMS errors, the true values of E[L] and V[L] are assumed to be the mean values obtained by conducting 100 runs of the MC simulation with 100000 sample points, and the RMS errors are normalized to the true values. The computation times (average time for each run) are proportional to the number of sample points in Figs. 5-3 and 5-4. As shown in Figs. 5-3 and 5-4, to generate a solution with the same accuracy, the PCE algorithm uses a much smaller number of sample points than the MC method. The PCE algorithm requires only 73 collocation points (accuracy level $l = 3$) and approximately 0.9 s to obtain a good approximate solution. On the other hand, the MC method requires over 100000 sample points and correspondingly over 1000 s to yield the same accuracy as the PCE method. Therefore, the PCE method provides an accurate approximate solution while dramatically reducing computational cost. Compared with the MC method, which is computationally expensive and intractable especially when used within the iterative process such as optimization process mentioned in Section 5.2, our proposed algorithm based on the PCE method can dramatically reduce the computational cost and therefore greatly enhance the computational efficiency.

In addition, we demonstrate that the probability distribution of L can be approximated as the Gaussian distribution. Although the actual probability distribution of the stochastic solution is unknown, it can be estimated by the Monte Carlo simulation as the post-processing of the PCE method. Since L is approximated as the function of the random variables of the wind errors in Eq. (2.10), L can be used as the surrogate model (input: random variables of the wind errors, output: L). Thus, by the Monte Carlo simulation of the random variables in Eq. (2.10), the histogram of L, accordingly the probability distribution of L, can be easily estimated. We use the simulation results of the conflict detection problem to compute the probability distribution and histogram of L. Figure 5-5 shows the Gaussian distribution obtained by the moment matching technique and the histogram of L computed by the Monte Carlo simulation with 100000 sample points. The mean and variance of the Gaussian distribution are 3.7240 nmi and 0.0418 nmi^2, respectively. It should be noted that the probability density is proportional to the frequency. As shown in Fig. 5-5, the Gaussian distribution can accurately approximate the histogram of L representing the probability distribution of L. Therefore, we have demonstrated the accuracy of approximating the probability distribution of L as the Gaussian distribution. By approximating the probability distribution of L as the Gaussian distribution, we can readily compute the conflict probability between any pair of aircraft given by Eq. (2.21). Although the Gaussian approximation can be employed for probabilistic conflict detection and resolution in this study, it should be noticed that if the distance between aircraft is much smaller (e.g., in the case of collision avoidance), the Gaussian approximation may deteriorate the estimation of conflict (or collision) probability and different distributions may be better.



(a) Number of sample points (b) Computation time

Figure 5-3: Normalized root mean square (RMS) errors in expected value of distance between aircraft.



(a) Number of sample points (b) Computation time

Figure 5-4: Normalized root mean square (RMS) errors in variance of distance between aircraft.

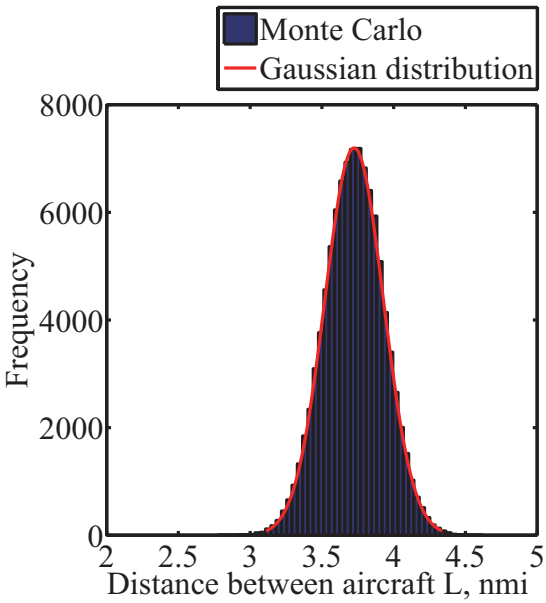


Figure 5-5: Comparison between Gaussian distribution (mean: 3.7240 nmi, variance: 0.0418 nmi^2) and histogram computed by Monte Carlo simulation with 100000 sample points for probability distribution of distance between aircraft.

5.2 Stochastic Optimal Control for Conflict Resolution

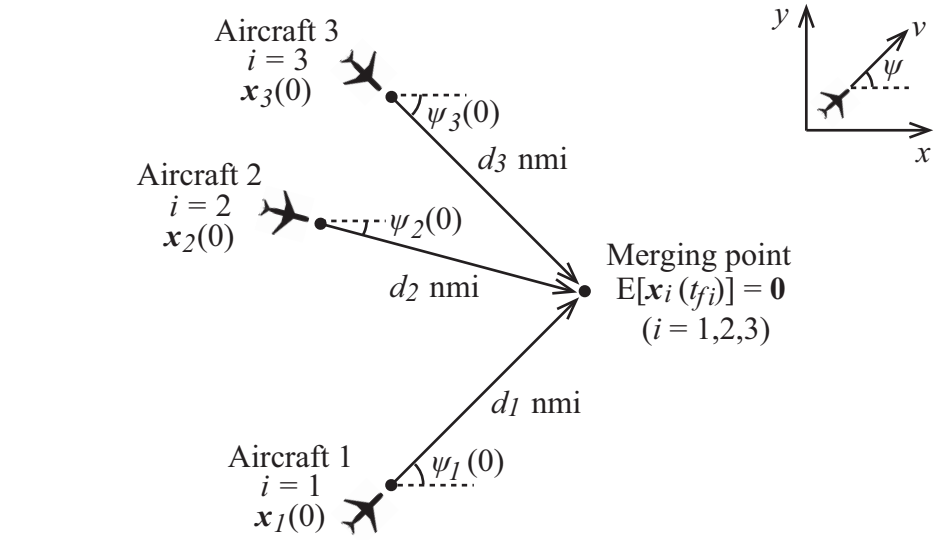
5.2.1 Problem Description

We consider the conflict resolution problem in two-dimensional horizontal plane in which the aircraft coming from different directions merge to the waypoint. As shown in Fig. 5-6, we consider two conflict scenarios: aircraft–aircraft conflict resolution problem in case 1; and aircraft–aircraft and aircraft–weather conflict resolution problem in case 2. Three aircraft are considered in case 1, and two aircraft and the moving convective weather region (cell) are considered in case 2. We assume that the shape of the convective weather cell is an ellipse as shown in Fig. 5-6b. An aircraft–aircraft conflict is defined by the required minimum separation standard established by ICAO [64], and the horizontal separation requirement between aircraft $d_{Haa\min}$ is set to 5 nmi for the en route airspace. The required minimum separation between aircraft and the convective weather cell $d_{Haw\min}$ is set to 0 nmi. It should be noted that the distance between aircraft and the convective weather cell is the shortest distance between the aircraft position and the ellipse representing the weather cell.

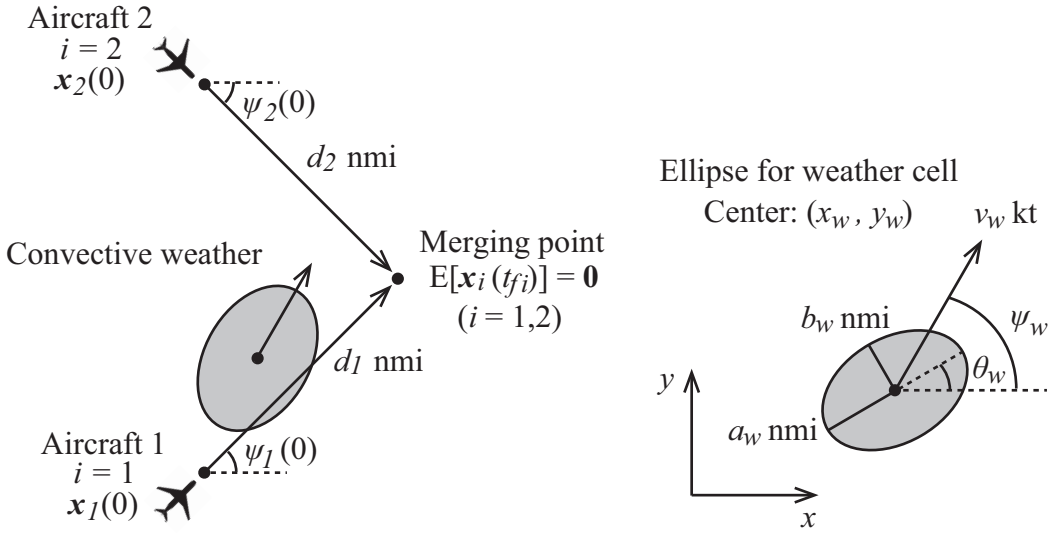
The merging point is set to $(x, y) = (0, 0 \text{ nmi})$ in both cases, and the initial condition $\boldsymbol{x}_i(0)$ ($i = 1, 2, 3$ in case 1 and $i = 1, 2$ in case 2) is determined geometrically by the initial heading angle $\psi_i(0)$ ($-\pi/2 \leq \psi_i \leq \pi/2$) and the direct distance between the initial position and the merging point d_i ($65 \leq d_i \leq 70$ nmi):

$$\boldsymbol{x}_i(0) = (-d_i \cos\psi_i(0), -d_i \sin\psi_i(0), \psi_i(0))^T \tag{5.1}$$

$\psi_i(0)$ and d_i are randomly selected, and the values of the parameters are shown in Tables 5.1 and 5.2. We assume that the subscript i indicates the aircraft sequences of reaching the merging point in ascending order of value i. Before solving the optimal control problem, the arrival sequences of aircraft are predetermined in ascending order of the magnitude of d_i, where d_i is given by satisfying the following condition: $d_i \leq d_j$ ($\forall i, j \in \{1, 2, 3\}$ in case 1 and $\forall i, j \in \{1, 2\}$ in case 2 : $i < j$). In case 2, as shown in



(a) Case 1



(b) Case 2

Figure 5-6: Conflict scenarios for conflict resolution problem.

Table 5.1: Parameters for conflict resolution problem in case 1

Parameters	Values
$\psi_1(0)$	0.11
$\psi_2(0)$	1.19
$\psi_3(0)$	-0.45
d_1, nmi	66.88
d_2, nmi	68.27
d_3, nmi	68.41

Table 5.2: Parameters for conflict resolution problem in case 2

Parameters	Values
$\psi_1(0)$	0.42
$\psi_2(0)$	-0.23
d_1, nmi	68.66
d_2, nmi	69.06
x_w, nmi	-30
y_w, nmi	-10
a_w, nmi	15
b_w, nmi	10
θ_w	$\pi/3$
v_w, kt	50
ψ_w	$\pi/4$

Fig. 5-6b, the shape of the convective weather cell is described as the ellipse with the rotation angle $\theta_w = \pi/3$, major radius $a_w = 15$ nmi and minor radius $b_w = 10$ nmi, and the initial position of the center of the ellipse is set to $(x_w, y_w) = (-30, -10$ nmi). The weather cell is moving with a constant velocity v_w of 50 kt, and the heading angle ψ_w of $\pi/4$. The values of these parameters for the moving convective weather cell are also shown in Table 5.2. We assume that the weather cell is not affected by the wind uncertainty and moves deterministically; however, the uncertainty contained in the moving weather cell is considered for the near-optimal conflict resolution problem in Section 5.3. In addition, the airspeed v is set to the same constant value of 400 kt, but the airspeed measurement error is also considered for more general conflict resolution problem in Section 5.3.

The terminal condition $\boldsymbol{x}_i(t_{fi})$ is given by the following equation:

$$\mathrm{E}\left[\boldsymbol{x}_i(t_{fi})\right] = (0,0,0)^T \tag{5.2}$$

where t_{fi} is the terminal time (time of arrival at the merging point): $t_{fi} \leq t_{fj}$ ($\forall i,j \in \{1,2,3\}$ in case 1 and $\forall i,j \in \{1,2\}$ in case 2 : $i<j$). Since $\boldsymbol{x}_i(t)$ contains the wind errors, the terminal condition in Eq. (5.2) is considered as the expected value.

As to the wind uncertainty, to quantify the effects of the wind correlation on the optimal conflict resolution trajectory, we consider the two different wind error models: the spatially correlated and non-correlated wind models. In the correlated case, the spatially correlated wind model is considered by using the KL expansion described in Section 2.1. As in the numerical simulations of conflict detection, the number of independent random variables N_{KL} in Eqs. (2.5) and (2.6) is set to three, and the total number of random variables is six. On the other hand, the wind correlation is completely ignored in the non-correlated case, which represents a simple baseline for comparison. The non-correlated wind error model is also widely used in ATM research because of its simplicity. In the non-correlated case, the wind error is assumed to be the Gaussian random variable with zero mean and the standard deviation $\sigma_w = 10.40$ kt [56], which is independent in the x and y directions and each aircraft. For the non-correlation model, the total number of independent random variables is six. For applying the PCE algorithm, on the basis of the numerical simulations of conflict detection in Section 5.1.2[2], the approximation order P is set to three and the accuracy level l is set to three. Accordingly, the number of collocation points is 73 for six random variables in both the correlated and non-correlated cases.

The aircraft dynamics are given by Eqs. (2.1)–(2.3), and the constraints on the variables are given as follows: $-\pi/2 \leq \psi \leq \pi/2$ and $-\pi/120 \leq u \leq \pi/120$ s^{-1}. We

[2]Even though constant ψ simplifies the problem and the stochastic solution in Section 5.1, the problem is still nonlinear and reasonably complex due to the spatially correlated wind error introduced in Eqs. (2.5) and (2.6). We are also able to obtain a good approximate solution when applying the proposed algorithms to a more complicated case in this section for a more general conflict resolution problem with the same number of collocation points as the simplified case in Section 5.1.

consider the stochastic safety constraint condition that the required minimum conflict probability is 0.1 to yield the effective conflict resolution trajectory [23]:

$$\Pr\left[\mathcal{C}_{ij}\right] \leq 0.1 \qquad (\forall i, j \in \{1,2,3\} : i < j) \qquad \text{in case 1} \tag{5.3}$$

$$\left.\begin{array}{l} \Pr\left[\mathcal{C}_{ij}\right] \leq 0.1 \\ \Pr\left[\mathcal{C}_{wi}\right] \leq 0.1 \end{array}\right\} \qquad (\forall i, j \in \{1,2\} : i < j) \qquad \text{in case 2} \tag{5.4}$$

where $\Pr\left[\mathcal{C}_{ij}\right]$ is the conflict probability between the ith and jth aircraft; and $\Pr\left[\mathcal{C}_{wi}\right]$ indicates the conflict probability between the ith aircraft and the convective weather cell. Both $\Pr\left[\mathcal{C}_{ij}\right]$ and $\Pr\left[\mathcal{C}_{wi}\right]$ are given by Eq. (2.21). For solving the stochastic optimal control problem to determine the conflict resolution trajectory, the following cost function J is minimized:

$$J = \sum_{i=1}^{N_a} \left(t_{fi} + 10^3 \int_0^{t_{fi}} |u_i(t)|^2 \mathrm{d}t \right) \tag{5.5}$$

where N_a is the number of aircraft: $N_a = 3$ in case 1 and $N_a = 2$ in case 2. Note that the cost function in Eq. (5.5) is a specific realization of the general form of a cost function in Eq. (3.15).

As described in Eqs. (5.1)–(5.5), the multiple-phase (three-phase in case 1 and two-phase in case 2) optimal control problems are formulated. In case 1, three aircraft ($i = 1, 2, 3$) and three conflicts ($\mathcal{C}_{12}, \mathcal{C}_{13}, \mathcal{C}_{23}$) are considered in the first phase ($t \in [0, t_{f1}]$); two aircraft ($i = 2, 3$) and one conflict ($\mathcal{C}_{23}$) are considered in the second phase ($t \in [t_{f1}, t_{f2}]$); and one aircraft ($i = 3$) is considered in the third phase ($t \in [t_{f2}, t_{f3}]$). In case 2, two aircraft ($i = 1, 2$) and three conflicts ($\mathcal{C}_{12}, \mathcal{C}_{w1}, \mathcal{C}_{w2}$) are considered in the first phase ($t \in [0, t_{f1}]$); one aircraft ($i = 2$) and one conflict ($\mathcal{C}_{w2}$) are considered in the second phase ($t \in [t_{f1}, t_{f2}]$). We apply the proposed stochastic optimal control method to the conflict resolution problem described in Section 5.2.1. With the stochastic optimal control method based on the pseudospectral method, the nonlinear optimal control problem can be solved, and the dynamic variables including the conflict resolution trajectory and maneuvers are optimized. Through the numer-

ical simulations, the effectiveness and performance of the stochastic optimal control method are evaluated and demonstrated.

5.2.2 Simulation Results

Figure 5-7 shows the expected values of the conflict resolution trajectories including the aircraft positions at time instant $t = t_{f1}$ in case 1. Figures 5-8 and 5-9 show the time histories of the heading angles and controls in case 1. Figure 5-10 shows the expected values of the conflict resolution trajectories including the aircraft positions at time instant $t = t_{f1}$ in case 2. The time histories of the heading angles and control variables in case 2 are shown in Figs. 5-11 and 5-12. Tables 5.3 and 5.4 indicate the terminal time at the merging point for each aircraft in cases 1 and 2, respectively. The characteristics of the conflict resolution trajectories in the correlated and non-correlated cases are similar to each other. In case 1, as shown in Fig. 5-7, aircraft 2 ($i = 2$, green line) and 3 ($i = 3$, red line) take a detour to avoid the potential conflicts, while aircraft 1 ($i = 1$, blue line) does not take a detour and flies toward the merging point without any conflict resolution maneuvers in both the correlated and non-correlated wind cases. The time histories of the heading angles and control variables of aircraft 1 in the correlated and non-correlated wind cases are also almost the same, as shown in Figs. 5-8 and 5-9. In addition, as shown in Table 5.3, the terminal time of aircraft 1 in the correlated and non-correlated wind cases are almost the same. It indicates that the trajectory of aircraft 1 is not affected by resolving conflict and these conflict resolution trajectories can minimize the cost function in Eq. (5.5). On the other hand, the conflict resolution trajectories of aircraft 2 and 3 in the correlated and non-correlated cases are also similar to each other. However, as shown in Fig. 5-7, the conflict resolution trajectories of aircraft 2 and 3 with the non-correlated wind model require a longer detour than the correlated one. Moreover, as shown in Table 5.3, the terminal times of aircraft 2 and 3 in the non-correlated wind case are also longer than that in the correlated one. It indicates that the distance between aircraft in the non-correlated wind case is necessary to be larger than the correlated wind case to guarantee safety and the resolution of the potential conflicts. That is, the distance

between aircraft can be smaller under the correlated wind uncertainty, which is the more realistic wind error model than the non-correlated one. The wind correlation can have a significant effect on the distance between aircraft and therefore the optimal conflict resolution trajectory.

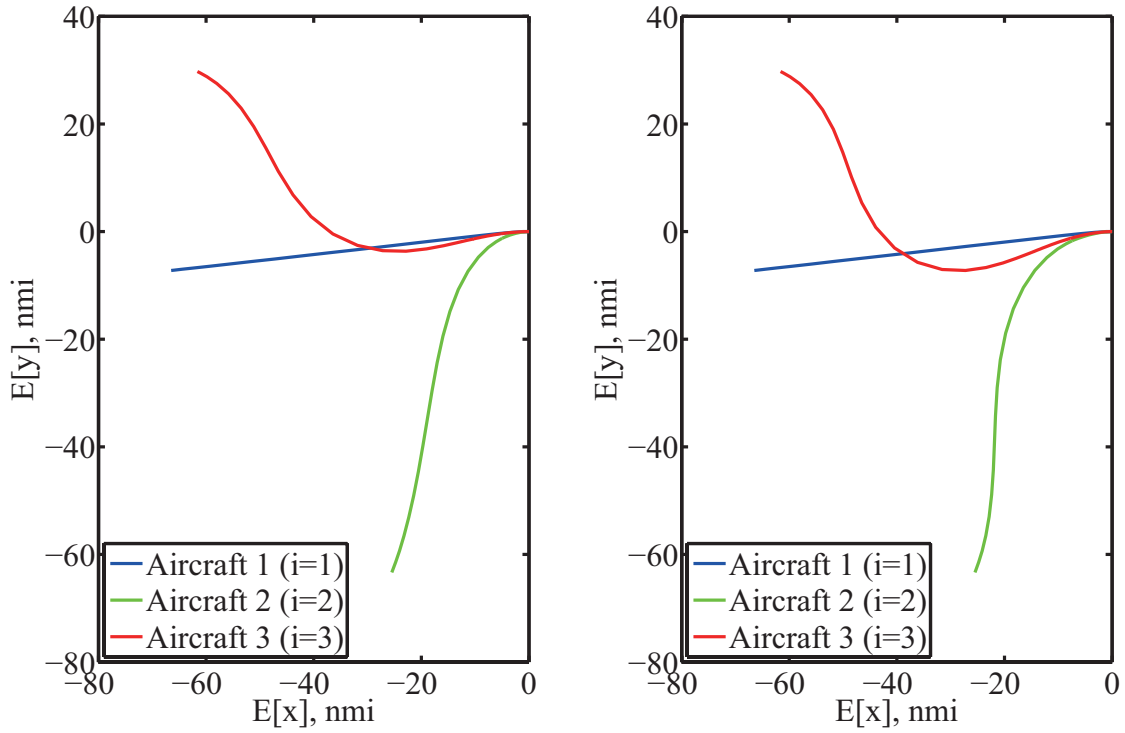
In case 2, the characteristics of the conflict resolution trajectories in the correlated and non-correlated cases are also similar to each other. As shown in Fig. 5-10, aircraft 1 ($i = 1$, blue line) and 2 ($i = 2$, red line) take a detour to avoid the potential conflicts in both the correlated and non-correlated wind cases. Aircraft 1 in case 2 takes a detour to resolve the aircraft–weather conflict in contrast to aircraft 1 in case 1 that does not take a detour and flies toward the merging point without any conflict resolution maneuvers. The conflict resolution trajectories of aircraft 1 in the correlated and non-correlated wind cases are almost the same, and the conflict resolution trajectory of aircraft 2 with the non-correlated wind model requires a longer detour than the correlated one. In addition, as shown in Table 5.4, the terminal time of aircraft 2 in the non-correlated wind case is longer than the correlated one, while the terminal times of aircraft 1 in the correlated and non-correlated wind cases are almost the same. It indicates that the wind correlation can have a significant effect on the distance between a pair of aircraft but less effect on the distance between aircraft and the weather cell. This is because the moving convective weather cell is not affected by the wind uncertainty.

Moreover, the separation margins are optimized to satisfy the stochastic safety constraint described in Eqs. (2.21), (5.3) and (5.4). Figure 5-13 shows the expected values of the separation margins in case 1, which are computed as $\mathrm{E}[L] - d_{H\min}$. Figure 5-14 shows the variances of the separations $\mathrm{V}[L]$ in case 1. Figure 5-15 shows the expected values of the separation margins in case 2, and Fig. 5-16 shows the variances of the separations in case 2. Figures 5-13 and 5-15 also indicate the standard deviation (SD) of the separation margin, where the SD (error bar) is obtained from $\mathrm{V}[L]$ in Figs. 5-14 and 5-16. Tables 5.5 and 5.6 show the minimum values of the expected values of the separation margins in Figs. 5-13 and 5-15. Since our proposed stochastic optimal control method incorporates the PCE algorithm, the expected

value and variance of the separation are easily computed. As shown in Figs. 5-14 and 5-16, the variance of the separation increases with time because the effect of the wind uncertainty accumulates over time, and the magnitude of the variance of the separation represents the effects of the wind uncertainty and correlation on the separation between aircraft. In the correlated wind case, the variance of the separation between the pair of aircraft can be smaller than the non-correlated wind case, because the wind errors experienced by any two aircraft are correlated with each other. On the other hand, the completely independent wind error in the non-correlated model leads to the much larger variance of the separation between the pair of aircraft than the correlated wind model, as shown in Figs. 5-14 and 5-16. Accordingly, as shown in Figs. 5-13 and 5-15, the SD of the separation in the non-correlated wind case is also much larger than the correlated wind case. As shown in Figs. 5-15 and 5-16, the variances and SDs of the separations between aircraft and the weather cell are much larger than those of the separations between the pair of aircraft in the correlated case, and the variances and SDs of the separations between aircraft and the weather cell in the correlated and non-correlated cases are similar to each other. This is because the weather cell is not affected by the wind uncertainty and the wind correlation can have less effect on the separation between aircraft and the weather cell. Thus, as shown in Figs. 5-13 and 5-15 and Tables 5.5 and 5.6, the separation margin between the pair of aircraft in the non-correlated wind case is necessary to be larger than the correlated wind case to satisfy the stochastic safety constraint described in Eqs. (2.21), (5.3) and (5.4), i.e., to guarantee the resolution of the potential conflicts. This results in an overestimation of the conflict probability in the non-correlated wind case. Therefore, using the correlated wind error model, which is the more realistic wind uncertainty than the non-correlated one, we can reduce the conservativeness in the estimation of the conflict probability and the separation margin under the non-correlated wind uncertainty.

To generate the optimal conflict resolution trajectory, the computation times for solving the stochastic optimal control problems for conflict resolution are approximately 38 s and 231 s in cases 1 and 2, respectively, by employing the computa-

tionally efficient PCE algorithm. According to the results of the conflict detection problem mentioned in Section 5.1.2, it can be computationally intractable to use the MC method instead of the PCE algorithm. Our proposed stochastic optimal control method that incorporates the PCE algorithm into the pseudospectral method can optimize the dynamic variables including the conflict resolution trajectory and maneuvers, and the resolution of the potential conflicts is guaranteed by incorporating the conflict detection algorithm and considering the stochastic safety constraint described in Eqs. (2.21), (5.3) and (5.4). The stochastic optimal control method can provide the effective separation margin and conflict resolution trajectory that can be guaranteed to avoid potential conflicts under the wind uncertainty. When the stochastic wind model is not considered, the separation between aircraft has no margin from the required minimum separation, which is not the robust conflict resolution trajectory at all under uncertain environments during the flight. In contrast, by considering the wind uncertainty, the stochastic optimal control method can provide the effective separation margins and conflict resolution trajectories that can be guaranteed to avoid potential conflicts under the wind uncertainty. In particular, using the correlated wind model, which is the more realistic wind uncertainty than the non-correlated one, we can reduce the conservativeness of separation distances under the non-correlated wind uncertainty by approximately 2.8 nmi on average. Accordingly, we can mitigate the air traffic congestion and increase the throughput capacity of the air traffic by using our proposed stochastic algorithms.



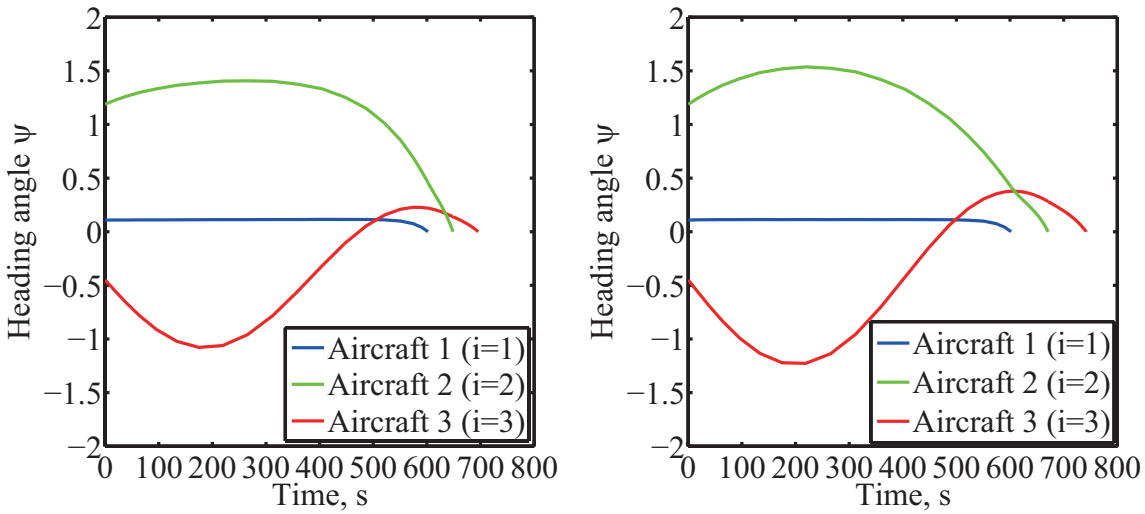
(a) Correlated wind

(b) Non-correlated wind

(c) Correlated wind ($t = t_{f1}$)

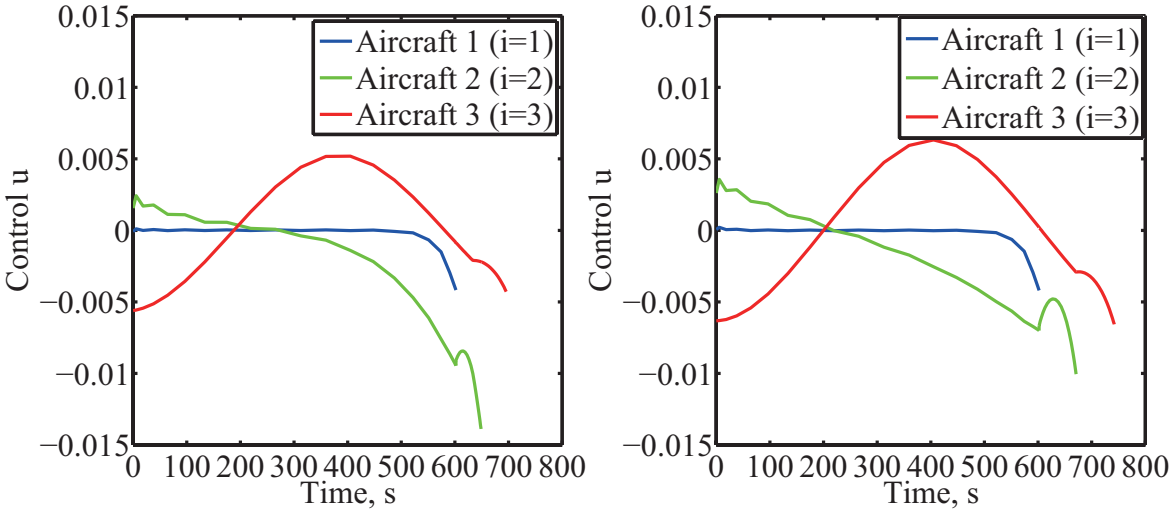
(d) Non-correlated wind ($t = t_{f1}$)

Figure 5-7: Optimal conflict resolution trajectory in case 1.



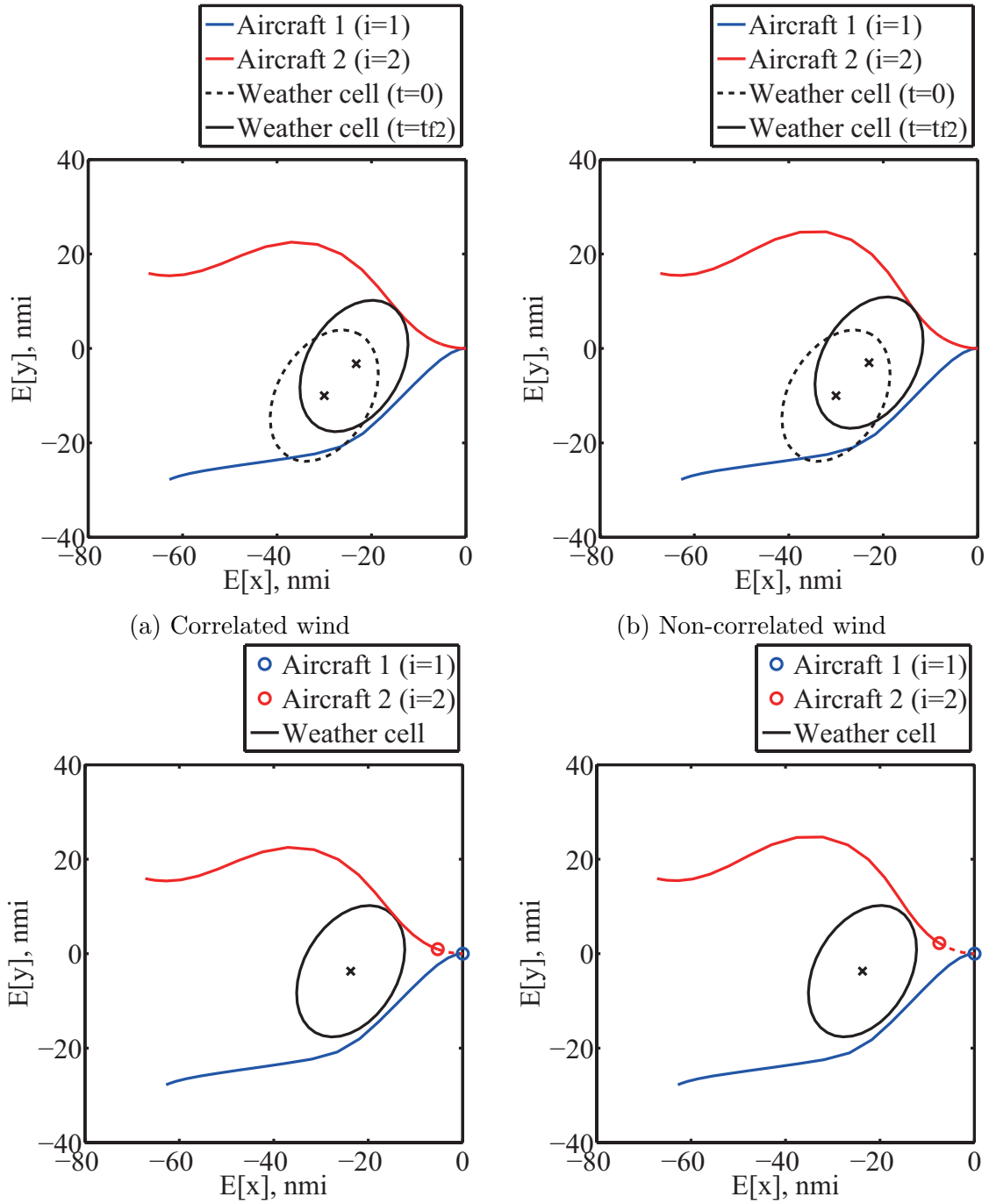
(a) Correlated wind (b) Non-correlated wind

Figure 5-8: Time history of heading angle in case 1.



(a) Correlated wind (b) Non-correlated wind

Figure 5-9: Time history of control variable in case 1.



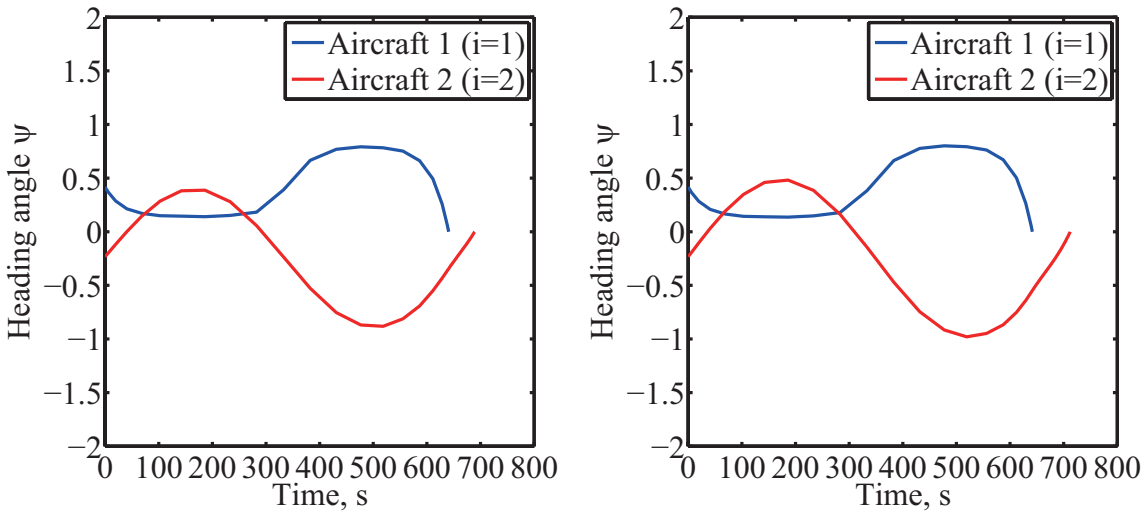
(a) Correlated wind

(b) Non-correlated wind

(c) Correlated wind ($t = t_{f1}$)

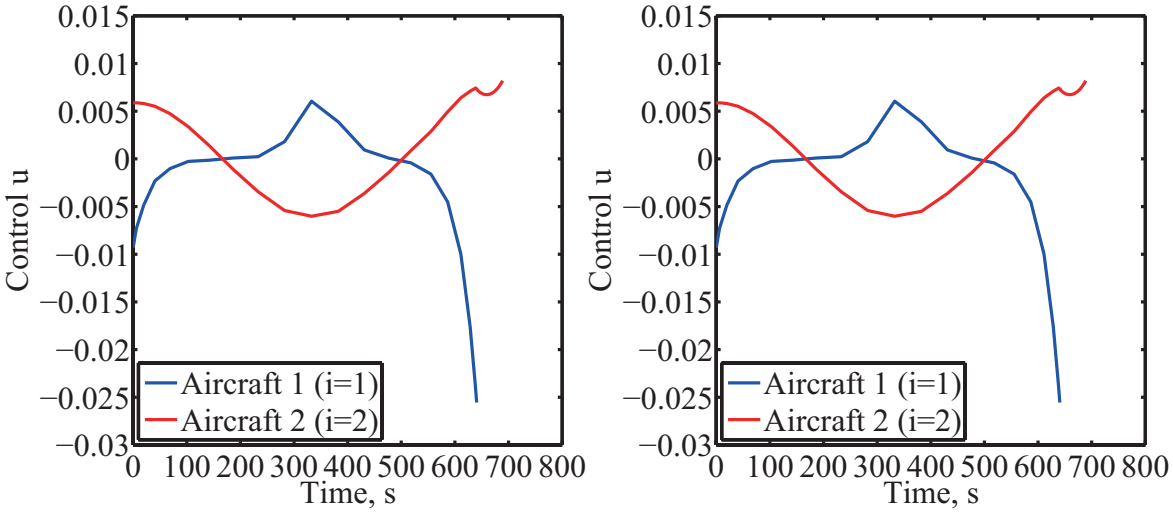
(d) Non-correlated wind ($t = t_{f1}$)

Figure 5-10: Optimal conflict resolution trajectory in case 2.



(a) Correlated wind (b) Non-correlated wind

Figure 5-11: Time history of heading angle in case 2.



(a) Correlated wind (b) Non-correlated wind

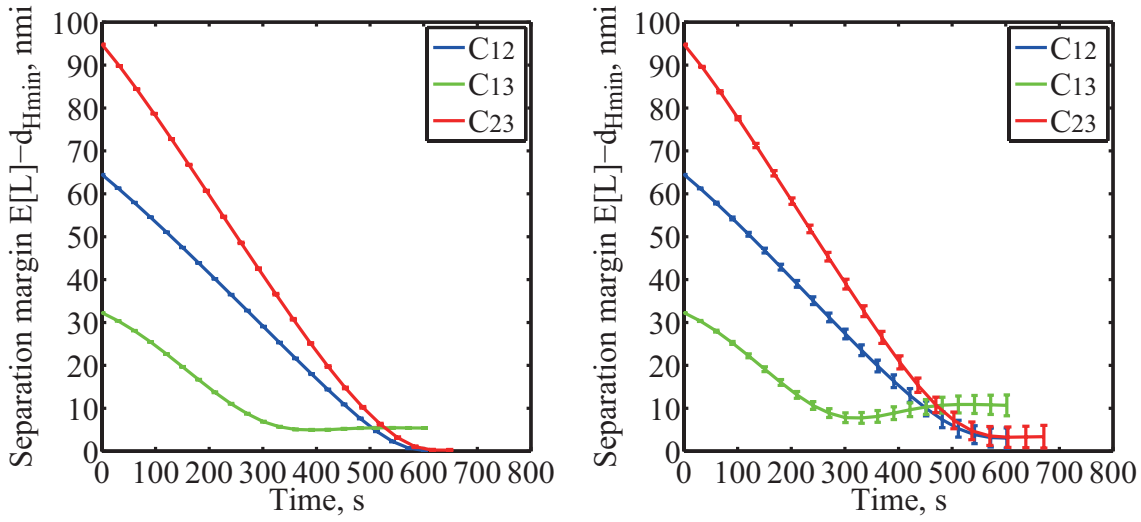
Figure 5-12: Time history of control variable in case 2.

Table 5.3: Terminal time at merging point in case 1

Variables	Correlated wind	Non-correlated wind
t_{f1}, s	601.97	601.97
t_{f2}, s	648.70	671.06
t_{f3}, s	695.78	742.23

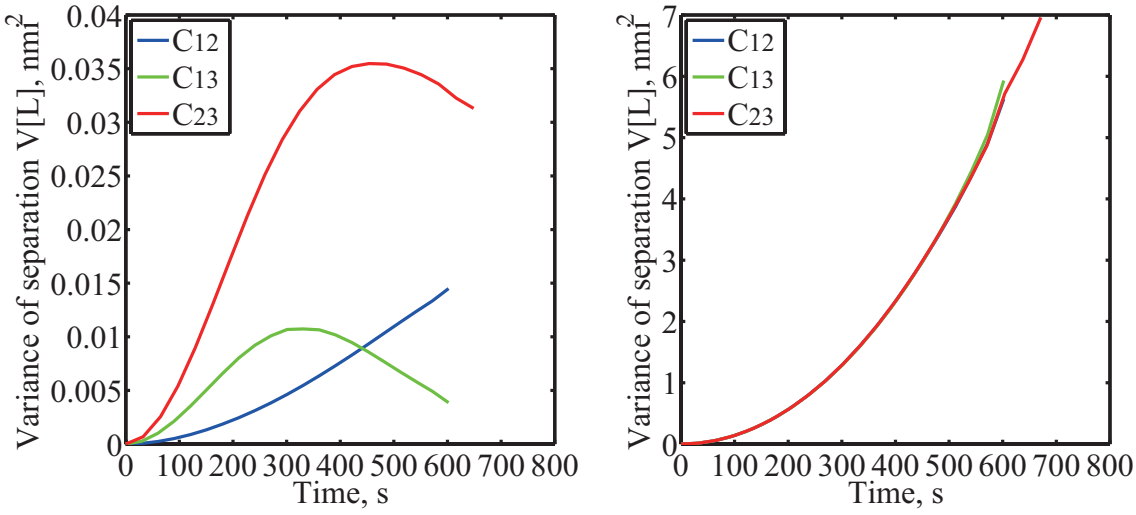
Table 5.4: Terminal time at merging point in case 2

Variables	Correlated wind	Non-correlated wind
t_{f1}, s	640.61	641.60
t_{f2}, s	689.13	712.24



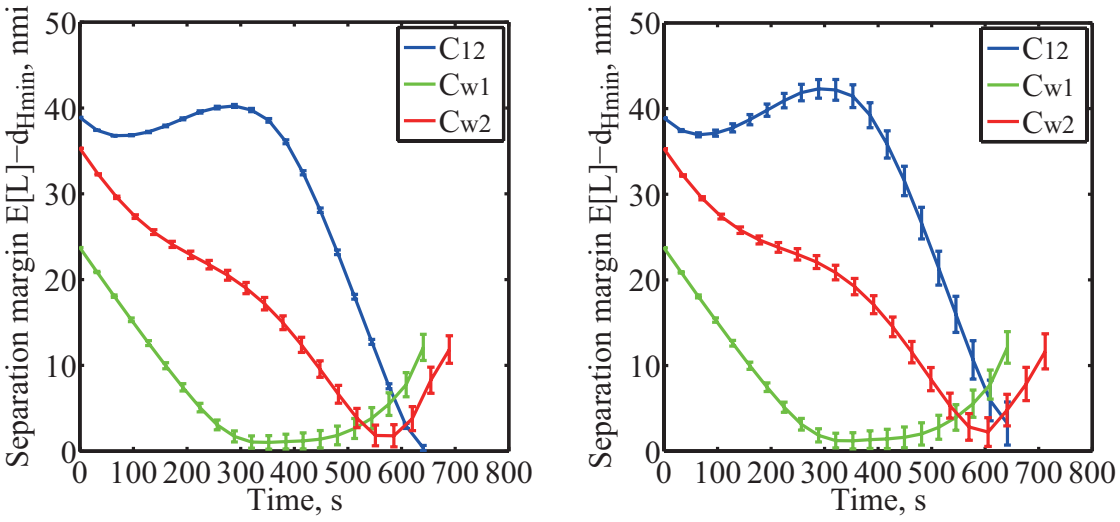
(a) Correlated wind (b) Non-correlated wind

Figure 5-13: Time history of expected values and standard deviation (error bar) of separation margin in case 1.



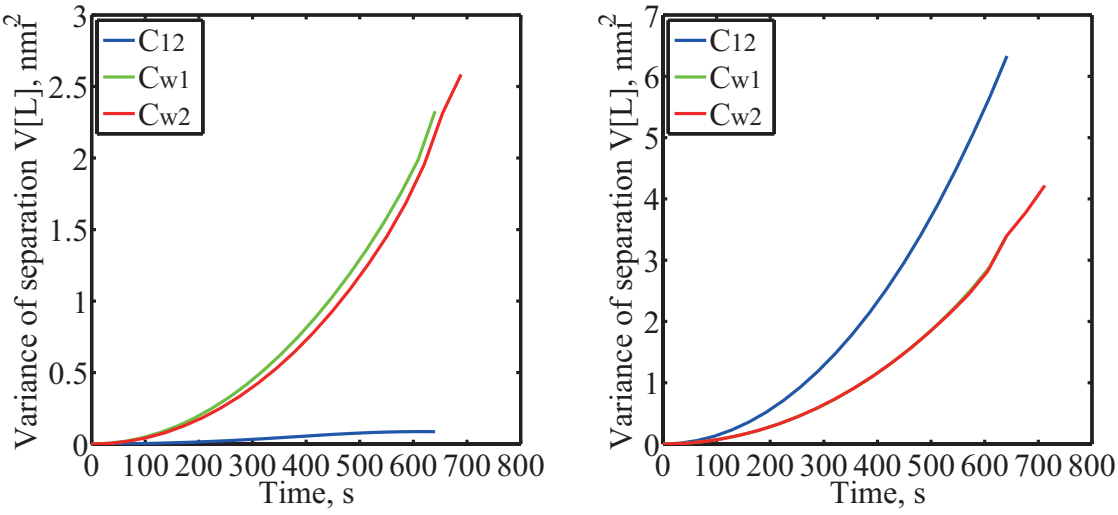
(a) Correlated wind (b) Non-correlated wind

Figure 5-14: Time history of variance of separation in case 1.



(a) Correlated wind (b) Non-correlated wind

Figure 5-15: Time history of expected values and standard deviation (error bar) of separation margin in case 2.



(a) Correlated wind (b) Non-correlated wind

Figure 5-16: Time history of variance of separation in case 2.

Table 5.5: Minimum separation margin (expected value) in case 1

Pairs	Correlated wind	Non-correlated wind
$\mathcal{C}_{12}$ (aircraft 1 and 2)	0.15 nmi	3.04 nmi
$\mathcal{C}_{13}$ (aircraft 1 and 3)	4.95 nmi	7.68 nmi
$\mathcal{C}_{23}$ (aircraft 2 and 3)	0.23 nmi	3.26 nmi

Table 5.6: Minimum separation margin (expected value) in case 2

Pairs	Correlated wind	Non-correlated wind
$\mathcal{C}_{12}$ (aircraft 1 and 2)	0.38 nmi	3.22 nmi
$\mathcal{C}_{w1}$ (weather and aircraft 1)	0.99 nmi	1.67 nmi
$\mathcal{C}_{w2}$ (weather and aircraft 2)	1.79 nmi	2.21 nmi

5.3 Stochastic Near-Optimal Control for Conflict Resolution

5.3.1 Problem Description

On the basis of the problem formulation for probabilistic conflict resolution described in Section 5.2, the near-optimal control problem for conflict resolution is formulated and solved. As shown in Fig. 5-6, we consider two conflict scenarios based on the conflict situations in Section 5.2: aircraft–aircraft conflict resolution problem in case 1; and aircraft–aircraft and aircraft–weather conflict resolution problem in case 2. In order to consider more general probabilistic conflict resolution problem, as to the uncertainties, we deal with the airspeed measurement error and the uncertainty contained in the moving convective weather cell as well as the spatially correlated wind uncertainty.

The surrogate models of the optimal conflict resolution trajectories are constructed by using the polynomial chaos kriging algorithm. We consider the discrete time steps: $t_0^{(p)} = T_1^{(p)} < \cdots < T_{N_t+1}^{(p)} = t_f^{(p)}$, where N_t is set to 20; and p denotes the phase ($p = 1, 2, 3$ in case 1 and $p = 1, 2$ in case 2). Note that the multiple-phase (three-phase in case 1 and two-phase in case 2) optimal control problems are formulated as described in Section 5.2, and $t_0^{(p)}$ and $t_f^{(p)}$ are the initial and terminal times in the pth phase, respectively ($t_f^{(p)} = t_0^{(p+1)}$). The surrogate models are built at each time step T_k ($k = 1, \ldots, N_t$). The input variables of the surrogate models $\boldsymbol{X}(T_k)$ are the current condition at the current time step T_k: the current state variables $\boldsymbol{x}_i(T_k)$ and airspeed $v_i(T_k)$ ($i = 1, 2, 3$) in case 1; and the current state variables $\boldsymbol{x}_i(T_k)$, airspeed $v_i(T_k)$ ($i = 1, 2$) and parameters for the convective weather cell $(x_w, y_w, a_w, b_w, \theta_w, v_x, \psi_w)$ at the current time step T_k in case 2. In case 1, the total numbers of the input variables are 12, 8 and 4 in the first, second and third phases, respectively. In case 2, the total numbers of the input variables are 15 and 11 in the first and second phases, respectively. The outputs of the surrogate models are the optimal states $\boldsymbol{x}_{i,k+1}^*(\boldsymbol{X}(T_k))$ at the next time step T_{k+1}, optimal controls $u_{i,k}^*(\boldsymbol{X}(T_k))$ at the current

Table 5.7: Standard deviations for variables of initial conditions in case 1 ($i = 1, 2, 3$)

Parameters	Values
x_i, nmi	1
y_i, nmi	1
ψ_i	$\pi/36$
v_i, kt	5.38

Table 5.8: Standard deviations for variables of initial conditions in case 2 ($i = 1, 2$)

Parameters	Values
x_i, nmi	1
y_i, nmi	1
ψ_i	$\pi/36$
v_i, kt	5.38
x_w, nmi	1
y_w, nmi	1
a_w, nmi	1
b_w, nmi	1
θ_w	$\pi/36$
v_w, kt	5
ψ_w	$\pi/36$

time step and optimal terminal times $t^*_{fi,k}(\boldsymbol{X}(T_k))$ at the current time step ($i = 1, 2, 3$ in case 1 and $i = 1, 2$ in case 2).

For applying the polynomial chaos kriging algorithm, the sample points of the initial conditions are generated by the Latin hypercube sampling method [143, 144]. The number of sample points N_s is set to 100, 200 and 300, and the surrogate models constructed with the different numbers of sample points are compared with each other. By using the stochastic optimal control method, N_s stochastic optimal control problems starting from the different initial conditions $\boldsymbol{X}^{(j)}(0) = \boldsymbol{X}^{(j)}(T_1^{(1)})$ ($j = 1, \ldots, N_s$) are solved for constructing the surrogate models of the optimal conflict resolution trajectories. The variables of the initial conditions are assumed to be Gaussian, and the means are consistent with the values of the initial conditions given by Eq. (5.1) and Tables 5.1 and 5.2 in Section 5.2. The standard deviations of these variables in cases 1 and 2 are shown in Tables 5.7 and 5.8, respectively. In Tables 5.7 and 5.8, the standard deviation of v, 5.38 kt, is determined based on the

speed measurement error and the air temperature error [30]. The airspeed indicators typically have errors with the standard deviation of 5 kt to the airspeed, and the air temperature error has the standard deviation of approximately 2 K, which leads to the airspeed error of about 2 kt. Assuming these two errors are independent, the standard deviation of 5.38 kt can be obtained. It should be noted that the airspeed measurement error is assumed to be constant in time.

The surrogate models are built with the set of N_s precomputed optimal solutions. Once the surrogate models are constructed, the near-optimal conflict resolution trajectories on the time interval between the current time and the terminal time can be obtained as the functions of the current conditions. Thus, the near-optimal solutions can be generated in real time from the surrogate models with the information of the current conditions without actually solving the computationally expensive stochastic optimal control problems. For assessing the surrogate models, we consider the randomly generated 100 different cases as the inputs of the surrogate models. Using these 100 cases, the near-optimal solution generated from the surrogate models and the exact optimal solution obtained by actually solving the optimal control problem are compared with each other.

5.3.2 Simulation Results

The near-optimal conflict resolution trajectories are generated from the surrogate models constructed by using the polynomial chaos kriging algorithm with the different numbers of sample points $N_s = 100, 200, 300$. In order to build the surrogate models, the stochastic optimal control problems starting from N_s selected initial conditions are solved. Figures 5-17 and 5-18 show the optimal solutions for the chosen $N_s = 200$ initial conditions in cases 1 and 2, respectively. By using the polynomial chaos kriging method, the surrogate models of the optimal conflict resolution trajectories are constructed from the set of the precomputed optimal solutions. Tables 5.9 and 5.10 indicate the maximum and RMS errors of the outputs of the surrogate models between the near-optimal and optimal solutions among 100 different cases and the comparison between the errors of the surrogate models built with the different

numbers of sample points. As shown in Tables 5.9 and 5.10, when N_s increases, the maximum and RMS errors generally become smaller, that is, the near-optimal solutions can be obtained more accurately. However, the increase of N_s from 200 to 300 does not provide significant improvements and changes in the maximum and RMS errors. It indicates that the maximum and RMS errors obtained by polynomial chaos kriging with 200 sample points are small enough to suggest that the surrogate models provide the approximate optimal conflict resolution trajectories accurately. In addition, the near-optimal solutions are generated from the surrogate models constructed by using three different surrogate modeling techniques, the polynomial chaos kriging, PCE and ordinary kriging methods, with $N_s = 200$ sample points. Tables 5.11 and 5.12 show the maximum and RMS errors of the outputs of the surrogate models between the near-optimal and optimal solutions among 100 different cases and the comparison between the errors of the surrogate models built with three different surrogate modeling techniques: the polynomial chaos kriging, PCE and ordinary kriging methods. As shown in Tables 5.11 and 5.12, the polynomial chaos kriging algorithm generally performs better than the PCE or kriging does. In reality, we do not have a prior knowledge that which approach performs better, PCE or kriging. The polynomial chaos kriging method can build more accurate surrogate models than either of the PCE or kriging method does. Therefore, the polynomial chaos kriging algorithm can perform better than the PCE or kriging does, and provide the highly accurate near-optimal solutions.

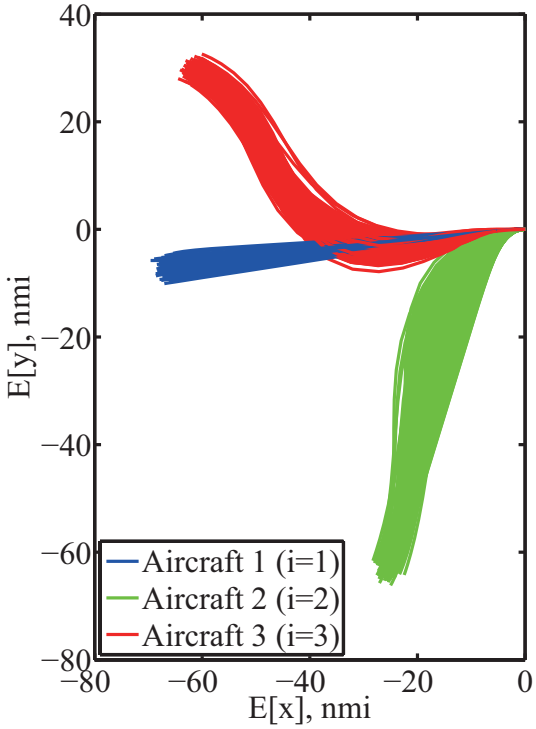
Tables 5.13 and 5.14 indicate the RMS errors of the expected values of the separations between the near-optimal and optimal solutions among 100 different cases. The near-optimal solutions are constructed by polynomial chaos kriging with $N_s = 200$ sample points. In Tables 5.13 and 5.14, the RMS errors are small enough to indicate that the surrogate models can generate the near-optimal solutions accurately. In addition, in all 100 cases, the near-optimal solutions also satisfy the stochastic safety constraint described in Eqs. (2.21), (5.3) and (5.4) as well as the optimal solutions. Thus, the near-optimal solutions can be guaranteed to avoid the potential conflicts without actually solving the stochastic optimal control problems, and it indicates that

the near-optimal solutions can be obtained accurately from the surrogate models.

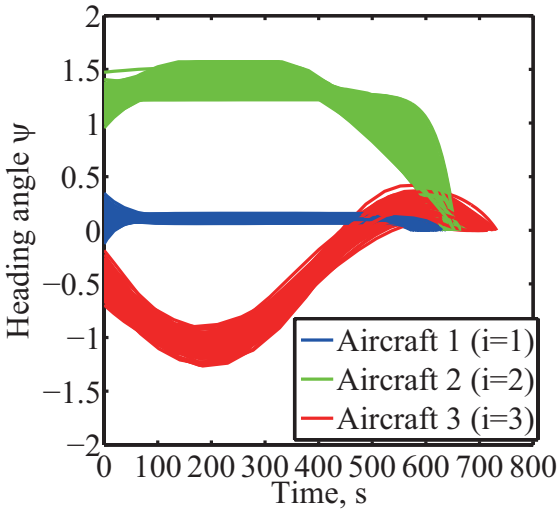
In order to demonstrate the accuracy and robustness of the near-optimal conflict resolution algorithm, Figs. 5-19 and 5-20 show several realizations of the near-optimal, optimal and nominal conflict resolution trajectories at the time steps $t = T_1^{(1)}, T_4^{(1)}, T_7^{(1)}, T_{10}^{(1)}, T_{13}^{(1)}, T_{16}^{(1)}$ in case 1. Figures 5-21–5-24 show the time histories of the near-optimal, optimal and nominal heading angles and controls at the time steps $t = T_1^{(1)}, T_4^{(1)}, T_7^{(1)}, T_{10}^{(1)}, T_{13}^{(1)}, T_{16}^{(1)}$ in case 1. Figures 5-25–5-30 also show several realizations of the near-optimal, optimal and nominal conflict resolution trajectories and the time histories of the near-optimal, optimal and nominal heading angles and controls at the time steps $t = T_1^{(1)}, T_4^{(1)}, T_7^{(1)}, T_{10}^{(1)}, T_{13}^{(1)}, T_{16}^{(1)}$ in case 2. The surrogate models are constructed by the polynomial chaos kriging method with $N_s = 200$ sample points. The solid lines denote the near-optimal solutions generated from the surrogate models with the current conditions (at the time steps $t = T_1^{(1)}, T_4^{(1)}, T_7^{(1)}, T_{10}^{(1)}, T_{13}^{(1)}, T_{16}^{(1)}$). The dashed lines are the optimal solutions obtained by actually solving the stochastic optimal control problems. The dotted lines indicate the nominal solutions, which are the optimal solutions in Section 5.2 representing a simple baseline for comparison. The circle markers denote the current conditions. In Figs. 5-19–5-24, the blue, green and red lines indicate the solutions of aircraft 1, 2 and 3 ($i = 1, 2, 3$) in case 1, respectively. In Figs. 5-25–5-30, the blue and red lines are the solutions of aircraft 1 and 2 ($i = 1, 2$) in case 2, respectively. As shown in Figs. 5-19–5-30, the near-optimal and optimal solutions closely match each other, and the near-optimal conflict resolution trajectories and maneuvers can be obtained accurately from the surrogate models with the information of the current conditions. The near-optimal solutions on the time interval between the current time and the terminal time can be generated from the surrogate models with the inputs of the current conditions, and these near-optimal solutions can be updated at the next time step based on the information of the conditions at the next time step. As shown in Figs. 5-19–5-30, we can obtain the different near-optimal solution depending on the current condition at each time step, which is also guaranteed to avoid the potential conflicts as well as the optimal solution. When the states on the precomputed optimal

trajectory deviate from the actual states due to the uncertainties, our proposed near-optimal feedback control method can accurately generate the near-optimal trajectory starting from the actual states in real time by using the surrogate models with the information of the actual states without solving another optimal control problem to obtain the correct optimal trajectory.

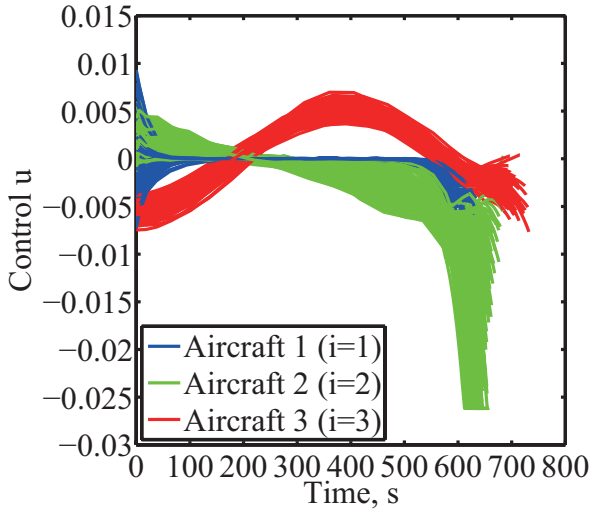
To generate the near-optimal conflict resolution trajectories, the computation times for constructing one surrogate model from a set of precomputed $N_s = 200$ optimal solutions by the polynomial chaos kriging method are approximately 46 s and 59 s on average in cases 1 and 2, respectively. In order to reduce the total computational cost, as mentioned in Section 4.3, the surrogate models can be built in parallel, and the stochastic optimal control problems can also be solved in parallel. Once the surrogate models are computed offline, the near-optimal solutions can be generated online from the surrogate models with the information of the current conditions. The surrogate models constructed by the polynomial chaos kriging method can accurately provide the approximate optimal conflict resolution trajectories that are guaranteed to avoid the potential conflicts without actually solving the computationally expensive stochastic optimal control problems. After constructing the surrogate models, each aircraft can determine its own conflict resolution trajectory based on the information of current conditions obtained from onboard equipment such as ADS-B. Through the numerical simulations, the effectiveness and performance of our proposed near-optimal conflict resolution algorithm are evaluated and demonstrated.



(a) Conflict resolution trajectory

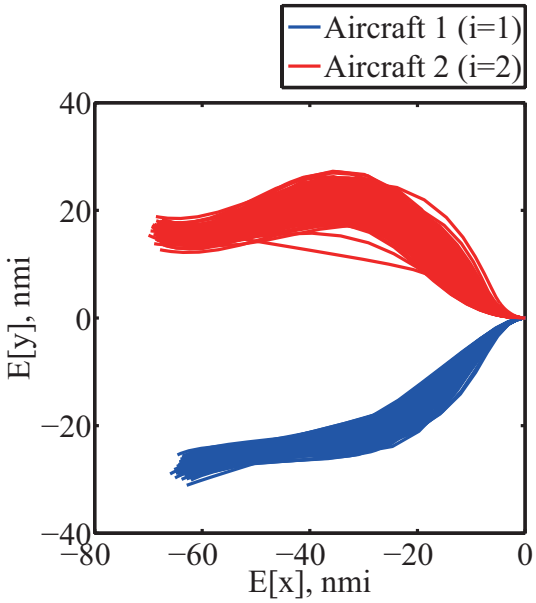


(b) Time history of heading angle

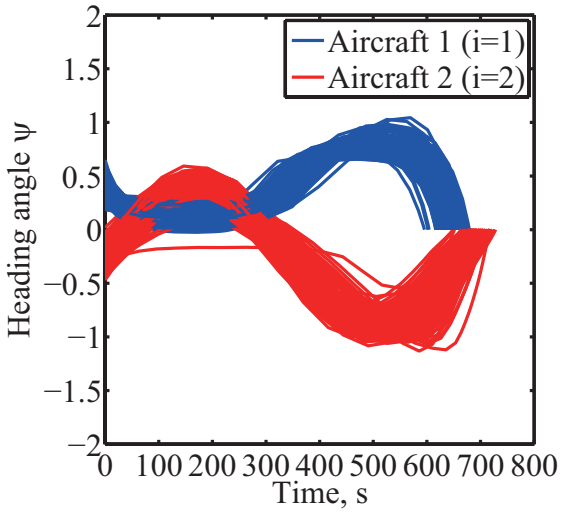


(c) Time history of control variable

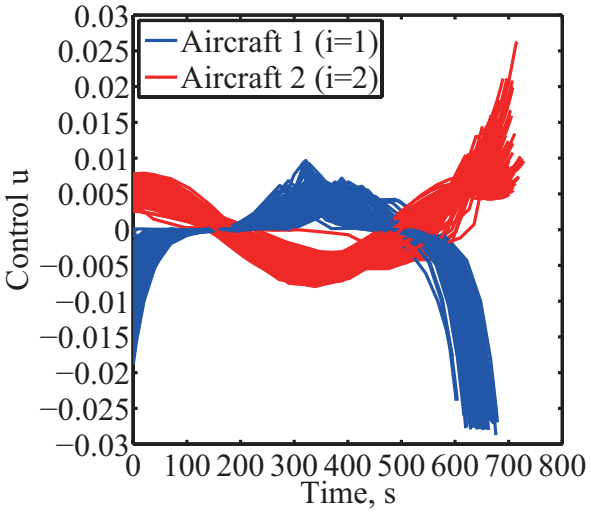
Figure 5-17: Precomputed optimal solutions in case 1 ($N_s = 200$).



(a) Conflict resolution trajectory



(b) Time history of heading angle



(c) Time history of control variable

Figure 5-18: Precomputed optimal solutions in case 2 ($N_s = 200$).

Table 5.9: Maximum and root mean square (RMS) errors of states, controls and terminal times between near-optimal and optimal solutions with different numbers of sample points in case 1

	$N_s = 100$		$N_s = 200$		$N_s = 300$	
Parameters	Maximum	RMS	Maximum	RMS	Maximum	RMS
x_1, nmi	2.31×10^{-3}	4.51×10^{-4}	1.31×10^{-3}	2.37×10^{-4}	1.95×10^{-3}	2.50×10^{-4}
y_1, nmi	1.71×10^{-2}	3.86×10^{-3}	1.01×10^{-2}	2.16×10^{-3}	1.04×10^{-2}	1.66×10^{-3}
ψ_1	1.52×10^{-2}	3.20×10^{-3}	7.64×10^{-3}	1.54×10^{-3}	1.10×10^{-2}	1.60×10^{-3}
x_2, nmi	2.39×10^{-1}	3.20×10^{-2}	5.47×10^{-2}	1.58×10^{-2}	8.87×10^{-2}	1.52×10^{-2}
y_2, nmi	1.43×10^{-1}	1.67×10^{-2}	2.05×10^{-2}	3.68×10^{-3}	9.31×10^{-3}	2.57×10^{-3}
ψ_2	7.30×10^{-2}	1.13×10^{-2}	2.18×10^{-2}	7.87×10^{-3}	2.34×10^{-2}	4.54×10^{-3}
x_3, nmi	1.26×10^{-1}	2.31×10^{-2}	4.99×10^{-2}	1.17×10^{-2}	3.85×10^{-2}	7.86×10^{-3}
y_3, nmi	2.72×10^{-1}	3.83×10^{-2}	5.01×10^{-2}	8.88×10^{-3}	3.43×10^{-2}	7.01×10^{-3}
ψ_3	7.48×10^{-2}	1.28×10^{-2}	2.03×10^{-2}	5.68×10^{-3}	1.74×10^{-2}	3.80×10^{-3}
u_1	1.88×10^{-3}	5.01×10^{-4}	9.68×10^{-4}	2.01×10^{-4}	1.28×10^{-3}	2.16×10^{-4}
u_2	1.78×10^{-2}	2.04×10^{-3}	2.02×10^{-3}	3.40×10^{-4}	1.02×10^{-3}	2.74×10^{-4}
u_3	1.38×10^{-2}	1.54×10^{-3}	9.69×10^{-4}	1.70×10^{-4}	7.73×10^{-4}	1.56×10^{-4}
t_{f1}, s	1.28×10^{-1}	3.51×10^{-2}	1.21×10^{-1}	2.55×10^{-2}	1.24×10^{-1}	2.23×10^{-2}
t_{f2}, s	12.93	3.26	6.09	1.10	4.21	8.33×10^{-1}
t_{f3}, s	11.76	3.32	5.01	1.00	4.11	8.66×10^{-1}

Table 5.10: Maximum and root mean square (RMS) errors of states, controls and terminal times between near-optimal and optimal solutions with different numbers of sample points in case 2

	$N_s = 100$		$N_s = 200$		$N_s = 300$	
Parameters	Maximum	RMS	Maximum	RMS	Maximum	RMS
x_1, nmi	4.07×10^{-1}	1.02×10^{-1}	1.02×10^{-1}	2.97×10^{-2}	6.05×10^{-2}	1.82×10^{-2}
y_1, nmi	1.17	2.35×10^{-1}	1.76×10^{-1}	4.71×10^{-2}	1.27×10^{-1}	3.77×10^{-2}
ψ_1	1.93×10^{-1}	5.20×10^{-2}	4.52×10^{-2}	1.39×10^{-2}	3.74×10^{-2}	7.44×10^{-3}
x_2, nmi	3.08×10^{-1}	6.61×10^{-2}	7.95×10^{-2}	2.33×10^{-2}	1.15×10^{-1}	1.37×10^{-2}
y_2, nmi	5.48×10^{-1}	1.66×10^{-1}	2.06×10^{-1}	4.62×10^{-2}	2.65×10^{-1}	4.25×10^{-2}
ψ_2	1.49×10^{-1}	3.50×10^{-2}	6.82×10^{-2}	1.49×10^{-2}	1.23×10^{-1}	1.68×10^{-2}
u_1	9.18×10^{-3}	2.02×10^{-3}	2.49×10^{-3}	7.29×10^{-4}	1.56×10^{-3}	4.83×10^{-4}
u_2	7.55×10^{-3}	1.39×10^{-3}	1.73×10^{-3}	3.96×10^{-4}	2.31×10^{-3}	3.35×10^{-4}
t_{f1}, s	8.18	1.86	1.22	2.60×10^{-1}	1.15	1.84×10^{-1}
t_{f2}, s	13.83	3.55	2.27	6.48×10^{-1}	1.24	2.27×10^{-1}

Table 5.11: Maximum and root mean square (RMS) errors of states, controls and terminal times between near-optimal and optimal solutions obtained by polynomial chaos kriging, PCE and ordinary kriging in case 1 ($N_s = 200$)

	Polynomial chaos kriging		PCE		Ordinary kriging	
Parameters	Maximum	RMS	Maximum	RMS	Maximum	RMS
x_1, nmi	1.31×10^{-3}	2.37×10^{-4}	1.50×10^{-3}	3.74×10^{-4}	3.05×10^{-3}	6.26×10^{-4}
y_1, nmi	1.01×10^{-2}	2.16×10^{-3}	1.31×10^{-2}	3.16×10^{-3}	3.44×10^{-2}	4.16×10^{-3}
ψ_1	7.64×10^{-3}	1.54×10^{-3}	1.38×10^{-2}	2.65×10^{-3}	8.70×10^{-3}	1.76×10^{-3}
x_2, nmi	5.47×10^{-2}	1.58×10^{-2}	1.15×10^{-1}	2.49×10^{-2}	9.85×10^{-2}	1.65×10^{-2}
y_2, nmi	2.05×10^{-2}	3.68×10^{-3}	1.97×10^{-2}	3.71×10^{-3}	2.51×10^{-2}	3.82×10^{-3}
ψ_2	2.18×10^{-2}	7.87×10^{-3}	2.82×10^{-2}	8.70×10^{-3}	3.66×10^{-2}	5.43×10^{-3}
x_3, nmi	4.99×10^{-2}	1.17×10^{-2}	4.50×10^{-2}	1.28×10^{-2}	7.88×10^{-2}	1.53×10^{-2}
y_3, nmi	5.01×10^{-2}	8.88×10^{-3}	3.86×10^{-2}	8.65×10^{-3}	4.70×10^{-1}	5.17×10^{-2}
ψ_3	2.03×10^{-2}	5.68×10^{-3}	1.71×10^{-2}	5.72×10^{-3}	4.04×10^{-2}	6.62×10^{-3}
u_1	9.68×10^{-4}	2.01×10^{-4}	1.63×10^{-3}	3.34×10^{-4}	8.36×10^{-4}	2.02×10^{-4}
u_2	2.02×10^{-3}	3.40×10^{-4}	2.64×10^{-3}	4.87×10^{-4}	1.56×10^{-3}	3.24×10^{-4}
u_3	9.69×10^{-4}	1.70×10^{-4}	1.41×10^{-3}	2.26×10^{-4}	1.18×10^{-3}	2.19×10^{-4}
t_{f1}, s	1.21×10^{-1}	2.55×10^{-2}	1.41×10^{-1}	3.35×10^{-2}	1.81×10^{-1}	3.28×10^{-2}
t_{f2}, s	6.09	1.10	6.78	2.21	5.16	1.16
t_{f3}, s	5.01	1.00	5.98	2.17	5.84	1.24

Table 5.12: Maximum and root mean square (RMS) errors of states, controls and terminal times between near-optimal and optimal solutions obtained by polynomial chaos kriging, PCE and ordinary kriging in case 2 ($N_s = 200$)

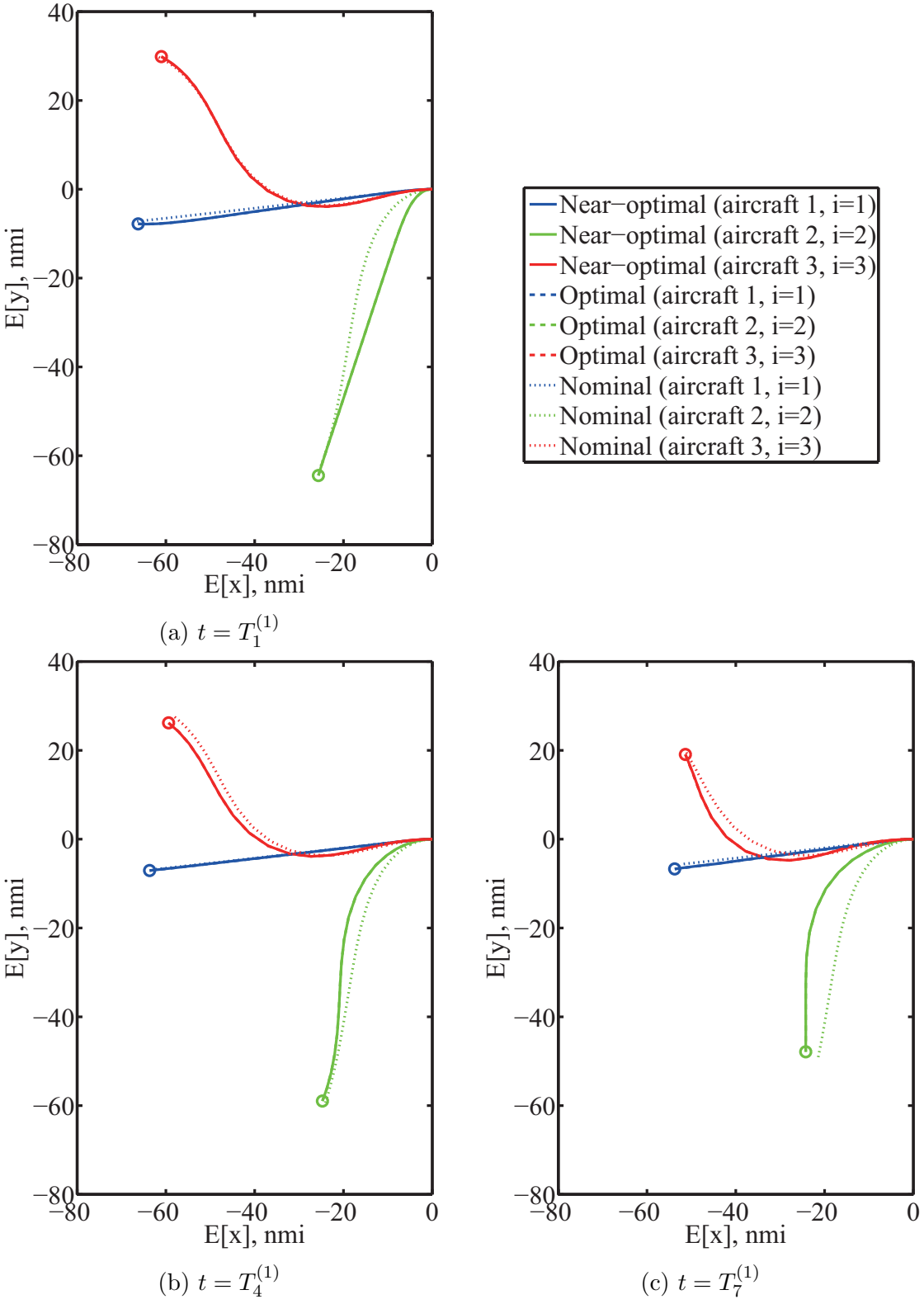
	Polynomial chaos kriging		PCE		Ordinary kriging	
Parameters	Maximum	RMS	Maximum	RMS	Maximum	RMS
x_1, nmi	1.02×10^{-1}	2.97×10^{-2}	9.99×10^{-2}	3.11×10^{-2}	1.57×10^{-1}	2.99×10^{-2}
y_1, nmi	1.76×10^{-1}	4.71×10^{-2}	1.76×10^{-1}	4.86×10^{-2}	2.35×10^{-1}	4.21×10^{-2}
ψ_1	4.52×10^{-2}	1.39×10^{-2}	4.97×10^{-2}	1.50×10^{-2}	6.89×10^{-2}	1.11×10^{-2}
x_2, nmi	7.95×10^{-2}	2.33×10^{-2}	7.66×10^{-2}	2.29×10^{-2}	2.16×10^{-1}	2.93×10^{-2}
y_2, nmi	2.06×10^{-1}	4.62×10^{-2}	2.35×10^{-1}	4.83×10^{-2}	2.49×10^{-1}	5.29×10^{-2}
ψ_2	6.82×10^{-2}	1.49×10^{-2}	6.07×10^{-2}	1.45×10^{-2}	8.07×10^{-2}	1.52×10^{-2}
u_1	2.49×10^{-3}	7.29×10^{-4}	2.43×10^{-3}	7.11×10^{-4}	3.16×10^{-3}	6.35×10^{-4}
u_2	1.73×10^{-3}	3.96×10^{-4}	1.89×10^{-3}	3.60×10^{-4}	4.37×10^{-3}	5.13×10^{-4}
t_{f1}, s	1.22	2.60×10^{-1}	1.33	2.68×10^{-1}	1.17	2.75×10^{-1}
t_{f2}, s	2.27	6.48×10^{-1}	2.37	7.01×10^{-1}	1.97	3.87×10^{-1}

Table 5.13: Root mean square (RMS) errors of expected values of separations ($N_s = 200$) in case 1

Pairs	RMS errors, nmi
$\mathcal{C}_{12}$ (aircraft 1 and 2)	1.31×10^{-2}
$\mathcal{C}_{13}$ (aircraft 1 and 3)	1.56×10^{-2}
$\mathcal{C}_{23}$ (aircraft 2 and 3)	1.07×10^{-2}

Table 5.14: Root mean square (RMS) errors of expected values of separations ($N_s = 200$) in case 2

Pairs	RMS errors, nmi
$\mathcal{C}_{12}$ (aircraft 1 and 2)	4.23×10^{-2}
$\mathcal{C}_{w1}$ (weather and aircraft 1)	2.22×10^{-2}
$\mathcal{C}_{w2}$ (weather and aircraft 2)	4.04×10^{-2}

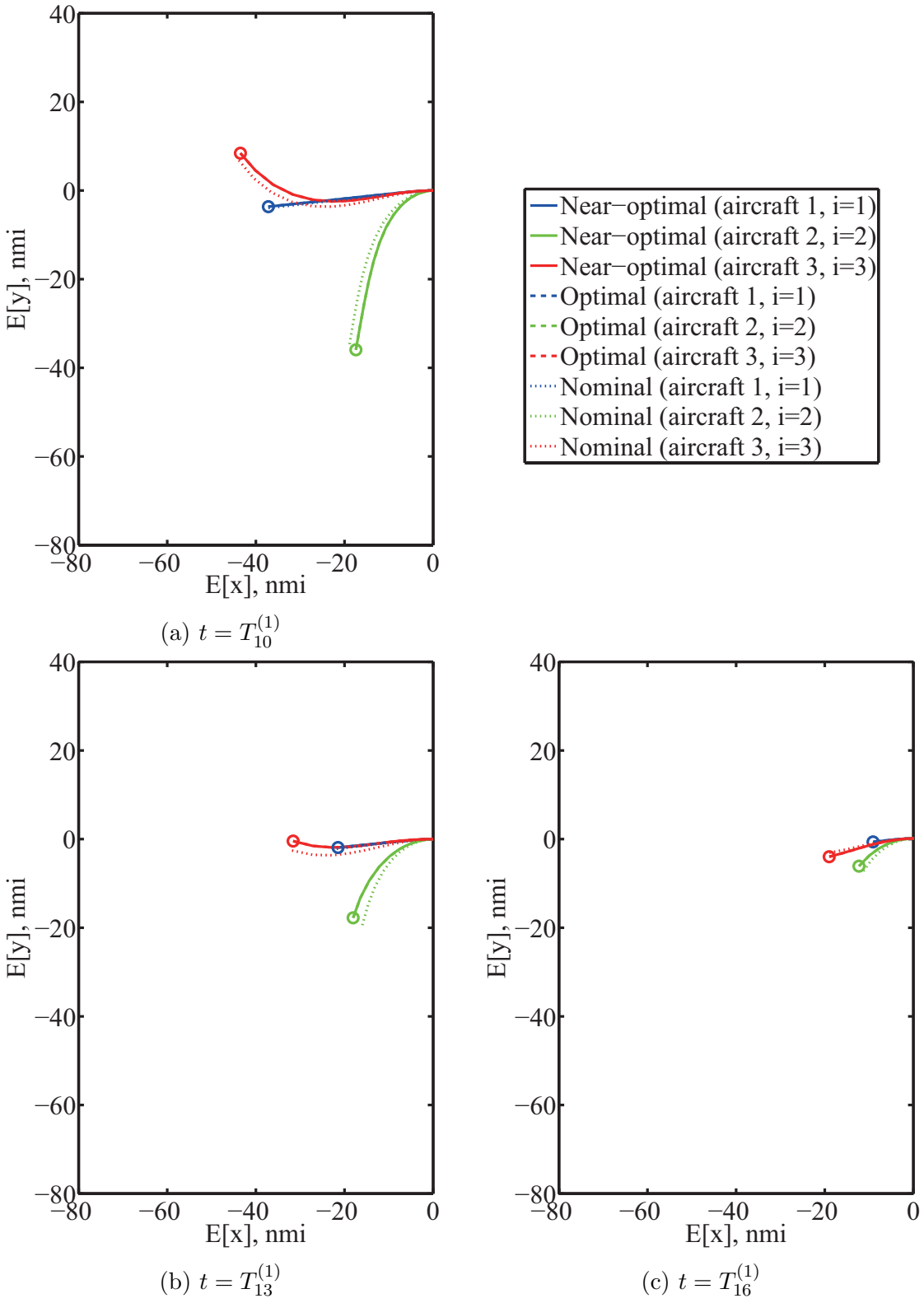


(a) $t = T_1^{(1)}$

(b) $t = T_4^{(1)}$

(c) $t = T_7^{(1)}$

Figure 5-19: Near-optimal, optimal and nominal conflict resolution trajectories in case 1 (at time instant $t = T_1^{(1)}, T_4^{(1)}, T_7^{(1)}$).

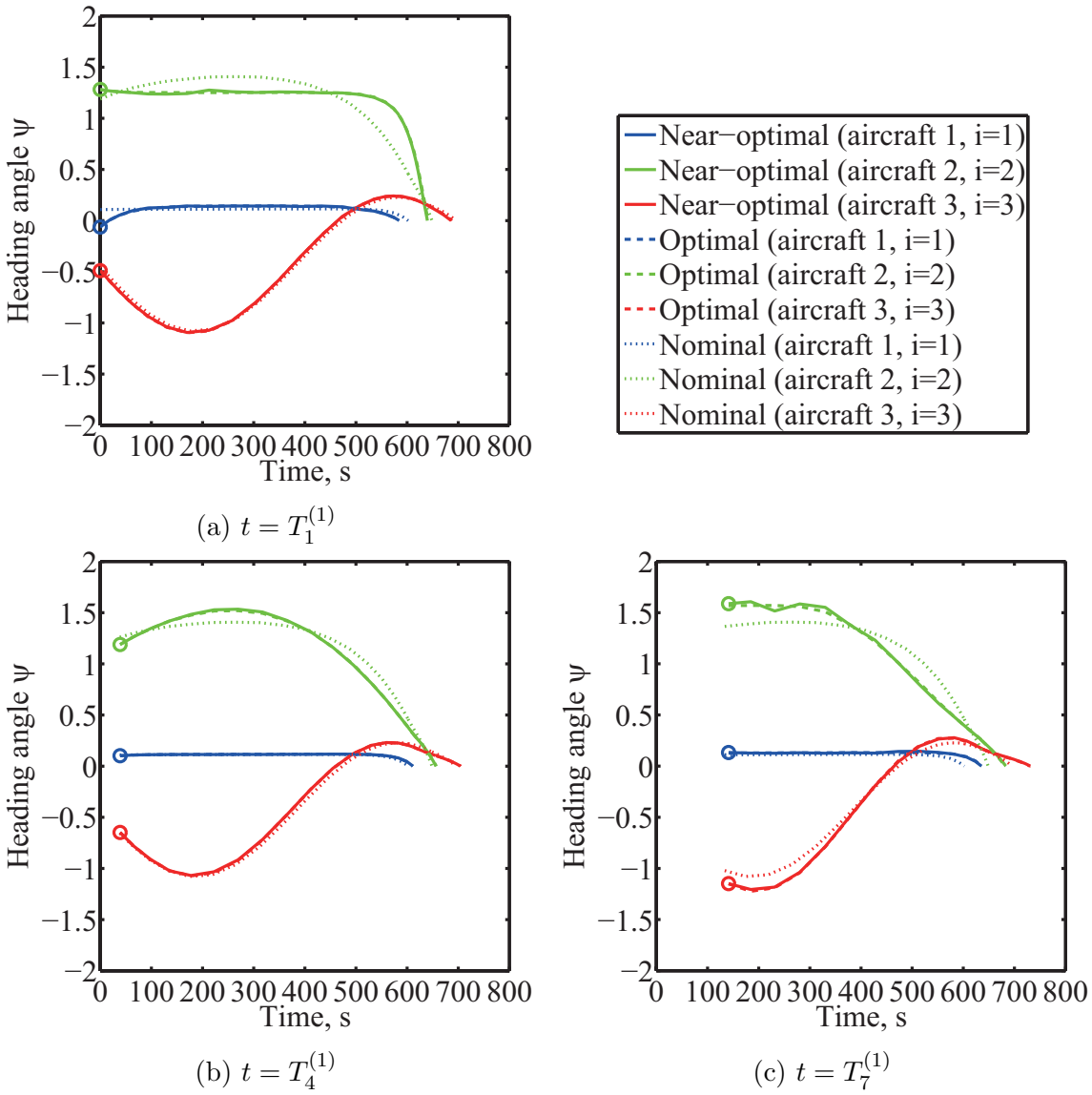


(a) $t = T_{10}^{(1)}$

(b) $t = T_{13}^{(1)}$

(c) $t = T_{16}^{(1)}$

Figure 5-20: Near-optimal, optimal and nominal conflict resolution trajectories in case 1 (at time instant $t = T_{10}^{(1)}, T_{13}^{(1)}, T_{16}^{(1)}$).

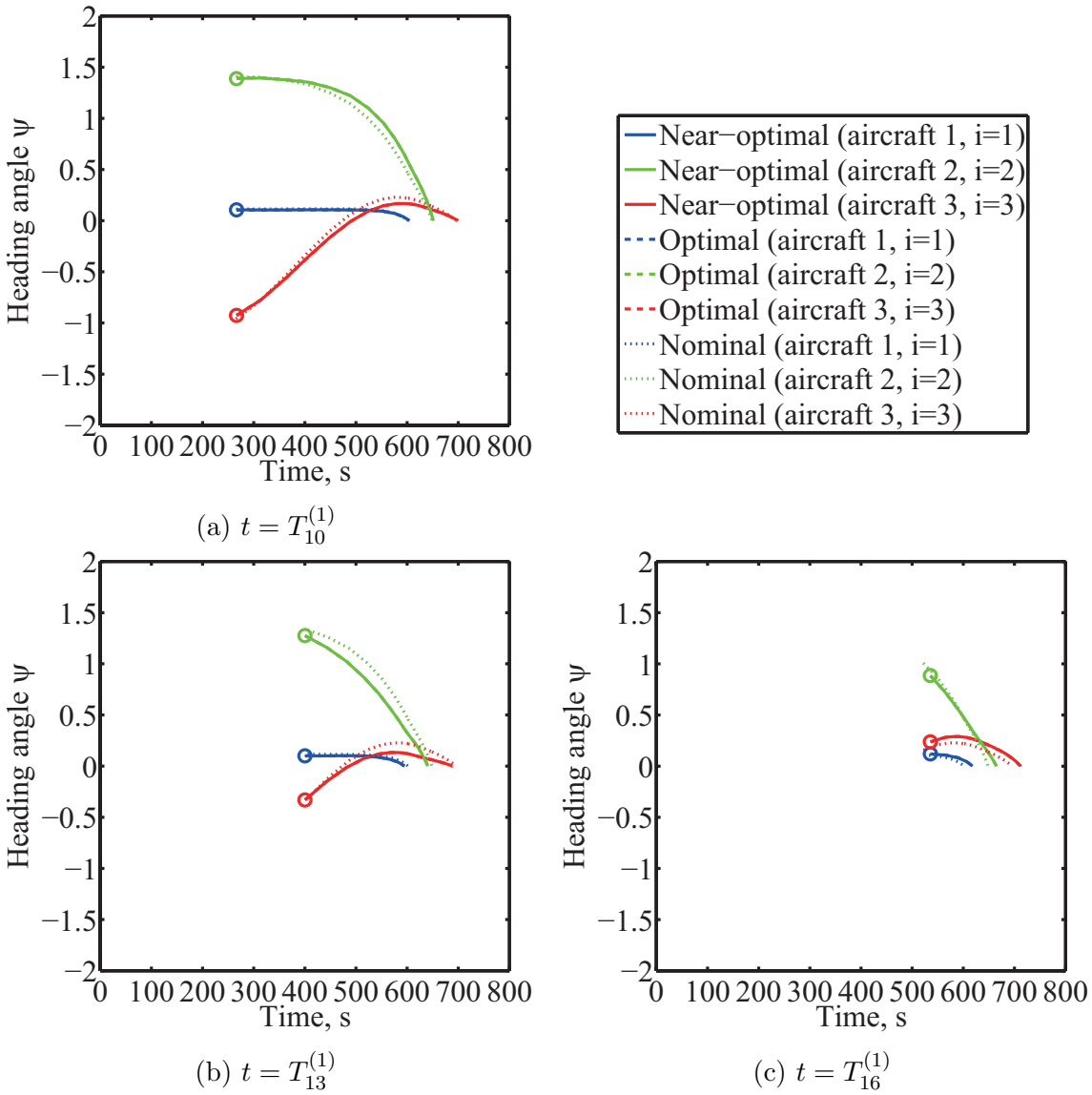


(a) $t = T_1^{(1)}$

(b) $t = T_4^{(1)}$

(c) $t = T_7^{(1)}$

Figure 5-21: Time histories of near-optimal, optimal and nominal heading angles in case 1 (at time instant $t = T_1^{(1)}, T_4^{(1)}, T_7^{(1)}$).

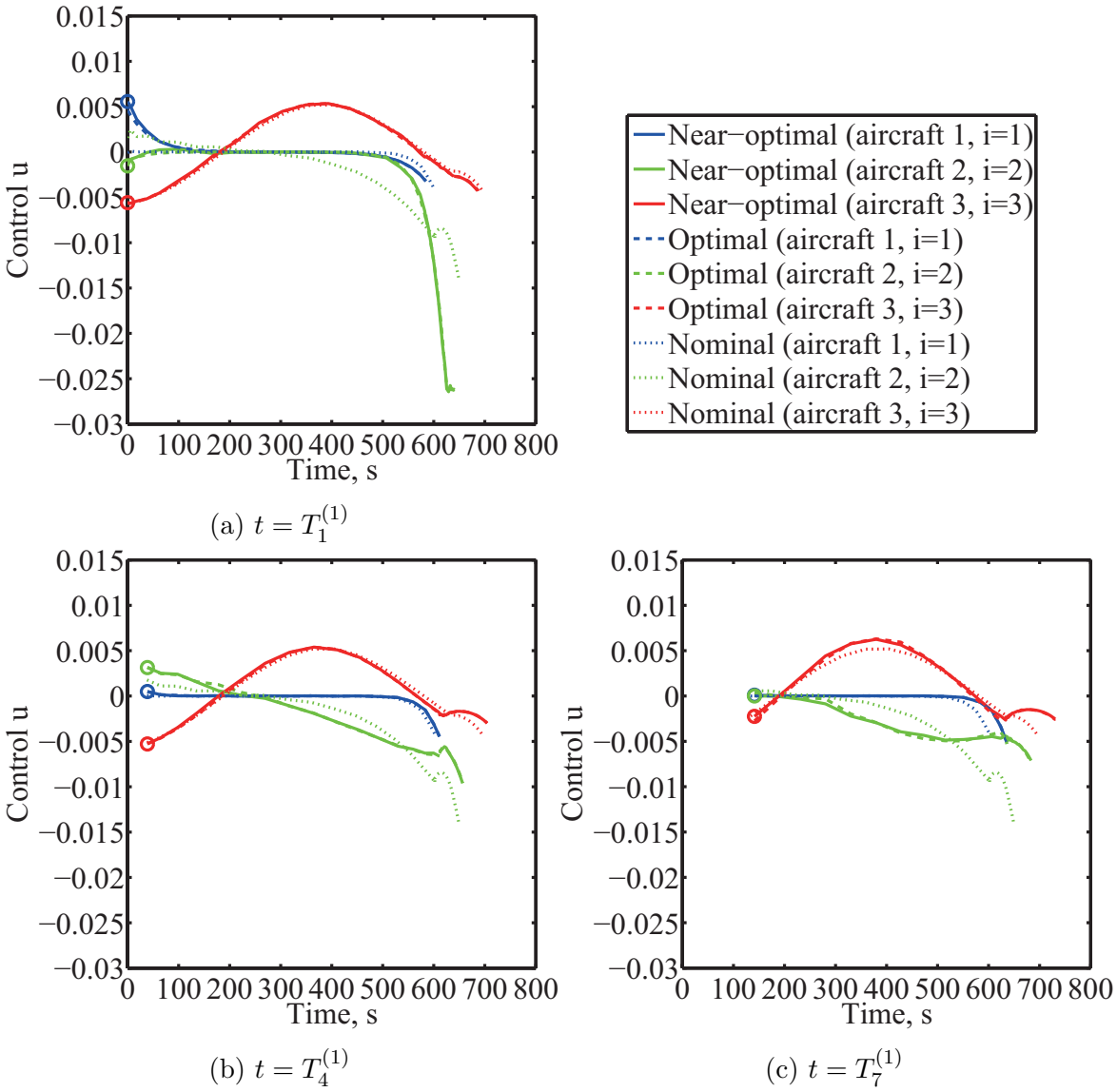


(a) $t = T_{10}^{(1)}$

(b) $t = T_{13}^{(1)}$

(c) $t = T_{16}^{(1)}$

Figure 5-22: Time histories of near-optimal, optimal and nominal heading angles in case 1 (at time instant $t = T_{10}^{(1)}, T_{13}^{(1)}, T_{16}^{(1)}$).

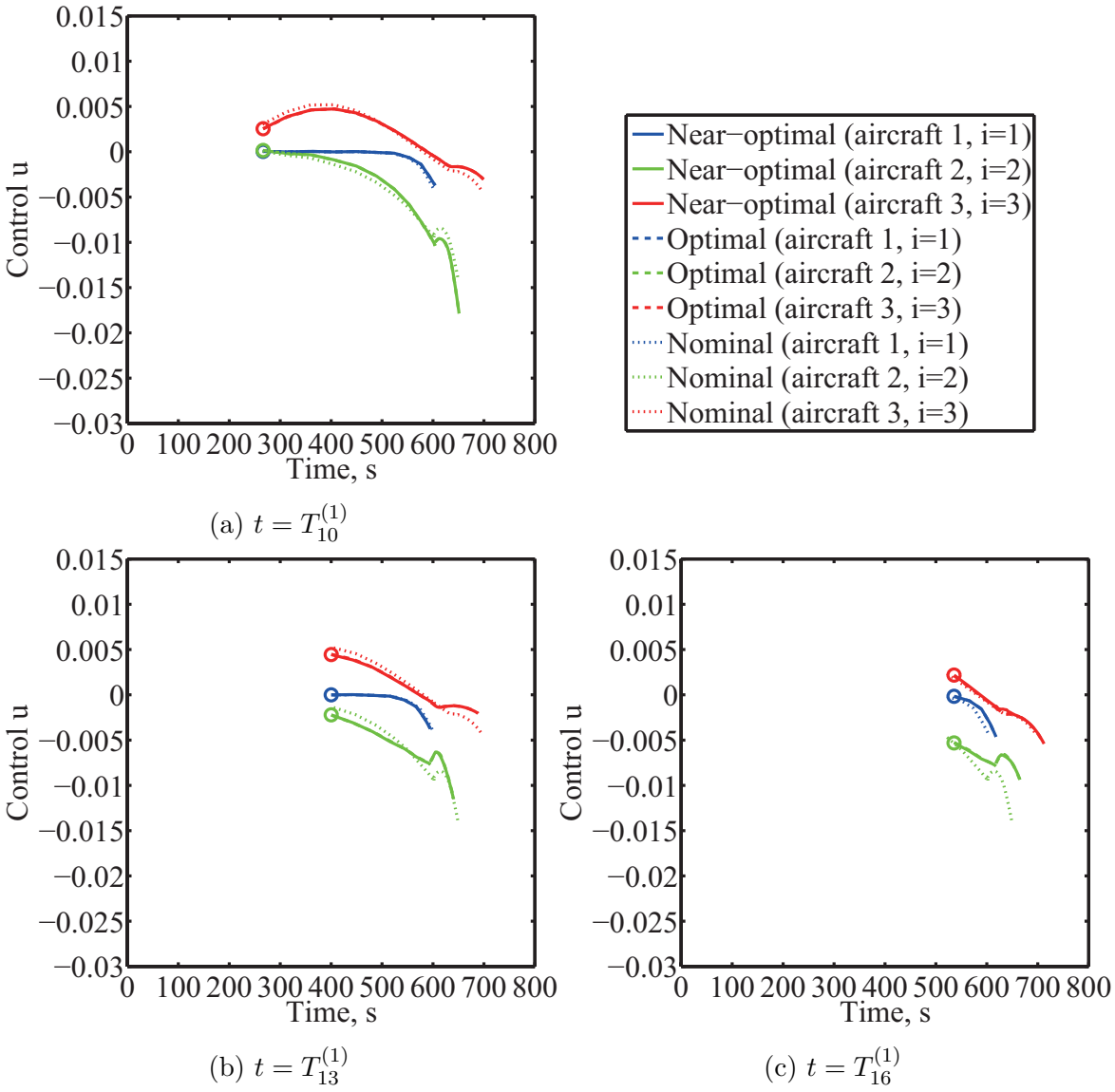


(a) $t = T_1^{(1)}$

(b) $t = T_4^{(1)}$

(c) $t = T_7^{(1)}$

Figure 5-23: Time histories of near-optimal, optimal and nominal controls in case 1 (at time instant $t = T_1^{(1)}, T_4^{(1)}, T_7^{(1)}$).

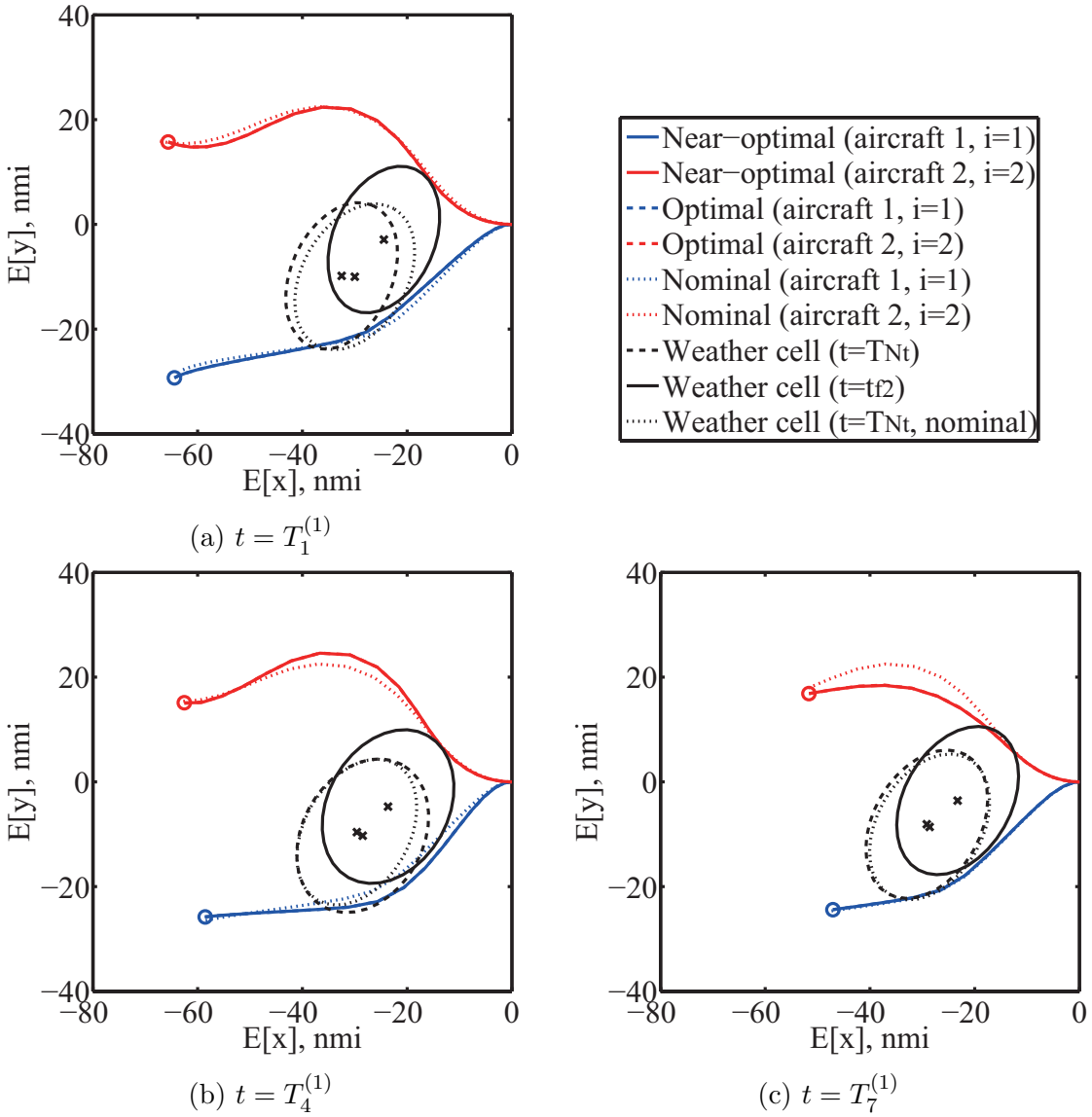


(a) $t = T_{10}^{(1)}$

(b) $t = T_{13}^{(1)}$

(c) $t = T_{16}^{(1)}$

Figure 5-24: Time histories of near-optimal, optimal and nominal controls in case 1 (at time instant $t = T_{10}^{(1)}, T_{13}^{(1)}, T_{16}^{(1)}$).

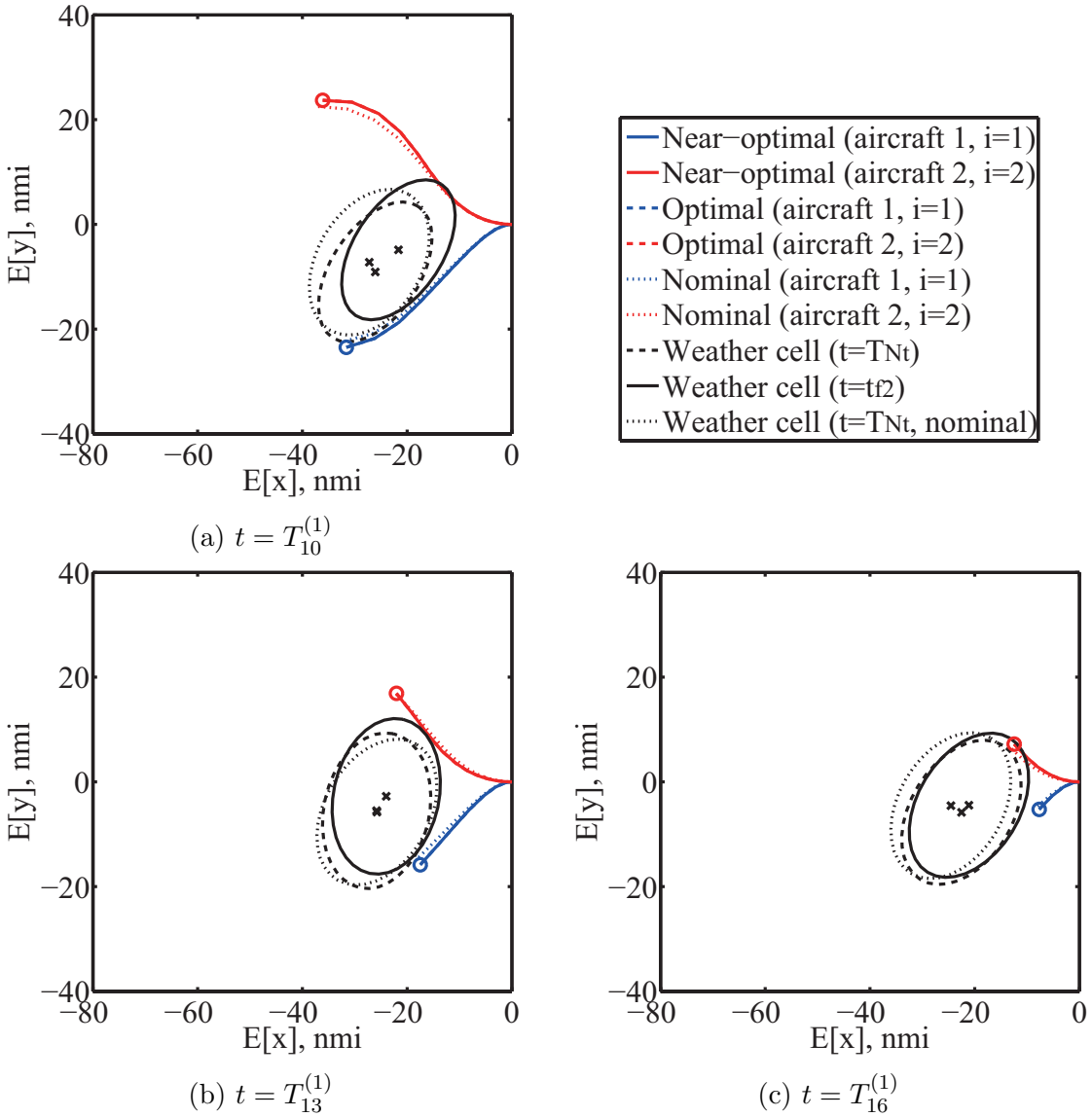


(a) $t = T_1^{(1)}$

(b) $t = T_4^{(1)}$

(c) $t = T_7^{(1)}$

Figure 5-25: Near-optimal, optimal and nominal conflict resolution trajectories in case 2 (at time instant $t = T_1^{(1)}, T_4^{(1)}, T_7^{(1)}$).

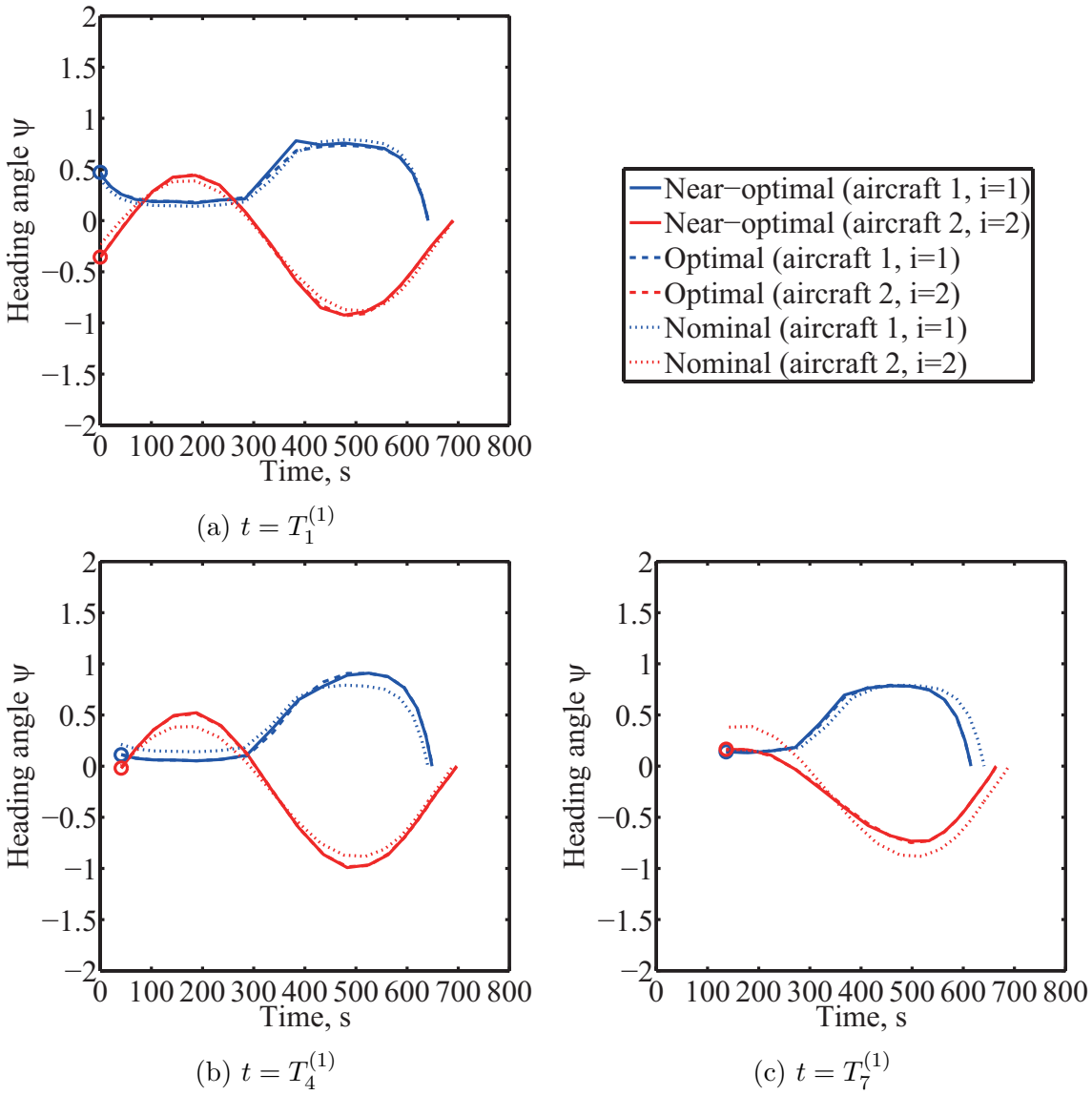


(a) $t = T_{10}^{(1)}$

(b) $t = T_{13}^{(1)}$

(c) $t = T_{16}^{(1)}$

Figure 5-26: Near-optimal, optimal and nominal conflict resolution trajectories in case 2 (at time instant $t = T_{10}^{(1)}, T_{13}^{(1)}, T_{16}^{(1)}$).

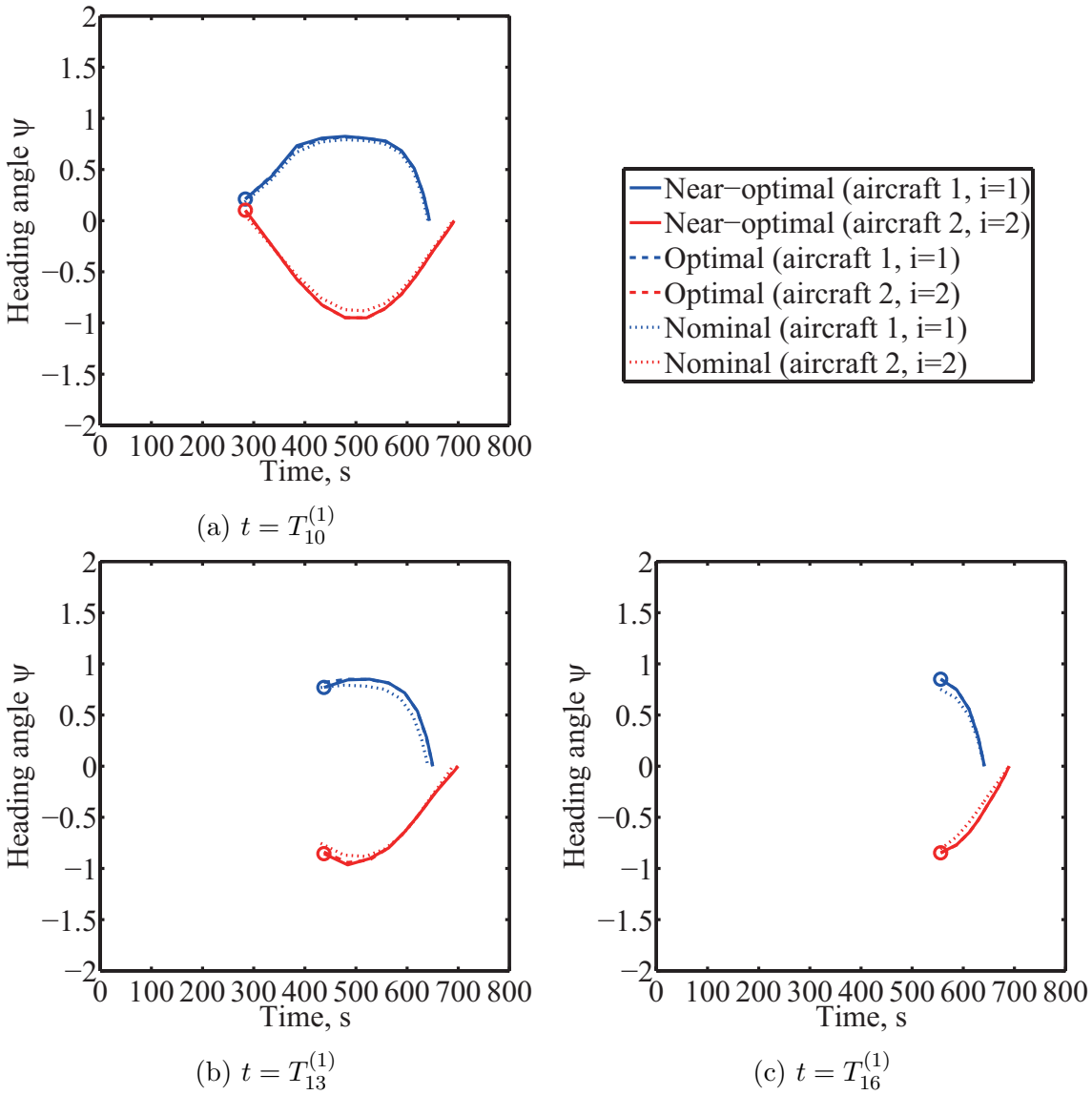


(a) $t = T_1^{(1)}$

(b) $t = T_4^{(1)}$

(c) $t = T_7^{(1)}$

Figure 5-27: Time histories of near-optimal, optimal and nominal heading angles in case 2 (at time instant $t = T_1^{(1)}, T_4^{(1)}, T_7^{(1)}$).

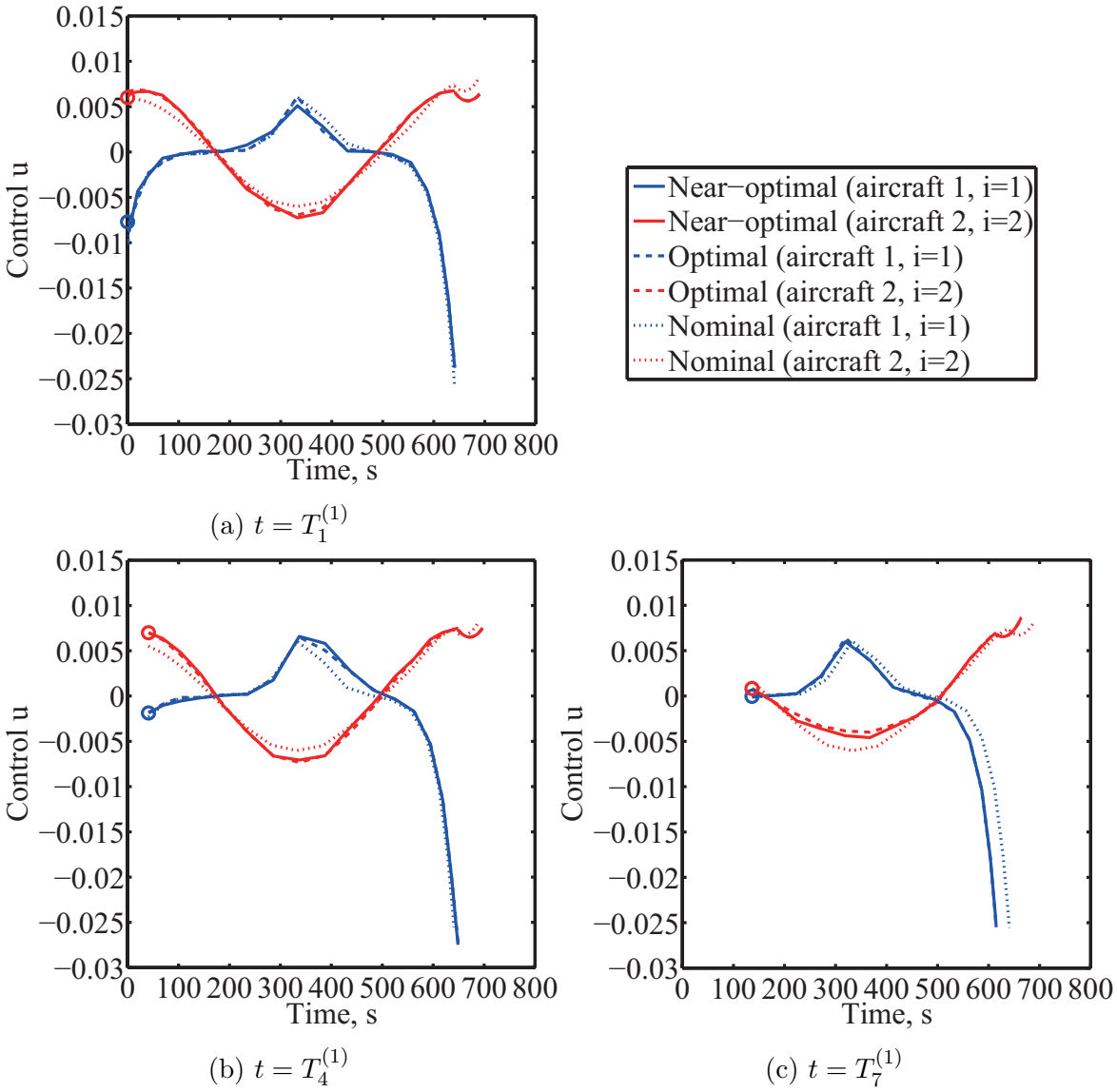


(a) $t = T_{10}^{(1)}$

(b) $t = T_{13}^{(1)}$

(c) $t = T_{16}^{(1)}$

Figure 5-28: Time histories of near-optimal, optimal and nominal heading angles in case 2 (at time instant $t = T_{10}^{(1)}, T_{13}^{(1)}, T_{16}^{(1)}$).

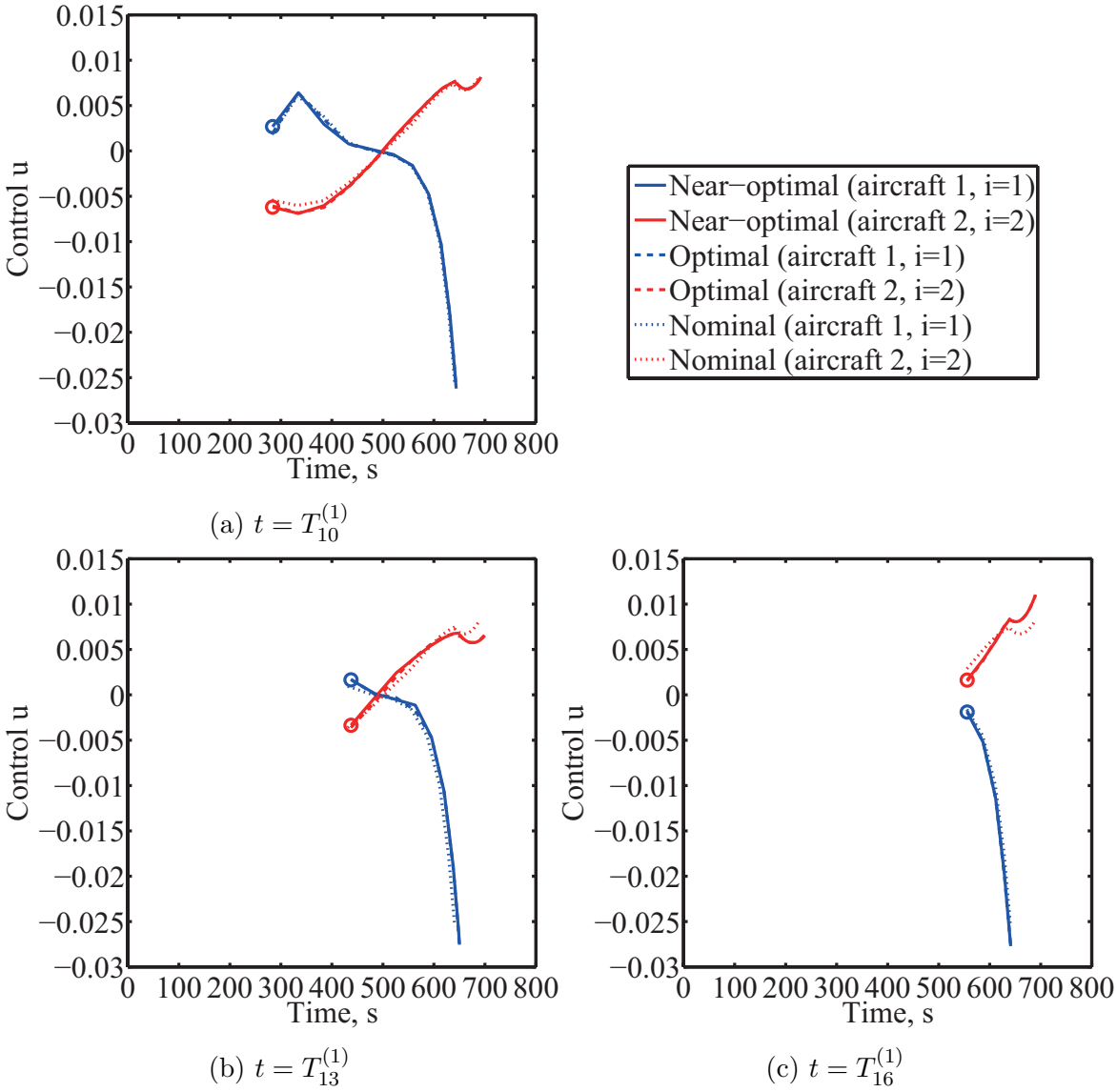


(a) $t = T_1^{(1)}$

(b) $t = T_4^{(1)}$

(c) $t = T_7^{(1)}$

Figure 5-29: Time histories of near-optimal, optimal and nominal controls in case 2 (at time instant $t = T_1^{(1)}, T_4^{(1)}, T_7^{(1)}$).



(a) $t = T_{10}^{(1)}$

(b) $t = T_{13}^{(1)}$

(c) $t = T_{16}^{(1)}$

Figure 5-30: Time histories of near-optimal, optimal and nominal controls in case 2 (at time instant $t = T_{10}^{(1)}, T_{13}^{(1)}, T_{16}^{(1)}$).

Chapter 6

Conclusions

6.1 Research Summary

In this dissertation, we have proposed the probabilistic aircraft conflict detection and resolution algorithms in the presence of uncertainty, especially for two kinds of conflicts: aircraft–aircraft and aircraft–weather conflicts. As the uncertainties during the flight, the wind prediction error, airspeed measurement error and the uncertainty contained in the moving convective weather region were considered. As to the wind uncertainty, the spatially correlated wind error was considered, and the Karhunen–Loève expansion was used to describe the correlated wind error. To detect aircraft midair conflicts, the statistical information of the distances between the pair of aircraft and between aircraft and the convective weather region was accurately computed by employing the polynomial chaos expansion (PCE) algorithm, and the conflict probabilities between the pair of aircraft and between aircraft and the convective weather region were computed by the conflict detection algorithm. Compared with the Monte Carlo method, which is computationally expensive and intractable especially when used within the iterative optimization process, the proposed algorithm based on the PCE method dramatically reduced the computational cost and greatly enhanced the computational efficiency.

In addition, the stochastic optimal control method combining the computationally efficient probabilistic conflict detection algorithm with the pseudospectral method was

applied to the two-dimensional aircraft–aircraft and aircraft–weather conflict resolution problem. The stochastic algorithm could generate the optimal conflict resolution trajectories for the correlated and non-correlated wind error models. By comparing the optimal conflict resolution trajectories for the correlated and non-correlated wind error models, the wind correlation could have significant effects on the separation between aircraft and the optimal conflict resolution trajectory, and the separation between aircraft under the correlated wind error could be smaller than that under the non-correlated error. Using the correlated wind model, we could reduce the conservative separation under the non-correlated wind uncertainty. Accordingly, we can mitigate the air traffic congestion and increase the throughput capacity of the air traffic.

Furthermore, we have proposed the stochastic near-optimal control method for conflict resolution based on the polynomial chaos kriging method. The near-optimal control method was also applied to the aircraft–aircraft and aircraft–weather conflict resolution problem. The polynomial chaos kriging method could accurately construct the surrogate models of the optimal conflict resolution trajectories from a set of the precomputed optimal solutions. Using the surrogate models, the approximate optimal conflict resolution trajectories on the time interval between the current time and the terminal time were obtained as the functions of the current conditions (i.e., the current states, airspeed and parameters for the convective weather cell). Once the surrogate models were constructed offline, the near-optimal conflict resolution trajectories could be obtained online from the surrogate models with the information of the current conditions without actually solving the computationally expensive stochastic optimal control problems. The proposed near-optimal conflict resolution algorithm could have the feature of optimal feedback control. When the states on the precomputed optimal conflict resolution trajectory deviate from the actual states due to the uncertainties, the proposed near-optimal feedback control method can accurately generate the near-optimal trajectory starting from the actual states in real time by using the surrogate models with the information of the actual states without solving another stochastic optimal control problem to obtain the correct optimal trajectory. Comparing the

near-optimal solutions generated from the surrogate models and the optimal solutions obtained by actually solving the stochastic optimal control problems, the near-optimal conflict resolution trajectories were accurately generated and guaranteed to avoid potential conflicts as well as the optimal solutions. Moreover, comparing the surrogate models built with three different surrogate modeling techniques, the polynomial chaos kriging, PCE and ordinary kriging methods, the polynomial chaos kriging method generally performed better than the PCE or kriging method did. Once the surrogate models are constructed and transferred to aircraft, aircraft can determine their own conflict resolution trajectories from the surrogate models based on the information of current conditions obtained from onboard equipment such as ADS-B. Through numerical simulations, the performance and effectiveness of the probabilistic conflict detection and resolution algorithms have been evaluated and demonstrated.

The proposed conflict detection algorithm allows the pilots and air traffic controllers to make a decision in real time on whether potential conflicts will occur within a look ahead time horizon, and the proposed near-optimal conflict resolution algorithm enables the pilots to determine optimal conflict resolution trajectories and maneuvers in real time. Consequently, our proposed probabilistic conflict detection and resolution algorithms can provide the automated advisories and conflict resolution trajectories for the pilots and air traffic controllers and have a potential to ultimately replace the tasks of the air traffic controllers.

6.2 Future Work

The accuracy and convergence rate of the proposed algorithms mainly depend on the structure of the wind model. The proposed algorithms require that the wind model is constructed by using random variables with known probability distributions. In addition, when the number of random variables for the wind model increases, the complexity of the problem increases. Although the wind error was assumed to be time-invariant in this study, the temporal variations in the wind model need to be considered for longer term conflict detection and resolution. If we consider not

only the spatial correlation but the temporal correlation in the wind model, the number of random variables and complexity of the problem will increase. Although the Gaussian random process model was used for the wind uncertainty in this study, a possible future research direction is to demonstrate the validity of the wind model by comparing the real historical aircraft reports and wind forecast data, and it may also be required to deal with other wind models with different probability distributions, unknown arbitrary wind models and the spatiotemporal correlation in the wind model.

In addition, on the basis of an extensive literature review, the wind prediction error was considered as the primary uncertainty during flight compared to other possible uncertainties such as navigation errors and pilots' intents. Although the wind error is one of the most important factors that have significant influence on the accuracy of aircraft trajectory prediction and conflict detection and resolution, an investigation of the relative contributions of the different possible input uncertainties may also be necessary.

Furthermore, another possible research direction is to further extend our proposed algorithms for the aircraft arrival sequencing in terminal areas. The most common conventional sequencing strategy is the *first come*, *first served* rule, however, the optimal sequences (in terms of fuel consumption or air traffic congestion) depend on multiple factors such as initial aircraft positions, aircraft weights, airspeeds and aircraft types. A probabilistic optimal aircraft sequencing algorithm can be investigated in future work.

Appendix A

Karhunen–Loève Expansion

With the Karhunen–Loève (KL) expansion [62, 63], the Gaussian random process $Z(\boldsymbol{x}, \boldsymbol{X})$ with zero mean and a covariance function $C(\boldsymbol{x}, \boldsymbol{x}')$ can be expanded as the following equation:

$$Z(\boldsymbol{x}, X) = \sum_{i=1}^{\infty} \sqrt{\lambda_i} g_i(\boldsymbol{x}) X_i \tag{A.1}$$

where $\boldsymbol{x} \in \mathbb{R}^{N_x}$ is the position vector defined over the domain $\boldsymbol{D}$; X_i is the independent Gaussian random variable; and λ_i and $g_i(\boldsymbol{x})$ are the eigenvalue and eigenfunction of the following integral equation in descending order of the magnitude of the eigenvalue λ_i, respectively:

$$\lambda_i g_i(\boldsymbol{x}) = \int_{\boldsymbol{D}} C(\boldsymbol{x}, \boldsymbol{x}') g_i(\boldsymbol{x}') \mathrm{d}\boldsymbol{x}'$$

The covariance function $C(\boldsymbol{x}, \boldsymbol{x}')$ can be decomposed by the following equation:

$$C(\boldsymbol{x}, \boldsymbol{x}') = \sum_{i=1}^{\infty} \lambda_i g_i(\boldsymbol{x}) g_i(\boldsymbol{x}') \tag{A.2}$$

For computational purpose, the infinite series in Eq. (A.1) is truncated in order to retain a finite number of terms:

$$Z(\boldsymbol{x}, X) = \sum_{i=1}^{N_{\mathrm{KL}}} \sqrt{\lambda_i} g_i(\boldsymbol{x}) X_i$$

where N_{KL} is the number of terms. Accordingly, the covariance function $C(\boldsymbol{x}, \boldsymbol{x}')$ in Eq. (A.2) is given by the following equation:

$$C(\boldsymbol{x}, \boldsymbol{x}') = \sum_{i=1}^{N_{\mathrm{KL}}} \lambda_i g_i(\boldsymbol{x}) g_i(\boldsymbol{x}')$$

We firstly consider the univariate Gaussian random process with zero mean and the following exponential covariance function:

$$C(x, x') = \exp(-\mu_x |x - x'|)$$

where μ_x is the parameter with the same unit as x^{-1} ($\mu_x > 0$), which is referred to the correlation length. It is assumed that the random process is defined over the range $[-T_x, T_x]$, and the eigenvalues and eigenfunctions are the solutions of the following integral equation:

$$\lambda_i g_i(x) = \int_{-T_x}^{T_x} \exp(-\mu_x |x - x'|) g_i(x') \mathrm{d}x' \qquad (i = 1, \ldots, N_{\mathrm{KL}}) \tag{A.3}$$

Eq. (A.3) can be rewritten as the following equation:

$$\lambda_i g_i(x) = \int_{-T_x}^{\bar{x}} \exp(-\mu_x (x - x')) g_i(x') \mathrm{d}x' + \int_{\bar{x}}^{T_x} \exp(\mu_x (x - x')) g_i(x') \mathrm{d}x' \tag{A.4}$$

Differentiating Eq. (A.4) with respect to x, we can obtain the following equation:

$$\lambda_i \frac{\mathrm{d}}{\mathrm{d}x} g_i(x) = -\mu_x \int_{-T_x}^{\bar{x}} \exp(-\mu_x (x - x')) g_i(x') \mathrm{d}x' + \mu_x \int_{\bar{x}}^{T_x} \exp(\mu_x (x - x')) g_i(x') \mathrm{d}x'$$

Differentiating again with respect to x, the following equation can be obtained:

$$\lambda_i \frac{\mathrm{d}^2}{\mathrm{d}x^2} g_i(x) = (-2\mu_x + \mu_x^2 \lambda_i) g_i(x) \tag{A.5}$$

The following new variable ω_i is introduced:

$$\omega_i^2 = \frac{2\mu_x - \mu_x^2 \lambda_i}{\lambda_i} \tag{A.6}$$

Using ω_i in Eq. (A.6), Eq. (A.5) can be rewritten as the following equation:

$$\frac{\mathrm{d}^2}{\mathrm{d}x^2} g_i(x) = \omega_i^2 g_i(x) \qquad (-T_x \leq x \leq T_x) \tag{A.7}$$

The boundary conditions associated with Eq. (A.7) can be given by the following equations:

$$\mu_x g_i(T_x) + \frac{\mathrm{d}}{\mathrm{d}x} g_i(T_x) = 0 \tag{A.8}$$

$$\mu_x g_i(-T_x) - \frac{\mathrm{d}}{\mathrm{d}x} g_i(-T_x) = 0 \tag{A.9}$$

It can be assumed that the solution to Eq. (A.7) has the following form:

$$g_i(x) = A\cos(\omega_i x) + B\sin(\omega_i x)$$

where $A \in \mathbb{R}$ and $B \in \mathbb{R}$ are the unknown variables. Applying the boundary conditions given by Eqs. (A.8) and (A.9), we can obtain the following transcendental equations:

$$\mu_x - \omega_i \tan(\omega_i T_x) = 0 \qquad \text{if } i \text{ is odd} \tag{A.10}$$

$$\omega_i + \mu_x \tan(\omega_i T_x) = 0 \qquad \text{if } i \text{ is even} \tag{A.11}$$

ω_i is the solution to either of Eqs. (A.10) and (A.11), and the resulting eigenfunction $g_i(x)$ is given by either of the following equations:

$$g_i(x) = \frac{\cos(\omega_i x)}{\sqrt{T_x + \frac{\sin(2\omega_i T_x)}{2\omega_i}}} \qquad \text{if } i \text{ is odd}$$

$$g_i(x) = \frac{\sin(\omega_i x)}{\sqrt{T_x - \frac{\sin(2\omega_i T_x)}{2\omega_i}}} \qquad \text{if } i \text{ is even}$$

The corresponding eigenvalue is given by the following equation:

$$\lambda_i = \frac{2\mu_x}{\omega_i^2 + \mu_x^2}$$

Again, it should be noted that λ_i and $g_i(x)$ are given in descending order of the magnitude of λ_i.

We further consider the two-dimensional Gaussian random process with the following covariance function:

$$C((x,y),(x',y')) = \exp(-\mu_x|x-x'|)\exp(-\mu_y|y-y'|)$$

where μ_x and μ_y are the parameters with the same units as x^{-1} and y^{-1}, respectively ($\mu_x > 0$ and $\mu_y > 0$). We would like to note that this form of the Gaussian random process is equivalent to the spatially correlated wind error model considered in this study. The random process is defined over the domain $\boldsymbol{D}$ ($x \in [-T_x, T_x]$ and $y \in [-T_y, T_y]$), and the eigenvalues and eigenfunctions are the solutions of the following integral equation:

$$\lambda_i g_i(x,y) = \int_{\boldsymbol{D}} \exp(-\mu_x|x-x'|)\exp(-\mu_y|y-y'|)g_i(x',y')\mathrm{d}x'\mathrm{d}y' \qquad (i = 1,\ldots,N_{\mathrm{KL}}) \tag{A.12}$$

It is assumed that λ_i and $g_i(x,y)$ have the following forms, respectively:

$$\lambda_i = \lambda_{x,j}\lambda_{y,k} \tag{A.13}$$

$$g_i(x,y) = g_{x,j}(x)g_{y,k}(y) \tag{A.14}$$

The combination of (j,k) is unique for each i, and the number of combinations is N_{KL}. Eq. (A.12) can be rewritten as the following equation:

$$\lambda_{x,j}\lambda_{y,k}g_{x,j}(x)g_{y,k}(y) = \int_{-T_x}^{T_x} \exp(-\mu_x|x-x'|)g_{x,j}(x')\mathrm{d}x' \int_{-T_y}^{T_y} \exp(-\mu_y|y-y'|)g_{y,k}(y')\mathrm{d}y' \tag{A.15}$$

The solution to Eq. (A.15) is the product of the two individual solutions to the

following equations:

$$\lambda_{x,j} g_{x,j}(x) = \int_{-T_x}^{T_x} \exp(-\mu_x |x - x'|) g_{x,j}(x') \mathrm{d}x' \tag{A.16}$$

$$\lambda_{y,k} g_{y,k}(y) = \int_{-T_y}^{T_y} \exp(-\mu_y |y - y'|) g_{y,k}(y') \mathrm{d}y' \tag{A.17}$$

Both Eqs. (A.16) and (A.17) are equivalent to Eq. (A.3) for the univariate Gaussian random process. Therefore, by differentiating Eqs. (A.16) and (A.17) twice with respect to x and y, respectively, we can obtain the following eigenvalues:

$$\lambda_{x,j} = \frac{2\mu_x}{\omega_{x,j}^2 + \mu_x^2} \tag{A.18}$$

$$\lambda_{y,k} = \frac{2\mu_y}{\omega_{y,k}^2 + \mu_y^2} \tag{A.19}$$

The associated eigenfunctions $g_{x,j}(x)$ and $g_{y,k}(y)$ are given by the following equations:

$$g_{x,j}(x) = \frac{\cos(\omega_{x,j} x)}{\sqrt{T_x + \frac{\sin(2\omega_{x,j} T_x)}{2\omega_{x,j}}}} \qquad \text{if } j \text{ is odd} \tag{A.20}$$

$$g_{x,j}(x) = \frac{\sin(\omega_{x,j} x)}{\sqrt{T_x - \frac{\sin(2\omega_{x,j} T_x)}{2\omega_{x,j}}}} \qquad \text{if } j \text{ is even} \tag{A.21}$$

$$g_{y,k}(y) = \frac{\cos(\omega_{y,k} y)}{\sqrt{T_y + \frac{\sin(2\omega_{y,k} T_y)}{2\omega_{y,k}}}} \qquad \text{if } k \text{ is odd} \tag{A.22}$$

$$g_{y,k}(y) = \frac{\sin(\omega_{y,k} y)}{\sqrt{T_y - \frac{\sin(2\omega_{y,k} T_y)}{2\omega_{y,k}}}} \qquad \text{if } k \text{ is even} \tag{A.23}$$

In Eqs. (A.18)–(A.23), $\omega_{x,j}$ and $\omega_{y,k}$ are the solutions to the following transcendental equations:

$$\mu_x - \omega_{x,j} \tan(\omega_{x,j} T_x) = 0 \qquad \text{if } j \text{ is odd}$$

$$\omega_{x,j} + \mu_x \tan(\omega_{x,j} T_x) = 0 \qquad \text{if } j \text{ is even}$$

$$\mu_y - \omega_{y,k} \tan(\omega_{y,k} T_y) = 0 \qquad \text{if } k \text{ is odd}$$

$$\omega_{y,k} + \mu_y \tan(\omega_{y,k} T_y) = 0 \qquad \text{if } k \text{ is even}$$

Therefore, λ_i and $g_i(x, y)$ in Eqs. (A.13) and (A.14) can be obtained by Eqs. (A.18)–(A.23). It should be noted that when $\mu_x = \mu_y$, λ_i and $g_i(x, y)$ in Eqs. (A.13) and (A.14) are given by the following equations:

$$\lambda_i = \lambda_{x,j}\lambda_{y,k} = \lambda_{x,k}\lambda_{y,j}$$
$$g_i(x, y) = \frac{1}{\sqrt{2}}\left(g_{x,j}(x)g_{y,k}(y) + g_{x,k}(x)g_{y,j}(y)\right)$$

Again, note that λ_i and $g_i(x, y)$ are given in descending order of the magnitude of λ_i.

Appendix B

Orthogonal Polynomials

There are several types of the orthogonal polynomials [145] such as Hermite and Legendre polynomials. Since only the Hermite polynomials are used in this study, we discuss the Hermite polynomials. The Hermite polynomial $He_n(x)$ ($n \in \mathbb{N}$) is given by the following differential equation:

$$\frac{\mathrm{d}^2}{\mathrm{d}x^2}He_n(x) - x\frac{\mathrm{d}}{\mathrm{d}x}He_n(x) + nHe_n(x) = 0 \tag{B.1}$$

$He_n(x)$ is orthogonal with respect to the weight function $w(x)$:

$$\int He_m(x)He_n(x)w(x)\mathrm{d}x = n!\delta_{mn}$$

where δ_{mn} is the Kronecker delta function that takes 1 if $m = n$ and 0 otherwise; and the weight function $w(x)$ is given by the standard Gaussian probability density function:

$$w(x) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x^2}{2}\right)$$

When x is the standard Gaussian random variable, $He_n(x)$ satisfies the following condition:

$$\mathrm{E}[He_m(x)He_n(x)] = n!\delta_{mn}$$

It should be noted that in this study we use the orthonormal Hermite polynomial given by the following equation:

$$he_n(x) = \frac{1}{\sqrt{n!}} He_n(x)$$

where $he_n(x)$ denotes the orthonormal Hermite polynomial.

In practice, $He_n(x)$ can be generated by the following recurrence relation:

$$He_0(x) = 1 \tag{B.2}$$

$$He_{n+1}(x) = xHe_n(x) - nHe_{n-1}(x) \tag{B.3}$$

Note that Eqs. (B.2) and (B.3) are equivalent to Eq. (B.1). By using the recurrence relation in Eqs. (B.2) and (B.3), the first six Hermite polynomials are given by the following equations:

$$\begin{aligned}
He_0(x) &= 1 \\
He_1(x) &= x \\
He_2(x) &= x^2 - 1 \\
He_3(x) &= x^3 - 3x \\
He_4(x) &= x^4 - 6x^2 + 3 \\
He_5(x) &= x^5 - 10x^3 + 15x
\end{aligned}$$

Table B.1: Collocation points and weights

Number of points	Collocation points	Weights
1	0	1
2	± 1	0.5
3	(± 1.7321, 0)	(0.1667, 0.6667)
4	(± 2.3344, ± 0.7420)	(0.0459, 0.4541)
5	(± 2.8570, ± 1.3556, 0)	(0.0113, 0.2221, 0.5333)

Accordingly, the first six orthonormal Hermite polynomials are also given by the following equations:

$$
\begin{aligned}
he_0(x) &= 1 \\
he_1(x) &= x \\
he_2(x) &= \frac{1}{\sqrt{2}}(x^2 - 1) \\
he_3(x) &= \frac{1}{\sqrt{3!}}(x^3 - 3x) \\
he_4(x) &= \frac{1}{\sqrt{4!}}(x^4 - 6x^2 + 3) \\
he_5(x) &= \frac{1}{\sqrt{5!}}(x^5 - 10x^3 + 15x)
\end{aligned}
$$

In addition, in the Gaussian quadrature rule, n collocation points are the roots of the nth order orthogonal polynomials, and the associated weights are given by the following equation:

$$\alpha^{(j)} = \frac{n!}{n^2(He_{n-1}(x^{(j)}))^2} \qquad (j = 1, \ldots, n)$$

where $x^{(j)}$ is the jth collocation point; and $\alpha^{(j)}$ is the associated weight. $\alpha^{(j)}$ satisfies the following condition:

$$\sum_{j=1}^{n} \alpha^{(j)} = 1$$

Table B.1 shows the collocation points and corresponding weights ($n = 1, 2, 3, 4, 5$).

Bibliography

[1] International Air Transport Association, "Annual Review 2014," 2014.

[2] International Civil Aviation Organization, "Global Air Transport Outlook to 2030 and Trends to 2040," Cir 333, 2013.

[3] International Civil Aviation Organization, "Global Air Traffic Management Operational Concept," Doc 9854AN/458, 2005.

[4] Federal Aviation Administration Joint Planning and Development Office, "Concept of Operations for the Next Generation Air Transportation System Ver. 3.2," 2010.

[5] SESAR Joint Undertaking, "European ATM Master Plan Edition 2," 2012.

[6] Ministry of Land, Infrastructure, Transport and Tourism Study Group for the Future Air Traffic Systems, "Long-term Vision for the Future Air Traffic Systems," 2010.

[7] Kuchar, J., and Yang, L., "A Review of Conflict Detection and Resolution Methods," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 1, No. 4, 2000, pp. 179–189.

[8] Eby, M. S., "A Self-Organizational Approach for Resolving Air Traffic Conflicts," *The Lincoln Laboratory Journal*, Vol. 7, No. 2, 1994, pp. 239–254.

[9] Tomlin, C., Pappas, G. J., and Sastry, S., "Conflict Resolution for Air Traffic Management: A Study in Multiagent Hybrid Systems," *IEEE Transactions on Automatic Control*, Vol. 43, No. 4, 1998, pp. 509–521.

[10] Bicchi, A., and Pallottino, L., "On Optimal Cooperative Conflict Resolution for Air Traffic Management Systems," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 1, No. 4, 2000, pp. 221–231.

[11] Frazzoli, E., Mao, Z. -H., Oh, J. -H, and Feron, E., "Resolution of Conflicts Involving Many Aircraft via Semidefinite Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 1, 2001, pp. 79–86.

[12] Pallottino, L., Feron, E. M., and Bicchi, A., "Conflict Resolution Problems for Air Traffic Management Systems Solved with Mixed Integer Programming," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 3, No. 1, 2002, pp. 3–11.

[13] Richards, A., and How, J. P., "Aircraft Trajectory Planning with Collision Avoidance Using Mixed Integer Linear Programming," *American Control Conference*, Anchorage, AK, USA, 2002, pp. 1936–1941.

[14] Omer, J., and Farges, J. -L., "Hybridization of Nonlinear and Mixed-Integer Linear Programming for Aircraft Separation With Trajectory Recovery," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 14, No. 3, 2013, pp. 1218–1230.

[15] Kuwata, Y., Richards, A., Schouwenaars, T., and How, J. P., "Distributed Robust Receding Horizon Control for Multivehicle Guidance," *IEEE Transactions on Control Systems Technology*, Vol. 15, No. 4, 2007, pp. 627–641.

[16] Yokoyama, N., "Model Predictive Control for Parallel Planning of Conflict-Free Trajectories for Multiple Aircraft," *Transactions of the Japan Society for Aeronautical and Space Sciences*, Vol. 57, No. 6, 2014, pp. 352–360

[17] Menon, P. K., Sweriduk, G. D., and Sridhar, B., "Optimal Strategies for Free-Flight Air Traffic Conflict Resolution," *Journal of Guidance, Control, and Dynamics*, Vol. 22, No. 2, 1999, pp. 202–211.

[18] Hu, J., Prandini, M., and Sastry, S., "Optimal Coordinated Maneuvers for Three-Dimensional Aircraft Conflict Resolution," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 5, 2002, pp. 888–900.

[19] Raghunathan, A. U., Gopal, V., Subramanian, D., Biegler, L. T., and Samad, T., "Dynamic Optimization Strategies for Three-Dimensional Conflict Resolution of Multiple Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 4, 2004, pp. 586–594.

[20] Dimarogonas, D. V., Loizou, S. G., Kyriakopoulos, K. J., and Zavlanos, M. M., "A Feedback Stabilization and Collision Avoidance Scheme for Multiple Independent Non-Point Agents," *Automatica*, Vol. 42, No. 2, 2006, pp. 229–243.

[21] Hwang, I., Kim, J., and Tomlin, C., "Protocol-Based Conflict Resolution for Air Traffic Control," *Air Traffic Control Quarterly*, Vol. 15, No. 1, 2007, pp. 1–34.

[22] Pallottino, L. , Scordio, V. G., Bicchi, A., and Frazzoli, E., "Decentralized Cooperative Policy for Conflict Resolution in Multivehicle Systems," *IEEE Transactions on Robotics*, Vol. 23, No. 6, 2007, pp. 1170–1183.

[23] Erzberger, H., Paielli, R., Isaacson, D., and Eshow, M., "Conflict Detection and Resolution in the Presence of Prediction Error," *1st USA/Europe Air Traffic Management R&D Seminar*, Saclay, France, 1997.

[24] Paielli, R., and Erzberger, H., "Conflict Probability Estimation for Free Flight," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 3, 1997, pp. 588–596.

[25] Yang, L., and Kuchar, J., "Prototype Conflict Alerting System for Free Flight," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 4, 1997, pp. 768–773.

[26] Yang, L., Yang, J. H., Kuchar, J., and Feron, E., "A Real-Time Monte Carlo Implementation for Computing Probability of Conflict," *AIAA Guidance, Navigation, and Control Conference*, Providence, RI, USA, AIAA Paper 2004-4876, 2004.

[27] Lauderdale, T. A., "Probabilistic Conflict Detection for Robust Detection and Resolution," *12th AIAA Aviation Technology, Integration, and Operations Conference*, Indianapolis, IN, USA, AIAA Paper 2012-5643, 2012.

[28] Prandini, M., Hu, J., Lygeros, J., and Sastry, S., "A Probabilistic Approach to Aircraft Conflict Detection," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 1, No. 4, 2000, pp. 199–220.

[29] Hu, J., Prandini, M., and Sastry, S., "Aircraft Conflict Prediction in the Presence of a Spatially Correlated Wind Field," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 6, No. 3, 2005, pp. 326–340.

[30] Chaloulos, G., and Lygeros, J., "Effect of Wind Correlation on Aircraft Conflict Probability," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 6, 2007, pp. 1742–1752.

[31] Liu, W., and Hwang, I., "Probabilistic Trajectory Prediction and Conflict Detection for Air Traffic Control," *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 6, 2011, pp. 1779–1789.

[32] Lecchini-Visintini, A., Glover, W., Lygeros, J., and Maciejowski, J. M., "Monte Carlo Optimization for Conflict Resolution in Air Traffic Control," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 7, No. 4, 2006, pp. 470–482.

[33] Kantas, N., Lecchini-Visintini, A., and Maciejowski, J. M., "Simulation-Based Bayesian Optimal Design of Aircraft Trajectories for Air Traffic Management," *International Journal of Adaptive Control and Signal Processing*, Vol. 24, 2010, pp. 882–899.

[34] Chaloulos, G., Crück, E., and Lygeros, J., "A Simulation Based Study of Subliminal Control for Air Traffic Management," *Transportation Research Part C: Emerging Technologies*, Vol. 18, No. 6, 2010, pp. 963–974.

[35] Margellos, K., and Lygeros, J., "Toward 4-D Trajectory Management in Air Traffic Control: A Study Based on Monte Carlo Simulation and Reachability Analysis," *IEEE Transactions on Control Systems Technology*, Vol. 21, No. 5, 2013, pp. 1820–1833.

[36] Eele, A., Maciejowski, J., Chau, T., and Luk, W., "Control of Aircraft in the Terminal Manoeuvring Area Using Parallelised Sequential Monte Carlo," *AIAA Guidance, Navigation, and Control Conference*, Boston, MA, USA, AIAA Paper 2013-5257, 2013.

[37] Eele, A., Maciejowski, J., Chau, T., and Luk, W., "Parallelisation of Sequential Monte Carlo for Real-Time Control in Air Traffic Management," *52nd IEEE Conference on Decision and Control*, Florence, Italy, 2013, pp. 4859–4864.

[38] Stone, J. E., Phillips, J. C., Freddolino, P. L., Hardy, D. J., Trabuco, L. G., and Schulten, K., "Accelerating Molecular Modeling Applications with Graphics Processors," *Journal of Computational Chemistry*, Vol. 28, No. 16, 2007, pp. 2618–2640.

[39] Lee, A., Yau, C., Giles, M. B., Doucet, A., and Holmes, C. C., "On the Utility of Graphics Cards to Perform Massively Parallel Simulation of Advanced Monte Carlo methods," *Journal of Computational and Graphical Statistics*, Vol. 19, No. 4, 2010, pp. 769–789.

[40] Liu, W., and Hwang, I., "Probabilistic Aircraft Midair Conflict Resolution Using Stochastic Optimal Control, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 15, No. 1, 2014, pp. 37–46.

[41] European Organization for the Safety of Air Navigation, "EUROCONTROL Specification for Short Term Conflict Alert," 2009.

[42] Kochenderfer, M. J., Holland, J. E., and Chryssanthacopoulos, J. P., "Next-Generation Airborne Collision Avoidance System," *Lincoln Laboratory Journal*, Vol. 19, No. 1, 2012, pp. 17–33.

[43] Federal Aviation Administration, "Introduction to TCAS II Version 7.1," 2011.

[44] Cole, R., Richard, C., Kim, S., and Bailey, D., "An Assessment of the 60 km Rapid Update Cycle (RUC) with Near Real-Time Aircraft Reports," Project Report NASA/A-1, MIT Lincoln Laboratory, Lexington, 1998.

[45] Xiu, D., and Karniadakis, G. E., "The Wiener–Askey Polynomial Chaos for Stochastic Differential Equations," *SIAM Journal on Scientific Computing*, Vol. 24, No. 2, 2002, pp. 619–644.

[46] Xiu, D., and Hesthaven, J. S., "High-Order Collocation Methods for Differential Equations with Random Inputs," *SIAM Journal on Scientific Computing*, Vol. 27, No. 3, 2005, pp. 1118–1139.

[47] Blatman, G., and Sudret, B., "Adaptive Sparse Polynomial Chaos Expansion Based on Least Angle Regression," *Journal of Computational Physics*, Vol. 230, 2011, pp. 2345–2367.

[48] Benson, D. A., Huntington, G. T., Thorvaldsen, T. P., and Rao, A. V., "Direct Trajectory Optimization and Costate Estimation via an Orthogonal Collocation Method," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 6, 2006, pp. 1435–1440.

[49] Garg, D., Patterson, M., Hager, W. W., Rao, A. V., Benson, D. A., and Huntington, G. T., "A Unified Framework for the Numerical Solution of Optimal Control Problems Using Pseudospectral Methods," *Automatica*, Vol. 46, No. 11, 2010, pp. 1843–1851.

[50] Patterson, M. A., and Rao, A. V., "Exploiting Sparsity in Direct Collocation Pseudospectral Methods for Solving Optimal Control Problems," *Journal of Spacecraft and Rockets*, Vol. 49, No. 2, 2012, pp. 364–377.

[51] Ghosh, P., and Conway, B. A., "Near-Optimal Feedback Strategies Synthesized Using a Spatial Statistical Approach," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 4, 2013, pp. 905–919.

[52] Tsiotras, P., and Sanz Diaz, R., "Real-Time Near-Optimal Feedback Control of Aggressive Vehicle Maneuvers," *Optimization and Optimal Control in Automotive Systems*, Vol. 455, 2014, pp. 109–129.

[53] Schöbi, R., Kersaudy, P., Sudret, B., and Wiart, J., "Combining Polynomial Chaos Expansions and Kriging," RSUQ-2014-001, ETH Zürich, Zurich, 2014.

[54] Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P., "Design and Analysis of Computer Experiments," *Statistical Science*, Vol. 4, No. 4, 1989, pp. 409–423.

[55] Cressie, N. A. C., *Statistics for Spatial Data*, Wiley, New York, 1991.

[56] Schwartz, B., Benjamin, S., Green, S., and Jardin, M., "Accuracy of RUC-1 and RUC-2 Wind and Aircraft Trajectory Forecasts by Comparison with ACARS Observations," *Weather Forecasting*, Vol. 15, No. 3, 2000, pp. 316–326.

[57] Benjamin, S., Schwartz, B., and Cole, R., "Accuracy of ACARS Wind and Temperature Observations Determined by Collocation," *Weather and Forecasting*, Vol. 14, No. 6, 1999, pp. 1032–1038.

[58] Gel, Y., Raftery, A. E., and Gneiting, T., "Calibrated Probabilistic Mesoscale Weather Field Forecasting: The Geostatistical Output Perturbation Method," *Journal of the American Statistical Association*, Vol. 99, No. 467, 2004, pp. 575–583.

[59] Monbet, V., Ailliot, P., and Prevosto, M., "Survey of Stochastic Models for Wind and Sea State Time Series," *Probabilistic Engineering Mechanics*, Vol. 22, No. 2, 2007, pp. 113–126.

[60] Lymperopoulos, I., and Lygeros, J., "Improved Multi-Aircraft Ground Trajectory Prediction for Air Traffic Control," *Journal of Guidance, Control, and Dynamics* Vol. 33, No. 2, 2010, pp. 347–362.

[61] Lymperopoulos, I., and Lygeros, J., "Sequential Monte Carlo Methods for Multi-Aircraft Trajectory Prediction in Air Traffic Management," *International Journal of Adaptive Control and Signal Processing*, Vol. 24, 2010, pp. 830–849.

[62] Loève, M., *Probability Theory II*, Springer-Verlag, New York, 1978.

[63] Ghanem, R. G., and Spanos, P. D., *Stochastic Finite Elements: A Spectral Approach*, Springer-Verlag, New York, 1991.

[64] International Civil Aviation Organization, "Procedures for Air Navigation Services Air Traffic Management," Doc 4444ATM/501, 2007.

[65] Wiener, N., "The Homogeneous Chaos," *American Journal of Mathematics*, Vol. 60, No. 4, 1938, pp. 897–936.

[66] Cameron, R. H., and Martin, W. T., "The Orthogonal Development of Non-Linear Functionals in Series of Fourier-Hermite Functionals," *Annals of Mathematics*, Vol. 48, No. 2, 1947, pp. 385–392.

[67] Loève, M., *Probability Theory I*, Springer-Verlag, New York, 1977.

[68] Askey, R., and Wilson, J., "Some Basic Hypergeometric Orthogonal Polynomials that Generalize Jacobi Polynomials," *Memoirs of the American Mathematical Society*, Vol. 54, No. 319, 1985.

[69] Oladyshkin, S., and Nowak, W., "Data-Driven Uncertainty Quantification Using the Arbitrary Polynomial Chaos Expansion," *Reliability Engineering & System Safety*, Vol. 106, 2012, pp. 179–190.

[70] Ghanem, R., and Dham, S., "Stochastic Finite Element Analysis for Multiphase Flow in Heterogeneous Porous Media," *Transport in Porous Media*, Vol. 32, No. 3, 1998, pp. 239–262.

[71] Ghanem, R., "Ingredients for a General Purpose Stochastic Finite Element Implementation," *Computer Methods in Applied Mechanics and Engineering*, Vol. 168, Nos. 1–4, 1999, pp. 19–34.

[72] Ghanem, R., "Stochastic Finite Elements with Multiple Random Non-Gaussian Properties," *Journal of Engineering Mechanics*, Vol. 125, No. 1, 1999, pp. 26–40.

[73] Sudret, B., and Der Kiureghian, A., "Stochastic Finite Element Methods and Reliability: A State-of-the-Art Report," Report No. UCB/SEMM-2000/08, University of California, Berkeley, 2000.

[74] Le Maître, O. P., Knio, O. M., Najm, H. N., and Ghanem, R. G., "A Stochastic Projection Method for Fluid Flow: I. Basic Formulation," *Journal of Computational Physics*, Vol. 173, No. 2, 2001, pp. 481–511.

[75] Le Maître, O. P., Reagan, M. T., Najm, H. N., Ghanem, R. G., and Knio, O. M., "A Stochastic Projection Method for Fluid Flow: II. Random Process," *Journal of Computational Physics*, Vol. 181, No. 1, 2002, pp. 9–44.

[76] Mathelin, L., and Hussaini, M. Y., "A Stochastic Collocation Algorithm for Uncertainty Analysis," Technical Report, NASA/CR-2003-212153, 2003.

[77] Xiu, D., and Karniadakis, G. E., "Modeling Uncertainty in Flow Simulations via Generalized Polynomial Chaos," *Journal of Computational Physics*, Vol. 187, No. 1, 2003, pp. 137–167.

[78] Knio, O. M., and Le Maître, O. P., "Uncertainty Propagation in CFD Using Polynomial Chaos Decomposition," *Fluid Dynamics Research* Vol. 38, No. 9, 2006, pp. 616–640.

[79] Sandu, A., Sandu, C., and Ahmadian, M., "Modeling Multibody Systems with Uncertainties. Part I: Theoretical and Computational Aspects," *Multibody System Dynamics*, Vol. 15, No. 4, 2006, pp. 369–391.

[80] Sandu, C., Sandu, A., and Ahmadian, M., "Modeling Multibody Systems with Uncertainties. Part II: Numerical Applications," *Multibody System Dynamics*, Vol. 15, No. 3, 2006, pp. 241–262.

[81] Xiu, D., *Numerical Methods for Stochastic Computations*, Princeton University Press, Princeton, 2010.

[82] Ghiocel, D. M., and Ghanem, R. G., "Stochastic Finite-Element Analysis of Seismic Soil–Structure Interaction," *Journal of Engineering Mechanics*, Vol. 128, No. 1, 2002, pp. 66–77.

[83] Choi, S. -K., Grandhi, R. V., Canfield, R. A., and Pettit, C. L., "Polynomial Chaos Expansion with Latin Hypercube Sampling for Estimating Response Variability," *AIAA Journal*, Vol. 42, No. 6, 2004, pp. 1191–1198.

[84] Berveiller, M., Sudret, B., and Lemaire, M., "Stochastic Finite Element: A Non Intrusive Approach by Regression," *European Journal of Computational Mechanics*, Vol. 15, 2006, pp. 81–92.

[85] Sudret, B., "Global Sensitivity Analysis Using Polynomial Chaos Expansions," *Reliability Engineering & System Safety*, Vol. 93, No. 7, 2008, pp. 964–979.

[86] Blatman, G., and Sudret, B., "An Adaptive Algorithm to Build Up Sparse Polynomial Chaos Expansions for Stochastic Finite Element Analysis," *Probabilistic Engineering Mechanics*, Vol. 25, No. 2, 2010, pp. 183–197.

[87] Doostan, A., and Owhadi, H., "A Non-Adapted Sparse Approximation of PDEs with Stochastic Inputs," *Journal of Computational Physics*, Vol. 230, No. 8, 2011, pp. 3015–3034.

[88] Constantine, P. G., Eldred, M. S., and Phipps, E. T., "Sparse Pseudospectral Approximation Method," *Computer Methods in Applied Mechanics and Engineering*, Vols. 229–232, 2012, pp. 1–12.

[89] Fagiano, L., and Khammash, M., "Simulation of Stochastic Systems via Polynomial Chaos Expansions and Convex Optimization," *Physical Review E*, Vol. 86, 2012, 036702.

[90] Peng, J., Hampton, J., and Doostan, A., "A Weighted l_1-minimization for Sparse Polynomial Chaos Expansions," *Journal of Computational Physics*, Vol. 267, 2014, pp. 92–111.

[91] Smolyak, S. A., "Quadrature and Interpolation Formulas for Tensor Products of Certain Classes of Functions," *Soviet Mathematics*, Vol. 4, 1963, pp. 240–243.

[92] Bungartz, H. -J., and Griebel, M., "Sparse Grids," *Acta Numerica*, Vol. 13, 2004, pp. 147–269.

[93] Heiss, F., and Winschel, V., "Likelihood Approximation by Numerical Integration on Sparse Grids," *Journal of Econometrics*, Vol. 144, No. 1, 2008, pp. 62–80.

[94] Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R., "Least Angle Regression," *The Annals of Statistics*, Vol. 32, No. 2, 2004, pp. 407–499.

[95] Blatman, G., and Sudret, B., "Efficient Computation of Global Sensitivity Indices Using Sparse Polynomial Chaos Expansions," *Reliability Engineering & System Safety*, Vol. 95, No. 11, 2010, pp. 1216–1229.

[96] Garcia–Cabrejo, O., and Valocchi, A., "Global Sensitivity Analysis for Multivariate Output Using Polynomial Chaos Expansion," *Reliability Engineering & System Safety*, Vol. 126, 2014, pp. 25–36.

[97] Sobol', I. M., "Global Sensitivity Indices for Nonlinear Mathematical Models and Their Monte Carlo Estimates," *Mathematics and Computers in Simulation*, Vol. 55, 2001, pp. 271–280.

[98] Betts, J. T., "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, 1998, pp. 193–207.

[99] Betts, J. T., *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, Society for Industrial and Applied Mathematics, Philadelphia, 2010.

[100] Pontryagin, L. S., Boltyanskii, V., Gamkrelidze, R., and Mischenko, E., *The Mathematical Theory of Optimal Processes*, Interscience, New York, 1962.

[101] Kirk, D. E., *Optimal Control Theory: An Introduction*, Prentice Hall, New Jersey, 1970.

[102] Bryson, A. E., and Ho, Y. C., *Applied Optimal Control: Optimization, Estimation, and Control*, Taylor & Francis, New York, 1975.

[103] Vlassenbroeck, J., and Van Doreen, R., "A Chebyshev Technique for Solving Nonlinear Optimal Control Problems," *IEEE Transactions on Automatic Control*, Vol. 33, No. 4, 1988, pp. 333–340.

[104] Vlassenbroeck, J., "A Chebyshev Polynomial Method for Optimal Control with State Constraints," *Automatica*, Vol. 24, No. 4, 1988, pp. 499–506.

[105] Elnagar, G., Kazemi, M., and Razzaghi, M., "The Pseudospectral Legendre Method for Discretizing Optimal Control Problems," *IEEE Transactions on Automatic Control*, Vol. 40, No. 10, 1995, pp. 1793–1796.

[106] Elnagar, G., and Kazemi, M. A., "Pseudospectral Chebyshev Optimal Control of Constrained Nonlinear Dynamical Systems," *Computational Optimization and Applications*, Vol. 11, 1998, pp. 195–217.

[107] Gong, Q., Kang, W., and Ross, I. M., "A Pseudospectral Method for the Optimal Control of Constrained Feedback Linearizable Systems," *IEEE Transactions on Automatic Control*, Vol. 51, No. 7, 2006, pp. 1115–1129.

[108] Benson, D. A., *A Gauss Pseudospectral Transcription for Optimal Control*, Ph. D. Dissertation, Massachusetts Institute of Technology, Cambridge, 2005.

[109] Huntington, G. T., *Advancement and Analysis of a Gauss Pseudospectral Transcription for Optimal Control Problems*, Ph. D. Dissertation, Massachusetts Institute of Technology, Cambridge, 2007.

[110] Fahroo, F., and Ross, I. M., "Costate Estimation by a Legendre Pseudospectral Method," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 2, 2001, pp. 270–277.

[111] Rao, A. V., Benson, D., Darby, C. L., Mahon, B., Francolin, C., Patterson, M., Sanders, I., and Huntington, G. T., "User's Manual for GPOPS Version 5.0: A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using *hp*-Adaptive Pseudospectral Methods," University of Florida, Gainesville, 2011.

[112] Rao, A. V., Benson, D. A., Darby, C. L., Patterson, M. A., Francolin, C., Sanders, I., and Huntington, G. T., "GPOPS: A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using the Gauss Pseudospectral Method," *ACM Transactions on Mathematical Software*, Vol. 37, No. 2, 2010, Article 22.

[113] Patterson, M. A. and Rao, A. V., "GPOPS-II: A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using *hp*-Adaptive Gaussian Quadrature Collocation Methods and Sparse Nonlinear Programming," *ACM Transactions on Mathematical Software*, Vol. 39, No. 3, 2013, Article 1.

[114] Garg, D., Hager, W. W., and Rao, A. V., "Pseudospectral Methods for Solving Infinite-Horizon Optimal Control Problems," *Automatica*, Vol. 47, No. 4, 2011, pp. 829–837.

[115] Garg, D., Patterson, M. A., Darby, C. L., Francolin, C., Huntington, G. T., Hager, W. W., and Rao, A. V., "Direct Trajectory Optimization and Costate Estimation of Finite-Horizon and Infinite-Horizon Optimal Control Problems Using a Radau Pseudospectral Method," *Computational Optimization and Applications*, Vol. 49, No. 2, 2011, pp. 335–358.

[116] Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Review*, Vol. 47, No. 1, 2005, pp. 99–131.

[117] Gill, P. E., Murray, W., and Saunders, M. A., "User's Guide for SNOPT Version 7: Software for Large-Scale Nonlinear Programming," University of California, San Diego, La Jolla, 2008.

[118] Prékopa, A., *Stochastic Programming*, Kluwer Academic Publishers, Dordrecht, 1995.

[119] Schwarm, A., and Nikolaou, M., "Chance-Constrained Model Predictive Control," *AlChE Journal*, Vol. 45, No. 8, 1999, pp.1743–1752.

[120] Blackmore, L., Ono, M., Bektassov, A., and Williams, B. C., "A Probabilistic Particle-Control Approximation of Chance-Constrained Stochastic Predictive Control," *IEEE Transactions on Robotics*, Vol. 26, No. 3, 2010, pp. 502–517.

[121] Fisher, J., and Bhattacharya, R., "Optimal Trajectory Generation with Probabilistic System Uncertainty Using Polynomial Chaos," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 133, No. 1, 2011, 014501.

[122] Cottrill, G. C., and Harmon, F. G., "Hybrid Gauss Pseudospectral and Generalized Polynomial Chaos Algorithm to Solve Stochastic Trajectory Optimization Problems," *AIAA Guidance, Navigation, and Control Conference*, Portland, OR, USA, AIAA Paper 2011-6572, 2011.

[123] Flanzer, T. C., Bower, G. C., and Kroo, I. M., "Robust Trajectory Optimization for Dynamic Soaring," *AIAA Guidance, Navigation, and Control Conference*, Minneapolis, MN, USA, AIAA Paper 2012-4603, 2012.

[124] Tiesler, H., Kirby, R. M., Xiu, D., and Preusser, T., "Stochastic Collocation for Optimal Control Problems with Stochastic PDE Constraints," *SIAM Journal on Control and Optimization*, Vol. 50, No. 5, 2012, pp. 2659–2682.

[125] Li, X., Nair, P. B., Zhang, Z., Gao, L., and Gao, C., "Aircraft Robust Trajectory Optimization Using Nonintrusive Polynomial Chaos," *Journal of Aircraft*, Vol. 51, No. 5, 2014, pp. 1592–1603.

[126] Simpson, T. W., Poplinski, J. D., Koch, P. N., and Allen, J. K., "Metamodels for Computer-Based Engineering Design: Survey and Recommendations," *Engineering with Computers*, Vol. 17, No. 2, 2001, pp. 129–150.

[127] Queipo, N. V., Haftka, R. T., Shyy, W., Goel, T., Vaidyanathan, R., and Tucker, P. K., "Surrogate-Based Analysis and Optimization," *Progress in Aerospace Sciences*, Vol. 41, No. 1, 2005, pp. 1–28.

[128] Forrester, A. I. J., and Keane, A. J., "Recent Advances in Surrogate-Based Optimization," *Progress in Aerospace Sciences*, Vol. 45, Nos. 1–3, 2009, pp. 50–79.

[129] Viana, F. A. C., Simpson, T. W., Balabanov, V., and Toropov, V., "Metamodeling in Multidisciplinary Design Optimization: How Far Have We Really Come?," *AIAA Journal*, Vol. 52, No. 4, 2014, pp. 670–690.

[130] Myers, R. H., and Montgomery, D. C., *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, Wiley, New York, 2002.

[131] Powell, M. J. D., "Radial Basis Functions for Multivariate Interpolation: A Review," *Algorithms for Approximation*, edited by Mason, J. C., and Cox, M. G., Clarendon Press, Oxford, 1987, pp. 143–167.

[132] Friedman, J. H., "Multivariate Adaptive Regression Splines," *The Annals of Statistics*, Vol. 19, No. 1, 1991, pp. 1–67.

[133] Smola, A. J., and Schölkopf, B., "A Tutorial on Support Vector Regression," *Statistics and Computing*, Vol. 14, No. 3, 2004, pp. 199–222.

[134] Kim, C., Wang, S., and Choi, K. K., "Efficient Response Surface Modeling by Using Moving Least-Squares Method and Sensitivity," *AIAA Journal*, Vol. 43, No. 11, 2005, pp. 2404–2411.

[135] Breitkopf, P., Naceur, H., Rassineux, A., and Villon, P., "Moving Least Squares Response Surface Approximation: Formulation and Metal Forming Applications," *Computers & Structures*, Vol. 83, Nos. 17–18, 2005, pp. 1411–1428.

[136] Coelho, R. F., Lebon, J., and Bouillard, P., "Hierarchical Stochastic Metamodels Based on Moving Least Squares and Polynomial Chaos Expansion," *Structural and Multidisciplinary Optimization*, Vol. 43, No. 5, 2011, pp. 707–729.

[137] Rasmussen, C. E., and Williams, C. K. I., *Gaussian Processes for Machine Learning*, MIT Press, Cambridge, 2006.

[138] Forrester, A. I. J., Sóbester, A., and Keane, A. J., *Engineering Design via Surrogate Modelling: A Practical Guide*, Wiley, New York, 2008.

[139] Martin, J. D., and Simpson, T. W., "Use of Kriging Models to Approximate Deterministic Computer Models," *AIAA Journal*, Vol. 43, No. 4, 2005, pp. 853–863.

[140] Bachoc, F., "Cross Validation and Maximum Likelihood Estimations of Hyper-Parameters of Gaussian Processes with Model Misspecification," *Computational Statistics & Data Analysis*, Vol. 66, 2013, pp. 55–69.

[141] Lophaven, S. N., Nielsen, H. B., and Søndergaard, J., "DACE: A MATLAB Kriging Toolbox," Technical Report IMM-TR-2002-12, Technical University of Denmark, Lyngby, 2002.

[142] Roustant, O., Ginsbourger, D., and Deville, Y., "DiceKriging, DiceOptim: Two R Packages for the Analysis of Computer Experiments by Kriging-Based Metamodeling and Optimization," *Journal of Statistical Software*, Vol. 51, No. 1, 2012, pp. 1–55.

[143] McKay, M. D., Beckman, R. J., and Conover, W. J., "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, Vol. 21, No. 2, 1979, pp. 239–245.

[144] Stein, M., "Large Sample Properties of Simulations Using Latin Hypercube Sampling," *Technometrics*, Vol. 29, No. 2, 1987, pp. 143–151.

[145] Abramowitz, M., and Stegun, I. A., *Handbook of Mathematical Functions: with Formulas, Graphs, and Mathematical Tables*, Dover Publications, New York, 1972.