Modeling, Control, and Optimization in Aeronautical Engineering
Modeling, Control, and Optimization in Aeronautical Engineering

Contents

Volume 2015, Article ID 979107, 2 pages

Terminal-Area Aircraft Intent Inference Approach Based on Online Trajectory Clustering, Yang Yang, Jun Zhang, and Kai-quan Cai
Volume 2015, Article ID 671360, 11 pages

An Airway Network Flow Assignment Approach Based on an Efficient Multiobjective Optimization Framework, Xiangmin Guan, Xuejun Zhang, Yanbo Zhu, Dengfeng Sun, and Jiaxing Lei
Volume 2015, Article ID 302615, 9 pages

Helicopter Control Energy Reduction Using Moving Horizontal Tail, Tugrul Oktay and Firat Sal
Volume 2015, Article ID 523914, 10 pages

A Robust $H_{\infty}$ Controller for an UAV Flight Control System, J. López, R. Dormido, S. Dormido, and J. P. Gómez
Volume 2015, Article ID 403236, 11 pages

A Novel Biobjective Risk-Based Model for Stochastic Air Traffic Network Flow Optimization Problem, Kaiquan Cai, Yaoguang Jia, Yanbo Zhu, and Mingming Xiao
Volume 2015, Article ID 742541, 10 pages

MRAC Control with Prior Model Knowledge for Asymmetric Damaged Aircraft, Xieyu Xu, Lingyu Yang, and Jing Zhang
Volume 2015, Article ID 247301, 10 pages

Coordinated Control of Slip Ratio for Wheeled Mobile Robots Climbing Loose Sloped Terrain, Zhengcai Li and Yang Wang
Volume 2014, Article ID 396382, 13 pages
Editorial

Modeling, Control, and Optimization in Aeronautical Engineering

Zheng Zheng,1 Kemao Peng,2 Wenbo Du,3 and Guangquan Zhang4

1School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China
2Temasek Laboratories, National University of Singapore, Singapore
3School of Electronic and Information Engineering, Beihang University, Beijing 100191, China
4Centre for Quantum Computation and Intelligent Systems, University of Technology, Sydney, NSW 2007, Australia

Correspondence should be addressed to Zheng Zheng; zhengz@buaa.edu.cn

Received 3 May 2015; Accepted 3 May 2015

The last decade has witnessed the rapid development of aeronautical engineering including all branches of applied sciences and technology dealing with aircraft and their support systems, which brings forward the higher request of safety, efficiency, and environmental protection. Modeling, control, and optimization of aeronautical engineering have played an increasingly important role in meeting aeronautical requirements, and they have drawn widespread attention from communities including control theory, intelligent optimization, system science, real-time distributed computing, electronic information engineering, and aeronautical engineering industry. Driven by such motivations, the main focus of this special issue is on the new theories, new technologies, and their applications in modeling, control, and optimization for aeronautical engineering systems.

The special issue is a collection of original research articles, whose authors and editors belong to academic or research institutions of five different countries from Asia, Europe, and Australia. The full papers in this issue can be broadly organized into two main categories: (i) modelling and optimization and (ii) control techniques.

Papers in category (i) are mainly concerned with network optimization of air transportation system and terminal-area operation. Recently, people come to realize that the air transportation system can be modeled as complex networks. Along this line, K. Cai et al. in “A Novel Biobjective Risk-Based Model for Stochastic Air Traffic Network Flow Optimization Problem” and X. Guan et al. in “An Airway Network Flow Assignment Approach Based on an Efficient Multiobjective Optimization Framework” modeled the flight operation as network flow optimization problem and proposed novel algorithms, respectively. Terminal-area operation is the bottleneck of the air transportation system. Y. Yang et al. investigated the aircraft intent inference approach and proposed an online trajectory clustering method, which will be meaningful to the efficiency and safety of busy terminal area.

Papers in category (ii) are focusing on the topics of rotary-wing aircrafts and robots. The rotary-wing aircraft is one of the popular platforms in the control community such as the helicopter and quadrotor. T. Oktay and F. Sal in “Helicopter Control Energy Reduction Using Moving Horizontal Tail” attempted to improve the flight duration by reducing consumption of the energy. X. Xu et al. in “MRAC Control with Prior Model Knowledge for Asymmetric Damaged Aircraft” intended to explore control of the wing-damaged aircraft, which is one of advanced topics to improve survivability of the fighters. Z. Li and Y. Wang in “Coordinated Control of Slip Ratio for Wheeled Mobile Robots Climbing Loose Sloped Terrain” attempted to explore the control of mobile robots in 3D environments. J. López et al. in “A Robust $H_{\infty}$ Controller for an UAV Flight Control System” implemented and validated a robust $H_{\infty}$ controller for an UAV to track different types of manoeuvres in the presence of noisy environment.

It should be mentioned that a special issue simply provides a snapshot of the field taken at a particular point in time. Due to the standard page limitations of a journal volume,
it can only include a relatively small number of papers. As a result, its coverage is by no means complete despite our best efforts.

Acknowledgments

Many individuals contributed to the success of this special issue. We take this opportunity to thank all the authors for their submissions. We are also indebted to a small army of referees who have put in the hard work and the long hours to review each paper in a timely and professional way. Editorial Assistant Kun Qiu provided valuable assistance. Last but not least, we are indebted to this journal for offering us this opportunity and for patiently waiting for the completion of the special issue.

Zheng Zheng
Kemao Peng
Wenbo Du
Guangquan Zhang
Terminal-Area Aircraft Intent Inference Approach Based on Online Trajectory Clustering

Yang Yang, Jun Zhang, and Kai-quan Cai

School of Electronic and Information Engineering, Beihang University, Xueyuan Road No. 37, Beijing 100191, China

Correspondence should be addressed to Kai-quan Cai; caikq@ee.buaa.edu.cn

Received 11 November 2014; Accepted 15 December 2014

Copyright © 2015 Yang Yang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Terminal-area aircraft intent inference (T-AII) is a prerequisite to detect and avoid potential aircraft conflict in the terminal airspace. T-AII challenges the state-of-the-art AII approaches due to the uncertainties of air traffic situation, in particular due to the undefined flight routes and frequent maneuvers. In this paper, a novel T-AII approach is introduced to address the limitations by solving the problem with two steps that are intent modeling and intent inference. In the modeling step, an online trajectory clustering procedure is designed for recognizing the real-time available routes in replacing of the missed plan routes. In the inference step, we then present a probabilistic T-AII approach based on the multiple flight attributes to improve the inference performance in maneuvering scenarios. The proposed approach is validated with real radar trajectory and flight attributes data of 34 days collected from Chengdu terminal area in China. Preliminary results show the efficacy of the presented approach.

1. Introduction

Terminal-area aircraft intent inference (T-AII) consists in predicting the future intended route for aircraft flying in the terminal airspace and can hence support conflict detection and resolution (CD&R) operations [1]. Indeed, in order to address the flight safety challenges posed by high traffic density operation in the terminal area, automatic CD&R tool resting on the aircraft intent information is taken as the main resolution strategy in the next generation air traffic management system master plans, such as, European SESAR [2] and US NextGen [3]. Recently, the AII problem has been widely studied in both academic and industrial contexts [4–9]. Additionally, the prototype systems for maintaining aircraft separation in terminal area based on aircraft intent have also been developed and tested [10–12].

Due to the uncertainties affecting the terminal-area operations, T-AII is a more difficult task than en route AII. This is because terminal-area air traffic control operates in a low altitude airspace, which can be strongly affected by severe meteorological conditions and may be subject to special use airspace restrictions. The route assigned to each aircraft is determined by the real-time airspace conditions and, as a result, no detailed flight plan can be provided in advance to pilots for the terminal flight phase. Furthermore, to maintain safe separation in a confined space, air traffic controllers issue clearances, such as vectoring, hold-on, and direct-to clearances, to direct aircraft, and, hence, pilots must implement maneuvers frequently. Clearances are oral and transmitted by a voice communication system, which aggravates the unpredictability of T-AII. These uncertainties of air traffic situation characterizing the terminal area bring new challenges for the AII problem.

The state-of-the-art AII approaches can be divided into two main classes depending on the kind of information that is used to infer the aircraft future intended route. In the first class of approaches, pilot actions are supposed to be available and are matched to the intended route as extracted from the en route flight plan [13, 14]. Various methods were then proposed to solve this best matching problem, such as those based on expert systems [15], plan recognition [16], and event tracking [17]. However, since pilot actions are hard to collect and transmit, these approaches are not widely used in practice. In the second class of approaches, an air traffic controller point of view is adopted and AII is realized through best fitting of the aircraft state observations.
to intent models [4, 5]; for example, heading angle is used to infer horizontal intent. Enhanced versions of this idea have been developed in [6–8], where both spatial and temporal aircraft information are adopted to improve intent inference performance. Obviously, the second class of approaches is much easier to be implemented and demonstrated.

Both classes of approaches are characterized by the following two properties, which make them not suitable for addressing the T-AII problem: (1) they rely on predefined flight plans (for example, in [4], 22 intent models based on the flight plan are investigated); and (2) they are all based on best fitting of observations to the intent model, which can be effective only in an en route scenario where maneuvers are quite limited.

In order to solve the specific challenges in the T-AII problem, this paper proposes a two-step T-AII approach where in the first step an intent model is derived and then, in the second step, intent inference is performed based on the identified intent model.

First of all, intent modeling is formulated as a trajectory clustering problem which is motivated by the idea of terminal path library [18, 19]. The intended routes are represented by the cluster centroids extracted from the trajectory data without the input of flight plans. However, the existing trajectory clustering approaches in aviation domain are offline in nature [20–27] with the aim of evaluating the operational performance, which are relying on the complete trajectory information obtained after operations. Hence, an online trajectory clustering approach is designed so as to recognize the real-time intended routes in which the cluster centroids are dynamically updated and reorganized based on the new input information of trajectory. Here, a flexible binary-tree structure [28, 29] is adopted.

In the second step, the T-AII is implemented with a probabilistic scheme integrating the multiple flight attributes. The attributes, such as, heading angle and route relevance, can be obtained through the trajectory data. Note that the route relevance expressing the relation of two connected routes is offered by the online trajectory clustering procedure. Other attributes, such as flight call sign, destination airports, aircraft size, and aircraft approach category, can be obtained from air traffic flow management system. Then, based on the current trajectory and flight attributes, the intended route with maximum probability is identified as the inferred intent at the time instant.

The proposed approach is verified on the radar trajectory and flight attributes data of 34 days including 8995 departure flights collected from Chengdu terminal area in China. Based on intended routes recognized from the online trajectory clustering procedure, the efficacy of T-AII is investigated in three intent inference scenarios with aircraft maneuvers.

The rest of this paper is organized as follows. Section 2 briefly outlines the T-AII problem with some examples. Section 3 introduces the proposed two-step T-AII approach and prescribes in detail the online trajectory clustering procedure and the probabilistic T-AII with multiple flight attributes. Some experimental results are presented in Section 4. Finally, Section 5 draws some conclusions.

2. Problem Description

The configuration of terminal area is normally shown as a circular region centered at the airport shown in Figure 1, where departure routes (blue lines) and arrival routes (grey lines) are always spatially separated for the sake of flight safety. The solid lines represent the available routes while the dashed lines represent the closed routes. Severe meteorological or special use airspace conditions are represented by yellow
3. T-AII Approach Based on Online Trajectory Clustering

Figure 2 gives the block diagram of the T-AII approach including two successive steps that are intent modeling and intent inference. The intent modeling step is divided into two parts: cluster building and cluster management. Radar trajectory at each time instant is applied to update the cluster centroids and dynamically reshape the structure of clusters based on the cluster control rules and operators, such as updating, splitting, merging, and deleting operators. In the intent inference step, several flight attributes are considered and the corresponding probabilities are extracted from both trajectory data and air traffic flow management system. The candidate intended route with the maximum probability summation of all attributes in logarithmic format is identified as the inferred intent. The detailed steps of the procedure are summarized as cluster building and cluster management.

(1) Cluster Building. Initially, set the first cluster $C_1 \in \mathbf{C}$ as the same positions of forpart $\alpha$ points of the trajectory $\{T^i_1\}^n_{j=1}$. Then, new trajectory $\{T^j_{i+1}\}_{i=1}^{n}$ is used to match any existing cluster $C_m \in \mathbf{C}$ based on the trajectory-cluster similarity (TCS) defined in the following:

$$\{\text{TCS}(T^j_i, C_m)\}_{j=1}^{\alpha} = \left\{ \min_{k=1, K_m} \frac{1}{\sigma_m^k} \sqrt{(x^j_i - \bar{x}^k_m)^2 + (y^j_i - \bar{y}^k_m)^2} \right\}_{j=1}^{\alpha} \quad (1)$$

If there is any $C_m \in \mathbf{C}$, let the similarity summation $\sum_{j=1}^{\alpha} \text{TCS}(T^j_i, C_m)$ satisfy the predefined similarity threshold; then the trajectory is assigned the specific cluster. Otherwise, a new cluster is created based on the trajectory information. Cluster set $\mathbf{C}$ is updated.

Clusters are dynamically manipulated through formulating the cluster set into the binary-tree structure based on the following cluster control rules and cluster building operators. The subsequent points $\{T^j_i\}_{j=\alpha+1}^{n}$ of each trajectory are the input information for the online process. In order to handle the trajectory point at time instant $j$, we also need to reserve the subsequent $\alpha - 1$ trajectory points. Note that an array of trajectory points is used to match the cluster instead of a single point so as to avoid the mismatch due to some irregular trajectory data. As a result, the TCS between the trajectory points $\{T^j_i\}_{j=1}^{\alpha}$ and cluster $C_m$ are denoted as an array $\{\text{TCS}(T^j_i, C_m)\}_{j=1}^{\alpha}$, where $\tau$ is the time instant. Here, some notations are given: reserving array is denoted as reserving, the lower and upper similarity thresholds are denoted as $S_l$ and $S_u$, and the lower and upper thresholds of array length are denoted as $L_l$ and $L_u$. 

3.1. Online Trajectory Clustering. In this section, we present the online trajectory clustering procedure. Let us denote by $\mathbf{C} = \{C_m\}_{m=1}^{M}$ the cluster set with $M$ elements organized in binary-tree structure. Each cluster is represented by the cluster centroids $C_m = \{\bar{x}^k_m, \bar{y}^k_m, \sigma^k_m\}_{k=1}^{K_m}$ along the time horizon $[1, K_m]$, where $\{\bar{x}^k_m, \bar{y}^k_m\}$ represents the horizontal position at discrete time instant $k$ and $\sigma^k_m$ represents variance of cluster in the corresponding position. The trajectories are denoted by $T = \{T^i_1\}, i = 1, \ldots, N, j = 1, \ldots, n$, where $T^i_j = \{x^i_j, y^i_j\}$ represents the position of trajectory $i$ at time instant $j$. $N$ represents the number of aircraft trajectories and $n$ represents the number of points for each trajectory. The detailed steps of the procedure are summarized as cluster building and cluster management.
The detailed cluster control rules are given as follows. Note that the trajectory point is determined as the outlier if it does not satisfy any of the following rules.

1. If the trajectory point $T_i$, $\tau = j, \ldots, j + (\alpha - 1)$, satisfies $TCS(T_i, C_m) \leq S_i$, the updating operator is active. Otherwise, the trajectory point $T_i$, $\tau = j, \ldots, j + (\alpha - 1)$, is inserted into the array reserving, and the updating and splitting operators are inactive. Then, trajectory is shifted to the next time instant $\tau + 1$.

2. Among the array reserving, if there exists a subarray $\{TCS(T_i, C_m)\}_{r=k}^{k+l-1}$ with length $l < L_1$ in which values of all elements are larger than $S_i$ and the similarity at time instant $k + 1$ satisfies $TCS(T_i, C_m) \leq S_i$, the updating operator is active. Then, the cluster $C_m$ is updated with trajectory information $\{TCS(T_i, C_m)\}_{r=k}^{k+l}$ and it empties the array reserving.

3. Among the array reserving, if there exists a subarray with length $l \geq L_1$ in which values of all elements are larger than $S_i$, the splitting operator is active. Then, the cluster $C_m$ is split into two separate clusters and it empties the array reserving.

4. Among the array reserving, if there exists a subarray with length $L_1 \leq l \leq L_n$ in which values of all elements are larger than $S_n$, the splitting operator is active. Then, the cluster $C_m$ is split into two separate clusters and it empties the array reserving.

Then, the two cluster building operators are described as follows.

1. Updating operator: when cluster $C_m$ is updated to the new cluster $C'_m$ based on the trajectory point information $T_i = (x'_i, y'_i)$, the corresponding centroid of $C_m = \{x^k_m, y^k_m, \sigma^k_m\}$ is updated based on (2), where $k^*$ is the index of centroid gained with the corresponding minimum TCS value and $\eta_1 \in [0, 0.5]$:

   \[
   \begin{align*}
   \left(\hat{x}^k_m\right)' &= \eta_1 \hat{x}^k_m + (1 - \eta_1) x'_i, \\
   \left(\hat{y}^k_m\right)' &= \eta_1 \hat{y}^k_m + (1 - \eta_1) y'_i, \\
   \left(\hat{\sigma}_m\right)' &= \eta_1 \hat{\sigma}_m + (1 - \eta_1) \left[\left(x'_i - \hat{x}^k_m\right)^2 + \left(y'_i - \hat{y}^k_m\right)^2\right].
   \end{align*}
   \] (2)

2. Splitting operator: when the cluster $C_m$ is split at time instant $k^*$, $C_m$ is separated into two new clusters $C'_m$ and $C''_m$. Then, a new cluster $C'''_m$ with default variance is created based on the subsequent trajectory points after time instant $k^*$; see Figure 3.

(2) **Cluster Management.** In the cluster building step, a tree structure based cluster set is generated. However, each cluster is followed by multiple subsequent clusters due to the frequent use of splitting operator. In order to recognize the main clusters, the clusters are managed through deleting and merging operators. Note that the number of trajectory points belonging to each cluster is counted in the cluster building step. And the cluster-cluster similarity (CCS) is defined in the following:

\[
CCS(C_m, C_n) = \frac{1}{K} \sqrt{\sum_{k=1}^{K} \left(\hat{x}^k_m - \hat{x}^k_n\right)^2 + \left(\hat{y}^k_m - \hat{y}^k_n\right)^2},
\] (3)

\[K = \min(K_m, K_n).\]

The cluster management operators are described as follows.

1. Deleting operator: if the number of trajectory points associated to the cluster is less than a threshold, the corresponding cluster is deleted. These trajectory points are classified as outliers.

2. Merging operator: when CCS of clusters $C_m$ and $C_n$ satisfies the predefined threshold, these clusters are merged and the new cluster is computed based on (4), where $\eta_2$ is defined as the proportion of number of trajectory points associated to the cluster $C_m$ in the total number of trajectory points associated to clusters $C_m$ and $C_n$:

\[
\begin{align*}
\left(\hat{x}^k_m\right)' &= \eta_2 \hat{x}^k_m + (1 - \eta_2) x'_n, \\
\left(\hat{y}^k_m\right)' &= \eta_2 \hat{y}^k_m + (1 - \eta_2) y'_n, \\
\left(\hat{\sigma}_m\right)' &= \eta_2 \hat{\sigma}_m + (1 - \eta_2) \hat{\sigma}_n, \\
& \quad k = 1, \ldots, \min(K_m, K_n).
\end{align*}
\] (4)

Two kinds of merging operators are shown in Figures 4 and 5. For parallel merging operator, the clusters $C_m$ and $C_n$ are divided into 3 new clusters separately denoted by $C'_m, C'''_{m+1}, C'''_{m+2}$ and $C''_n, C'''_{n+1}, C'''_{n+2}$. The corresponding clusters $C'_m$ and $C''_n$ are then merged into a new cluster $C'''_m$. For cascaded merging operator,
Table 1: Notation and definition of flight attributes.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
<th>Category</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS</td>
<td>Call sign (CS) is the identification of each flight, such as ABW285 and CAO1023.</td>
<td>Static</td>
<td>Discrete</td>
</tr>
<tr>
<td>DA</td>
<td>Destination airport (DA) is divided into several groups based on geographical location of destination airport, such as northeast and southwest.</td>
<td>Static</td>
<td>Discrete</td>
</tr>
<tr>
<td>AS</td>
<td>Aircraft size (AS) refers to the wake turbulence category, such as heavy, medium, and light.</td>
<td>Static</td>
<td>Discrete</td>
</tr>
<tr>
<td>AC</td>
<td>Aircraft approach category (AC) is defined based on the speed category on the approach phase, such as A, B, C, D, and E.</td>
<td>Static</td>
<td>Discrete</td>
</tr>
<tr>
<td>RR</td>
<td>Route relevance (RR) refers to the relation of the connected intended routes.</td>
<td>Dynamic</td>
<td>Discrete</td>
</tr>
<tr>
<td>HA</td>
<td>Heading angle (HA) is defined as the angular difference between the aircraft heading direction and the intended routes.</td>
<td>Dynamic</td>
<td>Continuous</td>
</tr>
</tbody>
</table>

3.2. T-AII with Multiple Attributes. Based on the online trajectory clustering procedure, the intended routes and relevance of routes can be derived from the cluster set. Then, intent inference step is implemented through comparing the probability summation of all flight attributes referencing to each candidate intended route. The notation and definition of flight attributes are given in Table 1. Based on the temporal property of the data, the attributes can be classified into static and dynamic categories. Moreover, the attributes can also be labeled as discrete data and continuous data.

For the discrete attributes including CS, DA, AS, AC, and RR, the probability is derived from its frequency in the data set defined as follows:

\[ P_i^k = \frac{\text{count} \left( A_i^k \right)}{\text{sum}_j \left( \text{count} \left( A_j^i \right) \right)}, \]  \hspace{1cm} \text{(5)}

where \( P_i^k \) is the probability of attribute \( i \) when it takes value of \( k \). The number of trajectory points associated to the attribute is derived from function count. Note that the probability of RR is defined as a conditional probability in (6), where the subsequent and preceding intended routes are denoted by \( C_m \) and \( C_n \) separately. The number of trajectory points associated to the clusters is counted based on the online trajectory clustering procedure:

\[ P_{RR}^{mn} = \frac{\text{count} \left( C_m, C_n \right)}{\text{count} \left( C_n \right)}. \]  \hspace{1cm} \text{(6)}

For the continuous attribute HA, the Gaussian random variable with zero mean and standard variance in [6] is adopted here, where \( \varphi_k \) is the angular difference between aircraft heading direction and direction to ending point of intended route denoted by \( C_k \):

\[ P_{HA}^k = \frac{1}{\sqrt{2\pi}\sigma} \exp \left( -\frac{\varphi_k^2}{2\sigma^2} \right). \]  \hspace{1cm} \text{(7)}

In order to determine the intended route for trajectory point \( T_j^i \) of trajectory \( i \) at time instant \( j \), the intended route set as the subset of cluster set \( C_j \subset C \) are investigated. For each intended route \( C_{kj} \in C_j, k = 1, \ldots, I_k \), the corresponding probabilistic summation is computed through the following
logarithmic format defined in (8), where $\omega_1, \ldots, \omega_6$ are the weight factors of attributes:

$$P(C_k | T_i^j) = \ln \left( (P_{CS})^{\omega_1} \cdot (P_{DA})^{\omega_2} \cdot (P_{AS})^{\omega_3} \cdot (P_{AC})^{\omega_4} \cdot (P_{RR})^{\omega_5} \cdot (P_{HA})^{\omega_6} \right)$$

$$= \omega_1 \ln P_{CS} + \omega_2 \ln P_{DA} + \omega_3 \ln P_{AS} + \omega_4 \ln P_{AC}$$
$$+ \omega_5 \ln P_{RR} + \omega_6 \ln P_{HA}. \quad (8)$$

Finally, the inferred intended route for trajectory point denoted by $k^*$ is recognized based on

$$k^* = \arg \max_{k=1, \ldots, T_k} P(C_k | T_i^j). \quad (9)$$

4. Experimental Results

4.1. Data Set. In the following experiments, radar trajectory and flight attribute data of 34 days from December 18, 2012, to January 20, 2013, collected from Chengdu terminal area are used. The radar trajectory data are limited in a circular area with radius 60 km centered in the position of Chengdu Shuangliu International Airport. The data consist of flight date and time, call sign, longitude, latitude, altitude, and ground speed with an update rate ranging from 1 to 5 seconds. Total 20854 primitive departure and arrival trajectories are collected. Based on the data preprocessing to eliminate some missing and fault data, 8995 departure valid trajectories with more than 250 trajectory points of each trajectory are used to verify the performance of the proposed approach. Note that rectangular coordinate system is applied in our experiments originated in the position of airport center. It is also worth pointing out that the same data set should be applied in the
Figure 7: Clustering results for departure trajectories are represented by blue lines. The radar trajectories are marked with grey lines.

whole validation process due to the continuity of the intent modeling and intent inference steps.

Flight attribute data are derived from the air traffic flow management system in Chengdu terminal area. The information contains flight date and time, call sign, destination airports, and aircraft type. Flight time and call sign are the keys to correlate the radar trajectory data. Here, 552 departure call signs are considered. The destination airports are divided into 7 categories based on the airport code and geographic information including northeast, north, northwest, southeast, east, southwest, and middle. Note that the international flights are classified based on their entering positions in the terminal area. Two kinds of aircraft size (heavy and medium) and three kinds of aircraft approach categories (B, C, and D) are considered.

4.2 Online Trajectory Clustering. Figure 6 reports the evolution of clustering process for departure trajectories. Plot (a) in Figure 6 shows the centroids of clusters starting from the runway represented by green lines. Subsequent clusters represented by blue and black lines are shown in plots (b) and (c). Plot (d) gives the final clustering results where the cluster is denoted by the original index, such as \( C_1 \) and \( C_{27} \). Note that clusters \( C_{14} \) and \( C_{26} \) have been deleted during the clustering process so that total 25 clusters are maintained. Furthermore, the clustering results are compared with the radar trajectories shown in Figure 7. In general, the clusters reflect the main routes. Note that some trajectories in the upper part of the figure are determined as outliers due to the low frequency in the data set.

We then investigate the available routes in different air traffic situations with 5 experiments shown in Figure 8. For each experiment, 2000 trajectories are selected randomly among the departure trajectories as the input to the clustering procedure. Clustering results are presented in plots (a), (b), (c), (d), and (e). Note that plot (b) shows the air traffic situation in which routes to the positive direction of \( X \) coordinate are rarely used. The situation shown in plot (d) reflects that flights are distributed in all directions.

4.3 T-AII with Multiple Attributes. By using the same trajectory data set and intended routes obtained in Section 4, eight main departure routes as shown in Figure 9 are selected as intended route set \( C_I \) to verify the T-AII approach, including \( C_{18}, C_{24}, C_2, C_{22}, C_9, C_{27}, C_{15}, \) and \( C_{20} \). These routes are organized in a binary-tree structure consisting of three intent inference scenarios shown in Figure 10. When aircraft takes off from the runway, the phase is marked as START state. The T-AII is triggered when the aircraft enters either route \( C_{18} \) or route \( C_{24} \). For example, based on trajectory data and flight attributes information collected in route \( C_{18} \), the first T-AII is to infer the likely intended routes (\( C_2 \) or \( C_{22} \)). Then, the second T-AII is followed if the aircraft is flying along route \( C_2 \) in which \( C_{15} \) and \( C_{20} \) are the possible intended routes. The third T-AII scenario refers to the case determining the possible route (\( C_9 \) or \( C_{27} \)) for aircraft in route \( C_{24} \).

Table 2 reports the intent inference accuracy of T-AII approach in three different cases (with only dynamic attributes, with only static attributes, and with both dynamic and static attributes), through adopting the following weight factors: \{0, 0, 0, 0, 0.5, 0.5\}, \{0.25, 0.25, 0.25, 0.25, 0.0, 0\}, and \{0.17, 0.17, 0.17, 0.17, 0.17, 0.17\} separately. Note that, in the maneuvering scenarios (turning), such as "\( C_{18} \) to \( C_{22} \)", "\( C_{24} \) to \( C_{9} \)" and "\( C_2 \) to \( C_{20} \)"”, the T-AII with only dynamic attributes results in quite low inference accuracy due to the ineffectiveness of dynamic attributes. This is because the dynamic attributes only contain short-term motional feature of aircraft. On the contrary, T-AII approach with only static attributes gains very high inference accuracy. One of the possible reasons is that static attributes consist of the long-term periodic characteristics for commercial flights, which are more efficient in the maneuvering scenario.
Figure 8: Clustering results for different air traffic situations.
Furthermore, the impact of each static attribute on inference accuracy is compared by independently applying the T-AII approach with each static attribute. Take the first scenario including “C18 to C2” and “C18 to C22” as an example. For the case “C18 to C2,” the lowest and the highest inference accuracy are 98.42% and 99.89% obtained from T-AII with DA and AS attributes, respectively. It can be seen that there is no obvious difference for each static attribute. However, the inference results of case “C18 to C22” are 87.40%, 20.18%, 1.30%, and 1.30% for the T-AII with CS, DA, AS, and AC, respectively. Here, the approaches with attributes AS and AC can gain the same accuracy level as that obtained with dynamic attributes shown in Table 2. On the contrary, the results by using attributes CS and DA are much better. One of the possible reasons is that the geographically related attributes DA and CS strongly affect the T-AII results in the turning maneuver case such as “C18 to C22.”

Then, confusion matrix in the first scenario is further investigated as shown in Table 3. It can be seen that most of trajectory points are associated with the straight route from C18 to C2 resulting in the high probability values for the two dynamic attributes (RR and HA) associated to “C18 to C2.” As a result, 13488 trajectory points are incorrectly determined to C2 rather than C22 based on T-AII approach with only dynamic attributes. However, when the static attributes are considered, the corresponding number of incorrect inferences is decreased to 1752.

5. Conclusion

This paper has introduced a two-step T-AII approach that integrated intent modeling and intent inference steps so as to address the uncertainties of air traffic situation in terminal area, especially the undefined flight routes and frequent maneuvers. The online trajectory clustering procedure is designed to recognize the real-time available routes merely based on radar trajectory rather than the predefined flight plan. Then, the T-AII approach with both dynamic and static attributes is proposed in which the corresponding inference accuracy in maneuvering scenario is increased. Although the results of the proposed approach are promising based on the verification with real radar trajectory and flight attributes data of 34 days collected from Chengdu terminal area, some
of the directions need to be further addressed in the future. For example, the outliers in the clustering procedure should be investigated in the perspective of air traffic controller. Also, more attributes related to flight should be discussed under the context of T-AII.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National Science Foundation for Young Scientists of China (Grants no. 61401011 and no. 61201314), the National ATC Scientific Research Project (Grant no. GKG201404004), and the National Key Technologies R&D Program of China (Grant no. 2012BAG04B01).

References


Research Article

An Airway Network Flow Assignment Approach Based on an Efficient Multiobjective Optimization Framework

Xiangmin Guan,1,2,3 Xuejun Zhang,1,2 Yanbo Zhu,1,2 Dengfeng Sun,4 and Jiaxing Lei1,2

1School of Electronic and Information Engineering, Beihang University, Beijing 100191, China
2National Key Laboratory of CNS/ATM, Beijing 100191, China
3Beijing Key Laboratory for Cooperative Vehicle Infrastructure Systems and Safety Control, Beijing 100191, China
4School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN 47907-2023, USA

Correspondence should be addressed to Xuejun Zhang; zhxj@buaa.edu.cn

Received 23 July 2014; Revised 7 September 2014; Accepted 18 September 2014

Academic Editor: Zheng Zheng

Copyright © 2015 Xiangmin Guan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Considering reducing the airspace congestion and the flight delay simultaneously, this paper formulates the airway network flow assignment (ANFA) problem as a multiobjective optimization model and presents a new multiobjective optimization framework to solve it. Firstly, an effective multi-island parallel evolution algorithm with multiple evolution populations is employed to improve the optimization capability. Secondly, the nondominated sorting genetic algorithm II is applied for each population. In addition, a cooperative coevolution algorithm is adapted to divide the ANFA problem into several low-dimensional biobjective optimization problems which are easier to deal with. Finally, in order to maintain the diversity of solutions and to avoid premature, a dynamic adjustment operator based on solution congestion degree is specifically designed for the ANFA problem. Simulation results using the real traffic data from China air route network and daily flight plans demonstrate that the proposed approach can improve the solution quality effectively, showing superiority to the existing approaches such as the multiobjective genetic algorithm, the well-known multiobjective evolutionary algorithm based on decomposition, and a cooperative coevolution multiobjective algorithm as well as other parallel evolution algorithms with different migration topology.

1. Introduction

With the development of civil aviation, the permanently increasing air traffic and the limited airspace resource have resulted in more and more serious congestion and flight delay [1–3]. Meantime, heavy congestion challenges the airspace safety and the flight delay costs the airline industry billions of dollars every year [4]. Hence, how to safely accommodate high levels of demand and to maximize the use of capacity-limited airspace and airport resources has become a major concern to both researchers and practitioners in air traffic management (ATM).

In recent years, FAA proposed the 4D-trajectory (4DT) operation concept in next generation air transportation system (NextGen), which includes three dimensions of position and a time description of the flight. Under the 4DT environment, flights can be accurately planned in both space and time. The airway network flow assignment (ANFA) approach can provide solutions from a global point of view with the aim of alleviating airspace congestion and reducing time cost by optimizing the 4D trajectories of all flights considered in the entire airspace. Therefore, it has become a focus of research interests.

However, the ANFA problem is a large-scale combinatorial optimization problem with complicated constraints as well as tightly coupled variables, which is difficult to solve in general. For instance, there are more than ten thousand flights flying over China every day on the air route network with more than one thousand waypoints, which generates a large number of tightly coupled decision variables and constraints. Hence, in order to get optimal flight plans for all flights, the ANFA problem has a very high computational complex. Besides, with consideration of reducing congestion and minimizing the induced delay, the problem usually has
multiple objectives which are all nondifferentiable or even noncontinuous.

Due to the importance of the ANFA problem, it has drawn a mass of attention of researchers [5]. Earlier work has focused on the slot-time allocation, known as ground holding in a single or multiple airport setting, in which delays propagate through the network [6, 7]. Besides, the flight level allocation has been considered to the flights in this problem [8, 9]. In addition, Bertsimas et al. [10, 11] considered the time slots and routes assignment simultaneously through an efficient deterministic approach. From a stochastic optimization point of view, Delahaye and Odoni proposed a genetic algorithm to solve the problem [12]. However, it often falls into the local optimum because of the large-scale decision variables. More recently, a cooperative coevolution (CC) algorithm was introduced into the resolution of the ANFA problem [13]. It adopted the divide-and-conquer strategy to divide the complex problem into several low-dimensional subproblems which becomes easier to deal with.

So far these works took the minimization of the airspace congestion or the flight delay as the sole objective [14, 15]. However, it might be more appropriate to consider both airspace congestion and extra flight cost and try to seek a good trade-off between them. Hence, Daniel et al. [16] formulated the problem into a biobjective optimization problem and solved it with the multiobjective genetic algorithm (MOGA). However, the application of existing multiobjective evolutionary algorithms (MOEAs) to the ANFA problem might lead to poor performance due to the scale of the problem. To the best of our knowledge, few works could effectively solve complicated multiobjective optimization problems with thousands of variables.

With the consideration of the reduction of the airspace congestion and the flight delay simultaneously, this paper formulates the ANFA problem as a multiobjective optimization model. Then, an efficient multiobjective optimization framework is presented to solve it. This framework employs the parallel computation and population diversity adjustment techniques to improve the optimization capability. Firstly, an effective multi-island parallel evolution algorithm (PEA) with multiple evolution populations is adopted. Besides, one-way ring migration topology is applied to exchange individuals among populations to improve the efficiency of the cooperation of populations. Secondly, the multiobjective evolutionary algorithm NSGA2 is used to optimize each population. In addition, a cooperative coevolution algorithm is adapted to divide the ANFA problem into several low-dimensional biobjective optimization problems which are easier to deal with. Finally, in order to maintain the diversity of solutions and avoid premature, a dynamic adjustment operator based on solution congestion degree is specifically designed. It can greatly improve the distribution of the nondominated solutions in the archive and maintain the population diversity by inheriting the nondominated solutions with low congestion degree in a high probability. Simulation results using the real traffic data from China air route network and daily flight plans demonstrate that the proposed approach can improve the solution quality effectively, showing superiority to the existing approaches such as the multiobjective genetic algorithm, the well-known multiobjective evolutionary algorithm based on decomposition (MOEA/D), and a CC-based multiobjective algorithm as well as other parallel evolution algorithms with different migration topology.

The rest of this paper is organized as follows. Section 2 introduces the formulation of the investigated biobjective ANFA problem. Section 3 describes the proposed multiobjective optimization framework in detail. Experimental study is presented in Section 4 to evaluate the effectiveness of our algorithm. Finally, Section 5 concludes this paper and discusses directions for further research.

2. Problem Formulation

The airway network can be modeled as a directed graph including the waypoints as nodes and segments as edges. Each flight can be considered as a single commodity on the network with a defined pair of origin-destination nodes. Besides, each flight has its predefined departure time slots and the optional routes. In this paper, with the consideration of safety and efficiency, the ANFA problem is formulated as a biobjective problem to reduce the airspace congestion and the total flight delay simultaneously.

Each flight (flight $k$) is associated with a pair of decision variables $(r_k, τ_k)$, where $r_k$ is a possible route and $τ_k$ is a feasible departure time slot. In addition, they are subject to the following:

1. $r_k ∈ Path_k = \{r_1, r_2, \ldots, r_{max}\}$, where Pathk is the set of all possible paths of flight $k$;
2. $t_{kmin} ≤ τ_k ≤ t_{kmax}$, where $t_{kmax}$ is the latest departure time slot and $t_{kmin}$ is the earliest departure time slot.

We can see that constraint (1) denotes a feasible route of each flight from a defined route set, while constraint (2) enforces the flight to depart at a predefined departure time slot.

The first objective function is the minimization of the airspace congestion. Here the workload of the sectors is used to indicate the airspace congestion, which is defined as follows.

It is generally known that the workload of a sector mainly depends on the monitoring workload and the coordination workload. Hence, the total workload of a sector $S_k$ at time $t$ can be roughly expressed by the following [16]:

$$W_{S_k}^t = W_{mosS_k}^t + W_{cosS_k}^t,$$

where $W_{mosS_k}^t$ is the monitoring workload and can be numerically estimated by

$$W_{mosS_k}^t = \begin{cases} 1 + M_{S_k}^t - C_{mS_k}^t, & \text{if } M_{S_k}^t > C_{mS_k}^t, \\ 0, & \text{else}, \end{cases}$$

where $M_{S_k}^t$ is related to the number of aircraft in sector $S_k$ at time $t$ and $C_{mS_k}^t$ is the monitoring critical capacity of the sector at time $t$. The Scientific World Journal
BEGIN
Initialize $M$ populations of size $p$ each. $g = 0$. Archive = NULL
WHILE $g < \text{max gen}$
    FOR population $j = 0: (M - 1)$
        Divide the problem into $n$ subcomponents based on cooperative co-evolution
        FOR each subcomponent
            Use the NSGA2 framework and differential evolution as the evolutionary algorithm to solve the subcomponent
        END FOR
        Archive = Obtain and update the non-dominated solutions based on the dynamic adjustment operator for solutions congestion degree
    END FOR
    IF (migration condition meet)
        Exchange individuals selected among populations based on a migration topology
    END IF
    $g = g + 1$
END WHILE
END

Algorithm 1: The framework of the proposed algorithm.

Besides, $W_{\text{co}}^{ct}$ is the coordination workload and can be numerically estimated by

$$W_{\text{co}}^{ct} = \begin{cases} 1 + C_t^{ct} - C_t^{ct}, & \text{if } C_t^{ct} > C_t^{ct}, \\ 0, & \text{else,} \end{cases} \quad (3)$$

where $C_t^{ct}$ is related to the flights passing through the boundary of sector $S_k$ at time $t$ and $C_t^{ct}$ is related to the number of aircraft passing the boundaries of sector $S_k$ at time $t$.

The goal of the optimization is to minimize the airspace congestion via spreading the congestion over several sectors. Hence, the objective is defined by the following [16]:

$$\min AC = \sum_{k=1}^{k=n_k} \left( \left( \sum_{t \in T} W_t^{ct} S_k \right)^{1-\varphi} \times \left( \max_{t \in T} W_t^{ct} S_k \right)^{\varphi} \right), \quad (4)$$

where $\varphi, (1 - \varphi) \in [0, 1]$ indicates the relative importance of the maximum congestion and the average congestion.

The second objective is the minimization of the extra flight cost (EFC) which includes the departure delay and the airborne delay caused by choosing a longer path than the shortest one. Then the second objective can be expressed as follows [16]:

$$\min EFC = \sum_{i \in F} \left( t_{r} - t_{prf} + (t_{r} - t_{prf}) \right)^2, \quad (5)$$

where $t_{r}$ and $t_{prf}$ denote the flight time of $r_i$ and the preferred shortest path, respectively, and $t_{prf}$ indicates the preferred departure time. The first part is the airborne delay, and the second part is the ground delay.

3. Optimization Method

It can be seen that the ANFA problem is a large-scale combinatorial optimization problem, and the objective functions are nonlinear and nondifferentiable. Black-box optimization methods such as EAs appeared to be promising to deal with this kind of problem [15–19]. Given the multiobjective model of the ANFA problem, basically any existing MOEA for discrete multiobjective optimization can be readily utilized. However, the ANFA problem involves thousands of tightly coupled decision variables which are difficult to solve. In addition, few existing MOEAs have been extensively investigated on problems of such a large scale while the scalability of EAs with respect to the number of decision variables is in general deemed to be poor; therefore a direct application of existing MOEAs might not obtain satisfactory solutions.

With the aim of avoiding prematurity and improving the convergence rate of this complex problem, a new multiobjective optimization framework is proposed. Firstly, an effective multi-island parallel evolution algorithm with multiple evolution populations is employed and one-way ring migration topology for exchange individuals among populations is applied to improve the efficiency of the cooperation of populations. Secondly, the multiobjective evolutionary algorithm NSGA2 is used to optimize each population. In addition, this paper introduces the idea of cooperative coevolutionary (CC) into the resolution of the multiobjective ANFA problems. The main idea is to divide the high-dimensional problem into low-dimensional subcomponents. The subcomponents work cooperatively to obtain better solutions. Finally, all the nondominated solutions are stored in an archive. As the ANFA problem is of such a large scale with thousands of variables, the main difficulty is how to maintain the diversity of the population to get better solutions and to improve the distribution of the solutions in the archive. Hence, in order to maintain the diversity of solutions and to avoid prematurity, a dynamic adjustment operator based on solution congestion degree is specifically designed for the ANFA problem. The framework is presented in Algorithm 1.
3.1. Multi-Island Parallel Evolution Algorithm. As parallel computers become more commonplace in scientific computing, it becomes more feasible to harness their power for use with evolutionary algorithms (EAs) [20]. Multi-island PEs consist of several populations, which can optimize simultaneously to avoid premature convergence. They have been successfully applied to find acceptable solutions to problems in different engineering domains [21, 22].

Suppose that there are $M$ islands and $N$ flights. Then each population can be denoted as

$$\text{pop}_i = \{i\text{div}_{i1}, i\text{div}_{i2}, \ldots, i\text{div}_{isp}\}, \quad 1 \leq i \leq M,$$

where $i\text{div}_{ij}$ is defined by

$$i\text{div}_{ij} = (r_{ij1}, r_{ij2}, r_{ij2}, \ldots, r_{ijN}, \tau_{ijN}), \quad 1 \leq j \leq \text{ps},$$

where $\text{ps}$ is the size of population.

3.2. Migration Topology. The migration topology is a key feature of the island model which determines the destination of the migrants, and it could greatly affect the quality of the solutions and the efficiency of algorithms. If two populations rarely communicate with each other, it is difficult for the best solution to spread which may prevent populations finding better solutions.

Currently, the main migration topologies are the one-way ring topology and the random topology [22]. The random topology delivers the migrants to a randomly selected population. However, it may result in inefficient communication. After many times of migration under the random topology, the best solution cannot be spread effectively among populations. In the one-way ring topology, populations are numbered, and the worst individual of a population is replaced by the best individual of the next population. The one-way ring topology can provide sufficient communication among populations and maintain the population diversity, which can effectively avoid premature convergence and improve the optimization capability.

3.3. Cooperative Coevolution For Each Population. Though the multi-island PEA uses several populations simultaneously, in fact each population is hard to avoid falling into a local optimum because the ANFA problem involves large-scale tightly coupled decision variables as presented in the previous sections. Hence, we introduce the cooperative coevolution algorithm for the optimization of each population to further improve the solution quality. The cooperative coevolution (CC) algorithm, adopting the divide-and-conquer strategy, divides the complex problem into several low-dimensional subproblems [23–25]. There are two critical issues in this approach.

(1) Decomposition Strategy. The decomposition strategy is a key feature of the cooperative coevolution framework which can greatly affect the capability and the efficiency of algorithms. In this work, the random grouping strategy is used which has been both theoretically and experimentally proved to be effective for the large-scale complex problem [23].

At each generation, each population is randomly divided into $ns$ disjoint subpopulations with the same population size:

$$\text{pop}_i = \{\text{sp}_1^i, \ldots, \text{sp}_{M}^i\}, \quad 1 \leq i \leq M$$

$$\text{sp}_i^j = \{s\text{id}_1^i, s\text{id}_2^i, \ldots, s\text{id}_{\text{ps}}^i\}, \quad 1 \leq j \leq ns$$

$$s\text{id}_j^i = (r_{j1}, r_{j2}, r_{j2}, \ldots, r_{jN}, (N/n_s)),$$

where $s\text{id}$ denotes the individual of each subpopulation and $\text{sp}_i^j$ indicates the subpopulation $j$ of population $i$.

(2) Subpopulation Optimization. Another critical point is optimization for each subpopulation. In this paper, the well-known multiobjective evolutionary algorithm NSGA2 is employed by each subpopulation [26]. Besides, differential evolution (DE) [27] is used in the framework to generate new solutions, because it is a simple yet effective algorithm for global optimization. DE is a randomized parallel searching algorithm. It begins with a random population, according to specific rules, for example, selection, crossover, and mutation. An optimized resolution is reached by retaining good individuals and discarding bad individuals. Compared with other optimization algorithms, DE has the advantages in global optimization as well as easy operation. Its operators are described below.

Suppose that $f(x)$ is the objective function and the goal is to minimize it.

In mutation, if current chromosome is $x_{ij,G}$, then choose three different chromosomes from current generation population named $x_{1j,G}$, $x_{2j,G}$, and $x_{3j,G}$. The mutation operator is defined by

$$V_{ij,G+1} = x_{ij,G} + F \cdot (x_{2j,G} - x_{3j,G}), \quad F \in [0, 2], \quad 1 \leq j \leq N,$$

where $F$ is a parameter which decides the scale of mutation. A new chromosome is generated by the crossover of $V_{ij,G+1}$ and $x_{ij,G}$ as follows:

$$U_{ij,G+1} = (u_{1j,G+1}, u_{2j,G+1}, \ldots, u_{Nj,G+1})$$

$$u_{jj,G+1} = \begin{cases} v_{jj,G+1}, & \text{rand}(j) \leq CR \lor j = \text{rnbr}(i) \\ x_{jj,G}, & \text{else} \end{cases},$$

where $\text{rnbr}(i)$ is a random integer number between 1 and $N$, which ensures that $U_{ij,G+1}$ gets at least one component from $V_{ij,G+1}$ component from $V_{ij,G+1}$ and $\text{rand}(j)$ is a uniformly distributed random number between 0 and 1.

After the evaluation of each chromosome, the chromosomes of next generation are chosen according to the following rule:

$$x_{ij,G+1} = \begin{cases} u_{ij,G+1}, & f(u_{ij,G+1}) \leq f(x_{ij,G}) \\ x_{ij,G}, & \text{else} \end{cases}$$

3.4. Dynamic Adjustment Operator Based on Solution Congestion Degree. As the optimization generation increases,
the population diversity decreases rapidly for more solutions congested at a local searching space [28, 29]. On the one hand, too crowded solutions will cause premature convergence. On the other hand, some local searching space with sparse solutions is not explored enough and needs more attention for better solutions. Hence, the more evenly the nondominated solutions distribute, the better the optimization is. In this paper, we propose a dynamic adjustment operator to improve the distribution of solutions and maintain the population diversity based on the congestion degree of solutions.

(1) Congestion Degree of the Nondominated Solutions in the Archive. The distance between solution $i$ and $j$ in archive is defined by

$$d_{i,j} = \frac{1}{2} \sqrt{(p_{i1}^{[1]} - p_{j1}^{[1]})^2 + (p_{i2}^{[2]} - p_{j2}^{[2]})^2}, \quad (12)$$

where $p_{i1}^{[1]}$ and $p_{i2}^{[2]}$ denote the value of the first and the second objective functions of solution $i$, respectively. Besides, the average distance of all solution pairs in the archive is described by

$$s = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} d_{i,j}, \quad (13)$$

where $n$ is the number of the solutions in archive.

Then the congestion degree of a nondominated solution can be expressed as

$$d_i = \frac{1}{n-1} \sum_{j=1, j\neq i}^{n} \delta_{i,j} \quad (14)$$

where

$$\delta_{i,j} = \begin{cases} 1 & \text{if } d_{i,j} \leq \gamma \cdot s \\ 0 & \text{if } d_{i,j} > \gamma \cdot s \end{cases}, \quad (15)$$

where $\gamma$ is a parameter which can be predefined. We can see that the larger $d_i$ is, the closer to other solutions solution $i$ is.

(2) Update the Archive. The solutions in the archive will be constantly updated during the evolution of the subpopulations. When the number of the nondominated solutions exceeds the maximum size of the archive, the solutions $i$ will be moved out from the archive in a probability

$$p_i = \frac{d_i^\alpha}{\sum_{j=1}^{n} d_j^\alpha}, \quad (16)$$

where $\alpha$ is a positive regulation factor. Equation (16) indicates that the higher $d_i$ is, the higher probability it will be moved out from the archive.

4. Experimental Studies

4.1. Database and Experimental Setup. The national route network of China consists of 1706 airway segments, 940 waypoints, and 150 airports. Note that the takeoff and landing phases of flights are truncated within a given radius (usually 10 NM) around airports. The traffic around airports is managed with specific procedures by the terminal control area (TCA) control services in these zones. The airspace is divided into many sectors, and Figure 1(a) shows the sectored airspace in China. The air traffic data was extracted from flight schedule database (FSD) of the summer in 2009 released by Civil Aviation Administration of China (CAAC). In order to better describe the difference between the algorithms’ performances, we consider two scenarios: 960 flights (the busiest one hour) and 1664 flights (the busiest three hours).

The parameters are set as follows: the number of populations $M = 5, \phi = 0.9, \rho = 0.1, r = 0.3, \gamma = 1/25$, and $ns = 10$. The mutation probability and the crossover probability of DE are 0.15 and 0.85.

The algorithms, such as our proposed method, MOGA, MOEA/D [30], and cooperative coevolution based algorithm,
in this work were implemented in C++, and the simulations were performed on a server with an E5620 2.4 GHz CPU with 12 GB RAM. For each algorithm, the results were collected and analyzed on the basis of 15 independent runs. Besides, the proposed approach was realized by multithreaded programming. Then, the optimization of all islands and all subcomponents of each population can proceed separately and simultaneously which can reduce the computation time. The parameters used in all experiments are listed in Table 1.

In order to evaluate the performance of the solutions obtained by each of the algorithms, three typical metrics are adopted: the convergence metric ($\gamma$) [26], the spread metric ($\Delta$) [31], and the hypervolume metric $I_H$ [32, 33]. $\gamma$ suggests the average Euclidean distance from the obtained nondominated solution set to the actual Pareto front. Note that it is difficult to find the actual Pareto front for most real-world optimization problems; so we use the best solution set obtained by these algorithms in 15 runs. $\Delta$ indicates the diversity of solutions along the Pareto front. $I_H$ can evaluate the convergence and the extent of spread of the solutions simultaneously without the real Pareto front. The smaller the first two indexes are, the better the algorithm is. On the contrary, the larger the third index is, the better the algorithm is.

### 4.2. Comparison with the Existing Methods.

In order to test the effectiveness of the proposed multi-island PEA framework, in this part, we will compare it with some existing algorithms, including the classical multiobjective genetic algorithm (MOGA), multiobjective evolutionary algorithm based on decomposition (MOEA/D), and a CC-based multiobjective algorithm (CCMA).

Tables 2 and 3 show the average value of $I_H$, $I_D$, and $\Delta$ over 15 independent runs of the algorithms for the two scenarios, respectively. In each row of the table, the best value is highlighted in boldface. It can be seen from the tables that PEA outperforms the other three algorithms in terms of $I_H$, $I_D$, and $\Delta$. Moreover, when the number of flights increases, PEA performs much better. It can be concluded that PEA has superiority to solve this large-scale problem.

Figure 2 shows the nondominated solutions obtained by the four algorithms. Specifically, the nondominated solutions of each algorithm were obtained over 15 runs. From Figure 2, it can be concluded that PEA performs the best because its solutions can dominate those obtained by other
Total delay

Airspace congestion \((n=960)\)

Total delay

Airspace congestion \((n=1664)\)

(a)

(b)

Figure 3: Comparison of PEA and PEA without DAO for 960 flights (a) and 1664 flights (b).

Table 3: Comparison of different algorithms for 1664 flights \((I_H, \gamma, \Delta)\).

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>(I_H)</th>
<th>(\gamma)</th>
<th>(\Delta)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOGA</td>
<td>3.3979e+12</td>
<td>2.1876e+07</td>
<td>1.0113</td>
</tr>
<tr>
<td>MOEA/D</td>
<td>7.4733e+13</td>
<td>1.1361e+07</td>
<td>1.0452</td>
</tr>
<tr>
<td>CCMA</td>
<td>1.8763e+14</td>
<td>4.7800e+06</td>
<td>0.9897</td>
</tr>
<tr>
<td>PEA</td>
<td>2.1909e+14</td>
<td>2.6247e+05</td>
<td>0.8029</td>
</tr>
</tbody>
</table>

Table 4: Comparison of different algorithms for 960 flights \((I_H, I_D, \Delta)\).

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>(I_H)</th>
<th>(I_D)</th>
<th>(\Delta)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PEA without DAO</td>
<td>9.2026e+10</td>
<td>7.0467e+04</td>
<td>1.2471</td>
</tr>
<tr>
<td>PEA</td>
<td>1.3937e+11</td>
<td>3.3957e+04</td>
<td>1.0050</td>
</tr>
</tbody>
</table>

Table 5: Comparison of different algorithms for 1664 flights \((I_H, I_D, \Delta)\).

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>(I_H)</th>
<th>(I_D)</th>
<th>(\Delta)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PEA without DAO</td>
<td>7.3890e+6</td>
<td>3.4212e+05</td>
<td>0.9395</td>
</tr>
<tr>
<td>PEA</td>
<td>1.3107e+7</td>
<td>1.2861e+05</td>
<td>0.7091</td>
</tr>
</tbody>
</table>

Tables 4 and 5 show the results obtained by PEA and PEA without DAO in terms of the values of the metrics over 15 independent runs of the algorithms when the number of flights is 996 and 1664, respectively. It is seen from the tables that PEA always outperforms the other algorithm in terms of \(I_H, I_D,\) and \(\Delta\).

Furthermore, like the first experiment, the nondominated solutions of the algorithms are shown in Figure 3. It shows that PEA performs much better than the other method and its nondominated solutions can dominate the solutions obtained by PEA without DAO. For the scenario of 1664 flights, PEA has the most nondominated solutions and spreads nicely in the objective space.

As the optimization generation increases, more solutions will be congested in some local searching space which may cause premature convergence. The dynamic adjustment operator based on the congestion degree can effectively improve the distribution of solutions by inheriting the nondominated algorithms. Besides, it can be seen that MOGA has the worst performance, and CCMA performs better than MOEA/D, but MOEA/D has good performance in terms of diversity.

From the experimental results, we find that PEA performs better than the other three methods for the two scenarios. It adopts an effective multi-island parallel evolution framework which can improve the optimization capability. Besides, the one-way ring migration strategy can further avoid premature. In addition, this paper introduces the cooperative coevolutionary (CC) into each population optimization via dividing the high-dimensional problem into low-dimensional subcomponents. The subcomponents work cooperatively to obtain better solutions. With the help of parallel computation, the computation is just about 30 minutes which is feasible for the ANFA problem.

4.3. Investigation of the Effectiveness of the Dynamic Adjustment Operator. In the previous section, the first set of experiments has justified the superiority of PEA to existing methods. The next experiment is designed to further investigate whether the dynamic adjustment operator (DAO) based on solution congestion degree contributes to the success of PEA.
solutions with low congestion degree in a high probability. Hence, it can avoid decreasing the diversity of all populations.

5. Conclusion and Future Work

With the aim of reducing the airspace congestion and the flights delay simultaneously, this paper formulates the airway network flow assignment (ANFA) problem into a multiobjective optimization model and presents a new multiobjective optimization framework to solve it. Firstly, an effective multi-island parallel evolution algorithm is employed to solve the problem by multiple evolution populations. Besides, one-way ring migration topology is applied to improve the efficiency of the cooperation of populations by exchanging individuals among populations. Secondly, the multiobjective evolutionary algorithm NSGA2 is used to optimize each population. In addition, a cooperative coevolution algorithm is adapted to improve the optimization capability by dividing the ANFA problem into several low-dimensional biobjective optimization problems. Finally, in order to maintain the diversity of solutions and avoid premature, a dynamic adjustment operator based on solution congestion degree is specifically designed. Simulation results using the real traffic data from the China air route network and daily flight plans demonstrate that the proposed approach can improve the solution quality effectively, showing superiority to the existing approaches such as the multiobjective genetic algorithm, the well-known multiobjective evolutionary algorithm based on decomposition, and a CC-based multiobjective algorithm as well as other parallel evolution algorithms with different migration topology. For future research, the ANFA problem with the influence of severe weather will be considered.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the National High Technology Research and Development Program of China (Grant no. 2011AA110101), the National Basic Research Program of China (Grant no. 2010CB731805), and the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (Grant no. 61221061).

References


Research Article

Helicopter Control Energy Reduction Using Moving Horizontal Tail

Tugrul Oktay and Firat Sal

College of Aviation, Erciyes University, 38039 Kayseri, Turkey

Correspondence should be addressed to Tugrul Oktay; tugruloktay52@gmail.com

Received 7 November 2014; Revised 14 February 2015; Accepted 1 April 2015

Academic Editor: Zheng Zheng

Helicopter moving horizontal tail (i.e., MHT) strategy is applied in order to save helicopter flight control system (i.e., FCS) energy. For this intention complex, physics-based, control-oriented nonlinear helicopter models are used. Equations of MHT are integrated into these models and they are together linearized around straight level flight condition. A specific variance constrained control strategy, namely, output variance constrained Control (i.e., OVC) is utilized for helicopter FCS. Control energy savings due to this MHT idea with respect to a conventional helicopter are calculated. Parameters of helicopter FCS and dimensions of MHT are simultaneously optimized using a stochastic optimization method, namely, simultaneous perturbation stochastic approximation (i.e., SPSA). In order to observe improvement in behaviors of classical controls closed loop analyses are done.

1. Introduction

Traditionally in order to control helicopters, collective and cyclic (i.e., longitudinal and lateral) rotor blade pitches are used. Presently almost all of the helicopters employ a swash-plate mechanism and pitch links (it consists of two circular plates and a ball bearing arrangement separating them; see [1] for more details) to transmit two cyclic and collective pitch commands to the blade root. However, this mechanism is heavy and complex and also causes important drag during high flight speeds. Throughout history some other control methods have been considered in order to avoid these drawbacks and also for some other reasons such as reduction of control energy and redundancy in case of failures. Some of these alternatives are using trailing edge flaps (TEFs) with (see [2–4]) and without (see [5–7]) classical swashplate mechanism, passive (see [8–10]) and active (see [11–13]) helicopter morphing, and MHT (see [14–18]). For example, in [2] TEFs were integrated into blades for the case of a failure of the pitch link making the blade free float in pitch. By this method catastrophic results of a pitch link failure were corrected. In [5] TEFs were replaced with a conventional swashplate mechanism. Via eliminating swashplate mechanism and using TEFs, important reduction in weight, drag, and cost and also improvement in rotor performance were obtained. Moreover, in [8] passive morphing was used in order to reduce helicopter FCS energy. In that study many blade parameters (e.g., blade length and blade chord length) were simultaneously optimized with helicopter FCS parameters in order to save FCS energy. Substantial reduction in helicopter FCS energy was obtained using passive morphing idea. In [11] active morphing was used to save helicopter FCS energy. In that study actively morphing parameters were blade chord length, blade length, blade twist, and main rotor angular speed. The main difference between this and previous study (i.e., passive morphing) was that for the active case the helicopter design parameters are able to change (except helicopter FCS) during flight, but in prescribed interval. Using active morphing idea significant reduction in helicopter FCS energy was obtained. MHT idea was firstly studied in [14] in 1953. In this study differential control of each side of a canted horizontal tail was permitted. In [15] collective control of horizontal tail was mechanically achieved. In this study it was claimed that a fixed horizontal tail is advantageous in order to improve longitudinal stability of helicopters in forward flight, but it is not enough during gliding and climbing flights. More recently in [16] a moveable horizontal tail to give the desired attitude at different flight speed for UH–60 was designed. Recently
in [17, 18] a moveable horizontal tail was designed in order to reduce TEF deflection for swashplateless helicopters since the stroke capacity of existing smart material actuators is not enough for the required TEFs inputs. Using MHT TEFs deflections were relieved.

Numerous helicopter FCS design methods have been studied throughout the years, in historical sequence classical pole placement techniques (see [19, 20]), simple feedback control methods (see [21, 22]), and modern control approaches depend on linear matrix algebra such as linear quadratic regulator (LQR) and linear quadratic Gaussian (LQG) techniques (see [23, 24]), $H_{\infty}$ control synthesis (see [25, 26]), and model predictive control (MPC) (see [27, 28]). In this paper, a modern constrained control method, namely, OVC, is chosen for the design of helicopter FCS. OVC has many advantages with respect to the other control strategies existing in the literature. First, these controllers are modified LQG controllers and they benefit from Kalman filters as state estimators. Second, variance constrained controllers apply second-order information (i.e., state covariance matrix; see [29, 30] for details) and this kind of information is very beneficial during multivariable control system design because all stabilizing controllers are parameterized in relation to the physically meaningful state covariance matrix. Last, for large and strongly coupled multi-input, multi-output (MIMO) systems as in air vehicles control and especially in this paper, variance constrained control methods give guarantees on the transient behavior of independent variables by enforcing upper limits on the variance of these variables.

Variance constrained controllers have been used for many aerospace vehicles (e.g., helicopters, see [8, 11, 31–36]; tiltrotor aircraft, see [37]; Hubble space telescope, see [38]; tensegrity structures, see [39]) in last thirty years. For instance, in [32] variance constrained controllers were used for helicopter FCS during maneuvers, specifically level banked turn and helical turn. In that paper, performance of them was also considered during failures of some helicopter sensors. Reasonable consequences (meaning that variance constraints on outputs/inputs were satisfied and also closed loop systems were exponentially stabilized) were found in terms of helicopter FCS. Robustness of the closed loop systems (obtained via integration of linearized helicopter model and FCS) with respect to some modeling uncertainties (i.e., variation of flight conditions and all helicopter inertial parameters) was also studied and it was found that these controllers have stability robustness with respect to modeling uncertainties.

In this paper, MHT is for the first time simultaneously designed with helicopter FCS. For this purpose, a specific variance constrained controller OVC is also used for the first time for FCS. It is important to note that when MHT is integrated with classical helicopter, the number of controls increases. This causes an important result. The number of trim unknowns increases with additional MHT controls. Nevertheless, there are no additional trim equations. Therefore, in order to solve the resulting nonlinear trim equations, a useful optimization algorithm is required. For its solution, a stochastic optimization method specifically simultaneous perturbation stochastic approximation (i.e., SPSA) (see [40, 41] for brief description of SPSA) is for the first time applied for the simultaneous trimming and FCS design problem since it is computationally cheap and effective during solving constrained optimization problems when it is impossible to compute derivatives such as gradients and Hessians, analytically as in the situation herein. This paper first presents helicopter models used for simultaneous MHT and FCS design. Second, MHT is illustrated and motions of it are described. Then, definition of applied FCS (i.e., OVC) is given briefly. After that, trimming the system (i.e., the one obtained via integration of helicopter, MHT, and FCS) via simultaneous trimming and FCS design idea is explained. Then, the specific optimization method, namely, SPSA, applied in order to trim the system is summarized. Finally, this simultaneous design idea is applied for Puma SA 330 helicopter and closed loop responses of classical helicopter and helicopter with MHT are compared.

2. Helicopter Model

The modeling approach of used helicopter models in this paper is presented in detail in [31, 42]. The essential modeling assumptions are given next. First of all, multibody system approach was used to include all helicopter components: fuselage, horizontal tail, tail rotor hub and shaft, landing gear, and fully articulated main rotor with 4 rigid blades with blade flapping and lagging hinges. Secondly, a static inflow formulation (i.e., Pitt-Peters formulation) was applied for helicopter main rotor downwash. Thirdly, linear incompressible aerodynamics was used for the main rotor blades, but an analytical formulation was applied for the modeling of fuselage.

The modeling procedure requires using physics principles and because of the assumptions described in the previous paragraph it directly led to helicopter dynamic models that consisted of finite sets of ordinary differential equations (ODEs). This mathematical structure is fairly beneficial for control system design since it assists the direct use of modern control theory, which relies on state space representations of the system’s dynamics, easily obtained from ODEs.

The modeling methodology summarized above was applied in Maple and it led to a nonlinear helicopter model in implicit form:

$$f(\dot{x}_n, x_n, u_n) = 0,$$

where $f \in \mathbb{R}^{28}$, $x_n \in \mathbb{R}^{25}$, and $u_n \in \mathbb{R}^{4}$. Here $x_n$ and $u_n$ are nonlinear state and control vectors, respectively, and $\mathbb{R}^{s}$ represents the linear space of $s$-dimensional real vectors, where “$s$” can be 28, 25, or 4. It should be noted that the inconsistency between the size of $f(28)$ and the size of $x_n(25)$ is due to the three static downwash equations. The 28 nonlinear equations in (1) are categorized as follows: 9 fuselage equations, 8 blade flapping and 8 blade lead-lagging equations, and 3 static main rotor downwash equations. The helicopter models obtained have too many terms, making its use in fast computation impractical. For that reason, a systematic model simplification technique, named ordering scheme, was applied to reduce the number of terms in the nonlinear ODEs. The ordering scheme iteratively deletes terms from an equation depending on their relative magnitude with respect to the other terms in that equation. Each term’s magnitude is
motion, 17 trim equations were found (i.e., flight conditions were applied for the nonlinear equations of considered are straight level flights. When the straight level equations). In this paper, for FCS design the nominal trajectories reasonably complex (i.e., with a total of 28 nonlinear equations). These equations were solved using MATLAB after trimming, the number or type of equations generated using physics principles; it just shortens the equations by retaining the dominant terms.

The model found after using the ordering scheme is still reasonably complex (i.e., with a total of 28 nonlinear equations). In this paper, for FCS design the nominal trajectories considered are straight level flights. When the straight level flight conditions were applied for the nonlinear equations of motion, 17 trim equations were found (i.e., 0 = 0 equations were deleted). These equations were solved using MATLAB for different straight level flight speeds. After trimming, the model was linearized using Maple, yielding continuous linear time-invariant (LTI) systems:

\[ \dot{x} = A x + B u, \]  
\[ (2) \]

Here \( x_p \) and \( u_p \) are the perturbed state and perturbed control vectors. Matrices \( A_p \) and \( B_p \) are of size \( 25 \times 25 \) and \( 25 \times 4 \). The state vector consists of 9 fuselage states, 8 blade flapping states, and 8 blade lead-lagging states. The control vector includes 3 main rotor controls (collective, \( \theta_c \), longitudinal cyclic, \( \theta_L \), and lateral cyclic, \( \theta_r \), blade pitch angles) and 1 tail rotor control (collective, \( \theta_T \)).

Puma SA 330 helicopter (see [31, 43]) was used to validate the models used in this paper. These models are leading to acceptable agreement on trim values, flight dynamics modes, and qualitatively similar flapping and lead-lagging mode behavior (see [43]). In Table 1 and Figure 1 some validation results show how the models correctly capture the dynamics of Puma SA 330 helicopter (see [31] for more validation data). For instance, most of the flight dynamics modes (linearized system eigenvalues) of the models for hover and straight level flights (i.e., 40 kts and 80 kts) match well the results reported in [43]. The mode displaying the largest discrepancy is the 4th mode (it is important to note that this is due to modeling discrepancy between the models used and [43]); nevertheless, the qualitative behavior is similar (they are both exponentially stable modes).

The qualitative behaviors of the blade flapping and lead-lagging modes are also identical with the ones given in [43] that the blade flapping modes are much farther away from the imaginary axis with respect to the blade lead-lagging modes and the magnitude of the frequency bound for the blade flapping modes is larger than the one for the blade lead-lagging modes (see Figure 2).

It is also required to note that all trim results obtained using our model also showed good correspondence with data given in the literature (see [31] for more details). For instance, the trim values for straight level flight at 40 kts were

\[ 40 \text{ kts } \mathbf{x}_0 \]
\[ = \begin{bmatrix} 0.2753, 0.0370, -0.0908, 0.4857, -0.0456, 0.0272, \\ -0.0795, 0.0592, 0.0252, 0, 0.0218, 0.0010, 0, 0.0082, 0, \\ -0.1252, 6.2236, 9.2217 \end{bmatrix}^T \]

(3)

Here \( \theta_c, \theta_L, \theta_r, \theta_T \), \( \beta_p, \beta_c, \beta_L, \beta_r \), and \( \phi_p, \phi_c, \phi_L, \phi_r \) vectors are trim values of conventional helicopter controls, blade flapping angles, blade lead-lagging angles, and Euler angles, respectively, and all are given in unit of radians. The trim vector of linear downwash is \( \{ \lambda_0, \lambda_{\phi_p}, \lambda_{\phi_c}, \lambda_{\phi_L}, \lambda_{\phi_r}, \lambda_{\beta_p}, \lambda_{\beta_c}, \lambda_{\beta_L}, \lambda_{\beta_r}, \lambda_{\theta_c}, \lambda_{\theta_L}, \lambda_{\theta_r}, \lambda_{\theta_T} \} \) where \( \lambda_{\phi_p}, \lambda_{\phi_c}, \lambda_{\phi_L}, \lambda_{\phi_r} \) are trims of collective and longitudinal cyclic downwash in m/s and \( \lambda_0 \) is the trim of wake skew angle given in radians.

Table 1: Flight dynamics modes comparison.

<table>
<thead>
<tr>
<th>Mode (rad/s)</th>
<th>( V_A = ) hover</th>
<th>( V_A = 40 \text{ kts} )</th>
<th>( V_A = 80 \text{ kts} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>0.2772 ± 0.5008i</td>
<td>-0.1543 ± 0.918i</td>
<td>-0.0434 ± 1.0846i</td>
</tr>
<tr>
<td>2nd</td>
<td>-0.0410 ± 0.5691i</td>
<td>0.0275 ± 0.3185i</td>
<td>-0.0143 ± 0.3253i</td>
</tr>
<tr>
<td>3rd</td>
<td>-0.2697</td>
<td>-0.0976</td>
<td>-0.0703</td>
</tr>
<tr>
<td>4th</td>
<td>-0.3262</td>
<td>-1.9944</td>
<td>-0.9817</td>
</tr>
<tr>
<td>5th</td>
<td>-1.2990 ± 0.2020i</td>
<td>-0.6536 ± 0.3536i</td>
<td>-0.6125 ± 0.2798i</td>
</tr>
</tbody>
</table>

(Padfield results are taken from [43]).
3. Illustration of MHT

MHT angles (i.e., collective and differential) are illustrated in Figure 3. Collective motion refers to the movement of left and right horizontal tails in the same direction and magnitude simultaneously. On the other hand, differential motion refers to the movement of them in the opposite direction and the same magnitude simultaneously.

Angle of attack for left and right horizontal tails is calculated using

\[
\alpha_{tp} = \alpha_p + \eta_0 + \eta_d,
\]

\[
\alpha_{tp} = \alpha_p + \eta_0 - \eta_d,
\]

where \(\alpha_p\) is the angle of attack of classical fixed horizontal tail and \(\alpha_{tp}\) and \(\alpha_{tp}\) are angle of attack for moving right and left horizontal tails, respectively. The last MHT control (i.e., \(i_o\)) is the control parameter of distance between helicopter center of gravity (cg) and horizontal tail (HT). The distance between cg and HT is found to be multiplying distance control parameter with the classical helicopter cg-HT distance.

4. Flight Control System (FCS)

For FCS, a variance constrained controller specifically output variance constrained control (OVC) is chosen. The OVC problem's description is given next.

For a given continuous linear time invariant (LTI) system

\[
\dot{x}_p = A_p x_p + B_p u_p + w_p, \quad y = C_p x_p, \quad z = M_p x_p + v
\]

and a positive definite input penalty matrix \(R > 0\), find a full order dynamic controller

\[
\dot{x}_c = A_c x_c + Fz, \quad u_p = Gx_c
\]

to solve the problem

\[
\min_{A_c, F, G} J = E_{\infty} u_p^T R u_p = \text{tr} (RG \Phi G^T)
\]

subject to

\[
E_{\infty} y_i^2 \leq \sigma_i^2, \quad i = 1, \ldots, n_y,
\]

where \(z\) represents sensor measurements, \(w_p\) and \(v\) are zero-mean uncorrelated Gaussian white noises with intensities \(W\) and \(V\), respectively, \(\sigma_i^2\) is the upper bound imposed on the \(i\)th output variance, and \(n_y\) is the number of outputs. The quantity \(J = E_{\infty} u_p^T R u_p\) is referred to as the control energy (or cost) and \(\Phi\) is the state covariance matrix computed using the OVC algorithm (see [44, 45]). Here \(E_{\infty} \triangleq \lim_{t \to \infty} E\) is the expectation operator. The solution to the OVC problem is obtained from a linear quadratic Gaussian (LQG) problem by choosing appropriately the output penalty \(Q > 0\). Specifically, \(Q\) is dictated by the constraints imposed on the output variances (i.e., \(\sigma_i^2\) in (8)) and it can be obtained using the iterative algorithm described in [44, 45]. After the algorithm converges and \(Q\) is found, the OVC parameters are computed using

\[
A_c = A_p + B_p G - FM_p,
\]

\[
F = XM_p^T V^{-1},
\]

\[
G = -R^{-1} B_p^T K,
\]

where \(X\) and \(K\) are obtained by solving the following two algebraic Riccati equations:

\[
0 = X A_p^T + A_p X - X M_p^T V^{-1} M_p X + W, \quad (10a)
\]

\[
0 = K A_p + A_p^T K - K B_p R^{-1} B_p^T K + C_p^T Q C_p, \quad (10b)
\]

Clearly, compared to LQG where the penalties are selected ad hoc, OVC has the advantage that the penalty \(Q\) is selected such that output variance constraints are satisfied.
convergence of SPSA was theoretically proved. Its short
46)]. Moreover, under certain conditions (see [41]) strong
solving constrained optimization problems (see [8, 11, 31, 41,
to estimate the gradient (see [40]). It is also successful in
expensive because it uses only two evaluations of the objective

5. Trimming and Simultaneous
MHT and FCS Design

Now it is required to define the simultaneous trimming and
FCS design problem for the helicopter with MHT. This prob-
lem makes use of the extra number of trim unknowns (i.e.,
the 3 MHT control trims) and the ability to create helicopter
linearized state-space models in terms of these MHT control
trims. Let \( x = \{\eta_0, \eta_d, l_0\} \) be the set of MHT control trims. The
problem of finding optimum trim values for MHT controls
can be obtained via changing the traditional OVC design
problem summarized in Section 4 if the dependencies \( A_p(x), B_p(x) \) are considered. It is important to note that here \( x \)
denotes the MHT controls trim values. During the control
problem formulation, \( u_p \) represents perturbed control vector
and includes the MHT controls. The FCS energy in (7) and
the expected values (i.e. \( E_{oc}u_p^T \), \( i = 1, \ldots, n_p \) in (8) are
now function of these MHT control trims also, in addition of
the control matrices \( (A_p, F, G) \). Therefore, the following
optimization problem is created:

\[
\min_{A_p, F, G, x} J = E_{oc}u_p^TRu_p
\]

subject to (5), (6), and (8). Furthermore, the components of
\( x \) are constrained (i.e., \( x_{min} \leq x \leq x_{max} \), see Table 2). This
new optimization problem is much more complicated than
traditional OVC design and how to solve it is discussed next.

6. Simultaneous Perturbation Stochastic
Approximation (SPSA)

The problem of finding the optimum values of the MHT
control trims during the simultaneous trimming and FCS
design problem summarized in Section 5 is much more diffi-
cult than the traditional OVC design due to the introduction
of the additional MHT trim optimization variables and the
associated constraints on them. Since there is complex depen-
dency between \( J \) and expected values of outputs of inter-
rest, computation of their derivatives with respect to these
variables is analytically impossible. This recommends the
application of certain stochastic optimization techniques.
In order to solve it, a stochastic optimization method, namely,
SPSA, is chosen. This method was successfully used in similar
complex constrained optimization problems (see [8, 11, 31,
41]) before. SPSA has many advantages. First, SPSA is inex-
pensive because it uses only two evaluations of the objective
to estimate the gradient (see [40]). It is also successful in
solving constrained optimization problems (see [8, 11, 31, 41,
46]). Moreover, under certain conditions (see [41]) strong
convergence of SPSA was theoretically proved. Its short
summary is given next.

<table>
<thead>
<tr>
<th>MHT parameters</th>
<th>Nominal trim value</th>
<th>Lower bound ( \Delta x_i/x_i )</th>
<th>Upper bound ( \Delta x_i/x_i )</th>
<th>Optimum trim value</th>
<th>Change ( \Delta x_i/x_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \eta_0 )</td>
<td>0 rad</td>
<td>-0.05</td>
<td>0.05</td>
<td>0.5022 rad</td>
<td>—</td>
</tr>
<tr>
<td>( \eta_d )</td>
<td>0 rad</td>
<td>-0.05</td>
<td>0.05</td>
<td>-0.5037 rad</td>
<td>—</td>
</tr>
<tr>
<td>( l_0 )</td>
<td>3.80 m</td>
<td>-0.05</td>
<td>0.05</td>
<td>2.9317 m</td>
<td>-0.2285</td>
</tr>
</tbody>
</table>

Let \( x \) denote the vector of optimization variables. For the
classical SPSA, if \( x_{[k]} \) is the estimate of \( x \) at \( k \)th iteration, then

\[
x_{[k+1]} = x_{[k]} - a_k g_{[k]} \tag{12}
\]

where

\[
g_{[k]} = \left[ \frac{\Gamma_x - \Gamma_-}{2d_k \Delta_{[k]}} \ldots \frac{\Gamma_x - \Gamma_-}{2d_k \Delta_{[k]}} \right]^T, \tag{13}
\]

\( a_k \) and \( d_k \) are gain sequences, \( g_{[k]} \) is the estimate of the
objective's gradient at \( x_{[k]} \), \( \Delta_{[k]} \in \mathbb{R}^p \) is a vector of \( p \) mutually
independent mean-zero random variables \( \{\Delta_{[k]_1}, \ldots, \Delta_{[k]_p}\} \)
satisfying certain conditions (see [47, 48]), and \( \Gamma_x, \Gamma_- \) and \( \Gamma_+ \)
are estimates of the objective evaluated at \( x_{[k]} + d_k \Delta_{[k]} \) and
\( x_{[k]} - d_k \Delta_{[k]} \), respectively. An adaptive algorithm considering
the requirement that the optimization variables must be
between lower and upper bounds was previously developed
and combined with OVC to solve the simultaneous actively
and passively morphing helicopter and FCS design problem
(see [8, 11]). The adaptation is via the gain sequences, \( a_k \) and
\( d_k \), and they are

\[
a_k = \min \left\{ a : 0.95 \min \left\{ \min \left( \vartheta_i \right), \min \left( \vartheta_i \right) \right\} \right\},
\]

\[
d_k = \min \left\{ \frac{d}{k^{0.9}}, 0.95 \min \left\{ \min \left( \vartheta_i \right), \min \left( \vartheta_i \right) \right\} \right\},
\]

where \( \vartheta_i \) and \( \vartheta_u \) are vectors whose components are \( (x_{[k]_i} -
\min) / \Delta_{[k]} \) for each positive \( \Delta_{[k]} \), \( (x_{[k]_i} - \max) / \Delta_{[k]} \) for
each negative \( \Delta_{[k]} \), respectively. Similarly, \( \vartheta_i \) and \( \vartheta_u \) are
vectors whose components are \( (x_{[k]_i} - \min) / g_{[k]} \) for each
positive \( g_{[k]} \) and \( (x_{[k]_i} - \max) / g_{[k]} \) for each negative \( g_{[k]} \),
respectively, and \( d, a, \alpha, \Phi, \Theta, \) and \( S \) are other SPSA parameters.
The reader interested in the details of this algorithm is
referred to [8, 11, 31].

In order to solve the simultaneous trimming and control
design problem for optimal MHT trim values, the following
algorithm is used in this paper.

\textbf{Step 1.} Set \( k = 1 \) and choose initial values for the optimization
parameters, \( x = x_{[1]} \), and a specific flight condition (e.g., \( V_A =
40 \text{ kts straight level flight} \)).

\textbf{Step 2.} Compute \( A_p \) and \( B_p \), design the corresponding OVC
using (9), (10a), and (10b), and find the current value of the
objective, \( \Gamma_k \) using (11); note that \( \Gamma_k = J_k \) for OVC.

\textbf{Step 3.} Perturb \( x_{[k]} \) to \( x_{[k]} + d_k \Delta_{[k]} \) and \( x_{[k]} - d_k \Delta_{[k]} \) and
solve the corresponding OVC problems to find \( \Gamma_+ \) and \( \Gamma_- \),
respectively. Then compute the approximate gradient, $g_k$, using (13) with $d_k$ given by (14).

**Step 4.** If $\|a_k g_k\| < \delta x$, where $a_k$ is given by (14) and $\delta x$ is the minimum allowed variation of $x_k$, or $k + 1$ is greater than the maximum number of iterations allowed, exit, else calculate the next estimate of $x$, $x_{k+1}$, using $x_{k+1} = x_k - a_k g_k$, set $k = k + 1$, and return to Step 2.

### 7. Results

#### 7.1. Helicopter FCS Energy Saving

It is first required to note that, for all of the numerical results reported in this paper, the sensor measurements ($z$ in (5)) were helicopter linear velocities, angular velocities, and Euler angles. The outputs of interest ($y$ in (5)) were the helicopter Euler angles. The tolerance used for all of the OVC designs was $10^{-7}$. Firstly, the nonlinear helicopter model including MHT was trimmed using simultaneous trim and FCS design idea. The output variance constraints on helicopter Euler angles were $\sigma^2 = 10^{-4}[1 \ 1 \ 0.1]$ while the inputs of interest were all traditional helicopter controls (i.e., 3 main rotor controls and 1 tail rotor control) and the additional MHT controls (i.e., collective and differential control and distance control parameter). The helicopter FCS energy obtained after simultaneous trimming and FCS design is labeled as $J_n$. Secondly, for the same flight condition, the same outputs and inputs of interest and the same constraints OVC were redesigned for the helicopter without MHT. The resulting helicopter FCS energy is labeled as $J_n$. In order to see the benefits of using MHT on helicopters, the relative variation of the helicopter FCS energy, $\gamma J$, was computed using $\gamma J = 100(J_n - J_f)/J_n$.

The adaptive SPSA algorithm summarized in Section 6 was applied in order to solve the simultaneous trimming and design problem using the SPSA parameters of $S = 5$, $\lambda = 0.602$, $a = 500$, $d = 20$, and $\Theta = 0.101$ via MATLAB software. For this design problem the algorithm was very effective in rapidly decreasing the helicopter FCS energy, $J$, converging quickly to a stable value, as seen in Figure 4 (see Table 2 for optimum MHT control trim values). Moreover, the FCS energy corresponding to the system obtained using simultaneous trimming and design was 59.4% lower than the FCS energy of system obtained using classical helicopter and traditional OVC (meaning that $\gamma J = 59.4\%$). The vector of trim values obtained after applying simultaneous trimming and design situation was

$$
\begin{bmatrix}
\phi_0, \theta_0, \phi_0, \theta_0, \\
-0.0620, 0.070, 0.0849, 0.1348, -0.0027, 0,
\end{bmatrix}
$$

In Figure 5, closed loop responses of helicopter Euler angle states are given when the 1st closed loop system (solid black line) and 2nd closed loop system (solid blue line) are both excited by white noise perturbations. From Figure 3 it can be easily seen that, for both classical helicopter and helicopter with MHT, the qualitative (i.e., shape of the response) and quantitative (i.e., magnitude of the response) behaviors of Euler angles are basically the same. This can be explained using the fact that the expected values ($E_{\infty} y_f^T$) of outputs of interest (i.e., helicopter Euler angles in this paper) are very close and satisfy the constraints ($E_{\infty} y_f^T \leq \sigma_f^2$).

In Figure 6, closed loop responses of helicopter linear and angular velocity states are given for the 1st closed loop system (solid black line) and 2nd closed loop system (solid blue line). Figure 6 shows that the linear and angular velocity states do not experience catastrophic behavior (meaning that fast and large variations do not occur). For both classical helicopter and helicopter with MHT, qualitative behaviors are similar. This nice behavior is clarified by the exponentially stabilizing effect of OVC (see [31] for more details).

In Figure 7, closed loop responses of all traditional helicopter controls (i.e., 3 main rotor and 1 tail rotor controls) are given for both classical helicopter and helicopter with MHT. The most important observation related to the traditional helicopter controls is that there is substantial reduction in the peaks of the absolute values of these controls if MHT is utilized. It can be easily seen from this figure that lateral cyclic blade pitch angle (i.e., $\theta_c$) experiences with smallest

---

**Figure 4: Cost optimization via SPSA.**
reduction. Moreover, the control variations are smooth and small.

In Figure 8, closed loop responses of MHT controls (i.e., collective and differential angles and distance control parameter) are given. It is clear from this figure that the peaks of the absolute values of all these additional controls are reasonable. Furthermore, they do not experience catastrophic behavior. Our extensive results also show that blade states do not experience catastrophic behavior and their qualitative behaviour is similar with/without MHT. This good behaviour is also clarified by the exponentially stabilizing effect of OVC.

8. Conclusions

Moving horizontal tail (MHT) idea is investigated in order to reduce helicopter flight control system (FCS) energy. Complex, control-oriented, physics-based nonlinear helicopter models are used for this purpose. Output variance constrained (OVC) controller is applied for helicopter FCS design. A stochastic optimization method is used in order to trim the helicopter during the simultaneous trimming and FCS design problem. Substantial FCS energy reduction (around 60%) is obtained using MHT. It is also important to note that this energy saving is obtained using small MHT control inputs. Nowadays such small changes are easily achievable and technologically feasible. It is also required to note that the FCS energy saving given in this paper is computed using linearized state-space model. In reality due to the nonlinearities it may be slightly different than this value.

Moreover, the qualitative behaviors of fuselage and blade states with/without MHT are similar and they do not display catastrophic behaviors. The outputs of interest (i.e., helicopter Euler angles) with/without MHT also display qualitatively and quantitatively similar behaviors while satisfying all of the output variance constraints. The peak values of traditional controls decrease with MHT clarifying the substantial reduction of FCS energy seen when MHT is applied.

Nomenclature

\[ p, q, r: \] Helicopter angular velocities, [rad/s]
\[ u, v, w: \] Helicopter linear velocities, [m/s]
\[ \phi_A, \theta_A, \psi_A: \] Helicopter Euler angles, [rad]
\[ J: \] Control energy, [J]
\( l_0 \) : Control parameter of distance between helicopter center of gravity and horizontal tail, \([\text{m}]\)

\( \beta_0, \beta_c, \beta_s \) : Collective and two cyclic blade flapping angles, \([\text{rad}]\)

\( \zeta_0, \zeta_c, \zeta_s \) : Collective and two cyclic blade lagging angles, \([\text{rad}]\)

\( \theta_0, \theta_c, \theta_s \) : Collective and two cyclic blade pitch angles, \([\text{rad}]\)

\( \theta_T \) : Collective tail rotor angle, \([\text{rad}]\)

\( \eta_0, \eta_d \) : Collective and differential moving horizontal tail angles, \([\text{rad}]\)

FCS: Flight control system

MHT: Moving horizontal tail

SPSA: Simultaneous perturbation stochastic approximation.

**Conflict of Interests**

The authors declare that they have no conflict of interests regarding the publication of this paper.

**Acknowledgment**

This work was supported by Research Fund of the Erciyes University, Project no. FBA-2013-4179.

**References**


A Robust $H_\infty$ Controller for an UAV Flight Control System

J. López,¹ R. Dormido,² S. Dormido,² and J. P. Gómez¹

¹Dynamic Systems Research Group, Universidad Politécnica de Madrid (ETSIA/EUITA), Plaza Cardenal Cisneros 3, 28040 Madrid, Spain
²Department of Computer Sciences and Automatic Control, UNED, Calle Juan del Rosal 16, 28040 Madrid, Spain

Correspondence should be addressed to R. Dormido; raquel@dia.uned.es

Received 27 November 2014; Revised 14 January 2015; Accepted 19 February 2015

Academic Editor: Zheng Zheng

Copyright © 2015 J. López et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The objective of this paper is the implementation and validation of a robust $H_\infty$ controller for an UAV to track all types of manoeuvres in the presence of noisy environment. A robust inner-outer loop strategy is implemented. To design the $H_\infty$ robust controller in the inner loop, $H_\infty$ control methodology is used. The two controllers that conform the outer loop are designed using the $H_\infty$ Loop Shaping technique. The reference vector used in the control architecture formed by vertical velocity, true airspeed, and heading angle, suggests a nontraditional way to pilot the aircraft. The simulation results show that the proposed control scheme works well despite the presence of noise and uncertainties, so the control system satisfies the requirements.

1. Introduction

There is a considerable and great interest in using unmanned air vehicles (UAVs) to perform a multitude of tasks [1]. UAVs are gaining more powerful skills to accomplish a wide range of missions with high efficiency and high accuracy rate. They are becoming vital warfare and homeland security platforms because they significantly reduce both the costs and the risk to human life and first-responder capabilities. UAVs have many typical applications such as intervention in industrial plants, natural disasters intervention, cooperation with other ground robots in demining operations, through aerial mapping, remote environmental research, pollution assessment and monitoring, fire-fighting management, security, for example, border monitoring, law enforcement, scientific missions, agricultural and fisheries applications, oceanography, or communications relays for wideband applications. Due to their numerous benefits, it would be nice to decrease the global cost of this type of aircraft. In this sense, the flight control design problem for low cost UAV still requires significant efforts, being the control and dynamic modeling of UAVs which is an attractive field of research.

The control of UAVs is not an easy task as the UAV is a multi-input multioutput (MIMO), under actuated, unstable, and highly coupled system. Many traditional control strategies have been used over the years for the control of UAVs, such as linear quadratic regulator (LQR) [2, 3].

Robust techniques have also been applied to design controllers to achieve robust performance and simultaneously guarantee stability when system deviates from its nominal design condition and/or is subjected to exogenous disturbances. In particular, robust $H_\infty$ control method by Zames [4, 5] has been used in flight control systems for both lateral and longitudinal dynamics of aircraft [6–8].

In this work, an inner-outer loop control architecture applied to the longitudinal and lateral flight motions is implemented using the $H_\infty$ Loop Shaping Design procedure [9, 10] to synthesize the inner-loop controller. The technique decouples the longitudinal and lateral dynamics and minimizes the cross effects involved. The feasibility of the controller is analyzed.

The control scheme is implemented on a 6-DOF non-linear simulation model. Different simulation results are presented to show the robustness of the proposed control architecture. The paper is structured as follows. Section 2 presents the aircraft model and its linearization. Section 3 describes the control problem, presenting the control objectives and the control scheme. Design results are analyzed in Section 4. Flight test results are presented in Section 5.


2. Aircraft Model

2.1. Fully Nonlinear Dynamic Model. The UAV is a 1/3 scaled down model of a Diamond Katana DA-20 shown in Figure 1. The main characteristics of the aircraft are as follows:

(i) span 3.9 m,
(ii) wing surface 1.47 square meters,
(iii) mean aerodynamic chord 0.39 m,
(iv) mass 18–30 kg,
(v) cruise velocity 130 km/h,
(vi) maximum velocity 200 km/h,
(vii) engine power 8 HP,
(viii) centre of gravity between 15 and 31% of mean aerodynamic chord.

Aircraft dynamics is described as a full 6-degree-of-freedom (DOF) 13-state high fidelity UAV nonlinear model. The nonlinear model has been developed in standard body axes centered at the aircraft center of gravity where \( x \) points forward, through the aircraft noise, \( y \) is directed to the starboard (right), and \( z \) is directed through the belly of the aircraft.

Using the notation given by Stevens and Lewis [11], the flight dynamic model that describes the rigid body motion of the aircraft is given by the following equations. Force equations are as follows:

\[
\begin{align*}
\dot{U} &= RV - QW + g_x + \frac{F_x}{m}, \\
\dot{V} &= -RU + PW + g_y + \frac{F_y}{m}, \\
\dot{W} &= QU - PW + g_z + \frac{F_z}{m}.
\end{align*}
\] (1)

Moment equations are as follows:

\[
\begin{bmatrix}
\dot{P} \\
\dot{Q} \\
\dot{R}
\end{bmatrix} = J^{-1} \begin{bmatrix}
L \\
M \\
N
\end{bmatrix} - \begin{bmatrix}
0 & -R & Q \\
R & 0 & -P \\
-Q & P & 0
\end{bmatrix} \begin{bmatrix}
P \\
Q \\
R
\end{bmatrix}. \tag{2}
\]

Kinematic equations are as follows:

\[
\begin{align*}
\dot{\phi} &= P + \tan \theta \left( Q \sin \phi + R \cos \phi \right), \\
\dot{\theta} &= Q \cos \phi - R \sin \phi, \\
\dot{\psi} &= \frac{(Q \sin \phi + R \cos \phi)}{\cos \theta}.
\end{align*}
\] (3)

Navigation equations are as follows:

\[
\begin{bmatrix}
\dot{p}_N \\
\dot{p}_E \\
\dot{h}
\end{bmatrix} = B^{-1} \begin{bmatrix}
U \\
V \\
W
\end{bmatrix}, \tag{4}
\]

where \( m \) is the mass; \((U, V, W)\) are the body axis velocity states; \((P, Q, R)\) are the body axis rates; \( \phi, \theta, \psi \) are the roll, pitch, and yaw angles, respectively; and \((p_N, p_E, h)\) are the north, east, and height positions. \( F = (F_x, F_y, F_z) \) represents the aerodynamic force vector and \( M = (L, M, N) \) represents the moment vectors. \( J \) is the aircraft inertia:

\[
J = \begin{bmatrix}
J_{XX} & 0 & -J_{XZ} \\
0 & J_{YY} & 0 \\
-J_{XZ} & 0 & J_{ZZ}
\end{bmatrix}; \tag{5}
\]

\( B \) is the inertial top body transformation matrix; \((g_x, g_y, g_z)\) is the gravity vector, which is the transformation of the \((0, 0, g)\) NED-frame gravity vector to the body axis frame, as shown below:

\[
\begin{bmatrix}
g_x \\
g_y \\
g_z
\end{bmatrix} = B \begin{bmatrix}
0 \\
0 \\
g
\end{bmatrix}, \tag{6}
\]

where

\[
B = \begin{bmatrix}
\cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\
-\cos \theta \sin \psi + \sin \phi \sin \theta \sin \psi & \cos \theta \cos \psi + \sin \phi \sin \theta \sin \psi & \sin \phi \cos \theta \\
\sin \theta \sin \psi + \cos \phi \sin \theta \cos \psi & -\sin \theta \cos \psi + \cos \phi \sin \theta \sin \psi & \cos \phi \cos \theta
\end{bmatrix}. \tag{7}
\]
The resulting model is described by a thirteen-state order model [12]. Due to the complexity and the uncertainty inherent to aerodynamic systems, the dynamic model was identified by a complete identification flight set through the full envelope. See Stevens and Lewis for details [11].

2.2. Linearized Dynamic Model. The nonlinear dynamic model described in Section 2.1 is linearized about certain trimmed operating conditions. This process is accomplished by perturbing the state and control variables from steady state.

The mathematical formulation of the dynamic system is modeled with standard continuous time invariant state space formulation given by (8). Where A is a $13 \times 13$ matrix, B a $13 \times 4$ matrix, C a $12 \times 13$ matrix, and D is a $12 \times 4$ matrix,

$$\dot{x} = Ax + Bu$$

(8)

and the state, output, and control vectors are, respectively,

$$x = [V_T \alpha \beta \theta \psi P Q R p_N p_E h \text{ pow}]^T,$$

$$y = [a_x a_y a_z P Q R \text{ lon lat } h \dot{p}_N \dot{p}_E \dot{h}]^T,$$

(9)

$$u = [\delta_{th} \delta_e \delta_a \delta_r]^T.$$

The state vector (x) components are true airspeed ($V_T$), angle of attack ($\alpha$), sideslip angle ($\beta$), roll angle ($\theta$), yaw angle ($\psi$), roll rate ($P$), pitch rate ($Q$), yaw rate ($R$), north position ($p_N$), east position ($p_E$), altitude ($h$), and power (pow).

The output vector (y) is formed by x-component of acceleration ($a_x$), y-component of acceleration ($a_y$), z-component of acceleration ($a_z$), roll rate ($P$), pitch rate ($Q$), yaw rate ($R$), longitude (lon), latitude (lat), altitude (h), north position derivative, east position derivative, and altitude derivative.

The control vector (u) is defined by throttle ($\delta_{th}$), elevator ($\delta_e$), aileron ($\delta_a$), and rudder ($\delta_r$).

The dynamics are linearized about a representative flight condition. This nominal condition is $V_T = 30 \text{ ms}^{-1}$, centre of gravity position equal to 25% of mean aerodynamic chord, $\phi = 0 \text{ rad}$, $\psi = 0 \text{ rad}$, $R = 0 \text{ rad}$, $P = 0 \text{ rad}$, $\theta = 0 \text{ rad}$, rate of climb = 0 rad, and lateral acceleration = 0 rad.

3. Control Technique

3.1. Control Objectives. The main objective is the design of a robust controller to track all types of input commands in a noisy environment. The controller has to be designed as a trade-off robustness and performance in order to fulfill the specifications described in this section.

3.1.1. Closed Loop Specifications. Stability of the aircraft, minimal overshoot, and reasonably long settling time are important constraints in the design. Translated into physical design goals, the controller must perform the following specifications:

(i) altitude response: overshoot < 5%, rise time < 5 s, and settling time < 20 s,
(ii) heading angle response: overshoot < 5%, rise time < 3 s, and settling time < 10 s,
(iii) flight path angle response: overshoot < 5%, rise time < 1 s, and settling time < 5 s,
(iv) airspeed response: overshoot < 5%, rise time < 3 s, and settling time < 10 s,
(v) cross coupling between airspeed and altitude: for a step in commanded altitude of 30 m, the peak value of the transient of the absolute error between airspeed and commanded airspeed should be smaller than 0.5 ms$^{-1}$; conversely, for a step in commanded airspeed of 2 ms$^{-1}$, the peak value of the transient of the absolute error between altitude and commanded altitude should be smaller than 5 m.

3.1.2. Gust Rejection. Second objective of the control system is to include robustness to gust effects on the aircraft. In this sense, turbulence can be considered as a stochastic process defined by its velocity spectra. For an aircraft flying at a cruise speed $U$, a commonly used velocity spectra for turbulence model is the Dryden spectra [13]:

$$\Phi_v = \frac{2L \sigma^2 (1 + 12 (L_v/U)^2 w^2)}{\pi U (1 + 4 (L_v/U)^2 w^2)^2},$$

(10)

where $w$ is the frequency in rad s$^{-1}$, $\sigma$ is the turbulence standard deviation, and $L_v$ is the turbulence scale length. The turbulence parameters values for severe gust conditions are given by [14]

$$\sigma = 0.1 + 0.00733h, \quad 300 < h < 600 \text{ m}$$

$$\sigma = 3.04 + 0.00244h, \quad 600 < h < 1400 \text{ m}$$

$$\sigma = 6.45 \text{ m/s}, \quad 1400 < h < 5800 \text{ m}$$

(11)

$$L_v = \frac{h}{(0.177 + 0.00274h)^{1/2}},$$

where $h$ is the altitude. Our gust rejection specification is to reject all disturbances below 13 rad s$^{-1}$.

3.1.3. Noise Rejection. Basically, the measured variables for the lateral control are the lateral acceleration and the yaw and roll rates measured in body fixed axis. For the selected sensors, the noise is high and concentrated in the frequency range above 30 rad s$^{-1}$. Thus, high frequency specification is that in which all noise spectra, which normally occur above 30 rad s$^{-1}$, should be rejected.

3.1.4. Robustness Specifications. The controller designed has to be robust against uncertainty in the plant model. The robust specifications are defined as follows.

3.2. Controller Design. The control architecture is based on that proposed by Tucker and Walker [13]. As Figure 2 shows, basically, it consists of two loops: an inner-loop controller to achieve stability and robustness to expected parameter uncertainty and an outer loop for tracking reference performances.

The design of the inner loop is focused on maintaining the vertical velocity deviation, the heading angle deviation, and the airspeed deviation near zero.

\[
M = \begin{bmatrix}
\frac{4^2}{s^2 + 2 \cdot 4s + 4^2} & 0 & 0 \\
0 & \frac{1.5^2}{s^2 + 2 \cdot 1.5s + 1.5^2} & 0 \\
0 & 0 & \frac{2.25^2}{s^2 + 2 \cdot 2.25s + 2.25^2}
\end{bmatrix}
\]  

(12)

The matching model is selected to accomplish desired behaviour of the vertical speed, airspeed, and roll angle to achieve the closed loop specifications detailed in Section 3.1. The cross coupling terms are zero, thus, defining the requirement for closed loop system as decoupled.

Two different controllers conform to the outer loop: the altitude controller and the heading angle-lateral deviation controller. Both controllers are synthesized using the $H_{\infty}$ Loop Shaping technique (see [9, 15, 16]).

Figure 3 shows the general framework used in the design process.

3.2.1. The Inner Loop Synthesis Procedure. Figure 4 shows the inner loop architecture. Its main goal is to minimize both the deviation to desired output and the control effort. $r_i \in \mathbb{R}^3$ is the reference input vector, whose components are the vertical speed, airspeed, and the roll angle. $u \in \mathbb{R}^4$ is the control signal. $z_1 \in \mathbb{R}^3$ is the vector of performance outputs. $z_2 \in \mathbb{R}^2$ is the vector of weighted control inputs. The feedback variables are the vertical speed, airspeed, the roll angle, the pitch rate, the yaw rate, the roll rate, and the sideslip.

The total plant $G_{total}$ is formed by the plant $G$ (the linearized UAV model), the actuators model, and the corresponding delays. These delays are modeled using the first order Pade approximations. They are used to represent plant uncertainties in the high frequency range such as modeling errors and neglected actuator dynamics. Four delays of 100 ms are included in the plant model, one in each input including the throttle.

The actuator model for $\delta_e, \delta_a,$ and $\delta_r$ is given by the first-order linear approximation $10/(s + 10)$ and the engine model is represented by $2/(s + 2)$.

The sensor noise is represented by means of white noise model. The standard deviations of the sensor noise corresponding to the output vector are $0.1 \text{ms}^{-2}$ for accelerations, $0.005 \text{rads}^{-1}$ for angular velocity, $5 \text{m}$ for position, and $0.5 \text{ms}^{-1}$ for velocity.

The controller $K$ is designed using the $H_{\infty}$ technique. It must guarantee the stability and follow an ideal model, the so-called matching model ($M$). That is, the closed loop system output $y_1$ is expected to match $y_m \in \mathbb{R}^3$, the output of the ideal model $M$.

The matching model $M$, which defines the behaviour of the vertical speed, the true speed, and the heading angle, consists of the following three second-order systems:

The cross coupling terms are zero, thus, defining the requirement for closed loop system as decoupled.
$W_1$ is related with reference tracking. So, its elements are selected as low pass filters. The yaw rate and roll rate are selected as pass band filters.

$W_2$ is devoted to minimize the control effort. This is why it is selected as a high pass filter, where its gain and bandwidth are chosen to allow low frequency control effort and to minimize high frequency control effort.

$W_3$ and $W_4$ are unity matrix. They weight turbulences and output disturbances, respectively.

The controller’s synthesis is accomplished using an iterative procedure. First, the weights are selected; then the controller $K$ is synthesized and finally the resulting system performances are analysed.

After this iterative process, the weights selected are the following:

$$W_1 = \text{diag}\left( \frac{3^2(s+1)}{s+2\cdot3s+3^2}, \frac{10\cdot500s}{s/0.001+1}, \frac{5\cdot500s}{s/0.001+1}, \frac{\gamma^2(s+1)}{s+2\cdot7s+7^2}, \frac{8\cdot500s}{s/0.001+1} \right).$$

$$W_2 = \text{diag}\left( \frac{0.5s/0.1+1}{s/0.008+1}, \frac{0.5s/0.1+1}{s/0.008+1}, \frac{0.5s/0.1+1}{s/0.008+1} \right).$$

(13)

After some iterations, a stabilizing controller $K(s)$ is determined. This controller minimizes the variables $z_1$ and $z_2$ (see Figure 3) which corresponds to deviation between the desired output, provided by the matching model and the real aircraft output and control effort. The subresulting suboptimal robust stability margin is $\gamma = 4.98$.

3.2.2. The Outer Loop Synthesis Procedure. Two different controllers conform to the outer loop: the altitude controller (see Figure 5) and the heading angle-lateral deviation controller (see Figure 6). The two outer-loop controllers are synthesized using the $H_{\infty}$ Loop Shaping technique [16].

Figure 5 shows the first problem to be solved, where $C$ is the controller and $W_1$ and $W_2$ are the weights used to tune the optimization. The simplified models of the plant used to synthesize these controllers are those defined in the matching model.

In the design of the altitude controller, an output integrator is used to provide height and vertical velocity outputs. An input integrator is used to improve the low frequency behaviour.

In a similar way in the heading angle-lateral deviation controller design an output integrator is used to provide yaw
angle and its derivative outputs. An input integrator is used to improve the low frequency behaviour.

The gamma values encountered are 3.18 and 2.5 for the altitude controller and the heading angle-lateral deviation controller, respectively.

The heading angle and lateral deviation controllers have been built together due to the hard interaction between the variables implied which motivates a tedious iterative process when individual controllers were designed. In this approach, these two controllers are synthesized jointly.

4. Design Results

The performance of a system can be represented by the sensitivity function $S$. The maximum singular value of $S$ is an important boundary in this case. By using the largest singular value, we are effectively assessing the worst case scenario. Performance specification means the minimization of the sensitivity function as much as possible for low frequencies. At the same time, control effort should be small in the high frequency range.

Figure 7 shows the sensitivity function. It is easy to see that our goal of minimizing the sensitivity at low frequencies has been achieved. At high frequencies, the gain is unity and around the bandwidth there is a peak in the response.

This behaviour of the sensitivity enables good tracking reference at the low frequency range and noise reduction and robustness in the high frequency range.

Figure 8 shows the control effort behaviour which is lower in the high frequency range as it was expected.

Since the $H_{\infty}$ controller designed produces a 46-size state space realization, it is necessary to apply controller reduction techniques. A final state realization for the controller of dimension 27 is achieved using Hankel minimum degree approximation (MDA) without balancing reduction method [17].

This method has been applied iteratively checking the frequency and time responses every step to evaluate the performance of the proposed UAV control scheme. One example of the time response in one step of this iterative process is shown in Figure 9.
The airplane desired reference is illustrated in Figure 11. The dashed line is the desired trajectory.

Figure 12 shows the airplane simulated trajectory tracked. The dashed line is the desired trajectory and the continuous line is the real one.

The controller is able to manage adequately the output and to control the control vector. Control variables evolutions are shown in Figure 13. The throttle varies around 2% and elevator, ailerons, and rudder present a smooth behaviour. The aileron and rudder are deflected by the controller to order the 45-degree change of direction. Immediately, a sustentation loose typical in this type of manoeuvres is suffered by the aircraft. To compensate this trend, the elevator acts to raise the noise of the aircraft and slightly increase the throttle to maintain the velocity.

Figure 13 confirms that the control variables remain far from its saturation values. The power demand is less than 40% and the elevator, aileron, and rudder demanded deflections are less than 5 degrees. In this case, if the altitude holder is not connected, in 5 s, the airplane suffers an altitude loss of 3 m and rapidly it recovers the desired altitude, in about 5 s more. The UAV quickly corrects its heading angle turning to reduce the error. In about 4.5 s, the error is null; however, the airplane continues turning. This is produced because of the lateral deviation. If the airplane stops its turning movement in 4.5 s, it would continue straight ahead along a parallel line to the desired trajectory. To reduce the lateral deviation, it must continue turning and augmenting, in a first stage, the heading angle error. Following this strategy, the controller gains its tracking heading angle and its lateral deviation reduction goal.

Figure 10 shows the effect of an incorrect order reduction. This performance is obtained when an order reduction is forced and the reduced controller is not able to maintain the desired specifications.

5. Simulation Tests Results

In order to validate the controller designed, a set of test cases have been developed. Below, an experience corresponding to 45-degree heading angle step response is shown. The results allow checking the performance of the aircraft in a noisy environment along this type of manoeuvre.

The airplane desired reference is illustrated in Figure 11. The dashed line is the desired trajectory.

Figure 12 shows the airplane simulated trajectory tracked. The dashed line is the desired trajectory and the continuous line is the real one.

The controller is able to manage adequately the output and to control the control vector. Control variables evolutions are shown in Figure 13. The throttle varies around 2% and elevator, ailerons, and rudder present a smooth behaviour. The aileron and rudder are deflected by the controller to order the 45-degree change of direction. Immediately, a sustentation loose typical in this type of manoeuvres is suffered by the aircraft. To compensate this trend, the elevator acts to raise the noise of the aircraft and slightly increase the throttle to maintain the velocity.

Figure 13 confirms that the control variables remain far from its saturation values. The power demand is less than 40% and the elevator, aileron, and rudder demanded deflections are less than 5 degrees. In this case, if the altitude holder is not connected, in 5 s, the airplane suffers an altitude loss of 3 m and rapidly it recovers the desired altitude, in about 5 s more. The UAV quickly corrects its heading angle turning to reduce the error. In about 4.5 s, the error is null; however, the airplane continues turning. This is produced because of the lateral deviation. If the airplane stops its turning movement in 4.5 s, it would continue straight ahead along a parallel line to the desired trajectory. To reduce the lateral deviation, it must continue turning and augmenting, in a first stage, the heading angle error. Following this strategy, the controller gains its tracking heading angle and its lateral deviation reduction goal.
Figure 9: Lateral deviation step response (correct order reduction).

Figure 10: Lateral deviation step response (incorrect order reduction).

Figure 11: Airplane desired trajectory.

Figure 12: Airplane real trajectory.

Figure 13: Control variables evolution during the 45-feet heading angle response.

6. Flight Test Results

For testing the whole system and the performance of the controller in flight, many real tests are accomplished. These tests are scheduled to validate in essence the physical design of the UAV, communications equipment, engine capabilities, and onboard software. A very important part of onboard software is the flight control system.

To manage the UAV platform, a ground station is developed (see Figure 14). It enables following the position and the attitude of the aircraft directly on a map shown in
The computer. It also allows showing the main variables of the UAV which are sent through a radio link. The ground station allows introducing a set of waypoints. The autopilot takes care of both navigation and stability of the plane. The mission is planned via waypoints, placing on a geo referred map the position of each waypoint at the beginning of the mission. This mission can be easily modified during its execution by adding/changing/removing waypoints in the map.

The system provides a user-friendly interface used to display the plane position in real time on a map during the mission and to monitor some UAV parameters such as battery levels, speed, position and orientation, or the sensors measurements. The system also provides a radio link which allows a continuous exchange of data between the plane and the control station.

In an emergency case, the aircraft can switch to a PIL (pilot in the loop) mode in which the plane can be teleoperated from the control station by using a control-stick while the onboard autopilot remains on sleep mode.

The selected test to illustrate the aptitudes of the autopilot designed is a circuit formed by four waypoints which is shown in Figure 15. The tracking reference trajectory is shaped for the waypoints labelled from one to four. The circles around the waypoint determine the instance when the reference input changes to the next waypoint (goal condition).

The reference is provided to the autopilot as a psi angle function and is built in a soft way using a combination of a step and a ramp. The reference is shown in the Figure 16.

Figure 17 shows how the UAV is capable of managing adequately the uncertainties and disturbances introduced by the modelling inaccuracies and the noisy output provided by the sensor. The response of the aircraft is not oscillating and it reaches the correct trajectory quickly when covering 600 m approximately which means 20 s at 30 ms$^{-1}$ of mean velocity.

The entire trajectory covered is around 160 s and the psi angle and lateral deviation error are minimized satisfactorily. A desired decoupling between lateral and longitudinal dynamics is achieved.

Figure 18 shows the noisy accelerations output provided by the inertial sensors to the controller.
Figure 18: Accelerations measured.

Figure 19 shows the onboard equipment mounted on the UAV. In Figure 20, the UAV during the test cases is shown.

7. Conclusions

The dynamics of the UAVs are highly nonlinear and continuously vary with time. Also, it is subjected to severe external disturbances. Due to this, dynamic and parametric uncertainties arise in the mathematical model of the UAVs over different operating conditions. This paper addresses the problem of designing a robust control system for UAVs in the presence of uncertainties using $H_{\infty}$ technique. The controller implemented allows the UAV to track all types of manoeuvres in the presence of noisy environment. The reference vector used, formed by vertical velocity, true airspeed, and heading angle, suggests a nontraditional way to pilot the aircraft that is based on commanding the desired reference vector and lets the controller select throttle position and surfaces deflections. This kind of pilot-machine interaction appears to be a more intuitive approximation.

The frequency domain analyses show that the proposed controller guarantees good performance, attenuating high frequency noise and also supplying suitable control signals. The tracking performance of the UAV is within the desired tracking performance range. The control efforts during soft manoeuvres are in the same way moderated. The first results obtained with the real UAV with the controller designed appeared to be very suitable.

The desired behaviour is introduced using a matching model ($M$). This architecture allows modifying the desired performances without varying the controller architecture.

The architecture selected to decouple the longitudinal and lateral dynamics provides very good performances. The outer-loop controller gives a very good behaviour in case of step responses, ramp responses, and combinations of these two input types. It is important to note that the outer loop shows a signal derivative at the input and this should be avoided. This signal derivative is not part of the real implementation. In this case, the inputs of the outer loop are provided directly by the GPS (height and vertical velocity). The specifications and robustness performances have been validated by mean of simulation and real tests.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work has been funded by the National Plan Project DPI2012-31303 of the Spanish Ministry of Economy and Competitiveness and FEDER funds.

References


A Novel Biobjective Risk-Based Model for Stochastic Air Traffic Network Flow Optimization Problem

Kaiquan Cai,1,2 Yaoguang Jia,1,2 Yanbo Zhu,2,3 and Mingming Xiao1,2

1School of Electronics and Information Engineering, Beihang University, Beijing 100191, China
2National Key Laboratory of CNS/ATM, Beijing 100191, China
3Aviation Data Communication Corporation, Beijing 100191, China

Correspondence should be addressed to Yanbo Zhu; zyb@adcc.com.cn and Mingming Xiao; buaxmm@163.com

Received 5 December 2014; Accepted 9 January 2015

1. Introduction

With the rapid growth of air traffic, airspace congestion and flight delays are becoming serious due to demand-capacity imbalances in air traffic management (ATM) system, especially in convective weather conditions. Severe airspace congestion and flight delays not only deteriorate the service quality of airlines but also raise operational costs. According to statistics, in 2013 [1], the average punctuality rate of flights all over the world was only 80%. China endures an even lower rate, which was 73.1% for big airlines and 70.0% for small airlines. Among all factors that lead to flight delays, the impact of inappropriate flight plan and convective weather takes up a weight of 59.2%.

The network-wide air traffic flow management (ATFM), through constructing global predeparture flight plans (including departure time, flight routes, and arrival time at the waypoints or arrival airports) for all the flights planned to fly over the air traffic network, aims to balance the air traffic demand and ATM system capacity. In ATM domain, considered as one of the effective ways to alleviate airspace congestion and flight delays, ATFM becomes increasingly highlighted in the expanding air transportation system and has drawn a mass of attention to researchers. Single airport ground-holding problem was firstly proposed to minimize ground delays by optimizing departure time of a flight when its arrival airport capacity is insufficient [2]. Later on, the concerns evolved to multiairport ground-holding problem [3–5], air traffic flow management problem [6, 7], and air traffic flow management rerouting problem [8–12]. All above problems can be collectively labeled as air traffic network flow optimization (ATNFO) problem. An air traffic network [13–15] can be modeled as airports network, airspace sector network (nodes include airports and sectors), or air route network. ATNFO aims to minimize the cost of flight delays by optimizing flight plan over a given network and time horizon, taking into account the capacity limits of airports or airspace sectors.

The above ATNFO problems share a common nature; that is, the capacities of airports and airspace sectors are
predetermined. However, due to the stochastic nature of the convective weather dynamics, it is hard to predict airport/airspace capacity accurately in future 24 hours (ATNFO is usually implemented more than 24 hours earlier to provide decision-making) in practice [16]. The convective weather dynamics often lead to capacity uncertainties for the airports or airspace sectors in a network. In the presence of capacity uncertainties, ATFM actions might be noneffective or impractical when adopting the deterministic ATNFO models. Hence, in recent years, the stochastic ATNFO (SATNFO) problem with uncertainties has aroused increasing concerns. Existing models for SATNFO problem can be classified into stochastic programming [17, 18] based methods and robust optimization [19, 20] based methods.

Deterministic models for ATNFO problem assume airports and airspace sectors capacity as constant values. However, stochastic programming based method considers the uncertainties of airports and sectors capacities by establishing probabilistic distributions for them and finds a solution that is feasible and optimal in some sense. The first attempt at stochastic programming in ATFM was made by Richetta and Odoni [21, 22]. They applied scenario tree [23, 24] to model the uncertainty of capacity and proposed a dynamic model for single airport ground-holding problem. Here, scenario tree is a kind of format to describe capacity change and its probability along time horizon. Later, the utilization of scenario tree was extended to stochastic model for multiairport ground-holding problem by Ball et al. [25]. In 2012, Agustín et al. [26] dealt with SATNFO problem under weather uncertainty by using stochastic model, which is an extension of their deterministic tight mixed 0-1 model [11]. In their work, a scenario tree which models the capacity uncertainties under storm scenarios is generated to represent the deterministic equivalent model for their stochastic mixed 0-1 program. Since scenarios in reality are numerous, especially when prolonging the time horizon (such as 24 hours) of ATFM, stochastic programming becomes intractable.

Robust optimization [27, 28] captures the probabilistic property of a problem by constructing appropriate limited uncertainty sets for uncertain parameters and then solves a solution that is feasible for all outcomes of the uncertainty sets. The solution is optimized in worst-case condition. Gupta and Bertsimas [29] and Bertsimas and Goyal [30] attempted to apply the idea of robust optimization to the capacity uncertainties in ATFM rerouting problem. Unfortunately, it is likely to get a very conservative solution and cause waste of airspace and time resource when using robustness optimization based method.

Both stochastic programming and robust optimization have the same basic idea for dealing with capacity uncertainties. Firstly, generate scenarios (scenario tree or uncertainty sets) to describe capacity uncertainties; then, optimize air traffic flow by using capacity scenarios as constraints. Their objective is to find a robust plan for all scheduled flights under uncertainty. But both methods have suffered computational intractability and given highly conservative solution.

In order to solve SATNFO problem efficiently, we propose a novel mathematic model that takes into account capacity uncertainties. In this model, we introduce the concept of operational risk to evaluate the effect of capacity uncertainties in ATFM and formulate it via probabilistic risk assessment methodology. A set of flight plans with lower operational risk is deemed to a more robust solution of SATNFO problem, which means the flight plans are more insusceptible to the capacity uncertainties of airports or airspace sectors. Hence, the SATNFO problem is formulated as a weighted biobjective 0-1 integer programming model with two objectives of minimizing the cost of total delays as well as the operational risks. The contributions of our novel SATNFO model are twofold. First, all appropriate scenarios are considered to evaluate the operational risk of the solution. The introduction of biobjective, that is, operational risk and delay cost, can provide a more efficient solution with appropriate weights and thus eliminate highly conservative solution. Second, without using capacity scenarios as constraints, the number of constraints in this model will not dramatically increase with the increasing number of scenarios. Therefore, we can efficiently avoid computationally intractable problem. Using real-world air traffic network data associated with simulated weather data, computation experiments demonstrate that our proposed model has generality that is suitable for solving SATNFO problem under different capacity scenarios. Moreover, the number of constraints is far less than stochastic model with nonanticipative constraints, which means the presented model has less computation complexity. Besides, we provide an appropriate range of weight coefficients to get a better tradeoff between operational risk and flight delay cost in practical ATFM decision-making.

The remainder of the paper is organized as follows. Section 2 presents the mathematical model of our SATNFO problem, including the definition of operational risk. Section 3 presents computation experiments that evaluate the efficacy of our SATNFO model, that is, the biobjective 0-1 integer programming model. Section 4 gives conclusion and future research.

2. Mathematical Model of SATNFO

In this section, our mathematical model of SATNFO problem is introduced in detail. Firstly, the operational risk is defined via probabilistic risk assessment method to evaluate the effect of capacity uncertainties on ATFM. And then, full mathematical formulation of SATNFO problem is presented.

Before the introduction of mathematical model of SATNFO problem, two concepts should be indicated here: (1) air traffic network: airspace is divided into sectors. Each sector or airport can be represented as a node in the network. If a flight is planned to fly from one node to another, there will be a directed arc between these two nodes; (2) capacity: for each sector or airport, there are a limited number of flights that can fly within it in every time unit. It can be easily influenced by convective weather.

2.1. Definition of Operational Risk. During the modeling, we introduce the concept of operational risk into SATNFO problem to deal with the uncertainties. Generally, operational risk is defined [31] as the risk of a change in value caused by
the fact that actual losses, incurred for inadequate or failed internal processes, people, and systems, or from external events, differ from the expected losses. The operational risk in SATNFO problem can be defined as the extra flight delay cost caused by unpredictable capacity changes and inappropriate flight plan. A robust set of flight plans for all scheduled flights receives less impact resultant from capacity uncertainties and thus with lower operational risk.

In order to quantify the operational risk induced by capacity uncertainties, we adopt the probabilistic risk assessment (PRA) method [32–34]. PRA is a methodology to evaluate risks associated with a complex engineered technological entity. It has been successfully used in agrochemicals analysis [35], nuclear power industry [36], and many other complex engineering fields [37]. In PRA method, the risk is characterized by two elements: the magnitude of all possible adverse consequences ($I_i$, $i = 1, \ldots, N$, suppose $N$ possible adverse consequences in total) and the likelihood of each consequence ($P_i$, $i = 1, \ldots, N$). And, the risk is defined as $\sum_{i=1}^{N} I_i \cdot P_i$. As for SATNFO problem, the magnitude of adverse consequence is the extra flight delay cost caused by capacity uncertainties, and the likelihood is determined by the capacity statistical distribution. According to PRA method, the steps to quantify the operational risk in SATNFO problem are as follows.

2.1.1. Capacity Statistical Distribution Modeling. In practical ATFM, the convective weather dynamics often lead to capacity uncertainties for the airports or airspace sectors in a network. A capacity statistical distribution represents possible values of capacity and its distribution and their changes with weather condition along time horizon. The modelling of capacity statistical distribution is built on the basis of the three assumptions and principles.

(1) The airspace is divided into $M$ regions (denoted by $\{S_1, S_2, \ldots, S_M\}$), and each region consists of several sectors and airports. The time period is divided into $Q$ time stages, and each time stage consists of certain smallest time units. When changes of weather spread in the airspace, it can only influence one of the regions during a time stage. Thus, the spreading process can be equally expressed as the capacity of this region decreasing lasting a certain time units.

(2) The extent of weather impact on capacity is divided into $N$ degrees, and the corresponding weather modes are denoted by $\{D_1, D_2, \ldots, D_N\}$.

(3) Let scenario tree [38] describe the capacity statistical distribution. The scenario tree may be various in different time period because of various weather conditions. One node on a scenario tree represents a certain weather mode in a certain region at a certain time stage, according to which we can calculate capacity and its probability of each region (i.e., a capacity distribution of the whole airspace). We call a node as a capacity scenario.

Based on above assumptions, the steps of modelling capacity statistical distribution can be described as follows.

(1) Get the initiative scenario, that is, a weather mode and a region it influenced in time stage 1. Let a scenario be denoted by $(P_{\text{region}}^{ij}, P_{\text{degree}}^{ij})$, where $P_{\text{region}}^{ij}$ represents the probability of the weather spreading from $S_k$ to $S_j$, and $P_{\text{degree}}^{ij}$ represents the probability of weather mode that transfers from $D_i$ to $D_j$. At the initial moment, if a weather mode whose degree is $D_1$ influences the $S_j$ region, we use $(P_{\text{0,1}}^{\text{region}}, P_{\text{0,1}}^{\text{degree}})$ to represent this scenario.

(2) Analyze the impact of weather along time horizon, including the weather spreading modes (decide which region will be influenced) and the degree of its impact. The impact of weather is uncertain. As time stage transfers from time stage $t$ to $t + 1$, the changes of weather may spread to different regions with a certain probability. And moreover, for a region, the impact degree of weather is also uncertain. Hence, a scenario can generate several subsequent scenarios with a certain decision probability. And, along the time horizon, a scenario tree can be established. Suppose $Q = 3$, $N = 1$, airspace is divided into 4 regions (shown in the left figure of Figure 1), weather spreading from north to south, and a scenario tree can be established as the right figure of Figure 1. In this scenario tree, the initiative scenario, that is, the root node $A$, is $(P_{\text{0,1}}^{\text{region}}, P_{\text{0,1}}^{\text{degree}})$.

(3) Based on the process of establishing scenario tree, we can get a dynamic probabilistic capacity distribution and decision probability. For example, in Figure 1, the probability of scenario $G$ happening is

\[ P(G) = P(A) \cdot P(C | A) \cdot P(G | C). \]  

(1)

Scenario $G$ represents a capacity distribution of the whole airspace at time stage 3. According to its information $(P_{\text{1,1}}^{\text{region}}, P_{\text{1,1}}^{\text{degree}})$, we can calculate the capacity for each sector or airport. And the corresponding probability $P(\text{capacity}_G)$ is $P(G)$. Furthermore, for each sector or airport, we can obtain its probability of its possible capacity at time stage $t$ according to all possible scenarios that may appear in stage $t$.

2.1.2. Extra Delay Cost Calculation. Extra flight delay cost is the magnitude of adverse consequence caused by capacity uncertainties. Flight delay may be caused if capacity decreases; that is, the extra delay cost resulting from capacity decreases. The number of flights flying in the sector or airport may exceed capacity which is decreased due to the dynamics of convective weather. Hence, some flights should execute airborne holding or ground holding, which leads to extra flight delays. And, the extra flight delay is regarded as the time for a flight to cross the sector with minimum velocity. To simplify the calculation we assume that there is no more flight coming during this waiting period, which means we take no account of the delay propagation in this model.
2.1.3. Operational Risk Formulation. According to PRA method, the operational risk of SATNFO problem can be defined as below:

\[ \text{Operational Risk} = \sum_{t, A} \text{extra delay} \times p \left( \text{capacity}_{i} \right). \]  

(5)

It can also be written as

\[ \text{Operational Risk} = \text{Risk}_{\text{airborne}} + \text{Risk}_{\text{ground}}. \]  

(6)

Risk_{\text{airborne}} is the risk from airborne holding and can be represented as below:

\[ \text{Risk}_{\text{airborne}} = \sum_{t,i} (N_{it} - C_{S_i}) \cdot \text{Pass}_{i} \cdot P \left( C_{S_i} \right), \]  

(7)

where \( P(C_{S_i}) \) is the probability that the capacity of sector \( i \) at time \( t \) is \( C_{S_i} \).

When the traffic flow in sector \( i \) at time \( t \) is \( N_{it} \), extra airborne holding time \( AED_{it} \) caused by the capacity decreases in sector \( i \) at a certain time \( t \) can be expressed as below:

\[ AED_{it} = (N_{it} - C_{S_i}) \cdot \text{Pass}_{i}. \]  

(2)

Risk_{\text{ground}} is the risk from ground holding that can be represented:

\[ \text{Risk}_{\text{ground}} = \sum_{t,d} (N_{dt} - C_{d_{dt}}) \cdot p \left( C_{d_{dt}} \right), \]  

(8)

where \( C_{d_{dt}} \) is the departure capacity of airport \( d \) at time \( t \).

Since the cost of airborne holding is much greater than ground holding, it is necessary to be considered in the definition of the operational risk. Hence, the total risk function should be improved as

\[ \text{Operational Risk} = w_{\text{airborne}} \cdot \text{Risk}_{\text{airborne}} + w_{\text{ground}} \cdot \text{Risk}_{\text{ground}}, \]  

(9)

where \( w_{\text{airborne}} > w_{\text{ground}}. \)

2.2. Mathematical Formulation of SATNFO Problem. In our SATNFO model, the minimization of operational risk, which evaluates the effect of capacity uncertainties on ATFM, is introduced as an objective. And, the minimization cost of total flight delays is the other objective.

The air traffic network in our SATNFO model is based on deterministic ATFMO model. To our knowledge, the 0-1 integer programming model [9] proposed by Bertsimas et al. (BLO model) has high computational efficiency and also accuracy to depict the real ATFM actions such as ground holding, airborne holding, speed control, and rerouting. Thus, we build our SATNFO model based on BLO model.

2.2.1. Decision Variable. We use the same definition of decision variable in BLO model:

\[ z_{i,t} = \begin{cases} 1 & \text{if flight } f \text{ arrive sector } i \text{ by time } t \\ 0 & \text{otherwise}. \end{cases} \]  

(10)

Note that we use “by” instead of “at.” This definition means once the decision variable is 1 at time unit \( t \), the value of decision variables in the subsequent time units must be 1. It could also be applied to airports. If flight \( f \) takes off from airport \( i \) by time \( t \), the decision variable is 1 (similarly, if flight \( f \) lands at airport \( j \) by time \( t \), the decision variable is 1).
2.2.2. Notations. The notations in the SATFNO model which are similar to BLO model are listed as follows:

- $K$: set of sectors,
- $K^f$: set of sectors that could be flown by fight $f$,
- $P$: set of airports,
- $P^f$: set of airports that could be flown by fight $f$,
- $T$: set of time units,
- $F$: set of fights,
- $A_i^f$: sets of sector $i$'s ancestor sectors for flight $f$,
- $S_i^f$: sets of sector $i$'s subsequent sectors for flight $f$,
- $Cid_p(t)$: ideal planning departure capacity of airport $p$ at time $t$,
- $Cia_p(t)$: ideal planning arrival capacity of airport $p$ at time $t$,
- $Cis_i(t)$: ideal planning capacity of sector $i$ at time $t$,
- $dep_f$: departure airport of flight $f$,
- $arr_f$: arrival airport of flight $f$,
- $Pass_{fij}$: minimum number of time units for air crafts to pass through sector $i$,
- $T^f_i[t^f_i, T^f_i]$ time window for fight $f$ to arriving $T^f_i$: the maximum time units for flight $f$ to arrive at sector $i$ (depart or arrive at airport $i$),
- $T^f_i$: the minimum time units for flight $f$ to arrive at sector $i$ (depart or arrive at airport $i$),
- $\overline{T^f_i}$: the maximum time units for flight $f$ to arrive at sector $i$ (depart or arrive at airport $i$).

2.2.3. Objective Functions. In SATFNO model, we formulate a weighted biobjective 0-1 integer programming model. The objective consists of two parts: the operational risk and total flight delay cost. Thus, the objective function is as below:

$$\text{objective function} = (1 - \beta) \cdot \text{delay cost} + \beta \cdot \text{Risk}_{\text{total}}$$

(11)

Note that the deterministic model is a particular case of this model when $\beta = 0$.

(1) Delay Cost. We refer to the definition in BLO model and classify the delay into two different parts: airborne holding AH and ground holding GH:

$$AH = \sum_{f \in F} \sum_{t \in T^f_i} \left( t - T^f_{i,t} \right) \cdot \left( z^f_{arr,t} - z^f_{arr,t-1} \right)$$
$$- \sum_{t \in T^f_i} \left( t - T^f_{i,t} \right) \cdot \left( z^f_{dep,t} - z^f_{dep,t-1} \right)$$

$$\text{GH} = \sum_{f \in F} \sum_{t \in T^f_i} \left( t - T^f_{i,t} \right) \cdot \left( z^f_{dep,t} - z^f_{dep,t-1} \right).$$

(12)

We calculate them separately and assign airborne holding a higher coefficient as it costs more resources than the other one. Then, we add up airborne holding and grounding holding to get total delay cost, which is

$$\text{delay cost} = \alpha_{\text{sh}} \cdot AH + \alpha_{\text{gh}} \cdot GH.$$  

(14)

(2) Operational Risk. According to our definition in the previous section, the operational risk is extra delay cost caused by unpredictable capacity changes. We have obtained operational risk through (1)–(9).

2.2.4. Constraints. Consider

$$\sum_{f \in F_{\text{dep}} = p} \left( z^f_{\text{dep}, t} - z^f_{\text{dep}, t-1} \right) \leq Cid_p(t) \quad \forall p \in P, t \in T$$

(15)

$$\sum_{f \in F_{\text{arr}} = p} \left( z^f_{\text{arr}, t} - z^f_{\text{arr}, t-1} \right) \leq Cia_p(t) \quad \forall p \in P, t \in T$$

(16)

$$\sum_{i \in K, t \in T} \left( \max \left\{ 0, z^f_{i,t} - \sum_{i' \in A^f_i} z^f_{i', t} \right\} \right) \leq Cis_i(t)$$

(17)

$$z^f_{i,t-1} - z^f_{i,t} \leq 0 \quad \forall t \in T, f \in F, i \in K_i$$

(18)

$$z^f_{i,t-1} \leq \sum_{i' \in A^f_i} z^f_{i', t} \quad \forall t \in T, f \in F, i \in K_i$$

(19)

$$z^f_{i,t} \leq \sum_{i' \in S^f_i} z^f_{i', t} \quad \forall f \in F, i \in K^f$$

(20)

$$\sum_{i' \in S^f_i} z^f_{i', t} \leq 1 \quad \forall f \in F, i \in K^f$$

(21)

$$z^f_{i,t} \leq 0 \quad \forall f \in F, i \in K^f \cap P^f, t \in T : t < T^f_i$$

(22)

$$z^f_{i,t} - z^f_{i,t-1} = 0 \quad \forall f \in F, i \in K^f \cap P^f, t \in T : t < T^f_i$$

(23)

$$z^f_{i,t} \in \{0,1\} \quad \forall f \in F, i \in K^f \cap P^f, t \in T^f_i$$

(24)

Constraints (15) and constraints (16) give the limitation of departure and arrival capacity of airport $p$ at time $t$. They ensure that the number of flights which take off from airport $p$ will not exceed the departure capacity of airport $p$ at time $t$ and that the number of flights which land on airport $p$ will not exceed the arrival capacity of airport $p$ at time $t$. Constraints (17) stipulate that the number of flights that arrive at sector $k$ will not exceed the capacity of sector $k$ at time $t$. Closely related to the definition of decision “by,” constraints (18) guarantee the time continuity of every flight. Constraints (19) state that if flight $f$ arrives at sector $i$ at time $t$, it must have been arrived at its ancestor node
before time $t - \text{pass}_{ij}$. Constraints (20) state that if flight arrives at sector $i$ it will arrive at its subsequent sectors. Constraints (21) present that, for every flight that has arrived at sector $k$, it can only arrive at one of its subsequent sectors. Constraint (22) and Constraint (23) present the time-window limitation for every flight and the sectors a flight will pass by. Constraints (22) and Constraint (23) guarantee that the flight $f$ cannot arrive at sector $i$ before the minimum time we planned. Constraints (23) ensure that flight $f$ should arrive at sector $i$ before the maximum time we planned or the flight will never arrive at sector $i$. Finally, constraints (24) state that decision variables are Boolean values which make our model a 0-1 programming model.

3. Experiments

In this section, two sets of experiments are carried out to evaluate the efficacy of our SATNFO model, that is, the biobjective 0-1 integer programming model. In the first set, we analyze the effect of parameter setting, that is, the weight coefficient of risk ($\beta$) in our SATNFO model, and thereby provide an appropriate range of weight coefficient to get a better tradeoff between operational risk and flight delay cost in practical ATFM decision-making. In the second set, we show the generality of our mathematical model by solving a SATNFO problem under various scenarios. After that, the computational complexity of our SATNFO model is analyzed.

As for the experimental data, we extract a real air traffic network from the northern airspace of China, which consists of 25 sectors. A time horizon of two hours and 12 minutes is considered, and it is divided into 22 unites, 6 minutes each. Two sets of flight plans, consisting of 55 flights and 220 flights, respectively, are considered to be optimized for different purposes, which will be explained later. Capacity instances are generated based on simulated weather data, and the corresponding generation method will be shown in Section 3.1. All experiments are performed using optimization programming language with CPLEX 12.5.1, on a PC with Inter core i5-3470 processor, 3.20 GHz, and 12 GB RAM Microsoft Windows 7 OS.

In the following of this section, the details of capacity instances generation, two sets of experiments, and computational complexity of our SATNFO model are presented in order.

3.1. Capacity Instances Generation

As mentioned previously, a scenario tree represents possible values of capacity and its distribution and their changes with weather condition along time horizon. In this subsection, we generate 6 different capacity instances, that is, 6 different scenario trees based on simulated weather data. The steps of capacity instances are described as follows.

(1) Partition of Airspace. The airspace consisting of 25 sectors is divided into 4 regions. Figure 2 is a schematic diagram of the airspace partition, thick lines are boundary of each region, and the value in sector denotes sector number. Weather has

![Figure 2: A schematic diagram of dividing airspace into 4 regions.](image-url)
weather is in $S_3$, it would spread to $S_4$. Otherwise, the changes of weather would leave this airspace and all capacities recover straight after. The definition of transfer probability is similar to mode 1.

**MODE 3: South-to-North Spreading.** In this mode, changes of weather influence $S_4$ first and then spread to $S_3, S_2$, and $S_1$ sequentially. The definition of transfer probability is similar to mode 1 and mode 2.

(4) **Capacity Instances.** Having gotten above three spreading modes and two sets of impact degrees of weather on capacity, we can get 6 capacity instances, denoted by the following:

- Instance 11 (MODE1, DEG1), Instance 12 (MODE1, DEG2),
- Instance 21 (MODE2, DEG1), Instance 22 (MODE2, DEG2),
- Instance 31 (MODE3, DEG1), Instance 32 (MODE3, DEG2).

According to each instance, we can generate its corresponding scenario tree based on the given weather data.

3.2. **Risk Weight Coefficient Analysis.** In this experiment, we analyze the effect of parameter setting, that is, the weight coefficient of risk $\beta$, in our SATNFO model. Hence, the relationship between $\beta$ and total delay and the relationship between $\beta$ and operational risk are analyzed concretely. For this experiment, a set of flight plans which consists of 55 flights is optimized under capacity instance “Instance 21.”

Table 1 shows corresponding delay and risk with various $\beta$. Figure 3 shows the relationship between $\beta$ and delay (left) and the relationship between $\beta$ and risk (right). From Figure 3, we find that operational risk decreases and total delay increases with the increase of $\beta$. It means that when we add risk as one objective of SATNFO problem, the optimized flight plans are relatively conservative, that is, allowing more delays for safety operation. Further, the optimized flight plans become more conservative with the increase of risk weight coefficient. Thus, decision-makers can choose their appropriate risk coefficient to optimize flight plans for all scheduled flights using our SATNFO model. According to their experience and the real condition, they can obtain an acceptable tradeoff between optimization and robustness.

More important, the analysis of Figure 3 can provide information that helps decision-makers to give an appropriate value of $\beta$ and get a better tradeoff. In the left figure, we can find that the delay cost increases almost linearly with the increase of $\beta$. In the right figure, the pace of risk decreasing with $\beta$ is fast at first and then slows down. When $\beta$ is less than 0.4, the risk will decrease obviously while the increase of delay cost is relatively slow; when $\beta$ is more than 0.4, the risk decreases become slow while the increase of delay cost is dramatic. Therefore, adjusting $\beta$ below 0.4 will help us to gain a small risk with a relatively lower delay cost.

3.3. **Generality Analysis of Our SATNFO Model.** In this experiment, we use our 6 generated capacity instances to test the generality of our SATNFO model. Moreover, we have increased the number of flights to 220. The value of $\beta$ is set to 0.4.

In each instance, capacity distribution changes according to its weather spreading mode and impact degree of weather on capacity. Table 2 illustrates the risk in deterministic model and our SATNFO model. The first column represents different capacity instances. The second and third columns list corresponding delay and operational risk obtained by our SATNFO model for each capacity instance. The last column is the operational risk if we do not consider changes of

<table>
<thead>
<tr>
<th>$\beta$</th>
<th>Delay</th>
<th>Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>138</td>
<td>275</td>
</tr>
<tr>
<td>0.3</td>
<td>235</td>
<td>225.4</td>
</tr>
<tr>
<td>0.4</td>
<td>246</td>
<td>219.2</td>
</tr>
<tr>
<td>0.5</td>
<td>276</td>
<td>216.2</td>
</tr>
<tr>
<td>0.6</td>
<td>305</td>
<td>213.8</td>
</tr>
<tr>
<td>0.7</td>
<td>338</td>
<td>211.8</td>
</tr>
<tr>
<td>0.8</td>
<td>396</td>
<td>200.8</td>
</tr>
</tbody>
</table>
Table 2: Delay, risk in SATNFO model, and risk in deterministic model.

<table>
<thead>
<tr>
<th>Instance</th>
<th>Delay-uncertain</th>
<th>Risk-uncertain</th>
<th>Risk-determine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance 11</td>
<td>2260</td>
<td>170.3</td>
<td>512.0</td>
</tr>
<tr>
<td>Instance 12</td>
<td>2151</td>
<td>63.6</td>
<td>296.0</td>
</tr>
<tr>
<td>Instance 21</td>
<td>2415</td>
<td>387.2</td>
<td>581.7</td>
</tr>
<tr>
<td>Instance 22</td>
<td>2318</td>
<td>240.8</td>
<td>338.7</td>
</tr>
<tr>
<td>Instance 31</td>
<td>2454</td>
<td>340.8</td>
<td>500.8</td>
</tr>
<tr>
<td>Instance 32</td>
<td>2365</td>
<td>252.3</td>
<td>285.4</td>
</tr>
</tbody>
</table>

Figure 4: Demonstration of risk in deterministic model and our SATNFO model.

Table 3: Numbers of constraints of SATNFO model and SPNAC model.

<table>
<thead>
<tr>
<th>Model</th>
<th>Flight</th>
<th>Time stage</th>
<th>Sector</th>
<th>Degradation degree</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPNAC</td>
<td>55</td>
<td>4</td>
<td>25</td>
<td>3</td>
<td>494971</td>
</tr>
<tr>
<td>SATNFO</td>
<td>55</td>
<td>4</td>
<td>25</td>
<td>3</td>
<td>42374</td>
</tr>
<tr>
<td>SATNFO</td>
<td>220</td>
<td>4</td>
<td>25</td>
<td>3</td>
<td>165530</td>
</tr>
</tbody>
</table>

3.4. Computational Complexity Analysis. If the numbers of sectors, flights, and time units are denoted by $S$, $F$, and $T$, respectively, then the dimension of decision variable is $|T| \cdot |F| \cdot |S|$. The upper bound of constraints is $2 \times |F| \times |T| \times |F| \times |S| + 2 \times |F| \times |T|$.

Table 3 shows the numbers of constraints in our proposed SATNFO model and in stochastic-programming with nonanticipative constraints (denoted by SPNAC model hereafter) [26]. From the table, we found that to solve a SATNFO problem with the same scale, the constraints in our proposed model are far more less than in the SPNAC model.

4. Conclusion and Future Research

In this paper, we successfully provide a novel weighted biobjective 0-1 integer programming model to deal with the uncertainty in SATNFO problem. In this model, two objectives are the minimization of the cost of total flight delays and the minimization of the operational risk. In order to evaluate the effect of capacity uncertainties on ATFM, the operational risk is specifically defined via probabilistic risk assessment method. By assigning different weight coefficients on risk and delay cost, some efficiency is compromised to make our flight schedule more robust. This model allows decision-makers to adjust risk weight coefficients, so that they can make tradeoff between risk and delay costs. Moreover, because we introduce the operational risk function into objective instead of using capacity constraints under uncertainties, we avoid the dramatic increase of the number of constraints and thus avoid suffering the computational intractability.

Through computation experiment, we find out an appropriate range of risk weight coefficient that enables decision-makers to gain a small risk with a relatively lower delay cost. Successful tests under different capacity distribution instances confirm the generality of our SATNFO model. Moreover, result shows the number of constraints is far less than nonanticipative constraints in stochastic programming, which illustrates the high computational quality of this model.

Our model could be further improved in the future. First, various kinds of uncertainty sources could be considered besides capacity changes. For example, the flight demand is also an uncertain element in air traffic flow management problem. Taking various kinds of uncertainties can make our model closer to the real-world application. Second, the choice between risk and delay cost is totally decided by setting risk difference. From this perspective, our SATNFO model is consistent with the reality, which illustrates its rationality.
weight, which is an artificial decision in this model. Although it is flexible, to some extent, the unreliability is inevitable. It is possible to introduce a feedback mechanism in future studies that the risk weight could be automatically adjusted according to the risk value during the optimization. In this case, we can avoid the unreasonable decision and make our model more reliable.

Conflict of Interests
The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments
This work was supported by the National Science Foundation for Young Scientists of China (Grant nos. 61401011 and 61201314), the National ATC Scientific Research Project (Grant no. GKG201404004), and the National Key Technologies R&D Program of China (Grant no. 2012BAG04B01).

References


Research Article

MRAC Control with Prior Model Knowledge for Asymmetric Damaged Aircraft

Xieyu Xu, Lingyu Yang, and Jing Zhang

1CNIGC Institution of Navigation and Control, Chedaogou No. 10, Haidian District, Beijing 100191, China
2Department of Automation Science and Electrical Engineering, Beihang University, Haidian District, Beijing 100191, China

Correspondence should be addressed to Xieyu Xu; xudavid501@163.com

Received 16 November 2014; Accepted 8 December 2014

Academic Editor: Kemao Peng

Copyright © 2015 Xieyu Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper develops a novel state-tracking multivariable model reference adaptive control (MRAC) technique utilizing prior knowledge of plant models to recover control performance of an asymmetric structural damaged aircraft. A modification of linear model representation is given. With prior knowledge on structural damage, a polytope linear parameter varying (LPV) model is derived to cover all concerned damage conditions. An MRAC method is developed for the polytope model, of which the stability and asymptotic error convergence are theoretically proved. The proposed technique reduces the number of parameters to be adapted and thus decreases computational cost and requires less input information. The method is validated by simulations on NASA generic transport model (GTM) with damage.

1. Introduction

It has been shown that the vehicle damage to airframe and engines had led to quite a few aircraft accidents and fatalities in recent years [1]. Asymmetric aircraft structural damage results in abrupt deterioration on flight performance and handling quality, which impairs safety. Researches to recover aircraft stability and handling qualities, like fault-tolerant flight control, reconfigurable control method, and so forth, have already become a hot spot in related academic fields.

Lots of studies have been conducted to recover control performance under aircraft failures or damages. NASA has launched several aircraft safety projects like IFCS [2] and AvSP [3], which innovate flight control algorithms and experiment on aircraft testbeds. Various flight control laws like the adaptive control designs [4–9], linear quadratic regulator (LQR) designs [6, 10, 11], and robust linear designs [12] were proposed and evaluated by flights. The pilot Cooper-Harper ratings showed that the proposed methods could retain the handling qualities under certain failure cases [4, 13].

Generally, the adaptive control algorithms are quite popular in this field. Reference [14] used an adaptive artificial neural network (ANN) control method to recover the handling qualities of an F-18 model. The method uses a pretrained ANN to provide a model inversion block with aerodynamics and handling characteristics and another online-adjusted ANN to compensate modeling error and failure conditions. Reference [15] extended the result of [14]. It changes the pretrained ANN to an online learning network, which adjusts the inversion block adaptively, resulting in a hybrid direct and indirect adaptive ANN control scheme. Reference [16] designed an angular rate controller using adaptive single hidden layer ANN method for an unmanned airborne vehicle. With a classic linear guidance law, several successful automated landings were made in flight experiments with 25% left wing loss.

Except for ANN adaptive laws, multivariable MRAC with various structures were also studied in this field. Reference [9] combined a direct MRAC method with a parameter estimator using gradient algorithm, obtaining a combined MRAC structure. References [6, 17] proposed a derivative-free MRAC using delayed weight update law. Reference [7] utilized the $L_1$ adaptive law [18] for a flight controller design. Reference [19] analyzed the plant characteristics by LDS decomposition of the high frequency gain matrix (HFGM) of the damaged aircraft. Realizing the signs of the leading
principle minors of the HFGM does not change before and after damage. Along with several other results and assumptions, an MRAC law with output feedback structure is proposed. The method is validated through simulations on a high-fidelity GTM model with left wing tip damage. A similar design can be found in [20], which uses a state feedback structure with adaptive gains instead of the original output feedback structure.

The methods mentioned above do not need to take the characteristics of damage conditions into account. As long as the assumptions hold, these methods can be applied to various damage cases and allow for relatively large parameter changes. However, the range of admissible parameter changes can be so large, in a way that is far enough to cover all damage cases. If proper descriptions can be used to model the concerned damage cases, it is possible to design a controller with parameters adjusted in a smaller region, reducing computation workload and improving performance.

Based on the analysis above, this paper models the structural damaged aircraft with polytope LPVs and model reference controllers (MRCs) are given, which explicitly integrates plant characteristics. The parameters updated by adaptive laws are the polytope interpolation coefficients, instead of control gains in traditional MRAC designs. The number of the adaptive parameters is therefore reduced. The paper is organized as follows. Section 1 introduces the background and design philosophy of the proposed method. Section 2 models and analyzes a damaged aircraft, namely, GTM. A polytope linear model is introduced to describe the concerned damage cases. Section 3 designs the control laws and models and analyzes a damaged aircraft, namely, GTM. A and design philosophy of the proposed method. Section 2 is organized as follows. Section 1 introduces the background

body dynamic equations are deduced. It should be noted that the sensors remain unmoved before and after damage, so that the point which the dynamic equations describe remains unchanged. The dynamic equation can be written as follows:

\[
\begin{bmatrix}
F \\
M
\end{bmatrix} = \begin{bmatrix}
m & -m \Delta I^s \\
m \Delta I^s & I
\end{bmatrix} \begin{bmatrix}
\dot{V} \\
\dot{\omega}
\end{bmatrix} + \begin{bmatrix}
m \omega^s \\
m \omega^s \Delta I^s \\
\omega^s I - mV \Delta I^s
\end{bmatrix} \begin{bmatrix}
V \\
\omega
\end{bmatrix}.
\]

\( F \) and \( M \) refer to the total force and moment relevant to original CG, respectively. \( m, I, \) and \( \Delta \) are the total mass, inertia in original CG, and the offset of CG reference to its origin location. \( V = [u, v, w]^T, \omega = [p, q, r]^T \) are the line speed and angular speed described in the origin body frame of the aircraft. Superscript \( as \) is the skew-symmetric matrix form of a vector, \( (\cdot)^s )y = x \times y \).

The line speed of a specific point generally changes with its location, only except that it moves parallel to the rotation axis of the rigid body. The calculation of the line acceleration has to take the centripetal and the Coriolis force into account, as long as the point is not located on the rotation axis. In short, due to the CG movement by the structural damage, the calculation of line speed becomes quite complicated, coupling with angular speed and angular acceleration. However, if the body frame is selected parallel to the original one, the direction of each axis does not change, and the values of the angular speed are the same wherever the frame is. Thus, the angular speed can be modeled in the body frame at the new CG without changing its value, while the line speed has to be modeled at the same point. Rewrite (1) as follows:

\[
\begin{bmatrix}
m \\
0
\end{bmatrix} \begin{bmatrix}
\dot{V} \\
\dot{\omega}
\end{bmatrix} + \begin{bmatrix}
-m \Delta I^s \\
0
\end{bmatrix} \begin{bmatrix}
V \\
\omega
\end{bmatrix} + \begin{bmatrix}
F \\
M
\end{bmatrix}.
\]

Since the value of angular speed remains unchanged, the symbol \( \omega \) is used as before. \( \Delta I \) and \( M \) are the inertia and moment represented in the new body frame located at CG after damage. The kinetic equations are the same as those of normal aircrafts. \((\phi, \theta, \psi) \) is the Euler angle. \([x_g, y_g, z_g] \) is the position vector relative to the ground:

\[
\begin{bmatrix}
\phi \\
\dot{\theta} \\
\dot{\psi}
\end{bmatrix} = \begin{bmatrix}
1 & \tan \theta \sin \phi & \tan \theta \cos \phi \\
0 & \cos \phi & -\sin \phi \\
0 & \sin \phi \sec \theta & \cos \phi \sec \theta
\end{bmatrix} \begin{bmatrix}
p \\
q \\
r
\end{bmatrix},
\]  

\[
\begin{bmatrix}
\dot{x}_g \\
\dot{y}_g \\
\dot{z}_g
\end{bmatrix} = \begin{bmatrix}
\cos \theta \cos \psi & \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi \\
\cos \theta \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & -\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi \\
-\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta
\end{bmatrix} \begin{bmatrix}
u \\
w
\end{bmatrix}.
\]  

2. Modeling of an Asymmetric Damaged Aircraft with Prior Knowledge

2.1. Nonlinear 6DOF Modeling. This paper studies the GTM from IRAC project in AvSP program carried out by NASA. Due to the limitation of modeling data, this paper only discusses the damage case of left outboard wing tip loss. With data and methods from [21], the characteristics of aerodynamics, mass, inertia, center of gravity (CG) movement, and so forth, can be modeled. With methods from [22], the rigid
With the sensors unmoved, the definitions of airspeed $V$, angle of attack $\alpha$, and sideslip $\beta$ are the same as before. The equations are as follows:

$$\alpha = \arctan \frac{w}{u},$$

$$\beta = \arcsin \frac{v}{V},$$

$$V = \|V\|_2.$$  

(4)

2.2. Polytope Linear Model. Generally, the nonlinear equations of an aircraft can be described as

$$\dot{x}(t) = f(x(t), u(t)).$$  

(5)

$x, u$ represent the state and input vectors, respectively. An operation point, or trim point, should be selected before linearization. It should be noted that the actual damage cases are unknown, which renders the trim point unavailable. A general solution is to select a common operation point for all damage cases. For example, the normal point $(x_0, u_0)$ in [20]. Because the normal point generally is not the trim point for damaged cases, an extra term $f_i$ is added to the state-space equations, as follows [23]:

$$\Delta \dot{x}(t) = \sum A_i \chi_i(t) \Delta x(t) + \sum B_i \chi_i(t) \Delta u(t) + \sum f_i \chi_i(t)$$

$$\chi_i(t) = \begin{cases} 1 & \text{if } (A(t), B(t), f(t)) = (A_i, B_i, f_i) \\ 0 & \text{otherwise} \end{cases}$$

(6)

where $A_i \in \mathbb{R}^{n \times n}$, $B_i \in \mathbb{R}^{n \times m}$ are unknown constant matrices, representing the state transition matrices and control matrices of different patterns and degrees of structural damage, respectively. $f_i \in \mathbb{R}^n$ is the unknown constant disturbance. Various control methods can be designed with (6) [20, 23]. However, $A_i, B_i, f_i$ are unknown constants with weak constraints like the signs of leading principle minors of high frequency gain matrix that do not change, which could lead to relatively large parameter variations. The actual damage cases may not cover all these variations. This paper believes that if proper definitions of different damage cases are introduced, the parameter variations can be limited, which facilitates the controller designs.

Based on the above analysis, the model of damaged aircraft is rewritten in a polytope form:

$$\Delta \dot{x} = A(p) \Delta x + B(p) \Delta u + f(p),$$

(7)

$$[A(p), B(p)] \in \Omega = \text{Co}[\{A_i^*, B_i^*\}, \ldots, \{A_l^*, B_l^*\}],$$

where $[A_i^*, B_i^*]$, $i = 1, \ldots, l$ are known constant matrices, the vertices of polytope $\Omega$. There exist nonnegative coefficients $\alpha_i(p)$; thus, $[A(p), B(p)]$ can be calculated by

$$[A(p), B(p)] = \sum_{i=1}^l \alpha_i(p) [A_i^*, B_i^*],$$

$$\sum_{i=1}^l \alpha_i(p) = 1, \quad 0 \leq \alpha_i(p) \leq 1.$$  

(8)

Parameter $p$ represents different damage patterns and degrees. For any specific damage case, it is assumed that a certain $p^*$ can describe the model of the damaged aircraft. $f(p)$ is the extra unknown constant term, for the aircraft is untrimmed under damage. $f(p)$ also captures other effects from structural damages and is not modelled in the polytope form. Finally the model is written as

$$\Delta \dot{x} = \sum_{i=1}^l \alpha_i(p) A_i^* \Delta x + \sum_{i=1}^l \alpha_i(p) B_i^* \Delta u + f(p).$$  

(9)

Compared to (6), the only unknown variables in (9) are $\alpha = [\alpha_1, \ldots, \alpha_l]^T \in \mathbb{R}^l$ and $f(p) \in \mathbb{R}^n$. Different damage cases simply differs in weighting coefficient $\alpha$ and constant $f(p)$, thus vastly reduces the number of unknown variables.

2.3. Vertex Calculation by High Order SVD Method. As stated before, the less the number of vertices is, the less complicated the controller is. It is crucial to balance between the accuracy and the complexity of the polytope aircraft model with damage of various patterns and degrees. The high order SVD (HOSVD) method is a decomposition method for high order arrays or data tensors. By representing the polytope LTI models into a tensor-product (TP) form, the HOSVD method can be used to reduce the number of vertices [24].

First, a parameter grid is generated on the concerned damage cases. For the left outboard wing tip damage case, a grid of $k$-element vector is used. Each node of the grid represents a different wing tip loss ratio to the semispan. The GTM model with damage is linearized at each node of the grid, written in the system matrix form

$$S_i = \begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix}.$$  

(10)

For every specific case in the concerned damage cases, the model $S(p)$ can be written as the interpolation of the linearized models. That is,

$$S(p) = \sum \alpha_i(p) S_i,$$  

(11)

wherein function $\alpha_i(p)$ is the interpolation coefficient function. The interpolation calculation can be rewritten in a compact TP form. Stack all the linearized models to make the model tensor $\mathcal{S}$ and all the coefficients to make the parameter tensor $\mathcal{U}$. Thus,

$$S(p) = \sum \alpha_i(p) S_i = \mathcal{S} \otimes \mathcal{U}(p).$$  

(12)

To accurately describe the original model, a relatively intense grid has to be generated. The algorithm based on HOSVD method from [24] is adopted in this paper to simplify the grid. By omission of small singular values, the number of models and corresponding coefficients can be reduced, without losing accuracy to the original ones. With this algorithm, the preliminary TP model can be approximated to

$$S(p) = \mathcal{S} \otimes \mathcal{U}(p) \approx \mathcal{S}^* \otimes \mathcal{U}^*(p).$$  

(13)

New models $S_i^*$ and interpolation coefficients $\alpha_i^*(p)$, $i = 1, \ldots, l$, can be extracted from the system tensor $\mathcal{S}^*$ and the
Damage ratio to wing semispan

The curves of the interpolation coefficients to wing tip loss ratios.

Figure 1: The curves of the interpolation coefficients to wing tip loss ratios.

The control algorithm proposed in this paper is not of LMI designs in [24]. The interpolation parameters satisfy \( \sum \alpha_j(i) = 1 \) only, which means extrapolation is admissible in the algorithm. The parameters are treated independent of each other, without restrictions in Figure 1. That is, the 3 parameters can vary freely on the plane \( \sum \alpha_j(i) = 1 \) in the 3D space, while Figure 1 restricts the parameters to a curve. As a result, it is expected that the polytope model can describe more damage cases by relaxing the restrictions on the parameters.

2.4. GTM Model Simplification. Assume the polytope model of GTM is written as

\[
\dot{x}_{\text{GTM}} = A(p)x_{\text{GTM}} + B(p)u + f_{\text{GTM}}(p),
\]

wherein

\[
x_{\text{GTM}}(t) = [q, p, r, V, \alpha, \theta, \beta, \phi]^T,
\]

\[
u(t) = [\delta_e, \delta_a, \delta_r]^T.
\]

The angular rates are to be controlled. Rewrite the state-space equations to treat angular rates as states, while the other signals related to angular rates are treated as measurable disturbances. Dividing the original states into two vectors yields

\[
x_{\text{GTM}}^T = [x^T, z^T],
\]

\[
x \triangleq [q, p, r]^T,
\]

\[
z \triangleq [V, \alpha, \theta, \beta, \phi]^T.
\]

Reformulate the state-space equation as follows:

\[
\begin{bmatrix}
\dot{x} \\
\dot{z}
\end{bmatrix} =
\begin{bmatrix}
A_{xx}(p) & A_{xz}(p) \\
A_{zx}(p) & A_{zz}(p)
\end{bmatrix}
\begin{bmatrix}
x \\
z
\end{bmatrix}
+ \begin{bmatrix}
B_x(p) \\
B_z(p)
\end{bmatrix}u + \begin{bmatrix}
f_x(p) \\
f_z(p)
\end{bmatrix}.
\]
Finally, the polytope model with angular rates as states is shown below:

$$\dot{x} = A_{xx}(p)x + A_{xz}(p)z + B_x(p)u + f_x(p). \quad (20)$$

The model is explained as follows. The states are defined as the angular rates. The state transition matrix is $A_{xx}$. The input of the model is separated into 2 parts, the first part of which is the remaining state $z$ with input matrix $A_{xz}$. The second part is the original input $u$ with input matrix $B_x$, which describes the effects of control surfaces. Because the damaged aircraft is not trimmed, a constant disturbance $f_x(p)$ is added to the equation. For the convenience of the following discussions, (20) is rewritten as

$$\dot{x} = A(p)x + H(p)z + B(p)u + f(p). \quad (21)$$

For the specific case of GTM, the linear model of the undamaged case is shown below:

$$\dot{x} = \begin{bmatrix} -3.0362 & 0 & 0 \\ 0 & -4.6576 & 0.5481 \\ 0 & -0.3932 & -1.0176 \end{bmatrix} x + \begin{bmatrix} 0 \\ -29.0039 \\ 0 \end{bmatrix} z + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} u + \begin{bmatrix} 53.0318 \\ -797.3925 \\ -86.6825 \end{bmatrix} f. \quad (22)$$

It can be seen from the above results that the attitude angles, namely, $\theta$ and $\phi$, have no effect on the angular rates, which can be omitted in the model. The omission is reasonable because the moment of aircraft can be generated only by aerodynamic moment and thrust, and the attitude angles have nothing to do with it. Thus, the state $z$ is simplified to $z = [V, \alpha, \beta]^T$.

Although the input matrix $B$ changes with damage, the characteristics of $B$, like the domination and signs of the diagonal elements, remain unchanged. The numeric changes of nondiagonal elements relative to diagonal ones can also be ignored. Based on the above conclusions, the $B$ matrix is treated as a constant before and after wing tip damage. In short, $B(p) \approx B$. \quad (24)

For the convenience of design, the $B$ matrix adopted in polytope model is the undamaged one.

3. The MRAC Design

Model reference scheme with state-tracking method is adopted to recover the control performance after the aircraft structural damage. The reference model is designed as follows:

$$\dot{x}_m = A_m x_m + B_m r. \quad (25)$$

Hurwitz state transition matrix $A_m$ and input matrix $B_m$ reflect the desired dynamic performance. It is known that given symmetric positive real matrix $Q$, there exists a symmetric positive real matrix $P$, satisfying

$$A_m^T P + P A_m + Q = 0. \quad (26)$$

Considering the angular motion of an aircraft can be directly altered by control surfaces, input matrix $B$ will have full rank to be invertible. The model can be rewritten as

$$\dot{x} = A_m x + B \times (u + B^{-1}(\sum \alpha_i A_i^* - A_m)x + B^{-1}H(p)z + B^{-1}f). \quad (27)$$

Define

$$\mu_i \triangleq B^{-1}(A_i^* - A_m) \quad \text{and} \quad \bar{f} \triangleq B^{-1}f. \quad (28)$$

Thus,

$$\dot{x} = A_m x + B \left( u + \left[ \mu_1 \cdots \mu_l \right] \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_l \end{bmatrix} + \bar{f} \right). \quad (29)$$
Rewrite the polytope constraint \( \sum_{i=1}^{l} \alpha_i = 1 \) to \( \alpha_1 = 1 - \sum_{i=2}^{l} \alpha_i \) and substitute
\[
\dot{x} = A_m x + B \left( u + \mu_1 \right) + \left[ \begin{array}{c} \alpha_2 \\ \vdots \\ \alpha_l \end{array} \right] + \tilde{f}.
\]

This implies
\[
A_m x + B \left( u + \mu_1 \right) = \tilde{x}.
\]

It is assumed that every state of the aircraft can be measured. Matrix \( \bar{M} \) is known, since signal \( \mu_1 \) can be measured and calculated with known signals. The parameter vector \( \bar{\alpha} \) concatenated from \( \alpha_i \) to \( \alpha_l \) and the constant offset \( \tilde{f} \) are unknown in actual damage cases. The control law can be designed as follows to alter the plant dynamic characteristics.

\[
u = B^{-1} B_m r - \mu_1 - \bar{M} \bar{\alpha} - \tilde{f}.
\]

However, \( \bar{\alpha} \) and \( \tilde{f} \) are unknown after damage. With certain-equivalence design principle [25], the control law is designed as
\[
u = B^{-1} B_m r - \mu_1 - \bar{M} \bar{\alpha} - \tilde{f}.
\]

\( \bar{\alpha} \) and \( \tilde{f} \) are estimates of \( \bar{\alpha} \) and \( \tilde{f} \) by adaptive laws, respectively. Define the parameter errors as \( \bar{\alpha} - \bar{\alpha} \) and \( \tilde{f} - \tilde{f} \) and substitute the control law into (30), resulting in
\[
\dot{x} = A_m x + B \left( u + \mu_1 \right) - \bar{M} \bar{\alpha} - \tilde{f}.
\]

The tracking error is defined as \( e = x - x_m \), the dynamics of which can be obtained as
\[
\dot{e} = A_m e - \bar{M} \bar{\alpha} - B \tilde{f}.
\]

The adaptive laws are designed using Lyapunov direct method. The Lyapunov function \( V \) is chosen to be
\[
V = e^T P e + \bar{\alpha}^T \bar{\Gamma}_a^{-1} \bar{\alpha} + \tilde{f}^T \bar{\Gamma}_f^{-1} \tilde{f}.
\]

\( \bar{\Gamma}_a, \bar{\Gamma}_f \) are positive symmetric matrices. The derivative of Lyapunov function \( V \) along the dynamics of system is
\[
\dot{V} = e^T P \dot{e} + e^T \dot{P} e + 2 \bar{\alpha}^T \bar{\Gamma}_a^{-1} \dot{\bar{\alpha}} + 2 \tilde{f}^T \bar{\Gamma}_f^{-1} \tilde{f}
\]
\[
= -e^T Q e - 2 e^T P B \bar{M} \bar{\alpha} - 2 e^T P B \tilde{f}
\]
\[
+ 2 \bar{\alpha}^T \bar{\Gamma}_a^{-1} \dot{\bar{\alpha}} + 2 \tilde{f}^T \bar{\Gamma}_f^{-1} \tilde{f}.
\]

The adaptive laws are selected as
\[
\dot{\bar{\alpha}} = e^T P B \bar{M} \bar{\Gamma}_a,
\]
\[
\dot{\tilde{f}} = e^T P B \bar{\Gamma}_f.
\]

Thus, the derivative \( \dot{V} \) becomes
\[
\dot{V} = -e^T Q e.
\]

The design process of the adaptive law in this paper is quite similar to the classical ones using the Lyapunov method. Properties like stability and asymptotic error convergence can also be proved using same procedures. All states in the system are bounded due to the positivity of \( V \) and seminegativity of \( \dot{V} \), so that the system is stable in the Lyapunov sense. The boundedness of states guarantees that \( V \) is bounded, which means \( V \) is uniformly continuous. Using Barbalat’s lemma, it can be concluded that \( V \) converges to zero asymptotically; that is, the tracking error \( e \to 0 \). In summary, the properties of the proposed method are stated as follows.

**Theorem 1.** The MRAC theme with control law (32) updated by adaptive law (37), when applied to the plant (30), guarantees the closed-loop signal boundedness and asymptotic state-tracking \( e = x - x_m \to 0 \).

**Remark 2.** Thanks to the popularity of the standard Lyapunov design procedure, lots of modifications can be applied to the control law, such as robust modifications [26], predictor-based or reference model modifications to improve transient performance [27–30], and adaptive law modifications [31].

**Remark 3.** The proposed method updates the interpolation coefficients of the feedback gains. The gains are designed beforehand for the vertices of the polytope model, which integrates prior information of different cases of damage into the control law. With prior knowledge of various damage cases, the performance of the design is expected to be improved. The HOSVD and model simplification process reduce the number of vertices and the amount of parameters. Less stimulation is required for the input signal [25].

**Remark 4.** The linearization of the original 6DOF nonlinear model and the vertex simplification using HOSVD method result in an approximated damaged aircraft model. There always exists modeling error between the real model of various damage cases and the developed polytope model. Similar to common adaptive algorithms, the update law has to be modified to ensure closed-loop stability [25]. Modifications like the projection operator have to be applied to the adaptive law in real cases.

**Remark 5.** It should be noted that the maximum deflection of control surfaces is finite in real aircrafts, which poses a constraint on \( u \) in (32) and is not taken into consideration in MRAC design process. It can be seen in (23) that some values in matrix \( H \) and vector \( f_0 \) are quite large, which might saturate \( u \) and cause the aircraft to be uncontrollable in some cases. Actually, if all control surfaces are limited within ±30°, which is even larger than that in [21], it is impossible to maintain a steady cruise in the operation point of (23) in 33% left wing tip loss case. The steady cruise means that no rotation is allowed; thus, the angular speed \([p, q, r]\) keeps zero, and the operation point means the other states in the
linearization model (23), namely, \([V, \alpha, \beta]\), are equal to zero, resulting in
\[
\begin{bmatrix}
-43.3124 & 2.6631 & -0.0442 \\
0.6455 & 22.0466 & 11.3155 \\
0.0211 & 1.8451 & -28.7231
\end{bmatrix}u + \begin{bmatrix}
53.0318 \\
-797.3925 \\
-86.6825
\end{bmatrix} = 0,
\]
\[
u = [\begin{array}{l}
-3.4640 \\
-36.4141 \\
0.6762
\end{array}]^T.
\]

It can be seen that the second element in \(u\) exceeds \(-30^\circ\), which means the aileron is saturated, concluding that it is impossible to balance the constant offset \(f_0\) solely by input \(u\).

However, it is feasible to maintain a straight cruise under some attack angle \(\alpha\) and sideslip \(\beta\), and it is possible to saturate the control surfaces under improper \(\alpha\) and \(\beta\). Due to the left wing tip loss, if the damaged aircraft is flown with relative large lift force, the left roll aerodynamic moment will be too large to be compensated by ailerons. Detailed discussions of flight performance under various damage cases may be found in [32, 33] and are beyond the scope of this paper.

4. Simulation Study

In this section, the developed adaptive law under the obtained linear polytope model is validated by simulations on a NASA GTM nonlinear model with damage. GTM is a 5.5% dynamically scaled twin-turbine powered aircraft model, designed and manufactured in the NASA AvSP program, dedicated to flight testing of research control laws in adverse flight conditions [34].

The damage case with loss of 20% semispan of the left wing tip is selected. Simulations are conducted on the nonlinear 6DOF GTM model, with maximum deflection of the control surfaces limited within \(\pm30^\circ\) degrees. The aircraft is flying normally at the beginning, injected with wing tip damage at 5 s during simulation. The simulation results with minor step commands are shown in Figure 3.

The step input starts at 1 second with value \([-1, 1, 1]\). Figure 3 shows that the response of the undamaged aircraft tracks the reference command from 0 to 5 seconds, which benefits from the proper selection of the initial values of \(\alpha\) and \(f_0\). The damage is injected at 5 seconds, resulting in some transient oscillations during 5–8 seconds. The magnitudes of the oscillations are relatively small and admissible. After the oscillations, the angular rate responses track the reference signals again. Another step input is injected at 10 seconds. The response signal tracks the reference consistently, which proves the performance of the proposed method. The deflections of control surfaces are shown in Figure 4.

Figure 4 shows that all the control surfaces are not saturated; thus, the angular rates are controllable. More explanations can be found in the responses of state \(z\) in (21), shown in Figure 5.

It can be seen that the responses of the other states are relatively small, especially for the angle of attack \(\alpha\). As in Remark 5, (23) indicates that positive \(\alpha\) with offset \(f_0\) can saturate the...
ailerons easily, meaning that inappropriate command might cause loss of stability. If the input is selected as \([q, p, r] = [1, 2, 0]\), simulation shows that positive \(\alpha\) will be generated, leaving the ailerons saturated, the response divergent, and the angular rates uncontrollable. Thus, the pitch angular rate command in this case is selected as \(-1°/s\) to obtain the negative \(\alpha\) response, which helps reduce the deflections of control surfaces and results in a controllable aircraft.

Further demonstration of tracking performance can be shown in sinusoidal command signals, as in Figure 6.

The sinusoidal inputs are arbitrarily chosen as \(q_c = \sin \sqrt{3}t\), \(p_c = 4 \sin 2t\), and \(r_c = \sin \sqrt{2}t\). The damage is injected as 20% left wing tip loss at 5 seconds. As in the step input case, the angular rates track the reference signals before damage, which validates the initial control parameters and the control structure. The transient oscillation lasts from 5 to 8 seconds in the simulation, the magnitude of which is relatively small and admissible. The transient ends at about 8 seconds and the response signals track the reference again. It can be concluded that the proposed MRAC method can restore the handling qualities after the structural damage.

The deflections of control surfaces are shown in Figure 7. It can be seen that the ailerons saturate shortly in the oscillations. However, the ailerons come back to about 18 degrees after the transient, and the saturation does not affect the overall performance of the design. The responses of \(z\) are shown in Figure 8.

It can be seen that the airspeed and air flow angles vary in the same sinusoidal pattern as that in the angular rates. The overall magnitude of state \(z\) is relatively small, which helps reduce the deflections of control surfaces, making it possible to track the reference signals.

In summary, the simulations show that the proposed MRAC method guarantees error convergence and stability in both step input and sinusoidal input. It can be concluded that the proposed method is validated by simulation.

5. Conclusions

In this paper, an MRAC method with prior model knowledge for asymmetric damaged aircraft has been developed. The method is designed for a series of linearized GTM models with the same operation points but with different left
outboard wing tip damage degrees. A polytope model for damaged aircraft with small amount of vertices is obtained first. By representing the vast models of various damage cases in a compact tensor-product form, the number of vertices is reduced by omission of smaller singular values using HOSVD algorithm. An extra constant offset term to the derivatives of states is introduced since the actual trimming values of damaged aircraft are unknown. By interpolating the vertex controllers, the model reference control can be achieved. An adaptive law is deduced to update the interpolation coefficients and the trimming values, finalizing the controller design. The method is theoretically proved with closed-loop stability and asymptotic tracking error convergence. Simulations on a nonlinear GTM with damage validate the design.

Some problems remain unsolved in this paper. It was assumed that the $B$ matrix is constant before and after damage, but actually it varies a bit. The result is a small tracking error in the simulation of sinusoidal inputs. Future research will address the various assumptions above. The performance of the design is expected to be improved.

**Conflict of Interests**

The authors declare that there is no conflict of interests regarding the publication of this paper.

**Acknowledgment**

This research was funded by National Natural Science Foundation of China (Grant no. 61273099).

**References**


[18] C. Cao and N. Hovakimyan, “Design and analysis of a novel L1 adaptive control architecture with guaranteed transient...


Research Article

Coordinated Control of Slip Ratio for Wheeled Mobile Robots Climbing Loose Sloped Terrain

Zhengcai Li and Yang Wang

School of Mechanical Instrumental Engineering, Harbin Institute of Technology, Harbin 150001, China

Correspondence should be addressed to Zhengcai Li; 12b908067@hit.edu.cn

Received 25 July 2014; Accepted 11 August 2014; Published 3 September 2014

1. Introduction

Wheeled mobile robots (WMRs) used in outdoor applications, such as for planetary exploration, often have to navigate loose sloped terrain. WMRs traversing on loose terrain will inevitably encounter the problem of slip between the rigid wheels and the loose soil [1]. An increase in this slip beyond a certain degree will cause the WMR to deviate from the predetermined trajectory and the power consumption to increase and possibly cause failures. Such adverse incidents have been reported for WMRs traversing on loose terrain. For example, the 2005 Opportunity Rover relapsed into loose sand dunes of Purgatory and took five weeks to escape [2]. Similarly, the 2009 Spirit Rover was jammed into the sandy soil of the Home Plate plateau of Mars, owing to which its “mobile” life ended and it became a fixed research platform [3]. The conventional control mechanism of a WMR is designed mainly for navigation in indoor environments, typically by assuming the wheel-terrain interaction as being rigid, that is, without considering the slip or considering it an external disturbance [4]. Indubitably, this assumption of rigidity is reasonable when the velocity and acceleration of WMRs are small or the surface of the traversing path is hard (both of which correspond to a large coefficient of friction) [5]. However, a WMR’s wheel-soil mechanics is substantially different in outdoor applications such as planetary exploration [6, 7], and, hence, the conventional rigidity-based control mechanism could cause loss of control of the WMR [8]. Thus, the control of WMRs on loose terrain is faced with unique challenges.

With this background, many researchers have focused on wheel-soil mechanics and considered it to be an important factor for achieving proper control of WMRs traversing on loose terrain and therefore conducted extensive research in this direction. For example, Iagnemma and Dubowsky verified through numerical analysis and tests that the rigidity-based conventional motion planning and control algorithm can cause severe slip and sinkage problems in WMRs [1]; they then modeled wheel-soil mechanics based on terramechanics [9] and also established the multiple-physical-based control mechanism for the navigation of WMRs on flat, loose terrain [9]. Further, Yoshida et al. extended the multiple-physics-based approaches and proposed a traction control method for reducing the slip ratio to a small value [10]. The method was verified through tests on dry sand, thus confirming that the wheel-slip-based control mechanism is effective in preventing severe sinkage and energy waste [10]. Furthermore,
in order to avoid incoordination among the wheels, Baumgartner et al. proposed a velocity synchronization algorithm [11]; they verified through experiments using field integrated design and operation rovers that the proposed method can reduce the required power and wheel slippage [11]. Ding et al. proposed an online soil parameter estimation method using the linear least-squares method [12] and a modified, simplified model for wheel-terrain interaction [13], and they compared the simplified model with the original one by numerical analysis [14] to proof the precision of the modified, simplified model. Now, with the development of wheel-soil interaction terramechanics for WMRs, the influence of the wheel slip ratio on the energy consumption and traction efficiency of WMRs is well understood [15]. Moreover, it has become feasible to develop more efficient control algorithms for both minimizing the energy consumption and compensating for the traction efficiency loss of WMRs as caused by wheel slip. However, current research is mainly focused on flat, loose terrain [16–19], and research on the control of WMRs traversing loose sloped terrain is rare. The presently available wheel-soil interaction mechanics models are too complex for control design [9]. The wheel-soil interaction model developed previously [10] involves numerous unknown dynamic time-varying parameters, which easily causes computation delay and control error. Therefore, the design of an effective control mechanism based on wheel-soil mechanics is considered a key research issue in the control of WMRs on loose terrain.

To this end, the present study investigates longitudinal slip-ratio-coordinated control of WMRs while they are climbing up a loose slope, through the planning and tracking of the slip based on an analysis of the wheel-terrain interaction mechanism. First, experimental motion analysis is conducted to establish a model for the wheel-soil interaction mechanism. Second, an online planning algorithm for slip ratio that is based on the goal of optimizing drive efficiency is built. Third, a dynamic model is developed for a six-wheel mobile robot climbing loose sloped terrain for the design of its control mechanism. Furthermore, a tracking control method is proposed, wherein individual wheels’ slip ratios are used as state variables and the planning slip ratios obtained by the drive efficiency optimization algorithm are used as the desired input. To improve the robustness and adaptability of the proposed tracking control method, an adaptive neural network designed with a weight error in its adaptability of the proposed tracking control method, an adaptive neural network is designed with a weight error in its adaptability of the proposed tracking control method. Finally, Section 7 concludes the study.

2. Slope-Based Wheel-Soil Dynamic Interaction Model

When WMRs traverse loose sloped terrain, the wheels may undergo rolling, slip, and sliding movements. The slip ratio is usually used to describe wheel slip and sliding and is expressed as follows:

\[ s = \frac{r \omega - v}{v} \]

where \( s \) is the slip ratio, which means the wheel slip ratio in our study; \( \omega \) is the actual angular velocity of the wheel; \( r \) is the effective radius of the wheel; \( v \) is the theoretical translational velocity of the wheel axis; and \( v \) is the actual translational velocity of the wheel axis. From (1), one can see that \(-1 \leq s \leq 1\). The condition \( s = 0 \) indicates pure rolling, wherein the wheel touches the ground at the wheel’s instantaneous velocity center; the relationship \( v = r \omega \) is satisfied in this condition. The condition \( s = 1 \) indicates that the wheels are in a pure slip state, that is, corresponding to time when \( v = 0 \). When \(-1 < s < 1\), the wheel is in the slip state, and, at this time, \( v < r \omega \). When \(-1 \leq s \leq 0\), the wheel is in the sliding state.

When the robot is climbing loose sloped terrain, the wheel-soil interaction is as shown in Figure 1. Here, \( F_N \) is the normal force, \( F_{DP} \) is the draw pull force, and \( T \) is the driving torque. Further, \( \theta_i \) denotes the wheel-soil interaction entry angle, \( \theta_j \) denotes the departure angle, and \( \theta_m \) denotes the angle of maximum normal stress. \( \tau \) and \( \sigma \) are the shear stress and normal stress, respectively, when the wheel-soil
interaction occurs at any point on the surface; \( z \) is the wheel sinkage, and \( \alpha \) is the slope angle.

On the basis of the Reece formula [15], the normal stress is expressed as

\[
\sigma = (k_1 + k_2) \left( \frac{h}{b} \right)^n,
\]

where \( k_1, k_2 \) are soil bearing characteristic parameters, \( h \) is the sinkage, \( n \) is the soil deformation index, and \( b \) is the wheel width.

Considering the influence of slope angle on the distribution of stress on the surface, the normal stress model for a robot climbing a loose slope is expressed as follows:

\[
\sigma_1(\theta) = \left( \frac{k_1}{b} + k_2 \right) (r \cos \alpha) (\cos \theta - \cos \theta_1)^n',
\]

\[
(\theta_m \leq \theta \leq \theta_1),
\]

\[
\sigma_2(\theta) = \left( \frac{k_1}{b} + k_2 \right) (r \cos \alpha)^n'
\]

\[
x \left\{ \cos \left[ \theta_1 - \frac{\theta - \theta_2}{\theta_m - \theta_2} \left( \theta_1 - \theta_m \right) \right] - \cos \theta_1 \right\}^n',
\]

\[
(\theta_2 \leq \theta \leq \theta_m).
\]

\( \theta_m \) is calculated by the empirical formula in (4). \( n' \) is the modified index for the condition of a robot climbing loose sloped terrain. These variables are calculated as

\[
\theta_m = (c_1 + c_2 s) \theta_1,
\]

\[
\theta_2 = c_3 \theta_1,
\]

\[
n' = c_1 + c_2 + c_3.
\]

In (6), \( c_1, c_2 \) are constants determined by soil properties. Generally, \( c_1 \approx 0.35, 0 \leq c_2 \leq 0.25. \) \( c_3 \) is the correction coefficient and is usually in the range \( 0 \leq c_3 \leq 0.075 [15]. \)

On the basis of the Janosi formula [15], we can determine the tangential stress, when the wheel is climbing the loose sloped terrain, as follows:

\[
\tau(\theta) = [c + \sigma(\theta) \tan \varphi] \left\{ 1 - \exp \left[ -j(\theta)/K \right] \right\},
\]

\[
j(\theta) = r \left[ (\theta_1 - \theta) - (1 - s) (\sin \theta_1 - \sin \theta) \right].
\]

In (7), \( c \) denotes the soil cohesion coefficient, \( \varphi \) denotes the soil friction angle, and \( K \) denotes the soil shear modulus of deformation.

Through the analysis of the wheel-soil interaction on loose sloped terrain, the correction equation for the concentrated force or torque of the wheel for WMRs climbing such a terrain can be established as

\[
F_N = rb \int_{\theta_2}^{\theta_1} \sigma(\theta) \cos \theta \, d\theta + rb \int_{\theta_2}^{\theta_1} \tau(\theta) \sin \theta \, d\theta = G \cos \alpha.
\]

In the horizontal direction, the balance equation is expressed as follows:

\[
F_{DP} = F_H - F_R = mv.
\]

Here, \( F_H \) is the soil thrust force and \( F_R \) is the resistance force, and these are expressed, respectively, as

\[
F_H = rb \int_{\theta_2}^{\theta_1} \tau(\theta) \cos \theta \, d\theta,
\]

\[
F_R = rb \int_{\theta_2}^{\theta_1} \sigma(\theta) \sin \theta \, d\theta + G \sin \alpha.
\]

Then, the drawbar pull force can be obtained by the summation of \( F_H \) and \( F_R \) as

\[
F_{DP} = -rb \int_{\theta_2}^{\theta_1} \sigma(\theta) \sin \theta \, d\theta - G \sin \varphi + rb \int_{\theta_2}^{\theta_1} \tau(\theta) \cos \theta \, d\theta.
\]

According to the balance of Euler equations for each wheel, we get

\[
T_i - T_{Ri} = I\omega_i \omega_i.
\]

Here, \( T_i \) is the motor driving torque and \( T_{Ri} \) is the resistance moment, expressed as

\[
T_{Ri} = r^2 b \int_{\theta_2}^{\theta_1} \tau(\theta) \, d\theta.
\]

3. Slip Planning for Six-Wheel Robot Climbing Loose Sloped Terrain

3.1. Relationship between Wheel Slip and Key Performance Indexes of WMR. The key performance indexes of a WMR include its traction efficiency (see Figure 18), thrust coefficient and traction coefficient, and drive efficiency. These indexes are indicators of the mobile performance of a WMR. From the literature [20], a linear combination of draw-pull force and the wheel supporting force approximately gives the equivalent driving torque as

\[
T = F_{DP} \cdot l + F_N \cdot e \approx F_{DP} \cdot r_s + F_N \cdot e,
\]

\[
F_{DP} \approx \frac{T}{r_s} - F_N \cdot \frac{e}{r_s},
\]

Here, \( F_i = T/r_s \) is the thrust force, which is produced by the soil deformation caused by the rotation of the motor-driven wheel. \( F_N \cdot e/r_s \) is the resistance of the soil, and \( e/r_s \) is the wheel resistance coefficient that reflects the resistance of the soil to prevent the rotation of the wheel. The wheel traction is the drawbar pull force minus the soil resistance generated by the traction. The wheel-soil interaction generates draw-pull force that drives the movement of the robot.

Equation (16) can be transformed as

\[
\frac{F_{DP}}{F_t} + \frac{e}{e F_t} \approx 1.
\]
Let PE = $F_{DP}/F_N = F_{DP}r_s/T$ denote the traction efficiency. It indicates how much driving force transforms into the effective draw-pull force. From (17), it is understood that the force generated by the motor rotation and thrust force is partly used to generate traction force and partly to overcome the resistance of the soil. Dividing (16) by the normal load $F_N$ gives

$$
\frac{F_{DP}}{F_N} \approx \frac{T}{(F_Nr_s)} - \frac{e}{r_s}. \tag{18}
$$

Let PC = $F_{DP}/F_N$ be the traction coefficient, which indicates the draw-pull force for providing traction under a unit load, and let TC = $T/(F_Nr_s)$ be the thrust coefficient, which indicates the motor-provided thrust force under a unit load. Then, we have

$$
PC = TC - \frac{e}{r_s}, \tag{19}
$$

$$
PE = \frac{(F_{DP}/F_N)}{(T/F_Nr_s)} = \frac{PC}{TC}. \tag{20}
$$

TE is an important index that denotes the wheel's drive efficiency. Combining with (1), we can express TE as

$$
TE = F_{DP} \cdot \frac{v}{T \omega} = F_{DP} \cdot \frac{r_s(1-s)}{T} = PE(1-s). \tag{20}
$$

From Figure 3, we can see that the key performance indexes of the WMR are optimal at a wheel slip ratio in the range of 0.1 to 0.4. The index TE is a function of PE, PC, and TC; therefore, if we use an optimization algorithm for TE, we can perform dynamic programming at each wheel slip ratio to ensure the least possible wheel sinkage and thus obtain maximal TE of the WMR.

### 3.2. Slip Ratio Planning Algorithm Based on Optimal Drive Efficiency

When robots climb loose sloped terrain with constant angular control (CAC), each parameter of the wheel-soil mechanics is likely to be different, implying that the wheel slip ratios would also possibly be different. This would inevitably lead to “incoordination” between wheels. Therefore, we aim to optimize the drive efficiency for achieving the desired wheel slip ratio and to control the drive torque of all wheels in a coordinated manner, to be able to track the desired slip ratio and reduce the wheel sinkage. Through tracking of the desired slip ratio for coordinating the energy distributions of each wheel, the drive efficiency of all the wheels can be ensured to be maximal. If the wheel width, radius, payload, and contact angle of the wheel with the terrain are known for a particular WMR, it is possible to determine the desired wheel slip ratio for climbing loose sloped terrain through drive efficiency optimization. The slip ratio planning goal based on drive efficiency optimization is as follows: minimum energy consumption and maximum drive efficiency for a unit running distance of a WMR. The following factors/conditions need to be considered to meet this goal: (1) the force equilibrium equation, (2) maintenance of contact of each of the wheels with the ground and a normal force greater than zero, (3) a soil resistance moment less than the maximum motor torque, and (4) wheel-soil mechanics.
Based on the factors/conditions of optimal TE, the expected slip ratio planning algorithm for a target WMR, that is,

$$\min \ J(s_i) = \sum_{i=1}^{6} \left| \frac{d(F_{DP}P_s(1-s_i)/T_i)}{ds_i} \right|,$$

(21)

should meet the following constraints:

s.t.  \( T_i = M_{RI} \leq T_{i,max} \)

\[ |\delta| < 1 \]

\[ F_{DP} = F_{DP}(\theta_{1,2}, \theta_{2,2}) \]

\[ F_{NI} = F_{NI}(\theta_{1,2}, \theta_{2,2}) \]

\[ T_i = T_i(\theta_{1,2}, \theta_{2,2}) \quad i = 1, \ldots, 6. \]

(22)

When the ground mechanics parameters are known, the flowchart of the planning algorithm for the optimal-TE-based desired slip ratio is as shown in Figure 4.

4. Dynamics Modeling of Six-Wheel Robot Climbing Loose Sloped Terrain Based on Wheel-Soil Mechanics

An optimal-TE-based slip ratio can ensure minimum energy consumption and maximum driving efficiency. To design the tracking control law for the optimal-TE-based desired slip ratio that would be in agreement with the actual slip ratio, we first perform dynamics modeling of climbing of WMRs on loose sloped terrain based on wheel-soil mechanics. Here,
Δα is the sinkage incline angle, |Δα| ≤ ζ, and ζ is a constant. A small ζ implies linearized derivation, sin(α + Δα) ≈ sin α + sin Δα, and, using (11)–(13),

\[
s_i = -\frac{(1 - s_i) \left[ \sum_{i=1}^{6} F_{DP}(\theta_{1i}, \theta_{2i}) - G \sin (\alpha + \Delta \alpha) \right]}{M_V} + \frac{r_s (1 - s_i)^2 \left[ T_i - T_{Ri}(\theta_{1i}, \theta_{2i}) \right]}{I_{WV}}.
\] (24)

Next, we define the output function as

\[
h(x) = [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6]^T = [s_1 \ s_2 \ s_3 \ s_4 \ s_5 \ s_6]^T.
\] (25)

Combining (23) and (24), we can obtain the standard form of affine nonlinear systems as

\[
\dot{x} = f(x) + g(x)u + d(x),
\]

\[
h(x) = x,
\]

where

\[
x = [s_1 \ s_2 \ s_3 \ s_4 \ s_5 \ s_6]^T,
\]

\[
u = [T_1 \ T_2 \ T_3 \ T_4 \ T_5 \ T_6]^T,
\]

\[
f(x) = [f_1(x) \ f_2(x) \ f_3(x) \ f_4(x) \ f_5(x) \ f_6(x)]^T,
\]

\[
f_i(x) = -\frac{(1 - s_i) \left[ \sum_{i=1}^{n=6} F_{DP}(\theta_{1i}, \theta_{2i}) - G \sin \alpha \right]}{M_V} + \frac{r_s (1 - s_i)^2 \left[ T_i - T_{Ri}(\theta_{1i}, \theta_{2i}) \right]}{I_{WV}},
\]

\[
g(x) = [g_1 \ \cdots \ g_6], \quad g_i(x) = \frac{r_s (1 - s_i)^2}{I_{WV}},
\]

\[
d(x) = [d_1(x) \ d_2(x) \ d_3(x) \ d_4(x) \ d_5(x) \ d_6(x)]^T,
\]

\[
d_i(x) = \frac{(1 - s_i) G \sin \Delta \alpha}{M_V}.
\] (27)

Then, (26), that is, an equation for a multiple-input, multiple-output (MIMO) system, can be decomposed into (28) for six single-input, single-output (SISO) subsystems. Further, we can design a controller for a SISO subsystem:

\[
\dot{x}_i = f_i(x) + g_i(x)u_i + d_i(x), \quad i = 1, \ldots, 6.
\] (28)

5. Radial Basis Function-Based Adaptive Sliding Tracking Control for Desired Slip Ratio

5.1. Approximation Properties of Radial Basis Function Network. A radial basis function (RBF) network is a kind of
a three-layer forward neural network. The first layer is the input, made up of source nodes. The input vector map is directly connected to the second hidden layer, and the third layer is the output. The hidden layer is a transformation function unit, called a Gaussian function. Generally, an RBF network is expressed as the excitation function of a hidden unit as follows:

$$\phi_i(\kappa) = \exp \left[ -\frac{1}{2b_i^2} \| \kappa - \xi_i \|^2 \right], \quad i = 1, 2, \ldots, n.$$  \hfill (29)

Here, $n$ is the hidden unit number, $\kappa$ is the network input vector, and $\xi_i$ and $b_i$ are the $i$th basis functions for the center and radius, respectively. The output layer node is the $i$th for nodes in the hidden layer output linear weighted sum:

$$y_i = \sum_{j=1}^{m} w_{ij} \phi_j(\kappa),$$  \hfill (30)

where $w_{ij}$ is the weight between $\phi_j$ and $y_i$.

For any continuous nonlinear function $f(\kappa)$, one can use the approximate representation of the RBF network as

$$f(\kappa) = W^T \phi(\kappa) + \varepsilon(\kappa),$$  \hfill (31)

where $W$ is the weight matrix of the RBF network and $\varepsilon(\kappa)$ is the reconstruction error of the neural circuits. The literatures [21, 22] have proved that the RBF network can approximate any continuous nonlinear function of arbitrary precision.

5.2. RBF-Based Adaptive Sliding Tracking Control Law. For the system given in (28), if $f_i$ and $g_i$ are precisely known, one can design the general sliding mode control system for tracking the desired slip ratio. When WMRs climb loose sloped terrain, the optimal slip ratio tracking error for the $i$th wheel is $e_i = s_i - s_{id}$, where $s_i$ is the $i$th actual feedback wheel slip ratio and $s_{id}$ is the wheel's desired slip ratio.

We express the sliding-mode surface as

$$S_i = \rho_i e_i = \rho_i (s_i - s_{id}),$$  \hfill (32)

where $\rho_i$ is a proportional coefficient, generally a normal number, and is used to determine the attenuation speed of the tracking error. The ideal sliding mode can be expressed as

$$\dot{S}_i = \rho_i (\dot{s}_i - \dot{s}_{id}) = 0.$$  \hfill (33)

In order to meet the arrival condition of sliding mode variable structure control and in the shortest time to reach the sliding mode surface, to ensure high robustness of the system, the following is the improved index-based control law:

$$\dot{S}_i = -\epsilon_i |S_i| \text{sgn}(S_i) - k_i S_i.$$  \hfill (34)

Here, $\epsilon_i > 0$, $k_i > 0$, and $\text{sgn}(S_i)$ is a symbolic function. Upon substituting (28) into (33), we obtain the system control law as

$$u_i = \frac{1}{\rho_i g_i(x,v)} \left[ \rho_i f_i(x,v) + \epsilon_i |S_i| \text{sgn}(S_i) + k_i S_i - \rho_i \dot{s}_{id} \right].$$  \hfill (35)

At this point, consider the following Lyapunov function candidate:

$$V = \sum_{i=1}^{6} V_i, \quad V_i = \frac{1}{2} S_i^2.$$  \hfill (36)

Differentiating it gives

$$\dot{V}_i = S_i \dot{S}_i = S_i (-\epsilon_i |S_i| \text{sgn}(S_i) - k_i S_i)$$

$$= -\epsilon_i S_i |S_i| \text{sgn}(S_i) - k_i S_i^2$$

$$= -\epsilon_i S_i^2 - k_i S_i^2 < 0,$$

where $V \geq 0$ and $\dot{V} < 0$ are guaranteed to be negative, implying $V \rightarrow 0$ and also $S \rightarrow 0$ and $\dot{S} \rightarrow 0$ as $t \rightarrow \infty$. Therefore, global stability is guaranteed by the Lyapunov theorem.

However, in actual circumstances, $f_i$ and $g_i$ contain many unknown parameters, some of which are dynamic time-varying and therefore cannot be known precisely. Hence, this problem imposes a major limitation on the ideal sliding mode controller given in (35). An RBF neural network possesses the property of an approximate arbitrary nonlinear function, which is not applicable for precision parameters. Therefore, we consider using neural networks with dynamic approximate values $\tilde{f}_i$ and $\tilde{g}_i$ of variables $f_i$ and $g_i$ with unknown parameters, based on the RBF neural network function approximation theory.

Let us assume that $\tilde{f}_i$, $\tilde{g}_i$ have approximation weights $w_{fi}$, $w_{gi}$ and that the $f_i$, $g_i$ approximation errors are $\varepsilon_{fi}$ and $\varepsilon_{gi}$, respectively. That is,

$$\tilde{f}_i(x) = \tilde{f}_i(x | w_{fi}) = w_{fi}^T \phi(x) + \varepsilon_{fi},$$  \hfill (38)

$$\tilde{g}_i(x) = \tilde{g}_i(x | w_{gi}) = w_{gi}^T \phi(x) + \varepsilon_{gi}. $$  \hfill (39)

Because the RBF neural network is used to approximate an unknown nonlinear function in the system, an approximation error inevitably exists. The neural network fitting error should be reduced as possible, so we define $\bar{w}_{fi}$ and $\bar{w}_{gi}$ as the optimal weights to estimate $w_{fi}$ and $w_{gi}$:

$$\bar{w}_{fi} = \arg \min_{w_{fi} \in \Omega_{fi}} \sup_{x \in \Omega_x} \left| \tilde{f}_i(x | w_{fi}) - f_i(x) \right|,$$  \hfill (40)

$$\bar{w}_{gi} = \arg \min_{w_{gi} \in \Omega_{gi}} \sup_{x \in \Omega_x} \left| \tilde{g}_i(x | w_{gi}) - g_i(x) \right|.$$  \hfill (41)
In (40) and (41), $\Omega_x \subset \mathbb{R}^n$, $\Omega_{f_i}$ and $\Omega_{g_i} \subset \mathbb{R}^m$, $\Omega_x$, $\Omega_{f_i}$, and $\Omega_{g_i}$ all are compact sets further and $n$ and $m$ are the input layer and hidden layer nodes, respectively. As a result, $\tilde{f}_i(x | \tilde{w}_{f_i})$ and $\tilde{g}_i(x | \tilde{w}_{g_i})$ are the optimal approximations of $f_i(x)$ and $g_i(x)$, respectively. Then, we define the weights of the neural network error as

$$
\eta_{f_i} = \tilde{w}_{f_i} - w_{f_i}, \quad \eta_{g_i} = \tilde{w}_{g_i} - w_{g_i},
$$

where $\dot{\eta}_{f_i} = \dot{\tilde{w}}_{f_i} - \dot{w}_{f_i}$, $\dot{\eta}_{g_i} = \dot{\tilde{w}}_{g_i} - \dot{w}_{g_i}$.

Taking the weights of the neural network, $w_{f_i}$ and $w_{g_i}$, as the adaptive control law, we get

$$
\dot{w}_{f_i} = -\gamma_{f_i} S_{f_i} \phi_{f_i}, \quad \dot{w}_{g_i} = \gamma_{g_i} S_{g_i} \phi_{g_i} u_i.
$$

Based on (35), (39), (40), and (43), we get the slip ratio tracking control law using RBF neural network adaptive sliding method, expressed as

$$
u_i = \frac{1}{\rho_i \tilde{g}_i(x)} \left[ \rho_i \tilde{f}_i(x) + \varepsilon_i \| S_i \| \text{sgn} (S_i) + k_i S_i - \rho_i s_{di} \right].
$$

5.3. Stability Analysis. Let us define the minimum approximation error of a neural network as

$$
\delta_i = \left[ \tilde{f}_i(x | \tilde{w}_{f_i}) - f_i(x) \right] + \left[ \tilde{g}_i(x | \tilde{w}_{g_i}) - g_i(x) \right] u_i.
$$
Then, substituting (44) and (45) into (37) gives
\[
\dot{S}_i = \rho_i \left( f_i (x) + g_i (x) u_i \right) - \rho_i \dot{s}_i \frac{d}{d_i}
\]
\[
= \rho_i \left( f_i (x) + \bar{g}_i (x | w_{gi}) u_i + g_i (x) u_i \right)
\]
\[
- \bar{g}_i (x | w_{gi}) u_i - \rho_i \dot{s}_i \frac{d}{d_i}
\]
\[
= f_i (x) - \bar{f}_i (x | w_{fi}) + \rho_i \dot{x}_i - k_i \text{sgn} S_i + g_i (x) u_i
\]
\[
- \bar{g}_i (x | w_{gi}) u_i - \rho_i \dot{s}_i \frac{d}{d_i}
\]
\[
= f_i (x) - \bar{f}_i (x | w_{fi}) + \left[ g_i (x) u_i - \bar{g}_i (x | w_{gi}) u_i \right]
\]
\[
- k_i \text{sgn} S_i
\]
\[
\dot{\bar{f}}_i (x | \bar{w}_{fi}) - \bar{f}_i (x | w_{fi})
\]
\[
+ \left[ \bar{g}_i (x | \bar{w}_{gi}) - \bar{g}_i (x | w_{gi}) \right] u_i - \delta_i - k_i \text{sgn} S_i
\]
\[
= \left( \bar{w}_{fi} - w_{fi} \right) \phi_{fi} (x) + \left( \bar{w}_{gi} - w_{gi} \right) \phi_{gi} (x) u_i
\]
\[
- \delta_i - k_i \text{sgn} S_i
\]
\[
= \eta_{fi} \phi_{fi} (x) + \eta_{gi} \phi_{gi} (x) u_i - \delta_i - k_i \text{sgn} S_i.
\]

Consider the following Lyapunov function candidate:
\[
V = \sum_{i=1}^{6} V_i,
\]
\[
V_i = \frac{1}{2} \left( S_i^2 + \frac{1}{\gamma_{fi}} \eta_{fi}^T \eta_{fi} + \frac{1}{\gamma_{gi}} \eta_{gi}^T \eta_{gi} \right).
\]

Differentiating it gives
\[
\dot{V}_i = S_i \dot{S}_i + \frac{1}{\gamma_{fi}} \eta_{fi}^T \dot{\eta}_{fi} + \frac{1}{\gamma_{gi}} \eta_{gi}^T \dot{\eta}_{gi}
\]
\[
= S_i \left( \eta_{fi}^T \phi_{fi} (x) + \eta_{gi}^T \phi_{gi} (x) u_i - \delta_i - k_i \text{sgn} (S_i) \right)
\]
\[
+ \frac{1}{\gamma_{fi}} \eta_{fi}^T \dot{\eta}_{fi} + \frac{1}{\gamma_{gi}} \eta_{gi}^T \dot{\eta}_{gi}
\]
\[
= S_i \eta_{fi}^T S_i + \frac{1}{\gamma_{fi}} \eta_{fi}^T \eta_{fi} + S_i \eta_{gi}^T \phi_{gi} (x) u_i
\]
\[
+ \frac{1}{\gamma_{gi}} \eta_{gi}^T \eta_{gi}
\]
\[
= \frac{1}{\gamma_{fi}} \eta_{fi}^T \left( \eta_{fi} + \gamma_{fi} S_i \phi_{fi} (x) \right)
\]
\[
+ \frac{1}{\gamma_{gi}} \eta_{gi}^T \left( \eta_{gi} + \gamma_{gi} S_i \phi_{gi} (x) u_i \right) - \delta_i S_i - k_i |S_i|^2
\]
\[
= -\delta_i S_i - k_i |S_i|^2.
\]

Based on the approximation theory of the RBF neural network, as long as numerous hidden layer nodes exist, the adaptive RBF neural network can give an infinitesimal value of the approximation error. Then,
\[
\dot{V}_i \leq 0, \quad \dot{V} \leq 0.
\]
Then, to reduce chattering that occurs usually, boundary layer methods are used, wherein a saturation function is employed instead of a signum function:

\[
\text{sat}(S_i, \phi_i) = \begin{cases} 
1, & S_i > \phi_i \\
 k_i S_i, & |S_i| \leq \phi_i, k_i \phi_i = 1 \\
-1, & S_i < -\phi_i,
\end{cases}
\]  

(50)

where \( \phi_i \) is the normal to the boundary layer thickness.

The flowchart for optimal slip ratio tracking control based on the RBF neural network adaptive sliding mode in the case of a WMR climbing loose sloped terrain is shown in Figure 7.

6. Simulation Experiment Based on RoSTDyn Platform

6.1. RoSTDyn. RoSTDyn is a multibody dynamics simulation platform for WMRs moving on loose sloped terrain; it uses a Vortex engine and VC++ and was developed by the RCAMC Laboratory. Here, VC++ was used to generate the WMR system, terrain module, and wheel-soil interaction mechanics model, and the kinetic function and scene function provided by the Vortex engine were used for calculations and 3D realization. The basic simulation outline is as shown in Figure 8. From the literature [23], it is known that the RoSTDyn robot provides precise simulations on loose sloped terrain through comprehensive testing.

6.2. Simulation of Control of Six-Wheel Lunar Rover Climbing Loose Sloped Terrain. Consider a six-wheel lunar rover as an example. When it is climbing a slope on the moon, the most influential factor is the slip ratio of the wheels. Using the proposed control algorithm (RBFAS) to adjust the drive torque to track the desired slip ratio, the lunar rover can effectively climb up a slope with the least possible wheel sinkage and maximum drive efficiency of all the wheels.

This simulation is performed as follows. The lunar rover is started from its initial stationary state. Then, the desired slip ratio based on the optimal TE is input into the slip ratio tracking algorithm. The wheel drive moment is adjusted to control the wheel slip ratio and obtain the optimal slip ratio.

The system parameters for the simulation are set as follows.

(a) Robot parameters are the following: \( M = 120 \text{ kg} \), \( m_w = 1.75 \text{ kg-m}^2 \), \( r = 0.15 \text{ m} \), \( b = 0.15 \text{ m} \), and \( h = 0.01 \text{ m} \).
Figure 13: Front-wheels driving torque comparison of different control strategy.

Figure 14: Middle-wheels driving torque comparison of different control strategy.

Figure 15: Behind-wheels driving torque comparison of different control strategy.

Figure 16: Power consumption comparison of different control strategy.

(b) Sandy soil characteristic parameters are the following: $k_1 = 1800$, $k_2 = 820000$, $g = 1.6333\ m/s^2$, $c = 520$, $\varphi = 42^\circ$, $K = 0.01732$, $c_1 = 0.35$, $c_2 = 0.042$, and $c_3 = 0.012$.

(c) Control parameters are the following: $\rho_i = 10$, $\sigma_i = 0.17$, $b_i = [10\ 10\ 10\ 10]^T$, $\gamma_{fi} = 5$, $\gamma_{gi} = 1$, and $\omega_{fi} = \omega_{gi} = [6\ 6\ 6\ 6]$.

Figure 9 shows images of the six-wheel lunar rover climbing the sloped terrain in the RoSTDyn 3D simulation platform. Figures 10, 11, 12, 13, 14, 15, and 16 show the key performance indices of WMR comparisons climbing up a 25° slope under RBFAS and CAC control strategies.

The simulation results in Figures 10–15 show that the RBFAS control algorithm given in (45) is superior to those using CAC. Figure 17 shows that the RBFAS control algorithm in (45) significantly reduces the energy consumption and drive efficiency of the six-wheel lunar rover.

7. Conclusion

In this study, we analyzed wheel-soil interaction based on traditional terrain mechanics for WMRs climbing loose sloped terrain and determined the influence of key performance indexes of the WMRs on the wheel slip ratio. We developed an online slip ratio planning algorithm based on the optimal drive efficiency (TE) of the WMRs. Next, using
the optimal-TE-based slip ratio as the input, the actual slip ratio as the state variable, and the wheel drive moment as the control input, we established a tracking system for the optimal-TE-based slip ratio using the method of nonlinear decoupling design. This was done with the aim of improving the robustness and adaptability of the tracking system. An adaptive neural network was used and a weight error in the weight rate was introduced in this network. The control stability of the system was confirmed using the Lyapunov method. Finally, full-scale simulations were performed to verify that the proposed control scheme not only retains the stability of the system but also improves the robot's mobile performance significantly.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was financially supported by the Natural Science Foundation of China (nos. 50975059 and 61005080), the National Postdoctoral Fund (no. 20100480994), the Postdoctoral Fund in Heilongjiang Province, and the “III” Project (no. B07018).

References


