

RESEARCH ARTICLE

Probabilistic Conflict Detection Using Heteroscedastic Gaussian Process and Bayesian Optimization

DUC-THINH PHAM^{ID}, YASH GULERIA, SAMEER ALAM^{ID}, AND VU DUONG^{ID}

Air Traffic Management Research Institute (ATMRI), Nanyang Technological University, Singapore 637460

Corresponding author: Duc-Thinh Pham (dtpham@ntu.edu.sg)

This work was supported in part by the National Research Foundation, Singapore; and in part by the Civil Aviation Authority of Singapore under the Aviation Transformation Programme.

ABSTRACT Conflict detection plays a crucial role in ensuring flight safety and efficiency and is a critical component of an air traffic control system. Despite the availability of tools to support air traffic controllers in identifying potential conflicts, their quality, and accuracy remain limited due to the challenge of accurately accounting for uncertainty when predicting flight trajectories. To tackle this issue, researchers have explored various studies focused on using probabilistic techniques to model aircraft dynamics and trajectory uncertainty. However, these approaches share several common shortcomings, including their assumptions about uncertainty distributions and the high computational costs of detecting and calculating the risk of conflicts. In response to these challenges, we propose a data-driven approach combining a multi-output generative model with a Bayesian Optimization algorithm to effectively model the uncertainty of aircraft trajectories and rapidly identify the probability of a conflict. Our approach employs the Heteroscedastic Gaussian Process to capture complex trajectory patterns and uncertainty from historical data directly. The proposed predictive model can effectively capture heteroscedastic noise from real data, leading to improved predictions. It achieves Kullback-Leibler divergence around 1 to 1.3 for all dimensions which reduces by $> 45\%$ for latitude, $> 24\%$ for longitude, and 4% for altitude compared to the classical homoscedastic GP approach. The method also boasts high-performance predictions for 4D trajectories including descending, climbing, and en-route phases. To pinpoint when two aircraft are most likely to experience a conflict, the Bayesian Optimization algorithm is adopted, which shows good performance in terms of computational efficiency and flexibility for probabilistic conflict detection. The proposed model achieves percentage error $< 0.25\%$ in estimating the conflict probability with computational cost $< 14s$. By addressing the challenges of uncertainty and computational complexity, our method demonstrates great potential to enhance flight safety and efficiency.

INDEX TERMS Air traffic management, ADS-B data, Bayesian optimization, probabilistic conflict detection, uncertainty modeling.

I. INTRODUCTION

Conflict detection and resolution (CDR) is critical in maintaining flight safety and efficiency in an air traffic control (ATC) system. The CDR problem is typically decomposed into two sub-problems: conflict detection (CD) and conflict resolution (CR). A conflict between any two aircraft is defined as the violation of vertical and lateral

minimum separations between them in en route airspace, e.g., $s_v \leq 1000$ feet or 10 flight level (FL) and $s_h \leq 5$ nautical mile (NM) [1]. In the CD problem, the primary task is to foresee potential conflicts between aircraft. As such, this task heavily relies on the prediction of the aircraft's future locations or trajectories. The stochastic nature of the air transportation environment poses challenges in predicting these future locations. This stochasticity originates from different factors such as weather, wind, unexpected airspace closure or other events, pilots' intention in free flight mode,

The associate editor coordinating the review of this manuscript and approving it for publication was Wentao Fan^{ID}.

etc. Therefore, incorporating these uncertainties during trajectory prediction and conflict detection is important to achieve accurate conflict detection.

Several approaches for conflict detection in structured as well as free flight operations have been proposed in the literature. These approaches can be classified into three main categories: deterministic, worst-case, and probabilistic. Probabilistic approaches are considered a promising direction since they can incorporate different types of uncertainties in conflict prediction. In a probabilistic setting, some studies considered empirical distribution model of future aircraft positions [2], [3] or modeling the dynamics of aircraft [4], [5], [6] with certainty. Other works include a probabilistic approach to identify conflicts at the merging waypoints [7] and the use of random walks with unknown intents to calculate the conflict probability at each moment of discretized time [8]. Two main approaches in probabilistic conflict detection are the analytical approach [3] and the Monte Carlo simulation [9], [10]. The analytical approach requires an actual closed-form solution which is often impractical due to the complexity of the real-world scenarios. It also requires imposing certain constraints on the aircraft dynamics and the incorporated uncertainty for the computation of conflict probability. Monte Carlo simulation is a more practical approach because it can incorporate a more complicated state-space model with flexible uncertainty modeling (e.g., non-Gaussian, multi-modal). Especially when additional information, such as weather or pilots' intent, is available, new uncertainties can be included. However, the most noticeable disadvantage of the Monte Carlo simulation is the high computational resource required to run a huge number of iterations. Yang et al. [10] propose an algorithm to speed up Monte Carlo simulation by simplifying the aircraft dynamics model. In one of the recent works, researchers have used a combination of analytical and monte-carlo simulation to address the issues of computational cost, albeit in 2-D (latitude and longitude), without significant improvement in computational speed [11]. The authors in [12], [13], [14], [15], and [16] also try to detect probabilistic conflict at a discretized time by computing the conflict probability at each moment with assumptions about uncertainty and aircraft dynamics. These approaches share common limitations, such as their assumption about uncertainty distribution and high computational cost for estimating conflict probability. Moreover, the computational cost of conflict detection is also a bottleneck in the study of conflict resolution and poses a challenge for evaluating a large number of candidate resolution maneuvers. To overcome those limitations, data-driven and machine-learning approaches have been investigated for trajectory prediction [17], [18], [19] without the need for an explicit model of weather and aircraft dynamics, etc. Since there are no explicit formulae to estimate conflict probability in these learning-based approaches, an efficient method based on the Monte Carlo simulation method is a potential candidate for estimating conflict probability between aircraft.

In this study, we combine the Gaussian Process (GP) to represent the uncertainty in flight trajectories with Bayesian Optimization (BayesOpt) to locate the point with the maximum conflict probability efficiently. The use of the Gaussian process for trajectory prediction is inspired by [20]. First, the Gaussian Process is used as a generative model to capture the complex trajectory pattern and its uncertainty directly from trajectories. This generative model must be able to address two primary challenges: the ability to produce multi-output values and the heteroscedasticity of data. Our work incorporates the 3-D position of the aircraft (longitude, latitude, and altitude) and, thus, is a multi-output problem. To handle this attribute, we train three GPs, one for each dimension. Thus, our generative model is, in fact, a set of three GP models. The work in [21] develops a Homoscedastic GP model to capture the consistent variance in data. In our work, the variations in longitude, latitude, or altitude of aircraft positions vary with time. Therefore, the heteroscedastic GP [22] is adopted to capture the variance of the considered data, while the classical homoscedastic GP is used as the baseline for comparison. After the development of the generative model, the current flight information (conditional points) such as current position and reference route is provided to the model for training the trajectory prediction model with uncertainties. These uncertainties are obtained by considering data variation on conditional points using a second GP. We also apply the Bayesian optimization algorithm for probabilistic conflict detection. We aim to obtain similar results as the Closest-Point-of-Approach (CPA) but in a probabilistic manner by identifying the time at which two aircraft have the highest probability of conflict. This is called Probabilistic Closest-Point-of-Approach (P-CPA). Since conflict is a rare event, the cost of estimating conflict probability is high. Our proposal method is faster and more flexible in finding P-CPA compared to the grid-search approaches and the existing methods. The proposed approach and the main results are also reported as a part of the first author's doctoral dissertation [23].

II. ADS-B DATA

In this study, Automatic Dependent Surveillance-Broadcast (ADS-B) data is used for training and evaluating the proposed approach. An en-route sector (Sector 2E) within Kuala Lumpur FIR, managed by the Singapore Air Traffic Center (ACC) for providing air traffic service from FL120 to FL360 inclusive, is selected for further analysis. Figure 1 depicts the spatial characteristics of the selected sector. It takes about 5 minutes on average for a typical flight to cross the sector. The sector contains 8 waypoints and is crossed by 8 Air Traffic Service (ATS) Route. There is one crossing in the sector and one converging point in the south of the sector. The spatial characteristics and the airspace structure of the sector are simple, therefore the scenario of the sector can be easily identified. We used the ADS-B data for December 2016 for analysis.

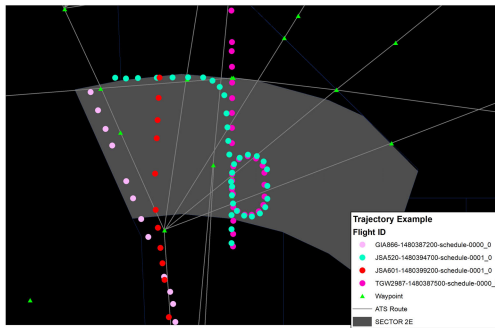


FIGURE 1. The layout of Sector 2E, with its waypoint and ATS routes.

The original ADS-B data set is a large data set with noise and missing data points. Moreover, with the given spatial information of Sector 2E, only a subset of trajectories should be considered and investigated. Thus, some data pre-processing steps are required:

- 1) First, we apply a 2D spatial filtering to filter out all trajectories which do not pass through the sector. A total of 12,141 flights that passed through Sector 2E in Dec 2016 were extracted.
- 2) A second 2D spatial filtering is applied on trajectories passing by Sector 2E to filter out data points outside of the sector. It is separated from Step 1 since the criteria for filtering can be changed in the future for different information extraction.
- 3) To deal with missing data points, we remove all the flight trajectories that have less than three data points in the sector. Thus, the final dataset size for training and evaluation of our approach is 9,082 flights.

Flight trajectories can be grouped using various metrics such as entry/exit points, maneuvers inside the sector, flight levels, etc. Different flight trajectory patterns between entry and exit points can be due to the maneuver instructions provided by the controllers or requests of pilots in response to traffic or to avoid regions of turbulent weather. From our observations and previous studies, given similar entry information, we can obtain similar 4D exit points for the flights. Thus, for en-route sectors, we propose grouping flights based on their entry and exit information to form flight routes, as shown in Figure 2. The variation, observed within each group of flights, is the uncertainty of a flight route. The final dataset contains 42 groups of flights which are utilized for training and evaluating the proposed approach.

III. OVERVIEW OF THE PROPOSED METHODOLOGY

The proposed approach (Figure 3) consists of two components: a predictive model and a Bayesian optimization model. In the first step, historical trajectories are clustered in terms of the locations of their entry to and exit from the airspace sector. Trajectories from the same cluster are considered to share a common entry-exit pair; however, due to uncertainty, their actual flight paths from the entry to the exit are varied. As discussed, the trained generative

model is a set of three Gaussian processes. To perform predictions for the incoming flights, recent positions and future reference positions of the flights are provided as conditional points. Given these conditional points, the dataset is filtered to identify a subset of trajectories that pass through all the conditional points. We then train our GP predictive models using filtered trajectories and learned flight position uncertainty.

The current work is focused on two aircraft conflict detection. Thus, predictive models are trained to predict the future locations of each aircraft and to be used for estimating their conflict probability. This conflict detection problem is formulated as an optimization problem, in which we identify a timestamp t when the conflict probability is the highest. The proposed Bayesian optimization algorithm is discussed in more detail in section V. For a given a timestamp t , we estimate conflict probability $P(t)$ using the Monte Carlo simulation method. This result is stored in the memory and also used to update a probabilistic model, called the surrogate. The surrogate function is an approximation function of the original function but needs less time to evaluate. Then the acquisition function, a mathematical technique for guiding the exploration of the parameter space during Bayesian optimization, uses the updated surrogate function to make the prediction of the next value of t to be used for evaluation. The searching process stops when the number of iterations i reaches the maximum number of evaluations Max_Eval .

This proposed approach is designed to perform probabilistic conflict detection in the short and medium term with a time horizon varying between 5 and 20 minutes. The predicted conflict probability can be provided to air traffic controllers (ATCO) to improve their situation awareness. A binary conflict detection can also be extracted from this probability by providing a threshold. Since air traffic control and air traffic are dynamic, the appropriate threshold should be adjustable and depending on each ATCO, which will be investigated in our future work for conflict detection with human input. Moreover, in this approach, the predictive models learn the heteroscedastic noises from the historical realization of the planned trajectory (e.g., conditional points), sufficient data is required to provide accurate estimation. In situations where the planned trajectory is non-nominal, the default uncertainty distributions can be used to generate the data for the input of this approach. In that case, the performance of this approach approximates the accuracy of the default uncertainty distributions which can be adopted from state-of-the-art studies for trajectory uncertainty modeling.

IV. PREDICTIVE MODEL FOR FLIGHT TRAJECTORY WITH LEARNED UNCERTAINTY

A. PROBABILISTIC TRAJECTORY PREDICTION USING GAUSSIAN PROCESS

Recently, machine learning methods such as hidden Markov models and random forests have been used for trajectory

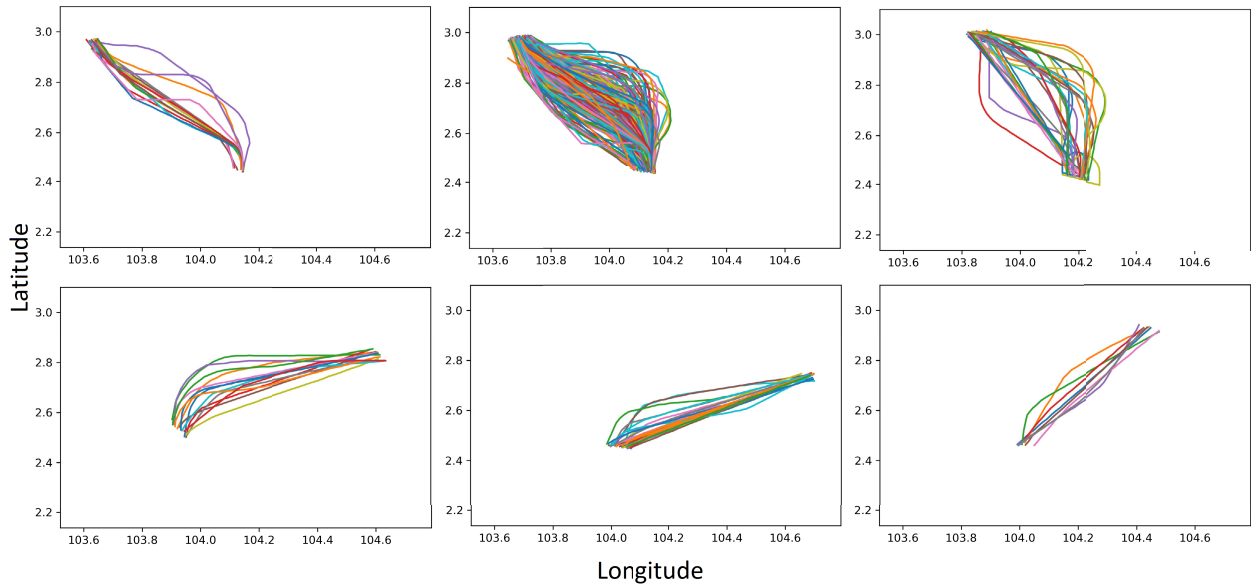


FIGURE 2. Examples of grouping trajectories based on entry and exit points.

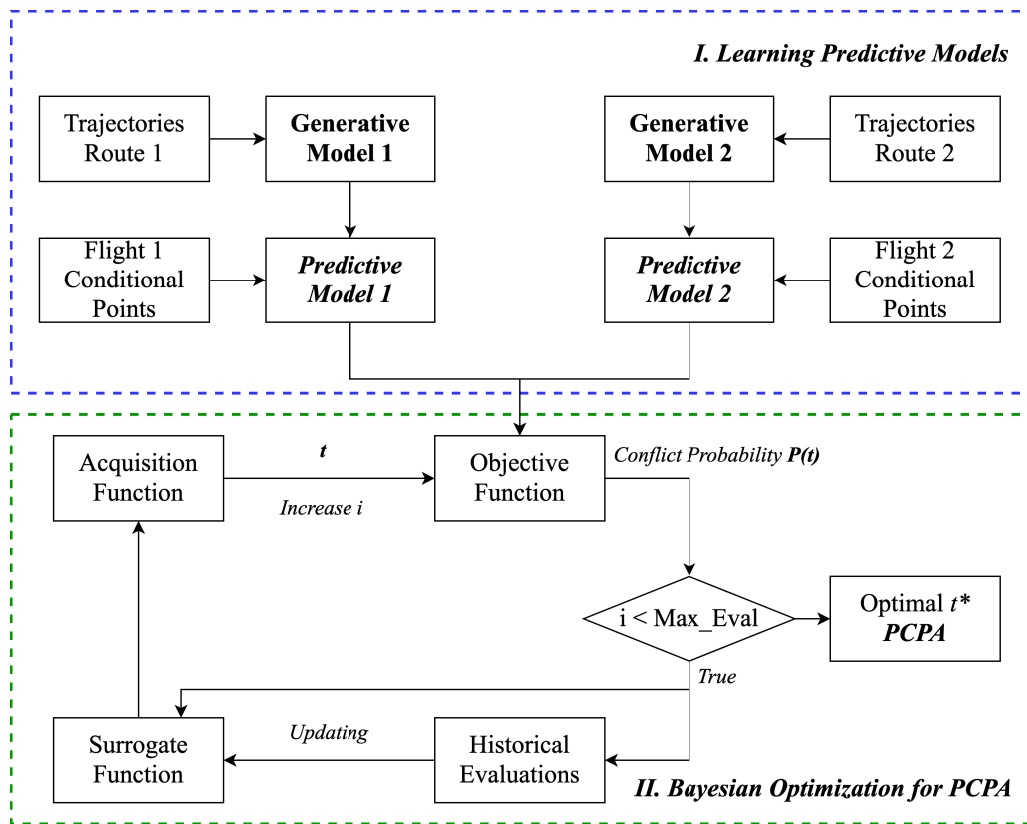


FIGURE 3. Illustrating approach for probabilistic conflict detection using Bayesian optimization.

prediction. However, when considering a predictive model with uncertainty, the Gaussian process is a promising approach because it can effectively model the dispersion of a trajectory. The Gaussian process is a collection of random variables, any finite number of which have (consistent) joint Gaussian distributions [24]. A Gaussian process

$f \sim \mathcal{G}(m(t), k(t, t'))$ is fully specified by its mean function $m(t)$ and covariance function $k(t, t')$ which need to be symmetric and satisfy Mercer's condition. Since our work involves 3-dimensional positions of aircraft over time, we train 3 Gaussian processes with time as input and latitude, longitude, and altitude as outputs of Gaussian processes.

1) HOMOSCEDASTIC NOISE AS THE INPUT INDEPENDENT NOISE

We consider a single cluster with H trajectories. For each trajectory i we will have N_i input vectors t paired with outputs y . Thus, the training data is of the form: $D_N = \{(t_j, y_j)\}_{j=1..N}$ with $N = \sum_{i=1}^H N_i$.

Gaussian process regression is a machine learning technique for inferring likely values of y for an input t as $y(t) = f(t) + \epsilon$ where f is a Gaussian process, $\epsilon \sim N(0, \sigma^2)$ is an input noise, and σ is a positive constant.

For the generative trajectory model, we train three one-dimensional models with t as the time and y as either the longitude, latitude, or altitude of the aircraft at time t . In this manner, we can specify the noise variance for each dimension independently. However, one drawback of this approach is that it does not model the correlation among the noises of the three dimensions.

2) HETEROSCEDASTIC NOISE AS THE INPUT INDEPENDENT NOISE

Instead of inferring y for an input t as $y(t) = f(t) + \epsilon$ where f is a Gaussian process, $\epsilon \sim N(0, \sigma^2)$ is the input noise with σ is a positive constant, we consider the model:

$$\begin{aligned} y(t) &= f(t) + \epsilon(t), \\ \epsilon(t) &\sim N(0, \sigma(t)), \end{aligned}$$

This is because it can capture the different noise levels along the trajectories. Furthermore, we use kernel regression to fit the noise-level function $\sigma(t)$.

3) AIRCRAFT TRAJECTORY PREDICTION GIVEN CONDITIONS POINTS

From the generative model, the mean function m^* and kernel k^* are obtained and used to sample K next locations of the aircraft $\hat{D}_K = (t_i, \hat{y}_i)_{i=1}^K = (T^K, \hat{y}^K)$ after observing its M previous locations $D_M = (t_i, y_i)_{i=1}^M = (T^M, y^M)$. This could be done by using the posterior distribution:

$$\begin{aligned} \hat{y}^K | T^K, T^M, y^M &\sim \mathcal{N}(k^*(T^K, T^M)(k^*(T^M, T^M) \\ &+ \sigma(T^K)I)^{-1}y^M, k^*(T^K, T^K) + \sigma(T^K)I \\ &- k^*(T^K, T^M)(k^*(T^M, T^M) + \sigma(T^K)I)^{-1}k^*(T^K, T^M)). \end{aligned} \quad (1)$$

Equation (1) can also be used to sample trajectories with given conditional points by setting $D_M = (T^M, y^M)$ as the set of conditional points. A conditional point is either the current position of the aircraft or reference points such as waypoints in the flight route. Instead of providing the model with the current positions only, we consider scenarios where flights have information on the reference routes from flight plans or from air traffic controllers. This setting can be used in conflict resolution where each maneuver can be considered as a reference flight route or set of conditional points and input into our model.

B. EXPERIMENT 1

This experiment is designed to evaluate the ability of the proposed probabilistic trajectory prediction model to capture the uncertainty from data using the heteroscedastic Gaussian process. The historical data will be utilized for training and evaluating the proposed approach. Its performance will be compared with a homoscedastic GP approach, adapted from [21], for accessing the advantages of our approach in modeling the heteroscedastic noise in real trajectory data.

The training process of the probabilistic trajectory prediction model is designed and implemented following Algorithm 1. The first part of the algorithm involves training the Gaussian processes to model the trajectories in each flight group with corresponding uncertainties. This step trains one Gaussian process for each data dimension (longitude, latitude, and altitude). In the second part, the prediction Gaussian process models are trained using the trained kernel of previous models as initial kernels and conditional points as input. The algorithm requires the following inputs for predictions: processed trajectories for each flight route (grouped trajectories), minimum number of trajectories (N_T), maximum number of data points for a subset of trajectories (NP_{MAX}), number of conditional points (N_{CP}) and initial radius value of each dimension for filtering trajectories with conditional points (Δ). Here, N_T is used to guarantee sufficient input data for training. However, due to the limitation of computational power, NP_{MAX} can be adjusted to generate a smaller subset of data for training the three Gaussian processes (high complexity $O(n^3)$). Furthermore, Δ is crucial for building training data for the predictive model. It is multiplied with normalized learned variance (*Normalized_V*) to compute the filtering region (within radius *Radius_Filtered*) for each conditional point. Only trajectories that pass through all filtering regions of conditional points remain in the new data set, which is then used to train predictive models. These input parameters can be modified depending on the particular application. In this experiment, several trials have been conducted for the appropriate parameters, and as the results, the selected parameters are [$N_T = 10, NP_{MAX} = 500, N_{CP} = 4, \Delta = [1NM, 1NM, 0.5FL]$]. We use *GaussianProcessRegression* from *SKlearn* for learning and in the case of heteroscedastic noise, a heteroscedastic kernel, adopted from [25], is used.

The historical trajectory data, described in Section II, are used with the train/test ratio of 70/30 for each flight group. The training data is used to train the generative models which capture the uncertainty of trajectories within each group. Noting that the selected training data of each group is sufficient for training purposes with more than $N_T = 10$ trajectories and $NP_{MAX} = 500$ data points. For the testing data, the conditional points are obtained by extracting the turning points from each trajectory using the Douglas-Peucker algorithm [26]. A set of conditional points for each trajectory will include the start and end locations and the set of extracted turning points. The size of those sets may vary between 3 to 5 in our dataset, depending on

Algorithm 1 Algorithm for Prediction Model

```

Input:  $Traj_{GID}, N_T, NP_{MAX}, N_{CP}, \Delta$ 
Result:  $GP_{Tf}$ 
1 if  $Traj_{GID} \geq N_T$  then
2    $Traj_{samples} \subseteq Traj_{GID}$  and  $|Traj_{samples}| \leq NP_{MAX}$ 
3    $GP_{Ts} = \text{Train\_Gaussian\_Process}(Traj_{samples})$ 
4    $CP(X, Y, Z, Time) =$ 
     Initializing_conditional_points( $N_{CP}$ )
5    $V = \text{Estimating\_Variance}(GP_{Ts}, CP.Time)$ 
6    $Normalized\_V = V / \min(V)$ 
7    $Radius_{Filtered} = (Normalized\_V[i] * \Delta[i])_{i=1}^{N_{CP}}$ 
8    $Traj_{Filtered} =$ 
     Filter_Conditional_Trajectories( $Traj_{GID}, CP,$ 
      $Radius_{Filtered}$ )
9    $GP_{Tf} = \text{Train\_Gaussian\_Process}(Traj_{Filtered},$ 
     kernels =  $GP_{Ts}.kernels$ )
10 end

```

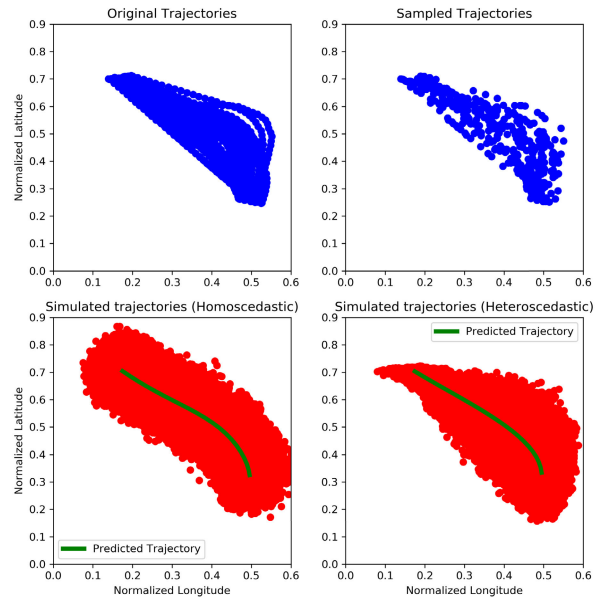


FIGURE 4. Learning ADS-B trajectory variation.

the pattern of each trajectory. Then they are utilized as input for Algorithm 1 to obtain the heteroscedastic GP models for probabilistic trajectory prediction.

For comparison, a baseline probabilistic trajectory prediction model using classical homoscedastic GP adapted from [21] is implemented, trained, and tested with the same set of data. The performances of those models are accessed both qualitatively (e.g., visualizing and observing the variation of the predicted trajectories from both approaches) and quantitatively (e.g., comparing the Kullback-Leibler (KL) divergence obtained from both approaches). The Kullback-Leibler (KL) divergence, also known as relative entropy, is a measure of how one probability distribution diverges from another. It is used in this experiment to quantify the difference between the probability distributions of the prediction model and the historical data. The formula for KL divergence between two probability distributions P and Q is defined as:

$$D_{KL}(P||Q) = \sum_i P(i) \log \left(\frac{P(i)}{Q(i)} \right) \quad (2)$$

where $D_{KL}(P||Q)$ represents the KL divergence from distribution Q to distribution P ; $P(i)$ and $Q(i)$ are the probabilities of an aircraft position i occurring in distributions P and Q , respectively.

C. EXPERIMENTAL RESULTS AND DISCUSSION

Using three models to handle multi-outputs may lead to issues with correlations among different dimensions. However, Figure 4 shows our prediction result for trajectory in 2 dimensions (longitude and latitude). In which, the mean predicted trajectories (green lines) of both models are smooth and fit to the trajectory data. The results show that combining predictive values from our trained models can still capture the distribution of multi-dimensional trajectory data.

Figure 5 illustrates the results of the generative models for three dimensions (longitude, latitude, and altitude) for

one flight route. The figures in the first column present the normalized values of the trajectories in each dimension over normalized time. Based on algorithm 1, the sampling step is applied, i.e. keeping 10% of data ≈ 500 data points. This will speed up the training of the Gaussian process and is also similar to the case where the number of data points is limited for fewer traffic routes. Two columns with black scatter plots on the right-hand side show the learning results. As we can see in columns 3 and 4 of Figure 5, the Gaussian process with homoscedastic kernel results is consistent in its variance over time for the three dimensions while the one with a heteroscedastic kernel replicates the distribution of raw data very well. Additionally, we can observe a bigger dispersion in latitude dimension than the others which is also reflected in the differences between the shapes of simulated data for both types of models. The simulated latitude values of the heteroscedastic model have a very small variance at the beginning, which significantly increases over time.

The results for predictive models using the heteroscedastic model for ADS-B data are illustrated in Figure 6. The second column of figures shows the results of the filtering process. Because the filtered radius will be proportional to the variance of the group at given time stamps, the filtered trajectories can also have big dispersion by keeping the trajectory quite far from conditional points. The results for predictive models can be observed in the third column of Figure 6 are interesting. The mean predicted trajectory can be used to anticipate the flight trajectory that flows the reference route (through conditional points) since it fits well with the shape of the trajectories in the data. Another interesting observation is the learned variance or uncertainty of the model. The sampling data from the model has shown its ability to capture not only the dispersion of filtered trajectories but also the full data. Moreover, the model with a heteroscedastic kernel

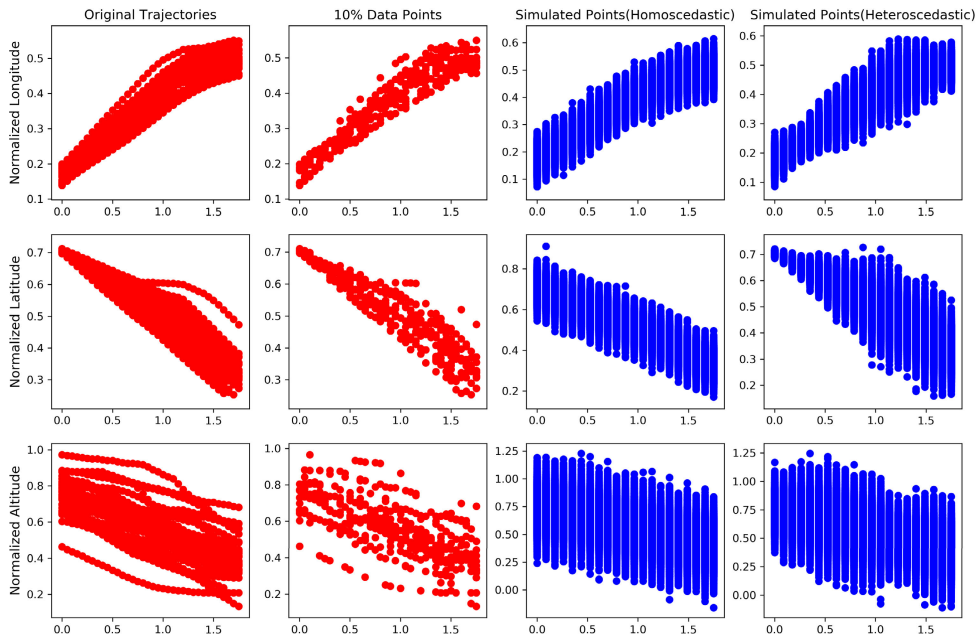


FIGURE 5. Example of learning ADS-B trajectory’s variances.

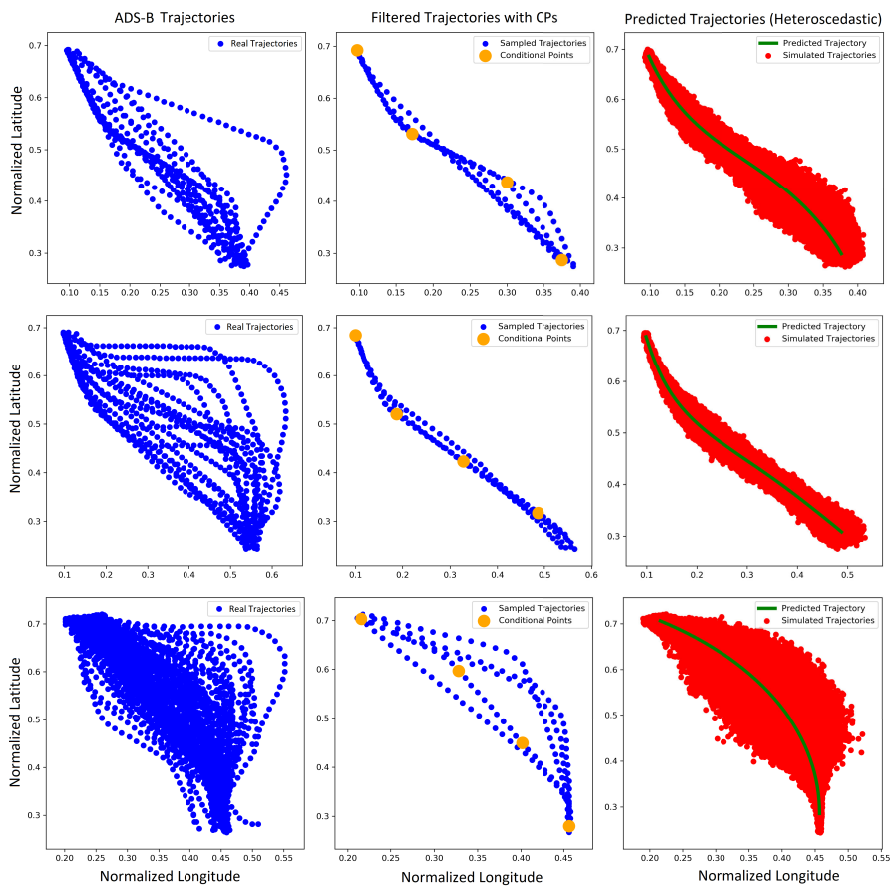


FIGURE 6. Prediction results from predictive model with ADS-B data.

provides good results in capturing the uncertainty from trajectory data as compared to the homoscedastic model. The comparison of two types of models using Kullback-Leibler

(KL) divergence is represented in Figure 7. From the left to the right, we can observe the results for different dimensions (longitude, latitude, and altitude). The heteroscedastic model

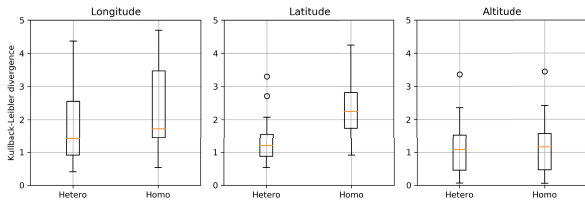


FIGURE 7. Comparing Kullback-Leibler divergences of heteroscedastic and homoscedastic models.

achieves robust performance with KL divergence around 1 to 1.3 for all dimensions. The performance of the two altitude predictive models is similar with the difference $\approx 4\%$, even though the median value of the heteroscedastic model is slightly smaller. A possible explanation is that the variance of altitude dimension in raw data is consistent over time which eliminates the benefits of our proposed approach. Our proposed approach outperforms the classical Gaussian process in modeling both longitude ($> 24\%$ reduction) and latitude ($> 45\%$ reduction), especially since there is a significant improvement in the performance of the model for latitude. We can conclude that heteroscedastic modeling can significantly better capture model trajectory data than homoscedastic modeling.

V. PROBABILISTIC CONFLICT DETECTION USING BAYESIAN OPTIMIZATION

A. BAYESIAN OPTIMIZATION

Bayesian Optimization is a class of machine-learning-based optimization methods that focuses on solving the problem

$$\max_{t \in A} F(t),$$

where A is the feasible set and F is a “black-box” objective function or is costly to be evaluated.

In contrast to GS, the BayesOpt algorithm keeps track of the past evaluation results to update a probabilistic model which maps the given parameters to the probability of a score s from objective function $F(t)$. The probabilistic model $P(s|t)$ is called a “surrogate” function with an input parameter, t , and a predicted score s . Bayesian optimization consists of five main steps:

- 1) Building surrogate probability model of objective function: The surrogate function is the probability representation of the objective function. There are several different forms of the surrogate function including Gaussian processes, Random Forest regression, and Tree-structured Parzen Estimator [27] which is used in this study. Instead of directly representing $P(s|t)$, the Tree-structured Parzen Estimator (TPE) applies the Bayes rule to get surrogate probability $P(s|t)$:

$$P(s|t) = \frac{P(t|s)P(s)}{P(t)}$$

where $P(t|s)$ is the probability of the parameter t given the score s of the objective function and is defined as:

$$P(t|s) = \begin{cases} l(t) & \text{if } s < s^* \\ g(t) & \text{if } s \geq s^* \end{cases}$$

Here, s^* is the threshold value of the objective function. $l(t)$ and $g(t)$ are two Gaussian Mixture Models (GMM) in which $l(t)$ is fitted using the data points in which their corresponding objective values are less than s^* and $g(t)$ is fitted using the remaining data points.

- 2) From the surrogate function, finding the next parameter t to be evaluated using Acquisition Function: Given the surrogate function, the acquisition function $A_{EI}(t)$ is the criteria to find the next parameter. This study uses the most common choice of this criteria, which is the expected improvement:

$$A_{EI}(t) = EI_{s^*}(t) = \int_{-\infty}^{s^*} s^*(s^* - s)P(s|t)ds$$

In this equation, t is the proposed parameter, and s is the actual value of the objective function using parameter t . The aim of this method is to maximize the Expected Improvement with respect to parameter t or find the best parameters t under the surrogate function $P(s|t)$. If $EI_{s^*}(t) > 0$, the parameter t is expected to yield a better result than the threshold value.

- 3) Apply the new parameter to the original objective function.
- 4) Update the surrogate model incorporating the new result.
- 5) Repeat steps 2-4 until maximum iterations or time is reached.

B. PROBABILISTIC CLOSEST POINT OF APPROACH - PCPA

We model the uncertainty in two trajectories by two sets of Gaussian processes f^1 and f^2 which are fitted to data, with the format $f(t) = (GP_{longitude}(t), GP_{latitude}(t), GP_{altitude}(t))$. Our purpose is to find the time in which two aircraft have the highest probability of conflict:

$$\bar{t} = \operatorname{argmax}_{t \in T} P(t), \tag{3}$$

where $P(t) = \mathbb{P}(\|f_{1,2}^1(t) - f_{1,2}^2(t)\| \leq s_h, |f_3^1(t) - f_3^2(t)| \leq s_v)$, with $f_{1,2}(t) = (GP_{longitude}(t), GP_{latitude}(t))$, and $f_3(t) = GP_{altitude}(t)$. In this work, we set $s_h = 5NM$ and $s_v = 10FL$.

Taking the advantage of Gaussian process, we can approximately compute $P(t)$ by Monte Carlo simulation:

$$P(t) \approx \frac{1}{N_s} \sum_{i=1}^{N_s} I_{A_i}(t),$$

where N_s is the number of MC runs, $A_i(t)$ is the event $\{\|f_{1,2}^{1,i}(t) - f_{1,2}^{2,i}(t)\| < s_h, |f_3^{1,i}(t) - f_3^{2,i}(t)| < s_v\}$, $\{f_{1,2}^{1,i}(t)\}_{i=1,\dots,N_s} \stackrel{\text{i.i.d}}{\sim} \mathbb{P}_{f_{1,2}^1}(t)$, $\{f_{1,2}^{2,i}(t)\}_{i=1,\dots,N_s} \stackrel{\text{i.i.d}}{\sim} \mathbb{P}_{f_{1,2}^2}(t)$ and $\{f_3^{1,i}(t)\}_{i=1,\dots,N_s} \stackrel{\text{i.i.d}}{\sim} \mathbb{P}_{f_3^1}(t)$, $\{f_3^{2,i}(t)\}_{i=1,\dots,N_s} \stackrel{\text{i.i.d}}{\sim} \mathbb{P}_{f_3^2}(t)$.

As discussed air traffic conflicts are rare events. As a result, estimating the conflict probability at each time t requires a large number of simulation runs N_s . We also need to evaluate the maximum value of $P(t)$ with respect to t (refer equation (3)). Therefore, the computational load is very high. To overcome this situation, we apply the BayesOpt technique

by choosing Tree Parzen Estimators (TPE) for the surrogate probability model and using Expected Improvement for the selection function.

C. EXPERIMENT 2

This experiment's main purpose is to evaluate BO's advantages for quickly estimating accurate conflict probability between a pair of aircraft. Algorithm 2 describes the conflict probability estimations of two flights (with uncertainty) using Bayesian optimization (BayesOpt). At a given timestamp t , we perform the sampling with size N_s for both flights: $[P_{1..N_s}^1]$, $[P_{1..N_s}^2]$. Then, for each pair, the distance between two aircraft $Dist(P_k^1, P_k^2)$ is compared with the minimum separation allowed (constraints). The conflict probability is computed directly from the number of pairs which violate the constraints over N_s . The output of these algorithms is the maximum conflict probability ($CProb$), the time to probabilistic closest point of approach ($Time - to - PCPA$), and the computational cost ($Cost$). The *hyperopt* Python package is used for the BayesOpt algorithm. Since we want to avoid any assumption about the search space for t , a uniform distribution is used to sample t . It is worth noting that the used package tries to minimize our defined objective function while our objective is to maximize conflict probability. Thus, the returned probability is multiplied by a negative one in the objective function. In BayesOpt, there are two terminating conditions: the convergence of maximum objective values or maximum number of evaluations (max_evals) is reached. In this study, we use max_evals as the terminating criterion.

Algorithm 2 Bayesian Optimization Method

Input: N, GP1_2, GP2_2, Duration, Threshold
Result: CProb, Time-to-PCPA, Cost

```

1 def objective(t):
2   Traj_1 = GP1_2.sample_y(t, N2)
3   Traj_2 = GP2_2.sample_y(t, N2)
4   Dist = Distance(Traj_1, Traj_2)
5   Prob_t = Conflict_Prob(Dist, Threshold)
6   return -Prob_t
7 end
8 space = hp.uniform('t', 0, Duration)
9 algo = tpe.suggest #Tree Parzen Estimators
10 trials = Trials()
11 Start_time = time.time()
12 Time-to-PCPA = hp.fmin(objective, space, algo,
   trials, max_evals)
13 CProb = get_max_prob(trials, Time-to-PCPA)
14 Cost = time.time() - Start_time

```

Algorithm 3, which is based on Grid-Search (GS) is developed as the baseline for comparison. The algorithm has steps similar to Algorithm 2, but the difference is how they find the value of t for evaluation. GS algorithm evaluates all values of t in a given range. Since the time interval (R) is given, only a predefined set of values t is evaluated. This

approach is mentioned and commonly used for probabilistic conflict detection, especially when finding the maximum conflict probability [12], [13], [14], [15], [16]. On the other hand, BayesOpt determines a sequence of values t depending on historical evaluations. This sequence of values of t can be continuous. By employing BayesOpt, the sample size is reduced, and consequently, lower computational cost is required.

Algorithm 3 Grid-Search Method

Input: N, R, GP1_2, GP2_2, Duration, Threshold
Result: CProb, Time-to-PCPA, Cost

```

1 Start_time = time.time()
2 Prob_T = []
3 for t ← 0 to Duration by R do
4   Traj_1 = GP1_2.sample_y(t, Ns)
5   Traj_2 = GP2_2.sample_y(t, Ns)
6   Dist = Distance(Traj_1, Traj_2)
7   Prob_t = Conflict_Prob(Dist, Threshold)
8   Prob_T.append(Prob_t)
9 end
10 CProb = max(Prob_T)
11 Time-to-PCPA = argmax(Prob_T)
12 Cost = time.time() - Start_time

```

For accessing the performance of those approaches, a scenario of two aircraft, illustrated in Figure 8, is generated by randomly sampling a pair of trajectories from two different groups of trajectories from historical data, mentioned in Section II. The first group, called group A, is the group of north-south flights that start near waypoint UKASAUKASA, passing by waypoint PEKLA and heading to waypoint PIBAP to exit the sector. The second group, called group B, is the group of east-west flights that start from waypoint LENDA, passing by waypoint PEKLA and heading to waypoint VMR to exit the sector. The travel time of flights in each group varies between 12 to 16 minutes. The starting time of the east-west flight is modified by adding an appropriate delay to ensure a certain level of conflict probability. In this case, the time differences at the crossing points of those two trajectories are limited to 1 minute. This is a crossing scenario where the conflict happens because of the uncertainty in the position of the aircraft at the given timestamp t . The learning process is applied and at the end, we can obtain two predictive models for two routes (called A, and B). These models are then used for evaluating BayesOpt as a probabilistic conflict detection algorithm.

Detecting aircraft conflict is a safety-critical task where the accuracy of the algorithm may have a strong impact on the decision of ATCO. In this work, a large number of MC simulations (e.g., sampling size N_s from $[0.5M, 1M, 2M]$) will be used to accurately estimate the potential conflict between any pairs of aircraft under high trajectory uncertainty. Besides, 5 different values of sampling interval R ($[1s, 5s, 10s, 30s, 60s]$) of the Grid-Search algorithm are

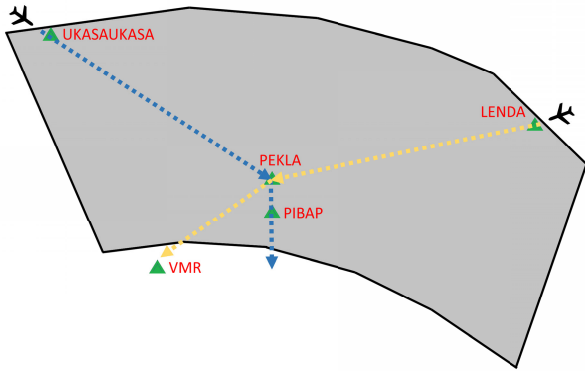


FIGURE 8. Illustration of the two nominal routes (referred to as route A in black and route B in yellow) for generating the scenarios for experiment 2. Route A represents a group of north-south flights, called group A, that pass by waypoints UKASAUKASA, PEKLA and PIBAP. Route B represents a group of east-west flights, called group B, that pass by waypoints LENDA, PEKLA, and VMR. Groups A and B are groups of flights from historical data.

selected by considering the common trajectory sampling rates which are usually used in trajectory prediction and conflict detection. Therefore, the GS simulation method is evaluated with 15 different configurations (3 different values of N_s and 5 different values of R). BayesOpt’s performance is experimented with and reported for 3 values of terminating condition, $Max_Eval \in [50, 100, 200]$. These parameter values are increased by a factor of 2 to perform parameter tuning and the following experimental results have shown their appropriateness for investigation of our approach’s performance. For the convenience of discussion, all experimental models are named with a similar format {Method}{Parameter}, e.g., GS01, GS05, or BO100. To distinguish models from the same methods for different sampling sizes, a corresponding prefix will be added to the name, e.g., 0.5M - GS01 vs. 2M - GS01. In this study, each model is evaluated 100 times to obtain the average model’s performance.

The computational cost and final results for P-CPA (e.g., Time to P-CPA and maximum P-CPA) will be reported and compared between different settings. Because the exhaustive search has been performed, the results of the GS with sampling interval 1s and sampling size 2M are chosen as the baseline for comparison in terms of the Time to P-CPA and Maximum P-CPA. The difference (e.g., percentage error PE) in the conflict probability (P-PCA) between the other models’ estimation $V_{Predicted}$ and the baseline $V_{Baseline}$ will be extracted and discussed to highlight the advantages and disadvantages of the proposed BO approach.

$$PE = \frac{|V_{Predicted} - V_{Baseline}|}{|V_{Baseline}|} \times 100\%$$

D. RESULT

The experimental is conducted as described and the average T-PCPA, P-CPA and computational cost are captured for discussion. First of all, all models can achieve the average T-PCPA values which approximate the T-PCPA at 2M-GS01.

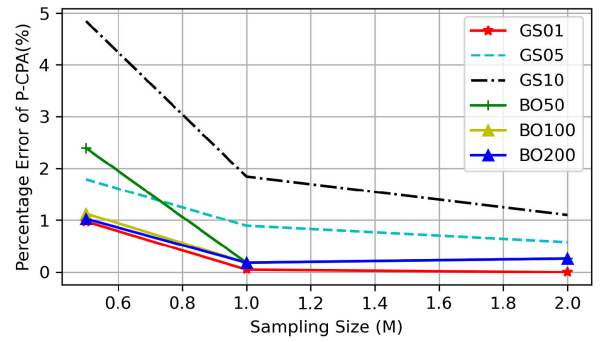


FIGURE 9. Illustration of the convergence of the P-CPA estimation with the increases in sampling size of Monte Carlo simulation (from 0.5M to 2M). The PE values of GS30 and GS60 are excluded from the figure (e.g., PEs $\geq 13.5\%$ and 29.5% , respectively).

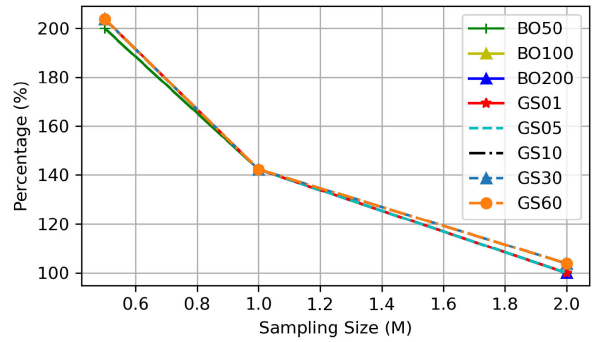


FIGURE 10. Illustration of the reduction in standard deviation (std) of the model’s estimation with the increases in sampling size of Monte Carlo simulation. The baseline (e.g., 100%) is the std of the Grid-Search 1s (GS01) with 2M MC simulation.

Moreover, the GS01 shows the sign of convergence at $N_s = 1M$, in which the PE of 1M-GS01 and 2M-GS01 is just 0.05%. All BO models approximate the baseline with ($N_s \geq 1M$) with the PE $< 0.25\%$. On the contrary, all GS methods with $R \geq 5s$ have high PE and only achieve PE $> 0.65\%$ at $N_s = 2M$. Since with $R \geq 5s$, the GS models can only evaluate the conflict probability at those discrete points (as their predicted T-PCPAs) with the expected deviation (approximately $\frac{R}{2}$) from the actual T-PCPA. Thus, the fine level of the grid or the value of R has a strong impact on the accuracy of the GS models’ estimation. The PEs for GS30 and GS60 are very high, e.g., 15% and 30% respectively, even at $N_s = 2M$. Those convergence behaviors can be observed in Figure 9. Since the PE values of GS30 and GS60 are too large, they are excluded from the figure for better visibility. Furthermore, with the increase in sampling size, the uncertainties of the estimation also decrease. For comparison, the standard deviation (std) of 2M - GS01 is used as the baseline (e.g., 100%) for computing the relative percentage of the std of the other models. Figure 10 depicts the reduction in relative std or uncertainties in the estimation of the evaluated models. Expectedly, the observed reductions are proportional to the increase in the sampling size of the MC simulation. F

Since the computational cost strongly depends on the sample size of MC simulation, the results, in Figure 11,

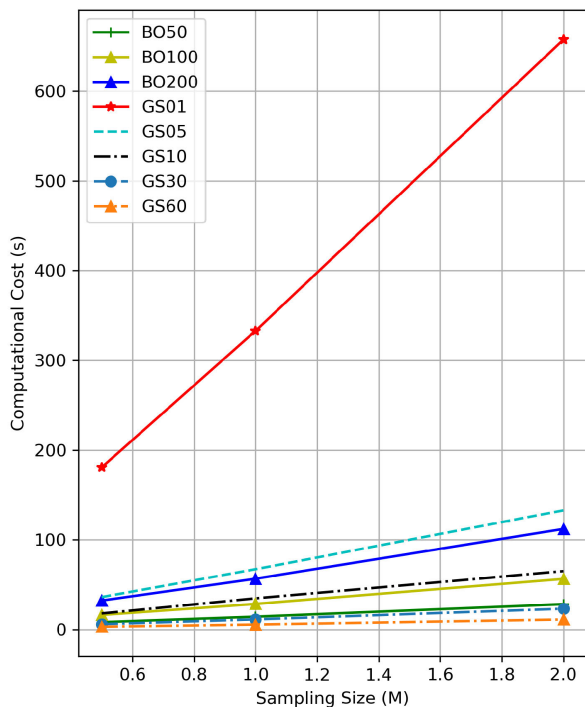


FIGURE 11. The computational cost (s) of all evaluated methods with the increases in sampling size.

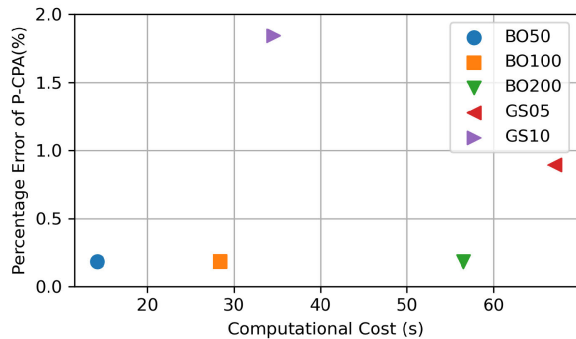


FIGURE 12. Illustration of the trade-off between the computational cost to find the maximum probabilistic and the of the P-CPA estimation with the increases in sampling size of the Monte Carlo simulation (from 0.5M to 2M). The PE values of GS30 and GS60 are off the chart and excluded from the figure (e.g., respectively 13.5% and 29.5%).

show a linear increase in terms of computational costs for all algorithms when the number of samples (N_s) is increased. When R increases, there are more values of t that should be evaluated, and hence, the cost of GS models also increases. Therefore, even though GS01 can achieve high accuracy in estimating P-CPA, its computational cost is much higher than the rest of the models. It needs more than 330 seconds to estimate the conflict probability with $N_s = 1M$ while the others just need less than 120 seconds. For the given conflict scenarios, with 20-minute trajectories, the cost for BayesOpt with 100 evaluations (BO100) is close to that of GS10 (with 120 evaluations). Moreover, the BO50 can achieve high performance with significantly less computational costs (e.g.,

<14s seconds at $N_s = 1M$). The only models that execute faster than BO50 are GS30 and GS60 but with a significant trade-off in model performance. For the GS05 model, it still needs 4.5x the running time of BO50 and 2.3x the running time of BO100 with a 2.6x increase in PE. Those trade-off between PE and computational costs, at $N_s = 1M$, is depicted in Figure 12. Since GS01 has a too-large computational cost (>320s), GS30 and GS60 have too-large PE (>13.5% and >29.5%), they are omitted from the figure for better visibility. As observed from the figure, BO50 and BO100 outperform the other models in both speed and accuracy for estimating the probability of conflict between a pair of aircraft. In short, the 1M - BO50 model can be considered as the “best” model with PE < 0.25% and computational cost < 14s.

VI. CONCLUSION

The results demonstrate the potential of our approach for trajectory prediction and conflict detection. Heteroscedastic Gaussian process regression can learn the uncertainty from data and perform better aircraft position prediction. Although some recent approaches try to increase the confidence of prediction by using more data and advanced machine learning techniques, the uncertainty from data still exists because of various reasons including human intervention and incomplete information. Our approach for predictive modeling can successfully capture heteroscedastic noises in trajectory data with robust performance. It achieves KL divergence around 1 to 1.3 for all dimensions which reduces by > 45% for latitude, > 24% for longitude, and 4% for altitude compared to the classical homoscedastic GP approach. Moreover, the results in experiment 2 have highlighted the benefits of Bayesian Optimization over the classical Grid-Search method in terms of speed and accuracy. The algorithm can work with continuous time, which means it can locate the conflict position without the impact of the selection of the sampling interval. The proposed approach can achieve less than 0.25% percentage error compared to the baseline with only 1/23 of the computational cost. Especially, when computing the conflict probability is expensive, the contribution of this method is more significant.

For future work, we believe that improving the multi-output problem could enhance the performance of the predictive model and speed up the training. Besides, the BayesOpt approach for probabilistic conflict detection can be generalized to work for the whole sector with multiple flights and conflicts.

ACKNOWLEDGMENT

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of National Research Foundation, Singapore, and the Civil Aviation Authority of Singapore.

REFERENCES

[1] M. Nolan, *Fundamentals of Air Traffic Control*. Boston, MA, USA: Cengage, 2010.

- [2] T. Lauderdale, "Probabilistic conflict detection for robust detection and resolution," in *Proc. 12th AIAA Aviation Technol., Integr., Operations (ATIO) Conf. 14th AIAA/ISSMO Multidisciplinary Anal. Optim. Conf.*, Sep. 2012, p. 5643.
- [3] R. A. Paielli and H. Erzberger, "Conflict probability estimation for free flight," *J. Guid., Control, Dyn.*, vol. 20, no. 3, pp. 588–596, May 1997.
- [4] M. Prandini, J. Hu, J. Lygeros, and S. Sastry, "A probabilistic approach to aircraft conflict detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 4, pp. 199–220, Dec. 2000.
- [5] J. Hu, M. Prandini, and S. Sastry, "Aircraft conflict prediction in the presence of a spatially correlated wind field," *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 3, pp. 326–340, Sep. 2005.
- [6] E. Hernández-Romero, A. Valenzuela, and D. Rivas, "Probabilistic multi-aircraft conflict detection and resolution considering wind forecast uncertainty," *Aerosp. Sci. Technol.*, vol. 105, Oct. 2020, Art. no. 105973.
- [7] Y. Matsuno and T. Tsuchiya, "Probabilistic conflict detection in the presence of uncertainty," in *Air Traffic Management and Systems*. Berlin, Germany: Springer, 2014, pp. 17–33.
- [8] S. Hao, Y. Zhang, S. Cheng, R. Liu, and Z. Xing, "Probabilistic multi-aircraft conflict detection approach for trajectory-based operation," *Transp. Res. C, Emerg. Technol.*, vol. 95, pp. 698–712, Oct. 2018.
- [9] H. A. Blom, J. Krystul, G. B. Bakker, M. B. Klompstra, and B. K. Obbink, "Free flight collision risk estimation by sequential MC simulation," in *Stochastic Hybrid Systems*. Boca Raton, FL, USA: CRC Press, 2006, pp. 254–286.
- [10] L. Yang, J. H. Yang, J. Kuchar, and E. Feron, "A real-time Monte Carlo implementation for computing probability of conflict," in *Proc. AIAA Guid., Navigat., Control Conf. Exhib.*, 2004, p. 4876.
- [11] S. Seyedipour, H. Nobahari, and M. Prandini, "A probabilistic approach to mid-term conflict detection with accuracy estimate," *J. Franklin Inst.*, vol. 359, no. 16, pp. 9193–9219, Nov. 2022.
- [12] C. Allignol, N. Barnier, N. Durand, and J.-M. Alliot, "A new framework for solving en route conflicts," *Air Traffic Control Quart.*, vol. 21, no. 3, pp. 233–253, Jul. 2013.
- [13] G. Chaloulos and J. Lygeros, "Effect of wind correlation on aircraft conflict probability," *J. Guid., Control, Dyn.*, vol. 30, no. 6, pp. 1742–1752, Nov. 2007.
- [14] W. Liu and I. Hwang, "Probabilistic trajectory prediction and conflict detection for air traffic control," *J. Guid., Control, Dyn.*, vol. 34, no. 6, pp. 1779–1789, Nov. 2011.
- [15] C. Allignol, N. Barnier, N. Durand, A. Gondran, and R. Wang, "Large scale 3D en-route conflict resolution," in *Proc. ATM Seminar, 12th USA/Europe Air Traffic Manag. RD Seminar*, 2017.
- [16] J. de la Mota, M. Cerezo-Magaña, A. Olivares, and E. Staffetti, "Data-driven probabilistic methodology for aircraft conflict detection under wind uncertainty," *IEEE Trans. Aerosp. Electron. Syst.*, early access, May 1, 2023, doi: 10.1109/TAES.2023.3250204.
- [17] C. Di Ciccio, H. van der Aa, C. Cabanillas, J. Mendling, and J. Prescher, "Detecting flight trajectory anomalies and predicting diversions in freight transportation," *Decis. Support Syst.*, vol. 88, pp. 1–17, Aug. 2016.
- [18] S. Ayhan and H. Samet, "Aircraft trajectory prediction made easy with predictive analytics," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 21–30.
- [19] R. Alligier, D. Gianazza, and N. Durand, "Predicting aircraft descent length with machine learning," in *Proc. Int. Conf. Res. Air Transp.*, 2016, pp. 1–9.
- [20] W. J. Eerland, S. Box, and A. Sóbester, "Modeling the dispersion of aircraft trajectories using Gaussian processes," *J. Guid., Control, Dyn.*, vol. 39, no. 12, pp. 2661–2672, Dec. 2016.
- [21] D.-T. Pham, M.-M. Ngo, N. P. Tran, S. Alam, and V. Duong, "Conflict prediction using generative model from ADS-B data," in *Proc. Integr. Commun., Navigat. Surveill. Conf. (ICNS)*, Apr. 2019, pp. 1–9.
- [22] Q. V. Le, A. J. Smola, and S. Canu, "Heteroscedastic Gaussian process regression," in *Proc. 22nd Int. Conf. Mach. Learn.*, 2005, pp. 489–496.
- [23] D.-T. Pham, "Machine learning-based flight trajectories prediction and air traffic conflict resolution advisory," Ph.D. dissertation, Univ. Paris Sci. Et Lettres, 2019.
- [24] E. Snelson, "Tutorial: Gaussian process models for machine learning," Gatsby Comput. Neurosci. Unit, UCL, Tech. Rep., 2006.
- [25] M. Jan Hendrik. (2016). *GP_Extras*. [Online]. Available: https://github.com/jmetzen/gp_extras
- [26] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica, Int. J. Geographic Inf. Geovisualization*, vol. 10, no. 2, pp. 112–122, Dec. 1973.
- [27] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyperparameter optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 2546–2554.



DUC-THINH PHAM received the Ph.D. degree in Paris science and letter (PSL) and in computer science, France, in 2019. He is currently a Senior Research Fellow with the Air Traffic Management Research Institute (ATMRI), Nanyang Technological University (NTU), where he is also the Leader of the Data-Driven Surface Movement Management Project with ATMRI. His research interests include deep reinforcement learning and application of machine learning in air traffic management.



YASH GULERIA received the B.S. degree in mechanical engineering from NIT Hamirpur, India, and the joint M.S. degree in aerospace engineering from Nanyang Technological University and Technical University of Munich. He is currently pursuing the Ph.D. degree with the Air Traffic Management Research Institute, Nanyang Technological University, Singapore. His research interests include delay propagation analysis, conflict detection and resolution and airside operation in air traffic management, and deep reinforcement learning in air transportation.



SAMEER ALAM received the M.S. degree in computer science from the Birla Institute of Technology, Mesra, India, in 1999, and the Ph.D. degree in computer science from The University of New South Wales, Canberra, Australia, in 2008. He is currently an Associate Professor with the School of Mechanical and Aerospace Engineering, Nanyang Technological University. His research interests include simulation, modeling, risk assessment, and the optimization of advanced air traffic management concepts. He is a Founding Member of the IEEE Computational Intelligence Society Chapter in Canberra. He is also a Chief Investigator of the ICAOMIDRMA Collision Risk Project and served as a Co-Chief Investigator for several projects of Air Service Australia and EUROCONTROL. He was a recipient of the Australian National University Science Medal, in 2011, the Tall Poppy Science Award, in 2011, and the Fresh Science Award, in 2009.



VU DUONG received the Ph.D. degree in artificial intelligence from Ecole Nationale des Ponts et Chaussées, France, in 1990. He is currently a Professor and the Director of the School of Mechanical & Aerospace Engineering, Air Traffic Management Research Institute (ATMRI), Nanyang Technological University, Singapore. He had also been the Head of Innovative Research and then a Senior Scientific Advisor with EUROCONTROL (1995–2012) and a member of the SESAR JU Scientific Committee (2010–2012).

• • •