Journal of Marine Science and Technology, Vol. 27, No. 3, pp. 246-256 (2019)
DOI: 10.6119/JMST.201906_27(3).0007

APPROACH FOR 4-D TRAJECTORY MANAGEMENT BASED ON HMM AND TRAJECTORY SIMILARITY

Yi Lin[1, 2, 3], Bo Yang[1, 2], Jianwei Zhang[1, 2], and Hong Liu[1, 2]

Key words: four-dimensional trajectory management, historical trajectory set, probabilistic statistical model, machine learning, hidden markov model, trajectory similarity.

ABSTRACT

To improve the accuracy and stability of flight trajectory prediction, a novel four-dimensional (4-D) trajectory management approach is proposed in this paper, which consists of the estimation and updating procedure. Historical flight trajectories are proved to be safe and feasible based on the real-time traffic situation, and serve as the data foundation of 4-D trajectory management in this paper. To achieve the goal of 4-D trajectory management, we firstly apply probabilistic statistical models and machine learning approach to predict the fly-over time and altitude of waypoints along the planning route before the flight takes off. Hidden Markov Models (HMMs) are regarded as the probabilistic model to represent the position and altitude transition patterns of the aircraft during the flight operation. The EM algorithm is applied to optimize model parameters of HMMs to fit the training data (historical trajectory set). Then the models with optimized parameters are used to predict the pre-takeoff 4-D trajectory by inferring an optimal hidden state sequence. Finally, after the flight takes off, we propose an algorithm to correct the pre-takeoff prediction results by considering the trajectory similarity between collected path of current execution and its historical trajectories. Simulations with real data show that the prediction results (fly-over time and altitude) of our proposed algorithm are more accurate than that of other existing methods, and would tend to be more credible after correcting with the proposed algorithm. Moreover, the prediction errors of our approach are stable during the whole flight, which is the bottleneck of existing d\eterministic models.

Paper submitted 02/10/17; revised 02/09/18; accepted 05/15/19. Author for correspondence: Hong Liu (e-mail: liuhblue@126.com).
[1] *National Key Laboratory of Fundamental Science on Synthetic Vision, Sichuan University, Chengdu, Sichuan, China.*
[2] *National Key Laboratory of Air Traffic Control Automatization System Technology, Sichuan University, Chengdu, Sichuan, China.*
[3] *Department of Civil and Environmental Engineering, University of Wisconsin-Madison, Madison, WI, USA.*

I. INTRODUCTION

4-D trajectory management is the foundation of Air Traffic Management (ATM) techniques, such as traffic flow forecasting, flight conflict detection, yaw alarm, correlation of flight plan and track path, et al. (Ayhan et al., 2016). An accurate 4-D trajectory is of great importance to predict the traffic situation of given airspaces, which can further ensure the operational safety of flights, maintain the order of air traffic and improve the traffic capacity. Different executions of a certain flight usually fly along a same planning route, but the fly-over time and altitude of each waypoint may be varied according to the real-time traffic situation. It is the fly-over time and altitude of waypoints that is very significant to predict the traffic situation in the local control area at given instants. Therefore, the basic purpose of the 4-D trajectory prediction for a flight in this paper is to estimate the fly-over time and altitude of waypoints along its planning route. In the research of air traffic control, the flight management generally can be divided into three steps: (a) Before a flight takes off, the fly-over time and altitude of waypoints along the planning route (pre-takeoff 4-D trajectory) are predicted to estimate the traffic situation strategically. (b) Once the flight takes off, collected tracks and its flight plan are correlated based on the flight information and the recognized track attributions. Then yaw alarm of certain flights, and potential conflict of aircraft pairs can be detected to ensure the flight safety based on the predicted trajectory. (c) After finishing the correlation of the track and flight plan, a series of track positions are collected by radar or ADS-B (Automatic Dependent Surveillance-Broadcast) for flights, and the pre-takeoff 4-D trajectory should be corrected to improve the efficiency of air traffic operation. From above descriptions, we can see that an accurate 4-D trajectory prediction plays a vital role in the air traffic control (ATC), which is also the reason why the issue has always been a hot research topic.

There are many existing works in this research field. Chen (Chen, 2012) divided the whole flight process into different stages (classic climb, cruise and descent, etc.) based on the flight profiles. By constructing and solving the kinematics and dynamics equations at every stage, the 4-D flight trajectory is estimated with preset conditions including aircraft parameters and aerodynamic models. However, due to the unclear boundary between different flight stages, the prediction results have a large deviation from the truth. Zhang and his colleagues (Zhang et al., 2014) proposed an algorithm to predict the flight trajectory based on the aircraft performance parameters, such as the speed and climb or descentrate. However, the proposed method did not take the change of the real-time flight environment into consideration, which causes unexpected prediction errors. Since the mass storage of historical trajectories, which contain the real-time environmental factors, is available with the developments of hardware, the following works focused on the historical trajectory mining (Kim, 2015). Authors in (Wandelt and Sun, 2015) proposed an efficient compression algorithm for saving storage space of historical trajectories. A method based on velocity correcting was presented (Tang et al., 2015, a) for predicting taxiing path. The approach routes in the terminal area were designed by analyzing frequent patterns of historical trajectories from collected radar data (Xie and Cheng, 2015). A trajectory estimation algorithm was proposed based on mining moving parameters of aircraft from historical flight trajectories by using machine learning tools (Song, 2012). Although these methods are data-driven, the accuracy of prediction results is still limited by the level of data mining. Only a few of hidden patterns of flight trajectories are mined to accomplish given tasks in those works. A method for extracting the nominal flight profile and revising airway meteorological forecasting (Tang et al., 2015, b) was proposed by mining transition patterns of historical flight paths. Trajectory prediction approach for the general aviation aircraft (Li et al., 2015) was proposed based on the data fusion theory. Researchers (Tang et al., 2015, c) also used the clustering algorithm to obtain moving patterns in different flight period, which are further used to predict the flight trajectory. Although the historical data has been introduced in later methods, the accuracy of estimation results are also not ideal because of the deterministic models without considering the randomness of flight condition. Consequently, probabilistic-based stochastic models were introduced to illustrate the randomness of the flight operation. As to the mathematic model (Hidden Markov Model), a classic research for mining trajectory patterns was proposed based on HMMs in (Jeung et al., 2007), which is the basis of this paper. Morzy also proposed an algorithm for mining trajectory patterns (Morzy, 2007) based on the HMM to track moving objects in local areas. To improve the model applicability, researchers in (Qiao et al., 2016) improved the algorithm (Morzy, 2007) by the parameter learning of self-adaptive mechanism. By combining the HMM with Gaussian Mixture Model (GMM), researchers (Lin et al., 2017) proposed an algorithm to model the motion trend of aircraft. The flight trajectory was predicted on the basis of the learned model with details, which include not only the fly-over time and altitude of waypoints, but also the motion state (longitude, latitude, altitude, speed and so on) on every update second. Ayhan and Samet proposed a stochastic trajectory prediction approach (Ayhan and Samet, 2016) to make better decisions and advisories for ATC by modeling the weather conditions and historical trajectories. A semi-Markov switching vector auto-regressive model-based anomaly detection in aviation systems was proposed (Melnyk et al., 2016) to monitor the aviation safety by considering the mechanical, environmental, and human factors during the flight operation. A comprehensive implementation for measuring the accuracy of trajectory prediction (Paglionee and Oaks, 2007) was proposed by parsing the actual and prediction trajectory with samples and measurements, and implementation details were also described in the paper.

Basically, existing approaches for 4-D trajectory prediction can be divided into three categories: kinematics and dynamics based, regression based, and probabilistic distributed based models. The first category considers the moving rules from the kinematics and dynamics view in different flight phase. The shortage of this type of approach is the phase division and the idealized simplicity (approximated constant speed or constant acceleration) for the kinematics and dynamics models. The second and third categories apply the historical data (only the fly-over time and altitude of waypoints) to predict the 4-D trajectory, but they are more likely a data engineering which only describes the features from rough levels and neglects the hidden transition patterns of the flight trajectory. Therefore, in this paper, we present a novel 4-D trajectory prediction algorithm based on probabilistic statistical model and machine learning approach. The research area is firstly divided into gridded cells to denote the flight state. To model the flight process more accurately, we introduce the HMM to describe stochastic features of the position and altitude transition during the flight operation. Two HMMs are applied to model the transitions for the position and altitude respectively, which we call them as position model and altitude model in following sections. In the position model, the observations and hidden states correspond to gridded map and route segments of the flight plan, while in the altitude model, they are customized flight levels denoting collected altitude of the flight trajectory and standard flight levels designed by Civil Aviation Administration of China (CAAC). Except some irregular conditions, such as flight returning or landing at an alternating airport, each execution of a certain flight in historical data usually traveled along the same planning route and flied over the same waypoint sequence. The historical trajectories are proved to be safe and feasible, and environment factors along the flying route are also considered during the actual operation of local flights. Therefore, parameters of the proposed flight models can be well optimized by mining historical trajectories. Based on the learned model, more accurate estimation of the fly-over time and altitude for waypoints can be achieved, which we call them as pre-takeoff prediction results. Once the flight takes off, we use an algorithm to correct the predicted fly-over time and altitude before departure by considering the trajectory similarity with its historical paths. From the view of the historical trajectory, only the position information (longitude, latitude, altitude, speed, heading, …) can be obtained from the data, which cannot be correlated with the planning route of flight plan. From the view of the flight plan, only the planning route or sequence of waypoint information is extracted, which cannot recognize the fly-over time and altitude of the waypoints. To estimate the 4-D flight trajectory, the proper way is to combine the flight plan with the collected trajectory. Generally, the position information of aircraft can be tracked by the surveillance equipment (observations), and the fundamental problem in this work is to estimate the fly-over time and altitude of waypoints which we cannot perceive directly. We apply HMM to take them as the hidden state sequence and associate with an observation sequence which fully takes the advantages of the dual stochastic progress in HMM model. Another merit of our proposed approach over existing methods is unnecessary to divide the flight process into different stages by using probabilistic characteristics. All in all, our main contributions in this work can be summarized as follows:

(a) An HMM based flight model is proposed in this paper, in which we applied the gridded map and flight level to generate the model observations to reduce the computational complexity.
(b) The representations of hidden states for HMMs are designed based on special characteristics of the flight operation, which are the segments of the planning route and standard flight levels of CAAC for the position and altitude model respectively.
(c) A trajectory similarity based correcting algorithm is proposed to improve the accuracy of prediction results after the flight takes off, which can further support the air traffic management.
(d) Experiments are conducted to determine pre-model parameters for HMMs, which is very important to the proposed model for illustrating the flight operation.

The rest of this paper is organized as follows. The related backgrounds are introduced in Section 2. Our proposed method is summarized in Section 3. Section 4 lists the learning and prediction algorithms. The trajectory similarity based correcting algorithm is proposed in Section 5. The simulation results are reported and discussed in Section 6. Conclusions are in Section 7.

II. RELATED BACKGROUND

Historical trajectories for all flights are stored in a database system (centralized or distributed), named HTSDB. Each single trajectory has been preprocessed by smoothing, denoising and interpolation algorithms to improve the data quality and keep a uniform updating interval (Zheng and Zhou, 2011; Ding et al., 2015; Vukovića, 2015). Search trees are constructed for each flight to improve the access efficiency. The database structure is sketched as follows:

$$\begin{aligned} HTSDB &= \left\{F_1, F_2, \ldots, F_{nf}\right\} \\ F_m &= \left\{T_1, T_2, \ldots, T_{nt}\right\} \\ T_n &= \left\langle p_1, p_2, \ldots, p_{np}\right\rangle \\ p_i &= \left[lon,\ lat, alt, t\right] \end{aligned} \tag{1}$$

There are nf trajectory sets of different flight saved in the database, in which a given flight $\boldsymbol{F}_m$ has been carried out nt times in history. The flight positions during every execution are collected by the surveillance equipment with same updating period (typically 4s). A single trajectory $\boldsymbol{T}_n$ is a time series of flight positions, which contains np discrete track positions with a same sampling interval. A track position p_i is given attributes of longitude, latitude, altitude, and time stamp.

HMM was proposed to predict the future information of the object by describing dynamic transitions among discrete states and the relationship between the sequence of observations and hidden states. HMM based approaches were widely used to estimate the "stay position" in the research of ground transportation (Alligier et al., 2015; Zahariand, 2015; Tang et al., 2016). The model parameters of the HMM are listed as follows:

$$\lambda_{HMM} = \{\boldsymbol{Y}, \boldsymbol{X}, \boldsymbol{A}, \boldsymbol{B}, \pi\} \tag{2}$$

where $\boldsymbol{Y} = \left\langle y_1, y_2, \ldots, y_T\right\rangle$ represents the observation sequence. $\boldsymbol{X} = \{x^{(1)}, x^{(2)}, \ldots, x^{(k)}\}$ is a finite set of hidden states. In HMM, every $y_i (1 \le i \le T)$ corresponds to a hidden state $x_i \in \boldsymbol{X}$, which further generates the hidden state sequence $\boldsymbol{S} = \left\langle x_1, x_2, \ldots, x_T\right\rangle$. $\boldsymbol{A}$ is the transition probability matrix of hidden states. $\boldsymbol{B}$ is the measurement probability matrix between the observation and hidden state. $\boldsymbol{\pi}$ is the initial distribution of the hidden state.

For any element $a_{ij} \in \boldsymbol{A}$, $a_{ij} = p(s_{t+1} = x^{(j)} \mid s_t = x^{(i)})$ represents the probability of having a hidden state $x^{(j)}$ at time $t+1$ given the hidden state $x^{(i)}$ at time t. Similarly, $b_{ij} = p(y_j \mid s_t = x^{(i)}), \forall b_{ij} \in B$ represents the probability that the observation is y_j on condition that the hidden state is x^i at time t, $\forall s_t \in \boldsymbol{X}, \forall y_t \in \boldsymbol{Y}$.

$$\begin{aligned} p(s_t \mid s_{t-1}, \ldots, s_1, y_{t-1}, \ldots, y_1) &= p(s_t \mid s_{t-1}) \\ p(y_t \mid s_t, \ldots, s_1, y_{t-1}, \ldots, y_1) &= p(y_t \mid s_t) \end{aligned}$$

This is the Markov property for the HMM, it is clear that the transition probability of the hidden state at time t depends only on the hidden state at time $t-1$, hidden states at time $t-2$ and before have no influence on the conditional distribution. Similarly, the observation at time t only depends on the hidden state s_t. The detailed definitions of HMM are referred in (Rabiner, 1989). In this work, the primary tasks are to determine the representations for the observations and hidden states of HMM based on unique characteristics of the flight trajectory.

III. FLIGHT MODEL

1. Position Model

1) Observation

The historical trajectory of a flight is a time series of track

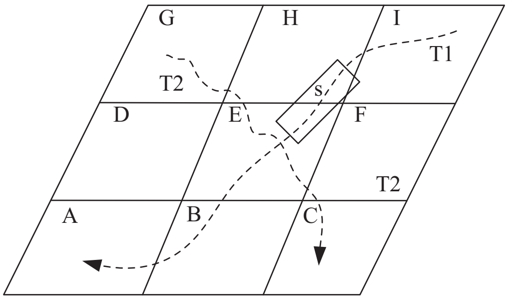

Fig. 1. Example of gridded cells.

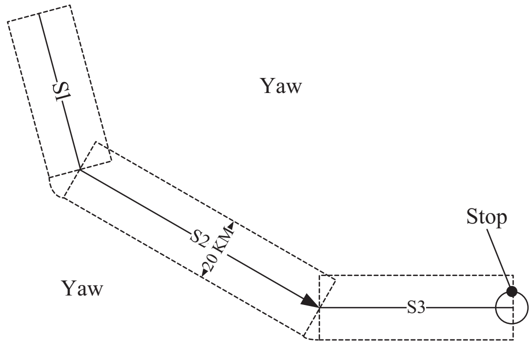

Fig. 2. Example of hidden states.

positions and updated in high frequency, which produce large amount of raw data. Basically, track positions of the flight can be regarded as the observations of the HMM directly. However, the values of track positions (longitude, latitude, altitude and timestamp) are sampled from a continuous space which will hugely aggravate the computational burden. Meanwhile, because of the high updating frequency for position collection, a long observation sequence makes the model traps into data amount disaster. On the other hand, the moving parameters of neighboring track positions along the flight path will not change sharply because of the constraints of aircraft performance for civil aviation. Therefore, to improve the computation efficiency, we propose a solution to substitute the track positions with fixed size cells. Based on the spatial extent of the cell, the track positions of trajectory (locations) can be represented by a sequence of cells sorted by the flight time. A simple example of the gridded cells is shown in Fig. 1.

In Fig. 1, the research area is divided into 9 gridded cells with same size, which are labeled as A (lower-left) - I (upper-right) depends on their locations. Any flight trajectory in the area can be expressed as a label sequence of gridded cells. For example, the grid label sequence of trajectory T1 and T2 are IHEBA and GHEFC respectively. After this procedure, the raw trajectory with continuous value of track positions will be represented by the sequence of cell labels with finite and discrete options, which can considerably improve the model efficiency. Obviously, the label sequence of the trajectory depends on the spatial granularity of the grid. There is a tradeoff between the prediction accuracy and computational complexity: we may obtain more accurate prediction results with heavy computing load by selecting a small size of cells (S_c). On the contrary, if the gridded cells are in a large size, the cost of calculation can be decreased at the expense of the prediction accuracy. Answer-loss problem would like to occur when making unreasonable partition for the research area since a large cell size may loss some important transitions of flight trajectory. Generally, only the areas in the envelope of planning route needs to be divided into gridded cells in this model.

2) Hidden State

The key to estimate the fly-over time and altitude for 4-D trajectory management is to find transition states for route segments during flight operation. In this model, the set of the hidden state is defined as possible segments of its flight planning route. By inferring the sequence of hidden states from the observations, we can find the transition patterns of the flight trajectory, which is used to predict the fly-over time and altitude of each waypoint. Based on the fact that the normal flight should be within the extent of route segments (Qiao et al., 2015), we define the hidden states of the position model as follows:

(a) Index of route segments, indicating that the aircraft flies in rectangular regions formed by covering 10 kilometers on both sides of the segment of adjacent waypoints along planning route.
(b) Stop (0), indicating that the aircraft flies into the region of the arrival airport.
(c) Yaw (-1), indicating that the aircraft deviates from the planning route and enters a yaw region.

The definition of hidden states covers all possible positions of an aircraft. By our proposed definitions, the set of hidden states is generated as the discrete index of route segments denoting the flight phase. The transitions of route segments (hidden states) in the position model and the altitude model indicate the fly-over time and altitude of given waypoints.

The rectangular region *s* in Fig. 1 is a hidden state based on our definition. For given flight plan, the waypoints can be extracted from the AFTN (Aeronautical Fixed Telecommunication Network) and hidden states of the model can be created by connecting two neighboring waypoints.

In Fig. 2, the outermost rectangle is the research area of the gridded map, which covers the planning route of the flight. Based on our definition, there are 5 hidden states for this flight, i.e., 3 route segments (rectangle with dot line marked as S1, S2, S3), stop (circle), and yaw (the other regions in the gridded map). The flight trajectory is converted to the label of the gridded map, which represents the spatial extent, but the aircraft positions did not correlate with the flight plan information. From this point, the route segments can also describe the spatial correlations by parsing the flight plan. By combining the trajectory and flight plan, the 4-D trajectory can be predicted from historical trajectories. Here, we summary the position model as follows:

(a) The observation is denoted by the label sequence of our

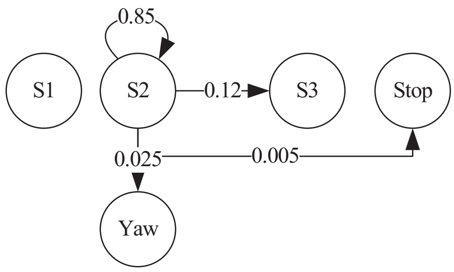

Fig. 3. Example of transition probability of S2.

proposed gridded map method, which describes the information of flight trajectory, and

(b) The set of hidden states in our proposal is the route segments, stop and possible yaw area, which illustrates the transition patterns from flight plan.

3) Hidden State Transition Probability Matrix

In HMM, the hidden state transition probability matrix is defined as the probability distribution of transition patterns between hidden states (i.e., route segments, stop and yaw in this paper). This hidden state transition probability matrix $\boldsymbol{A}$ can be optimized based on the known historical trajectories once the set of hidden state is designed. Due to the irreversible particularity of flight operation, the transition of hidden states in the model must be one of the following four cases:

(a) Index of current state, indicating that the aircraft still flies in current route segment, or
(b) Index of the next segment along the planning route, indicating that the aircraft flies to the next route segment, or
(c) -1, indicating that the aircraft deviates from the planning route and enters the yaw regions, or
(d) 0, indicating that the aircraft reaches its destination.

Fig. 3 shows an example of hidden state transitions for the flight plan in Fig. 2. It indicates that when the current state is S2, the probability of next hidden state remaining S2 is 0.85, the probability of being S3 is 0.12, while probability of yaw and stop state are 0.025 and 0.005 respectively. The sum of transition probabilities from a certain hidden state to other hidden states equals 1, which indicates that all possible transitions between any two of hidden states are considered in this model.

4) Measurement Probability Matrix

Measurement probability matrix is a description of the probability distribution between the observation and hidden state at a given prediction moment. In our model, the matrix clarifies the conditional probability of observations given certain hidden states, i.e., $p(y = y_i \mid x = s_j)$. The sum of each row in the matrix equals 1 indicates that all observations can be measured by hidden states implicitly. In general, the measurement probability matrix can be optimized based on historical trajectories. Intuitively, the measurement probability of the hidden state in this work can be represented by the proportion of track positions with each route segment (hidden state) in different gridded cells (observation). In this sense, the measurement probability of hidden state $\boldsymbol{s}$ in Fig. 1 can be computed below:

$$p(y = E \mid x = s) = \frac{3}{7},\ p(y = H \mid x = s) = \frac{4}{7}$$

5) Initial Distribution of Hidden State

A typical HMM application is to evaluate the probability of an observation sequence given optimized model parameters, which can be expressed as follows mathematically:

$$\begin{aligned} p(\boldsymbol{Y} \mid \lambda) &= \sum\nolimits_X p(\boldsymbol{Y}, S \mid \lambda) = \sum_{s_1 = x^{(1)}}^{x^{(k)}} \cdots \sum_{s_T = x^{(1)}}^{x^{(k)}} p(y_T, \ldots y_1, s_T \ldots s_1 \mid \lambda) \\ &= \sum_{s_1 = x^{(1)}}^{x^{(k)}} \cdots \sum_{s_T = x^{(1)}}^{x^{(k)}} [p(y_T \mid s_T) p(s_T \mid s_{T-1}) \ldots p(y_1 \mid s_1)\pi] \\ &= \sum_{s_1 = x^{(1)}}^{x^{(k)}} \cdots \sum_{s_T = x^{(1)}}^{x^{(k)}} \pi \prod_{t=2}^{T} a_{s_t s_{t-1}} b_{y_t s_t} \end{aligned} \tag{3}$$

The Markov Property is used to simplify the equation in (3), in which the conditional probability a_{ij} and b_{ij} can be obtained from the transition probability matrix $\boldsymbol{A}$ and measurement probability matrix $\boldsymbol{B}$. However, the initial distribution of hidden state $\boldsymbol{\pi}$ needs to be modeled in advance according to the empirical information (Alligier et al., 2015). In this paper, due to all aircraft take off from their departure airport, we initialize the initial distribution of hidden state $\boldsymbol{\pi}$ as a uniform distribution whose mean and standard deviation can be optimized from training data. Intuitively, the initial position of flight locates at the starting point of the runway. However, due to the measurement error of surveillance systems, the value is also not a deterministic one. Therefore, it is reasonable to describe the initial position of the flight by a probabilistic distribution in our proposed method.

2. Altitude Model

According to the HMM, the five parameters in (2) should be well represented for the altitude model. Actually, the altitude model can be built in a similar way with that of the position model. Firstly, just like the position model, we select customized flight levels as the observation of the altitude model rather than using the real collected altitude to reduce the size of value space. The altitude of airspace extends from 0 to 14900 meters in China Mainland. We divide the altitude into customized flight levels, in which we need to determine the altitude interval parameter, just like the cell size in the position model. The altitude interval will be verified by experiments to obtain more accurate prediction. In the altitude model, the hidden states are designed as the standard flight levels of CAAC, from level 0 to level 45, whose details are referred in (Xu, 2014). The hidden state transition probability matrix and measurement probability matrix in altitude model are generated by a similar way in the position model. Finally, as for the initial distribution of the hidden state, we apply one-dimensional Gaussian distribution whose mean and standard deviation can be optimized by the altitude of first collected track position for each execution from training data. Similarly, the initial altitude of aircraft should be the elevation of the departure airport, but we use a probabilistic distribution to describe it by considering the measurement error of surveillance systems. The initial value of the altitude is sampled from the learned distribution and converted to the flight level which corresponds to the initial hidden state. In summary, the altitude model is built based on following rules:

(a) The observations are denoted by the label sequence of customized flight levels, and
(b) The set of hidden states is the standard flight levels designed by CAAC, and
(c) The transition of hidden states comprises of current flight level, adjacent upper and lower flight levels and the elevation of arrival airport, in which the first two categories are used to describe the transition of hidden states and the last one indicates the terminal of the prediction sequence.

IV. PARAMETER LEARNING

The Baum-Welch algorithm, a special case of the expectation-maximization (EM) algorithm, is a classic algorithm for learning HMM parameters (Qiao et al., 2015). The working steps of the EM algorithm can be explained as follows:

(1) E-Step: calculating the maximum likelihood estimation by the model parameters in current step;
(2) M-Step: calculating the maximum likelihood parameters by the maximize value in E-Step.

The two steps are executed iteratively to obtain optimal model parameters. In HMM, we write the target equation of parameter optimization as (4) since both the hidden state and observation in our work are discrete.

$$\lambda^{(g+1)} = \arg\max_{\lambda} \underbrace{\left(\int_{x \in X} \ln\left(p\left(\mathbf{Y}, x \mid \lambda\right)\right) p\left(x, \mathbf{Y} \mid \lambda^{(g)}\right) dx \right)}_{\rho\left(\lambda, \lambda^{(g)}\right)} \tag{4-a}$$

$$\rho\left(\lambda, \lambda^{(g)}\right) = \sum_{x_0=1}^{k} \cdots \sum_{x_T=1}^{k} \left(\ln \pi + \sum_{t=1}^{T} \ln a_{x_{t-1}, x_t} + \sum_{t=1}^{T} \ln b_{x_t}\left(y_t\right) \right) p\left(x, Y \mid \lambda^{(g)}\right) \tag{4-b}$$

Therefore, the optimization of the EM algorithm for HMM can be achieved by optimizing the three items in (4-b) respectively. The classic Lagrange multiplier will be used in E- and M-step of the algorithm to obtain model parameters (Rabiner, 1989). As mentioned before, the sum of the transition probability of a certain hidden state and its measurement probability with observations are 1, which serves as the constrains for the Lagrange multiplier. Since EM has a very heavy computation

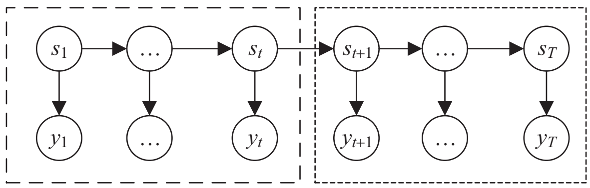

Fig. 4. Diagram of forward-backward algorithm.

cost, an improved forward-backward algorithm is applied in this paper to decrease the computational complexity from k^T to k^2T (Rabiner, 1989), where k is the total number of hidden states, and T is the sequence length. The computational complexity also supports the necessity of our proposal to improve the learning efficiency by replacing the collected position and altitude with the gridded cell and customized flight level respectively.

In the forward-backward algorithm, the left and right parts in Fig. 4 are defined as (5) and (6) respectively:

$$\partial_i(t) = p(y_1, \ldots, y_t, s_t = x^{(i)} \mid \lambda) \tag{5}$$

$$\beta_i(t) = p(y_{t+1}, \ldots, y_T, s_t = x^{(i)} \mid \lambda) \tag{6}$$

On the basis of the following latent variables:

$$\begin{aligned} \xi_{ij}(t) &= p(s_t = x^{(i)}, s_{t+1} = x^{(j)} \mid Y, \lambda) \\ &= \frac{\partial_i(t) a_{ij} \beta_j(t) b_{jy_{t+1}}}{\sum_{i \in X} \sum_{j \in X} \partial_i(t) a_{ij} \beta_j(t) b_{jy_{t+1}}} \end{aligned} \tag{7}$$

$$\gamma_i(t) = p(s_t = x^{(i)} \mid Y, \lambda) = \frac{\partial_i(t) \beta_i(t)}{\sum_{i \in X} \partial_i(t) \beta_i(t)} \tag{8}$$

The optimal estimation of model parameters can be derived as follows (Rabiner, 1989):

$$\pi = \gamma_i(1)$$

$$a_{ij} = \sum_{t=1}^{T-1} \xi_{ij}(t) / \sum_{t=1}^{T-1} \gamma_i(t)$$

$$b_{jk} = \sum_{t=1, y_t = k}^{T-1} \gamma_j(t) / \sum_{t=1}^{T-1} \gamma_j(t)$$

V. PREDICTION CORRECTING

After the flight takes off, a series of real-time track positions are collected by surveillance equipment. Based on the collected track positions, we correct our pre-takeoff 4-D trajectory prediction results by presenting an algorithm named Trajectory Similarity based Updating Algorithm. The historical trajectories (a sequence of track positions), which have the higher similarity with real-time collected path, are more credible when correcting the pre-takeoff 4-D trajectory. Therefore, only the most similar trajectories are used to correct the pre-takeoff 4-D trajectory in this work (Zahariand, 2015). There are several measurements can be used to evaluate the similarity between two trajectories (time series data), such as Euclidian Distance, Dynamic Time Warping (DTW) Distance, Longest Common Subsequence (LCSS), Edit Distance with Real Penalty (ERP), Edit Distance on Real Sequences (EDR), et al. (Zheng and Zhou, 2011). In the database, the historical trajectories are completed ones from departure to arrival airport for given flight, while the real-time path is only a sub-sequence of trajectory until the correcting instant. By analyzing the applicability of the mentioned measurements, the DTW distance is the most proper tool for evaluating the similarity of trajectories with different length.

Given trajectories $A = a_1, \ldots, a_n$ and $B = b_1, \ldots, b_m$, and the attributions of a_i and b_i are same as p_i in (1) except t. To measure the distance between two positions, all positions with latitude and longitude are converted into a same projected coordinate system to keep a unified unit (in meter). The m and n are the length of corresponding sequence. Let $Head(A)$ and $Rest(A)$ denote a_1 and $a_2, \ldots, a_n$ respectively, the DTW distance can be defined as follows:

$$DTW(A, B) = \begin{cases} 0, m = 0 \textbf{ and } n = 0 \\ \infty, m = 0 \textbf{ or } n = 0 \\ d_{head} + d_{rest}, otherwise \end{cases} \tag{9}$$

$$\begin{aligned} d_{head} &= d(Head(A), Head(B)) \\ d_{rest} &= \min \begin{cases} DTW(A, Rest(B)) \\ DTW(Rest(A), B) \\ DTW(Rest(A), Rest(B)) \end{cases} \end{aligned} \tag{10}$$

The notation $d(p_a, p_b)$ in (10) is the Euclidian Distance of two spatial points in the 3-D space. The computational complexity of DTW algorithm is mnq, where q is the computational complexity of the Euclidian Distance algorithm. Once the aircraft passes a waypoint of the planning route, the updating algorithm is executed as follows:

(1) Find the most similar l historical trajectories (in this paper $l = 5$) for the flight in the database, whose DTW distance with collected path are $dtw_1, \ldots, dtw_1$ respectively;
(2) Calculate and normalize the weights for each historical trajectory in the correcting procedure:

$$w_i = \exp\left(-dtw_i\right) / \sum_{j=1}^{l} \exp\left(-dtw_j\right) (i = 1, \ldots, l) \tag{11}$$

(3) Correct the pre-takeoff 4-D trajectory with (12), where the superscript ‘c’ and ‘o’ denote corrected and pre-takeoff prediction results respectively. t_k^c and h_k^c are the corrected fly-over time and altitude respectively. The superscript i represents the index of selected historical trajectories. k is the index of the waypoint along the flight planning route from the departure airport. By the correcting algorithm, the prediction result for each waypoint will be updated when the aircraft go through a waypoint.

$$t_k^c = (t_k^o + \sum_{i=1}^{l} w_i t_k^i) / 2, \; h_k^c = (h_k^o + \sum_{i=1}^{l} w_i h_k^i) / 2 \tag{12}$$

Table 1. Basic flight information of training data.

Identity	Departure	Arrival	Flight time	Cruising speed	Cruising altitude
FL1	SWA	PEK	175 min	800 km/h	9800 m
FL2	CTU	PEK	150 min	800 km/h	10800 m

VI. EXPERIMENT AND DISCUSSION

In this section, several experiments are firstly conducted to optimize the parameters which affect the observations and hidden states of HMMs. The cell size of the gridded map and altitude interval (hereafter we call them as pre-model parameters) need to be verified by experiments. Based on the pre-model parameters, the optimal model parameters of HMMs can be learned from historical data, and then the pre-takeoff 4-D trajectory is predicted by the proposed model. The ground truth value of test samples is regarded as the real-time data to correct the pre-takeoff 4-D trajectory by the proposed correcting algorithm. We apply two real flights in our database to test the proposed approach, whose basic information is listed in Table 1.

The training data in this work is collected historical trajectories of given flights from March 1, 2015 to February 28, 2016. There are 327 and 301 historical trajectories for flight FL1 and FL2 after removing the low-quality data, which are used to train the proposed HMM models. The main purpose of the simulation is to predict the 4-D trajectory (fly-over time and altitude of all waypoints on the planning route) for the test flights. A total of 15 flight executions for each flight after February 29, 2016 serve as the test data, i.e., about 5% of the amount of the training data. We also own the real flight data as test samples to evaluate the prediction performance of different methods. According to the planning route of flight plans, there are 9 waypoints along the planning route for FL1, while 10 waypoints for flight FL2. The detailed positions of planning route for given flights are shown in Fig. 5, in which we also specify the departure and arrival airport for each flight. In Fig.5, the rectangle and triangle denote the departure and arrival airport respectively, while the circles are the waypoints of their planning route.

To show the further performance superiority, we also compare our prediction results with that of existing methods, including kinematics-dynamics approach (KDA), regression model of his-

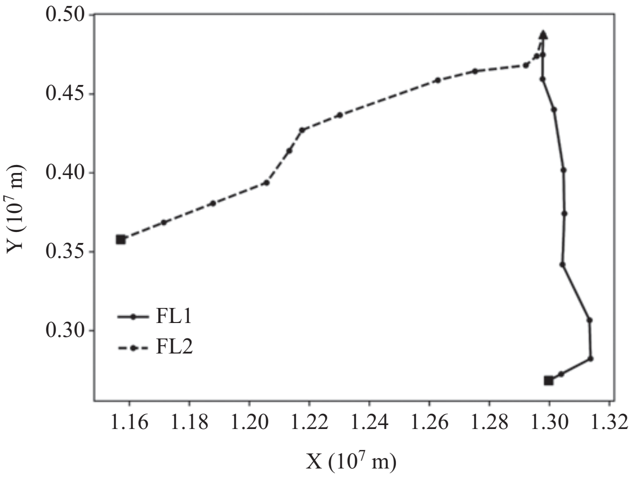

Fig. 5. Planning route of flights in the training data.

torical trajectory (RMHT) and probabilistic distributed based model (PDBM), to evaluate the accuracy and stability of the prediction results. The three comparative approaches are described as follows briefly:

(a) The KDA method is based on the kinematics-dynamics rules. The whole flight process is divided into different phases, in which different kinematics-dynamics patterns are applied to calculate the 4-D trajectory information.
(b) The RMHT method is a regression-based one, whose regression equation is the linear polynomial according to the certain application. The parameters of the regression model are also learned from the sample data. To this extent, it is also a type of machine learning approach.
(c) The core idea of PDBM is that the values of fly-over time and altitude of each waypoint are subject to a probabilistic distribution. In our work, two-dimensional Gaussian distribution (one dimension for fly-over time, the other for the fly-over altitude) is used to describe the data features. The mean and standard deviation of the distribution are the parameters which can be optimized from training data.

In this paper, the KDA and RMHT methods are implemented based on the details in (Chen, 2012) and (Hamed, 2013) respectively, while the PDBM approach is implemented based on the details in (Song, 2012). The main hardware configurations of our training server are summarized as: 2*Intel Xeon E5-2650 CPUs 2.80 GHz, 64GB memory, and 4TB hard disk. All implementations in this work are programmed using Python. In this section, two measurements are applied to evaluate the performance of prediction results:

(a) Waypoint errors:

$$t_e^i = \sum_{j=1}^{15} \left| t_p^{i,j} - t_r^{i,j} \right|,\ h_e^i = \sum_{j=1}^{15} \left| h_p^{i,j} - h_r^{i,j} \right| \tag{13}$$

where the $t_\bullet^\bullet$ and $h_\bullet^\bullet$ denote the prediction error of the fly-over time and altitude respectively, i is the waypoint index along the planning route for given flight. The subscript p and r are predicted results and real values respectively. 15 is the total number of test flights.

(b) Mean and standard deviation of waypoint errors: mean and standard deviation of prediction error series for the whole trajectory: $t_e^1, \ldots, t_e^L$ and $h_e^1, \ldots, h_e^L$, where L is the number of waypoints on the planning route.

1. Pre-Model Parameters

In this section, different cell size and altitude interval are selected for modeling observations in simulations to check the prediction performance. The mean and standard deviation of prediction results (without correction) are used to evaluate the performance, and further to determine the most appropriate pre-model parameters. A larger cell size and altitude interval is highly recommended to improve the efficiency of our algorithm if a similar prediction accuracy can be achieved. We select the side length of gridded cells from 1 to 10 km (corresponding to the area of cells from 1 to 100 km^2) and the altitude interval from 10 to 100 m to conduct different experiments. Elbow rule[1] is a good guidance for parameters selection. In case of ensuring the prediction accuracy, larger pre-model parameters are inclined to be selected to reduce the computational complexity for model learning and prediction.

From the experiment results (Fig. 6 and Fig. 7), we can draw the following conclusions.

(1) The mean errors of the predicted fly-over time and altitude become larger with the increasing of the cell size and altitude interval for both FL1 and FL2. However, the standard deviations of prediction errors of the fly-over time and altitude float within a narrow range.
(2) For both the flight FL1 and FL2, there is a sharp increase for the prediction errors of the fly-over time when the cell size is greater than 16 km^2 which is regarded as a better option for the cell size selection in this paper based on the Elbow rule.
(3) For both the flight FL1, there is a sharp increase for the prediction errors of the fly-over altitude when the altitude interval is greater than 30 m. Therefore, the optimal options of the altitude interval for them are 30 meters in this paper based on the Elbow rule.

2. Evaluation of Prediction Results

After determining pre-model parameters, the observations

[1] It is a term in Machine Learning course instructed by Andre Ng. When encountering a tradeoff between the computing complexity and predicting accuracy, we select the maximum parameters before the deteriorating of predicting accuracy to reduce the computing complexity.

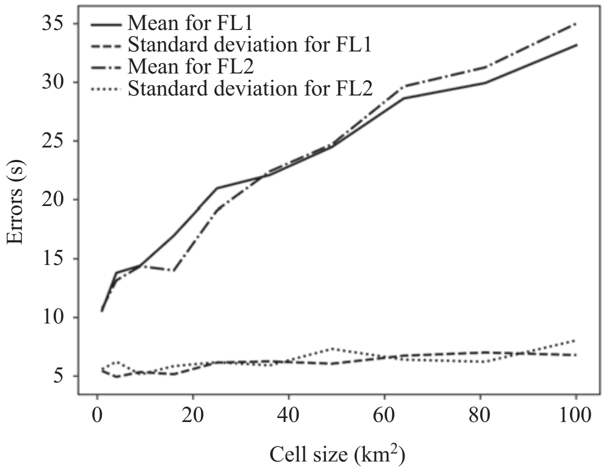

Fig. 6. Mean and standard deviation errors of predicted fly-over time with different cell sizes.

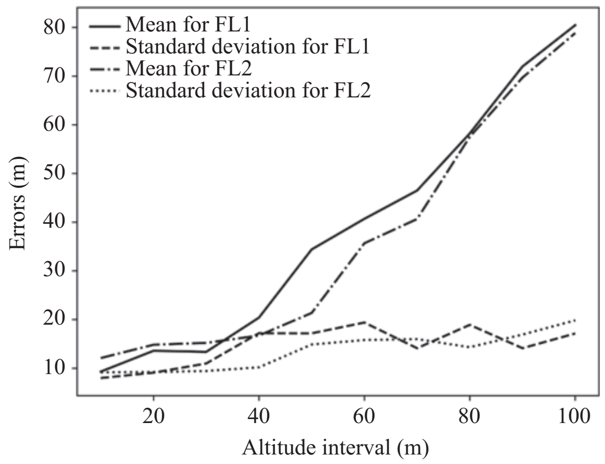

Fig. 7. Mean and standard deviation errors of predicted fly-over altitude with different altitude interval.

and hidden states of HMMs are generated from raw training data (historical trajectories). For flight FL1, there are 12 hidden states (10 route segments, stop and yaw) for the position model while 497 hidden states (customized flight levels) for the altitude model. When it comes to the flight FL2, the number of hidden states for the position and altitude model are 13 and 373 respectively. EM based algorithm is applied to learn optimal parameters for HMMs, and the pre-takeoff 4-D trajectory for test flights are predicted by learned models. Finally, the proposed correcting algorithm is used to improve the accuracy of pre-takeoff prediction results. To show further superiority over existing approaches, we conduct several experiments with KDA, RMHT and PDBM methods and compare the prediction results with that of our proposed approach (without correction). The comparison of prediction results (fly-over time and altitude of each waypoint) with different methods for FL1 and FL2 are listed in Table 2 and Table 3. In the two tables, the values before and after "/" denote the fly-over time (in seconds) and altitude (in meters)

Table 2. Comparison of waypoint errors for FL1 with different methods.

Waypoints	KDA	RMHT	PDBM	HMM*
1st waypoint	35.1/38.7	23.9/32.0	25.6/38.7	11.5/10.0
2nd waypoint	24.5/31.3	20.1/20.0	17.8/19.3	14.3/17.3
3rd waypoint	9.3/18.7	14.5/18.7	12.0/12.0	18.6/6.0
4th waypoint	12.3/20.0	12.4/17.3	10.5/19.3	8.5/13.3
5th waypoint	15.1/10.0	13.3/8.7	9.2/7.3	9.2/18.0
6th waypoint	18.6/20.7	13.9/10.7	16.7/18.7	11.6/10.0
7th waypoint	13.3/21.3	11.5/20.7	8.9/10.0	17.1/18.0
8th waypoint	13.9/30.0	22.5/19.3	19.1/19.3	20.0/12.0
9th waypoint	22.7/32.0	17.2/21.3	24.5/21.3	15.0/12.7
Arrival airport	34.1/0.0	26.7/0.0	20.3/0.0	18.2/0.0

Table 3. Comparison of waypoint errors for FL2 with different methods.

Waypoints	KDA	RMHT	PDBM	HMM*
1st waypoint	40.4/50.0	30.0/36.7	33.7/29.3	16.0/19.3
2nd waypoint	24.9/38.0	21.9/30.7	17.7/30.0	14.5/17.3
3rd waypoint	11.9/30.0	12.5/23.3	10.4/29.3	12.5/9.3
4th waypoint	9.5/11.3	10.8/18.0	8.8/10.7	11.7/12.0
5th waypoint	10.9/11.3	15.1/12.7	11.8/6.0	8.6/7.3
6th waypoint	14.0/11.3	14.3/11.3	15.1/21.3	10.3/9.3
7th waypoint	12.1/21.3	16.7/20.7	14.6/13.3	11.3/23.3
8th waypoint	18.7/21.3	20.4/22.0	20.1/8.0	18.7/11.3
9th waypoint	21.5/21.3	25.1/20.0	20.1/10.0	13.0/21.3
10th waypoint	16.5/12.7	14.5/12.0	17.1/10.7	9.9/20.7
Arrival airport	33.7/0.0	37.6/0.0	28.1/0.0	19.6/0.0

errors for given waypoint respectively. The correcting results obtained by the proposed algorithm for FL1 and FL2 are reported in Table 4 and Table 5. The representations of each cell in the table are same with that of Table 2 and Table 4, while the placeholder "-/-" means the data has been collected.

Noted that the prediction error of the fly-over altitude in the arrival airport is 0 since its elevation is known to us. From the comparison of prediction errors with different methods and the corrected results of the proposed method based on the correcting algorithm, we can summarize the experiments as follows:

(a) Even without the correcting procedure, prediction results obtained by the proposed approach are more accurate and stable than that of other methods (KDA, RMHT and PDBM), which support the effectiveness of the proposed approach on mining the transition patterns of flight trajectories.
(b) Unlike comparative algorithms that the prediction errors in the takeoff and landing stages are worse than that of in the cruise stage, the prediction errors obtained by the proposed approach are steady during the whole flight operation.
(c) By checking the prediction errors with the ground truth data, the corrected results are closer to real collected trajectory

Table 4. Corrected results for FL1 by the proposed algorithm.

Correcting	1st	2nd	3rd	4th	5th	6th	7th	8th	9th
2nd waypoint	4.0/12.0	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-
3rd waypoint	9.4/10.7	6.5/12.7	-/-	-/-	-/-	-/-	-/-	-/-	-/-
4th waypoint	11.9/8.0	8.4/8.7	2.0/6.7	-/-	-/-	-/-	-/-	-/-	-/-
5th waypoint	16.1/6.7	14.1/6.7	8.5/12.0	9.7/10.0	-/-	-/-	-/-	-/-	-/-
6th waypoint	13.0/10.0	14.9/7.3	14.2/10.0	12.0/8.7	7.4/12.0	-/-	-/-	-/-	-/-
7th waypoint	19.5/10.7	19.5/10.7	15.2/11.3	15.4/9.3	10.1/8.0	9.5/8.0	-/-	-/-	-/-
8th waypoint	14.4/10.0	16.7/16.0	12.5/9.3	13.9/10.0	16.3/7.3	10.5/5.3	15.5/4.7	-/-	-/-
9th waypoint	13.9/10.0	14.2/8.0	15.9/10.0	17.2/8.7	14.9/8.0	12.0/14.0	16.6/10.7	17.5/5.3	-/-
Arrival airport	16.7/0.0	12.3/0.0	13.1/0.0	14.1/0.0	13.6/0.0	15.0/0.0	14.8/0.0	19.5/0.0	14.2/0.0

Table 5. Corrected results for FL2 by the proposed algorithm.

Correcting	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
2nd waypoint	6.4/10.7	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-
3rd waypoint	11.9/10.7	7.0/7.3	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-
4th waypoint	13.1/12.7	10.1/12.7	4.0/10.7	-/-	-/-	-/-	-/-	-/-	-/-	-/-
5th waypoint	9.3/12.7	10.4/10.7	7.3/8.7	8.5/12.7	-/-	-/-	-/-	-/-	-/-	-/-
6th waypoint	11.1/9.3	7.8/10.7	9.6/10.7	10.0/7.3	6.5/10.7	-/-	-/-	-/-	-/-	-/-
7th waypoint	18.5/11.3	13.7/10.0	13.5/12.7	12.2/8.7	9.5/6.0	8.3/8.0	-/-	-/-	-/-	-/-
8th waypoint	15.9/21.3	13.1/10.7	11.3/10.7	14.2/16.7	11.8/10.0	9.9/10.0	12.5/10.0	-/-	-/-	-/-
9th waypoint	11.8/12.7	10.3/9.3	13.5/9.3	14.1/12.7	10.9/6.0	10.7/8.7	10.1/8.7	11.3/10.0	-/-	-/-
10th waypoint	10.5/12.7	13.3/10.0	11.7/9.3	11.5/7.3	13.5/20.7	10.9/14.0	13.7/8.7	9.9/9.3	7.0/6.0	-/-
Arrival airport	14.7/0.0	12.3/0.0	17.5/0.0	13.1/0.0	15.2/0.0	12.2/0.0	15.5/0.0	14.6/0.0	10.1/0.0	11.7/0.0

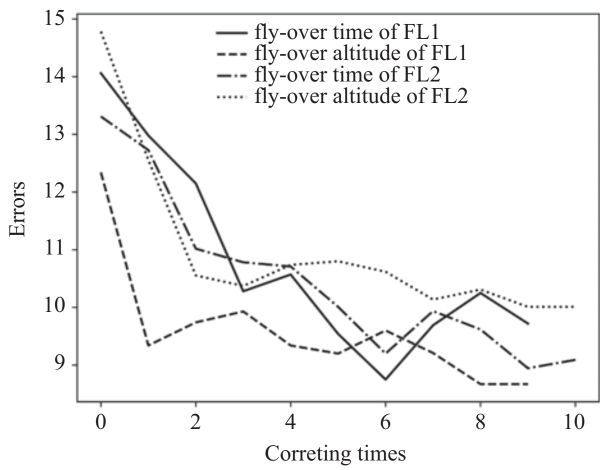

Fig. 8. Mean of the fly-over time and altitude after different correcting times.

compared to the pre-takeoff prediction results, which is also shows the validity of the proposed correcting algorithm. The mean errors of fly-over time and altitude after different correcting time are shown in Fig. 8, in which the errors of fly-over time and altitude are shown in seconds and meters respectively. The horizontal axis denotes the correcting times, where 0 indicates the mean errors of pre-takeoff prediction results. As shown in the figure, the mean errors of both fly-over time and altitude for flight FL1 and FL2 are generally in decline with the executions of the correcting procedure.

(d) It can be seen that the values of prediction error for FL1 are generally less than that of FL2. By analyzing the raw training data, we find that the flight trajectories of FL2 are distributed in a more divergent value space, while the trajectory distribution of FL1 is more convergent. Moreover, there are more departure and landing flights in CTU airport (departure of FL2), which causes more flow control issues and further impacts the prediction accuracy.

VII. CONCLUSION

In this paper, we present a stochastic probabilistic statistical model to illustrate the operation progress of the flight to predict its 4-D trajectory. The gridded map and altitude interval are used to generate the observations for the proposed HMM models. An EM based algorithm is applied to learn optimal parameters from the training data. Then the fly-over time and altitude of waypoints along planning route are predicted by learned models. In addition, once the flight takes off, the 4-D trajectory of flight is updated by correcting the pre-takeoff prediction results based on a trajectory similarity-based algorithm. Simulation results demonstrate that the proposed algorithm obtains more accurate and stable prediction results compared to other comparative methods.

After this work, we will further improve the accuracy and stability of 4-D trajectory by building an integrated HMM model for both the flight position and altitude. Moreover, we also plan to use the neural network-based algorithm to correct the pre-takeoff 4-D trajectory.

ACKNOWLEDGEMENTS

This research was supported by the National Natural Science Foundation of China (Grant Nos. 71573184) and the project of Office of China Air Traffic Control Commission (Grant Nos. GKG201403004) and the National Key Foundation for Exploring Scientific Instrument of China (Grant Nos. 2013YQ490879).

REFERENCES

Alligier, R., D. Gianazza and N. Durand (2015). Machine learning and mass estimation method for ground-based aircraft climb prediction. IEEE Trans. on Intell. Transp. Syst. 16(6), 3138-3149.

Ayhan, S. and H. Samet (2016). Aircraft trajectory prediction made easy with predictive analytics. Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery & Data Mining, San Francisco, California, USA, 21-30.

Chen, Z. J. (2012). Theory and Method of Airspace Management. Science Press, Beijing, China.

Ding, Z. M., B. Yang, R. H. Güting and Y. G. Li (2015). Network-matched trajectory-based moving-object database-models and applications. IEEE Trans. on Intell. Transp. Syst. 16(4), 1918-1928.

Hamed, G. M., D. Gianazza, M. Serrurier and N. Durand (2013). Statistical prediction of aircraft trajectory: regression methods vs point-mass model. 10th USA/Europe Air Traffic Management Research and Development Seminar, Chicago, IL, USA, 1-10.

Hostettler, R., W. Birk and L. M. Nordenvaad (2015). Joint vehicle trajectory and model parameter estimation using road side sensors. IEEE Sensors Journal 15(9), 5075-5086.

Jeung, H., H. T. Shen and X. F. Zhou (2007). Mining trajectory patterns using hidden markov models. 2007 International Conference on Data Warehousing and Knowledge Discovery, Regensburg Germany, 470-480.

Kim, Y. K., J. K. Han and H. Park (2015). Trajectory prediction for using real data and real meteorological data. Ubiquitous Computing Application and Wireless Sensor 331(1), 89-103.

Li, L. C., S. L. He, J. Zhang and B. Ran (2016). Short-term highway traffic flow prediction based on a hybrid strategy considering temporal-spatial information. Journal of Advanced Transportation 50(8), 2029-2040.

Li, Z., S. H. Li and X. L. Wu (2015). General aircraft 4D flight trajectory prediction method based on data fusion. 2015 International Conference on Machine Learning and Cybernetics (ICMLC), Guangzhou, China, 309-315.

Lin, Y., J. W. Zhang and H. Liu (2017). An algorithm for trajectory prediction of flight plan based on RMBP. Frontiers of Information Technology & Electronic Engineering 19(7), 905-916.

Melnyk, I., A. Banerjee, B. Matthews and N. Oza (2016). Semi-markov switching vector autoregressive model-based anomaly detection in aviation systems. Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery & Data Mining, San Francisco, California, USA, 1065-1074.

Morzy, M. (2007). Mining frequent trajectories of moving objects for location prediction. Proc. 5th Int. Conf. Mach. Learn. Data Mining Pattern Recognition, Leipzig, Germany, 667-680.

Paglionee, M. M. and R. D. Oaks (2007). Implementation and metrics for a trajectory prediction validation methodology. AIAA Guidance, Navigation and Control Conference and Exhibit, Hilton Head, South Carolina, USA, 1-18.

Qiao, M. Y., W. Bian, Y. D. Xu and D. C. Tao (2015). Diversified hidden markov models for sequential labeling. IEEE Transactions on Knowledge & Data Engineering 27(11), 2947-2960.

Qiao, S. J., D. Y. Shen, X. T. Wang, N. Han and W. Zhu (2016). A self-adaptive parameter selection trajectory prediction approach via hidden markov models. IEEE Trans. on Intell. Transp. Syst. 16(1), 284-296.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of IEEE 77(2), 257-286.

Song, L. L. (2012). A 4-D trajectory prediction method based on set of historical trajectory. Computer Technology and Development 12(22), 11-14.

Tang, K. S., S. F. Zhu, Y. Q. Xu and F. Wang (2016). Modeling drivers' dynamic decision-making behavior during the phase transition period: an analytical approach based on hidden Markov model theory. IEEE Trans. on Intell. Transp. Syst., 17(1), 206-214.

Tang, X. M., P. Chen and Y. Zhang (2015). 4D trajectory estimation based on nominal flight profile extraction and airway meteorological forecast revision. Aerospace Science and Technology 45(1), 387-397.

Tang, X. M., J. W. Gu, Z. Y. Shen and P. Chen (2015). A flight profile clustering method combining twed with K-means algorithm for 4D trajectory prediction. 2015 Integrated Communication, Navigation and Surveillance Conference (ICNS), Herdon, VA, USA, S3-1 - S3-9.

Tang, X. M., L. Zhou, Z. Y. Shen and M. Tang (2015). 4D trajectory prediction of aircraft taxiing based on fitting velocity profile. 15th COTA International Conference of Transportation Professionals, Beijing, China, 1-12.

Vukovića, N., M. Mitićbm and Z. Miljković (2015). Trajectory learning and reproduction for differential drive mobile robots based on GMM/HMM and dynamic time warping using learning from demonstration framework. Engineering Applications of Artificial Intelligence 45, 388-404.

Wandelt, S and X. Q. Sun (2015). Efficient compression of 4D-trajectory data in air traffic management. IEEE Trans. on Intell. Transp. Syst. 14(2), 844-853.

Xie, A. M. and P. Cheng (2015). 4D approaching trajectory design in terminal area based on radar data. Applied Mechanics & Materials 740, 731-735.

Xu, C. (2014). Comparison and thought of three flight level system reforms of China. China civil aviation 12, 29-31.

Zahariand, A. and J. Jaafar (2015). Combining hidden markov model and case based reasoning for time series forecasting. Communications in Computer & Information Science 513, 237-247.

Zhang, J. F., H. X. Jiang and X. G. Wu (2014). 4D trajectory prediction based on BADA and aircraft intent. Journal of Southwest Jiaotong University 49(3), 553-558.

Zheng, Y. and X. F. Zhou (2011). Computing with Spatial Trajectories. Springer, New York, USA.