This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Learning Probabilistic Trajectory Models of Aircraft in Terminal Airspace From Position Data

Shane T. Barratt, Mykel J. Kochenderfer, and Stephen P. Boyd

***Abstract*— Models for predicting aircraft motion are an important component of modern aeronautical systems. These models help aircraft plan collision avoidance maneuvers and help conduct off-line performance and safety analyses. In this paper, we develop a method for learning a probabilistic generative model of aircraft motion in terminal airspace, the controlled airspace surrounding a given airport. The method fits the model based on a historical dataset of radar-based position measurements of aircraft landings and takeoffs at that airport. We find that the model generates realistic trajectories, provides accurate predictions, and captures the statistical properties of the aircraft trajectories. Furthermore, the model trains quickly, is compact, and allows for efficient real-time inference.**

***Index Terms*— Air traffic control, predictive models, machine learning, unsupervised learning, Gaussian mixture model, clustering methods.**

I. Introduction

THE United States National Airspace System is undergoing a multibillion dollar modernization to increase safety, efficiency, and predictability of air transport, also known as NextGen [1]. A major focus of the modernization is on the portion of flight that occurs in the terminal airspace surrounding an airport, which represents the greatest safety risk. In fact, 56% of fatal air-related accidents worldwide occur in the takeoff, initial climb, final approach, and landing phases of the flight, even though these four phases constitute only 6% of flight time [2].

Probabilistic trajectory models of aircraft in the terminal region are important for the development and assessment of air traffic control technologies. These trajectory models can be used to predict the future trajectories of aircraft, detect conflicts, and recommend avoidance maneuvers. In addition, they can be used to assess the safety and operational performance of new technologies and procedures in simulation before deployment. It is critical, however, that the trajectory models be accurate representations of how aircraft actually behave in the airspace. This paper focuses on the construction of probabilistic trajectory models of aircraft in terminal airspace from radar-based surveillance data.

Manuscript received April 23, 2018; revised August 16, 2018; accepted September 26, 2018. This work was supported by the National Science Foundation Graduate Research Fellowship under Grant DGE-1656518. The Associate Editor for this paper was Q. Zhang. *(Corresponding author: Shane T. Barratt.)*

S. T. Barratt and S. P. Boyd are with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305 USA (e-mail: sbarratt@stanford.edu; boyd@stanford.edu).

M. J. Kochenderfer is with the Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305 USA (e-mail: mykel@stanford.edu).

Digital Object Identifier 10.1109/TITS.2018.2877572

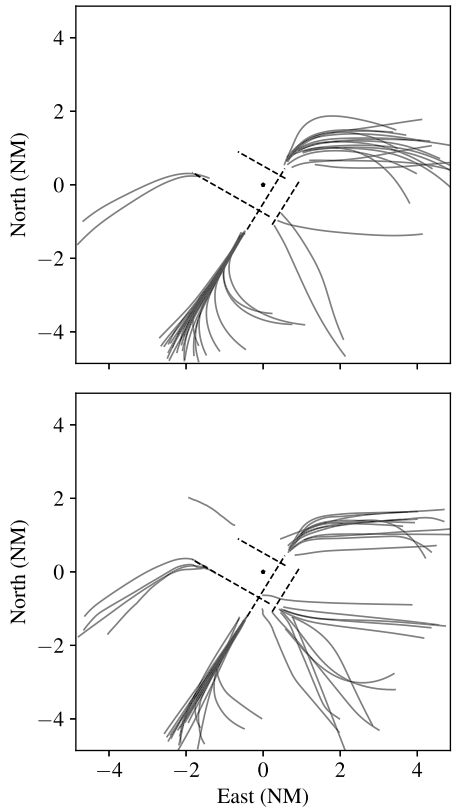


Fig. 1. Lateral view of takeoffs from KJFK. One figure is of held-out test trajectories and one is of trajectories generated by the model. Try to guess which one is real and which one is generated before looking at their true identities.[1]

A. Related Work

We begin by summarizing methods for modeling aircraft trajectories in unstructured airspace, *i.e.*, outside of the terminal region. One line of work involves predicting future aircraft motion by estimating the aircraft's state and propagating that estimate through physical equations of motion [5], [6]. Other methods include synthesizing trajectories by combining individual segments defined by different modes of operation [7] and estimating vertical paths based on the weight of the aircraft [8]. Another method proposed by Kochenderfer *et al.* [9], [10] is to learn Bayesian network statistical representations of dynamic variables from historical radar data. Given trajectory predictions and their covariances for two aircraft, Paielli and Erzberger [11] derive a method to solve for *conflict probability*, defined as the probability that the aircraft become too close. Other methods have been proposed

[1]Top is generated, bottom is held-out test samples.

1524-9050 © 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

to efficiently calculate conflict probabilities, including probability flow [12], [13] and importance sampling [14]. Many of these trajectory prediction models are used in production aviation systems, including the Center TRACON Automation System (CTAS) [15], [16], and for evaluating the Traffic Alert and Collision Avoidance System (TCAS) [17] and ACAS X, the next generation of TCAS [18]. Realistic airport simulations are also used to validate NextGen [19].

There are also hybrid dynamics models of aircraft in unstructured airspace, which explicitly take into account both continuous and discrete dynamic behavior. Prandini *et al.* [20] propagate aircraft state through a stochastic hybrid dynamical system model to perform conflict detection. Combining the stochastic hybrid dynamical system model with a method for inferring the aircraft's navigational intent leads to a commonly used set of modeling tools for aircraft trajectory prediction [21]–[23]. A recently proposed method incorporates the hybrid dynamics model with a Bayesian intent model [24]. These methods, for the most part, focus on learning probabilistic trajectory models for aircraft in unstructured airspace. Modeling aircraft in terminal regions is a fundamentally different task that cannot be solved with models for unstructured operation. It is difficult to apply such models because the structure of the arrival and departure procedures at airports leads to different behavior.

One way to learn models for the terminal region is by using the airport's published instrument procedures. However, doing this leads to a model that has no knowledge of the statistical variation in how the procedures are executed. A way to capture the statistical variation observed in practice is to learn directly from historical surveillance data.

Given the large amount of surveillance data available, several papers have proposed using machine learning techniques to learn models of terminal airspace. One method involves clustering *turning points*, defined as spatial positions where a substantial change of heading occurs. Gariel *et al.* [25] identify and cluster these so-called turning points across a dataset of trajectories, then treat a trajectory as a (discrete) sequence of the cluster labels, then cluster those (discrete) sequences using the least common subsequence algorithm. Mahboubi and Kochenderfer [26] find that the turning point model does not perform well on real, noisy radar data, and extend the work of Gariel *et al.*by representing the transitions between turning points at small airports using a Gaussian hidden semi-Markov model.

Another promising direction is to cluster trajectories at the level of position measurements, which is also the principle behind air traffic flow modeling of en route traffic [27], [28]. The main challenge in clustering this way is that the sequences are of different lengths. Li *et al.* [29], [30] use the density-based spatial clustering of applications with noise (DBSCAN) algorithm to cluster time-aligned trajectories and use the learned clusters for anomaly detection around airports. They also experiment with clustering the trajectories after performing principal component analysis. Hong and Lee [31] deal with the varying length issue by clustering with the dynamic time warping (DTW) algorithm to discover traffic patterns. They regress an aircraft's arrival time based on its similarity to the learned clusters and discuss how their method could augment terminal radar approach controllers. Conde Rocha Murca *et al.* [32] handle the varying length issue by resampling trajectories to make them the same length. They then cluster the resampled trajectories using the DBSCAN algorithm and construct classifiers to assign new flights to clusters. Their main application area is resource management. These three lines of research are the closest to ours.

B. Proposed Method

In this paper, we propose a method for learning probabilistic trajectory models of aircraft in terminal airspaces directly from position measurements. The model training procedure does not rely on other (likely unavailable) information, *e.g.*, velocity measurements, flight plans, or air traffic control instructions. Through an unsupervised clustering algorithm, the method is able to discover the departure and approach procedures for an airport, without ever accessing the airport's instrument procedures. The method then fits a generative model based on the intra-cluster covariance matrices.

The main limitations of the three most similar lines of work are: 1) they use ad-hoc methods for dealing with the varying length issue; 2) they do not use the learned clusters to construct a full probabilistic model of motion; and 3) the DTW and DBSCAN algorithms have trouble scaling to large datasets (our datasets are over 100 times larger).

Our method deals with the varying length issue, inspired by Li *et al.* [29], by time-aligning trajectories around the time they are closest to the runways and extrapolating the shorter trajectories so that they are all the same length. This alignment and extrapolation allows us to compute the similarity between trajectrories directly using the Euclidean norm. Using this similarity metric, we use the K-means clustering algorithm to cluster the trajectories. Our method then constructs a probabilistic model (a Gaussian mixture model) from the clusters, which leads to accurate inference and realistic generation of trajectories. The benefits of such an approach are that it is a compact model, trains quickly (which allows for retraining), efficient to sample from and perform inference with, and most importantly easy to understand and ultimately implement.

We find that our method performs well on real, noisy data, and is able to capture relevant statistical properties of the terminal airspace environment. We also find that trajectories sampled from the model are realistic. In Fig. 1, we show samples generated from an instance of our model trained on takeoffs at the KJFK airport next to held-out test trajectories. We evaluate our method throughout the paper on the KJFK airport, for illustration purposes, though applying the method to other airports is trivial. In fact, another major advantage of our method is that it is relatively automatic; it can produce a generative model from radar data with little to no supervision. Our method can create generative models for thousands of airports, automatically, from (poor, noisy, and intermittent) radar position data.

C. Limitations

Our method has several limitations that must be addressed. It is unable to incorporate the intent of the air traffic controller,

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

is unable to model the complex spatio-temporal interaction between aircraft, and can only fit models based on historical data (and thus cannot be used to test new instrument procedures).

D. Summary of Contributions

- A linear time trajectory reconstruction procedure given noisy, irregular position measurements.
- A procedure for automatic unsupervised learning of a Gaussian mixture model for aircraft motion that captures correlation between position dimensions over time.
- Publicly available code for constructing the model, running the experiments, and pre-trained models for the KJFK airport.[2]

II. Radar Data

Our method expects a set of aircraft trajectories denoted $\tau_1, \ldots, \tau_{N_{\text{traj}}}$. Each trajectory τ_i is represented by a set of time and position measurement ordered pairs, *i.e.*, $\tau = \{(t_i, \hat{p}_i)\}_{i=1}^{N_{\text{meas}}}$ where the time is $t_i \in \mathbf{Z}_+$ and the measurements are $\hat{p}_i \in \mathbf{R}^3$.

A. FAA Radar Dataset

The radar dataset we use in this paper is derived from the Federal Aviation Administration (FAA) multi–sensor fusion tracker [33], which fuses radar detections of a single aircraft by multiple sensors into a single track. The tracker uses primary radars, mode A/C SSRs, mode S SSRs, Wide Area Multilateration systems, and ADS–B receivers. Each of these surveillance sensor systems has different accuracies, updating rates (1–12 Hz), and susceptibility to anomalies. The data was preprocessed and fused into timestamped entries of target address (used to group entries into flights), position (measured in WGS84 latitude/longitude), pressure altitude (meters) and optionally geometric altitude, lateral velocity, and vertical velocity. We ignore lateral/vertical velocities in our analysis, as it is unclear whether they are derived from the position measurements or are actual velocity measurements. The data contains six months of flights starting March 2012 from Central Florida, New York City, and Southern California.

We describe the procedure for filtering the data that corresponds to a particular airport's terminal airspace, defined as the center of its runways $p_{\text{ref}} \in \mathbf{R}^3$ in earth–centered, earth–fixed (ECEF) coordinates. We convert every position measurement to east–north–up (ENU) coordinates determined by the origin p_{ref}. We only keep the position measurements that are less than 5NM in the "east" or "north" dimensions and less than 3000 feet in the "up" dimension, *i.e.*, in the terminal airspace of the airport. Then we iterate through measurements associated with each target address. Because target addresses are reused, we split sequences of measurements that have a gap in measurements of more than 30 s into separate trajectories. This filtering results in a set of trajectories specific to a given airport in the format expected by our method, where each trajectory is a set of time and aircraft position measurement pairs. We shift time so that the first measurement in each trajectory is always at $t = 0$.

[2]https://github.com/sisl/terminal-airspace-models

B. Landing and Takeoff Separation

We model landings and takeoffs separately, as these are inherently different modes of operation. There are many ways to classify a trajectory as a landing or takeoff; we used the following heuristic approach.

For each trajectory, let $T = \max_i t_i$ (the time of the last measurement in that trajectory), $\mathrm{c} = \operatorname{argmin}_i \|p_i\|_2$ (the index of the measurement closest to the center of the runways), $\dot{z}_{\text{avg}}$ be the average "up" velocity in ft/min, and R be approximately the length of the longest runway (2000 m for KJFK). The trajectory is then classified as a

- *landing* if $(\|p_{\mathrm{c}}\|_2 < R) \wedge (\|p_T\|_2 > R) \wedge (\dot{z}_{\text{avg}} < -200\,\text{ft/min} \wedge (t_c/T > 0.95)$
- *takeoff* if $(\|p_{\mathrm{c}}\|_2 < R) \wedge (\|p_0\|_2 > R) \wedge (\dot{z}_{\text{avg}} > 200\,\text{ft/min} \wedge (t_c/T < 0.05)$

We used this rule because it seemed to work well with the FAA radar dataset. Trajectories that do not match either of these rules are discarded.

For landings, we discard the position measurements after the index c, making the measurement closest to the origin as the last measurement. Similarly, for takeoffs, we ignore the position measurements before the index c. For landings, we reverse time in the trajectory by setting $t_i = T - t_i$, making landings and takeoffs both start near the origin. Reversing time for landings imposes the constraint that for both landings and takeoffs $t_i = 0$ is the point that the trajectory is closest to the runways. From now on, we assume that landings and takeoffs are processed separately.

C. Preprocessing

The only preprocessing we do is scale each dimension in each measurement by a separate positive scalar. Pre-scaling allows the dimensions to be compared equally when using ℓ_2 norms, and is basic standardization practice in machine learning. For aircraft trajectories, we scale the "up" dimension by 10. This is a heuristic, guided by the fact that glide slopes are approximately 3° to 5° ($1/\sin 5° \approx 11$) and by the ratio of the dimensions of the terminal airspace (5 NM/3000 feet ≈ 10). We reverse the scaling after we learn the model.

III. Model Learning

This section outlines the steps one performs to construct the model. The steps are to reconstruct the trajectories, cluster them, and then fit a generative model to the clusters.

A. Trajectory Reconstruction

There are several complications with the trajectories in their original format. First, the trajectories can have any number (including zero) measurements at a given time step. Second, the measurements are noisy. Third, we need to make the trajectories the same duration.

We tackle the third complication by enforcing that the trajectories are a common length, denoted T_{com} in units

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

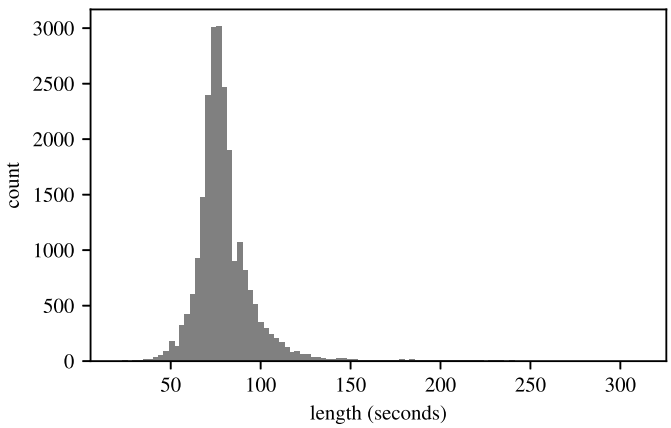


Fig. 2. Histogram of the length of takeoff trajectories at KJFK in seconds. The median is 70 seconds.

of seconds. We ignore trajectories that are significantly shorter than T_{com} seconds (*e.g.*, 30 seconds shorter). We extrapolate trajectories that are shorter than T_{com} and truncate trajectories that are longer than T_{com}.

Evidently, the choice of T_{com} will have an effect on the model learned, and should be optimized for the application at hand. For example, having a larger T_{com} leads to less accurate trajectories, as more trajectories are extrapolated, and a smaller dataset, as more trajectories are too short. For illustration, we set T_{com} to the median of the trajectory durations across the data (see Fig. 2 for the histogram of raw trajectory lengths at KJFK).

We interpolate, smooth, and extrapolate by solving the (least-squares) optimization problem

$$\underset{P}{\text{minimize}} \ \|AP - \hat{P}\|_F^2 + \lambda_1 \|D_2 P\|_F^2 + \lambda_2 \|D_3 P\|_F^2, \quad (1)$$

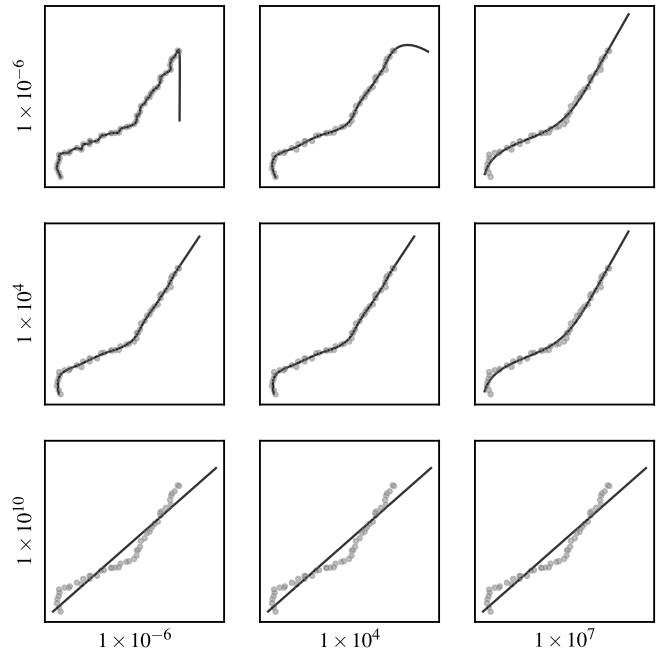
where the optimization variable $P \in \mathbf{R}^{N\times 3}$ is the reconstructed trajectory and its ith row is the position of the reconstructed trajectory at time i, the length $N = \max\{T, T_{\text{com}}\}$, the diagonal matrix $A \in \mathbf{R}^{N\times N}$ has A_{ii} equal to one if there are one or more measurements at time i and zero otherwise, each row in the measurement matrix $\hat{P} \in \mathbf{R}^{N\times 3}$ is the average of the measurements at that time and zero otherwise,[3] $D_2 \in \mathbf{R}^{N-2\times N}$ is the second-order difference matrix representing the acceleration operator, $D_3 \in \mathbf{R}^{N-4\times N}$ is the third-order difference matrix representing the jerk operator, $\lambda_1, \lambda_2 \in \mathbf{R}_+$ are regularization hyper-parameters, and $\|\cdot\|_F$ is the Frobenius norm of a matrix. We include the form of the (banded) matrices D_2 and D_3 for the sake of completeness

$$\begin{aligned}(D_2)_i &= e_i - 2e_{i+1} + e_{i+2}\\ (D_3)_i &= -e_{i-1} + 2e_i - 2e_{i+2} + e_{i+3},\end{aligned} \quad (2)$$

where $(A)_i$ denotes the ith row of A and e_i denotes the ith standard unit vector.

The first term of (1) encourages the reconstructed trajectory to be close to the measured points and the second and third terms encourage it to have low acceleration and jerk on average. The regularization parameters λ_1 and λ_2 are selected individually for each trajectory through out-of-sample validation.

[3]This formulation is equivalent to having an objective function where the ℓ_2 loss is averaged over each measurement per time step.



(a)

		λ_2		
		1×10^{-6}	1×10^{4}	1×10^{7}
λ_1	1×10^{-6}	1.842	1.318	2.590
	1×10^{4}	1.250	**1.247**	2.590
	1×10^{10}	5.239	5.239	5.239

(b)

Fig. 3. Trajectory reconstruction procedure. (a) Lateral view of measurements and reconstructed trajectory (including a 20 s extrapolation) for various settings of the regularization parameters. A low λ_1 and λ_2 leads to a jagged trajectory (top left) whereas high λ_1 or λ_2 leads to a linear trajectory (bottom right). (b) Out-of-sample validation losses for various regularization parameters. The optimal out-of-sample validation loss is bolded and achieved by $\lambda_1 = 1 \times 10^4$ and $\lambda_2 = 1 \times 10^4$.

We perform out-of-sample validation by randomly holding out measurements from that trajectory, fitting trajectories with varying λ_1 and λ_2 on the measurements that were not held out, and selecting the parameters that have the lowest loss (*i.e.*, (1) with $\lambda_1 = \lambda_2 = 0$) on the held-out measurements. Fig. 3 shows the reconstructed trajectories on the Pareto-optimal surface that were generated by varying λ_1 and λ_2, as well as the respective out-of-sample losses, for a simulated aircraft trajectory.

The formulation could potentially be improved by switching to a different loss, *e.g.*, Huber or ℓ_1, and by adding constraints on the reconstructed trajectory [34]. We stick to the form in (1) mainly for the sake of simplicity of implementation; solving (1) reduces to solving a 9–banded linear system, which is extremely fast (and linear in N). We believe that the trajectory reconstruction procedure described here could be useful for trajectory smoothing of a wide variety of vehicle data.

B. Clustering

After the trajectories are reconstructed, they all have the same length. Vectorizing a trajectory (stacking the columns of the matrix P) makes it a column vector $p^{(i)} \in \mathbf{R}^{3T_{\text{com}}}, i =$

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

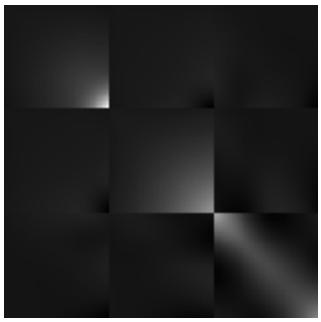


Fig. 4. Sample covariance matrix for the most probable cluster for takeoffs at KJFK. Higher values of covariance are a lighter shade. The first third of the matrix corresponds to the east dimension, the second third corresponds to north dimension over time, and the final third corresponds to up dimension over time. Uncertainty increases with time, as aircraft are more likely to deviate from their nominal path later in the takeoff maneuver.

$1, \ldots, N_{\text{traj}}$. The first, second, and third parts of the vector are the east, north, and up positions over time.

Since aircraft in terminal airspace, for the most part, follow pre-defined procedures and only slightly deviate from them, it is reasonable to believe that the trajectories partition well into clusters. Each cluster should have trajectories that follow a similar path. One way we can measure the similarity between two trajectories $p^{(1)}$ and $p^{(2)}$ is using their Euclidean distance $\|p^{(1)} - p^{(2)}\|_2$. (One could use other distance metrics, but the Euclidean distance works well with the Gaussian mixture model we eventually fit.) Given a number of groups, denoted K, we want to choose K centers μ_j and assignments $c_i \in \{1, \ldots, K\}$ to each trajectory to minimize the total distance between each point and its assigned cluster center. We do this using the K-means++ algorithm [35], [36].

The K-means procedure results in K *archetypal trajectories* μ_j and N_{traj} assignments c_i. For each cluster, we calculate the intra-cluster empirical covariance matrices $Q_j \in \mathbf{S}_{++}^{3T_{\text{com}}}$. Fig. 4 provides a visualization of one of the intra-cluster covariance matrices. We let the vector π denote cluster frequencies, where π_j is the fraction of trajectories in cluster j.

C. Gaussian Mixture Model

Our intra-cluster sample covariance matrices are noisy. We would like to remove as much of this noise as possible, so we truncate the singular values of the empirical covariance matrix Q_j. (We note that there are more sophisticated methods, *e.g.*, [37].) The approximation is given by $\tilde{Q}_j = U_j \Sigma_j U_j^T$, where U_j is a matrix with the first r singular vectors, and Σ_j is a diagonal matrix with the first r singular values. The columns of U_j can be interpreted as the *principal deviations* for motion in cluster j. (Note that if we save just the principal deviations instead of Q_j it significantly reduces the overall size of the trained model and speeds up sampling and inference.) As a final step, we perform the inverse of the preprocessing in Section II, which involves dividing the "up" dimension by 10 and reversing time for landings.

We keep the top $r = 5$ principal deviations for aircraft at KJFK because it captures most of the intra-cluster variance, though in practice r should be chosen using the procedure in Section III-D. Fig. 5 provides a lateral view of the top five principal deviations for the most probable cluster for takeoffs at KJFK.

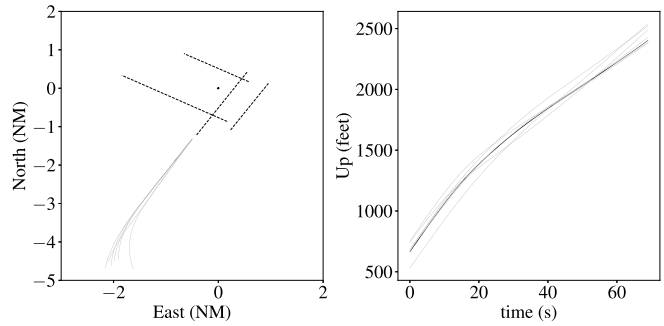


Fig. 5. Archetypal trajectory (black) and top-5 principal deviations (gray) for the most probable cluster of the model for takeoffs at KJFK with $K = 20$.

We now can construct a generative probabilistic model. Our model for aircraft trajectories in terminal airspaces is the following Gaussian mixture model (GMM):

$$\begin{aligned} j &\sim \text{Categorical}(\pi) \\ p &\sim \mathcal{N}(\mu_j, \tilde{Q}_j). \end{aligned} \tag{3}$$

It is a well-known fact that conditioning a Gaussian distribution on measurements gives another Gaussian distribution. This makes GMMs easy to sample from and perform inference with.

This fact leads to a convenient interpretation of the assumed trajectory distribution. Because the marginal distribution of a Gaussian is another Gaussian, the first measurement of the trajectory comes from a Gaussian distribution. Then the position at time t conditioned on the previously sampled points similarly follows a Gaussian distribution. Drawing a full trajectory from the model is equivalent to successively drawing from Gaussian distributions dependent on the past.

1) Positive Definiteness of Covariance Matrices: Because of the low-rank approximation step, our covariance matrices are now positive semidefinite and no longer positive definite, which is a requirement to have a valid multivariate density. We can deal with this inconsistency by restricting the density to a $\mathbf{rank}(Q_j)$ subspace where the Gaussian distribution is supported, which results in the following density:

$$f(x) = (\det{}^*(2\pi Q_j))^{-\frac{1}{2}} \exp(-\frac{1}{2}(x-\mu_j)^T Q_j^{\dagger}(x-\mu_j)), \tag{4}$$

where $Q_j^{\dagger}$ is the Moore-Penrose inverse of Q_j and $\det^*$ is the pseudo-determinant, or the product of the non-zero singular values [38].

2) Sampling: Similarly, to sample from cluster j we sample $z \sim \mathcal{N}(0, I_r)$ and then emit $\mu_j + U_j \Sigma_j^{1/2} z$. The sample is just an archetypal trajectory plus a random linear combination of the principal deviations weighted by the square root of the singular values, hence the name principal deviations. Fig. 6 shows a cluster center and samples from that cluster's distribution for the most probable cluster at KJFK.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

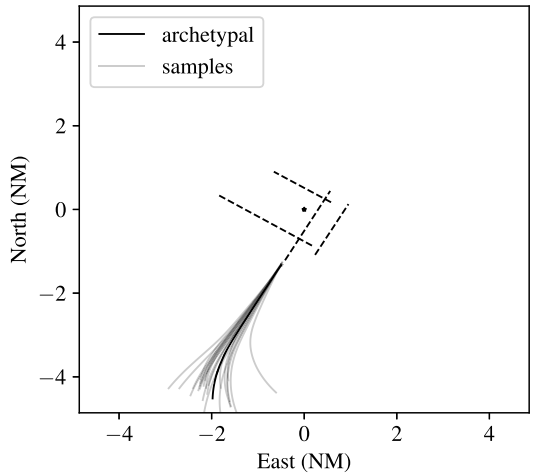


Fig. 6. Samples from the most probable cluster takeoff model at KJFK with $K = 20$. The cluster covariance naturally captures plausible takeoffs at KJFK.

D. Automatic Model Generation

We now discuss the claim, made in Section I, that the method can go from radar data to generative model with little to no supervision. The trajectory reconstruction procedure (Section III-A) can be performed independently on each trajectory and requires no parameters. The clustering procedure (Section III-B) requires only one parameter, K. The generative model fitting (Section III-C) also requires only one parameter, r. We can select K and r, automatically, using an out-of-sample validation procedure that depends on the downstream task. First, we split the dataset into a training and held-out set. We fit generative models with varying values of K and r, calculate the performance measure (which depends on the downstream task) of the generative model on the held-out data, and then use the model that achieves the highest performance on the held-out set. Section IV demonstrates this procedure for the tasks of generation and inference.

IV. Experiments

We evaluate our method on takeoffs and landings at the KJFK airport. Details on the layout of KJFK are given in the Appendixes. The plots in the paper are for the KJFK airport with $K = 50$ clusters for takeoffs and $K = 70$ clusters for landings, as those seemed to be reasonable choices for the number of clusters for the purpose of illustration. In practice, K should be chosen using the procedure in Section III-D.

A. Archetypal Trajectories

We first sanity check the model by visualizing the learned centers. Fig. 7 shows the learned centers for landings at KJFK and Fig. 8 shows the centers for takeoffs at KJFK. The centers seem to capture the plausible ways of approaching and departing KJFK. Overall, the glide slope for landings seems to be of constant slope and the climb rates for takeoffs seem to start high and then level off. (The different phases for takeoffs likely correspond to the form of the takeoff procedures at KJFK.) Several of the archetypal trajectories look super-imposed, which seems redundant. But in fact,

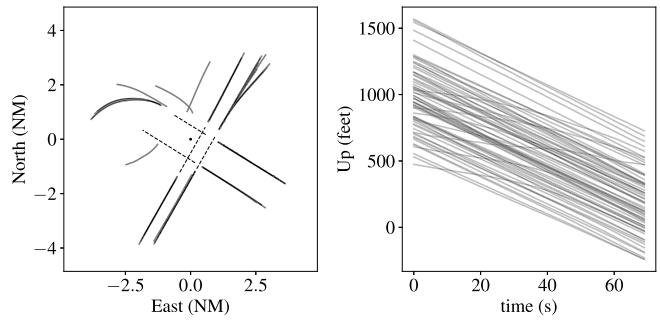


Fig. 7. Model for landings at KJFK with $K = 70$. Left, lateral view of archetypal trajectories. Right, archetypal trajectories vertical profile. Many of the archetypal trajectories overlap in lateral view but not in vertical profile. Each center corresponds to a slightly different glide slope.

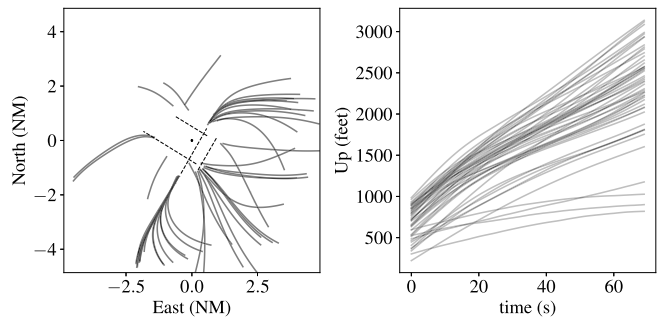


Fig. 8. Model for takeoffs at KJFK with $K = 50$. Left, archetypal trajectories lateral view. Right, archetypal trajectories vertical profile. As with landings, each cluster corresponds to different climb rates.

different archetypes can capture not only the procedures, but also approaches at different glide slopes, airspeeds, and wind conditions. For example, we might end up with clusters for smaller and larger aircraft because they have different characteristics. These nuances are not coded into the procedure, but rather learned directly from the data.

One observation worth discussing is that there seem to be clusters that represent invalid approaches/departures, *e.g.*, a runway does not exist there or clusters have a negative/constant vertical profile. These invalid clusters correspond to a small π_j, and thus low probability, under the model. That there are clusters representing invalid maneuvers likely means that there are anomalous trajectories in the data. These anomalous trajectories could be filtered using outlier detection, but that is out of the scope of this paper.

B. Statistical Properties

To evaluate the statistical properties of the model, we randomly split the trajectories into a training set and heldout set, using a 75-25 training-test split. Then, we trained the model on the training set and evaluated its performance on the held-out set. To evaluate whether or not the model is able to capture the overall statistics of vehicle motion in the environment, we devised several visual and quantitative experiments and tested the model over different values of K.

We have a set of held-out test samples and independently sampled trajectories from the model. We first plot the overall histogram of positions, with a logarithmic color scale, for generated landings at KJFK (Fig. 9), held-out landings at

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

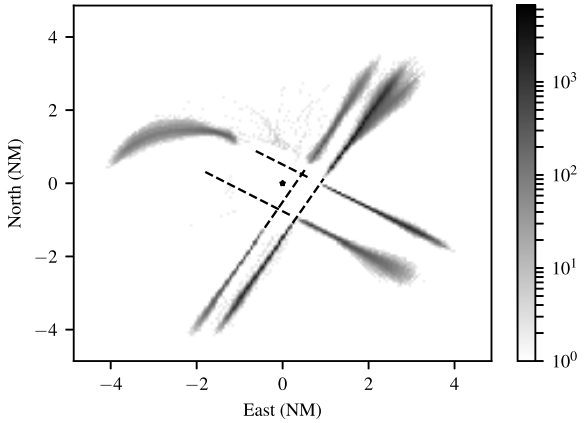


Fig. 9. Lateral view of log-histogram of **generated** landing flight tracks. Compare with Fig. 10.

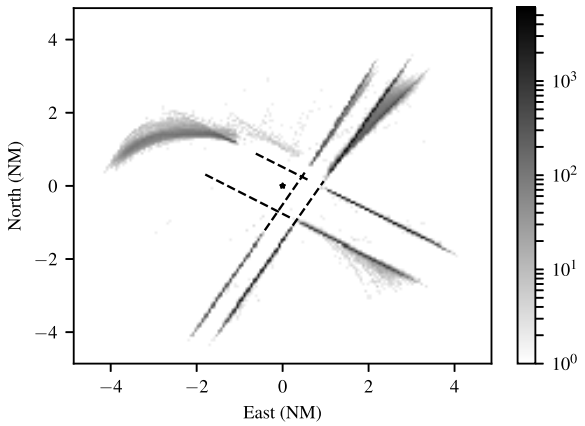


Fig. 10. Lateral view of log-histogram of **real held-out** landing flight tracks. Compare with Fig. 9.

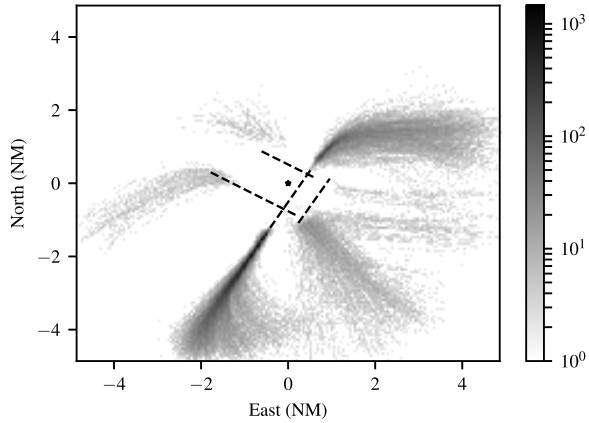


Fig. 11. Lateral view of log-histogram of **generated** takeoff flight tracks. Compare with Fig. 12.

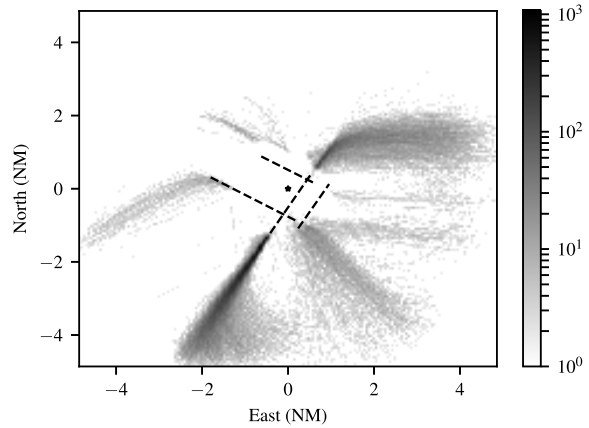


Fig. 12. Lateral view of log-histogram of **real held-out** takeoff flight tracks. Compare with Fig. 11.

KJFK (Fig. 10), generated takeoffs at KJFK (Fig. 11) and held-out takeoffs at KJFK (Fig. 12). The $5 \times 5\text{NM}^2$ grid is split into 400×400 bins making each bin $23.15 \times 23.15\,\text{m}^2$.

We also would like to make sure that the model captures realistic physical properties of the trajectories outside of just positions, so we calculate the sample longitudinal velocity, vertical velocity, and turn rate for generated samples and compare that to those in held-out data. Fig. 15 in the Appendix shows the generated/held-out turn-rate distributions for takeoffs at KJFK. (The figures for vertical velocity and turn rate were omitted for the sake of space.)

Next we performed the out-of-sample validation procedure described in Section III-D for two tasks: generation and prediction. To evaluate generation, we calculate the Kullback-Leibler divergence (KL-divergence) between the (smoothed) empirical distributions of position, longitudinal velocity, vertical velocity, and turn rate,

$$D_{\text{KL}}(P \| Q) = -\sum_i P(i) \log \frac{Q(i)}{P(i)}. \tag{5}$$

We then average the KL-divergence across the four distributions to get an overall measure of distributional similarity.

To evaluate prediction, we give the model the first 10 measurements of every held-out trajectory, calculate the closest cluster, and calculate the posterior mean conditioned on those 10 measurements. We then calculate the root-mean-square (RMS) error between the posterior mean (the prediction) and the actual rest of the trajectory. Fig. 13 shows these two metrics, generation and inference, for varying values of K at KJFK. (The parameter r was set to 5 in all experiments.)

For KJFK, from the out-of-sample validation procedure, we can conclude that the optimal number of clusters K for prediction for takeoffs is 10 (RMS = 322.22) and for landings is 20 (RMS = 106.86). The inflated RMS value for takeoffs is likely because it is challenging to predict where an aircraft will go given just the direction it took off in, but for landings the model is able to identify the procedure from only 10 measurements. Increasing K, in the case of prediction, leads to *overfitting*, where the generative model fits the training data too well and fails to generalize to held-out data. However, for generation, increasing K does not lead to overfitting. This result is likely because different clusters may represent, *e.g.*, different aircraft types and wind conditions, which are not coded into the airport procedures. These clusters are not important for RMS-based prediction, but are important for accurately capturing the statistical properties of the terminal airspace environment.

C. Pilot Turing Test

In Fig. 1 we showed samples from the model and held-out test samples side-by-side and asked the reader to guess which

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

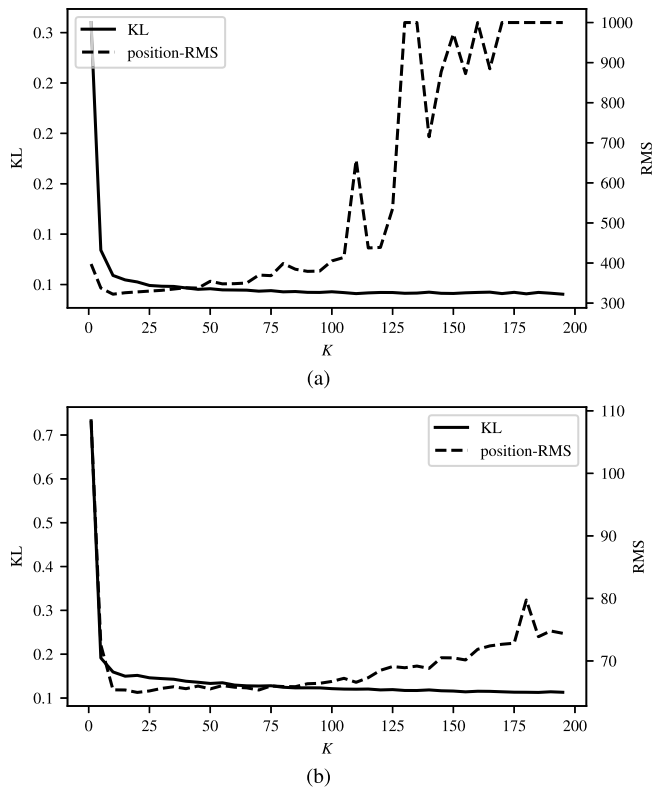


Fig. 13. Model for KJFK takeoffs. Log-plot of average KL-divergence and prediction RMS error vs. number of clusters (lower is better in both cases). The RMS error was clipped at 1000 m for visualization purposes. (a) takeoffs, optimal KL is 0.002 and the optimal RMS is 322.22 m ($K = 10$); (b) landings, optimal KL is 0.005 and the optimal RMS is 106.86 m ($K = 20$).

one was generated and which one was real. Throughout the experiments, we found the generated samples to be realistic. To evaluate how realistic the samples are, we devised what we call a "Pilot Turing Test" (PTT), akin to the Turing Test in Artificial Intelligence [39].

To perform a PTT, we first take 25 random samples generated from the model and 25 random held-out samples. We added a small amount of random noise to the real and synthetic samples to obscure radar anomalies. Then, we give the pilot 25 pairs of generated and real trajectories in random order, where each plot has a lateral view of the trajectory and its vertical profile, as in Fig. 8. We ask the pilot to label one trajectory in each pair as real and one as generated and calculate the accuracy of their predictions. We want the accuracy to be around 50%, which indicates that the generated and real trajectories are indistinguishable.

We conducted a PTT with two licensed pilots for landings/takeoffs at KJFK. The first pilot (Pilot 1) is a professional Flight Test Captain for a U.S. airline. The second pilot (Pilot 2) is a professional Boeing 747 pilot with over 13 000 hours of total flying experience. The resulting accuracies for both pilots can be found in Table I. The files used to conduct the PTT can be found in the code. On average, the pilots achieved an accuracy of 54%, which means the pilots performed slightly better than random guessing. We can conclude that our model is able to generate trajectories that are, in practice, visually indistinguishable from real trajectories.

TABLE I
PILOT TURING TEST ACCURACIES

Pilot 1		Pilot 2	
Landings	Takeoffs	Landings	Takeoffs
52 %	52 %	44 %	68 %

V. APPLICATIONS

This section describes how the generative model can be used to generate trajectories and perform inference. The model can also be adapted for other applications, including anomaly detection, using the clusters to adjust procedures, and reactionary motion planning.

A. Generation

As described in Section III, we can draw samples from the assumed probability distribution over aircraft motion. We draw samples by first sampling from the categorical distribution over clusters, sampling the vector $z \sim \mathcal{N}(0, I_r)$, and then emitting the trajectory $\mu_j + U_j \Sigma_j^{1/2} z$. If we further condition the categorical distribution over clusters on external factors such as time of day or previously sampled clusters, we can construct 3D simulations of aircraft operating in particular terminal airspaces. We could even integrate it into an airport queuing model [40] to construct even more realistic simulations.

The ability to generate samples could be particularly useful for evaluating decision support systems. For example, the model could be used for verification of AutoATC, an automated air traffic control system for nontowered airports [41]. The model would provide 3D simulations of the terminal airspace, feed the resulting position traces into a simulation of AutoATC, and identify failure cases where AutoATC failed to provide the correct guidance to avoid a collision. It could also be used to perform ablation studies, *e.g.*, by increasing the "pressure" on the airport by increasing the number of clusters sampled per time period or scaling the covariance matrices in the clusters to emit more erratic trajectories.

B. Inference

As described in Section III, we can condition each cluster's Gaussian distribution on partial measurements. By performing Bayesian inference in the Gaussian mixture model, we can calculate a probability distribution over clusters and time index in that cluster's distribution. For each of these possibilities, we can also calculate the posterior distribution over future and past measurements by conditioning that cluster's multivariate Gaussian on the partial measurements. We demonstrate the result of this procedure for an approach at KJFK in Fig. 14.

Being able to perform inference directly leads to several applications. If we had, for example, 10 radar measurements of an aircraft approaching an airport, we can immediately provide non-trivial answers to useful questions (we provide the answers for the inference that was performed in Fig. 14):

1) Which runway will the aircraft land on? Runway 13L.
2) How long until the aircraft lands? ≈ 60 s.
3) Which approach procedure is the aircraft performing? PARKWAY VISUAL RWY 13L/R.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

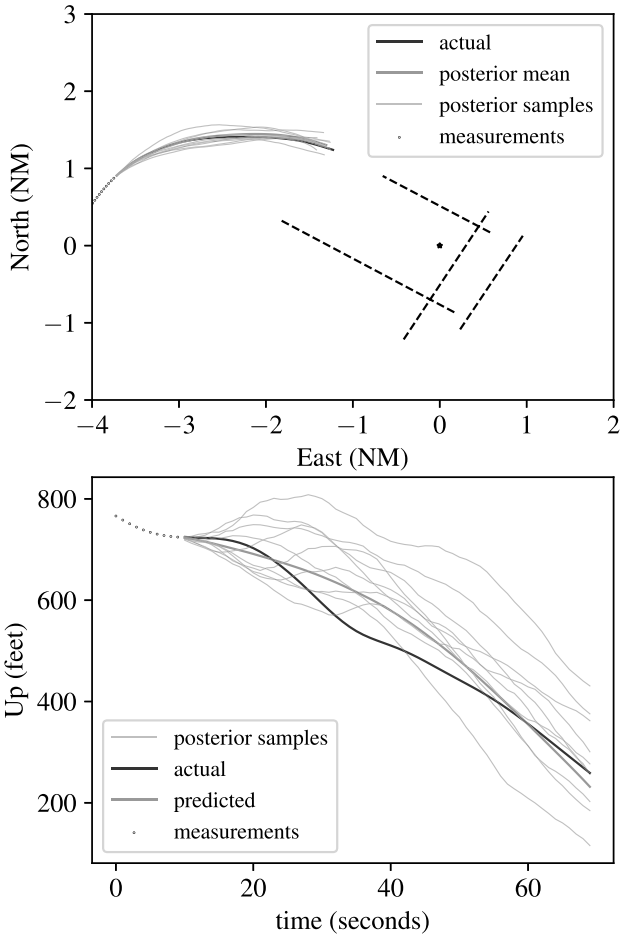


Fig. 14. Demonstration of inference for a KJFK landing model with $K = 20$. Given 10 samples (dots) of a landing trajectory (black), the model produces a prediction of where the aircraft will go (dark gray) and samples from the posterior distribution (light gray). The average RMS error for the prediction is 77.41 m.

If they were departing, we could answer a similar set of questions:

1) How long until the aircraft exits the terminal airspace?
2) Which direction will the aircraft exit the terminal airspace?
3) Which departure procedure is the aircraft performing?

If we had the posterior distribution for two aircraft, we could perform a Monte Carlo simulation to calculate a conflict probability by sampling a large number of possible trajectories from both posterior distributions and calculating the proportion that violated some conflict criteria. This posterior distribution could also be used as a model of other aircraft in an on-board planning system for collision avoidance, or as input to an automated air traffic control system.

VI. Conclusion

In this paper, we proposed a method for learning probabilistic motion models for aircraft in terminal airspaces directly from position measurements. Our method could be extended to a larger terminal airspace, *e.g.*, < 50NM from the airport, and be used to aid terminal radar approach controllers as in Hong and Lee [31]. We also believe that a slightly altered version of our method could be applied to vehicles in other contexts. It is likely to work well for highly structured environments, *e.g.*, loading dock for trucks, parking lots, intersections with stoplights/stop signs, or shipping ports. On the other hand, the method will not perform as well in unstructured environments, such as interstate freeways and in unstructured airspace.

The main limitation and hence direction for future work is to incorporate pairwise interaction. We can assume that aircraft actively avoid each other, which affects the probability distribution over future motion. We believe that modeling pairwise interaction can be achieved with a framework similar to ours.

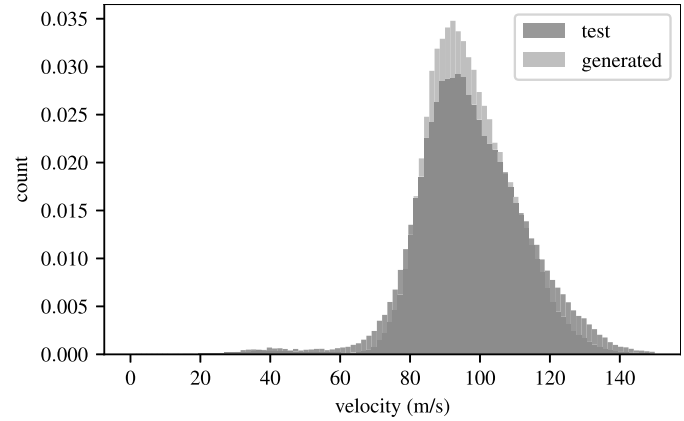


Fig. 15. Model for takeoffs at KJFK with $K = 50$. Histogram of lateral velocities for generated and held-out test data.

Appendix A
KJFK Airport

The John F Kennedy International Airport (KJFK) airport is a towered airport in Queens, New York with four runways. As of 2016, KJFK has 1256 aircraft operations per day, 91% commercial, 7% air taxi, 2% transient general aviation and less than 1% military [42].

Acknowledgments

The authors acknowledge the FAA and the MITRE Corporation for sharing the surveillance data. They also thank E. Maki for the thoughtful discussions, and R. Abraham and G. Knox for their participation in the Pilot Turing Test.

References

[1] (2018). *Modernization of U.S. Airspace*. [Online]. Available: https://www.faa.gov/nextgen
[2] "Statistical summary of commercial jet airplane accidents," Boeing, Chicago, IL, USA, Tech. Rep., 2016.
[3] M. Enriquez, "Identifying temporally persistent flows in the terminal airspace via spectral clustering," in *Proc. USA/Eur. Air Traffic Manage. Res. Develop. Seminar*, 2013.
[4] D. Vasquez and T. Fraichard, "Motion prediction for moving objects: A statistical approach," in *Proc. Intl. Conf. Robot. Autom.*, vol. 4, Apr./May 2004, pp. 3931–3936.
[5] G. Chatterji, B. Sridhar, and K. Bilimoria, "En-route flight trajectory prediction for conflict avoidance and traffic management," in *Proc. AIAA Guid., Navigat., Control Conf.*, Jul. 1996, p. 3766.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

[6] G. Chatterji, "Short-term trajectory prediction methods," in *Proc. AIAA Guid., Navigat., Control Conf.*, 1999, p. 4233.

[7] R. Slattery and Y. Zhao, "Trajectory synthesis for air traffic automation," *J. Guid., Control Dyn.*, vol. 20, no. 2, pp. 232–238, 1997.

[8] A. W. Warren and Y. S. Ebrahimi, "Vertical path trajectory prediction for next generation ATM," in *Proc. 17th Digit. Avionics Syst. Conf.*, vol. 2, Oct./Nov. 1998, p. F11–1.

[9] M. Kochenderfer, J. Kuchar, L. Espindle, and J. Griffith, "Uncorrelated encounter model of the national airspace system," Massachusetts Inst. Technol., Lincoln Lab., Lexington, MA, USA, Project Rep. ATC-345, 2008.

[10] M. J. Kochenderfer, M. W. M. Edwards, L. P. Espindle, J. K. Kuchar, and J. D. Griffith, "Airspace encounter models for estimating collision risk," *J. Guid., Control, Dyn.*, vol. 33, no. 2, pp. 487–499, 2010.

[11] R. A. Paielli and H. Erzberger, "Conflict probability estimation for free flight," *J. Guid., Control Dyn.*, vol. 20, no. 3, pp. 588–596, Oct. 1997.

[12] C. E. van Daalen and T. Jones, "Fast conflict detection using probability flow," *Automatica*, vol. 45, no. 8, pp. 1903–1909, Aug. 2009.

[13] L. J. Pienaar and T. Jones, "The application of probability flow for conflict detection near airports," in *Proc. AIAA Guid., Navigat., Control Conf.*, Jan. 2015, p. 1325.

[14] J. Chryssanthacopoulos, M. Kochenderfer, and R. Williams, "Improved Monte Carlo sampling for conflict probability estimation," in *Proc. AIAA Struct., Struct. Dyn., Mater. Conf.*, Apr. 2010, p. 3012.

[15] R. Coppenbarger, "Climb trajectory prediction enhancement using airline flight-planning information," in *Proc. AIAA Guid., Navigat., Control Conf.*, 1999, p. 170.

[16] W. Chan, R. Bach, and J. Walton, "Improving and validating CTAS performance models," in *Proc. AIAA Guid., Navigat., Control Conf.*, Aug. 2000, p. 4476.

[17] FAA, "Minimum operational performance standards for traffic alert and collision avoidance system II (TCAS II)," RTCA, Washington, DC, USA, Tech. Rep. RTCA DO-185B, Jun. 2008.

[18] M. Kochenderfer, J. Holland, and J. Chryssanthacopoulos, "Next generation airborne collision avoidance system," *Lincoln Lab. J.*, vol. 19, no. 1, pp. 17–33, 2012.

[19] E. Harkleroad, A. Vela, J. Kuchar, B. Barnett, and R. Merchant-Bennett, "Risk-based modeling to support nextgen concept assessment and validation," Lincoln Lab, Lexington, MA, USA, Tech. Rep. ATC-405, 2013.

[20] 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.

[21] J. L. Yepes, I. Hwang, and M. Rotea, "New algorithms for aircraft intent inference and trajectory prediction," *J. Guid., Control, Dyn.*, vol. 30, no. 2, pp. 370–382, 2007.

[22] I. Hwang and C. E. Seah, "Intent-based probabilistic conflict detection for the next generation air transportation system," *Proc. IEEE*, vol. 96, no. 12, pp. 2040–2059, Dec. 2008.

[23] 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, 2011.

[24] C. Lowe and J. How, "Learning and predicting pilot behavior in uncontrolled airspace," in *Proc. AIAA Infotech Aerosp.*, 2015, p. 1199.

[25] M. Gariel, A. N. Srivastava, and E. Feron, "Trajectory clustering and an application to airspace monitoring," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1511–1524, Dec. 2011.

[26] Z. Mahboubi and M. Kochenderfer, "Learning traffic patterns at small airports from flight tracks," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 4, pp. 917–926, Apr. 2017.

[27] M. Gariel, "Toward a graceful degradation of air traffic management systems," Ph.D. dissertation, Georgia Inst. Technol., Atlanta, GA, USA, 2010.

[28] A. Marzuoli, M. Gariel, A. Vela, and E. Feron, "Data-based modeling and optimization of en route traffic," *J. Guid., Control, Dyn.*, vol. 37, no. 6, pp. 1930–1945, 2014.

[29] L. Li, M. Gariel, R. J. Hansman, and R. Palacios, "Anomaly detection in onboard-recorded flight data using cluster analysis," in *Proc. IEEE/AIAA 30th Digit. Avionics Syst. Conf.*, Seattle, WA, USA, Oct. 2011, pp. 4A4-1–4A4-11.

[30] L. Li, J. Hansman, R. Palacios, and R. Welsch, "Anomaly detection via a Gaussian mixture model for flight operation and safety monitoring," *Transp. Res. C, Emerg. Technol.*, vol. 64, pp. 45–57, Mar. 2016.

[31] S. Hong and K. Lee, "Trajectory prediction for vectored area navigation arrivals," *J. Aerosp. Inf. Syst.*, vol. 12, no. 7, pp. 490–502, 2015.

[32] M. C. R. Murca, R. DeLaura, R. J. Hansman, R. Jordan, T. Reynolds, and H. Balakrishnan, "Trajectory clustering and classification for characterization of air traffic flows," in *Proc. 16th AIAA Aviation Technol., Integr., Oper. Conf.*, 2016, p. 3760.

[33] C. Jagodnik, J. Stella, and D. Varon, "Fusion tracking in air traffic control," *J. Air Traffic Control*, vol. 50, no. 1, pp. 15–21, May 2008.

[34] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[35] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, pp. 129–137, Mar. 1982.

[36] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proc. ACM-SIAM Symp. Discrete Algorithms*, Jan. 2007, pp. 1027–1035.

[37] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, Jul. 2008.

[38] C. Rao, *Linear Statistical Inference and Its Applications*, vol. 2. New York, NY, USA: Wiley, 1973.

[39] A. Turing, "Intelligent machinery, a heretical theory," *Philos. Math.*, vol. 4, no. 3, pp. 256–260, 1996.

[40] H. Idris, J.-P. Clarke, R. Bhuva, and L. Kang, "Queuing model for taxi-out time estimation," *Air Traffic Control Quart.*, vol. 10, no. 1, pp. 1–22, 2002.

[41] Z. Mahboubi and M. J. Kochenderfer, "Autonomous air traffic control for non-towered airports," in *Proc. USA/Eur. Air Traffic Manage. Res. Develop. Seminar*, 2015, pp. 1–6.

[42] (2018). *AirNav: KJFK—John F Kennedy Int. Airport*. [Online]. Available: http://www.airnav.com/airport/KJFK

Shane T. Barratt received the B.S. degree in electrical engineering and computer science from the University of California at Berkeley in 2017. He is currently pursuing the Ph.D. degree in electrical engineering with Stanford University.

Mykel J. Kochenderfer received the B.S. and M.S. degrees in computer science from Stanford University in 2003 and the Ph.D. degree from the University of Edinburgh in 2006. He was with MIT Lincoln Laboratory, where he worked on airspace modeling and aircraft collision avoidance. Since 2013, he has been an Assistant Professor of aeronautics and astronautics at Stanford University.

Stephen P. Boyd received the B.A. degree in mathematics from Harvard University in 1980, and the Ph.D. degree in electrical engineering and computer science from the University of California at Berkeley in 1985. He then joined the faculty at Stanford University. He is currently a Samsung Professor of engineering and a Professor of electrical engineering with the Information Systems Laboratory, Stanford University, with courtesy appointments in computer science and management science and engineering.