Probabilistic Modeling of Aircraft Trajectories for Dynamic Separation Volumes

Timothy A. Lewis*

NASA Langley Research Center, Hampton, VA 23681

With a proliferation of new and unconventional vehicles and operations expected in the future, the *ab initio* airspace design will require new approaches to trajectory prediction for separation assurance and other air traffic management functions. This paper presents an approach to probabilistic modeling of the trajectory of an aircraft when its intent is unknown. The approach uses a set of feature functions to constrain a maximum entropy probability distribution based on a set of observed aircraft trajectories. This model can be used to sample new aircraft trajectories to form an ensemble reflecting the variability in an aircraft's intent. The model learning process ensures that the variability in this ensemble reflects the behavior observed in the original data set. Computational examples are presented.

Nomenclature

$E[\cdot]$	expectation	$f(\cdot)$	true probability distribution
Φ	set of feature functions	i	feature function index
$H[\cdot]$	histogram	k	time index
L	number of feature functions	λ	Lagrange multiplier
Λ	set of Lagrange multipliers	μ	sample mean approximation
Ω	support set of trajectories	$p(\cdot)$	maximum entropy probability distribution
$\mathbf{R}$	set of real numbers	r	yaw rate [deg/s]
R^{obs}	set of observed trajectories	$\mathbf{r}$	yaw rate trajectory vector
R^{syn}	set of synthetic trajectories	t	time [s]
S	Shannon entropy	$\mathbf{u}$	position of airspace hazard
T	number of time steps	$\mathbf{v}$	velocity [kt]
d	separation distance from airspace hazard	$\mathbf{x}$	position [nmi]
$\phi(\cdot)$	feature function	ψ	yaw angle [deg]

I. Introduction

The ability to accurately predict the trajectory of an aircraft is the cornerstone of many air traffic management (ATM) functions. Trajectory prediction enables functions such as the detection of traffic conflicts between aircraft, which is critical for separation assurance, as well as the calculation of the estimated time of arrival of an aircraft at a given position, important for schedule and flow management.

Trajectory prediction (also called trajectory generation) is at the core of most ATM automation systems under development, and much work has been devoted to the problem of accurate trajectory prediction in this context.[1–3] The standard approach to trajectory prediction in ATM is constructed on the basis of a nominal trajectory plus some method to account for uncertainty due to factors such as wind[4] and variations in vehicle performance.[5]

The nominal trajectory is based on a notion of the ideal trajectory intent of an aircraft in the absence of any uncertainty. In today's operations, this intent is derived from the aircraft's flight plan or current clearance and usually takes the form of a series of speeds, headings, and altitudes to be flown in sequence. A trajectory prediction algorithm may employ this intent information in a simplified kinematic motion model,[6] or in a physics-based aircraft performance model such as the Base of Aircraft Data.[7]

Trajectory prediction will be even more important to the many new and unconventional uses of the airspace that are expected in the coming years. Of immediate concern is proliferation of unmanned aircraft systems (UAS) in

*Aerospace Engineer, Crew Systems and Aviation Operations Branch, M/S 152, AIAA member.

the national airspace in the next decade. A huge variety of missions have been proposed for these vehicles, ranging from environmental monitoring, infrastructure inspection, search and rescue, and package delivery. Looking further, advances in UAS technology may enable a revolution in personal air vehicles (PAVs) operating in an on-demand, point-to-point fashion. In both the UAS and PAV applications, we can envision large numbers of vehicles routinely operating at closer proximity to each other, as well as to buildings and people on the ground, than is acceptable in civil aviation today.

If we take an *ab initio*[8] approach to the design of an airspace system to accommodate such density and diversity of new vehicles and missions, we are faced with the issue that the trajectory prediction and separation assurance approaches to managing today's traffic will not be sufficient in this new world. This paper attempts to address two factors here.

First, we can conceive of many circumstances where an aircraft's trajectory intent is unavailable or unreliable. In such cases, it will be necessary to incorporate the uncertainty of intent into the trajectory prediction. With a probabilistic model of an aircraft's intent, a trajectory prediction algorithm can make more accurate predictions of the range of likely future positions of that aircraft, with applications to separation assurance, airspace complexity management, and other important functions.

Second, with the diversity of aircraft that may be expected in the future, it may not be practical or feasible to build such a model by hand for each aircraft type in each operational scenario. Instead, we propose a data-driven approach to learning a probabilistic model of intent behavior from a set of observed trajectories. Such a learning process could be conducted off-line from a database of trajectories to build an initial model, and then updated in real time based on observations made in flight. Such an approach could be conducted automatically in an unsupervised fashion.

With these goals in mind, this paper describes a preliminary approach to probabilistic trajectory prediction with uncertain intent. The paper describes a modeling process employing a set of feature functions to constrain a maximum entropy probability distribution based on a set of observed trajectories. New trajectories are then sampled from this model in a series of experiments to demonstrate the ability to reproduce the observed aircraft behavior. In future work, this model is intended for application to derive a dynamic separation volume, yielding the optimal separation distance between two aircraft in a given encounter.

II. Background

The proposed approach to trajectory modeling is a generative model. A *generative* model in probability and statistics is one that allows for the generation of sample observations from a distribution of interest. Compare with a *discriminative* model, which allows for the classification between given samples. For instance, in an image processing context, a discriminative model may allow for the classification of a set of images into different categories, e.g., cats, dogs, trees, apples, etc. Whereas, a generative model may allow the generation of new images with the characteristics of a given category, e.g., the generation of new and previously unobserved images in the category of human faces.[9] This generative process is sometimes referred to as "hallucination" or "dreaming," since the computer algorithm synthesizes new and recognizable images that were not part of the training set.[10]

Generative models have unique advantages in problems of prediction. In the problem of air traffic trajectory prediction, we can draw sample trajectories from such a model and use them as the basis for probabilistic conflict detection and other functions. In this Monte Carlo approach to trajectory prediction, the samples form an ensemble of predictions representing the variability of the future trajectory of an aircraft. For instance, by comparing the ensembles between two aircraft, we can obtain a distribution of their closest point of approach, and derive the probability that this distance will be below a certain allowable separation standard. If this probability is above a given threshold, the aircraft can be alerted to initiate appropriate action.

Such an application will only be as successful as the goodness of the model generating the ensemble. The development of techniques to build useful and accurate generative models for complex phenomena is an ongoing area of research in many fields. This paper is inspired by computer vision work in texture modeling and related problems. In particular, the Filters, Random Fields, and Maximum Entropy method by Zhu et al.[11] applied a set of feature functions to constrain a maximum entropy probability distribution, leading to a model which reproduced the characteristics of texture patterns from natural images. The feature-constrained maximum entropy approach has been used in number of other applications.[12–16]

The basic process of feature-constrained maximum entropy model learning is employed here, with application to the yaw rate trajectory of an aircraft. By applying this modeling approach to aircraft trajectories, we seek to produce a model which generates new trajectories with the same statistical characteristics of the observation data set. The present application is restricted to the 2D, constant speed motion of an aircraft, but future work will expand to full trajectories.

III. Methodology

A. Aircraft trajectory representation

Consider the 2D, horizontal trajectory of an aircraft traveling at constant speed v in an inertial frame with no wind. Under these assumptions, we can parameterize the trajectory via the yaw rate r as a function of time t. The relationship between $r(t)$ and the position $\mathbf{x}(t)$ is

$$\dot{\mathbf{x}}(t) = \mathbf{v}(t) \tag{1}$$

$$\mathbf{v}(t) = [v\cos\psi(t),\ v\sin\psi(t)] \tag{2}$$

$$\dot{\psi}(t) = r(t) \tag{3}$$

where the position $\mathbf{x}$ and velocity $\mathbf{v}$ are vectors in $\mathbf{R}^2$ and ψ is the yaw angle in $\mathbf{R}$.

We approximate the continuous trajectory by discretizing $r(t)$ as

$$\mathbf{r} = [r_1, r_2, \ldots, r_k, \ldots, r_T] \tag{4}$$

where $\mathbf{r}$ is a vector in $\mathbf{R}^T$ whose elements r_k are the yaw rate samples at time indices $k \in \{1, 2, \ldots, T\}$. Given $\mathbf{r}$, the position history $\{\mathbf{x}_k\}$ can be obtained through discrete time integration of Eqs. (1–3). The forward Euler method was used for simplicity in all of the results in this paper.

B. Maximum entropy probability distribution

The following presentation is written in terms of $\mathbf{r}$ in order to make clear the connection to the yaw rate trajectories of interest in this paper. However, the general development here is applicable to any real-valued vector.

Let Ω be the support set of possible yaw rate trajectories in a given circumstance, e.g., of the aircraft of a given type in a given airspace. These trajectories are random, and we assume the existence of an unknown true probability distribution $f(\mathbf{r})$ that is the source of this randomness. Let $R^{\text{obs}} = \{\mathbf{r}_1, \mathbf{r}_2, \ldots\} \subset \Omega$ be a set of observed trajectories assumed to have been drawn from $f(\mathbf{r})$.

Let $\Phi = \{\phi_1, \phi_2, \ldots\ \phi_i, \ldots \phi_L\}$ be a set of feature functions $\phi_i : \mathbf{R}^T \to \mathbf{R}$. (Subscript i will be used to index these features throughout the paper.) Each feature is a function that maps the T-dimensional yaw rate trajectory vector to a scalar value. For each feature, we can define its true expectation with respect to $f(\mathbf{r})$ as

$$E_f[\phi_i(\mathbf{r})] = \int_\Omega f(\mathbf{r})\phi_i(\mathbf{r})d\mathbf{r}\,, \quad i = 1, 2, \ldots, L \tag{5}$$

Although $f(\mathbf{r})$ is unknown, assume that we know the value of each true feature expectation. We seek to approximate $f(\mathbf{r})$ by a distribution $p(\mathbf{r})$ whose feature expectations match those of the true distribution:

$$E_p[\phi_i(\mathbf{r})] = E_f[\phi_i(\mathbf{r})]\,, \quad i = 1, 2, \ldots, L \tag{6}$$

If Eq. (6) is true for many such features, then $p(\mathbf{r})$ is a functional approximation for $f(\mathbf{r})$.

There are many possible distributions that satisfy the constraints of Eq. (6), and so the problem becomes how to choose from among them. The principle of maximum entropy suggests we choose the distribution that assumes the least additional information beyond what is known. Hence, the problem is to choose the distribution that maximizes Shannon's entropy,

$$S = -\int_\Omega p(\mathbf{r}) \log p(\mathbf{r}) d\mathbf{r} \tag{7}$$

subject to the feature constraints.

The solution to this problem is of the form of the well-known Gibbs distribution

$$p(\mathbf{r}; \Lambda) = \frac{1}{Z(\Lambda)} \exp\left(-\sum_{i=1}^{L} \lambda_i \phi_i(\mathbf{r})\right) \tag{8}$$

where $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_i, \ldots, \lambda_L\}$ is a set of Lagrange multipliers and $Z(\Lambda)$ is a normalizing constant. If Λ is chosen such that the feature constraints in Eq. (6) are satisfied, then $p(\mathbf{r}; \Lambda)$ is the maximum entropy (maxent) distribution subject to those constraints.

C. Feature histograms

Since $\mathbf{r}$ is considered here as a random variable, so is $\phi_i(\mathbf{r})$ a random variable with its own distribution. The above formulation uses the true expectation of each scalar feature distribution to constrain the maxent distribution. However, this expectation only captures the mean of each feature, whereas higher-order information about its distribution is lost. We may also want to incorporate features of higher dimensionality, e.g., a feature that maps $\mathbf{r}$ to another vector rather than a scalar.

To better represent the complete distribution of each feature in the maxent model, we will instead constrain the model to match the histogram of each feature over R^{obs}. This amounts to introducing new scalar features corresponding to the value of each bin in each feature histogram.

Let $h(\mathbf{y}, z)$ be the value of the normalized histogram applied to the elements of a vector $\mathbf{y} = [y_1, y_2, \ldots, y_M] \in \mathbf{R}^M$ over the bin defined by the half-open interval $z = [z_{\min}, z_{\max}) \subset \mathbf{R}$. This value is

$$h(\mathbf{y}, z) = \frac{1}{M} \sum_{i=1}^{M} I_z(y_j) \tag{9}$$

$$I_z(y) = \begin{cases} 1 & \text{if } y \in z \\ 0 & \text{if } y \notin z \end{cases} \tag{10}$$

The complete histogram vector of $\mathbf{y}$ over a set of bins $Z = \{z_1, z_2, \ldots, z_B\}$ is

$$H(\mathbf{y}) = [h(\mathbf{y}, z) \mid z \in Z] \in \mathbf{R}^B \tag{11}$$

Now, we redefine each feature function as $\phi_i : \mathbf{R}^T \to \mathbf{R}^M$, where M may be different for each feature. (If $M = 1$, then the feature is a scalar function as in the previous derivation.) Then, $H[\phi_i(\mathbf{r})]$ is the complete histogram of the feature response vector $\phi_i(\mathbf{r})$. Substituting feature histograms for the features in the previous derivation, Eqs. (6) and (8) become

$$E_p\{H[\phi_i(\mathbf{r})]\} = E_f\{H[\phi_i(\mathbf{r})]\}\,, \quad i = 1, 2, \ldots, L \tag{12}$$

$$p(\mathbf{r}; \Lambda) = \frac{1}{Z(\Lambda)} \exp\left(- \sum_{i=1}^{L} \langle \boldsymbol{\lambda}_i, H[\phi_i(\mathbf{r})] \rangle \right) \tag{13}$$

where $\boldsymbol{\lambda}_i \in \mathbf{R}^B$ is a vector of Lagrange multipliers corresponding to feature i and $\langle \cdot\,, \cdot \rangle$ denotes the scalar product of two vectors.

Since the true distribution $f(\mathbf{r})$ is unknown, we cannot compute the true expectation $E_f\{H[\phi_i(\mathbf{r})]\}$ exactly. However, it can be approximated using the sample mean of the feature histograms over R^{obs}. Likewise, $E_p\{H[\phi_i(\mathbf{r})]\}$ can be approximated by the sample mean over a set of synthetic observations R^{syn} sampled from $p(\mathbf{r}; \Lambda)$:

$$E_f\{H[\phi_i(\mathbf{r})]\} \approx \boldsymbol{\mu}_i^{\text{obs}} = \frac{1}{N^{\text{obs}}} \sum_j H[\phi_i(\mathbf{r}_j)]\,, \;\; i = \{1, 2, \ldots, L\}, \;\; \mathbf{r}_j \in R^{\text{obs}} \tag{14}$$

$$E_p\{H[\phi_i(\mathbf{r})]\} \approx \boldsymbol{\mu}_i^{\text{syn}} = \frac{1}{N^{\text{syn}}} \sum_j H[\phi_i(\mathbf{r}_j)]\,, \;\; i \in \{1, 2, \ldots, L\}, \;\; \mathbf{r}_j \in R^{\text{syn}} \tag{15}$$

Here, $\boldsymbol{\mu}_i^{\text{obs}}$ and $\boldsymbol{\mu}_i^{\text{syn}}$ are vectors of dimension B whose elements are the sample mean approximations to the expected value of the corresponding bins in each feature histogram, derived from the samples in R^{obs} and R^{syn} respectively. The values N^{obs} and N^{syn} are the number of samples in R^{obs} and R^{syn} respectively.

D. Learning Λ

In order to draw samples from Eq. (13), we must find the set Λ of Lagrange multiplier vectors $\boldsymbol{\lambda}_i$ that satisfies Eq. (12) using the sample mean approximations in Eqs. (14) and (15). This can be accomplished through an iterative gradient ascent procedure described in Algorithm 1.

First, $\boldsymbol{\mu}_i^{\text{obs}}$ is calculated from Eq. (14) for each pre-defined feature. Then, all $\boldsymbol{\lambda}_i$ in Λ are initialized to zero vectors of appropriate length. At each step j, samples R^{syn} are drawn from Eq. (13) by Gibbs sampling (Algorithm 2) with the

current Λ. The value of $\boldsymbol{\mu}_i^{\text{syn}}$ is calculated from Eq. (15) using the new samples in R^{syn}. Finally, Λ is updated at the end of each step according to

$$\boldsymbol{\lambda}_i^{(j+1)} = \boldsymbol{\lambda}_i^{(j)} - \alpha \left(\boldsymbol{\mu}_i^{\text{syn}} - \boldsymbol{\mu}_i^{\text{obs}} \right), \quad i = 1, 2, \ldots, L \tag{16}$$

where α is a step factor controlling the speed of convergence. This process is repeated until the distance between the observed and synthetic features histograms is sufficiently small:

$$\left\| \boldsymbol{\mu}_i^{\text{syn}} - \boldsymbol{\mu}_i^{\text{obs}} \right\| < \varepsilon \tag{17}$$

This process iteratively adjusts the Lagrange multipliers until the feature histograms of the synthetic samples from the maxent distribution match those of the observed samples from the true distribution. The Lagrange multipliers effectively weight the bins of each feature histogram to satisfy the constraints in Eq. (12). Once the algorithm has produced a good Λ, the distribution $p(\boldsymbol{r}; \Lambda)$ is completely determined and can be used for sampling applications.

Algorithm 1 Maxent model learning.

Require: R^{obs}, Φ
1: $\boldsymbol{\mu}_i^{\text{obs}} \leftarrow$ Eq. (14)
2: $\boldsymbol{\lambda}_i \leftarrow \mathbf{0},\ i = 1, 2, \ldots, L$
3: **repeat**
4: Gibbs sample R^{syn} from $p(\mathbf{r}; \Lambda)$ (Algorithm 2)
5: $\boldsymbol{\mu}_i^{\text{syn}} \leftarrow$ Eq. (15)
6: $\boldsymbol{\lambda}_i \leftarrow \boldsymbol{\lambda}_i - \alpha(\boldsymbol{\mu}_i^{\text{syn}} - \boldsymbol{\mu}_i^{\text{obs}})$
7: **until** $\|\boldsymbol{\mu}_i^{\text{syn}} - \boldsymbol{\mu}_i^{\text{obs}}\| < \varepsilon$
8: **return** Λ

E. Gibbs sampling

The Λ-learning process and applications of the maxent distribution require the ability to draw samples from $p(\mathbf{r}; \Lambda)$ in Eq. (13). This is generally difficult, especially if the dimensionality of $\mathbf{r}$ is large. There are a number of methods to sample from difficult distributions of high dimensionality. This paper uses the Gibbs sampling method. The principle of Gibbs sampling is that it is easier to sequentially sample from a series of one-dimensional conditional distributions than it is to sample directly from the highly dimensional joint distribution.

The Gibbs sampling procedure is briefly described here in Algorithm 2. Begin with an initial value of $\mathbf{r}$. In this paper, $\mathbf{r}$ is initialized to a vector of random noise. At each step j, choose one element r_k of $\mathbf{r}^{(j)}$ at random. Next, draw a new sample for r_k from the marginal distribution

$$r_k \sim p(r_k \mid r_1, r_2, \ldots, r_{k-1}, r_{k+1}, \ldots, r_T; \Lambda) \tag{18}$$

Equation (18) is approximated by a histogram obtained by substituting a discretized range of values for r_k into Eq. (13) while holding $r_1, r_2, \ldots, r_{k-1}, r_{k+1}, \ldots, r_T$ constant at their current values. (The constant $Z(\Lambda)$ can be set to 1 and ignored in this evaluation.) The value of Eq. (18) for each value of r_k is used to form a categorical distribution from which to draw a new r_k. The new r_k is then substituted into $\mathbf{r}^{(j)}$ to yield $\mathbf{r}^{(j+1)}$. When this process is repeated many times, the sequence of $\mathbf{r}^{(j)}$ vectors constitutes a set of samples from $p(\mathbf{r}; \Lambda)$.

Algorithm 2 Gibbs sampling.

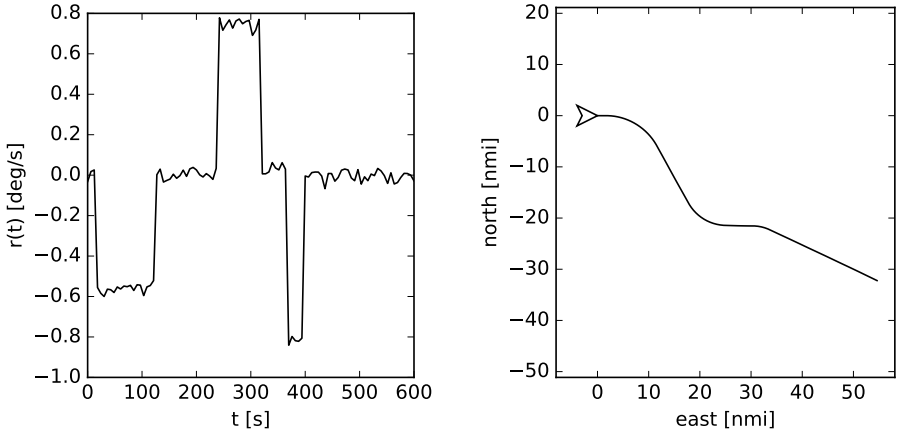
Require: Λ, $\mathbf{r}$
1: **for** $j = 1$ to n **do**
2: Choose k uniformly at random from $\{1, 2, \ldots, T\}$
3: $P \leftarrow \{\ p([r_1, r_2, \ldots, r_k, \ldots, r_T]; \Lambda) \mid r_k \in \{r_{\min}, \ldots, r_{\max}\}\ \}$
4: $P \leftarrow P/\Sigma P$ (normalization)
5: Draw r_k from categorical($\{r_{\min}, \ldots, r_{\max}\}; P$)
6: $\mathbf{r} \leftarrow [r_1, r_2, \ldots, r_k, \ldots, r_T]$
7: **end for**
8: **return** Sequence of $\mathbf{r}$ values

IV. Experiments

A. Observed trajectories

While the intent is to apply this method to a set of real-world air traffic trajectory data, the preliminary application in this paper uses a set of randomly generated surrogate data for simplicity. The trajectories generated by this model serve as the observations in R^{obs}.

Observed yaw rate trajectories are generated according to a procedure that produces a series of segments alternating between straight-and-level flight and level turns of random rate and duration. Figure 1 shows an example trajectory generated by this model. Positive yaw rates indicate left (counter-clockwise) turns. Each aircraft is initialized with initial position $\mathbf{x} = [0, 0]$ nmi and velocity $\mathbf{v} = [400, 0]$ kt in the eastward direction. Since only the yaw rate is varied in this model, each aircraft always travels with a constant speed of 400 kt. All examples and results in this paper use $T = 100$ time steps over a duration of 600 s for each trajectory.



(a) Yaw rate trajectory in time.

(b) Horizontal trajectory in space. Initial position and heading are marked by the chevron.

Figure 1: Example aircraft trajectory in R^{obs}.

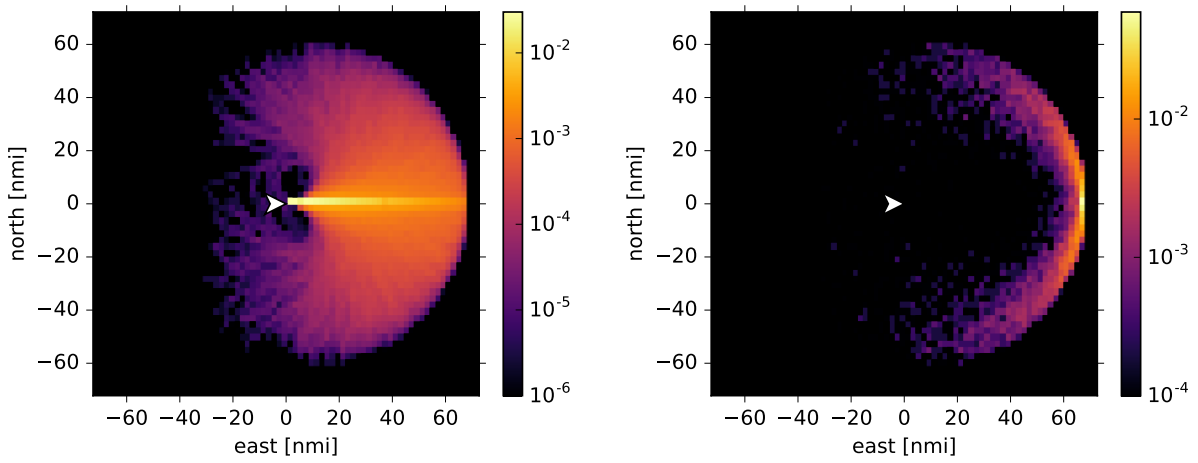
A total of 10,000 trajectories were generated to constitute R^{obs}. Figure 2a shows a 2D histogram of the horizontal positions of these trajectories at all sample times. The numeric value of the color scale is the number of points found in each cell divided by the total number of positions in the given set. Note the color scale is logarithmic. Black indicates areas where no aircraft flew. The highest intensity (yellow) is concentrated from the origin on an eastward trajectory. The orange space indicates the "fan" of trajectories spreading out from the initial position in either direction. The darker, purple areas behind the origin show the rare aircraft that managed to turn around and start flying westward. The fraction of points in the dark areas behind the origin (10^{-5} to 10^{-6}) is much smaller than in the bright areas ahead of it (10^{-2} to 10^{-3}).

Figure 2b shows the same but only for the positions of the aircraft at the final time step ($t = 600$ s). The histogram reveals that after 10 min of flight, an aircraft in this data set is more likely to be found somewhere ahead of its initial position at a range of about 60 nmi. This is not surprising since the maximum range of an aircraft traveling at a constant speed of 400 kt over 10 min is 66.7 nmi.

Although these histograms are but one way of visualizing the statistical characteristics of the observation set, we would expect the synthetic observations generated by through the maximum entropy distribution modeling process to yield similar results if $p(\mathbf{r}; \Lambda)$ is a good approximation for $f(\mathbf{r})$.

B. Feature selection

The modeling process described here requires a set of feature functions to capture statistical behavior found in the observed trajectories. We can conceive of a wide variety of potential features, capturing information about the statistical variability of each aircraft's course, altitude, and speed, possibly conditioned on the type of aircraft and phase of flight. We can also propose characteristics of the overall traffic pattern such as the density of aircraft in a given region, the



(a) Aircraft positions at all times.

(b) Aircraft positions at $t = 600$ s.

Figure 2: Histograms of aircraft positions for trajectories in R^{obs}. Initial position and heading marked by the chevron.

typical separation distance and conflict geometry of any given traffic encounter, and so on. The expansion of the current method to a larger set of features capturing a wider variety of behavior will be the subject of future work.

In this paper, two primary features are demonstrated. The first feature $\phi_1(\mathbf{r})$ is simply the identify function

$$\phi_1(\mathbf{r}) = \mathbf{r} \tag{19}$$

yielding the yaw rate trajectory vector itself. This function is chosen to represent the "DC" component of the yaw rate trajectory signal, and its inclusion ensures that the model reproduces the distribution of yaw rate values observed in R^{obs}.

The histogram approximation for feature 1, obtained from Eq. (14) over R^{obs}, is shown in Fig. 3a. Note the logarithmic scale. A bin count of $B = 19$ is used for all results in this paper. The histogram indicates that values for $\phi_1(\mathbf{r}) = \mathbf{r}$ are concentrated in the middle bin interval of $[-0.06, 0.06)$. This corresponds to the straight-and-level segments observed in Fig. 1a. In these segments, the yaw rate signal has zero mean and a standard deviation of 0.025 deg/s as defined by the generating model. Since the aircraft spend most of their time flying straight and level, it is intuitive that the yaw rate sample is within the middle bin at most time steps.

The histogram has secondary peaks on either side of zero centered at roughly ±0.5 deg/s. This corresponds to the mean value of the yaw rate trajectory in any given turn segment, also dictated by this particular trajectory model. The maximum of these "turn segment" peaks is smaller than the central "straight segment" peak, reflecting that the aircraft spend less time in turns than they do flying straight.

The second feature is defined by the forward difference of the yaw rate trajectory signal:

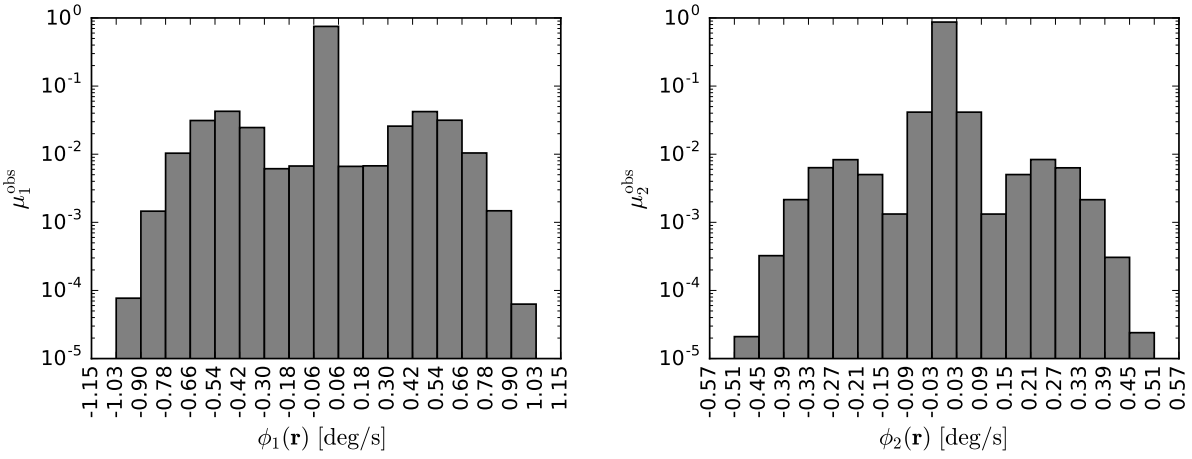
$$\phi_2(\mathbf{r}) = [r_{k+1} - r_k \mid 1 \leq k < T] \tag{20}$$

The forward difference corresponding to Fig. 1a is shown in Fig. 4. The spikes in this figure correspond to the changes in yaw rate experienced at the start and end of each turn segment.

Whereas feature 1 provides information about each yaw rate sample considered independently, feature 2 provides information about the sequential correlation in the signal. That is to say, feature 2 provides information to account for the fact that successive yaw rate samples within a given trajectory are not totally independent. Figure 3b shows the corresponding histogram for feature 2.

C. Maxent model

First, the maxent modeling process will be demonstrated in the degenerate case with no features. This is equivalent to setting all Lagrange multipliers to zero. The effect is that $p(\boldsymbol{r}; \Lambda)$ in Eq. (13) simply becomes the uniform distribution, assigning equal probability density to all $\mathbf{r} \in \Omega$. This is an intuitive consequence of the maximum entropy principle.



(a) Feature 1

(b) Feature 2

Figure 3: Sample mean approximation for the expected value of the feature histograms over the observation set.

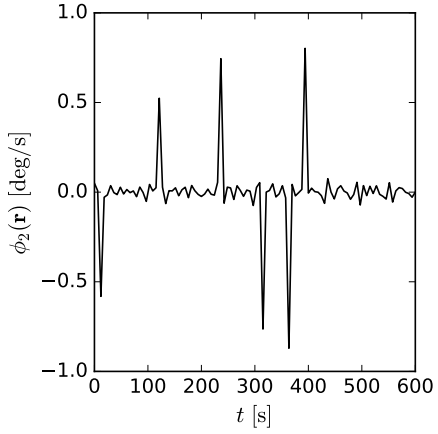
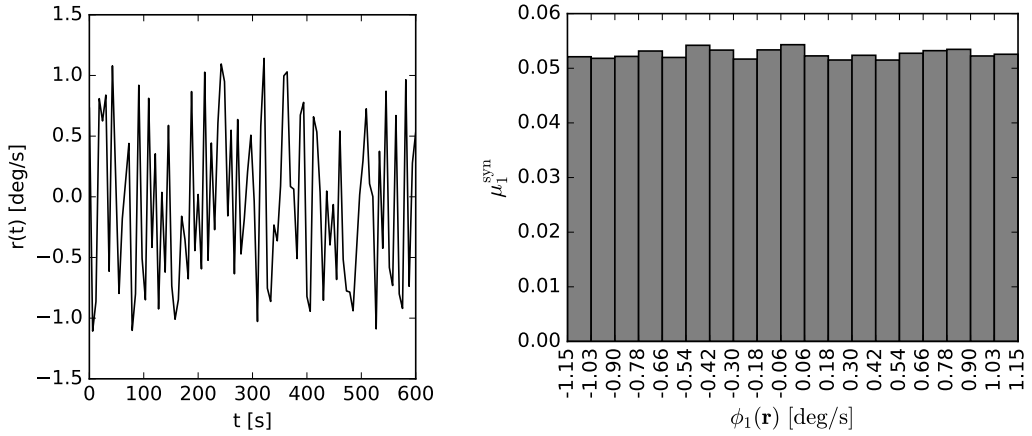


Figure 4: Forward difference of the example yaw rate trajectory in Fig. 1a.

In the case where there are no feature constraints (i.e., no information), the uniform distribution is the one that has the maximum entropy.

Figure 5a shows a single example trajectory produced as the result of Gibbs sampling R^{syn} from this model with no features. Aircraft produced with this model exhibit a continuously varying random walk behavior vs. the more realistic straight and turn segment behavior observed in Fig. 1a. The distribution of yaw rate at any given time is uniform (Fig. 5b), reflecting that this model has no additional information from any features.

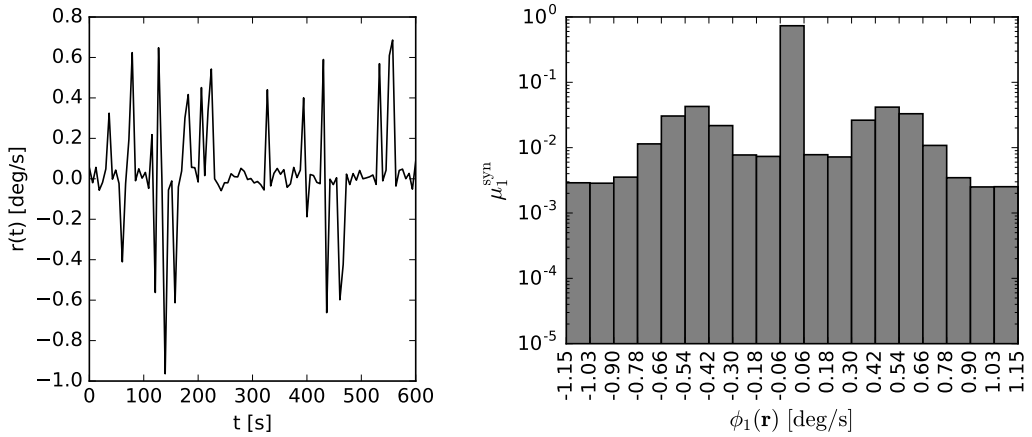


(a) Example yaw rate trajectory.

(b) Histogram of feature 1.

Figure 5: Results of R^{syn} with no features.

Next, feature 1 is added to the model. After learning Λ with feature 1, the model produces yaw rate trajectories such as the one in Fig. 6a. As can be seen in Fig. 6b, the learning process reproduces the histogram of feature 1 (compare with Fig. 3a) with the exception of the small values ($< 10^{-3}$) in the tails. Running the learning algorithm for more iterations with a smaller value for ε would produce better convergence. While the model reproduces values in $\mathbf{r}$ with the correct distribution, the sequential time dependent structure is not represented. The effect is that the yaw rate trajectory signal in Fig. 6a has values of the correct magnitude and frequency but does not have the step-like behavior of the straight and turn segments seen in the observed trajectories.



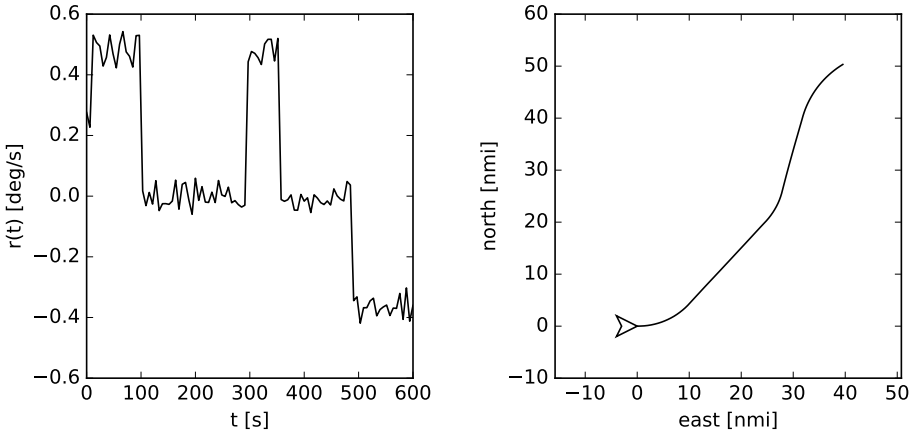
(a) Example yaw rate trajectory.

(b) Histogram of feature 1.

Figure 6: Results of R^{syn} with feature 1.

Finally, the model is learned with both features 1 and 2. Figure 7 shows the an example trajectory produced by this model. With the inclusion of both features, the model reproduces not only the correct distribution of yaw rate values

but also the straight and turn segment behavior seen in R^{obs}.

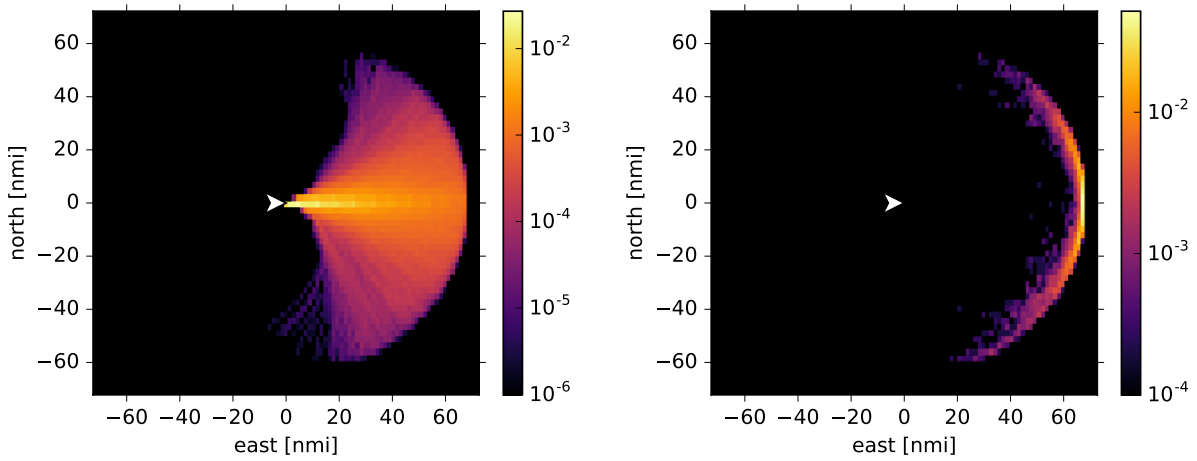


(a) Yaw rate trajectory in time.

(b) Horizontal trajectory in space.

Figure 7: Example aircraft trajectory in R^{syn} with features 1 and 2.

Figure 8 shows the 2D histograms for the results including both features. Compare with Fig. 2. Qualitatively, this model reproduces the shape the distributions in the higher probability areas ($> 10^{-4}$). However, the lower probability areas ($< 10^{-5}$) are underrepresented. For instance, this model does not produce as many trajectories in the darker areas behind the origin as seen in Fig. 2a. This can be attributed to the Gibbs sampling process. Since this is a Monte Carlo sampling method with a finite number of iterations, it can be difficult to obtain good sample coverage in areas of low probability. This is important if the model is to be used to obtain probabilities for rare events, such as a near mid-air collision.



(a) Aircraft positions at all times.

(b) Aircraft positions at $t = 600$ s.

Figure 8: Histograms of aircraft positions for trajectories in R^{syn} with features 1 and 2.

D. Features as airspace constraints

The feature functions convey information in the maxent model about the statistical behavior of the aircraft trajectories of interest. Features 1 and 2 capture some information about the variability of the yaw rate as observed in the original set of trajectory observations. Feature functions can also be designed to convey a rich variety of additional information, including operational constraints that may be imposed on the aircraft.

As a final example, the model is extended with an additional feature to capture the constraint that aircraft will avoid a given geometrically defined area. For example, we may have information that aircraft will avoid a given restricted airspace or region of convective weather. This feature is defined as

$$\phi_3(\mathbf{r}) = \begin{cases} [1,0] & \text{if } \min_{\mathbf{x}_k} \|\mathbf{x}_k - \mathbf{u}\| < d, \quad k \in \{1,2,\ldots,T\} \\ [0,1] & \text{otherwise} \end{cases} \tag{21}$$

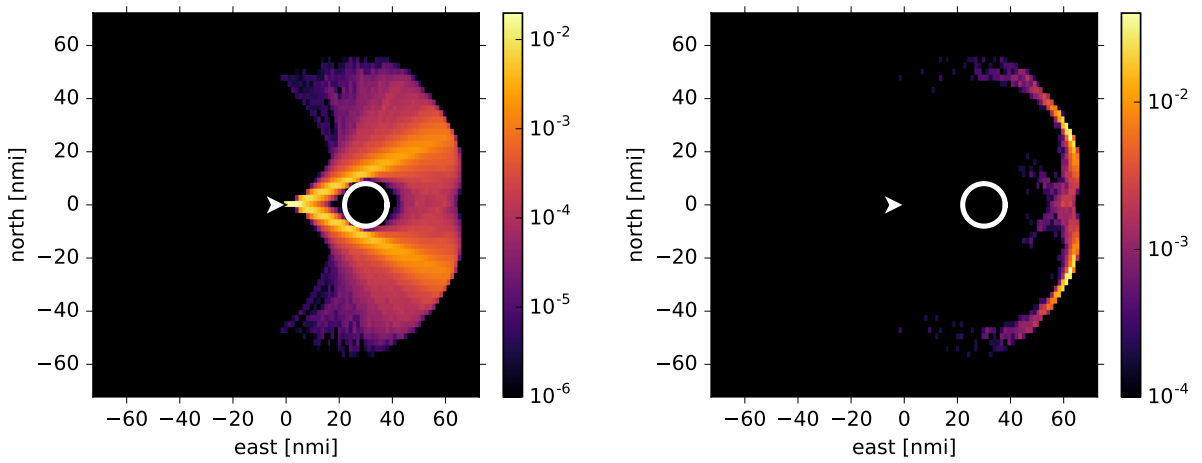
where $\{\mathbf{x}_k\}$ are the position vectors obtained from integrating Eqs. (1–3) for the given $\mathbf{r}$, where $\mathbf{u} \in \mathbf{R}^2$ is the position of the center of the given airspace hazard, and d is the minimum allowable separation distance from the hazard. The feature $\phi_3 : \mathbf{R}^T \rightarrow \mathbf{R}^2$ maps the yaw rate trajectory to a one-hot vector indicating whether that trajectory passes through the hazard or not.

The maxent learning process is repeated, now including features 1, 2, and 3. Recall that Algorithm 1 requires the feature constraints in the form of $\boldsymbol{\mu}_i^{\text{obs}}$, which is the sample mean approximation of the histogram of each feature over the set of observations R^{obs}. However, the samples in the R^{obs} used previously do not exhibit any such airspace constraint as intended by feature 3. However, we can impose this constraint directly by setting $\boldsymbol{\mu}_3^{\text{obs}}$ as

$$\boldsymbol{\mu}_3^{\text{obs}} = [0,1] \tag{22}$$

indicating that $E_p\{H[\phi_3(\mathbf{r})]\}$ should be equal to $[0,1]$ if aircraft never enter this region. This is useful because it shows we can apply constraints that will be reproduced by the maxent model, even if those constraints were not observed in the original set of observations.

Figure 9 shows the 2D histograms with the airspace constraint feature applied in addition to features 1 and 2. The airspace hazard in this case is marked by the white circle. The figure shows all aircraft turning left or right to avoid the hazard.



(a) Aircraft positions at all times.

(b) Aircraft positions at $t = 600$ s.

Figure 9: Histograms of aircraft positions for trajectories in R^{syn} with features 1, 2, and airspace constraint feature 3.

V. Conclusion

The method presented here is a preliminary approach to learning a probabilistic model of aircraft horizontal trajectory behavior. The method employs a set of feature functions to constrain a maximum entropy probability distribution. After learning the parameters from a set of trajectory observations, the model is then sampled to produce new, synthetic trajectories that can be used for a number of ATM applications, including separation assurance.

While the basic approach is presented here, much work remains to adapt the method to practical use. Whereas the maxent method and related methods have received years of development in the field of computer vision, comparable statistical modeling methods in ATM are still in their infancy. To start, the example here is restricted to the 2D, constant

speed trajectory of an aircraft. The method should be extended to the vertical dimension, and also to account for varying speed. The method should also be extended to account for other sources of uncertainty, including wind.

The selection of features is an art in itself. A formal process for designing and selecting which features should be incorporated into the model is needed. Formal measures of the goodness of the learned model are also necessary to characterize the quality of the model's predictions.

The computational efficiency of the method should be addressed, particularly for real-time implementation. The current process for learning and sampling requires many iterations, and improvements are necessary to support an online application. Additionally, the ability of the method to predict rare events needs to be addressed.

Finally, future work will demonstrate the application of this method to separation assurance, in particular to the problem of defining a dynamically optimized separation volume that is appropriate for each aircraft encounter.

References

[1]Slattery, R. and Zhao, Y., "Trajectory Synthesis for Air Traffic Automation," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 2, March–April 1997, pp. 232–238.

[2]Warren, A., "Trajectory Prediction Concepts for Next Generation Air Traffic Management," *3rd USA/Europe Air Traffic Management Research and Development Seminar*, June 2000.

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

[4]Lee, A. G., Weygandt, S. S., Schwartz, B., and Murphy, J. R., "Performance of Trajectory Models with Wind Uncertainty," AIAA Paper 2009-5834, Aug. 2009.

[5]Coppenbarger, R. A., "Climb Trajectory Prediction Enhancement Using Airline Flight-Planning Information," AIAA Paper 99-4147, Aug. 1999.

[6]Hagen, G. E. and Butler, R. W., "Toward a Formal Semantics of Flight Plans and Trajectories," NASA/TM–2014-218662, Dec. 2014.

[7]Nuic, A., Poinsot, C., Iagaru, M.-G., Gallo, E., Navarro, F. A., and Querejeta, C., "Advanced Aircraft Performance Modeling for ATM: Enhancements to the BADA Model," *24th Digital Avionics Systems Conference*, Oct. 2005.

[8]Alexandrov, N., "Control of Future Air Traffic Systems via Complexity Bound Management," AIAA Paper 2013-4419, Aug. 2013.

[9]Ranzato, M. A., Susskind, J., Mnih, V., and Hinton, G., "On Deep Generative Models with Applications to Recognition," *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[10]Graves, A., "Generating Sequences With Recurrent Neural Networks," arXiv:1308.0850 [cs.NE], 2013.

[11]Zhu, S. C., Wu, Y., and Mumford, D., "Filters, Random Fields, and Maximum Entropy (FRAME): Towards a Unified Theory for Texture Modeling," *International Journal of Computer Vision*, Vol. 27, No. 2, 1998, pp. 107–126.

[12]Berger, A. L., Della Pietra, V. J., and Della Pietra, S. A., "A maximum entropy approach to natural language processing," *Computational Linguistics*, Vol. 22, No. 1, March 1996, pp. 39–71.

[13]Phillips, S. J., Anderson, R. P., and Schapire, R. E., "Maximum entropy modeling of species geographic distributions," *Ecological Modeling*, Vol. 190, 2006, pp. 231–259.

[14]Yu, D., Deng, L., and Acero, A., "Using continuous features in the maximum entropy model," *Pattern Recognition Letters*, Vol. 30, 2009, pp. 1295–1300.

[15]Porway, J. M., *A Hierarchical and Contextual Model for Learning and Recognizing Highly Variant Visual Categories*, Ph.D. thesis, University of California, Los Angeles, 2010.

[16]Xie, J., Hu, W., Zhu, S.-C., and Wu, Y. N., "Learning Sparse FRAME Models for Natural Image Patterns," *International Journal of Computer Vision*, Vol. 114, No. 2, Sept. 2015, pp. 91–112.