

Target tracking and Estimated Time of Arrival (ETA) Prediction for Arrival Aircraft

Kaushik Roy*

Benjamin Levy†

Claire J. Tomlin‡§

The problem of developing a unified algorithm for arrival aircraft target tracking and Estimated Time of Arrival (ETA) prediction is approached from a hybrid linear systems approach. Discrete-time hybrid state models are derived and two state estimation algorithms, the Interacting Multiple Model (IMM) and particle filtering with resampling, are implemented for target tracking. Along with the standard Markov chain model for discrete mode changes, the idea of autonomous transitions, or mode changes which depend on the continuous state, are utilized in filtering. The IMM algorithm with autonomous transitions incorporated in discrete mode estimation is developed as an effective ETA predictor. The IMM algorithm is also found to be more efficient than particle filters in terms of run-time and target tracking accuracy. Tracking is performed on observed and simulated data to have RMS errors of less than 50 ft in position and less than 10 ft/s in velocity. ETA predictions are made within 30 seconds of actual landing time for time horizons of nearly 20 minutes.

I. Introduction

The increase of air traffic in the United States to record levels requires improvements in the efficiency of the Air Traffic Control (ATC) system. One bottleneck in the flow of aircraft is traffic on the surface of and in the air surrounding airports. This bottleneck is especially noticeable in weather events such as thunderstorms, which can disrupt arrival airflow at a handful of airports, leading to large delays throughout the national airspace, reducing efficiency dramatically. Previous work has concentrated on routing and scheduling of arrival flows under weather restrictions (see for example, Krozel and Penny [9] and Prete and Mitchell [16]). Even without events severely restricting flow, efficiency of aircraft traffic around airports can be significantly degraded under congestion.

One requirement for efficient flow of air traffic near an airport is accurate knowledge of aircraft position and adequate prediction of future aircraft movement. Without accurate information, maintaining safe separation between aircraft requires much more conservative control techniques, leading to larger taxi times and delays in the air. Also, airport ground crews can be more effectively deployed if arrival and departure times are accurately predicted. Companies such as FedEx depend on appropriate scheduling of unloading and loading crews for their nightly package redistribution at Memphis.¹⁸ Thus, efficiency in both travel time and airline costs can be improved through more accurate tracking and prediction of aircraft states.

To better understand the dynamics of aircraft arriving at an airport, consider the tracks of aircraft landing at Lambert International Airport, in Saint Louis, Missouri (STL), as shown in Figure 1. At many airports, like STL, there is a *four cornerpost* setup, in which aircraft must pass through one of four arrival fixes on entry into the Terminal Radar Approach Control airspace. Under east flow conditions, aircraft landing on runways 12L or 12R pass through one of four fixes: LORLE to the northwest, PETTI to the northeast, KAYLA to the southwest, and QBALL to the southeast. Though aircraft entering a given fix follow the same general path, individual tracks are not identical. Indeed, for arrivals through PETTI onto Runway 12L, it is clear that aircraft can follow very different tracks (Figure 2). Differences are due to configuration of surrounding aircraft, environmental parameters, and the arbitrary choices of individual controllers. The

*Ph.D. Candidate, Electrical Engineering, Stanford University.

†Sensis Corporation, East Syracuse, NY.

‡Associate Professor, Aeronautics and Astronautics, Courtesy Associate Professor, Electrical Engineering; Director, Hybrid Systems Laboratory, Stanford University.

§Roy and Tomlin are supported by NASA grant NCC2-5536.

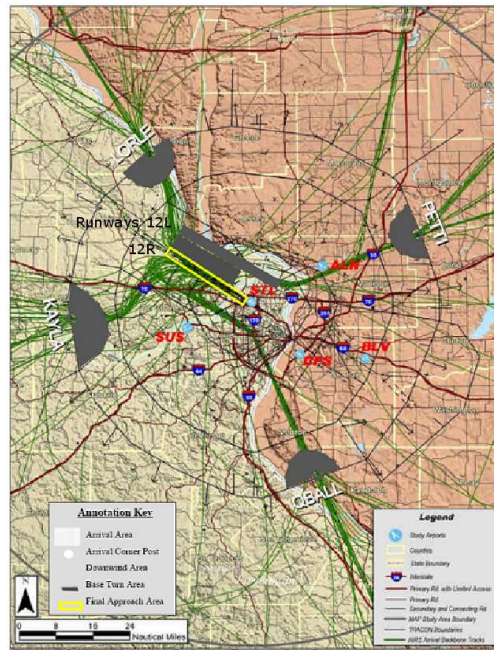


Figure 1. Arrival aircraft tracks under east flow superimposed on map of Saint Louis area. Four cornerpost fixes, LORLE, PETTI, KAYLA, and QBALL, and runways 12L and 12R are labelled.

accurate tracking and prediction of arrival aircraft position at STL provides an example that can be applied at airports throughout the country.

This paper develops a framework in which target tracking and estimated time of arrival (ETA) prediction are performed using an integrated algorithm. First, a stochastic linear model of aircraft motion and noisy observations is developed; state estimates are performed using a Kalman filter. Aircraft dynamics, however, can be very different under different flight modes; that is, straight flight motion and turning motion cannot be represented by a single linear model. We note that in Figure 2, aircraft tracks indeed consist of both straight segments and turning segments. Current techniques often utilize nonlinear models with extended Kalman filtering to track aircraft trajectories consisting of multiple flight modes. However, aircraft motion in each flight mode (straight or turning) can be well represented by simpler (linear) dynamic models. Also, different modes may have different noise characteristics; for example, process noise may be greater under coordinated turns, where multiple flight surfaces are controlled in concert to obtain a constant turn rate. It is also noted that a model with multiple modes can easily incorporate new modes. Scenarios can be designed for specific days, environmental parameters, or surrounding aircraft configurations. Separating flight modes into different system models also allows the specification of the likelihood of flight mode transitions (through autonomous transitions, for example).

Therefore, a stochastic hybrid linear system model with two modes (straight and turning) is used to model aircraft motion. Target tracking, equivalent to continuous state estimation, is part of the hybrid state estimation problem. Solutions to this problem include particle filtering and the Interacting Multiple Model (IMM). These algorithms can also be used to predict the estimated time of arrival. However, this prediction requires a more accurate understanding of expected aircraft motion in the future. While target tracking has the benefit of observations to estimate position, the ETA prediction problem requires estimation without observations. To approach this problem, we consider hybrid models with autonomous transitions. That is, the mode transition matrix is allowed to vary with the continuous state of the system. This allows the evolution of the state in a nominal manner even without measurements, which in turn allows ETA prediction. Observed data obtained by Sensis Corporation at Saint Louis Lambert International Airport and simulated scenarios are used to evaluate the performance of particle filtering and IMM with and without autonomous transitions.

The organization of the paper is as follows. In the next section, the problem of state estimation, as pertaining to aircraft tracking, is explored. Hybrid systems and two approaches to hybrid state estimation,

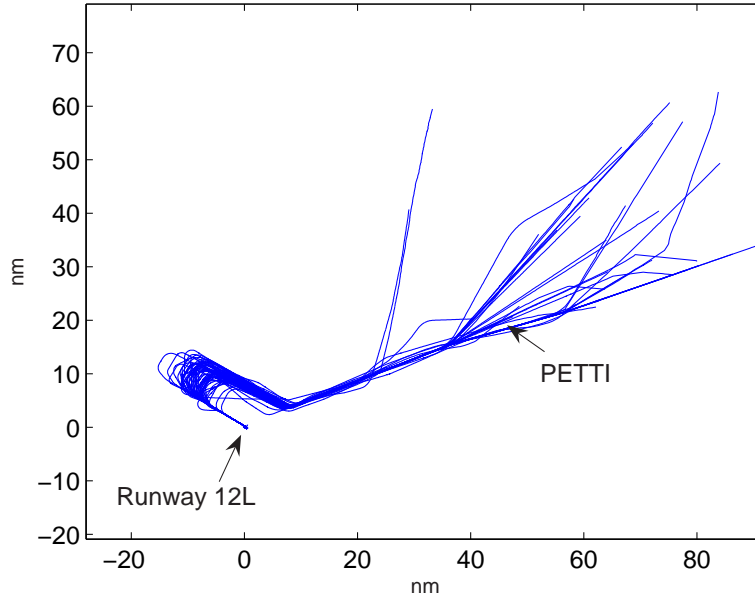


Figure 2. Arrival aircraft trajectories into STL, runway 12L, through PETTI fix. Data from April 21, 2005 under east flow conditions (collected by Sensis Corporation).¹⁶

particle filters and the Interacting Multiple Model (IMM) are introduced. Section III discusses the problem of state prediction. While state estimation involves a prediction estimate over one time step, the problem discussed in this section involves a longer time horizon (i.e. - estimated time of arrival). Hybrid systems with autonomous transitions are introduced. In section IV, the presented theory is applied to aircraft state estimation and arrival time prediction. Appropriate parameters for the hybrid system model are determined from available data. The hybrid state estimation algorithms are evaluated on target tracking of observed and simulated data. The IMM algorithm with autonomous transitions is used for ETA prediction on simulated data. Also, run-time issues are considered. Finally, conclusions and future work directions are presented in Section V.

II. Target Tracking

As discussed in the introduction, ATC strategies require accurate tracking of aircraft position, velocity, and other state variables. Target tracking is one specific instance of a more general problem of state estimation, where noisy measurements and (possibly nonlinear) models of the system dynamics are available. To derive appropriate algorithms for target tracking, it is useful to define the goals of state estimation for particular system models. The overall idea of state estimation is to estimate the internal state of a system at time k given noisy observations of output variables for all times before and up to k .

Discrete-time Stochastic Linear System DTSLs Consider a linear system with discrete-time, continuous-state dynamics:

$$\begin{aligned} x_c(k+1) &= Ax_c(k) + Bw(k) \\ z(k) &= Cx_c(k) + Dv(k), \end{aligned} \tag{1}$$

where $x_c \in \mathbb{R}^n$ and $z \in \mathbb{R}^p$ are the state and output of the system, respectively. The terms $w \in \mathbb{R}^m$ and $v \in \mathbb{R}^r$ are the uncorrelated, white Gaussian process noise and measurement noise with zero means and covariances Q and R , respectively. The evolution of the state and the observation of the state are defined by system matrices $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$, and $D \in \mathbb{R}^{p \times r}$ respectively.

Current algorithms used for ATC target tracking (i.e. - state estimation) are based on constant gain Kalman filters, known as $\alpha - \beta$ or $\alpha - \beta - \gamma$ filters.^{2,4} To understand the effectiveness of state estimation of a DTSL model of aircraft target tracking, we present an example of such a system: a linear model of the motion of an aircraft maintaining straight and level flight at a constant velocity, with measurements taken with sampling interval T . The continuous state variable $x_c(t)$ consists of position, velocity, and acceleration in each of two dimensions; that is, $x_c = [x \dot{x} \ddot{x} y \dot{y} \ddot{y}]^T$. Also, we consider process and measurement noises specific to this system denoted w_{cv} and v_{cv} , respectively. The system model is given by

$$\begin{aligned}
 x_c(k+1) &= \begin{bmatrix} 1 & T & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} x_c(k) + \begin{bmatrix} T^2/2 & 0 \\ T & 0 \\ 0 & 0 \\ 0 & T^2/2 \\ 0 & T \\ 0 & 0 \end{bmatrix} w_{cv}(k) \\
 z(k) &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} x_c(k) + v_{cv}(k).
 \end{aligned} \tag{2}$$

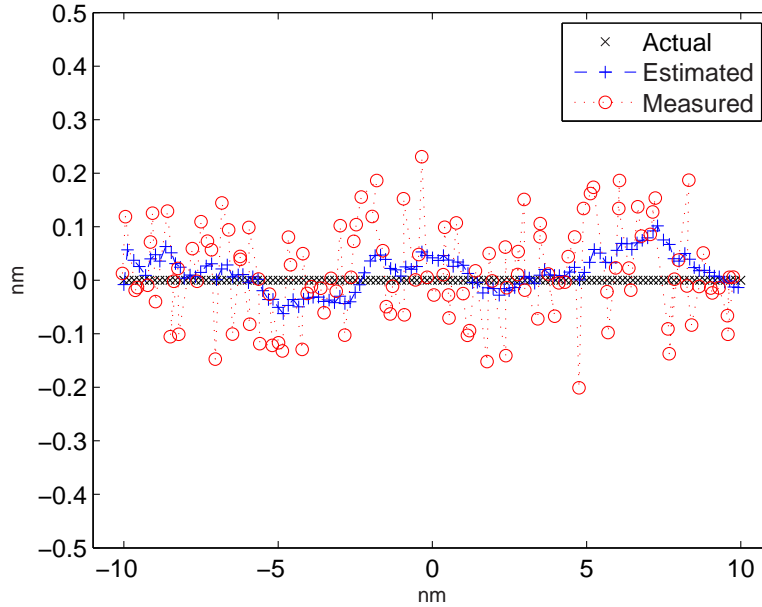


Figure 3. Actual, measured, and estimated position of aircraft travelling due east at constant velocity of 1/6 nm/sec; the noise covariances are $Q = \text{diag}([10^{-6}, 10^{-6}])$ and $R = \text{diag}([10^{-2}, 10^{-3}, 10^{-2}, 10^{-3}])$. Kalman filtered estimates are significantly more accurate than measured points.

Consider an aircraft with initial position $(x, y) = (-10, 0)$ and velocity $(v_x, v_y) = (1/6, 0)$, where positions are given in nautical miles (nm) and velocities in nm/s. Acceleration is assumed to be 0, yielding $x_c(0) = [-10, 1/6, 0, 0, 0, 0]^T$. The target continues on this constant velocity trajectory for 2 minutes, with measurements taken every second ($T = 1$). The noise covariances are $Q = \text{diag}([10^{-6}, 10^{-6}])$ and $R = \text{diag}([10^{-2}, 10^{-3}, 10^{-2}, 10^{-3}])$. In Figure 3, we see the actual aircraft trajectory (marked by 'x'), the measured trajectory (marked by 'o'), and the position estimates (marked by '+') obtained by using a Kalman filter with the system model given by (2). We note that the filtered position estimates are much closer to the actual trajectory than the measurements.

As discussed in the introduction, it is apparent that a single linear model is insufficient in describing the full dynamics of aircraft motion. In the next subsection, the idea of modelling multiple modes of flight through the use of hybrid systems is introduced. The two subsections following describe algorithms used to perform state estimation on hybrid systems: particle filters and the Interacting Multiple Model (IMM).

A. Hybrid systems

Accurate tracking of aircraft is difficult under unanticipated changes in flight mode (i.e. - change from straight flight at constant speed to coordinated turn). A single Kalman filter tuned to a given system model cannot adequately capture the full dynamics of aircraft flight. Even a nonlinear model with an associated unscented Kalman filter may not be able to succinctly capture the disparate dynamics of flight in different modes. To deal with this issue, a variety of adaptive state estimation algorithms have been proposed;^{3, 7, 10, 14, 20} in this paper, we utilize algorithms which compute state estimates using a bank of Kalman filters matched to different modes of the systems.

For planar motion of an aircraft, it is convenient to classify the dynamics either as constant velocity (CV) or coordinated turn (CT). It is noted that tracking of aircraft near airports is generally treated as a planar problem, partly for the simplicity of reduced dimensionality. For the purposes of this paper, the 2D model suffices for capturing the differences in straight and turning flight; future work may include extending mode definitions to 3D. For each mode, we define an appropriate (linear) model for the evolution of the continuous states of the aircraft. That is, the state of the aircraft can be classified into discrete modes (e.g. - CV or CT) and continuous values (position, velocity, acceleration). More formally, we define a discrete-time, hybrid linear system.

Discrete-time Stochastic Hybrid Linear System DTSHLS Consider a set of N linear systems with discrete-time, continuous-state dynamics:

$$\begin{aligned} x_c(k+1) &= A_j x_c(k) + B_j w_j(k) \\ z_j(k) &= C_j x_c(k) + D_j v_j(k), \end{aligned} \quad (3)$$

$j \in \{0, \dots, N-1\}$, where $x_c \in \mathbb{R}^n$ and $z \in \mathbb{R}^p$ are the continuous state and output, respectively. For mode j , the terms $w_j \in \mathbb{R}^m$ and $v_j \in \mathbb{R}^r$ are the uncorrelated, white Gaussian process noise and measurement noise with zero means and covariances Q_j and R_j , respectively. In each mode, the evolution of the state and the observation of the state are defined by system matrices $A_j \in \mathbb{R}^{n \times n}$, $B_j \in \mathbb{R}^{n \times m}$, $C_j \in \mathbb{R}^{p \times n}$, and $D_j \in \mathbb{R}^{p \times r}$ respectively. The discrete mode of the hybrid system, denoted $x_d(k)$, $x_d(k) \in \{0, \dots, N-1\}$, determines which linear system model is used to update the state and output at time k . The evolution of the discrete state is described by the mode transition matrix, denoted H . Entries H_{ij} represent the probability that the system will transition from mode j to i ; note that the transition matrix need not be independent of time or continuous states x_c .

Given a full description of a **DTSHLS** (discrete modes and continuous state dynamics for each mode), we would like to perform hybrid state estimation. That is, we would like to estimate both the discrete state $x_d(k)$ and the continuous state $x_c(k)$ given the observation history $z(l)$, $l \in \{0, 1, \dots, k\}$. The problem of hybrid state estimation has been explored by many authors;^{3, 10, 14, 20} two methods are presented later in this section.

We can apply the definition of a hybrid linear system to the aircraft motion model. First, the discrete state of an aircraft is defined as $x_d(k) = 0$ if the aircraft is in constant velocity mode and $x_d(k) = 1$ if the aircraft is in a coordinated turn at time k . The continuous state $x_c(k)$ is composed of position, velocity, and acceleration states in two dimensions ($x_c(k) = [x \dot{x} \ddot{x} y \dot{y} \ddot{y}]^T$). The evolution of the continuous state in each mode is defined through a **DTSLS** model. For mode 0, constant velocity, this system is as described above in Equations (2). For mode 1, coordinated turn, the system model is

$$\begin{aligned}
x_c(k+1) &= \begin{bmatrix} 1 & T & T^2/2 & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & T^2/2 \\ 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} x_c(k) + \begin{bmatrix} T^2/2 & 0 \\ T & 0 \\ 1 & 0 \\ 0 & T^2/2 \\ 0 & T \\ 0 & 1 \end{bmatrix} w_{ct}(k) \\
z(k) &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} x_c(k) + v_{ct}(k),
\end{aligned} \tag{4}$$

where process and measurement noises specific to this mode are denoted v_{cv} and w_{cv} , respectively.

We also assume the discrete mode follows a time-invariant Markov model: the transition matrix is fixed with respect to time and the continuous state. We also consider a symmetric transition matrix:

$$H = \begin{bmatrix} 1 - \alpha & \alpha \\ \alpha & 1 - \alpha, \end{bmatrix} \tag{5}$$

where α is the probability of changing modes. The term α is often set to 0.05, which is used in this example and throughout the paper. If α is set larger, mode changes are expected to occur much more often, which can lead to instability in switching under noisy conditions. Similarly, using a smaller α results in fewer mode changes, and the hybrid state estimates may end up forced into one mode or the other.

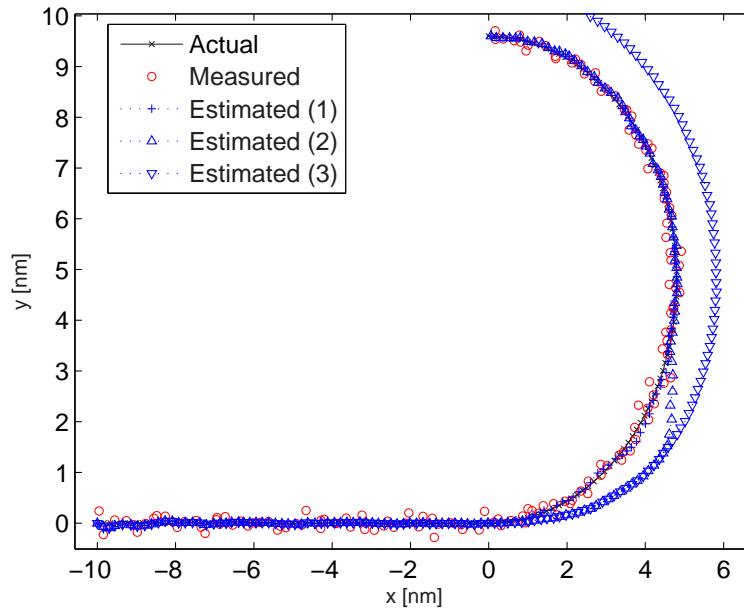


Figure 4. Actual, measured, and estimated position of aircraft travelling due east at constant velocity for 60 seconds and then turning left at 2 degrees/sec for 90 seconds. Estimate (1) assumes correct mode assumptions (CV for 60 seconds followed by CT for 90 seconds). Estimate (2) remains in CV mode for 90 seconds, and estimate (3) remains in CV mode throughout. Noise covariances are $Q = \text{diag}([10^{-6}, 10^{-6}])$ and $R = \text{diag}([10^{-2}, 10^{-3}, 10^{-2}, 10^{-3}])$. Note that all estimated positions are accurate only when the discrete mode assumptions are correct.

Consider an aircraft with same initial conditions as in the single-mode example from above, but in which case the aircraft makes a turn after 1 minute of CV flight. This turn is to the left at 2 degrees/sec and lasts 90 seconds. In Figure 4, we show the estimated trajectory under three different assumptions: (1) if

the aircraft is assumed to have the correct mode (CV the first minute, CT the next 90 seconds), (2) if it transitions to CT mode 1 minute late, and (3) if it remains in CV flight the entire time. It is clear that the estimates are best when multiple modes are used and when the mode changes are timed accurately, highlighting the benefits of using a hybrid model to represent aircraft motion.

Target tracking for this model requires hybrid state estimation; results are improved when assuming complete knowledge of the discrete mode. In cases where the discrete mode $x_d(k)$ is known for all k , the system matrices A, B, C, D are specified as functions of time, and the hybrid linear system reduces to a time-varying linear system. Because the system model is known, Kalman filtering (using the model j whenever $x_d(k) = j$) is a valid approach for target tracking. The problem with this approach is that for a hybrid system with N modes and K time steps, there are N^K possible discrete trajectories $x_d(k)$. The algorithms presented in the next two subsections drastically prune this set of possible discrete states to approximately estimate the continuous state.

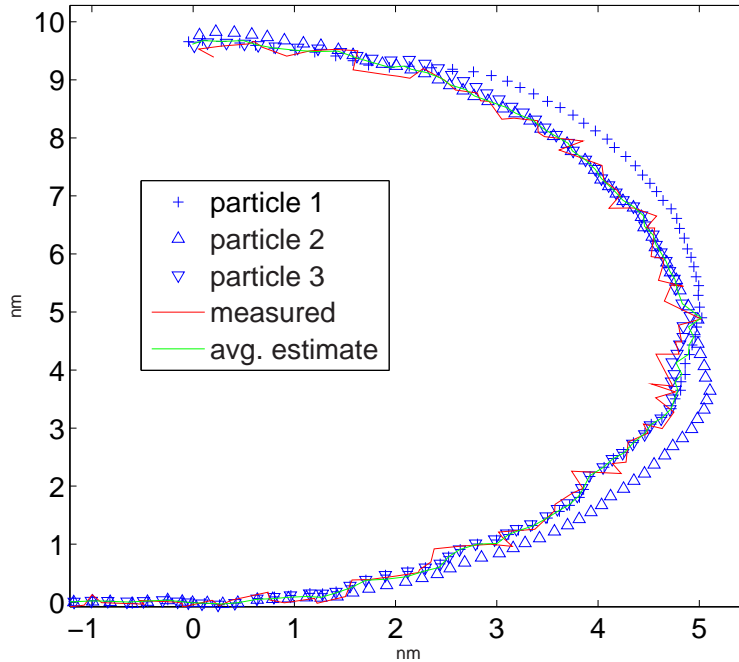


Figure 5. Actual, measured, and estimated position of aircraft travelling due east at constant velocity for 60 seconds and then turning left at 2 degrees/sec for 90 seconds. Each particle is a hybrid trajectory; estimates tend to drift when discrete mode assumptions are wrong. Note that because the particles are independent, the errors are also independent. Resampling allows reduction in error by choosing the estimates which perform best to seed the next time step.

B. Particle filters

One method of hybrid state estimation is to use the more general approach of particle filters. Particle filters involve the tracking of many individual trajectories (particles) to generate a single state estimate. The efficacy of this numerical method is due to the exponential growth in available computational power and the ability to accurately track many independent particles to obtain an accurate aggregate estimate.⁶⁻⁸

For the hybrid state estimation problem, a given particle is a hybrid state trajectory: discrete mode $x_d(k)$ and continuous state $x_c(k)$ at all times $k \in \{0, \dots, T\}$. A basic form of particle filtering consists of three steps: initialization, prediction, and resampling. Initialization of particles implies initializing the hybrid state at time $k = 0$ by sampling from the a priori distribution. That is, we sample the discrete state distribution $p(x_d(0))$ and the continuous state distribution $p(x_c(0))$ to obtain a starting point for each particle. The second step, prediction, consists of three substeps. First, we find the hybrid state of the particle at the

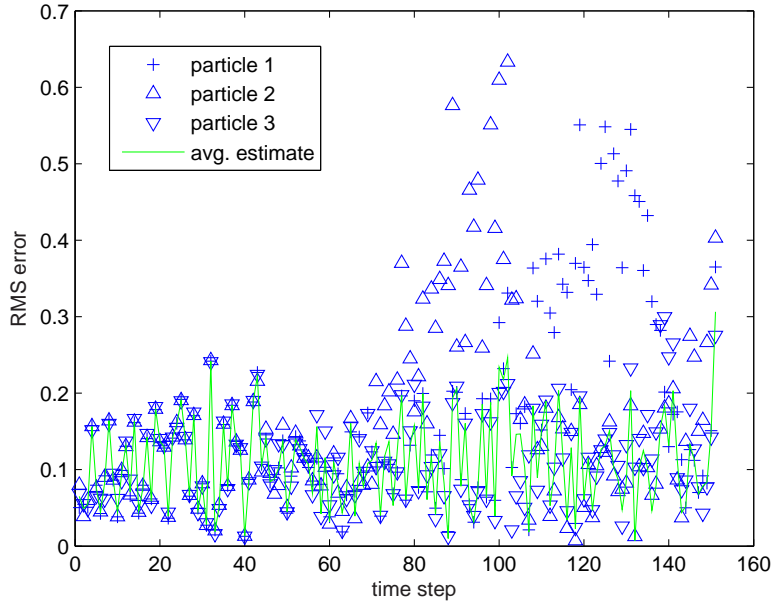


Figure 6. RMS position errors for estimates made in Figure 5. Note that though individual particles are most inaccurate at different points in time, the weighted average estimate always performs best.

next time step. The discrete mode is evolved using the appropriate state transition matrix. A Kalman filter matched to this discrete mode is then used to update the continuous state. Given this update of the hybrid state trajectory, importance weights are determined. These weights are defined as the probability that the observed measurement can be explained by the hybrid state estimate: $p(z(k)|(x_d(k), x_c(k)))$ and can be determined by noting that the system is linear and the noise is Gaussian. Therefore, the expression has a Gaussian distribution with known parameters. Finally, these weights are normalized to determine the relative “success” of each particle at that time step.

The third step, resampling, is added to particle filtering to drastically improve performance. This step involves drawing a new set of particles from the available particles using the normalized importance weights as the distribution function of the new particles. These resampled particles are then input to the system at the next time step. This process ensures that at each time step, the hybrid state estimates made by particles will have higher likelihood.

As an example, we consider the hybrid system model of aircraft motion presented in the previous subsection. We assume the estimate at time zero has the distribution:

$$x_d(t) = \begin{cases} 0 & w/p & 0.7 \\ 1 & w/p & 0.3 \end{cases} \quad (6)$$

and

$$x_c(t) = [-10 \ 1/6 \ 0 \ 0 \ 0 \ 0] + N(0, \text{diag}([10^{-2}, 10^{-3}, 10^{-3}, 10^{-2}, 10^{-3}, 10^{-3}])). \quad (7)$$

That is, at time zero, there is a 70% percent chance that the initial discrete state of any particle is 0 and the initial continuous state of any particle has the normal distribution shown above. We consider the trajectory shown in Figure 5. Also shown are the noisy measurements available to the estimation algorithm and the trajectories of 3 particles. Resampling is not implemented in this example; each particle independently steps through time, updating the discrete mode according to the Markov model and the continuous state according to the mode-matched Kalman filter. The norm of the residual for each particle’s estimate is shown in Figure 6. We see that at different times, different particles are performing with different levels of accuracy; the average estimate performs much better. Using resampling, the poor estimates can be ignored completely and much more robust estimates can be made. That is, resampling ensures that only “good” particles are

used for the next time step. Also, a weighted average of the particle state estimates are used in determining the overall estimate for the algorithm; this too ensures outlier estimates do not affect the state estimate significantly.

The next subsection discusses a different approach to the hybrid state estimation problem, the Interactive Multiple Model. This model generates a single estimate for each mode at each time step rather than propagating trajectories for many particles.

C. Interactive Multiple Model (IMM)

Compared to the particle filter approach of combining the estimates of many separate particle trajectories, the Interacting Multiple Model (IMM) takes a different approach in pruning the possible combinations of discrete modes over time. IMM keeps an estimate of the probability distribution associated with the discrete mode and estimates the continuous state for each possible discrete mode.^{5, 11} The probability that the system is in mode j at time k is denoted $\mu_j(k)$. The continuous state estimate for mode j at time k is denoted $\hat{x}_j(k)$. At each time step, the output of the algorithm is the most likely mode ($\arg \max_j (\mu_j(k))$) and a weighted continuous state estimate ($\sum_{j=0}^{N-1} \mu_j(k) \hat{x}_j(k)$).

At each time step, the IMM algorithm performs four blocks: mixing, Kalman filter updates, mode probability update, and state estimate/covariance combination. The mixing block determines initial conditions (at time k) for the Kalman filter updates. First, the mixing probability, which is the probability that the system is in mode i at time k given that it is in mode j at time $k-1$. This probability, denoted $\mu_{ij}(k|k-1)$, is determined as

$$\mu_{ij}(k|k-1) = \frac{1}{c_j} H(i, j) \mu_j(k-1), \quad (8)$$

where c_j is a normalization constant. The input to each Kalman filter is adjusted by weighting the output of each Kalman filter from the previous time step by the mixing probability. These inputs are denoted $\hat{x}_{0j}(k)$ and $P_{0j}(k)$, respectively.

The second block is a bank of Kalman filters, one tuned to each mode of the system, which are run in parallel. The measurement at time k is denoted $z(k)$. For filter j , the inputs $\hat{x}_{0j}(k)$ and $P_{0j}(k)$ and the measurement are used to obtain state and covariance estimates denoted $\hat{x}_j(k)$ and $P_j(k)$, respectively. The likelihood function of mode j is a measurement of how likely it is that the model in Kalman filter j is correct and is defined as

$$\Lambda_j(k) = \mathcal{N}(r_j(k); 0, S_j(k)), \quad (9)$$

where $r_j(k) = z(k) - C_j \hat{x}_j(k)$ is the residual and $S_j(k)$ is the covariance of the Kalman filter.

The third block generates an updated mode estimate, using the likelihood functions from the Kalman filters. That is,

$$\mu_j(k) = \frac{1}{c_j} \Lambda(j) \sum_i \mu_{ji}(k|k-1). \quad (10)$$

The quantities $\mu_j(k)$ estimate the distribution of the discrete state of the system; that is, $\mu_j(k) = P(x_d(k) == j)$.

The final block combines the state and covariance estimates from each Kalman filter, by weighting them by the appropriate mode estimate, to generate output estimates. Specifically,

$$\begin{aligned} \hat{x}(k) &= \sum_j \hat{x}_j(k) \mu_j(k) \\ P(k) &= \sum_j \{P_j(k) + [\hat{x}_j - \hat{x}(k)] \times [\hat{x}_j - \hat{x}(k)]^T\} \mu_j(k) \\ \hat{m}(k) &= \arg \max_j \mu_j(k), \end{aligned} \quad (11)$$

where $\hat{m}(k)$ is the mode estimate at time k .

The numerical example from the previous subsection is used to show the performance of IMM. Specifically, we have an aircraft moving east for 60 seconds and then turning left at 2 degrees per second for 90 seconds. The noisy measurements and state estimates of the IMM algorithm are shown in Figure 7; the estimates are very accurate, especially as compared to any one of the particles from Figure 5. Mode estimates are shown in Figure 8; the mode change in the estimate occurs at time 70, 10 seconds after the actual mode change occurs. It is common in IMM estimation to have a delay in actually estimating a mode change; however, the trajectory estimate remains robust as the probability of CT mode starts increasing just after time 60.

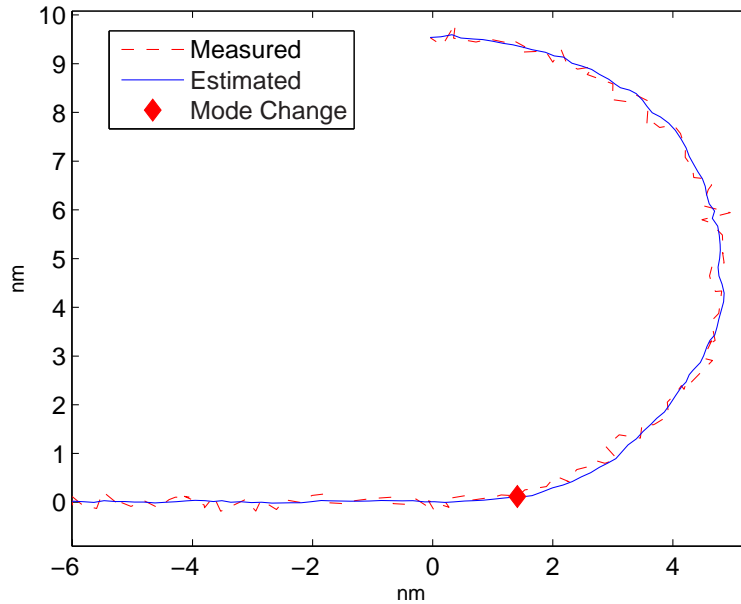


Figure 7. Actual, measured, and estimated position of aircraft travelling due east at constant velocity for 60 seconds and then turning left at 2 degrees/sec for 90 seconds. Estimates generated using IMM algorithm track the actual trajectory very well. The mode change estimated by the algorithm is marked by the solid diamond.

Overall, the estimated trajectory determined by IMM is more accurate than any single particle in particle filtering; the tradeoff between accuracy and run-time due to the number of particles is explored further in Section IV. The next section discusses the challenges of predicting the estimated time of arrival.

III. Prediction

In this section, the problem of estimated time of arrival (ETA) prediction is explored. ETA prediction is an important tool in many applications. Just in the past decade, there has been a proliferation of rapid transit systems, including BART and Muni in the San Francisco Bay Area and Metrorail in the Washington, D.C. area, which have begun publishing ETA predictions both online and at transit stops. Algorithms used to predict ETA for buses and rail systems have also become much more sophisticated. While previous methods focused on aggregate measures, such as average time between two waypoints,¹² newer ideas include linear models which represent the location of vehicles as functions of time.^{13, 15, 17, 19}

In the field of air traffic ETA prediction, recent work has also focused on techniques involving aggregation of past data. For example, prediction algorithms for taxi times on the airport surface have often developed average values for time between runway and gate, or at best developed averages for several unique congestion scenarios.¹ However, these aggregate models cannot adapt estimates to changing conditions; for example, if a specific aircraft has to be routed around a set of runways, the aircraft would have a significantly longer taxi time than other aircraft during this time frame. Thus, the ETA prediction algorithm needs to incorporate dynamic conditions such as aircraft state, other aircraft states, and environmental variables.

This paper treats state tracking and ETA prediction as a problem to be solved under a single framework, a discrete-time stochastic hybrid linear system model with an appropriate filtering technique (either IMM or particle filtering). Both methods utilize Kalman filters to update continuous state estimates; Kalman filtering consists of a state evolution/prediction step and a measurement update step. When measurements are unavailable, state evolution can be carried out without the correction obtained through the measurement update. Prediction of the state for a longer time horizon requires evolving the state over time without measurement updates (since measurements are unavailable for the future). As a result, accurate prediction requires an accurate understanding of future aircraft behavior. That is, without discrete state knowledge,

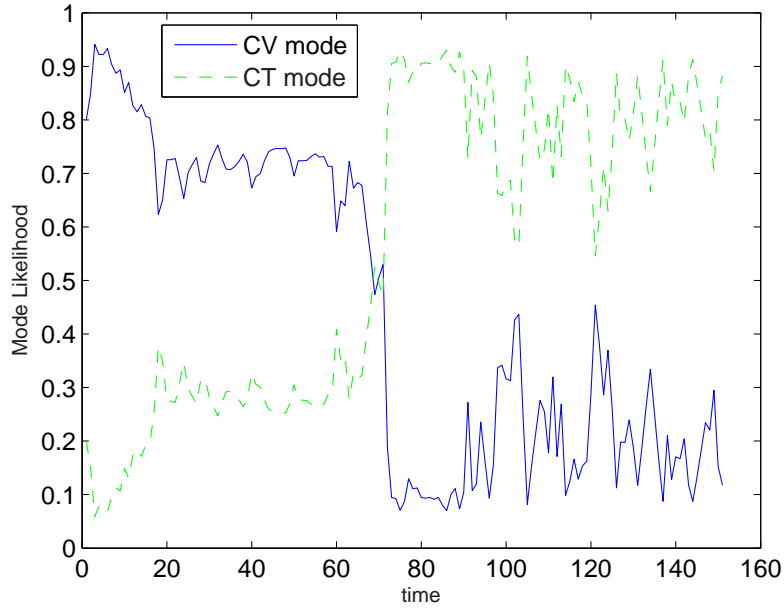


Figure 8. IMM mode estimates for scenario shown in 7. Note that the algorithm determines a mode change at time 70, which is 10 seconds after the actual mode change. However, the probability of being in CT mode starts increasing at the time of the mode change; as a result, position estimates are still accurate.

the evolution of the continuous state cannot be done accurately. For this reason, particle filtering cannot be used effectively for predictive purposes. The stochastic nature of the system implies a nonzero probability of mode transitions (otherwise the system would be time-varying but deterministic). Hence, all particles in the filter will at some time switch to the wrong mode, at which time the evolution of the state will become inaccurate. Measurements help determine the validity of discrete mode estimates and the likelihood of continuous state estimates; without measurements, particle filters are not able to determine an estimate for the arrival time.

For the same reason, a Markov model is also not suitable for ETA prediction. In the previous section, hybrid systems were introduced to deal with the issue of unanticipated flight mode changes. In this context of tracking, mode changes were expected to follow a Markov model; that is, the probability of transitioning from one mode to another is fixed. For any filtering algorithm evolving without measurements, a fixed (nonzero) transition probability implies discrete state predictions will eventually become inaccurate, leading to inaccurate continuous state predictions. In the next subsection, we introduce the idea of mode transition probabilities that are functions of the continuous state. Such a model allows the evolution of the continuous state to approach some prior expectation which can be then used to predict ETA. The following subsection discusses changes in the acceleration noise model that are used to accurately represent the gradual decrease in speed prior to landing.

A. Autonomous transitions

Accurate tracking and prediction both require accurate models of aircraft motion. In Section II, stochastic hybrid linear systems were introduced to handle multiple separate modes of flight. In that development, it is assumed that mode changes follow a Markov model; that is, there exists a constant transition matrix which describes the probability of changing from one mode to another at any give time step. However, it is clear that aircraft, especially those making maneuvers just before landing, do not make mode changes at random. While tracking requires anticipating mode changes and then validating the estimate through measurements, prediction requires looking ahead in time with no current data. Instead, prior information such as average time to traverse a given distance or time to landing for the aircraft ahead are used to guess the behavior of the given aircraft. We aim to codify this prior information into the hybrid system model through the use of

autonomous transitions. Autonomous transitions are discrete mode changes triggered by continuous state variable changes (e.g. - such as turning a heater on if the temperature falls below 68 degrees). Autonomous transitions can also be represented by a mode transition matrix that is a function of the continuous state. In terms of the heater example, we may set it to turn on with increasing probability as the temperature drops.

In the context of aircraft performing landing maneuvers, there are two assumptions used to determine the variability of the mode transition matrix with respect to the aircraft state. First, it is assumed that aircraft in constant velocity mode (CV) follow straight line segment paths to a point in space before turning. Given a set of data (arrival aircraft tracks) which exhibit alternating CV and CT modes, the locations of transitions from CV to CT mode are tabulated. It is further assumed that aircraft in coordinated turn (CT) mode turn until reaching a specified heading, at which point the mode switches back to CV. Thus, the headings at the end of a turn are tabulated to locate transitions from CT to CV mode. Almost all aircraft end coordinated turns at the same heading, so the distributions on heading are very narrow compared to the relatively broad spatial distribution of points where CV to CT mode changes occur.

Given the distributions of mode change locations (in physical space or in heading coordinates), the means of the distributions define a nominal track; that is, the average path for a given arrival aircraft. The distribution of the state variables at transition points help determine the variability of the transition matrix in regions around the mean transition point. In other words, given the aircraft location and heading (i.e. - continuous state), the mode transition matrix can be obtained. In the next section, this idea is developed further and parameters are numerically determined from STL arrival data.

The nominal track for arrival aircraft is necessary for arrival time prediction. Without knowledge of future aircraft path (specifically, knowledge of expected mode transitions), one cannot accurately determine future positions of an aircraft. This is due to the fact that even though dynamics can be accurately modelled in the current flight mode, state estimates break down if mode changes are not accurately modelled. With the nominal track, which is an expected set of flight segments for which dynamics models are available, the future state of an aircraft can be accurately determined, assuming the aircraft remains on the nominal track. Errors in ETA prediction are still due to differences between the nominal and the actual track, but these errors are greatly reduced if the distribution of all arrival tracks is narrow relative to the nominal track. Future work will focus on determining scenarios under which the arrival tracks closely match the nominal track.

B. Acceleration noise model changes

For accurate ETA predictions, the slow decrease in aircraft speed before landing must also be incorporated into the stochastic hybrid linear model. This decrease is small (less than 0.5 nm/hr each second) and can be ignored in target tracking for reasonably frequent measurements (e.g. - at least one measurement every few seconds). However, since time horizons for ETA predictions are on the order of tens of minutes, the decrease in speed over this time horizon needs to be included in estimating time to landing. This acceleration is added to the model by changing the process noise w to have nonzero mean. Numerical examples are provided with the STL data set in the next section.

The changes to acceleration noise model are important mostly in ETA prediction, while adjusting the mode transition model to include autonomous transitions is beneficial to estimating unknown mode changes in ETA prediction as well as in recognizing mode changes quickly in target tracking. In the following section, we analyze the performance of particle filters and IMM in target tracking and ETA prediction.

IV. Simulations and Results

Having developed a stochastic hybrid linear model to model aircraft motion, we look to validate the model and evaluate the various implementations discussed in the previous sections. Specifically, we test the accuracy of incorporating the possibility of autonomous transitions in the transition matrix, using resampling in particle filtering, and the differences between IMM and particle filters. Data used in this section comes courtesy of Sensis Corporation and is taken from Lambert International Airport in Saint Louis, Missouri (STL). The data is first parsed to generate appropriate parameters for the stochastic hybrid linear system used to model aircraft motion as well as the autonomous transition matrix to model mode changes. Next, we compare the various algorithms in tracking an aircraft as it comes in for landing at STL. Both observed and simulated data are used in this comparison. Run-time issues are also presented. The efficacy of these

algorithms in predicting ETA is also analyzed for a simulated scenario based on observed data.

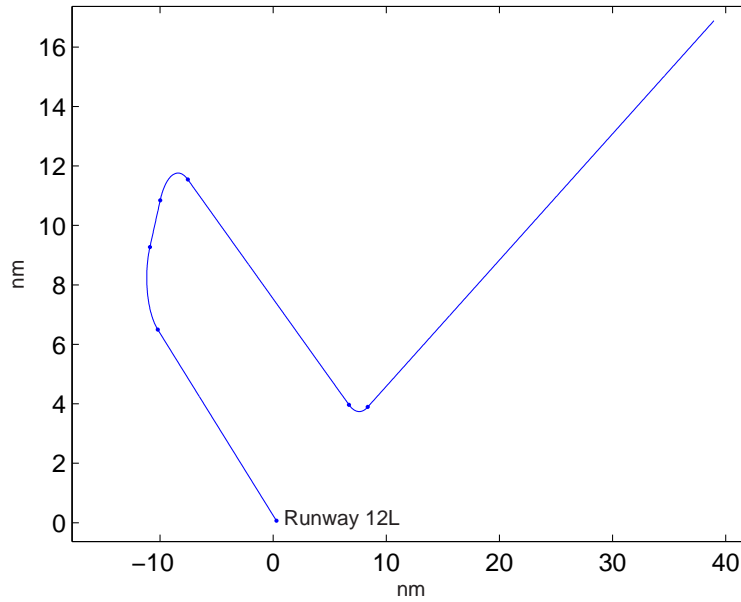


Figure 9. Nominal arrival track; to derive this track, the transition points of available arrival tracks are tabulated. The mean transition points are then used to define the transitions of the nominal arrival track; these points are marked on the plot. Available arrival tracks are for flights landing at STL runway 12L on April 21, 2005.

A. Derivation of parameters for STL from given data set

To perform target tracking and ETA prediction with a hybrid system model, appropriate parameters must be set for the dynamics in each mode of the system as well as for the transitions between modes. To determine these parameters, we consider a set of arrivals coming from the northeast of STL (through arrival fix PETTI) and landing on runway 12L; trajectories of these flights are shown in Figure 2. Data taken for these arrivals are commonly available to a radius of 40nm from the airport and were collected by Sensis Corporation on April 21, 2005. Trajectories start with southwesterly flight (heading approximately 250), turning right to a heading of 300 (downwind), then turning left 180 degrees for final approach to the runway, located at the origin on the figure. Looking at the velocity profiles of the flights, we find that speed generally decreases on the straight segments and remains much more constant during turns. Inspired by the constant velocity and coordinated turn modes from Section II, we define two modes corresponding to the straight and turning portions of flight as shown in Figure 9. This figure shows a nominal approach based on the aggregation of all arrivals shown in Figure 2. Consider Figure 10, which shows a portion of the arrival tracks from Figure 2. Marked on this plot (by x's) are locations of the first mode change, from CV to CT flight. The mean of this distribution is marked by a diamond; the nominal track is found by using the mean transition point for each transition.

The dynamics in straight and turning segments are as developed in Equations (2) and (4), respectively, from Section II. The discrete mode evolution is broken into two cases. First, in the static transition matrix case, we assume mode changes occur with 5% probability. That is, the transition matrix is

$$H = \begin{bmatrix} 0.95 & 0.05 \\ 0.95 & 0.05 \end{bmatrix}. \quad (12)$$

The second possibility incorporates autonomous transitions. The distribution of transition points from the STL data set are used to determine the transition matrix as a function of the continuous state. The distribution of the first mode change is noted in Figure 10. As aircraft traverse (entering from the top

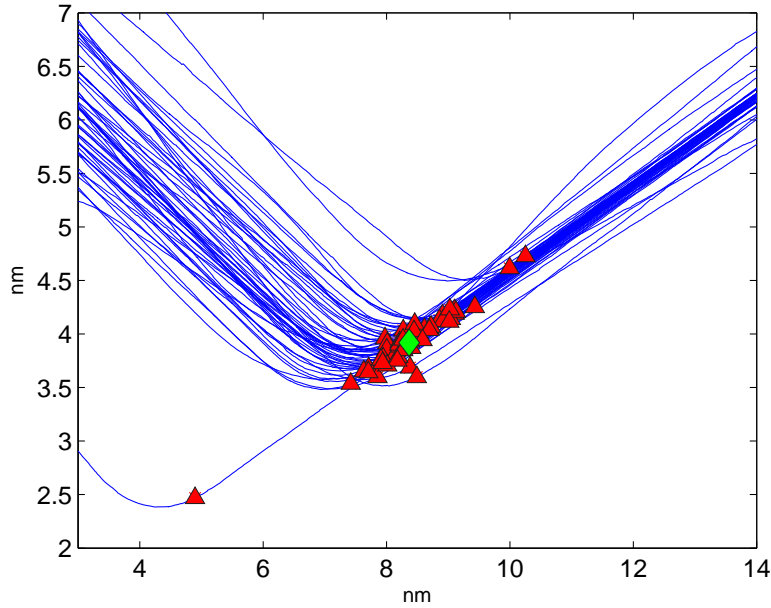


Figure 10. Portion of arrival aircraft tracks near where aircraft start to turn toward downwind leg. The points at which aircraft start the turn are marked by solid triangles; the mean of these points is marked with solid diamond. This point is the first marked point on Figure 9.

right) the region shown in this plot, it becomes more and more likely that it will start a turn to the right. This information can be encoded by increasing the probability of transitioning from CV to CT mode from top right to bottom left (with contours running parallel to the nominal track heading of approximately 250 degrees). We similarly encode each CV to CT transition by marking the region inside which the transitions have occurred in data and defining the probability of transition to be an increasing function along the flight path (for this paper, a linear function is chosen). For CT to CV transitions, we note that the CT mode continues until specific headings are attained. Thus, the probability of transition out of CT is defined by the heading (which is a function of the velocity of the aircraft; i.e. - derived from the continuous state). Specifically, the probability of transitioning to CV mode is 0.05 when the heading is more than 5 degrees away from desired and 0.95 when the desired heading is reached, with linear interpolation in between.

In the next subsections, two hybrid state estimation algorithms (IMM vs. particle filters), and two mode transition schemes (Markov model vs. autonomous transitions) are compared. The parameters derived in this section are used in the models used below; simulated scenarios are also generated from these parameters.

B. Target tracking

In this subsection, we compare different methods for target tracking using the hybrid system model for aircraft motion. First, IMM methods using either a Markov model or autonomous transition model for mode changes are compared to determine which maintains the best estimates of the discrete mode. This comparison is performed using actual data from arrivals into STL. It is customary to compare state estimates to the true state values, but true state values are unavailable for measured data.

An alternative, to generate a simulated scenario using the nominal arrival track and estimated measurement noise parameters, is used to properly analyze the performance of the different algorithms. We find that all four methods have similar efficacy as measured by RMS position and velocity error, but the methods using autonomous transitions clearly outperform those using a Markov model for estimating the discrete mode changes. Finally, run-time issues between the methods are considered.

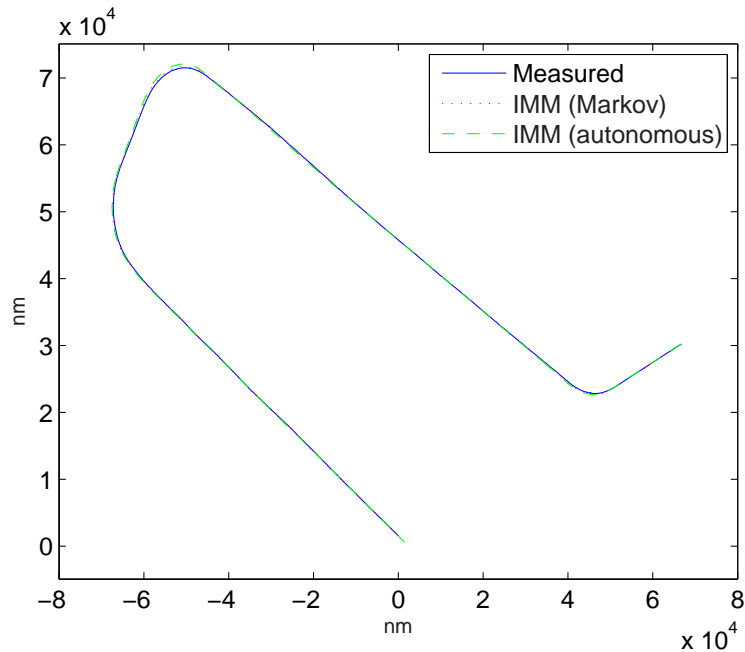


Figure 11. Measured and estimated trajectories for an observed arrival. Estimates made using IMM algorithms; both Markov and autonomous transition models are very accurate.

1. IMM estimates for measured arrival track data

In Figure 11, the measured trajectory for an aircraft landing at STL is shown (solid line). The dotted line shows the estimates generated by using IMM with a Markov model for mode changes, while the dashed line shows estimated for IMM with an autonomous transition model. The estimates are nearly coincident with the measurements, with slight overshoot in estimation at the beginning of turns. To illustrate the estimates in another way, the aircraft heading is also plotted as a function of time in Figure 12. Again, both methods exhibit good tracking of heading, removing much of the noise in the signal while tracking turns effectively. In Figure 13, the mode estimates for the two methods are shown. Actual mode changes are shown by vertical lines, while the estimates shown are the probability that the aircraft is in CT mode. Incorporating autonomous transitions in the discrete state model leads to decreased confusion in the mode estimates.

Both algorithms perform reasonably well in position estimates and mode estimates, with inaccuracies occurring in a delay in estimating the change from CV to CT mode. This delay is reduced when autonomous transitions are incorporated. Also, using autonomous transitions leads to more distinct likelihoods of modes than for the Markov model. The advantage of using autonomous transitions, that mode estimates are maintained more accurately, is thus demonstrated using this data set. The next section analyzes the performance of all four algorithms on simulated data.

2. Tracking of simulated arrival data

To compare tracking algorithms, the measure used is RMS error between the actual and the estimated continuous state. This measure requires a ground truth for the continuous state, which cannot be obtained for measured data. A simulated scenario based on the STL data set is used for this analysis. To generate this scenario, a point is chosen for each transition point from the appropriate distribution (the mean of each distribution was used to generate the nominal track). These points define a trajectory for the arrival aircraft; this trajectory, plotted in Figure 14, is used as the baseline for generating RMS error estimates. Tracking is performed using both IMM and particle filters, each with a Markov model or an autonomous transition model for discrete mode changes. Because the estimated trajectories are again very close to the measurements, only RMS position errors are plotted in the figures below. The first plot, Figure 15, shows the RMS position errors for the IMM algorithm with Markov or autonomous transition models for discrete

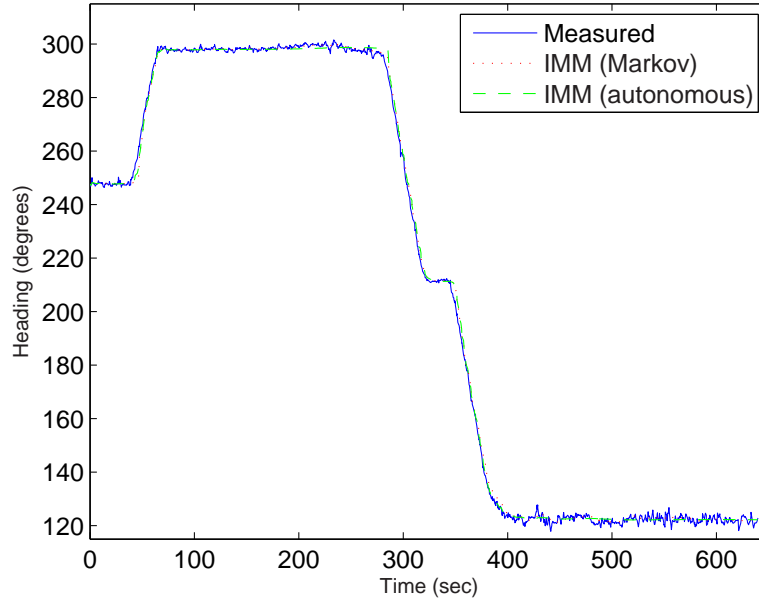


Figure 12. Measured and estimated headings for the same arrival as in Figure 11.

mode changes. The second plot, Figure 16, is the same but for particle filters using 10 particles. The same scenario run with 100 particles per filter yields Figure 16. In Table 1, the average RMS position and velocity errors are presented for 100 runs for the two IMM algorithms and the two particle filter algorithms (with 10, 20, and 100 particles). We note that adding particles leads to a slight improvement in tracking. It is also interesting to note that the IMM algorithm performs better than particle filters with 20 particles and similar to particle filters with 100 particles. Finally, there is a small improvement in tracking error due to using autonomous transitions. However, this result is highly sensitive to the simulation trajectory; using autonomous transitions with a simulation trajectory further away from nominal will have worse results.

3. Run time considerations for target tracking

Although particle filters have the benefit of maintaining many state and prediction estimates, as well as the flexibility of adding particles to obtain improved accuracy, the run-time costs of particle filters can be

Table 1. Average RMS position and velocity errors determined with 100 runs for each algorithm on simulated data; N represents the number of particles for a given particle filter. We find that more particles lead to better results; also, the autonomous transition model slightly outperforms the Markov model. The IMM algorithm outperforms all but the $N = 100$ particle filter.

Algorithm	RMS Position Error (ft)	RMS Velocity Error (ft/s)
1) IMM (Markov)	35.7	10.2
2) IMM (Autonomous)	35.0	9.7
3) Particle Filter(Markov), $N = 10$	42.8	10.4
4) Particle Filter(Autonomous), $N = 10$	39.0	9.9
5) Particle Filter(Markov), $N = 20$	38.8	10.1
6) Particle Filter(Autonomous), $N = 20$	36.3	9.6
7) Particle Filter(Markov), $N = 100$	35.8	9.9
8) Particle Filter(Autonomous), $N = 100$	31.5	9.6

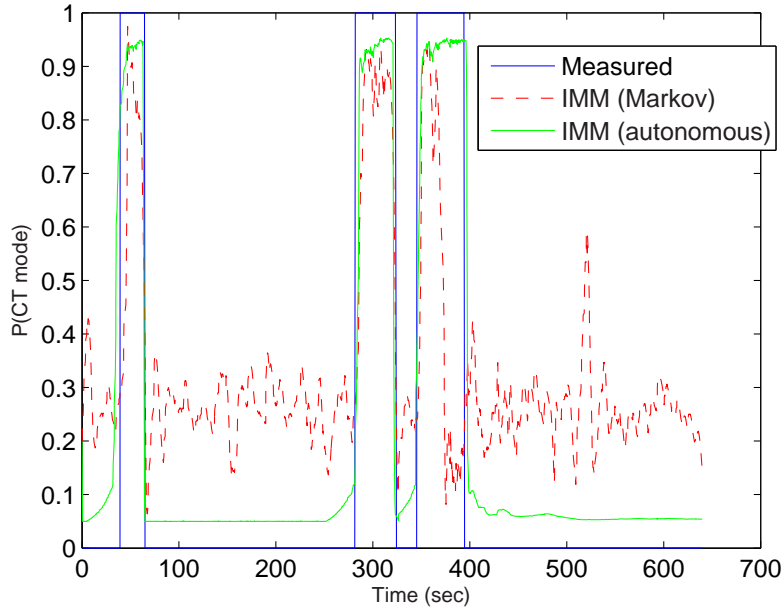


Figure 13. Mode estimates for IMM algorithms for arrival depicted in Figure 11. Actual mode changes marked by vertical lines. The output mode would be obtained by rounding the mode estimates plotted here; thus, both techniques are generally correct in estimating the discrete mode. However, the autonomous transition model leads to more certain mode estimates, which is important in target tracking and, more importantly, in predicting ETA.

quite expensive. Running the IMM algorithm on the simulated scenario takes an average of 3 seconds (on a laptop with a 1.3 GHz Intel Pentium M processor). On the other hand, running a particle filter with 10 particles takes close to 11 seconds on average. Including autonomous transitions makes a minor difference for either algorithm, since the mode transition matrix calculation is essentially a lookup table. The bulk of the computational costs come from Kalman filtering, which require matrix inverses and must be computed at each time step; data for a single arrival consists of 1000-2000 data points. The IMM algorithm runs two Kalman filter updates per time step (one for each discrete mode), while the particle filter runs N updates, where N is the number of particles (i.e. - 20). Thus, we expect the particle filter to take an order of magnitude longer to run. Because overhead costs such as mode mixing and resampling are similar for each algorithm, the overall ratio in run time is not quite 10. Table 2 shows that run time scales slightly sublinearly with the number of Kalman filter updates per time step, as expected. In a real-time environment such as air traffic control, the added accuracy of a larger set of particles may not be useful if tracking and ETA estimates are required with high frequency.

C. Estimated Time of Arrival (ETA) prediction

As discussed in section III, only one of the four algorithms presented above can effectively perform ETA prediction. Methods modelling discrete mode changes using a Markov model are unable to estimate future mode changes without measurements. As a result, these methods are not capable of predicting ETA. Additionally, particle filtering methods cannot capture future mode changes effectively without a more sophisticated model for mode changes. Even with autonomous transition models, mode changes in particle trajectories are random processes and occur more often than in the actual trajectory. That is, even though mode changes are greatly reduced through modelling mode changes effectively with autonomous transitions, particles tend to change modes often, rendering effective continuous state estimates impossible without measurements. As a result, the only effective method for ETA prediction amongst the four presented is the IMM algorithm modelling discrete mode changes with autonomous transitions.

The simulated trajectory used in the previous subsection is used in this section for purposes of testing

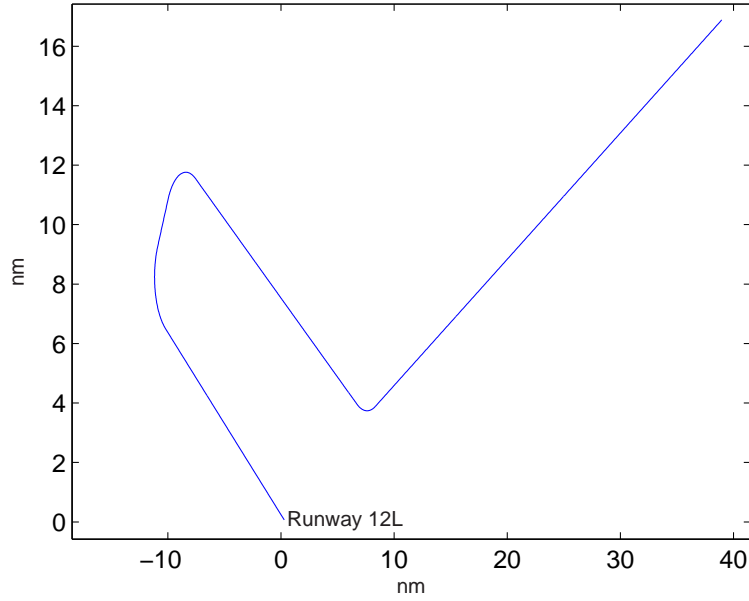


Figure 14. Trajectory used for simulation; trajectory obtained by adding noise to the location of the transition points that define the nominal trajectory from Figure 9 as well as adding noise to individual position measurements.

ETA prediction. The actual time of arrival is at $t = 1125$, where the simulation starts at time 0. The prediction algorithm evolves the state without measurements but still requires the time-consuming Kalman filter updates. ETA predictions are made every 10 seconds; these predictions are plotted (o's) in Figure ???. The horizontal line in the figure is the actual arrival time. At $t = 10$ seconds, the first ETA prediction is made: $t = 1154.9$. The error is just under 30 seconds. The error remains relatively steady near 35 seconds until the first turning segment, which leads to some variation in the ETA. However, throughout the downwind leg ($t \in [600, 800]$), the error again approaches 35 seconds. The main cause in the overestimation of the time of arrival is in the attempt to classify all possible arrivals with a single nominal track. Because of the differences between the nominal and the scenario track, especially in the turns to final approach, there is a 50 second swing in the ETA as the aircraft makes turns to final approach. The error over the last two minutes of flight is under 5 seconds; this error is mostly attributable to the assumption that the runway is at the origin (each runway is in fact slightly off the origin) and that the aircraft slow significantly in the

Table 2. Average run-time determined with 100 runs for each algorithm on simulated data; N represents the number of particles for a given particle filter. Note that particle filter run-times scale nearly linearly in the number of particles. Also, the IMM algorithms are clearly superior in terms of run-time.

Algorithm	Average run-time (sec)	Average run-time per time-step (m)
1) IMM (Markov)	3.04	1.3
2) IMM (Autonomous)	3.01	1.3
3) Particle Filter(Markov), $N = 10$	10.9	4.9
4) Particle Filter(Autonomous), $N = 10$	12.0	5.3
5) Particle Filter(Markov), $N = 20$	21.8	9.7
6) Particle Filter(Autonomous), $N = 20$	23.6	10.5
7) Particle Filter(Markov), $N = 100$	110.9	49.2
8) Particle Filter(Autonomous), $N = 100$	118.7	52.7

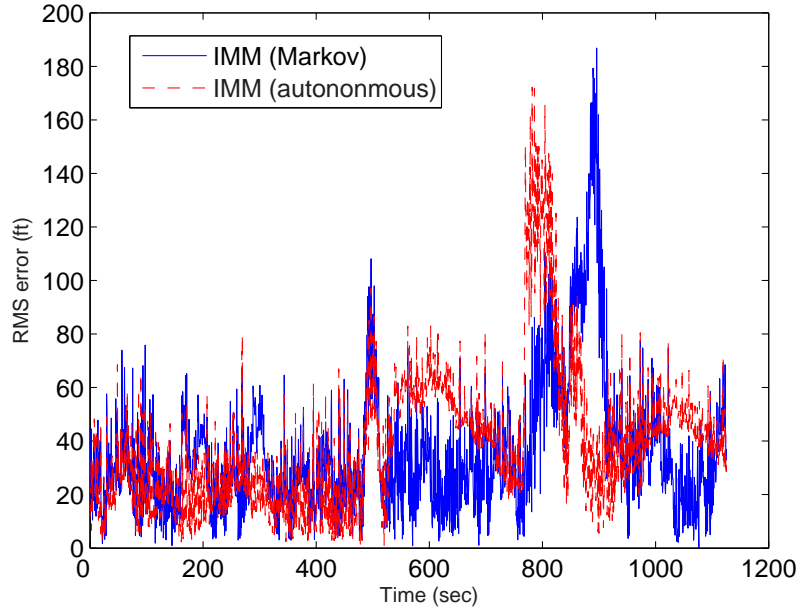


Figure 15. RMS position errors for IMM algorithms for scenario shown in Figure 14. Both algorithms have comparably low error, with higher errors found just after mode transitions.

seconds around landing.

The run-time of the prediction algorithm requires more consideration than that of the tracking algorithm. While Table ?? shows that tracking can take several seconds, this computational cost is spread out over a long time period if running the algorithm online. However, running a prediction online still requires computation from the current time to the estimated arrival time, which is on the order of the total track time for early predictions. That is, a single prediction made at a large distance from the airport can take upwards of 4** seconds. This may be quite sufficient for most applications, but extending the complexity of the prediction algorithm at each time step may lead to predictions taking longer than is available for online algorithms. Such concerns must be considered when extending the ETA prediction algorithm to larger number of modes or specialized scenarios. In this section, a framework for target tracking and ETA prediction has been presented to obtain comparisons between available hybrid state estimation models. The advantages of the IMM algorithm and using autonomous transitions, in terms of ETA predictive capability, are noted. Also, we find that IMM is clearly superior to particle filters in terms of run-time.

V. Conclusions

The problems of tracking aircraft near an airport and predicting arrival times have been formulated in a single stochastic hybrid linear system framework. Hybrid state estimation algorithms, IMM and particle filtering, as well as mode transition models, Markov and autonomous transitions, are considered for this framework. All methods perform reasonably well in target tracking. Particle filters using 100 particles perform best in terms of RMS error, while IMM performs just behind the $N = 100$ particle filter and ahead of particle filters with fewer particles. IMM algorithms run much faster than particle filters of any number; hence it is concluded that IMM with either mode transition model is a more efficient scheme for target tracking. Prediction of ETA is a difficult task for algorithms using particle filters or Markov models; the result is that only the IMM algorithm incorporating autonomous transitions can effectively predict ETA. Errors are found to be on the order of 30 seconds at a time horizon of 15 minutes.

These results, however, are highly sensitive to the nominal trajectory created from observed data and the simulated trajectory, which is a noisy version of the nominal. It is imperative in future work to improve the parameters used in the hybrid linear system model. That is, we intend to develop parameter sets that are

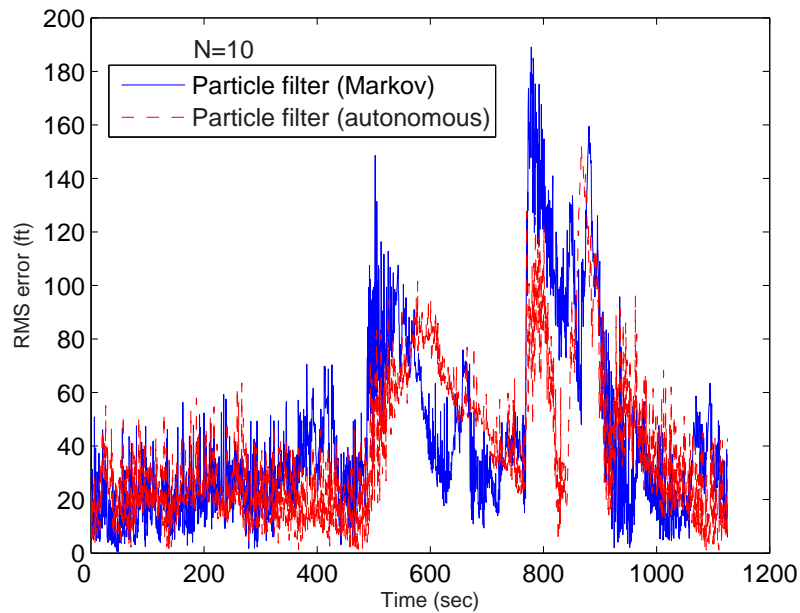


Figure 16. RMS position errors for particle filter algorithms (10 particles) for scenario shown in Figure 14. Again, higher errors are an indication of a mode transition.

tailored to specific scenarios (e.g. - morning flights on Sundays or congested periods after runway direction swaps). This update will lead to more accurate tracking and ETA prediction by appropriately choosing nominal trajectories for any observed data. The idea of using autonomous transitions to model discrete mode changes in hybrid systems can be used in other areas of air traffic control. For example, aircraft in enroute airspace generally follow jetways. Conformance monitoring algorithms can use techniques such as autonomous transitions to better detect unsafe mode changes or other unexpected maneuvers.

References

- ¹K. ANDERSSON, F. CARR, E. FERON, and W.D. HALL. Analysis and modeling of ground operations at hub airports. In *Proceedings of the 3rd USA/Europe Air Traffic Management R&D Seminar*, Napoli, Italy, June 2000.
- ²Y. BAR-SHALOM and T.F. FORTMANN. *Tracking and Data Association*. Academic Press, 1988.
- ³Y. BAR-SHALOM and X.R. LI. *Estimation and Tracking: Principles, Techniques and Software*. Artech House, Boston, MA, 1993.
- ⁴A.P. BLOM, R.A. HOGENDOORN, and B.A. VAN DOORN. *Design of a multisensor tracking system for advanced air traffic control*, volume 2 of *Multitarget-Multisensor Tracking: Application and Advances*, pages 31–63. Artech House, 1990.
- ⁵H.A.P. Blom and Y. Bar-Shalom. The interacting multiple model algorithm for systems with markovian switching coefficients. *IEEE Transactions on Automatic Control*, 33(8):780–783, August 1988.
- ⁶S. FUNIAK, L.J. BLACKMORE, and B.C. WILLIAMS. Gaussian particle filtering for concurrent hybrid models with autonomous transitions. *Journal of Artificial Intelligence Research (submitted)*, 2004.
- ⁷M.W. HOFBAUR and B.C. WILLIAMS. Mode estimation of probabilistic hybrid systems. In Claire Tomlin and Mark R. Greenstreet, editors, *Hybrid Systems: Computation and Control*, volume 2289 of *Lecture Notes in Computer Science*, Stanford, CA, March 2002. Springer.
- ⁸X. KOUTSOUKOS, J. KURIEN, and F. ZHAO. Estimation of distributed hybrid systems using particle filtering methods. In *Hybrid Systems: Computation and Control*, volume 2623 of *Lecture Notes in Computer Science*, pages 298–313, Prague, April 2003. Springer.
- ⁹J. KROZEL and S. PENNY. Comparison of algorithms for synthesizing weather avoidance routes in transition airspace. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Providence, Rhode Island, Aug. 2004.
- ¹⁰G. LAINIOTIS. Partitioning: A unifying framework for adaptive systems i: Estimation. Number 64, August 1976.
- ¹¹X.R. Li and Y. Bar-Shalom. Design of an Interacting Multiple Model Algorithm for air traffic control tracking. *IEEE Transactions on Control Systems Technology*, 1(3):186–194, September 1993.
- ¹²W.H. LIN and R.L. BERTINI. Modeling schedule recovery processes in transit operations for bus arrival time prediction. *Journal of Advanced Transportation*, 38(3):347–365, 2004.

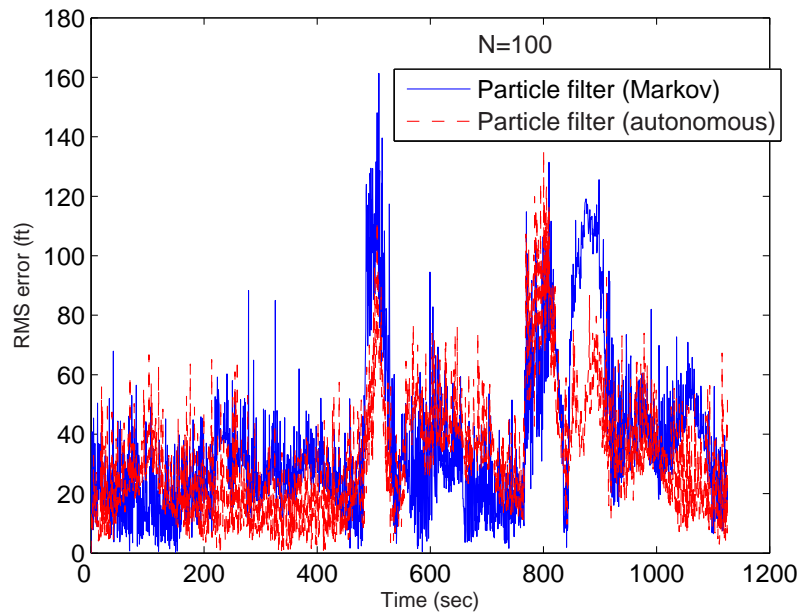


Figure 17. RMS position errors for particle filter algorithms (100 particles) for scenario shown in Figure 14. Errors are slightly lower for 100 particles than for 10 particles.

¹³A. MATHEJA, C. ZIMMERMANN, M. LARNOULD, M. BERNARD, and M. MISKA. Wabis - an information and operating system for inland waterways. In *Second European Inland Waterway Navigation Conference*, Budapest, June 2001.

¹⁴E. MAZOR, A. AVERBUCH, Y. BAR-SHALOM, and J. DAYAN. Interacting multiple model methods in tracking: A survey. *IEEE Transactions on Aerospace and Electronic Systems*, 34(1):103–123, 1998.

¹⁵National Research Council. *An Experimental Study on Real Time Bus Arrival Time Prediction with GPS Data*, volume 2, Washington D.C., January 1999.

¹⁶J. PRETE and J. MITCHELL. Safe routing of multiple aircraft flows in the presence of time-varying weather data. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Providence, Rhode Island, Aug. 2004.

¹⁷R. RAJBHANDARI. *Bus Arrival Time Prediction using Stochastic Time Series and Markov Chains*. PhD dissertation, New Jersey Institute of Technology, Interdisciplinary Program in Transportation, January 2005.

¹⁸J. B. SHAH. Fedex's hub of supply chain activity. *Electronics Supply & Manufacturing*, July 2001.

¹⁹A. SHALABY and A. FARHAN. Prediction model of bus arrival and departure times using avl and apc data. *Journal of Public Transportation*, 7(1):41–61, 2004.

²⁰D. SWORDER and J.E. BOYD. *Estimation Problems in Hybrid Systems*. Cambridge University Press, 1999.

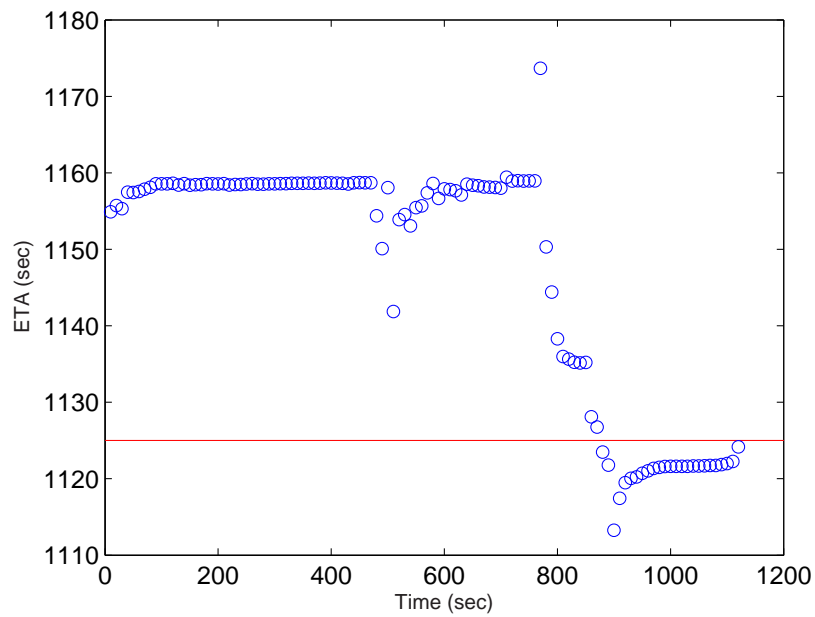


Figure 18. ETA predictions (o's) as a function of time and actual time of arrival (horizontal line). Data based on simulated scenario. Predictions are off by about 30 seconds at a time horizon of 15 minutes. The predictions become more accurate closer to the end of the trajectory because there is less uncertainty in the future, both in terms of discrete mode and continuous states.