18th International Conference on Information Fusion
Washington, DC - July 6-9, 2015

# An Efficient Algorithm for Aircraft Conflict Detection and Resolution Using List Viterbi Algorithm

**Vesselin P. Jilkov  X. Rong Li  Jeffrey H. Ledet**
Department of Electrical Engineering
University of New Orleans
New Orleans, LA 70148, U.S.A.
Email: {vjilkov, xli, jhledet}@uno.edu

***Abstract*—This paper addresses the air traffic control problem of conflict detection and resolution (CDR) under intent uncertainty, in a multiple model (MM) trajectory information processing framework. The conflict detection is based on a predicted probability of conflict. The problem of conflict resolution (CR) is formulated as one of a chance-constrained model predictive control, whereby the constraint is imposed on the probability of conflict, set to guarantee a desired level of safety (in a probabilistic sense). An efficient algorithm for CDR is proposed that utilizes a List Viterbi Algorithm for finding an optimal, conflict-free, MM-maneuver sequence for CR. The capability and computational efficiency of the proposed algorithm are demonstrated by simulation of several "sense-and-avoid" UAV encounter scenarios that involve one or more intruders and horizontal and vertical MM-maneuvers for CR.**

***Index Terms*—Air traffic control, NextGen, conflict detection and resolution, collision avoidance, multiple model, list Viterbi algorithm.**

## I. Introduction

Safety is the primary concern of Air Traffic Management (ATM). It includes ensuring appropriate separation between aircraft in an increased density of future traffic. A conflict occurs when a separation criterion is violated. Examples of standard separation criteria are the minimum vertical separation between flight levels and the minimum horizontal separation between aircraft at the same flight level. Timely and reliable detection of potential conflicts is vital for preventing collisions. Conflict detection (CD) is performed using predicted aircraft future trajectories. If a potential conflict is detected, the trajectories of the aircraft involved in the conflict are re-planned for conflict resolution (CR). With reference to the look-ahead time horizon conflict detection and resolution (CDR) is typically performed at three different levels (see e.g., [1]): i) long range, where flight plans and airline schedules are composed (e.g., daily) to ensure that airport and sector capacities are not exceeded; ii) mid-range (of the order of tens of minutes), where the flight plans are modified online based on actual flight information to ensure adequate aircraft separation; iii) short range over horizons of seconds to minutes, carried out both on the ground at the ATC level and on board the aircraft at the Flight Management System level. A major challenge of the ATM is to achieve autonomous CDR of individual aircraft at short and mid-ranges, without the intervention of air traffic controllers [2].

Research supported in part by NASA/LEQSF(2013-15)-Phase3-06 through grant NNX13AD29A.

The large number of existing methods to automate CDR can be classified in different ways, including broadly grouped into three categories: geometric, force field, and probabilistic [3]. In this paper, we follow the probabilistic approach which provides general and systematic means to account for and overcome the uncertainties inherent in the CDR problem. The majority of probabilistic CDR methods are based on computing the probability of conflict (PC) – the probability that the distance between a pair of aircraft becomes smaller than a specified minimum separation distance [1], [4], [5]. However, the exact computing of PC is intractable analytically and the known methods resort to some sort of approximation. A more detailed discussion on the approximation techniques for PC can be found in [6]. In [6] we have shown that in an advanced multiple model (MM) trajectory prediction framework such as NextGen (see e.g., [4], [7]–[10]), the separation vector between two aircraft has a Gaussian mixture (GM) distribution, and we have proposed an efficient randomized algorithm for estimating PC by utilizing the information from MM aircraft trajectory prediction. In this paper we use our method from [6] for the purpose of MM-based CR and collision avoidance (CA) under intent uncertainty.

A considerable research effort for CR and CA using probabilistic models has been going on in the recent years (see e.g., [11]–[20]). [20] formulates the problem for cooperative midair CR, addressing aircraft-aircraft and aircraft-weather conflicts, as a classic stochastic optimal control problem and proposes to compute an approximate solution through approximating the stochastic differential equation of motion by a Markov chain. The method involves time and state-space discretization, which makes it very intense computationally. Another, rigorous approach, addressing CA, is based on modeling the problem as a Markov decision process (MDP, or more generally, as a partially observed MDP) [12], [16]–[19]. However, discrete state POMDP algorithms require discretization of

the high-dimensional continuous state-space of the problem and impose a prohibitive computational burden for practical implementation. One idea to circumvent the computational problems due to discretization of the state space is to use Monte Carlo (MC) methods. Within the POMDP model this idea was implemented in, e.g., [17], where the Monte Carlo Value Iteration algorithm was employed to cope with the continuous state-space. A Markov chain MC optimization algorithm similar to the Simulated Annealing was proposed in [11] for sampling optimizers for a finite-horizon open-loop control (similar to model predictive control (MPC)), and, furthermore, a solution to the problem in a stochastic MPC formulation was proposed in [13], based on Sequential MC. [14] also uses Sequential MC for a stochastic MPC formulation, which also includes probabilistic constraints on the optimal trajectories and control input. Even though the Monte Carlo methods avoid regular discretization of the state space, their computational complexity is still very high.

As discussed above, while many theoretically rigorous and elegant solutions are available in the literature, their practical implementation requires tremendous computational resources. For many practical applications, e.g., a UAV "sense-and-avoid" encounter, the computational resources are quite limited and those methods are infeasible at present. From the practical point of view, the primary goal in a close near mid-air aircraft conflict (NMAC) encounter is to guarantee safety and trajectory optimization is secondary – once the conflict is resolved, return to the original flight path, or re-routing to the original next way point can be done via standard navigational procedures. That is why our approach takes a middle ground—we formulate the CR (CA) problem as one of optimizing the control input (not the flight path) and impose hard constraints on the flight path to provide safety, as a trade off between optimality and computational complexity at a guaranteed level of safety. We use the MM trajectory prediction framework, as described in [6], and seek to minimize a CR-maneuver cost over the trellis graph of possible model sequences over a finite look-ahead time horizon, subject to a hard constraint on the probability of conflict. For this problem, we propose an algorithm for CR that finds efficiently an optimal maneuver sequence subject to the safety constraint. The proposed algorithm is an extension of the unconstrained sequential List Viterbi algorithm of [21] to the probabilistically constrained case of our CR formulation. The capability and computational efficiency of the proposed algorithm are demonstrated by simulation of several "sense-and-avoid" UAV encounter scenarios involving one or more intruders and horizontal and vertical CA maneuvers.

## II. Background

This section outlines our MM trajectory prediction framework and computation of the predicted PC which are needed for the remainder of this paper. It is based on [6].

### A. Predicted Probability of Conflict

PC is the probability that the distance between two aircraft is smaller than a specified minimum separation distance. As specified by the current aviation standards, the horizontal (2D) and vertical separation distances (1D) are commonly treated separately, [1], [22], which amounts to a cylindrical protected zone. For the sake of generality we consider a 3D ellipsoidal protected zone (proposed in [7], [9]) where the horizontal and vertical separations are treated jointly. More specifically, let $A$ and $B$ denote two aircraft with position vectors $x_A = (\mathsf{x}_A, \mathsf{y}_A, \mathsf{z}_A)'$ and $x_B = (\mathsf{x}_B, \mathsf{y}_B, \mathsf{z}_B)'$ in an inertial (East-North-Up) Cartesian coordinate system $O$xyz, and let the distance vector be $\rho_{AB} = x_A - x_B$. If $\lambda_{\mathsf{xy}}$ and $\lambda_{\mathsf{z}}$ are the given horizontal and vertical separation thresholds, respectively, the "weighted" distance vector is $\Lambda\rho_{AB}$, where $\Lambda = \mathrm{diag}\{1/\lambda_{\mathsf{xy}}, 1/\lambda_{\mathsf{xy}}, 1/\lambda_{\mathsf{z}}\}$, and the ellipsoidal protected zone is defined as

$$\mathcal{R}^{\lambda} = \left\{\rho \in \mathbb{R}^3 : \|\Lambda\rho\| \le 1\right\} \tag{1}$$

where $\|a\| = (a'a)^{\frac{1}{2}}$ is the Euclidean norm of a vector $a$.

If $f_{\rho_{AB_k}}(\rho)$ is the probability density function (PDF) of the distance vector $\rho_{AB_k}$ at time $k$, then the instantaneous PC is

$$PC_k = P\left\{\left\|\Lambda\rho_{AB_k}\right\| \le 1\right\} = \int_{\mathcal{R}^{\lambda}} f_{\rho_{AB_k}}(\rho)d\rho \tag{2}$$

and the maximal PC over a time interval $(k, k+N]$ is

$$PC_{(k,k+N]} = \max_{0<n\le N} PC_{k+n} \tag{3}$$

In our CDR logic, $k$ is the "current" time and $N$ is the prediction time horizon. The corresponding predicted PC is computed via (2) and (3), respectively, based on predicted densities $\hat{f}_{\rho_{AB_{k+n}}}(\rho|z^k) \approx f_{\rho_{AB_{k+n}}}(\rho)$, $n = 1, 2, \ldots, N$, where $z^k = \{z_1, \ldots, z_k\}$ is the available sensor data.

### B. Multiple Model Trajectory Prediction

Apparently, the accuracy of the predicted PC relies heavily on that of the predicted aircraft state distributions. Commonly in the literature they are assumed (or approximated as) Gaussian. In reality, however, this approximation can be quite inaccurate in the presence of significant uncertainties, such as aircraft intent and flight mode (e.g., takeoff, level cruise, landing, ascent, descent, turning). Multiple model estimation is the state-of-the-art approach to meet the challenges of trajectory prediction, which is particularly effective with handling multiple modes of operation [23].

In an MM framework, the typical flight modes of a civilian aircraft can be described, e.g., by the following kinematic models [24]: constant velocity (CV), constant acceleration (CA) and constant turn (CT) in the horizontal plane, and constant height (CH) and constant climb/descent (CD) in the vertical direction. A flight mode is then modeled as a combination of a horizontal and a vertical model. Flight dynamics is described by a hybrid system (HS) model, including the model set and the transition between models describing flight mode changes. If no information about the aircraft intent (e.g.,

flight plan and waypoints) is available, the transition model is purely a priori and most often Markov/semi-Markov models are used. If some intent information is available, the transition model can be adapted based on this information and the current or predicted state [4], [8]–[10], [25]. Furthermore, the intent information can be significantly better incorporated for MM trajectory estimation within the framework of the variable structure MM (VSMM) estimation [26].

Without loss of generality, we consider the Markov Jump Linear System (MJLS) model:

$$x_k = F_{m_k} x_{k-1} + G_{m_k} w_k \tag{4}$$

$$z_k = H_{m_k} x_k + v_k \tag{5}$$

where $k = 1, 2, \ldots$ is the time index, $x_k \in \mathbb{R}^{n_x}$ is the continuous kinematic state, $m_k \in \mathbb{M} = \left\{m^{(1)}, m^{(2)}, \ldots, m^{(M)}\right\}$ is the discrete modal state, $z_k \in \mathbb{R}^{n_z}$ is the sensor measurement, and $w_k \sim \mathcal{N}(0, Q_k)$ and $v_k \sim \mathcal{N}(0, R_k)$ are mutually independent white process and measurement noises, respectively. The model sequence $\langle m_k \rangle$ is a Markov chain with transition and initial probabilities, respectively:

$$P\{m_k = m^{(j)} | m_{k-1} = m^{(i)}\} = \pi_{ij} \tag{6}$$

$$P\{m_0 = m^{(i)}\} = \mu_0^{(i)} \tag{7}$$

We assume a probabilistic trajectory prediction that is concerned with finding (or approximating) the PDFs $f(x_{k+n}|z^k)$, $n = 1, \ldots, N$, where $k$ is the current time, $z^k = \{z_1, \ldots, z_k\}$ is the available sensor data, and $N$ is the look-ahead horizon for prediction. Typically, an MM filter (e.g., GPB, IMM or VSMM [23]) provides state estimates $\hat{x}_{k|k}^{(i_k)}$, associated covariances $P_{k|k}^{(i_k)}$ and model probabilities $\mu_{k|k}^{(i_k)}$, $i_k = 1, \ldots M$. The filter density is approximated as a Gaussian mixture (GM)

$$f(x_k|z^k) = \sum_{i_k=1}^{M} \mu_{k|k}^{(i_k)} \mathcal{N}\left(x_k; \hat{x}_{k|k}^{(i_k)}, P_{k|k}^{(i_k)}\right) \tag{8}$$

Furthermore, the prediction is based on the motion model given by (4) and (6) with $\hat{x}_{k|k}^{(i_k)}$, $P_{k|k}^{(i_k)}$, and $\mu_{k|k}^{(i_k)}$ being the "initial" condition.

Let a model sequence in the time interval $[k, k+N]$ be denoted by

$$\mathfrak{M}_k^{k+N} \triangleq (m_k^{(i_k)}, \ldots, m_{k+N}^{(i_{k+N})}) \in \mathbb{M}^{N+1} \tag{9}$$

where $i_{k:k+N} \triangleq (i_k, \ldots, i_{k+N})$ is the underlying sequence of model indices.

Then

$$f(x_{k+N}|z^k) = \sum_{i_{k:k+N}} f(x_{k+N}|\mathfrak{M}_k^{k+N}, z^k) P\{\mathfrak{M}_k^{k+N}|z^k\} \tag{10}$$

where

$$f(x_{k+N}|\mathfrak{M}_k^{k+N}, z^k) = \mathcal{N}\left(x_{k+N}; \hat{x}_{k+N|k}^{(i_{k:k+N})}, P_{k+N|k}^{(i_{k:k+N})}\right) \tag{11}$$

$$P\{\mathfrak{M}_k^{k+N}|z^k\} = \mu_{k|k}^{(i_k)} \pi_{i_k i_{k+1}} \ldots \pi_{i_{k+N-1} i_{k+N}} \tag{12}$$

TABLE I
RANDOMIZED ALGORITHM FOR ESTIMATING $PC_{k+n}$ [6]

- Initialization
  - Choose $R_{MC}$ = number of MC runs
  - Set $\widehat{PC}_{k+n} = 0$
- For $r = 1, \ldots, R_{MC}$
  - Sample random index $l \sim \left\{\mu_{AB_{k+n}}^{(l)} : l = 1, \ldots, M_{AB_{k+n}}\right\}$
  - Sample random vector $\rho \sim \mathcal{N}\left(\rho; \hat{\rho}_{AB_{k+n}}^{(l)}, P_{AB_{k+n}}^{(l)}\right)$
  - If $\|\Lambda\rho\| \leq 1$ then $\widehat{PC}_{k+n} = \widehat{PC}_{k+n} + 1$
- Set $\widehat{PC}_{k+n} = \widehat{PC}_{k+n}/R_{MC}$

and $\hat{x}_{k+N|k}^{(i_{k:k+N})}, P_{k+N|k}^{(i_{k:k+N})}$ are obtained recursively by the Kalman filter prediction equations for each model sequence $\mathfrak{M}_k^{k+N}$ with the initial $\hat{x}_{k|k}^{(i_k)}, P_{k|k}^{(i_k)}$.

For the MJLS, the predictor (10)–(12) is exact, except for the MM filter part. Its computation increases exponentially as the time horizon $N$ increases, but $N$ is fixed. It may be feasible for not large $N$. Otherwise, the computation can be reduced by some of the well known soft (GPB, IMM) or hard decision (B-best selection) techniques [23] to limit the size of the model-switching tree. For example, [4], [8], [9] all use directly the IMM scheme for MM trajectory prediction.

### C. Computing the Predicted PC

It was shown in [6] that in the MM trajectory prediction setting, outlined in the previous section, the predicted PDFs $f_{\rho_{AB_{k+n}}}(\rho)$ of the distance vectors $\rho_{AB_{k+n}} = x_{A_{k+n}} - x_{B_{k+n}}$ between two aircraft $A$ and $B$, for $n = 1, 2, \ldots, N$, are Gaussian mixtures:

$$f_{\rho_{AB_{k+n}}}(\rho) = \sum_{l=1}^{M_{AB_{k+n}}} \mu_{AB_{k+n}}^{(l)} \mathcal{N}\left(\rho; \hat{\rho}_{AB_{k+n}}^{(l)}, P_{AB_{k+n}}^{(l)}\right) \tag{13}$$

where $\hat{\rho}_{AB_{k+n}}^{(l)} = \hat{x}_{A_{k+n|k}}^{(l)} - \hat{x}_{B_{k+n|k}}^{(l)}$ and $P_{AB_{k+n}}^{(l)} = P_{A_{k+n|k}}^{(l)} + P_{B_{k+n|k}}^{(l)}$.

An estimate of the instantaneous predicted $PC_{k+n}$, $n = 1, 2, \ldots, N$, can be computed by the randomized algorithm, given in Table I. Good approximation properties (in a probabilistic sense) of such randomized estimation of PC are discussed in [1], where quantitative bounds on the approximation error can be found. The high accuracy and efficiency of the algorithm were demonstrated in [6].

## III. PROPOSED CDR APPROACH

### A. Problem Formulation

We consider decentralized and non-coordinated (possibly non-cooperative) environment where an aircraft $A$ should detect possible conflicts with nearby intruders $B_1, B_2, \ldots$ under uncertainty regarding their motion intent, and generate collision-free own-flight paths.

The motion of all aircraft is modeled by the hybrid system (4). For the own aircraft $A$, the flight mode sequence $\mathfrak{M}_k^{k+N}$ is

known according to the flight plan to the next waypoint (WP) or final destination. For the other aircraft (potential intruders) the intent is not exactly known and the intent uncertainty of each potential intruder is modeled by a Markov random sequence $\langle m_k \rangle$ with known initial and transition probabilities, as given by (6) and (7).

It is assumed that at each time $k$ (current time) all filtered estimates $\hat{x}^A_{k|k}$ and $\hat{x}^{B_b}_{k|k}$, $b = 1, 2, \ldots$, are available. The CD strategy, under consideration in this paper, is to predict the GM PDFs of all intruders and, given the own aircraft mode sequence $\mathfrak{M}^{k+N}_k$, compute the predicted $PC^{(b)}_{k+n}$, $n = 1, \ldots, N;\ b = 1, 2, \ldots,$ where $N$ is the look-ahead time horizon and $b$ denotes intruder $B_b$. This can be efficiently done by our method, described in Sec. II.

A conflict alert is triggered "ON" iff

$$\max_{(b, 1 \le n \le N)} PC^{(b)}_{k+n} \ge \delta \tag{14}$$

where $\delta$ is a threshold.

Upon a conflict alert, the current mode sequence $\mathfrak{M}^{k+N}_k$ of the own aircraft needs to updated to provide safety, i.e., to guarantee $\max_{(b, 1 \le n \le N)} PC^{(b)}_{k+n} < \delta$ along the updated flight path. However, any change in the current flight mode sequence incurs extra cost in terms of fuel, flight-path deviation, time delay, etc. Let $c_{k+n}(m^{(i)}, m^{(j)})$, $i, j \in \{1, 2, \ldots, M\}$ be the cost of transition from mode $m^{(i)}$ at time $k + n - 1$ to mode $m^{(j)}$ at time $k + n$, $n = 1, 2, \ldots, N$. Then, we formulate the collision avoidance problem as the following "chance-constrained" model predictive control problem:

*Minimize:*

$$J(\mathfrak{M}^{k+N}_k) = \sum_{n=1}^{N} c_{k+n}(m^{(i_{k+n-1})}, m^{(i_{k+n})}) \tag{15}$$

*subject to:*

$$PC^{(b)}_{k+n} < \delta,\ n = 1, \ldots, N;\ b = 1, 2, \ldots \tag{16}$$

Simply put, as formulated the problem is to find the least expensive maneuver at a guaranteed safety level (in a probabilistic sense). It is clear that the formulation of the total cost, given by (15), has some limitations. While it optimizes the cost of maneuvering, it does not optimize the flight path explicitly but just guarantees that it is conflict-free. Consequently, it does not guarantee reaching the next WP, or return to the originally commanded flight-path. Nevertheless, we find strong arguments in support of this formulation: First, the primary goal in a close NMAC encounter is to guarantee safety and a trajectory optimization is secondary – once the conflict is resolved, return to the original flight path, or re-routing to the next way point can be done via standard navigational procedures. Second, including explicit optimization of the ownship trajectory in the total cost (15) leads to a dramatic increase in the complexity of the resulting stochastic optimal control problem. While many theoretically rigorous and elegant solutions are available in the literature (as discussed in the introduction), their practical implementation requires tremendous computational resources. For many practical applications, e.g., UAV "sense-and-avoid",

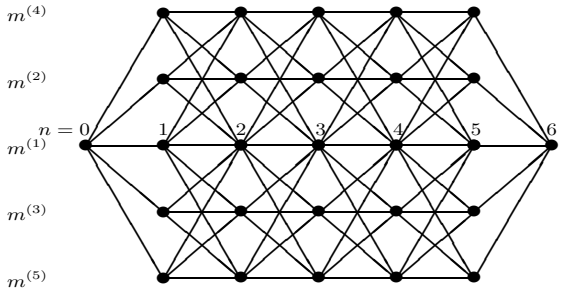


Fig. 1. Model Switching Trellis with $M = 5$ and $N = 6$.

the computational resources are quite limited and those methods are infeasible at present. In this aspect, our formulation is attractive, given the fact that we have a very efficient method to find the globally optimal solution to problem (15)–(16). This method is presented next.

### *B. Finding Optimal Solution*

The unconstrained optimization problem (15) is one of finding a best (least costly) path through a trellis digraph. Fig. 1 illustrates a typical model switching trellis (at current time $k$) with $M = 5$ models (nodes) and look-ahead time horizon $N = 6$ (i.e., ending time is $k+N$). Without loss of generality, it is assumed that at the start and end times ($k$ and $k + N$, respectively) the own aircraft is in flight mode $m^{(1)}$ (e.g., a CV motion). In our implementation (Sec. IV) we used the complete trellis with $m^{(3)}$, $m^{(5)}$ being right CT models and $m^{(2)}$, $m^{(4)}$ being their symmetrical left CT models. (Note that the depicted trellis is not complete, e.g., switching from a left CT to a right CT (and vise versa) is only possible through the CV model.)

Finding an unconstrained best path through a trellis can be easily and efficiently done via Dynamic Programming, e.g., via the popular Viterbi algorithm (VA) [27]. The constrained problem (15)–(16) is more complicated. The obvious idea of searching for a best path, after determining the feasible (conflict-free) trellis (a subset of the complete trellis) first is a dead end because it needs evaluation of PC over all paths. Since the PC evaluation is the computational bottleneck in this problem, our idea is to organize the search of the best paths in a sequential manner (in an increasing order of costs) and then check feasibility (i.e. whether a candidate is conflict-free) so that PC is only evaluated over low-cost candidate paths. Thus, our strategy is to find the best path and check its feasibility, if it is not feasible we search for the second best path and check its feasibility, if the second best is infeasible, we continue with the search for the third best past, and so on, until we find a best path which is feasible (conflict-free).

The problem of finding the $L$ best paths through directed graphs has been well studied in the area of computational geometry and many general algorithms exist. More efficient algorithms, particularly tailored to the special case of trellis graphs are available in the communications literature [21],

[28], [29], where algorithms for finding the $L$ best paths through a trellis are generally referred to as the List Viterbi Algorithms (LVAs). For the purpose of finding an optimal solution to the constrained problem (15)–(16), in principle any LVA can be used; however, the sequential LVA (SLVA) [21] best suits to our problem for two reasons: first, it finds the next best path recursively based on the previous best paths, and second, the search is organized via forward VA-like passes through the trellis, which would allow us to re-use already computed PC along the conflict-free parts of the candidate forward paths.

Next, we outline the idea of the unconstrained SLVA of [21], which later (in Sect. III-C) we extend to the constrained case of our CDR formulation.

For simplicity, let the trellis states be denoted by $i \in \{1, 2, \dots, M\}$ (instead of our MM notation $m^{(i)} \in \mathbb{M}$). Without loss of generality we assume that the initial state of the trellis at time $n = 0$ is 1. Let $\lambda_n^{(i)}$ be the minimum cost to reach state $i$ at time $n$ (from the known state 1 at time $n = 0$), and $j_n(i) \in \{1, 2, \dots, M\}$ be the state occupied at time $n-1$ by the best path into state $i$ at time $n$.

The first step of SLVA is to obtain the best path via VA. For convenience we provide the VA in Table II.

In SLVA the $l$th best path $p^{(l)}$ is found in a sequential manner, based on previously found $(l-1)$ best paths $p^{(1)},\ p^{(2)}, \dots,\ p^{(l-1)}$, sorted in a non-decreasing order of costs:

$$p^{(l)} = \text{SLVA}(p^{(1)}, p^{(2)}, \dots p^{(l-1)}),\ l = 2, 3, \dots \qquad (17)$$

where SLVA symbolizes one recursive step of the algorithm. First, $p^{(1)}$ can be found through VA, as given above. Finding $p^{(2)}$, based on $p^{(1)}$, utilizes the fact that $p^{(2)}$ has exactly one forward split point from $p^{(1)}$, i.e., after $p^{(2)}$ leaves $p^{(1)}$ at some instant and then it merges with $p^{(1)}$ at a later instant, it cannot split anymore till the end, because any other split will result in a higher cost. So, if $i_n^*$ is the state occupied by $p^{(1)}$ at time $n$, the forward cost-to-go $\lambda_n^{(i_n^*)}(2)$ of $p^{(2)}$ can be written as

$$\lambda_n^{(i_n^*)}(2) = \min\{(\lambda_{n-1}^{(i_{n-1}^*)} + c_n(i_{n-1}^*, i_n^*)), \min_{0 \le j \le M,\ j \ne i_n^*} (\lambda_{n-1}^{(j)} + c_n(j, i_n^*))\} \qquad (18)$$

The first term in (18) represents the 2nd best path to $i_n^*$ which has merged to $p^{(1)}$ no later than time $n-1$. The second term represents the 2nd best path to $i_n^*$ which has merged to $p^{(1)}$ no earlier than time $n$. In the forward pass of the algorithm, the one with the min cost remains in contention to become $p^{(2)}$ and the time of the last best merge, say $n_m$ is recorded. Then $p^{(2)}$ is the 2nd best path to time $n_m - 1$, determined by the above recursion, and is equal to $p^{(1)}$ from time $n_m$ until the end. Finding $p^{(3)}$ from $p^{(1)}$ and $p^{(2)}$ can be organized in a similar manner. A formal description of the algorithm is given in [21] and will not be provided here due to space limitation.

### C. Proposed CDR Algorithm

A formal algorithm description is given in Table III.

TABLE II
VITERBI ALGORITHM

1) Initialization: $(n = 1)$

$$\lambda_1^{(i)} = c_1(1, i),\ j_1(i) = 1; \quad i = 1, \dots, M$$

2) Recursion: $(n = 2, \dots, N-1)$

$$\lambda_n^{(i)} = \min_{1 \le j \le M} (\lambda_{n-1}^{(j)} + c_n(j, i));\ i = 1, \dots, M$$

$$j_n(i) = \arg \min_{1 \le j \le M} (\lambda_{n-1}^{(j)} + c_n(j, i))$$

3) Termination: $(n = N)$

$$\lambda_N^{(1)} = \min_{1 \le j \le M} (\lambda_{N-1}^{(j)} + c_N(j, 1))$$

$$j_N(1) = \arg \min_{1 \le j \le M} (\lambda_{N-1}^{(j)} + c_N(j, 1))$$

4) Backtracking:

$$i_N^* = 1;\ i_n^* = j_{n+1}(i_{n+1}^*),\ n = N-1, \dots, 1$$

5) Best sequence: $(1, i_1^*, \dots, i_{N-1}^*, 1)$

The CD part of the algorithm implements simply (14). In the CR part we employ the SLVA search strategy. However, a straightforward implementation, i.e, finding the next best path $p^{(l)}$ and then computing $PC_{k+n}^{(b)}(p^{(l)})$ from the beginning $n = 1$ is naive and inefficient – it would lead to repeating (possibly many times) the computation of PC along the initial portions of the $p^{(l)}$ which are common with its predecessors $p^{(1)}, p^{(2)}, \dots p^{(l-1)}$, and for which PC was already calculated (and also carry information about feasibility of these portions). In order to eliminate repeated computation of PC and utilize the feasibility information of the already tested paths, we implement the following idea. First, for each infeasible (conflicting) sequence we store the earliest conflict times (with all intruders) $n_c^{(b)}(p^{(l)}) = \min_{1 \le n \le N}\{n : PC_{k+n}^{(b)}(p^{(l)}) \ge \delta\}$. Second, for each next best path $p^{(l)}$ found we store the earliest split times with all its predecessors: $n_s^{(l|j)} = \min_{0 \le n \le N}\{n :$ $p^{(l)}$ diverges from $p^{(j)}\}$. This can be easily done within the forward pass of SLVA. Then, we search for the largest split time from a conflict-free parent subsequence $n_s^{(l|j_0)} = \max_{1 \le j \le l-1}\{n_s^{(l|j)} : n_s^{(l|j)} < n_c^{(b)}(p^{(j)})\}$. If no such time exists, then $p^{(l)}$ is infeasible, because for all predecessors of $p^{(l)}$ the conflict occurred before the split. Thus $p^{(l)}$ is rejected without computing PC. Otherwise, if $n_s^{(l|j_0)}$ is found then the portion of $p^{(l)}$ from time 1 to $n_s^{(l|j_0)}$ is known to be conflict-free, and computing $PC_{k+n}^{(b)}(p^{(l)})$, in order to check feasibility of $p^{(l)}$, can start from $n_s^{(l|j_0)} + 1$.

## IV. SIMULATION

We simulated several "sense-and-avoid" (S&A) encounter scenarios, where the own-ship (e.g., an unmanned aircraft) needs to predict the trajectories of one or more "intruders" (e.g., an airliner or other unmanned aircraft) under the uncertainty in the intruders' intents (possible maneuvers) in

TABLE III
PROPOSED CDR ALGORITHM

Initialize: $l = 1,\ p^{(1)} = (1, 1, \ldots, 1)$.
At time $k$:

**1.** *Conflict Detection:*
For $b = 1, 2, \ldots;\ n = 1, \ldots, N$
Compute $PC^{(b)}_{k+n}(p^{(l)})$
If $PC^{(b)}_{k+n}(p^{(l)}) \geq \delta$ then go to 2.
$k = k + 1$ go to 1.

**2.** *Conflict Resolution:*
Search trellis for best CR maneuver sequence:

**a.** Find conflict times $n^{(b)}_c(p^{(l)}),\ b = 1, 2, \ldots$:

$$n^{(b)}_c(p^{(l)}) = \min_{1 \leq n \leq N} \{n : PC^{(b)}_{k+n}(p^{(l)}) \geq \delta\}$$

**b.** Find next best path & split times:

$$\begin{aligned} l &= l + 1 \\ p^{(l)} &= \text{SLVA}(p^{(1)}, p^{(2)}, \ldots p^{(l-1)}) \\ n^{(l|j)}_s &= \min_{0 \leq n \leq N} \{n : p^{(l)} \text{ diverges from } p^{(j)}\}, \\ & j = 1, \ldots, l-1 \end{aligned}$$

**c.** Find largest split time from conflict-free parent subsequence:

$$n^{(l|j_0)}_s = \max_{1 \leq j \leq l-1} \{n^{(l|j)}_s : n^{(l|j)}_s < n^{(b)}_c(p^{(j)})\}$$

**d.** If $\{n^{(l|j_0)}_s\} = \emptyset$ then go to 2(a)

**e.** For $b = 1, 2, \ldots;\ n = n^{(l|j_0)}_s + 1, \ldots, N$
Compute $PC^{(b)}_{k+n}(p^{(l)})$
If $PC^{(b)}_{k+n}(p^{(l)}) \geq \delta$ then go to 2(a)
CR sequence $= p^{(l)}$, recalculate guidance from state at time $k+N$ to next WP (or destination), go to 1.

order to avoid a conflict. S&A has received a great deal of attention recently since S&A capability is required for unmanned aircraft systems (UAS) in order to allow such systems in the civil airspace [12], [18], [19].

We considered two types of CDR scenarios with horizontal and vertical resolution own-ship maneuvers, respectively.

### A. Horizontal CDR

The horizontal scenarios are set up in a horizontal plane $O$xy. The dynamic model of the own-ship $(A)$ is given by (4) with $M = 5$ models: $m^{(1)} =$ CV (constant velocity), $m^{(2)} =$ CT (constant turn) with $\omega = 1\,^{\circ}/s$ (left turn), $m^{(3)} =$ CT with $\omega = -1\,^{\circ}/s$ (right turn), $m^{(4)} =$ CT with $\omega = 2\,^{\circ}/s$, $m^{(5)} =$ CT with $\omega = -2\,^{\circ}/s$, where $\omega$ denotes the turn rate. The CV and CT models are [24]:

$$x_k = \begin{bmatrix} 1 & \frac{\sin \omega_k T}{\omega_k} & 0 & -\frac{1-\cos \omega_k T}{\omega_k} \\ 0 & \cos \omega_k T & 0 & -\sin \omega_k T \\ 0 & \frac{1-\cos \omega_k T}{\omega_k} & 1 & \frac{\sin \omega_k T}{\omega_k} \\ 0 & \sin \omega_k T & 0 & \cos \omega_k T \end{bmatrix} x_{k-1} + G_k w_k \tag{19}$$

where $x_k = [\ \mathrm{x}\ \ \dot{\mathrm{x}}\ \ \mathrm{y}\ \ \dot{\mathrm{y}}\ ]'$ is the state vector, $T$ is the sampling interval, $\omega_k \in \{\omega^{(1)}, \ldots, \omega^{(5)}\}$, and $w_k$ is the process noise. Note that $\omega^{(1)} = 0$ because CV $= \lim_{\omega \to 0}$ CT$(\omega)$.

The model transition costs used are

$$c(i, j) = |\omega^{(j)} - \omega^{(i)}|,\ i, j = 1, \ldots, 5 \tag{20}$$

and $m^{(1)}$ is the initial and final trellis state (as in Fig. 1). The look-ahead time horizon used is $N = 10$.

Each one of the intruders $(B)$ is modeled through the hybrid system (4), (6), (7), and has three models: $m^{(1)} =$ CV, $m^{(2)} =$ CT with $\omega = 1\,^{\circ}/s$, and $m^{(3)} =$ CT with $\omega = -1\,^{\circ}/s$. The flight mode initial and transition probabilities are $\mu^{(1)}_0 = 0.8,\ \mu^{(2)}_0 = 0.1,\ \mu^{(3)}_0 = 0.1$ and $\pi_{11} = 0.8,\ \pi_{12} = 0.1,\ \pi_{21} = 0.9,\ \pi_{22} = 0.1,\ \pi_{31} = 0.9,\ \pi_{32} = 0.0$.

The values of the other parameters of the models used in the simulation are: $T = 5\,\text{sec}$, the process noise covariance $Q = q^2 I$ with $q = 5m/s^2$, the number of MC samples to compute PC (as given in Table I) $R_{MC} = 1000$, the horizontal separation distance threshold $\lambda_{\text{xy}} = 5km$, and the threshold on PC for conflict alert $\delta = 10^{-3}$.

Figures 2 and 3 show two scenarios with one intruder (top) and two intruders (bottom), respectively. Integer labels indicate time steps. The uncertainty in intruders' trajectories is illustrated by scatter plots sampled from the predicted mixtures, where a red zone indicates the intruders' predicted density at the time that a conflict is deemed to occur. A blue dashed line shows the planned own-ship path to the next waypoint, and a blue thick line shows the minimum cost, conflict-free path computed by the proposed CDR algorithm. The figures illustrate the capability of the algorithm to successfully resolve a conflict with the minimum maneuver cost. Each scenario was repeated 5000 times with random maneuver sequences of the intruder(s) and no conflict occurred with the own-ship minimum maneuver cost trajectory.

### B. Vertical CDR

For simplicity, the vertical scenarios are set up in a vertical plane $O$xz. The dynamic model of the own-ship $(A)$ is given by (4) with $M = 5$ models: $m^{(1)} =$ CH (constant height), i.e., with zero vertical velocity, $m^{(2)} =$ CC (constant climb) with $v^{(2)} = 10m/s$, $m^{(3)} =$ CD (constant descent) with $v^{(3)} = -10m/s$, $m^{(4)} =$ CC with $v^{(4)} = 20m/s$, $m^{(5)} =$ CD with $v^{(5)} = -20m/s$, where $v$ is the vertical velocity and the nearly CH/CC/CD model is defined as

$$x_k = \begin{bmatrix} 1 & T & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} x_{k-1} + \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} v_k + \begin{bmatrix} \frac{T^2}{2} & 0 \\ T & 0 \\ 0 & \frac{T^2}{2} \end{bmatrix} w_k \tag{21}$$

where $x_k = [\ \mathrm{x}\ \ \dot{\mathrm{x}}\ \ \mathrm{z}\ ]'$ is the state vector, $T$ is the sampling interval, $v_k \in \{v^{(1)}, \ldots, v^{(5)}\}$, and $w_k$ is the process noise. The model transition costs used are

$$c(i, j) = |v^{(j)} - v^{(i)}|,\ i, j = 1, \ldots, 5 \tag{22}$$

and $m^{(1)}$ is the initial and final trellis state (as in Fig. 1). The look-ahead time horizon used is $N = 10$.

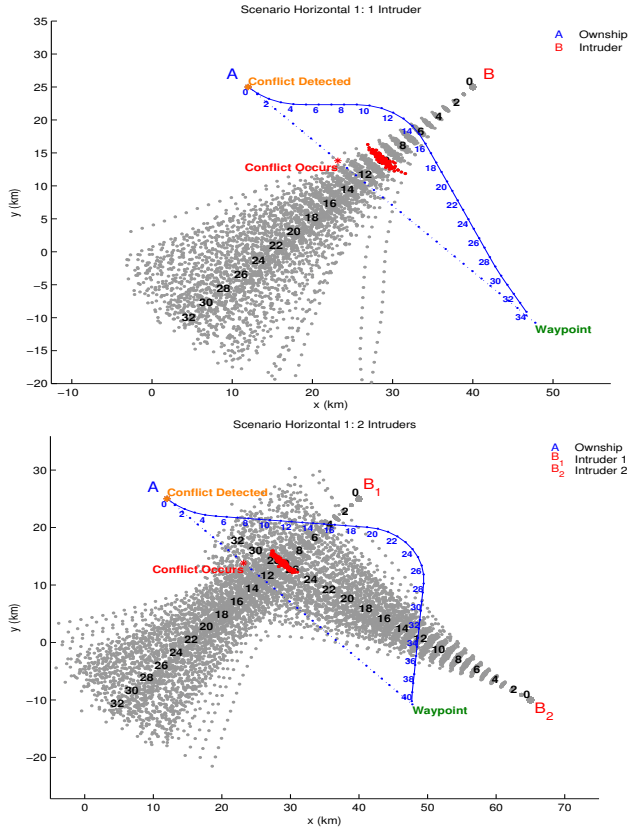


Fig. 2. Horizontal Scenario (H1): One Intruder (top), Two Intruders (bottom).

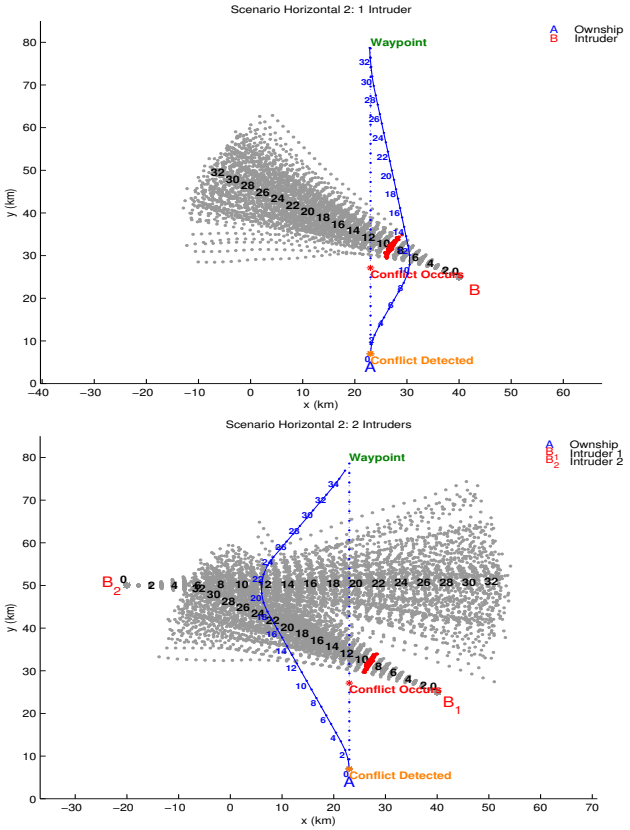


Fig. 3. Horizontal Scenario (H2): One Intruder (top), Two Intruders (bottom)

Each one of the intruders $(B)$ is modeled through the hybrid system (4), (6), (7), and has three models: $m^{(1)} = \text{CH}$ with $v^{(1)} = 0m/s$, $m^{(2)} = \text{CC}$ with $v^{(2)} = 10m/s$, and $m^{(3)} = \text{CD}$ with $v^{(3)} = -10m/s$. The flight mode initial and transition probabilities are $\mu_0^{(1)} = 0.8$, $\mu_0^{(2)} = 0.1$, $\mu_0^{(3)} = 0.1$ and $\pi_{11} = 0.8$, $\pi_{12} = 0.1$, $\pi_{21} = 0.9$, $\pi_{22} = 0.1$, $\pi_{31} = 0.9$, $\pi_{32} = 0$.

The values of the other parameters of the models used in the simulation are: $T = 5\,\text{sec}$, the process noise covariance $Q = q^2 I$ with $q = 1m/s^2$, the number of MC samples to compute PC $R_{MC} = 1000$, the vertical separation distance threshold $\lambda_z = 1000m$, and the threshold on PC for conflict alert $\delta = 10^{-3}$.

Figure 4 shows one scenario with one intruder (top) and two intruders (bottom), respectively. It illustrates the capability of the algorithm to successfully resolve a conflict with the minimum maneuver cost. The scenario was repeated 5000 times with random vertical maneuver sequences of the intruder(s) and no conflict occurred with the own-ship minimum maneuver cost trajectory.

### C. Computational Efficiency

The computational efficiency of the proposed CDR algorithm (Sec. III-C), referred to as the constrained SLVA (CSLVA), is compared with a direct implementation of the known SLVA, i.e, find the next best (unconstrained) path first, and then check its feasibility via computing the PCs over this path.

Comparative results from the above scenarios are presented in Table V. Columns "Search Depth" give the number of the best unconstrained paths in the sorted search list that were found and tested in order to obtain the best constrained (conflict free) path by each algorithm, respectively. This number is the same for both algorithms. Columns "# PC Checks" give the total number of instantaneous PCs computed by each algorithm, respectively. Column "Speedup" gives the ratio of computation times of SLVA and CSLVA, given in the fifth and sixth columns, respectively.

In all cases the proposed CSLVA appears more efficient than the direct SLVA. This is mostly due to the fact (illustrated by the comparison of columns three and four) that CSLVA dramatically reduces the number of PCs computed as compared to SLVA. The amount of improvement is scenario dependent. For the more difficult horizontal scenarios the speedup can be quite significant (as high as 4.83 times) but for the much easier vertical scenarios the speedup is not that significant (can be as low as 1.12 times). In general, it seems safe to conclude that the more difficult the scenario, the more savings on PC computation, and the greater the speedup.

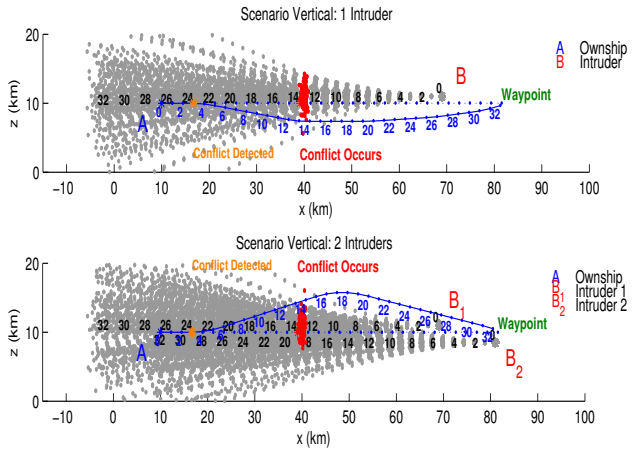


Fig. 4. Vertical Scenario (V): One Intruder (top), Two Intruders (bottom).

TABLE IV
COMPUTATIONAL PERFORMANCE: CSLVA *vs.* SLVA

| Scenario ID | Search Depth | # PC Checks | | Comp. Time (sec) | | **Speedup** (times) |
|---|---|---|---|---|---|---|
| | | SLVA | CSLVA | SLVA | CSLVA | |
| H1:1 | 7433 | 59265 | 1424 | 12.09 | 2.50 | **4.83** |
| H1:2 | 28102 | 227152 | 3796 | 38.66 | 8.38 | **4.61** |
| H2:1 | 3274 | 20864 | 368 | 5.69 | 1.29 | **4.41** |
| H2:2 | 3275 | 21003 | 379 | 5.61 | 1.80 | **3.11** |
| V:1 | 361 | 5073 | 1473 | 1.79 | 1.59 | **1.12** |
| V:2 | 1832 | 26057 | 7795 | 5.54 | 3.51 | **1.57** |

## V. CONCLUSIONS & FURTHER RESEARCH

Available optimal methods for aircraft conflict detection and resolution (CDR) have excessive computational requirements.

The proposed CDR approach and optimal algorithm can find minimum-cost conflict-free control input sequences through a very efficient List Viterbi Algorithm-based search.

The simulation results of several "sense-and-avoid" UAV encounter scenarios have demonstrated the capability of the proposed algorithm to successfully resolve conflicts that involve one or more intruders by horizontal and vertical multiple model maneuvers, as well as its computational efficiency.

Future work includes better tuning of the maneuvering cost matrix for realistic aircraft models and comprehensive statistical performance evaluation and analysis.

## REFERENCES


[1] M. Prandini, J. Hu, J. Lygeros, and S. Sastry, "A Probabilistic Approach to Aircraft Conflict Detection," *IEEE Trans. on Intelligent Transportation Systems*, vol. 1, no. 4, pp. 199–220, 2000.

[2] J. Planning and D. Office, "NextGen Avionics Roadmap, Version 2.0," September 30, 2011.

[3] J. K. Kuchar and L. C. Yang, "A Review of Conflict Detection and Resolution Modeling Methods," *IEEE Trans. on Intelligent Transportation Systems*, vol. 1, no. 4, pp. 179–189, 2000.

[4] I. Hwang and C. E. Seah, "Intent-Based Probabilistic Conflict Detection for the Next Generation Air Transportation System," *IEEE Proceedings*, vol. 96, no. 12, pp. 2040–2059, 2008.

[5] R. A. Paielli, H. Erzberger, D. Chiu, and K. R. Heere, "Tactical Conflict Alerting Aid for Air Traffic Controllers," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 32, no. 1, 2009.

[6] V. P. Jilkov, X. R. Li, and J. H. Ledet, "Improved Estimation of Conflict Probability for Aircraft Collision Avoidance," in *Proc. 2014 International Conf. on Information Fusion*, Salamanca, Spain, July 2014.

[7] H. A. P. Blom and G. J. Bakker, "Conflict Probability and Incrossing Probability in Air Traffic Management," in *Proc. Conference on Decision and Control*, Las Vegas, Nevada, U.S.A., Dec. 2002.

[8] J. L. Yepes, I. Hwang, and M. Rotea, "New Algorithms for Aircraft Intent Inference and Trajectory Prediction," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 30, no. 2, pp. 370–382, 2007.

[9] W. Liu and I. Hwang, "Probabilistic Aircraft Trajectory Prediction and Conflict Detection for Air Traffic Control," *AIAA Journal on Guidance, Control and Dynamics*, vol. 34, no. 6, pp. 1779–1789, 2011.

[10] Y. Lu and X. R. Li, "Intent Based Trajectory Prediction by Multiple Model Prediction and Smoothing," in *AIAA Guidance, Navigation, and Control Conference*, Kissimmee , FL, USA, Jan. 2015.

[11] A. L. Visintini, W. Glover, J. Lygeros, and J. Maciejowski, "Monte Carlo Optimization for Conflict Resolution in Air Traffic Control," *IEEE Trans. on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 470–482, 2006.

[12] S. Temizer, M. J. Kochenderfer, L. P. Kaelbling, T. Lozano-Perez, and J. K. Kuchar, "Unmanned Aircraft Collision Avoidance using Partially-Observable Markov Decision Processes," MIT, Lincoln Laboratory, Project Report ATC-356, 2009.

[13] N. Kantas, J. Maciejowski, and A. Lecchini-Visintini, "Sequential Monte Carlo for Model Predictive Control," in *Nonlinear Model Predictive Control*. Springer, 2009, pp. 263–273.

[14] L. Blackmore, M. Ono, A. Bektassov, and B. C. Williams, "A Probabilistic Particle-Control Approximation of Chance-Constrained Stochastic Predictive Control," *IEEE Trans. on Robotics*, vol. 26, no. 3, pp. 502–517, 2010.

[15] P.-J. Nordlund and F. Gustafsson, "Probabilistic Noncooperative Near Mid-Air Collision Avoidance," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 47, no. 2, pp. 1265–1276, 2011.

[16] J. P. Chryssanthacopoulos and M. J. Kochenderfer, "Accounting for State Uncertainty in Collision Avoidance," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 4, pp. 951–960, 2011.

[17] H. Bai, D. Hsu, M. J. Kochenderfer, and W. S. Lee, "Unmanned Aircraft Collision Avoidance using Continuous-State POMDPs," in *Robotics: Science and Systems*, Los Angeles, CA, 2011.

[18] M. J. Kochenderfer, J. E. Holland, and J. P. Chryssanthacopoulos, "Next Generation Airborne Collision Avoidance System," *Lincoln Laboratory Journal*, vol. 19, pp. 17–33, 2012.

[19] M. J. Kochenderfer, J. P. Chryssanthacopoulos, and R. E. Weibel, "A New Approach for Designing Safer Collision Avoidance Systems," *Air Traffic Control Quarterly*, vol. 20, no. 1, pp. 27–45, 2012.

[20] W. Liu and I. Hwang, "Probabilistic Aircraft Midair Conflict Resolution Using Stochastic Optimal Control," *IEEE Trans. on Intelligent Transportation Systems*, vol. 15, no. 1, pp. 37–46, 2014.

[21] N. Seshadri and C.-E. Sundberg, "List Viterbi Decoding Algorithms with Applications," *IEEE Trans. on Communications*, vol. 42, no. 2/3/4, pp. 313–323, 1994.

[22] R. Paielli and H. Erzberger, "Conflict Probability Estimation for Free Flight," *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 3, pp. 588–596, 1997.

[23] X. R. Li and V. P. Jilkov, "Survey of Maneuvering Target Tracking. Part V: Multiple-Model Methods," *IEEE Trans. Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1255–1321, Oct. 2005.

[24] ——, "Survey of Maneuvering Target Tracking. Part I: Dynamic Models," *IEEE Trans. Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1333–1364, Oct. 2003.

[25] R. Rastgoufard, V. P. Jilkov, and X. R. Li, "Incorporating World Information into the IMM Algorithm via State-Dependent Value Assignment," in *Proc. 2014 International Conf. on Information Fusion*, Salamanca, Spain, July 2014.

[26] X. R. Li and Y. Bar-Shalom, "Multiple-Model Estimation with Variable Structure," *IEEE Trans. Automatic Control*, vol. AC-41, no. 4, pp. 478–493, Apr. 1996.

[27] G. D. Forney, "The Viterbi Algorithm," *Proc. IEEE*, vol. 61, no. 3, pp. 268–278, Mar. 1973.

[28] C. Nill and C.-E. Sundberg, "List and Soft Symbol Output Viterbi Algorithms: Extensions and Comparisons," *IEEE Trans. on Communications*, vol. 43, no. 2/3/4, pp. 277–287, 1995.

[29] M. Roder and R. Hamzaoui, "Fast Tree-Trellis List Viterbi Decoding," *IEEE Trans. on Communications*, vol. 54, no. 3, pp. 453–461, 2006.