

Interactive Multiple Model Hazard States Prediction for Unmanned Aircraft Systems (UAS) Detect and Avoid (DAA)

Adriano Canolla* Michael B. Jamoom,[†] and Boris Pervan[‡]

Illinois Institute of Technology

This research aims at improving Detect and Avoid (DAA) functions in Unmanned Aircraft Systems (UAS) using multiple model estimation to track maneuvering intruders. This research builds on previous work that used predefined aircraft encounter trajectories. An established encounter model generates the intruder trajectories while multiple models are introduced to improve intruder dynamics estimation. A new method using the Kalman prediction phase inside the Interactive Multiple Model (IMM) algorithm is presented to estimate DAA Hazard States (time to closest point of approach, horizontal miss distance, and vertical separation). The efficiency of the trajectory estimation has direct implication on the estimation of the intruder trajectory in relation to the own aircraft. The methods described in this research can aid a certification authority in determining if a DAA system is sufficient for UAS integration into the National Airspace System.

I. Introduction

This research describes new methods to apply safety standards in Detect and Avoid (DAA) functions for Unmanned Aircraft Systems (UAS) using target tracking and encounter models. With the expanding range of applications for UAS operations, the United States Congress mandated the Federal Aviation Administration (FAA), through the FAA Modernization and Reform Act of 2012, to develop necessary requirements for integration of UAS into the National Airspace System.¹ One of the challenges for the FAA to meet this mandate is to ensure that safety targets are met. While a manned aircraft's pilot relies on human vision to see and avoid non-cooperative intruders (those not employing a transponder or Automatic Dependent Surveillance-Broadcast, ADS-B),² a UAS requires sensors to detect and avoid intruder aircraft.

Previous work used predefined linear encounter trajectories (either with constant velocity or linear acceleration) for the sensor uncertainty analysis, focusing on potential borderline cases as a test for the methodology.³ In this work, an established encounter model generates the intruder trajectories. This better accounts for the likelihood of different types of encounters, to include potential worst case intruder trajectories. In addition, where previous research implemented a single Kalman filter to estimate the relative intruder aircraft dynamics,⁴ the methods in this research aim to improve the intruder dynamics estimation using the Multiple Model Adaptive Estimation (MMAE) concept to track the maneuvering target.⁵ This research serves as a continuation effort to evaluate the MMAE tracking applied to realistic trajectories on the UAS DAA capabilities for realistic intruder encounters.

I.A. Well Clear / CPA

Since a UAS will not have a pilot on board, it will have to replicate the pilot vision through appropriate sensors. DAA sensors include radar, Laser/Light Detection and Ranging (LIDAR), Electro-Optical (EO),

*PhD Candidate, Department of Mechanical, Materials and Aerospace Engineering, 10 W 32nd Street, Room 243, Student Member AIAA.

[†]Senior Research Associate, Department of Mechanical, Materials and Aerospace Engineering, 10 W 32nd Street, Room 243, Member AIAA.

[‡]Professor, Department of Mechanical, Materials and Aerospace Engineering, 10 W 32nd Street, Room 243, Associate Fellow AIAA.

acoustic, and Infrared (IR).^{6,7} Until 2014, the self-separation (SS) concept of keeping aircraft well clear of each other had never been fully defined, despite “well clear” being a widely recognized term by the FAA and International Civil Aviation Organization (ICAO).⁸ Recently, the Radio Technical Commission for Aeronautics (RTCA) Special Committee-228 (SC-228), in their Phase I Detect and Avoid (DAA) Minimum Operational Performance Standards (MOPS), defined the Well Clear Threshold (WCT) as the time τ to horizontal closest point of approach (CPA), of 35 seconds, a horizontal miss distance (MD) of 4000 ft, and a vertical miss distance of 450 ft.⁹ In this paper, we consider tau, HMD, and vertical miss distance to be “hazard states,” which define the hazard of a loss of well clear between two aircraft.

When an intruder cannot remain well clear, a collision avoidance maneuver is required to avoid a near mid-air collision (NMAC). NMAC limits are defined as 500 ft laterally and 100 ft vertically from the own aircraft¹⁰ and are shown, along with the WCT and CPA, in Figure 1.

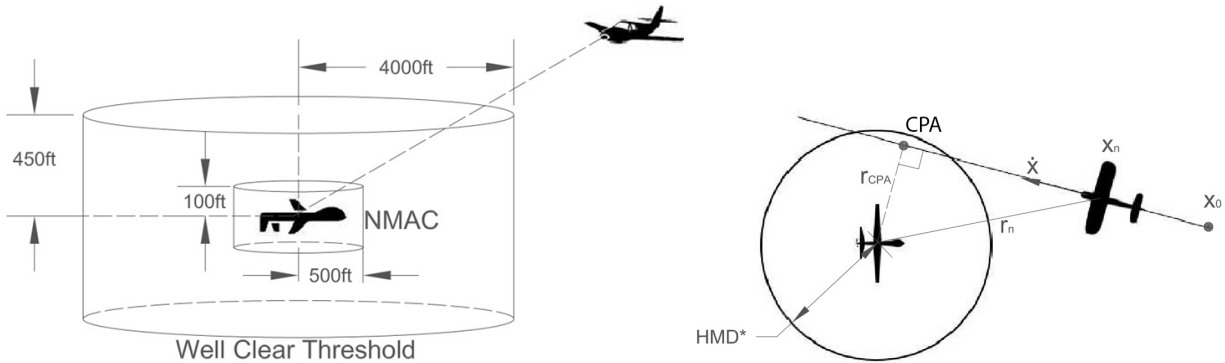


Figure 1: Well Clear Threshold, CPA, and NMAC

This work employs WCT and Self-Separation limits, although the same methodology could be applied to NMAC and collision avoidance limits.

I.B. Target Tracking

The DAA system of the UAS (the own aircraft) must detect a non-cooperative intruder aircraft (a target) then estimate its trajectory. Uncertainty in the measurement due to sensor noise will give poor state estimates when the intruder is first detected by the sensors, but as the DAA system gets more sensor measurements, our estimated trajectory error decreases.

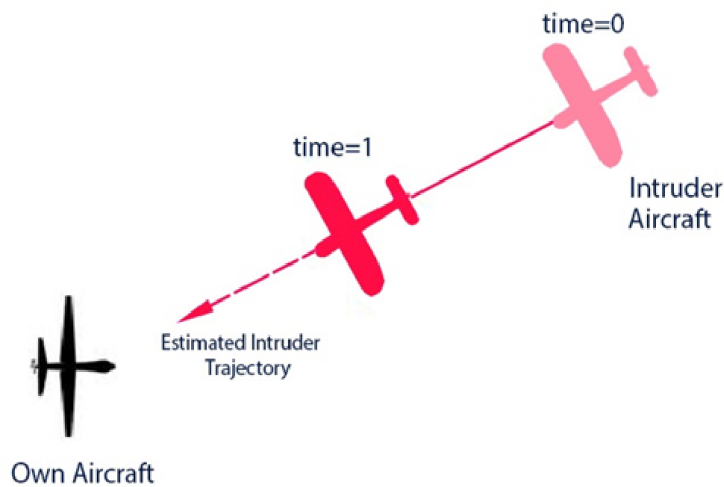


Figure 2: Intruder aircraft tracking

When tracking maneuvering intruders (with nonlinear trajectories), there is limited knowledge of changes

in intruder behavior beyond the measurement uncertainty due to sensor noise.¹¹ The standard Kalman filter (KF) with a single motion model is limited in performance for such problems due to ineffective responses to dynamics changes as the target maneuvers. In these cases, state estimation can be improved using Multiple Model Adaptive Estimation (MMAE) to account for different potential intruder behaviors.¹²

I.B.1. Multiple Model Adaptive Estimation

In MMAE, it is assumed that the current active dynamic model is one of a discrete set of r models, each of which is the basis for a separate filter.¹² These filters run in parallel to estimate the states of targets with changing dynamics. For now, we assume that for each model we have a prior probability and that the probabilities of switching from model i to model j in the next time step are known. The latter can be represented as a transition probability matrix of a Markov chain characterizing the mode transitions. Such systems are called hybrid since they have both continuous (noise) and discrete (mode or model) uncertainties.¹²

I.B.2. IMM Algorithm

The optimal approach to filtering the states of a multiple model system requires running filters for every possible model sequence. That is, for r models, r^n optimal filters must be implemented to process measurements at time step n . Except for the simplest models applied over very short intervals, an approximation is needed in practical applications of multiple model systems. One of the most popular practical algorithms for the MMAE is the Interacting Multiple Model (IMM) estimator, whose model changes according to a finite-state, discrete-time Markov chain.^{12, 13}

The IMM estimator is a suboptimal hybrid filter that has been shown to achieve an excellent compromise between performance and complexity.^{14, 15} In the IMM, each state estimate is computed under each possible current (timestep t) model using r filters, with each filter using a different combination of the previous (timestep $t - 1$) model-conditioned estimates, which are the mixed initial conditions. The regime switching is usually modeled by a finite state homogeneous Markov chain, with a-priori known transition probabilities. As depicted in Figure 3, this algorithm can be divided into five stages: (1) calculation of mixing probabilities, (2) mixed initial states and covariances, (3) filtering, (4) mode probability update and (5) state estimate combination (output only, stage five is not part of the recursion). The transition matrix of the Markov Chain - used in the mode probability update - is assumed constant over time.

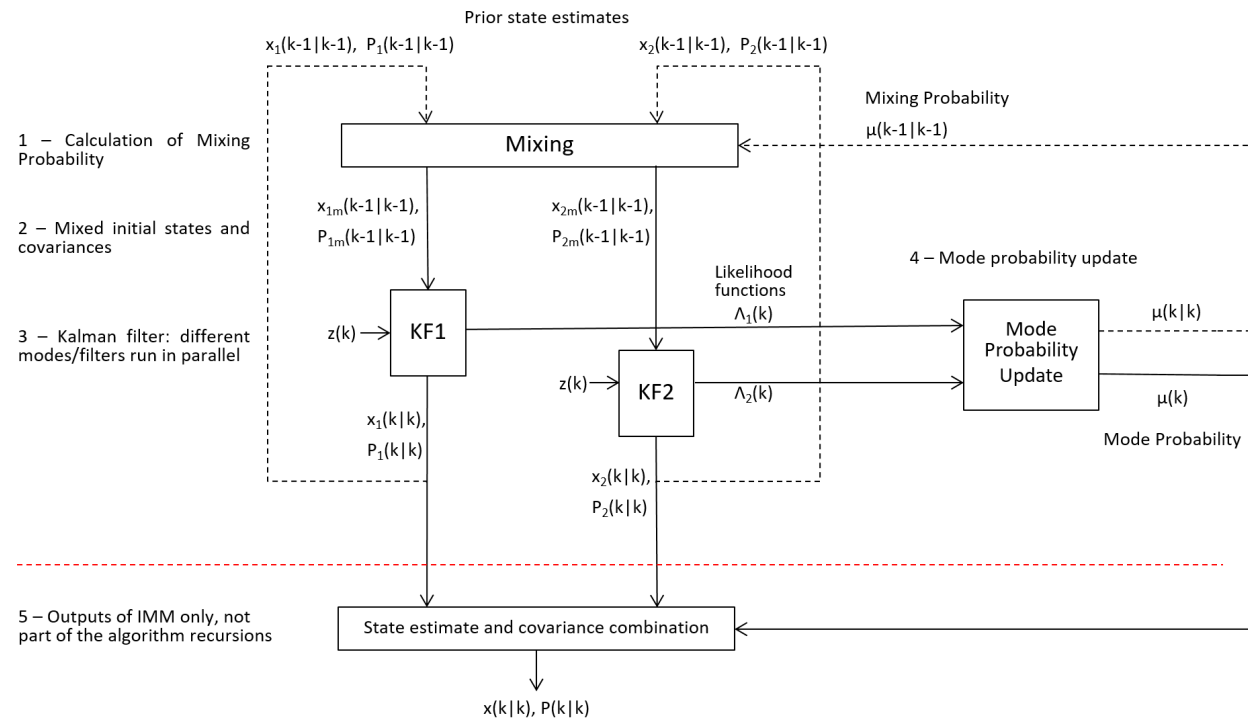


Figure 3: IMM Algorithm

I.C. Contributions

The main research goal is to develop new methods for the UAS DAA system, addressing the system safety performance with a maneuvering intruder and developing a new method for Hazard States estimation, advancing previous work that used predefined, linear trajectories^{4,5}. We generate random trajectories using an established Encounter Model to account for realistic trajectories and apply intruder tracking using a Multiple Model Adaptive Estimation (MMAE) algorithm. The algorithm chosen was the Interactive Multiple Model (IMM). In previous work, we demonstrated the IMM performance comparing it with the Perfect IMM, which is used as a benchmark for the mode adaptation analysis.⁵ In this work, we use the Hazard States estimation to analyze the IMM and general methodology performance. A new method for the Hazard States estimation was developed to account for the maneuvering intruders, since the calculations used in the MOPS only consider a linear, non-accelerated trajectory.⁹

I.D. Paper Outline

After this introductory section, Section II introduces the methodology of the multiple model estimation method as well as the new method for the Hazard States estimation. Section III includes a further analysis on the Interactive Multiple Model algorithm (IMM) selected among the MMAE algorithms for this work. Section IV presents results for the new estimation method for a particular trajectory and for a Monte Carlo run. Finally, Section V provides conclusions and opportunities for future research.

II. Methodology

In this section we are going to present the methodology behind the IMM algorithm, starting from the Kalman Filters that correspond to each individual IMM mode. Following the formal definition of the Hazard States we present the methodology behind the new method for the Hazard States estimation.

II.A. Relative Intruder State

Each state space model can be expressed with equations of the following form:

$$\mathbf{x}_k = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1} + \mathbf{q}_{k-1} \quad (1)$$

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{r}_k \quad (2)$$

where:

- $\mathbf{x}_k \in R^n$ are the system states (intruder Cartesian position, velocity, and acceleration in inertial coordinates) at time step k .
- \mathbf{F}_{k-1} is the transition matrix of the dynamic model.
- \mathbf{G}_{k-1} is the transition matrix of the dynamics of the own aircraft.
- \mathbf{u}_{k-1} are the own aircraft states in inertial coordinates.
- $\mathbf{q}_{k-1} \sim N(0; \mathbf{Q}_{k-1})$ is the process noise at time step $k - 1$.
- $\mathbf{z}_k \in R^m$ is the cartesian relative intruder measurement at time step k .
- \mathbf{H}_k is the measurement model matrix.
- $\mathbf{r}_k \sim N(0; \mathbf{R}_k)$ is the measurement noise on time step k .
- The initial distribution for the state is $\mathbf{x}_0 \sim N(0; \hat{\mathbf{P}}_0)$, where $\hat{\mathbf{P}}_0$ is typically large based on the lack of initial knowledge about the target.

The Kalman filter has two steps: the prediction step, where the next state of the system is predicted given the previous measurements, and the update step, where the current state of the system is estimated given the measurement at that time step. The equation for these steps are as follows:¹²

Prediction:

$$\bar{\mathbf{x}}_k = \mathbf{F}_{k-1} \hat{\mathbf{x}}_{k-1} + \mathbf{G}_{k-1} \mathbf{u}_{k-1} \quad (3)$$

$$\bar{\mathbf{P}}_k = \mathbf{F}_{k-1} \hat{\mathbf{P}}_{k-1} \mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1} \quad (4)$$

Update:

$$\mathbf{S}_k = \mathbf{H}_k \bar{\mathbf{P}}_k \mathbf{H}_k^T + \mathbf{R}_k \quad (5)$$

$$\mathbf{K}_k = \bar{\mathbf{P}}_k \mathbf{H}_k^T \mathbf{S}_k^{-1} \quad (6)$$

$$\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \bar{\mathbf{x}}) \quad (7)$$

$$\hat{\mathbf{P}}_k = \bar{\mathbf{P}}_k - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^T \quad (8)$$

Different from fixed radar applications (air traffic control as an example) in which the tracker is static, the UAS DAA system will contain tracking sensors mounted on our unmanned aircraft. In this way, standard maneuvers such as a coordinated turn (CT) are viewed differently in the aircraft relative frame. We have to consider the transitions between these reference frames in the formulation.

The inertial to relative frame transformation is performed as shown in the Appendix. Our final Equation (9) is analogous to the Kalman filter Equation (3).

$$\mathbf{x}_{k+1}^{rel} = \mathbf{F}_k^I \mathbf{x}_k^{rel} + \mathbf{G}_k \mathbf{u}_k \quad (9)$$

where:

- \mathbf{x}^{rel} is the state vector of the intruder aircraft in the own aircraft-centered relative frame.
- \mathbf{F}^I is the state transition matrix of the intruder aircraft.

II.B. Dynamic Models of Nonlinear Intruder Estimation

To build an MMAE framework (as in Figure 3) you will use multiple KF to estimate nonlinear intruder trajectories. - An encounter model was used to determine the different ways an intruder can maneuver - These encounter model results were used to define six individual models (or modes) of intruder maneuvering.

II.B.1. Encounter Model

We will have three uses for the Encounter Model, being the intruder trajectory generation, helping defining the modes (dynamic models) for the IMM algorithm and also generating the values of the Transition Matrix of the Markov Chain.

The MIT Lincoln Lab Encounter Model that we use is based on radar data collected across the U.S.¹⁶ The model uses dynamic variables to construct random aircraft trajectories that are statistically similar to those observed in the radar data. It gives as output the initial conditions [airspace class A , flight level L , velocity v , linear acceleration \dot{v} , vertical velocity \dot{z} , turn rate $\dot{\psi}$] and control variables (linear acceleration \dot{v} , turn rate $\dot{\psi}$, and the vertical velocity \dot{z}) for each timestep. After initializing our simulated intruder trajectory with the initial conditions, we are interested in updating the new trajectory states based on the control variables generated.

II.B.2. Models of Intruder Trajectory Dynamics

Based on the potential ways the encounter model can build an intruder trajectory, we will at least require models for straight level flight, turns, linear acceleration, and climb/descent. This results in six models/modes, three different modes for 2D movement and the same three modes in 3D with the addition of the vertical velocity \dot{z} to account for climb/descent:

- Mode 1 - Constant Velocity

- Mode 2 - Constant Velocity 3D (non-zero \dot{z})
- Mode 3 - Coordinated Turn
- Mode 4 - Coordinated Turn 3D (non-zero \dot{z})
- Mode 5 - Constant Linear Acceleration
- Mode 6 - Constant Linear Acceleration 3D (non-zero \dot{z})

The possibility of using additional modes was explored in previous work and has not shown improvement over the six modes.⁵

For Modes 1 and 2, we have six states:

$$\mathbf{x} = \begin{bmatrix} x & \dot{x} & y & \dot{y} & z & \dot{z} \end{bmatrix}^T \quad (10)$$

Modes 3 through 6 have nine states:

$$\mathbf{x} = \begin{bmatrix} x & \dot{x} & \ddot{x} & y & \dot{y} & \ddot{y} & z & \dot{z} & \ddot{z} \end{bmatrix}^T \quad (11)$$

The implementations of the Constant Velocity (CV) and Constant Acceleration (CA) filters are relatively straightforward. For the kinematically constrained Coordinated Turn (CT) model (modes 3 and 4), the state transition matrix is defined to model a constant-speed turn along the trajectory which is defined by the state estimates of velocity and acceleration. Then, the transition matrices are based on the dynamics for the intruder \mathbf{F}^I and own aircraft \mathbf{F}^O are:

$$\mathbf{F}^I = \begin{bmatrix} 1 & \frac{\sin(\omega T)}{\omega} & \frac{(1-\cos(\omega T))}{(\omega^2)} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \cos(\omega T) & \frac{\sin(\omega T)}{\omega} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\omega \sin(\omega T) & \cos(\omega T) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \frac{\sin(\omega T)}{\omega} & \frac{(1-\cos(\omega T))}{(\omega^2)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cos(\omega T) & \frac{\sin(\omega T)}{\omega} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\omega \sin(\omega T) & \cos(\omega T) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & T & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (12)$$

$$\mathbf{F}^O = \begin{bmatrix} 0 & T & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & T & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & T & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (13)$$

$$\mathbf{G} = \mathbf{F}^I - \mathbf{F}^O = \begin{bmatrix} 1 & \frac{\sin(\omega T)}{\omega} - T & \frac{(1-\cos(\omega T))}{(\omega^2)} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \cos(\omega T) - 1 & \frac{\sin(\omega T)}{\omega} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\omega \sin(\omega T) & \cos(\omega T) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \frac{\sin(\omega T)}{\omega} - T & \frac{(1-\cos(\omega T))}{(\omega^2)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cos(\omega T) - 1 & \frac{\sin(\omega T)}{\omega} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\omega \sin(\omega T) & \cos(\omega T) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (14)$$

where T is the simulation timestep.

$$\mathbf{u} = \begin{bmatrix} 0 & \dot{x}_{own} & 0 & 0 & \dot{y}_{own} & 0 & 0 & \dot{z}_{own} & 0 \end{bmatrix}^T \quad (15)$$

where \dot{x}_{own} , \dot{y}_{own} and \dot{z}_{own} are the own aircraft velocity components. Here we have assumed that our own aircraft has a constant velocity vector during the encounter. Strictly, this assumption is not necessary at this point in the general development. However, we write it here as an example because we will use it the performance analysis later in the paper.

Then the turning rate ω is defined by:

$$\omega = \frac{\|\mathbf{a}\|}{\|\mathbf{v}\|} = \frac{\sqrt{\ddot{x}^2 + \ddot{y}^2 + \ddot{z}^2}}{\sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}} \quad (16)$$

where \mathbf{a} and \mathbf{v} are the accelerations and velocity vectors, in the inertial frame.

Considering the case where we have only relative position measurements, for example, the measurement matrix is:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (17)$$

The matrices for the other modes can be found in the Appendix.

II.B.3. Transition Matrix Probabilities

The IMM estimator assumes that the Transition Matrix of the Markov Chain governing the mode jumps is known. However, it is very difficult to determine the appropriate matrix quantities and identify a Markov transition law that optimally fits the unknown target motion.¹³

Fortunately, we used the encounter model itself, which is based on these radar observations¹⁶ to get the best available mode transition matrix. To do this, we ran the Encounter Model 10⁶ times and recorded each mode transition, in order to get all the likelihood of mode changes during the encounters. We used these runs to build the Transition Matrix, outlined in equation (44) in the Appendix.

II.C. Hazard States

The DAA MOPS introduces the term DAA Well Clear (DWC) as “a temporal and/or spatial boundary around the aircraft intended to be an electronic means of avoiding conflicting traffic”.⁹ DWC is synonymous with the WCT. The MOPS mathematically defines DWC in the following equation for a loss of DWC (LoWC):

$$LoWC = [0 \leq \tau_{mod} \leq \tau_{mod}^*].and.[HMD \leq HMD^*].and.[-h \leq d_h \leq h^*] \quad (18)$$

with $\tau_{mod}^* = 35sec$, $HMD^* = 4000ft$ and $h^* = 450ft$.

The MOPS LoWC threshold is synonymous with the WCT. Based on the DAA MOPS, these variables, or Hazard States, are:

- Horizontal Closest Point of Approach (CPA), or Horizontal Miss Distance (HMD)
- Modified Tau (τ_{mod})
- Vertical Separation (h)

There are two ways to determine the Horizontal CPA, r_{CPA} . One is the time based definition in the DAA MOPS and the other is geometrically determined based on a linear intruder trajectory. The geometric r_{CPA} can be derived from the definition¹⁷ and its expression is:

$$r_{CPA} = \frac{\dot{y}x - \dot{x}y}{\sqrt{\dot{x}^2 + \dot{y}^2}} \quad (19)$$

The importance of the Hazard States is in defining the WCT (Well Clear Threshold) Violation or Loss of DAA Well Clear, which happens when all three are below their thresholds. The previous analysis⁵ was done using the MOPS formulas, for Horizontal CPA (HMD) and τ_{mod} :⁹

$$HMD = r_{CPA} = \sqrt{(d_x + v_{rx}t_{CPA})^2 + (d_y + v_{ry}t_{CPA})^2} \quad (20)$$

$$\tau_{mod} = \max(0, -\frac{d_x v_{rx} + d_y v_{ry}}{v_{rx}^2 + v_{ry}^2}) \quad (21)$$

where:

- $v_{rx} = \dot{x}_2 - \dot{x}_1$ is the relative horizontal velocity in the x direction
- $v_{ry} = \dot{y}_2 - \dot{y}_1$ is the relative horizontal velocity in the y direction
- $d_x = x_2 - x_1$ is the current horizontal separation on the x direction
- $d_y = y_2 - y_1$ is the current horizontal separation on the y direction

The vertical separation for alerting performance requirements are based on actual altitudes of the own and intruder aircrafts. We assume constant vertical velocity to account for the lookahead time alert criteria. These MOPS equations for the Hazard States are based upon a linear trajectory with constant velocity being extended in the future. As explained in Section II.E, a linear trajectory can be a problematic approximation for a nonlinear trajectory.

II.D. IMM Estimation Error Sources

The IMM algorithm has estimation errors that are not directly related with its calculated covariances. The total error in the Hazard States and trajectory prediction will be a sum of different error sources. Some will depend on the quality of the implementation of the algorithm, others on the inherently noisy measurements from the DAA system. In summary, we can divide the main contributors of the error into four parts:

- Modeling error: This is partly due to the modeling errors, where a maneuver, by its nature, does not follow the assumptions of the standard dynamic models used in the IMM.¹⁵
- Sensor noise: Random noise on the sensor measurements.
- Mode transition: A relatively large estimation error is needed for the algorithm to recognize sudden changes in system modes, i.e. the adaptation might not be rapid enough. We will introduce later the Perfect IMM, in which the true mode of the target is known, for performance comparison.
- Prediction error: Since our analysis covers maneuver changes along the simulation and we cannot predict future intruder intent, we have an additional source of error which consists of the prediction error of the estimated Hazard States (in Section II.C) from the actual Hazard States, which is only known after an encounter has ended.

In a real IMM, even with perfect measurements, we would still have uncertainty in trajectory estimation due to the approximate motion models used.

II.E. New Hazard States Estimation Method

To minimize prediction errors on the Hazard States, we introduce a new method instead of using the linear trajectory approximation. The approach is to use the IMM prediction in the Kalman prediction step as the estimation for the Hazard States. We now have four different ways to define the Hazard States:

- MOPS: using the formula as defined by the MOPS in Equation (20) and true trajectory data
- IMM-MOPS: using the formula as defined by the MOPS in Equation (20) and the best IMM trajectory estimation
- IMM-Predicted: the estimated Hazard States calculated at each timestep, using the new IMM prediction method for extending the future trajectory
- True: only known after the encounter is completed (influenced by future mode switches)

The method can be used to estimate any of the defined Hazard States, but as a benchmark we start using the CPA/HMD since it is more affected by the maneuvering trajectories. It is interesting to note that the difference between MOPS CPA and IMM-MOPS CPA are only due to estimation errors in the IMM estimation, since they use the same method. While an estimated CPA was valid for a particular epoch, maneuver changes result in changes to the True CPA, which can only be known until after an encounter simulation is complete. The True CPA is an analysis tool, to be used for a benchmark comparison to which method gives a better prediction.

An example of the MOPS, IMM-MOPS, and IMM-Predicted CPAs are depicted in Figure 4. The difference between CPA_{MOPS} and $CPA_{IMM-MOPS}$ is due to the error in the IMM state estimation when compared to the true trajectory and states. The new CPA estimation method is shown as $CPA_{IMM-PRED}$. Under maneuvers, the Hazard States estimation will be changing at each timestep.

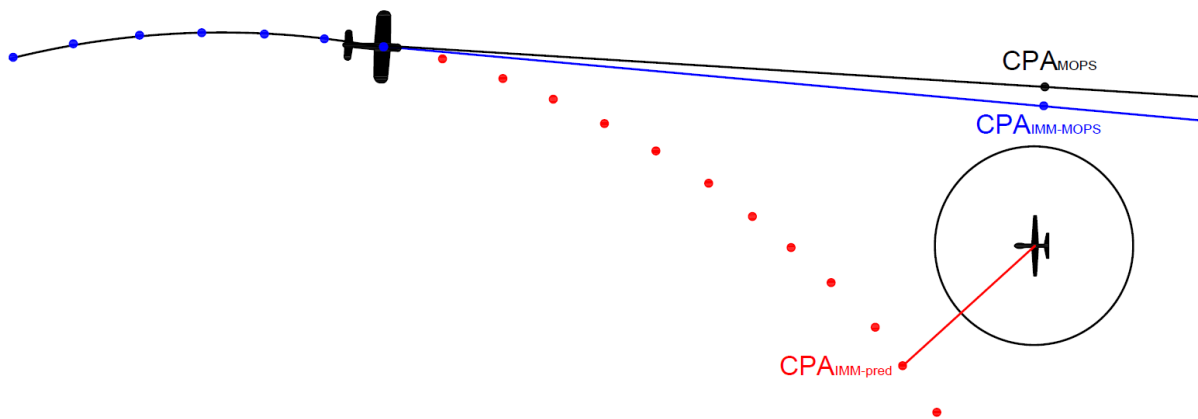


Figure 4: CPA estimation

We don't know future aircraft intent (and possible maneuver changes), so the True CPA can only be known after the encounter is completed. Using the knowledge of the current maneuver can improve our estimation based on the IMM prediction. After a maneuver occurrence in the future (as shown in Figure 5), the new estimation gets better as the IMM recognizes the maneuver and adjusts for a new prediction subsequently.

We should note that, even though the focus on the representation is on the CPA, the Time to CPA can also be extracted using the same method, as well as the vertical separation threshold. We chose the CPA since it is easier to visualize graphically and also it represents the greater change in the calculation between the new method and the linear formula.

III. Analysis

III.1. Simulation Steps

First, we generate and extract the initial conditions and control variables from Encounter Model. These are only scalars (velocity, linear acceleration, turn rate, and the vertical velocity) so we need to provide a initial

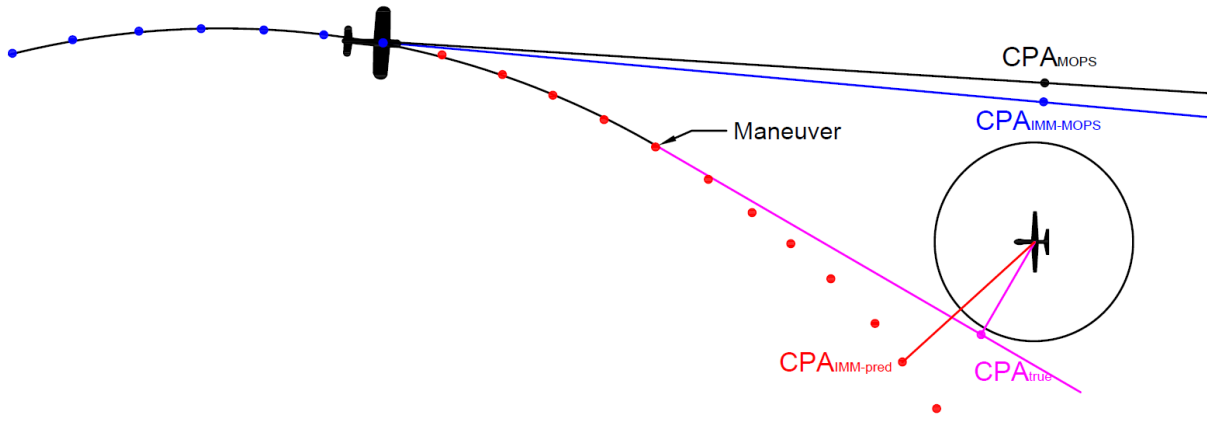


Figure 5: Future maneuver change

position and direction in relation to our own aircraft. We initialize the position of the intruder aircraft on the surface of an “encounter cylinder” centered on the own aircraft, at a random heading angle. The probability of being initialized on the top, bottom, or side surfaces of the encounter cylinder is proportional to the surface area of each. We apply sampling rejection for the initial position conjugated with the initial control variables from the Encounter Model. If the intruder is not in an inward trajectory from the surface, we repeat the initialization.

Since the trajectory dynamics are not provided in the encounter model, we build the intruder trajectory using point-mass kinematics to update the aircraft states. The trajectory is then inserted into the IMM estimation algorithm, using the Encounter Model outputs as inputs for the true intruder trajectory. Finally, we get the simulation outputs for analysis as shown in Figure 6.

In this way the IMM estimator is being fed with measurements that are effectively perfect. The usefulness of this idealistic exercise is discussed in Section II.D, IMM Estimation Error Sources.

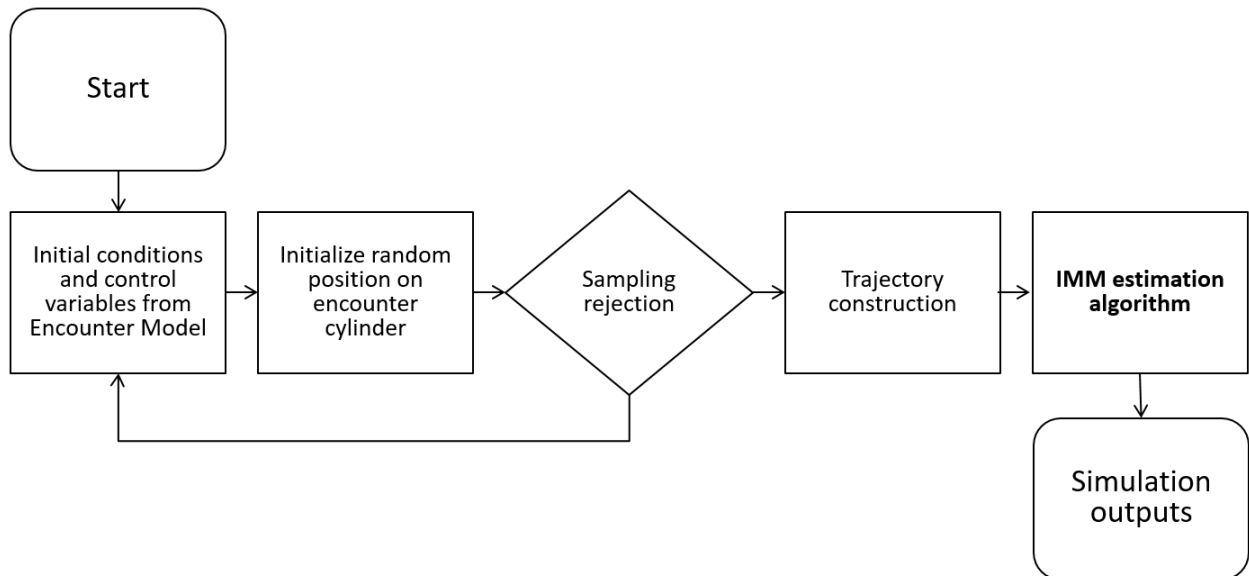


Figure 6: Simulation Structure

III.A. Perfect IMM (PIMM)

In these simulations, the true motion of the target is known. The output from the intruder is defined by the encounter model, and the PIMM switches to the correct motion model at each mode change. Since the PIMM always selects one motion model as soon as the mode change occurs, it defines the lower bound on the IMM estimation error for any given sensor error model, since the mode transition error will be zeroed. In our DAA sensor analysis, the importance of this lower bound is that we know that we cannot improve the state estimation on the IMM beyond the PIMM results, given the same modes and conditions regardless of sensor quality.

III.A.1. IMM vs Perfect IMM

In order to address the four errors sources in Section II.D, sensor noise was zero in this analysis, isolating the other three errors. The mode transition error is eliminated by using the PIMM. We can look at how the mode transition errors affect our results when comparing the IMM to the PIMM under the same conditions. Finally, the errors on Table 1 are the combination of modeling error and prediction error.

In previous work, we showed the IMM performance vs the Perfect IMM vs the Kalman Filters that correspond to each one of the different modes.⁵ We ran 10^5 simulations and calculated RMS of CPA error for each one of the filters. Since each of the KF or IMM modes took some time to stabilize, we calculated the RMS values of the CPA error starting at different timesteps T_s to account for the adaptation time. The CPA estimation error mean RMS for each mode, IMM, and PIMM at 10, 15, and 20 seconds are displayed in Table 1:

	$T_s = 10s$	$T_s = 15s$	$T_s = 20s$
Mode 1	1318.9	1294.9	1311.0
Mode 2	1318.9	1294.9	1311.0
Mode 3	2411.7	2439.6	2483.9
Mode 4	2409.1	2438.3	2481.6
Mode 5	4046.3	3765.1	3547.6
Mode 6	4046.3	3765.1	3547.6
IMM	112.5	107.6	103.3
PIMM	111.5	107.0	102.7

Table 1: Mean RMS of CPA error [ft] estimation of 10^5 trajectories

The performance of the IMM and the Perfect IMM is negligible when considering the RMS error. The main difference was caused by the adaptation lag between a mode change and the algorithm response, although that was only a few seconds for most of the trajectories.

III.A.2. Data Structure

For each timestep of the main IMM, we have to run the prediction algorithm until the end of the simulation. In comparison with the IMM, in which we were dealing with only the state estimation of the current timestep (diagonal of the matrix shown in Figure 7), we now have to deal with a much large dataset, since the state estimates for each different filter and the IMM are preserved for all future timesteps. Each one of the elements (state estimation, X) are calculated for each one of the modes and then the best estimation is combining them under the IMM algorithm. The elements on the main diagonal represent a state estimate for the current epoch while the other elements are estimated in the future using the IMM prediction.

IV. Results

First we have an example of a particular trajectory, and then a Monte Carlo run to analyze the general performance of the algorithm. In both cases, the simulations were done following the flowchart in Figure 6. For all the different encounters simulated, the own aircraft was assumed to have a linear trajectory with constant velocity of 200 kt, while the generated trajectory for the intruder aircraft is randomly generated by

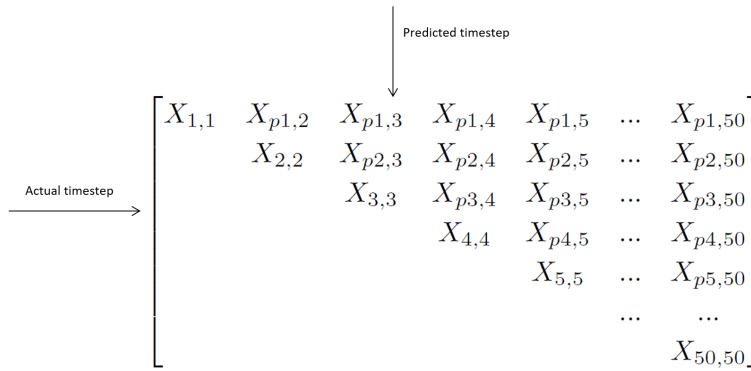


Figure 7: Actual timesteps and predicted timesteps

the encounter model. Each of the trajectories were simulated for 60 seconds of total simulation time with a sample rate of 1 Hz.

IV.A. Maneuvering Trajectory Example

To represent a complex scenario, we chose a trajectory in which we have all control variables non-zero as shown in Figure 8 for this simulation. The trajectory throughout the Encounter Cylinder is shown in Figure 9 with both cylinders shown; the encounter cylinder (blue) and the WCT (red). The encounter cylinder was assumed to have a ± 3000 ft height and 5.4 NM radius^{9,18}

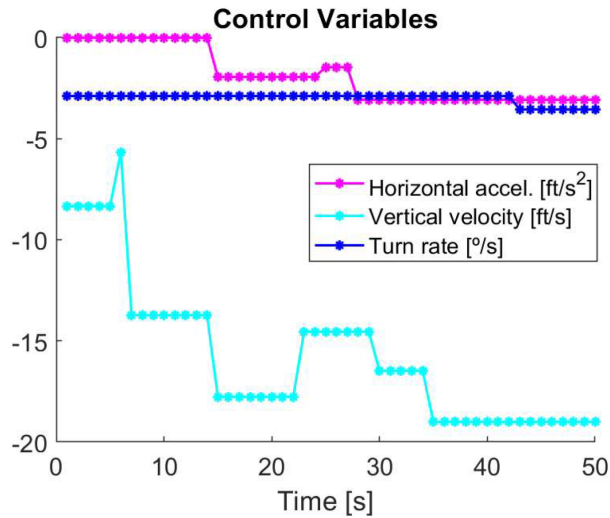


Figure 8: Control variables for the simulation

The predicted trajectories are shown in Figure 10. The true trajectory is shown in red, while the other are predicted trajectories starting each at 10s intervals, starting after 10s of simulation. As expected the largest deviations in the prediction happens after future maneuvers.

For each timestep, the horizontal CPA is estimated and knowing the True CPA after the simulation is finished we plotted a graph comparing the CPA estimation using the new method against the MOPS calculation. The CPA estimation for this trajectory is shown in Figure 11.

We can see that in this example (with a highly non-linear trajectory) our prediction method gives a better prediction than the MOPS CPA formula, which is expected since the formula this MOPS formula extrapolates linearly an intruder trajectory.

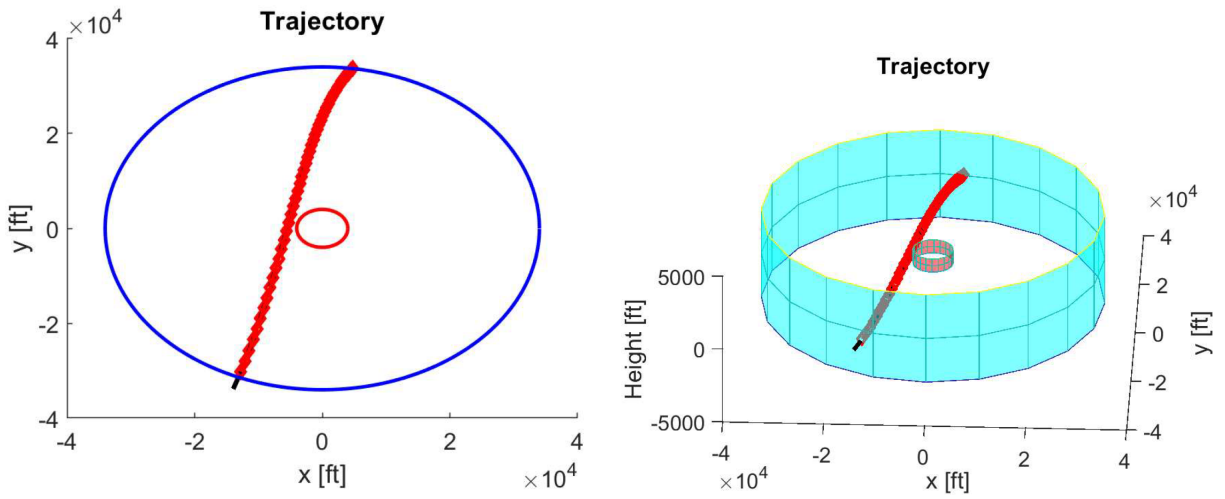


Figure 9: Encounter trajectory in two different views

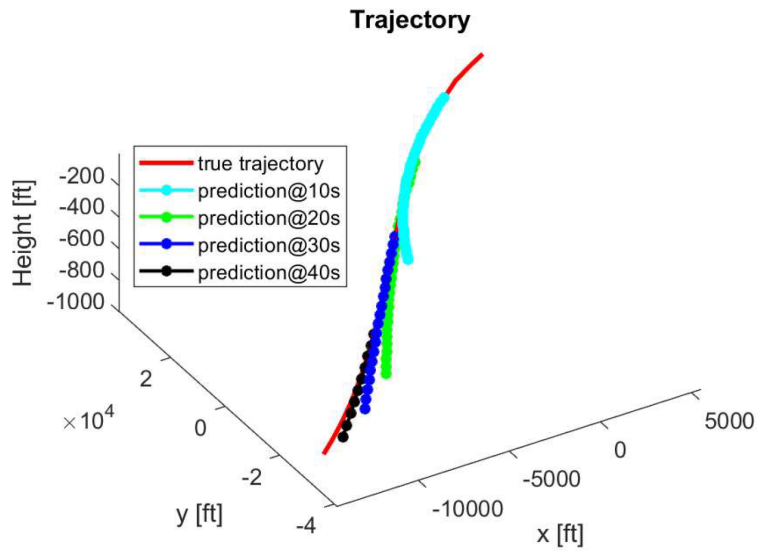


Figure 10: Predicted trajectories

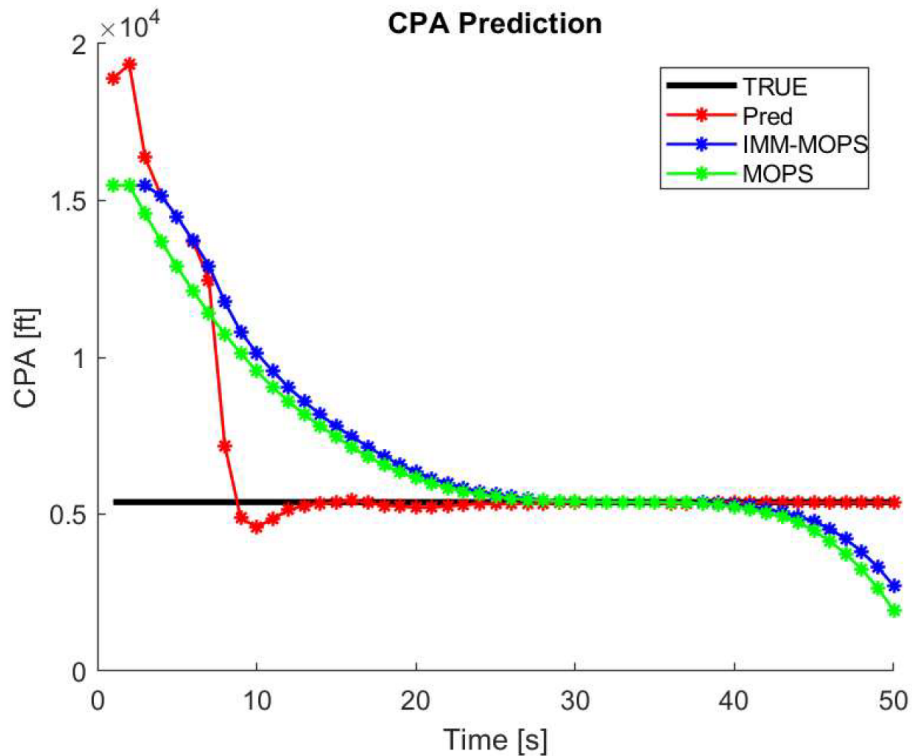


Figure 11: CPA estimation comparison

IV.B. Monte Carlo Results

In order to compare the general performance of the new estimation method, we ran 10^5 different encounters comparing it and the other methods with the True CPA value, as shown in Table 2:

	$T_s = 5s$	$T_s = 10s$	$T_s = 20s$
IMM-MOPS	758.8	572.3	560.7
True-MOPS	648.6	564.0	559.3
IMM Prediction (new)	507.1	108.0	101.5

Table 2: Mean True CPA error [m] estimation of 10^5 encounters

It is important to note the significant difference between the results shown here and in the Table 1. First, we have shown that the CPA estimation errors using the standard formula (True-MOPS and IMM-MOPS) are considerably reduced when using the IMM estimation (IMM Prediction). Our new CPA prediction method considerably reduced this estimation error.

V. Conclusion

We used an encounter model to generate realistic intruder trajectories, including nonlinear intruder maneuvers, to estimate hazardous encounters for the UAS DAA problem. The intruder dynamics estimation was done using multiple models to track the maneuvering intruder aircraft. A new method to estimate Hazard States (horizontal CPA, τ_{mod} and vertical separation) based on the IMM prediction was introduced. We first examined a sample hazardous encounter, then we accomplished a Monte Carlo analysis of 10^5 hazardous encounters. The results showed our new IMM method for estimating horizontal CPA was a significant improvement over methods that linearly extrapolate intruder trajectories.

Future work includes error analysis for realistic DAA sensors and process noise, field of regard considerations, and a safety evaluation based on probabilities of loss of well clear (and NMAC).

Appendix

Inertial to relative frame transformation

The inertial to relative frame transformation is performed as follows:

$$\mathbf{x}_{k+1}^{rel} = \mathbf{x}_{k+1}^I - \mathbf{x}_{k+1}^O \quad (22)$$

$$\mathbf{x}_{k+1}^I - \mathbf{x}_{k+1}^O = \mathbf{F}_k^I \mathbf{x}_k^I - \mathbf{F}_k^O \mathbf{x}_k^O \quad (23)$$

$$\mathbf{x}_{k+1}^I - \mathbf{x}_{k+1}^O = \mathbf{F}_k^I \mathbf{x}_k^I - \mathbf{F}_k^O \mathbf{x}_k^O - \mathbf{F}_k^I \mathbf{x}_k^O + \mathbf{F}_k^I \mathbf{x}_k^O \quad (24)$$

$$\mathbf{x}_{k+1}^{rel} = \mathbf{F}_k^I \mathbf{x}_k^{rel} + [\mathbf{F}_k^I - \mathbf{F}_k^O] \mathbf{x}_k^O \quad (25)$$

$$\mathbf{x}_{k+1}^{rel} = \mathbf{F}_k^I \mathbf{x}_k^{rel} + \mathbf{G}_k \mathbf{u}_k \quad (26)$$

where:

- \mathbf{x}^{rel} is the state vector of the intruder aircraft in the own aircraft-centered relative frame.
- \mathbf{x}^I is the state vector of the intruder aircraft in the inertial frame.
- \mathbf{x}^O is the state vector of the own aircraft in the inertial frame.
- \mathbf{F}^I is the state transition matrix of the intruder aircraft.
- \mathbf{F}^O is the state transition matrix of the own aircraft.

We define $[\mathbf{F}^I - \mathbf{F}^O] = \mathbf{G}_k$ and $\mathbf{x}_k^O = \mathbf{u}_k$ as our "input" velocity of our own aircraft to transform inertially-defined intruder maneuvers to the relative frame. Our final Equation (26) is analogous to the Kalman filter Equation (3).

Transition Matrices of the Modes

For Modes 1-2:

$$\mathbf{x} = \begin{bmatrix} x & \dot{x} & y & \dot{y} & z & \dot{z} \end{bmatrix}^T \quad (27)$$

$$\mathbf{F}^O = \begin{bmatrix} 0 & T & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & T & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (28)$$

$$\mathbf{G} = \mathbf{F}^I - \mathbf{F}^O \quad (29)$$

$$\mathbf{u} = \begin{bmatrix} 0 & \dot{x}_{own} & 0 & \dot{y}_{own} & 0 & \dot{z}_{own} \end{bmatrix}^T \quad (30)$$

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (31)$$

For Modes 3-6:

$$\mathbf{x} = \begin{bmatrix} x & \dot{x} & \ddot{x} & y & \dot{y} & \ddot{y} & z & \dot{z} & \ddot{z} \end{bmatrix}^T \quad (32)$$

$$\mathbf{F}^O = \begin{bmatrix} 0 & T & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & T & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & T & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (33)$$

$$\mathbf{G} = \mathbf{F}^I - \mathbf{F}^O \quad (34)$$

$$\mathbf{u} = \begin{bmatrix} 0 & \dot{x}_{own} & 0 & 0 & \dot{y}_{own} & 0 & 0 & \dot{z}_{own} & 0 \end{bmatrix}^T \quad (35)$$

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (36)$$

- Mode 1 - Constant velocity

$$\mathbf{F}^I = \begin{bmatrix} 1 & T & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & T & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (37)$$

- Mode 2 - Constant velocity 3D

$$\mathbf{F}^I = \begin{bmatrix} 1 & T & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & T & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (38)$$

- Mode 3 - Coordinated Turn Model (kinematics constraints)

$$\mathbf{F}^I = \begin{bmatrix} 1 & \frac{\sin(\omega T)}{\omega} & \frac{1-\cos(\omega T)}{\omega^2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \cos(\omega T) & \frac{\sin(\omega T)}{\omega} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\omega \sin(\omega T) & \cos(\omega T) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \frac{\sin(\omega T)}{\omega} & \frac{1-\cos(\omega T)}{\omega^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cos(\omega T) & \frac{\sin(\omega T)}{\omega} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\omega \sin(\omega T) & \cos(\omega T) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (39)$$

- Mode 4 - Coordinated Turn Model (kinematics constraints) - 3D

$$\mathbf{F}^I = \begin{bmatrix} 1 & \frac{\sin(\omega T)}{\omega} & \frac{1-\cos(\omega T)}{\omega^2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \cos(\omega T) & \frac{\sin(\omega T)}{\omega} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\omega \sin(\omega T) & \cos(\omega T) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \frac{\sin(\omega T)}{\omega} & \frac{1-\cos(\omega T)}{\omega^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cos(\omega T) & \frac{\sin(\omega T)}{\omega} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\omega \sin(\omega T) & \cos(\omega T) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & T & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (40)$$

- Mode 5 - Constant Linear Acceleration

$$\mathbf{F}^I = \begin{bmatrix} 1 & T & T^2/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & T^2/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (41)$$

- Mode 6 - Constant Linear Acceleration - 3D

$$\mathbf{F}^I = \begin{bmatrix} 1 & T & T^2/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & T^2/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & T & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (42)$$

Transition Matrix and Probability Vector (6 modes)

Initial mode probabilities:

$$\boldsymbol{\mu}_{ip} = \begin{bmatrix} 0.55117 & 0.15393 & 0.10708 & 0.04781 & 0.09292 & 0.04709 \end{bmatrix} \quad (43)$$

Transition Matrix of the Markov Chain:

$$\mathbf{M}_{ij} = \begin{bmatrix} 0.98038 & 0.00692 & 0.00562 & 0.00093 & 0.00508 & 0.00108 \\ 0.02536 & 0.96022 & 0.00246 & 0.00509 & 0.00200 & 0.00487 \\ 0.03263 & 0.00280 & 0.95063 & 0.00646 & 0.00624 & 0.00123 \\ 0.01280 & 0.01891 & 0.01456 & 0.94566 & 0.00172 & 0.00636 \\ 0.02551 & 0.00250 & 0.01098 & 0.00077 & 0.95314 & 0.00710 \\ 0.00926 & 0.01860 & 0.00204 & 0.01147 & 0.01049 & 0.94814 \end{bmatrix} \quad (44)$$

Acknowledgments

This work is supported by the Brazilian agency Capes through a scholarship from Ciencia sem Fronteiras (Science without Borders) program / Processo No. 99999.012923/2013-03.

References

- ¹Federal Aviation Administration, "Integration of Civil Unmanned Aircraft Systems (UAS) in the National Airspace System (NAS) Roadmap," Nov 2013.
- ²Edwards, M., "A Safety Driven Approach to the Development of an Airborne Sense and Avoid System," *Infotech@Aerospace 2012*, American Institute of Aeronautics and Astronautics (AIAA), Garden Grove, CA, Jun 2012, pp. 1–12.
- ³Jamoom, M. B., Canolla, A., Pervan, B., and Joerger, M., "Unmanned Aircraft System Sense and Avoid Integrity: Intruder Linear Accelerations and Analysis," *Journal of Aerospace Information Systems*, Vol. 14, No. 1, Jan 2017, pp. 53–67.
- ⁴Jamoom, M. B., Joerger, M., and Pervan, B., "Sense and Avoid for Unmanned Aircraft Systems: Ensuring Integrity and Continuity for Three Dimensional Intruder Trajectories," *Institute of Navigation GNSS+*, Tampa, FL, Sep 2015.
- ⁵Canolla, A. C., Jamoom, M. B., and Pervan, B., "Unmanned Aircraft Systems Detect and Avoid Sensor Hybrid Estimation Error Analysis," *17th AIAA Aviation Technology, Integration, and Operations Conference*, American Institute of Aeronautics and Astronautics, Jun 2017.
- ⁶Yu, X. and Zhang, Y., "Sense and avoid technologies with applications to unmanned aircraft systems: Review and prospects," *Progress in Aerospace Sciences*, Vol. 74, Apr 2015, pp. 152–166.
- ⁷Zeitlin, A. D., "Sense & Avoid Capability Development Challenges," *IEEE Aerospace and Electronic Systems Magazine*, Vol. 25, No. 10, Oct 2010, pp. 27–32.
- ⁸Cook, S. P., Brooks, D., Cole, R., Hackenberg, D., and Raska, V., "Defining Well Clear for Unmanned Aircraft Systems," *AIAA Infotech @ Aerospace*, American Institute of Aeronautics and Astronautics (AIAA), Kissimmee, FL, Jan 2015, pp. 1–20.
- ⁹RTCA SC-228; DO-365, "Minimum Operational Performance Standards (MOPS) for Detect and Avoid (DAA) Systems," May 2017.
- ¹⁰Federal Aviation Administration, "Sense and Avoid (SAA) for Unmanned Aircraft Systems (UAS), SAA Workshop Second Caucus Report," Jan 2013.
- ¹¹Mazor, E., Averbuch, A., Bar-Shalom, Y., and Dayan, J., "Interacting multiple model methods in target tracking: a survey," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 34, No. 1, 1998, pp. 103–123.
- ¹²Bar-Shalom, Y., Li, X. R., and Kirubarajan, T., *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*, Wiley-Interscience, 2008.
- ¹³Radosavljevi, Z., "Determination of the Transition Probabilities for the Interacting Multiple Model Probabilistic Data Association Estimator," *Scientific Technical Review*, Vol. LVII, No. 2, 2007, pp. 31–37.
- ¹⁴Hwang, I., Balakrishnan, H., and Tomlin, C., "Performance analysis of hybrid estimation algorithms," *42nd IEEE International Conference on Decision and Control*, IEEE, Dec 2003.
- ¹⁵Li, X. and Bar-Shalom, Y., "Performance prediction of the interacting multiple model algorithm," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 29, No. 3, Jul 1993, pp. 755–771.

¹⁶Kochenderfer, M. J., Kuchar, J. K., Espindle, L. P., and Griffith, J. D., “Uncorrelated Encounter Model of the National Airspace System,” Project Report ATC-345, Massachusetts Institute of Technology, Lincoln Laboratory, 2008.

¹⁷Jamoom, M. B., *Unmanned Aircraft System Sense and Avoid Integrity and Continuity*, Ph.D. thesis, Illinois Institute of Technology, May 2016.

¹⁸RTCA SC-228; DO-366, “Minimum Operational Performance Standards (MOPS) for Air-to-Air Radar for Traffic Surveillance,” May 2017.