AIAA 2016-3674

AIAA Aviation
13-17 June 2016, Washington, D.C.
AIAA Modeling and Simulation Technologies Conference

Simulation Comparison of Collision Avoidance Algorithms for Multi-Rotor Aircraft

Eric Mueller,[*] and Mykel J. Kochenderfer[†]
Stanford University, Stanford, CA 94305

This paper describes the implementation of four different collision avoidance algorithms designed for multi-rotor aircraft. The primary contribution of this paper is the evaluation of each using a common set of assumptions, simulation capabilities, and metrics. The first algorithm is an extension of the next-generation manned aircraft algorithm that will soon be deployed worldwide; it poses the collision avoidance problem as a partially observable Markov decision process and solves offline for the approximately optimal solution using dynamic programming. The second algorithm is based on receding-horizon model predictive control, and the third is based on potential field methods. The fourth algorithm is an adaptation of a tactical conflict detection and resolution algorithm that uses candidate trajectory predictions to determine a preferred resolution. A batch simulation system generates trajectories and computes aggregate metrics related to each algorithm's performance, allowing direct comparison of the benefits and drawbacks of each approach. A second contribution of this paper is a description of the first known encounter model of hobbyist unmanned aircraft trajectories. This model is used to generate trajectories that are representative of the airspace. All algorithms have the flexibility to provide different tradeoffs between separation from an intruder and the trajectory deviation necessary to achieve that separation. The algorithm generated using dynamic programming delivers excellent maximum deviation performance with only slightly smaller separations than the model predictive control algorithm, and it does so with less than half of the required velocity change of the other algorithms.

I. Introduction

Collision avoidance algorithms for multi-rotor aircraft differ from those designed for manned fixed-wing aircraft in several key respects. First, multi-rotor aircraft capable of hovering or moving in any direction without the typical heading-rate limits of fixed-wing aircraft mean that longitudinal maneuvers (i.e., speed changes) are an important degree of freedom in the algorithm's choice of resolution trajectories. Longitudinal maneuvers are seldom used for collision avoidance on a fixed-wing aircraft because they do not increase separation from an intruder as quickly. A second important difference between small multi-rotor aircraft and manned fixed-wing aircraft is that, because the former will not have an onboard pilot and therefore likely possess a high degree of autonomy, the collision avoidance maneuvers may be entirely automatic. Removing humans from the control loop means that maneuver updates may be passed directly to the vehicle's control system, while a human-in-the-loop algorithm would need to consider response time delays and avoid changing maneuver recommendations unless there is a significant safety benefit.[1] Several of the approaches considered in this paper take direct advantage of these two characteristics to improve the algorithm's performance over what would be possible with a manned fixed-wing aircraft.

A common method for avoiding collision with other aircraft and obstacles is to create a set of candidate trajectories that are within the aerodynamic constraints of the vehicle being controlled, and to then select the trajectory that satisfies the separation requirements and minimizes some cost function.[2,3] One such algorithm designed for small unmanned aircraft creates a set of branching trajectories as candidates for resolution maneuvers and then removes any that would violate the separation standard. Only horizontal maneuvers

[*] Graduate Student, Aeronautics and Astronautics, Stanford University, and Research Aerospace Engineer, NASA, Eric.Mueller@nasa.gov. AIAA Associate Fellow.

[†] Assistant Professor, Aeronautics and Astronautics, Stanford University, AIAA Senior Member.

are sampled because including longitudinal or vertical maneuvers would create too many combinations of resolutions for the platform to consider.[4] Another algorithm uses mixed-integer linear programming to select an optimal set of control actions over a receding horizon to achieve both trajectory-following and collision-avoidance goals.[5] This approach can work in three dimensions,[6] but it requires the online optimization of control actions that can be computationally expensive. However, practical demonstration of this robust receding horizon predictive control technique has been achieved on a small quadrotor aircraft.[7] Another approach to collision avoidance uses an analogy with electrically charged particles to calculate attractive and repulsive forces for goal states and intruders, respectively, to determine the best current action without optimizing, or even necessarily predicting, the future trajectory.[8] Each of these algorithmic approaches can provide benefits for the particular vehicle type and mission for which they are intended, and may have advantages in terms of computational load, memory requirements or safety/risk assessment over their competitors.

The primary contribution described in this paper is the evaluation of several collision avoidance algorithms using the same simulation framework, modeling and simulation assumptions, uncertainty histories, and metrics. Although many such algorithms have been evaluated and shown to work both in simulation and flight test, comparison of performance is difficult because the evaluation methodologies used in each research effort differ significantly. One of the four algorithms evaluated in this paper was developed as an extension to the ACAS X family of collision avoidance algorithms being deployed for large, fixed-wing aircraft[1,9] and is described in detail in a companion paper[10] and dissertation,[11] while the other three were developed as benchmarks for performance comparisons with the new algorithm. A secondary contribution is the introduction of an encounter model for hobbyist unmanned aircraft that may be used to create realistic intruder trajectories against which to evaluate the algorithms. Section II introduces the simulation system, metrics and uncertainties used to evaluate the algorithms, along with an overview of the four collision avoidance algorithms and the dynamic model used to simulate aircraft encounters. The encounter model is presented in Section III, and the results of the algorithm comparison are given in Section IV.

II. Experimental Setup

This section describes relevant aspects of the Julia programming language-based simulation system used to compare the performance of the collision avoidance algorithms.[12] That system creates state and dynamic uncertainty histories, applies the uncertainties to encounters with intruder aircraft, determines and executes resolutions, and computes aggregate statistics to quantify the performance of each algorithm.

A. Collision Avoidance Algorithms

This paper compares four very different collision avoidance algorithms. The first algorithm is a recent extension of an existing family of collision avoidance algorithms being deployed for large fixed-wing aircraft.[1,9,13,14] It uses a partially observable Markov decision process (POMDP) formulation that is solved with dynamic programming using an approximation called QMDP.[15] Detailed information may be found in a companion paper.[10] The second algorithm uses a receding horizon approach based on model predictive control (MPC).[5–7,16] This algorithm was implemented in simulation as if unlimited computation resources were available, which could make real-world implementation challenging (though this method has been shown to be feasible in certain applications[17]). The third approach is based on potential field (PF) methods in which the intruder generates a repulsive "force" and the reference trajectory an attractive one.[8,18–21] The fourth algorithm uses trajectory predictions to detect close encounters and determine a preferred resolution maneuver. This last algorithm was adapted from the horizontal vector turn (HVT) algorithm, which was originally developed for large fixed-wing aircraft.[2,22–24] All four algorithms are described elsewhere in detail.[11]

B. Aircraft Encounter Dynamics

The four algorithms used an identical set of system dynamics and uncertainties in order to provide a common framework for comparing their performance. The eight state variables that characterize the problem in the horizontal dimension are defined in Fig. 1: two relative range states (r_x and r_y), two velocities for the ownship (v_{ox} and v_{oy}) and two for the intruder (v_{ix} and v_{iy}), and two states that indicate the ownship's absolute displacement from the reference trajectory at that time (d_x and d_y). The reference trajectory, shown in Fig. 1 for the current time step as a yellow star, is normalized to unit velocity in the x-axis and zero velocity in the y-axis. This simplification reduces the number of state variables required to specify the reference trajectory and avoids loss of generality because the coordinate frame of the ownship may be rotated and scaled into the algorithm's coordinate frame to find the appropriate action. All distance units, not only the reference trajectory units, are normalized to preserve the flexibility of the algorithm for aircraft with difference performance envelopes and for applications with different separation requirements. The allowable actions for these algorithms are lateral (perpendicular to the reference trajectory) and longitudinal (parallel to the reference trajectory) accelerations of up to $1\,\mathrm{s}^{-2}$. Additional details about the dynamics of the system may be found elsewhere.[11]

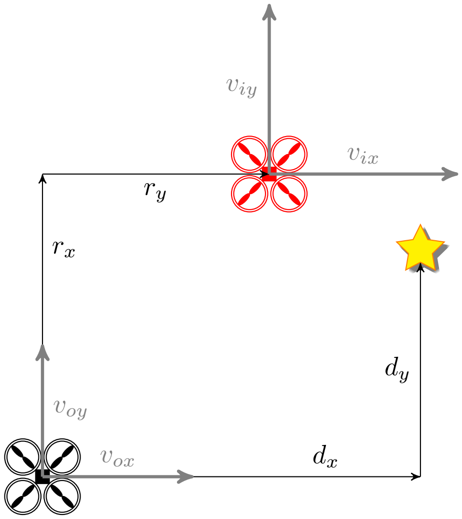


Figure 1: State variables used to formulate the collision avoidance algorithm

C. Algorithm Evaluation Metrics

The collision avoidance algorithms are evaluated primarily according to the separation they achieve from an intruder and the deviation from the reference trajectory that results from this separation. Only the most basic metrics are described here and presented in Section IV. Extensive simulation results of more than a dozen metrics are provided in a dissertation.[11]

The separation metric for a single trajectory is the horizontal range at the closest point of approach (CPA); this metric is the one most directly related to the safety of an encounter. The performance of an algorithm may be evaluated over a collection of trajectories through the use of "aggregate" metrics. The first of these is the rate at which near mid-air collisions (NMAC) occur, wherein a CPA under 1.0 constitutes an NMAC. The occurrence of these encounters is essentially a "failure" of the algorithm. The radius of the NMAC volume is the basis for the distance normalization described in the previous section. Another aggregate metric is the 5th percentile threshold separation, $r_{5\%CPA}$. The value of this metric is the CPA separation equal to the 5th percentile encounter, so 5% of encounters had a smaller CPA separation and 95% had larger minimum separations. This metric is useful as a proxy for the level of risk of the most serious encounters, however it must be considered in conjunction with the trajectory deviation to understand whether a particular value of the metric is too large or too small.

The deviation from the reference trajectory may also be measured in different ways, such as the mean deviation over an entire trajectory or the overall maximum deviation. Different unmanned aircraft applications may prefer to minimize one or the other, but this paper focuses on the maximum deviation. The aggregate trajectory deviation metric presented here is simply the median value, among a set of encounters, of the maximum trajectory deviation in each encounter.

The final metric that will be used to compare algorithms is the time-integrated action sequence, which in a discrete-time simulation is simply the sum of the absolute values of the acceleration commands multiplied by the time step. Although removing humans from the collision avoidance loop means that issuing maneuver commands or changing commands frequently is not a serious deficiency, algorithms that require less velocity change would be preferred over those that issue larger or more sustained acceleration commands.

D. Uncertainty

The lack of accurate knowledge about where aircraft are and will be in the future is the most significant challenge facing a collision avoidance algorithm. This uncertainty comes from many sources: inaccurate surveillance sensors, imprecise knowledge of the attitude of the ownship, the inability for the ownship navigation system to precisely track its reference trajectory, unknown characteristics of the environment like wind speeds and atmospheric temperature and pressure, delays in execution of commanded collision avoidance maneuvers, and lack of access to the intended future path of the intruder.

The first type of uncertainty arises primarily from surveillance system errors in the measurement of the ownship and intruder's states. This "state uncertainty" is added to the true values of each state variable during simulations.[25] Although the concept for collision avoidance in this research assumes that vehicles exchange state data derived by onboard GPS receivers, accelerometers and other sensors, the algorithms are not fundamentally dependent on this assumption. The normally distributed sensor errors used in the optimization and simulation evaluations are shown in Table 1, however it is important to note that any type of distribution may be used to model uncertainty.

Table 1: State and dynamic uncertainties

Uncertainty Parameter	Used in	Mean	Standard Deviation
Ownship acceleration, N_{ox}, N_{oy}	Opt and sim	$0\,\mathrm{s}^{-2}$	$0.30\,\mathrm{s}^{-2}$
Intruder acceleration, N_{ix}, N_{iy}	Opt and sim	$0\,\mathrm{s}^{-2}$	$0.45\,\mathrm{s}^{-2}$
Relative range, r_x, r_y	Simulation	0	0.30
Ownship absolute velocity, v_{ox}, v_{oy}	Simulation	$0\,\mathrm{s}^{-1}$	$0.075\,\mathrm{s}^{-1}$
Intruder absolute velocity, v_{ix}, v_{iy}	Simulation	$0\,\mathrm{s}^{-1}$	$0.15\,\mathrm{s}^{-1}$
Distance to trajectory, d_x, d_y	Simulation	0	0.15

The uncertainty in the dynamics of the collision avoidance encounter is implemented as disturbance accelerations on both the intruder and ownship at every time step during the encounter, and the values of this uncertainty are given in Table 1. The larger uncertainty for the intruder is meant to capture the lack of information about that aircraft's intended path. These dynamic uncertainties encompass not only the inability to precisely command a particular acceleration, but also the cumulative effects of wind uncertainty, attitude measurement and navigational system uncertainty.

Uncertainty in the intended flight path of an intruder becomes the dominant source of error when the time horizon of the prediction is sufficiently large. The point at which this occurs depends on the frequency with which the intruder accelerates (i.e., turns or changes its vertical rate) and is complex to determine in general. While the time horizons of collision avoidance for large fixed-wing aircraft are typically short enough that intent uncertainty is not a dominant source of trajectory prediction error, small multi-rotor unmanned aircraft can accelerate so quickly with respect to their typical cruising velocities that this source is important to consider. The approach to modeling intruder intent is the subject of Section III.

Realistic state and dynamic uncertainty histories are created by filtering the output of a white noise model of each state variable and acceleration noise term. Real-world uncertainties frequently arise from sources that are persistent, so filtering of the uncertainty histories is necessary to make them correlated with time. The filtering is done by applying a Gaussian kernel with a standard deviation of 1 s. The correlation of the state and noise variables is only between the individual variables and time, cross correlations between the variables are not modeled.

The uncertainty values shown in Table 1 are referred to as "1.0× uncertainty." Five different levels of uncertainty were computed, each a fraction of this full amount of uncertainty, to understand the effect that uncertainty has on the algorithm's performance. When referring to an uncertainty factor, the fraction will be reported and followed by a × to indicate that it is a multiplicative factor of the full uncertainty.

E. Intruder Trajectories

The selection of realistic and appropriate intruder trajectories is critical to the accurate evaluation of a collision avoidance algorithm. Two different classes of intruder trajectories are used in this research: intruders nominally moving with uniform velocity and maneuvering intruders that follow realistic trajectories similar to real-world unmanned aircraft. Only the uniform velocity trajectories are subject to dynamic uncertainty because the maneuvering intruders are subject to intent uncertainty. The measurement of the state of every aircraft in either class is subject to state uncertainty. All ownship trajectories are also subject to dynamic uncertainty. The initial conditions of each encounter are set for direct collision (i.e., separation at CPA is 0.0), however the effects of dynamic uncertainty may divert one or both aircraft from a direct collision.

The pool of intruder aircraft moving with uniform velocity and the full range of relative headings and speeds allows evaluation of an algorithm's performance across the spectrum of collision avoidance encounters. The relative heading is the difference between the ownship and intruder heading at CPA, so a value of 180° is a head-on encounter. Each combination of velocity and heading is replayed ten times under different sequences of state and dynamic uncertainty in order to ensure a range of reasonable behavior is obtained for each encounter type. The parameters that were systematically varied for this class of intruder and their values are shown in Table 2. A total of 1320 uniform velocity encounters were simulated for each simulation experimental condition (i.e., level of state and dynamic uncertainty).

Table 2: Encounter parameters for uniform velocity intruders

Encounter Parameter	Minimum Value	Step Size	Maximum Value
Relative heading (deg)	30	30	330
Relative velocity (s^{-1})	0.25	0.25	3.0
Number of uncertainty histories	10		

The intruder trajectories generated by the encounter model are meant to simulate realistic flights by hobbyist unmanned aircraft and represent the most difficult encounter conditions: maneuvering (or, equivalently, accelerating) intruders. In some cases it may be impossible for the ownship to avoid a close encounter when an intruder maneuvers at close range to the ownship with high relative velocity. The details of these intruder trajectories are described in Section III. A total of 7000 encounter model intruders were simulated for each experimental condition.

III. Hobbyist Unmanned Aircraft Encounter Model

This section describes the development of the first known encounter model for hobbyist unmanned aircraft. Encounter models have been used for decades to evaluate the performance and risk of airspace technologies and procedures, so without a model of the trajectories of this class of aircraft it is challenging to establish the degree to which their operations under a future airspace concept will be safe. The following sections briefly introduce the importance of encounter models, the work to improve the data quality of the trajectories that constituted the input to the model, the creation of statistical representations of the relationships between trajectory features, and the resulting simulated trajectories. Additional detail may be found in a dissertation.[11]

A. Encounter Model Background

Aircraft encounter models allow collision avoidance algorithm designers and other aviation researchers to generate artificial aircraft trajectories that are statistically representative of the flights of real aircraft. These trajectories are used to exhaustively test algorithms,[1] airspace procedures,[26] and new technologies[27] to determine how they will work in the real world. They also allow an assessment of the risk such changes could present to the airspace, a task they are well-suited for because trajectories can be created in the large numbers required to evaluate the probabilities of rare events.[28] Encounter models have been developed for many different types of aircraft conducting a variety of operations: an uncorrelated model of general aviation aircraft interactions,[29] a correlated model of aircraft interactions involving air traffic control,[30] and several models of unconventional aircraft (e.g., balloons, hang gliders, parachutists).[31] The models have even been used as part of the case justifying replacement of mature, safety-critical aviation systems.[32] No aircraft encounter models currently exist for small unmanned aircraft of the kind used by hobbyists and for which the QMDP collision avoidance algorithm was developed.

B. Aircraft Trajectory Data

A large database of unmanned aircraft flights was located at DroneShare.com, an online repository that allowed users to create profiles for their vehicles, post the telemetry data associated with their flights and view the flights of other users. The telemetry log files of more than 75,000 flights were available for download, along with parameter files that provided information about the unmanned aircraft types, autopilot settings, battery characteristics and other important vehicle properties. This database was used to develop the statistical models that form the basis of the encounter model described in the rest of this section.

C. Processing Raw Track Data

The conversion of raw telemetry log information into track data suitable for developing an encounter model required a significant amount of manual data examination and correction. This process was carried out until no significant outliers were detected and aggregate metrics related to data quality were reasonable. It was not expected that every data quality problem was to be resolved before proceeding to the feature extraction step, nor was it necessary to achieve realistic intruder trajectories.

D. Feature Determination

This model augments prior models[29–31] with three new features including the aircraft's range from the origin, R, radial-relative heading, ψ_{rr}, and radial-relative heading rate, $\dot{\psi_{rr}}$. It is expected that aircraft at larger ranges will be more likely to turn towards the operator's position at the origin because the operator's ability to see and control the unmanned aircraft degrades with distance. In addition, because an operator using this mode of control generally turns to face their aircraft, the heading of the aircraft relative to the radial line (from operator to aircraft) is expected to be more important than its heading relative to north. For example, to the operator the decision to make a turn is more dependent on whether the vehicle is moving towards ($\psi_{rr} = 180\,\mathrm{deg}$) or away ($\psi_{rr} = 0\,\mathrm{deg}$) than whether those directions are north ($\psi = 0\,\mathrm{deg}$) or south ($\psi = 180\,\mathrm{deg}$). The subset of features used to create the preliminary hobbyist unmanned aircraft encounter model are summarized in Table 3.

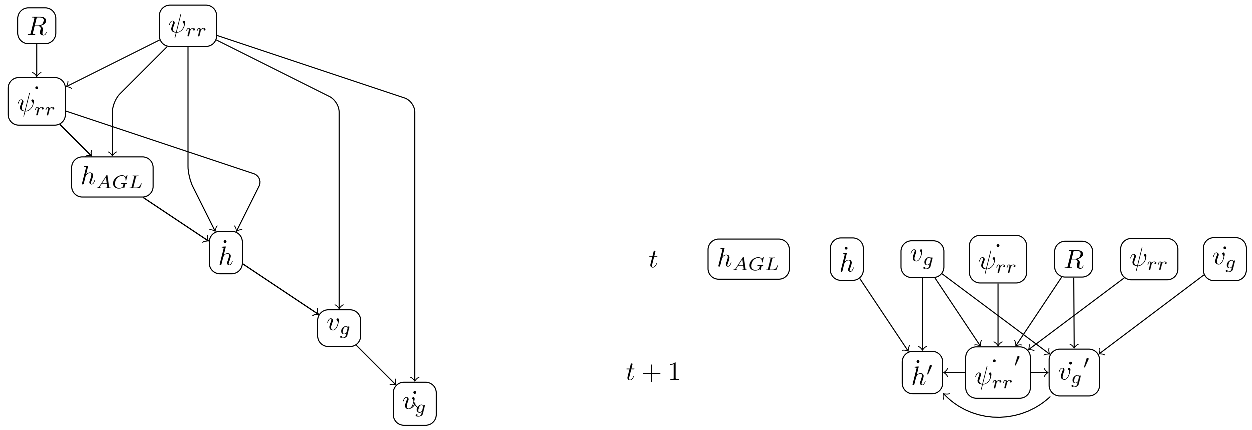
Table 3: Features used in preliminary encounter model

Feature	Description	Discretization
R	range from origin/operator's position	$\left(0, 11, 29, 101, 822, 10^4\right)\,\mathrm{ft}$
h_{AGL}	altitude above ground level	$(0, 10, 50, 100, 400, 5000)\,\mathrm{ft}$
$\dot{h}$	altitude rate (MSL and AGL identical)	$(-300, -23.5, -3.6, 3.6, 24.9, 300)\,\mathrm{ft/\,min}$
v_g	ground speed	$(0.0, 0.03, 0.1, 0.3, 1.75, 40.0)\,\mathrm{kts}$
$\dot{v_g}$	acceleration with respect to the ground	$\left(-5.0, -0.46, -5.1 \times 10^{-6}, 4.9 \times 10^{-5}, 0.46, 5.0\right)\,\mathrm{kts/s}$
ψ_{rr}	radial-relative heading	$(0, 45, 90, 135, 180, 225, 270, 315, 360)\,\mathrm{deg}$
$\dot{\psi_{rr}}$	radial-relative heading rate	$(-50, -8.7, -0.032, 0.032, 8.1, 50)\,\mathrm{deg\,/s}$

E. Bayesian Network Determination and Sampling

A discrete Bayesian network encodes the conditional probabilities of a feature taking on a particular value given the values of a set of parent features. A large data set of feature values (i.e., those extracted from a set of unmanned aircraft trajectories at each time step) may be used to determine the most likely probabilistic relationship between the variables. A number of Bayesian network search techniques exist,[15] most of them based on heuristics that add candidate edges between nodes to maximize a Bayesian score. The GeNIe software package, developed by the Decision Systems Laboratory of the University of Pittsburgh,[33] was used to conduct the structure search. The conditional probabilities in the network are determined directly from the observed frequency with which combinations of feature values appear.

The encounter model consists of two different Bayesian networks. The first, shown in Fig. 2a, represents the initial state distribution and is used to select a representative set of features with which to initialize the intruder aircraft's trajectory. The second network is a dynamic Bayesian network that consists of the set of features represented in Table 3, plus three features that represent the state of the aircraft at a subsequent time step. Additional details about the creation and sampling of the networks may be found elsewhere.[11]



(a) Initial distribution

(b) Transition distribution

Figure 2: Bayesian networks for intruder encounter modeling

F. Encounter Modeling

The intruder trajectory is only part of the encounter; a trajectory representing the ownship and controlled by the collision avoidance algorithm must intersect that of the intruder. A collision point along the intruder's trajectory is randomly selected. That point must satisfy encounter selection criteria set by the algorithm evaluator, and any trajectories that fail to meet the criteria are discarded. The velocity of the intruder at the CPA was required to lie within a range of $0.25\,\mathrm{s}^{-1}$ and $3.0\,\mathrm{s}^{-1}$. This prevents the sampling of a large proportion of intruders with nearly zero velocity.

Table 4: Encounter set parameters

Parameter	Values	Units
Relative headings at CPA	$45, 90, 135, 180, 225, 270, 315$	deg
Ownship initial velocity	1.0	s^{-1}
Nominal trajectory duration	120	s
Initial time to CPA	15	s

Each intruder trajectory meeting the selection criteria is rotated to seven different orientations according to Table 4 to ensure a range of relative headings at CPA are tested. An example of a set of seven encounters derived from a single sampled trajectory is shown in Fig. 3: each of the red lines represents one trajectory, and a single trajectory incorporates × symbols at each time step to provide a sense of the intruder's velocity during the encounter. The ownship trajectory is represented by a short green line with zero lateral deviation that starts at -15.0 and ends at the origin. In this encounter set the intruder makes a sharp right turn shortly after CPA, which can be challenging to resolve for certain ownship collision avoidance maneuvers.

The other parameters listed in Table 4 are needed to fully specify the encounter. The ownship velocity is always initialized to the nominal trajectory velocity used in the optimization step of the QMDP algorithm: unit velocity in the longitudinal direction and zero velocity in the lateral. The intruder trajectory is created by sampling from the transition Bayesian network until it is 120 s long. The time to CPA at which to initialize the ownship is 15 s, and this parameter together with the selected CPA location on the intruder

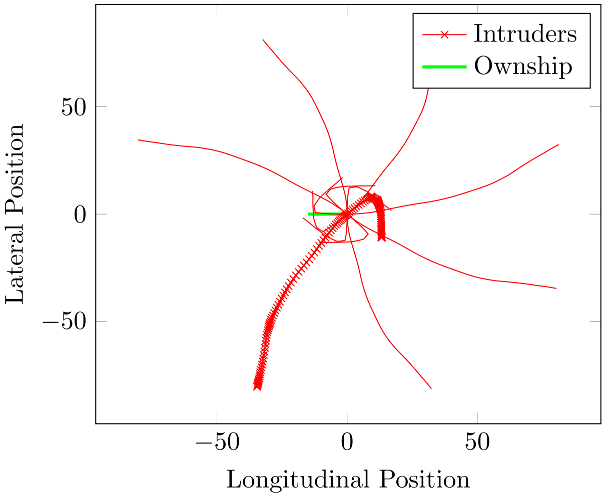


Figure 3: A set of simulated trajectories forming an encounter set

trajectory and nominal ownship velocity specifies the initial position of the ownship. The intruder follows this trajectory throughout the encounter (without any added dynamic uncertainty, though it is still subject to state uncertainty) and without executing any collision avoidance maneuvers.

IV. Results

This section presents the results of the algorithm evaluation. It first shows the tradeoff between separation and trajectory deviation that is achievable by different algorithm designs, and then selects a single design for each algorithm for the remaining results. These individual algorithm designs are compared first on a trajectory level, then as distributions of metrics for a pool of encounters, and finally aggregated to provide a concise metric comparisons as a function of intruder type and uncertainty.

A. Separation-Deviation Operating Curve

It is important to compare the performance of the algorithms not only at a single operating point, but also compare the tradeoffs between separation and deviation that are achievable by each. This analysis can show that certain algorithms are better at meeting larger separation requirements while other algorithms are more suited to meeting smaller trajectory deviation preferences. It also provides a user of a collision avoidance algorithm with a set of design points from which to select parameters that meet particular separation or deviation requirements.

The design points resulting from a sweep of separation parameters are shown in Fig. 4. On this chart the separation metric is the inverse 5% CPA distance so that lower values (higher separations) are preferred. The deviation metric is also better when its value is lower, so the ideal design points are in the bottom left portion of the chart. The non-dominated points for each algorithm, defined as those designs for which no other design can improve on both metrics simultaneously, are connected together to represent the Pareto-optimal front for that algorithm. Three algorithms (excluding MPC) perform similarly at the lower left design points, all clustering closely together. When smaller deviations and separations are desirable the PF algorithm performs best, with the QMDP approach also producing good behavior. The HVT algorithm, constrained to issue circular arc resolution maneuvers, is unable to achieve mean separations much below 1 without detracting from the deviation metric. In contrast, when larger deviations are allowable (in order to increase separation) the HVT algorithm performs well.

The MPC algorithm stands alone in delivering equivalent separation performance to the other three algorithms but with 0.5 to 1.0 units lower trajectory deviation. As described in sub-Section A, MPC is able to achieve the improved performance because it directly optimizes a set of continuous accelerations for a closely spaced sequence of action points without any computation time constraints. The manifestation of this optimal trade off is that the CPA is nearly always located at the point of maximum trajectory deviation, a desirable characteristic the resolution maneuvers of other algorithms do not always match.

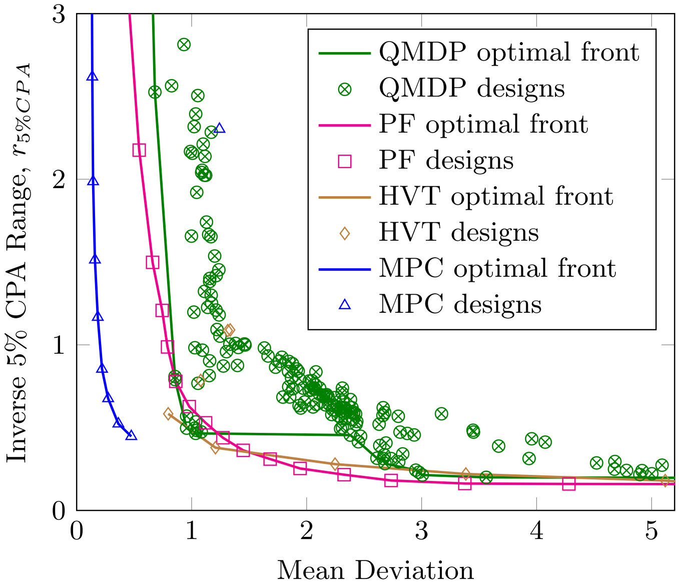


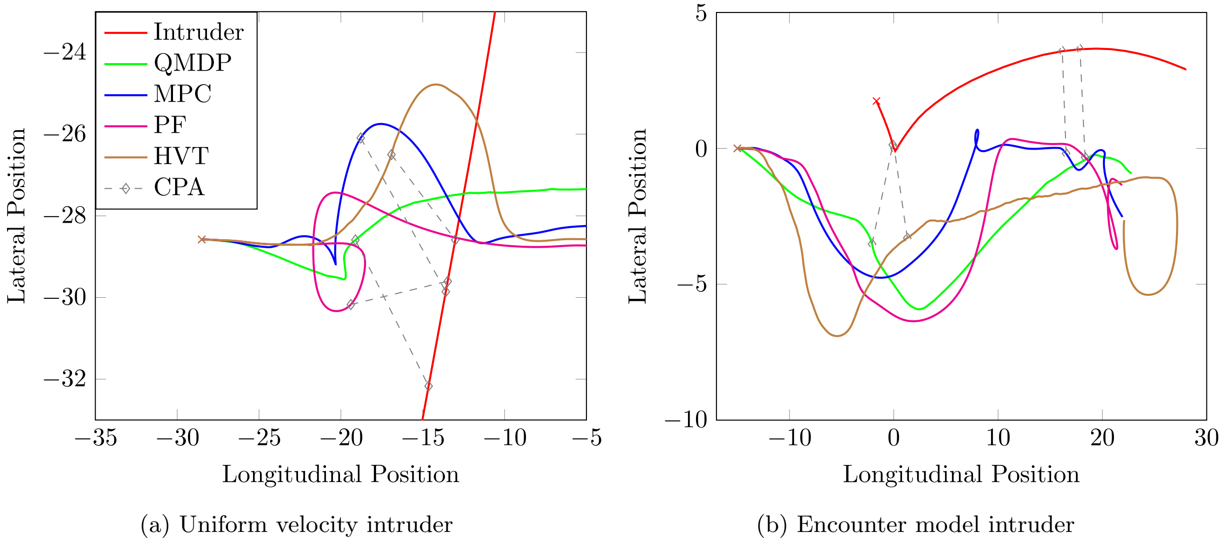
Figure 4: Separation-deviation operating curves for all algorithms

B. Individual Trajectory Comparisons

An important test of any collision avoidance algorithm is whether the maneuvers it recommends are consistent with the types of maneuvers an aircraft operator would select when faced with the same encounter. Trajectories typical of challenging encounters with uniform velocity intruders and encounter model trajectories are described in this section. These two encounter situations expose characteristic behavior from each of the algorithms that is useful for understanding the aggregate performance metrics presented later.

The separations achieved by each algorithm during the high-speed crossing encounter shown in Fig. 5a were relatively good when compared with an NMAC size of 1.0 unit, ranging between 4.4 (HVT) and 6.4 (MPC). The maximum deviations were also typical, being lowest for QMDP (3.42), MPC (3.43) and HVT (3.82), while PF fared poorly at 7.92. These CPA locations of the aircraft are shown as gray diamonds connected by dashed lines. MPC and PF make at least two reversals during the encounter, with the latter even circling around a complete 360 deg before returning to the reference trajectory. This circling behavior leads to the very large maximum deviation from the reference trajectory. In contrast, QMDP comes to a nearly complete stop to allow the intruder to pass by and then catches up to the reference trajectory with very little lateral deviation, though its slow return to zero lateral error leads to the highest mean deviation among the algorithms. The HVT algorithm has good separation performance and a reasonable maximum deviation metric value, though most of that deviation occurred in the lateral dimension and may be less desirable from an unmanned aircraft operator perspective.

The encounter model trajectory shown in Fig. 5b illustrates the very different intruder behavior that that model produces, along with typical responses from each of four algorithms. The intruder makes an initial maneuver in the negative lateral direction before pausing, changing directions and moving in an arc in the positive longitudinal direction. All the algorithms avoid the initial intruder acceleration segment, with MPC and PF quickly returning to the reference trajectory. This aggressive maneuvering means the CPA for those algorithms is towards the end of the encounter rather than after the initial maneuver, but all separations were acceptable. The mean trajectory deviations were larger than is typical for each algorithm (2.2 to 4.5) because of the longitudinal maneuver of the intruder, but the maximum deviations were in line with those of other encounters.



(a) Uniform velocity intruder

(b) Encounter model intruder

Figure 5: Typical algorithm trajectories for two classes of intruder

C. Distributions of Algorithm Metrics

This section presents the empirical cumulative distribution functions (CDF) of separation and deviation metrics across the entire pool of either uniform velocity intruders or encounter model trajectories, providing a basis for comparison of the algorithms' performance across a large number of trajectories.

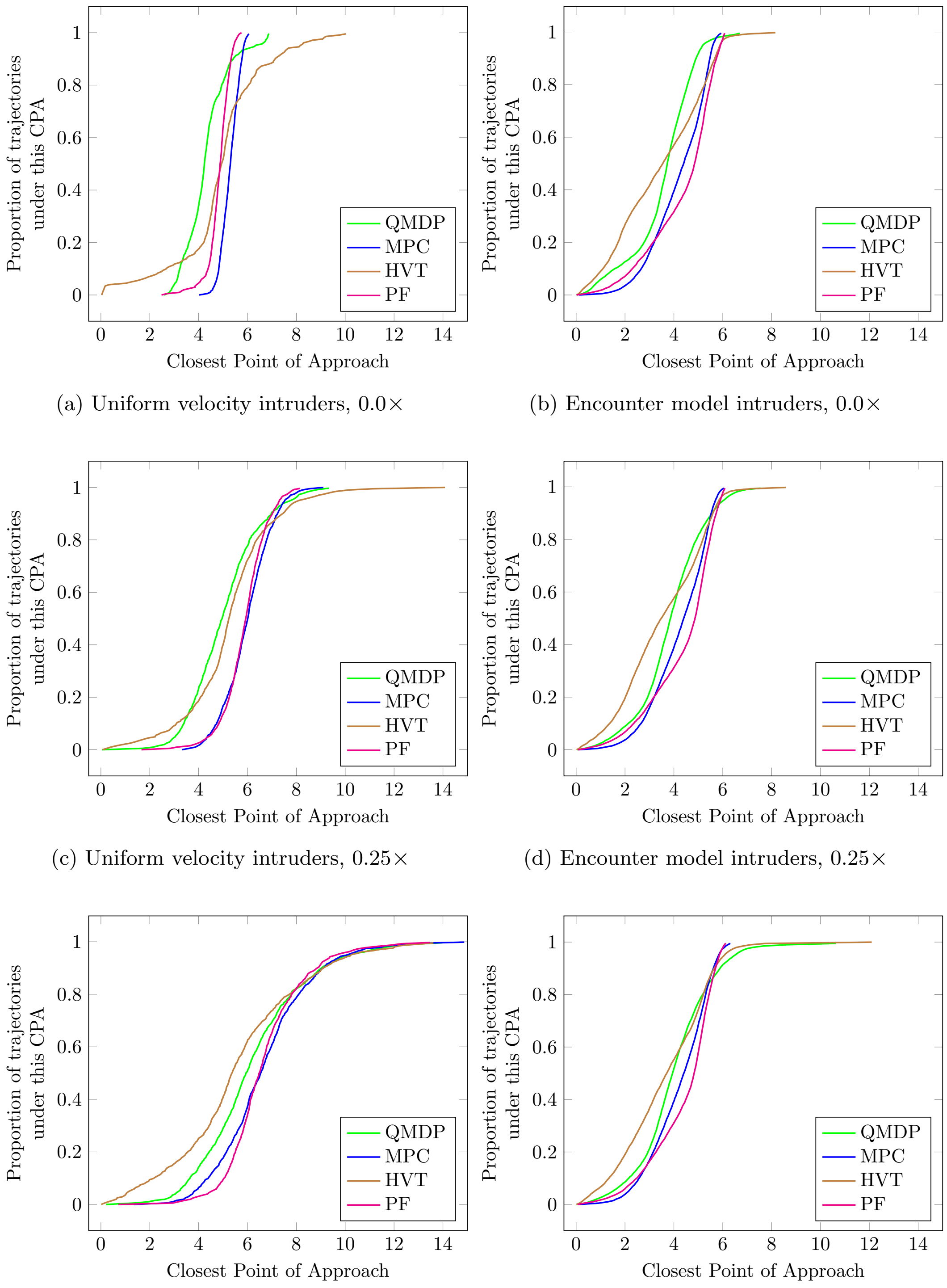
1. Separation at CPA

The key separation metric is the minimum distance between the aircraft over the entire encounter trajectory, and many of the aggregate and derived metrics come from this fundamental measure of the risk in a particular encounter. Distributions of the CPA separations of the four algorithms for uniform velocity intruders and encounter model trajectories at three different levels of uncertainty are shown in Fig. 6. Each curve in the charts represents the proportion of the relevant trajectory set that had a CPA, represented on the x-axis, that is *lower* than the given y-axis value. The point at which a horizontal line drawn through the y-axis at a value of 0.5 intersects each algorithms' CDF is the median value of the CPA separation. The point at which a vertical line drawn through the x-axis at a value of 1.0 intersects each algorithms' CDF is the proportion of encounters that resulted in NMACs.

Comparing the figures in the left column of Fig. 6, representing different levels of uncertainty for the uniform velocity intruders, with the figures in the right column, the encounter model trajectories, it is clear that a higher proportion of close encounters occurred during simulations of the latter. This result was expected and is due to changes in the intent of the intruder. These sudden and sustained accelerations change the heading of an aircraft and can be impossible for an algorithm to completely avoid. While several of the algorithms had no NMACs at the 0.0× and 0.25× uncertainty levels with uniform velocity intruders, all algorithms experienced these failures in the encounter model simulations.

In general, the PF and MPC algorithms deliver the largest CPA distances and the fewest close encounters. For the closest encounters, those under 3.0, the MPC algorithm has the fewest proportion for all combinations of uncertainty and intruder trajectory type. The PF algorithm closely matched the MPC algorithm in aggregate performance across many regimes, even surpassing it substantially with uniform velocity intruders and 1.0×uncertainty. The QMDP algorithm's performance approached that of the other two as uncertainty increased, achieving nearly the same distribution of minimum separations with encounter model intruders and 1.0×uncertainty. This algorithm is specifically designed to be robust to a variety of uncertainty types, further examination may find that at larger values of uncertainty or with a different set of encounter model intruders the QMDP algorithm performs better than its competitors.

The HVT algorithm is an outlier in terms of the number of close encounters for several of the combinations of uncertainty and intruder trajectory type, most notably the encounter model intruders at 0.0× and 0.25×



(a) Uniform velocity intruders, 0.0×

(b) Encounter model intruders, 0.0×

(c) Uniform velocity intruders, 0.25×

(d) Encounter model intruders, 0.25×

(e) Uniform velocity intruders, 1.0×

(f) Encounter model intruders, 1.0×

Figure 6: Cumulative distributions of CPA separations by algorithm

uncertainty. This result seems to be related to a feature of the algorithm that does not necessarily return a maneuver if it cannot find one that satisfies the algorithm-specific separation standard. In its original tactical separation application, this feature of HVT was useful because it indicated to the human air traffic controller that a resolution was not currently available. Given the two-minute time horizons and 5 nmi separations the algorithm was designed to alert at, there was still plenty of time to find an alternate resolution maneuver. Further development of the algorithm would be necessary to revise this feature if it were to be used for small aircraft collision avoidance.

2. Maximum Trajectory Deviation

The maximum trajectory deviation recorded over the entire encounter trajectory may be more useful than the mean deviation for a variety of applications. Distributions of the maximum deviations of the four algorithms for uniform velocity intruders and encounter model trajectories at three different levels of uncertainty are shown in Fig. 7. Each curve in the charts represents the proportion of the relevant trajectory set that had a maximum deviation, represented on the x-axis, that is *lower* than the given y-axis value.

The maximum deviation results are relatively consistent for both pools of intruders and at different levels of uncertainty, though the proportion of large maximum deviations with uniform velocity intruders at 1.0× uncertainty is higher than the other conditions. In all cases shown in Fig. 7 the MPC and QMDP maneuvers have the same maximum deviations for 60 to 90% of the encounters; the degree to which they deviate is essentially identical for the encounters requiring the least amount of deviation. For the small percentage of remaining encounters the MPC algorithm is able to find a maneuver with a smaller maximum deviation. Referring back to the CDFs of CPA separations in Fig. 6, the MPC algorithm had a larger separation even in these cases of smaller deviations, indicating that it is superior to the QMDP algorithm at balancing deviation and separation in this minority of encounter situations.

The PF algorithm performs worse on the maximum deviation metric than either MPC or QMDP, and in the case of 0.0× uncertainty it does much worse. The HVT algorithm also performs poorly on this metric, though it is not significantly worse than PF for the encounter model intruders and has smaller deviations for a subset of the uniform velocity intruders. The HVT algorithm's inability to command a maneuver other than a constant-radius turn and the lack of a mechanism with which to compensate for uncertainty other than making larger deviations account for its last-place performance on this metric.

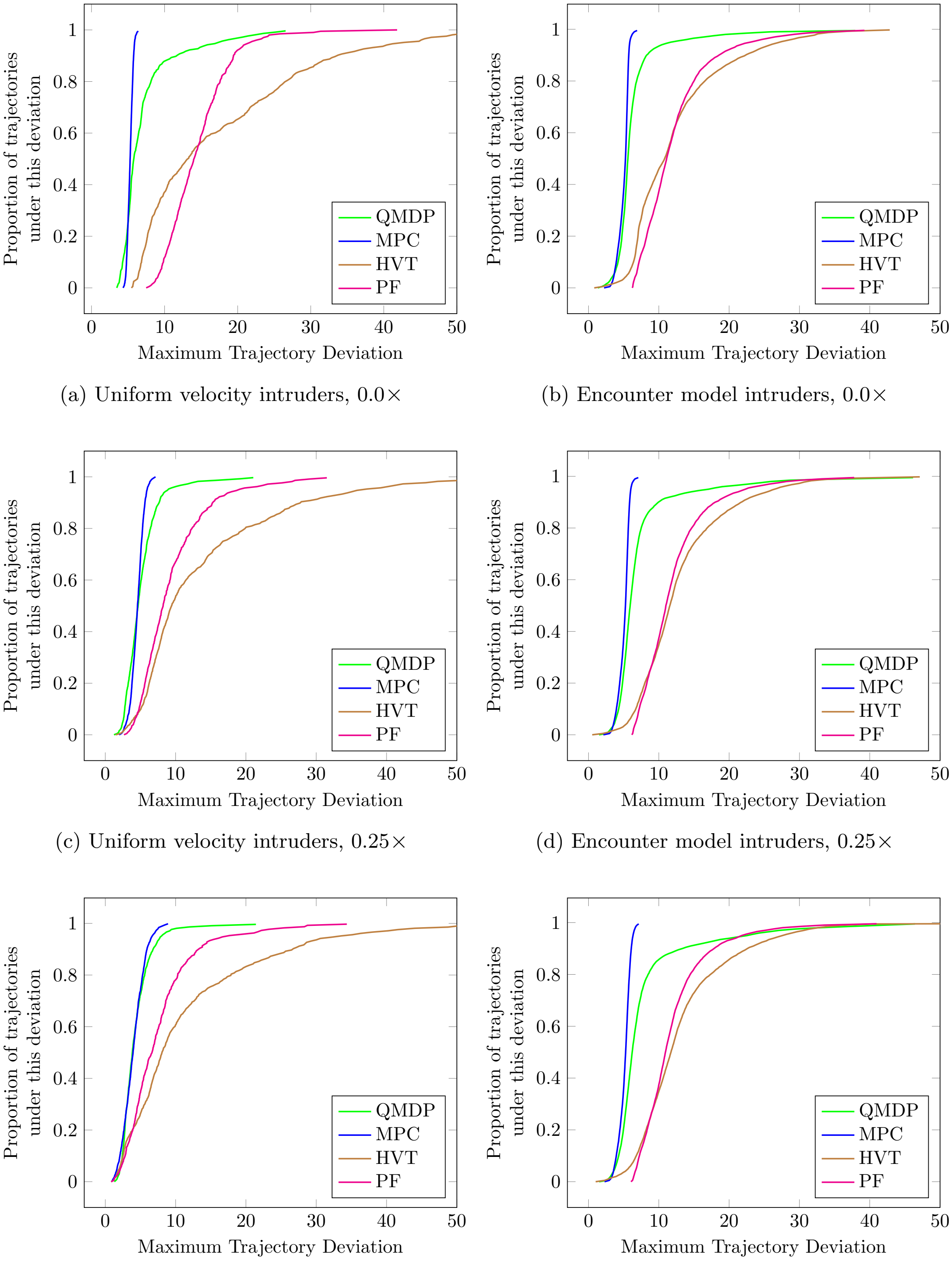
D. Aggregate Algorithm Comparison Metrics

This section uses aggregate metrics to combine the separation and deviation results from the previous section into a compact and concise summary of performance for each algorithm as a function of the level of uncertainty. These charts clarify the observations presented in the previous sections without providing the detailed distributions and trajectory histories that explain why the results occur.

1. Near Mid-Air Collisions

The proportions of NMACs that occurred as a function of algorithm and uncertainty factor for the uniform velocity encounters and encounter model intruders are shown in Fig. 8. For the uniform velocity intruders, both PF and MPC had virtually no NMACs, while QMDP had no more than a handful of them out of the 1320 encounters for any uncertainty level (maximum percentage was 0.45%). The HVT algorithm had a much higher rate for reasons described earlier, with peak values at both 0.0×and 1.0×uncertainty. The peak at the low level of uncertainty results almost entirely from head-on encounters with high speed intruders in which the algorithm does not find a satisfactory maneuver and therefore intentionally returns nothing. A small change to the algorithm would allow it to return the maneuver that would maximize the CPA separation rather than no maneuver.

The proportion of NMACs is much higher for the encounter model trajectories than the uniform velocity intruders. The MPC algorithm is able to limit the rate to between 0.45 and 0.57%, a rate that is insensitive to the level of uncertainty. The PF algorithm delivered the next lowest rate, ranging from 1.7 to 2.0%, that was again insensitive to uncertainty. For most of the uncertainty values the QMDP algorithm was on par with PF, 2.4 to 3.0%. Interestingly, the highest rates for the QMDP algorithm occurred at the lowest uncertainties: 5.9% at 0.0× uncertainty and 3.0% at 0.125× uncertainty. Although the assumed degree of uncertainty used to calculate the belief state did not change between these conditions, the reduced or



(a) Uniform velocity intruders, 0.0×

(b) Encounter model intruders, 0.0×

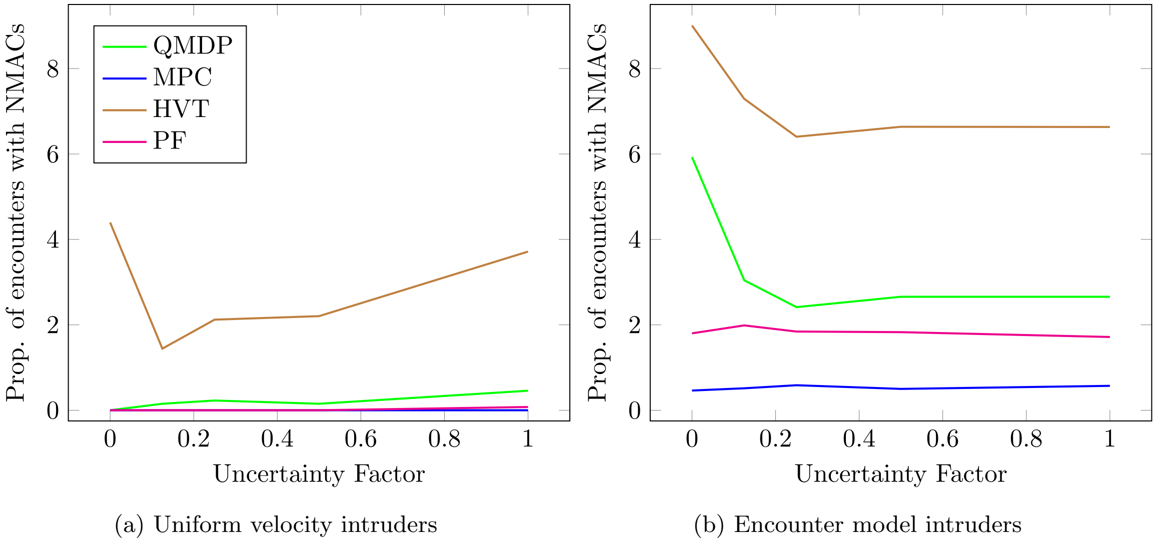
(c) Uniform velocity intruders, 0.25×

(d) Encounter model intruders, 0.25×

(e) Uniform velocity intruders, 1.0×

(f) Encounter model intruders, 1.0×

Figure 7: Cumulative distributions of maximum trajectory deviation by algorithm



(a) Uniform velocity intruders

(b) Encounter model intruders

Figure 8: Proportion of NMACs as a function of algorithm and uncertainty factor

non-existent state uncertainty may have prevented the beliefs from "considering" the sustained accelerations that were possible in the encounter model. At higher levels of uncertainty the QMDP algorithm may have taken action because of the state uncertainty and unwittingly increased separation that prevented an NMAC. Extensions to the QMDP algorithm are exploring the degree to which belief uncertainty can remove these NMACs.

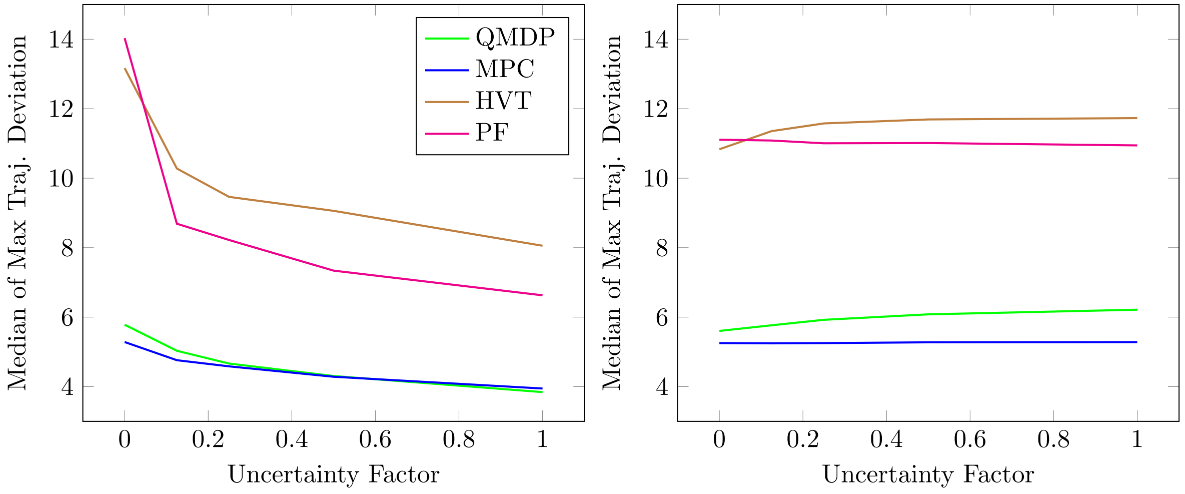
2. Median Value of Maximum Trajectory Deviation

The median values of the maximum deviation among all the trajectories for the uniform velocity and encounter model intruders as a function of uncertainty are shown in Fig. 9. As described in the sections showing the individual CDFs, uncertainty in the uniform velocity intruder condition has the effect of lowering the maximum deviation for each algorithm because dynamic uncertainty diverts the intruders from a direct collision and makes the required deviation smaller. The MPC and QMDP algorithms' performance on the maximum deviation metric is nearly identical except for a small discrepancy at 0.0× uncertainty. This similar performance indicates that both are creating maximum deviations that are likely near the limit of what deviation is required to achieve a given separation. The PF and HVT algorithms both do poorly at 0.0× uncertainty but substantially improve at higher values, and the PF algorithm is better at each evaluated level than HVT.

The maximum deviations for the encounter model intruders are largely insensitive to the level of uncertainty, though the median value for QMDP does increase very slightly with increasing uncertainty. Unlike in the uniform velocity case, MPC outperforms QMDP slightly against encounter model intruders, but their scores are approximately equal. The PF algorithm is a distant third place, while HVT performs only slightly worse than PF.

3. 5% Threshold Separation

The 5% threshold separation is shown as a function of algorithm and uncertainty factor for the uniform velocity encounters and encounter model intruders in Fig. 10. Because this metric is driven only by the CPA separation of the 5th percentile encounter, which itself does not have a target separation requirement, the values achieved by each algorithm must be interpreted in the context of their maximum trajectory deviations. For example, although the MPC and PF algorithms have desirable 5% separations of 4.0 to 5.0 in the uniform velocity case, the MPC algorithm delivers the separation with about half the maximum trajectory deviation of PF. The QMDP algorithm has a slightly smaller separation of about 3.0, but has the same small deviation as MPC. The encounter model trajectories show an even smaller decrease in QMDP's separation relative to PF with a much larger improvement in maximum deviation. An algorithm user

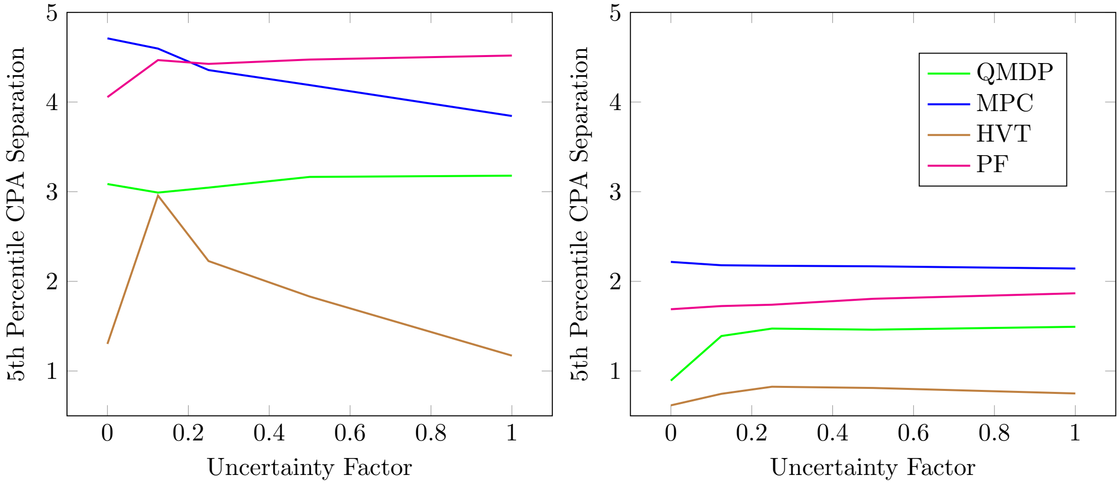


(a) Uniform velocity intruders

(b) Encounter model intruders

Figure 9: Median value of maximum trajectory deviation as a function of algorithm and uncertainty factor

choosing between PF and QMDP (perhaps MPC cannot be implemented in real time) would have to decide whether gaining 1.0 unit of separation is worth increasing the maximum trajectory deviation by 5.0. The separations with the encounter model trajectories are lower for every algorithm and level of uncertainty than for the uniform velocity intruders, confirming that this pool of intruders presents more challenging collision avoidance encounter situations.



(a) Uniform velocity intruders

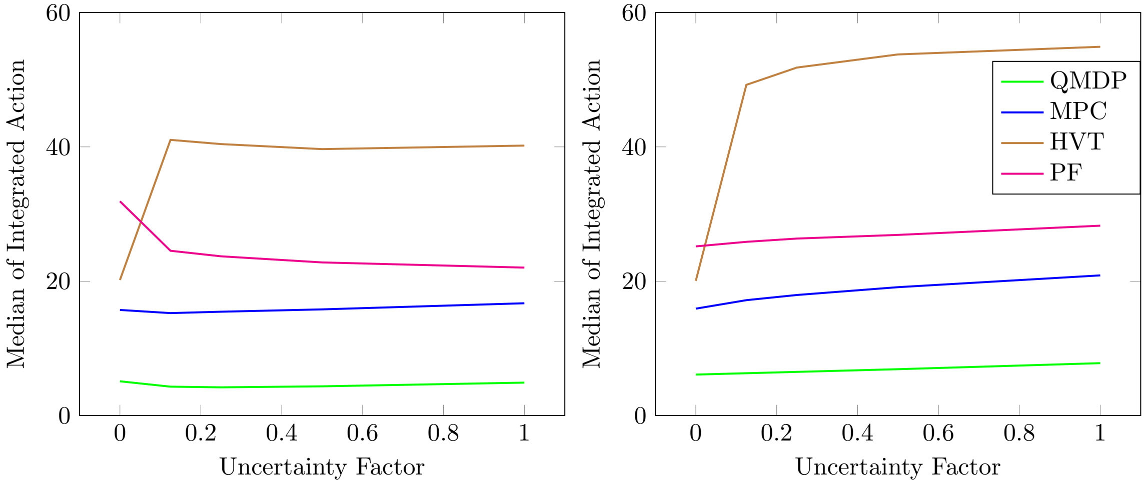
(b) Encounter model intruders

Figure 10: Separation distance violated by 5% of intruders as a function of algorithm and uncertainty factor

E. Integrated Action

The integrated action over a trajectory is essentially the net velocity change required by the algorithm to avoid an intruder. The integrated action is shown as a function of algorithm and uncertainty factor for uniform velocity intruders and encounter model trajectories in Fig. 11. The QMDP algorithm is clearly superior to the others for both types of intruder trajectories and across all levels of uncertainty, requiring only 40 to 43% of the integrated action of the next nearest algorithm, MPC, for encounter model intruders and only 28 to 36% for uniform velocity intruders. The primary reason for QMDP's superiority on this metric is that it allows only a single unit magnitude acceleration in one axis at a time, while MPC is free to choose any continuous value of acceleration up to the limit of ±1.0. This limit means that QMDP will frequently sacrifice trajectory tracking performance because an acceleration of 1.0 may cause an overshoot that will result in a lower reward than commanding 0.0 acceleration and accepting the deviation error.

An interesting aspect of this action metric is that it appears to have very little dependence on the level of uncertainty. Control systems that have to contend with dynamic uncertainty frequently expend more control authority to counteract that uncertainty. With the exception of HVT, the algorithms in these encounters are not tracking a reference trajectory while making their collision avoidance maneuvers. Instead, they are making a control decision at each step in the simulation that is largely dependent on the intruder's relative state, not the immediate value of state or dynamic uncertainty. The uncertainty does not change the relative state significantly. The integrated action metric presents different insights into the performance of each algorithm that may be useful in certain types of UAS applications.



(a) Uniform velocity intruders

(b) Encounter model intruders

Figure 11: Median integrated action as a function of algorithm and uncertainty factor

V. Conclusions

Four algorithms have been designed and implemented in a simulation framework that allows direct comparison of their performance under a variety of encounter situations using separation and trajectory deviation metrics. The first algorithm (QMDP) is an extension of the next-generation manned aircraft algorithm that poses the collision avoidance problem as a partially observable Markov decision process and solves for the approximately optimal solution using dynamic programming. The second algorithm (PF) is based on a potential field method, and the third (HVT) is an adaptation of a tactical conflict detection and resolution algorithm that uses candidate trajectory predictions to determine a preferred resolution. The fourth algorithm (MPC) is based on receding-horizon model predictive control. Two different pools of intruder trajectories are used to evaluate each algorithm. The first is a set of largely uniform velocity intruders that encounter the collision avoidance algorithm-equipped ownship at a comprehensive set of relative heading angles and velocities. The second set of intruder trajectories was created from a novel encounter model that is based on the statistical characteristics of hobbyist unmanned aircraft extracted from a database of 75,000 such flights. Unique state and dynamic uncertainty histories were added to each intruder trajectory to facilitate a fair comparison between the algorithms.

Comprehensive evaluation of the algorithms showed that each may be tuned to provide approximately the same tradeoff between separation from other aircraft and deviation from the reference trajectory. The algorithms were tuned by selecting different parameter values that delivered larger or smaller intruder separations, allowing the user of such an algorithm to select the design that meets the separation requirements of their application. The unlimited computational resources available to the model predictive control algorithm in this simulation environment allowed it to surpass the others on both separation and deviation metrics, though in a real-world implementation of the algorithm it may not be possible to achieve this same level of performance and robustness. The potential field algorithm was generally robust to accelerating intruders and consistently ensured those aircraft did not get so close they posed a safety risk. However, in some encounters this robust separation capability came at the expense of large trajectory deviations. The algorithm that was adapted from a fixed-wing tactical separation algorithm suffered from several implementation issues and frequently had large trajectory deviations without guaranteeing that close encounters would be prevented. The newly developed algorithm based on the offline solution of a partially observable Markov decision process delivered excellent maximum deviation performance with only slightly smaller separations than the model predictive control algorithm. The new algorithm achieves this performance with less than half the required velocity change of any of the other algorithms.

Acknowledgments

The authors would like to thank NASA for supporting Eric Mueller during the completion of his PhD research, support that resulted, in part, in the work presented in this paper. Andrew Weinert, working at MIT Lincoln Laboratories under contract to NASA, discovered and downloaded the initial hobbyist trajectory database, which was the key precursor to development of the encounter model. The authors also wish to thank the FAA TCAS Program Manager, Neal Suchy.

References

[1] Kochenderfer, M. J., Chryssanthacopoulos, J. P., Kaelbling, L. P., Lozano-Perez, T., and Kuchar, J. K., “Model-Based Optimization of Airborne Collision Avoidance Logic,” Project Report ATC-360, Massachusetts Institute of Technology, Lincoln Laboratory, 2010.

[2] Erzberger, H. and Heere, K., “Algorithm and Operational Concept for Resolving Short-Range Conflicts,” *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 224, No. 2, 2010, pp. 225–243.

[3] Chryssanthacopoulos, J. P. and Kochenderfer, M. J., “Analysis of Open-loop and Closed-loop Planning for Aircraft Collision Avoidance,” *IEEE International Conference on Intelligent Transportation Systems*, October 2011, pp. 212–217.

[4] Klaus, R. and McLain, T., “A Radar-Based, Tree-Branching Sense and Avoid System for Small Unmanned Aircraft,” *AIAA Guidance, Navigation and Control Conference*, August 2013.

[5] Richards, A. and How, J. P., “Aircraft Trajectory Planning with Collision Avoidance using Mixed Integer Linear Programming,” *American Control Conference*, May 2002.

[6] Kuwata, Y. and How, J. P., “Three Dimensional Receding Horizon Control for UAVs,” *AIAA Guidance, Navigation and Control Conference*, August 2004.

[7] Kuwata, Y., *Trajectory Planning for Unmanned Vehicles using Robust Receding Horizon Control*, Ph.D. thesis, Massachusetts Institute of Technology, 2007.

[8] Nieuwenhuisen, M., Schadler, M., and Behnke, S., “Predictive Potential Field-Based Collision Avoidance for Multicopters,” *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XL-1/W2, September 2013, pp. 293–298.

[9] Kochenderfer, M. J. and Chryssanthacopoulos, J. P., “Robust Airborne Collision Avoidance through Dynamic Programming,” Project Report ATC-371, Massachusetts Institute of Technology, Lincoln Laboratory, Jan. 2011.

[10] Mueller, E. and Kochenderfer, M. J., “Multi-Rotor Aircraft Collision Avoidance using Partially Observable Markov Decision Processes,” *AIAA Modeling and Simulation Technologies Conference*, 2016.

[11] Mueller, E. R., *Multi-Rotor Aircraft Collision Avoidance using Partially Observable Markov Decision Processes*, Ph.D. thesis, Stanford University, 2016.

[12] Bezanson, J., Karpinski, S., Shah, V. B., and Edelman, A., “Julia: A Fast Dynamic Language for Technical Computing,” *Computing Research Repository*, Vol. abs/1209.5145, 2012.

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

[14] Amato, C., Chowdhary, G., Geramifard, A., Ure, N. K., and Kochenderfer, M. J., “Decentralized Control of Partially Observable Markov Decision Processes,” *IEEE Conference on Decision and Control*, Florence, Italy, 2013.

[15] Kochenderfer, M. J., *Decision Making Under Uncertainty: Theory and Application*, MIT Press, 2015.

[16] Chao, Z., Ming, L., Shaolei, Z., and Wenguang, Z., “Collision-free UAV formation flight control based on nonlinear MPC,” *International Conference on Electronics, Communications and Control*, Sept 2011, pp. 1951–1956.

[17] Ross, S., Pineau, J., Paquet, S., and Chaib-draa, B., “Online Planning Algorithms for POMDPs,” *Journal of Artificial Intelligence Research*, Vol. 32, 2008, pp. 663–704.

[18] Khatib, O., “Real-time obstacle avoidance for manipulators and mobile robots,” *IEEE International Conference on Robotics and Automation*, Vol. 2, March 1985, pp. 500–505.

[19] Sigurd, K. and How, J., “UAV Trajectory Design Using Total Field Collision Avoidance,” *AIAA Guidance, Navigation and Control Conference*, 2003.

[20] Argyle, M., Chamberlain, C., and Beard, R., "Chain-Based Path Planning for Multiple UAVs," *IEEE Conference on Decision and Control and European Control Conference*, December 2011.

[21] Sahawneh, L., Beard, R. W., Avadhanam, S., and Bai, H., "Chain-based Collision Avoidance for UAS Sense-and-Avoid Systems," *AIAA Guidance, Navigation, and Control Conference*, 2013.

[22] Erzberger, H. and Heere, K., "Algorithm and Operational Concept for Resolving Short-Range Conflicts," *International Congress of the Aeronautical Sciences*, 2008.

[23] Paielli, R. A., Chiu, H. E. D., and Heere, K. R., "Tactical Conflict Alerting Aid for Air Traffic Controllers," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 1, 2009, pp. 184–193.

[24] Bach, R., Farrell, C., and Erzberger, H., "An Algorithm for Level-Aircraft Conflict Resolution," Tech. Rep. CR-2009-214573, NASA, 2009.

[25] Jarrell, J. A., *Employ Sensor Fusion Techniques for Determining Aircraft Attitude and Position Information*, Ph.D. thesis, West Virginia University, 2008.

[26] Weibel, R. E., Edwards, M. W. M., and Fernandes, C., "Establishing a Risk-Based Separation Standard for Unmanned Aircraft Self Separation," *AIAA Aviation Technology, Integration and Operations Conference*, 2011.

[27] Griffith, J. D., Kochenderfer, M. J., and Kuchar, J. K., "Electro-optical system analysis for sense and avoid," *AIAA Guidance, Navigation, and Control Conference*, Honolulu, Hawaii, 2008.

[28] Kochenderfer, M. J., Edwards, M. W. M., Espindle, L. P., Kuchar, J. K., and Griffith, J. D., "Airspace Encounter Models for Estimating Collision Risk," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 2, 2010, pp. 487–499.

[29] Weinert, A. J., Harkleroad, E. P., Griffith, J. D., Edwards, M. W. M., and Kochenderfer, M. J., "Uncorrelated Encounter Model of the National Airspace System Version 2.0," Project Report ATC-404, Massachusetts Institute of Technology, Lincoln Laboratory, 2013.

[30] Kochenderfer, M. J., Espindle, L. P., Kuchar, J. K., and Griffith, J. D., "Correlated Encounter Model for Cooperative Aircraft in the National Airspace System," Project Report ATC-344, Massachusetts Institute of Technology, Lincoln Laboratory, 2008.

[31] Edwards, M. W. M., Kochenderfer, M. J., Kuchar, J. K., and Espindle, L. P., "Encounter Models for Unconventional Aircraft," Project Report ATC-348, Massachusetts Institute of Technology, Lincoln Laboratory, 2009.

[32] Kochenderfer, M. J., Chryssanthacopoulos, J. P., and Weibel, R. E., "A new approach for designing safer collision avoidance systems," *Air Traffic Control Quarterly*, Vol. 20, No. 1, 2012, pp. 27–45.

[33] Druzdzel, M. J., "SMILE: Structural Modeling, Inference, and Learning Engine and GeNIe: A development environment for graphical decision-theoretic models," *AAAI National Conference on Artificial Intelligence*, 1999.