17th AIAA Aviation Technology, Integration, and Operations Conference

The Generic Resolution Advisor and Conflict Evaluator (GRACE) for Detect-And-Avoid (DAA) Systems

Michael Abramson *
Crown Consulting Inc, Moffett Field, California, 94035

Mohamad Refai †
Crown Consulting Inc, Moffett Field, California, 94035

Confesor Santiago ‡
NASA Ames Research Center, Moffett Field, California, 94035

The paper describes the Generic Resolution Advisor and Conflict Evaluator (GRACE), a novel alerting and guidance algorithm that combines flexibility, robustness, and computational efficiency. GRACE is "generic" in that it makes no assumptions regarding temporal or spatial scales, aircraft performance, or its sensor and communication systems. Accordingly, GRACE is well suited to research applications where alerting and guidance is a central feature and requirements are fluid involving a wide range of aviation technologies. GRACE has been used at NASA in a number of real-time and fast-time experiments supporting evolving requirements of DAA research, including parametric studies, NAS-wide simulations, human-in-the-loop experiments, and live flight tests.

Glossary

ACAS	Airborne Collision Avoidance System
CPA	Closest Point to Approach
DAA	Detect-and-Avoid
DWC	DAA Well Clear
GCE	Generic Conflict Evaluator
GCS	Ground Control Station
GRA	Generic Resolution Advisor
GRACE	Generic Resolution Advisor and Conflict Evaluator
HITL	Human-in-the-Loop
JADEM	Java Architecture for DAA Extensibility and Modeling
NAS	National Airspace
NMAC	Near Mid-Air Collisions
TCAS	Traffic Alert and Collision Avoidance System
TCP	Trajectory Change Point
UAS	Unmanned Aircraft Systems
VFR	Visual Flight Rules

*Senior Engineer, M/S 210-8, michael.abramson@nasa.gov
†Senior Engineer, M/S 210-8, mohamad.s.refai@nasa.gov
‡Aerospace Engineer, Aviation Systems Division, M/S 210-10, Member AIAA, confesor.santiago@nasa.gov

I. Introduction

Aviation is undergoing rapid transformation, with increasing levels of automation and new technology and operational concepts continually being introduced. A key factor common to all is operational safety. Safe separation between aircraft has been and continues to be a key driving force in air traffic management and related research. In today's airspace, separation is maintained by controllers using generous separation criteria and by pilots using highly subjective means dependent on visual acuity. While this has proven sufficient in the past, increasing airspace density and technical innovations are pushing the limits of the system. Introduction of Unmanned Aerial Systems (UAS) into the airspace would further complicate the situation because unmanned aircraft cannot "see and avoid" other traffic.

Integration of UAS in the National Airspace System (NAS) therefore requires a robust separation capability to meet FAA requirement to "see and avoid" and, when passing proximate traffic, to remain "well clear" as mandated for manned aircraft in 14 CFR §91.113,[1] "Rules of the Road". The Detect-and-Avoid (DAA) concept was introduced to satisfy this requirement for unmanned aircraft. DAA systems are intended to enable UAS to remain "well clear" and avoid collisions with other airborne traffic,[2–4] and to do so they require an objective definition of "well clear" (referred to as DAA Well Clear or DWC). DAA is required to provide detection and guidance to maintain DWC and, where DWC is lost, to provide recovery guidance to regain it.

DAA systems are envisioned as a means to maintain a low level of risk of Near Mid-Air Collisions[6] (NMAC). To accomplish this goal, it is essential to investigate a number of objective definitions of Well Clear[7] and to quantify the levels of risk achieved by proposed DAA concepts. To that end, NASA has developed a portable software system, Java Architecture for DAA Extensibility and Modeling (JADEM), that supports various DAA algorithms and can interface with different fast-time and real-time simulation tools and live systems.[8] The target users of JADEM are researchers who need to simulate the main features of proposed DAA systems before those systems are fully developed by manufacturers. The availability of a general purpose, flexible, and fast conflict detection and resolution algorithm was critical for the purpose. The Generic Resolution Advisor and Conflict Evaluator (GRACE)[8] proved suitable for that role. In addition to its core alerting and resolution features, GRACE powers JADEM's implementation of the recently introduced bands and well clear recovery guidance concepts (see Ref. 8).

GRACE was used in a number of NASA NAS-wide fast-time simulations, real-time Human-in-the-Loop (HITL) simulations with participation of experienced UAS pilots and air traffic controllers, and flight tests with live aircraft.[9–18] The complex and rapidly evolving requirements of these studies demanded and created ample opportunities for extensive evaluation of GRACE in a broad range of operational scenarios and applications.

The goal of this paper is to describe GRACE with sufficient algorithmic detail to be of interest to researchers involved in air traffic simulations and to developers of practical DAA systems for manufacture and deployment. The next section provides an overview of previous work related to development of conflict detection and resolution algorithms. Section III describes GRACE features and algorithmic details. Section IV provides sample results of GRACE evaluation. Finally, concluding remarks are presented in Section V.

II. Background

A number of conflict detection and resolution algorithms have been proposed and actively used for decades for various research projects (see Ref. 4, 19 and the references therein). Several such algorithms were inspired by the Traffic Alert and Collision Avoidance System (TCAS)[20] that has been very successful in preventing mid-air collisions for manned aircraft. These systems rely on the cooperative behavior of aircraft, which actively communicate their state and intent data,[21,22] hence said systems cannot be used with non-cooperative aircraft.

Some algorithms were specifically designed for certain types of onboard sensors, such as radars[21] or optical sensors.[23] Other algorithms, such as Jointly Optimal Collision Avoidance, are in principle sensor-agnostic,[24] but are tuned for a specific aircraft response model. Another next-generation algorithm, the Airborne Collision Avoidance System (ACAS), proposed to replace TCAS,[25] is more flexible and can be adapted to different aircraft, including UAS, by updating the probabilistic lookup table driving the ACAS threat logic. However, these algorithms do not use separation standards that would be needed to maintain Well Clear as defined by RTCA Special Committee 228.[8] Note that DAA support is currently being added

to ACAS-Xu (the version of ACAS supporting UAS).

Other algorithms, more suitable for Separation Assurance in Air Traffic Control, include the Profile Selector En Route (PFS-E) component of the Center / TRACON Automation System,[26] the Advanced Airspace Concept Autoresolver, and the Tactical Separation-Assured Flight Environment (TSAFE).[27] However for use in DAA, these algorithms would require non-trivial modifications to accommodate smaller look-ahead times and spatial separation standards, introduction of temporal separation, the lack of trajectory intent information, and more frequent update rates.

More recent algorithms, "inspired by nature," include "force field" methods,[28–30] the closely related "light propagation" model,[31] and "navigation function-based" algorithms that model aircraft as particles moving in a potential field formed by other aircraft with the same "charge" and hence generating a "repulsive force."[32,33] These algorithms demonstrated their effectiveness in guaranteed collision avoidance, but they can lead to overcosts and large deviations from nominal routes.[35]

A number of algorithms are based on optimization methods, which provide a very general framework for non-linear systems (see Ref. 4 and references therein). These methods typically use a mathematically rigorous Hamilton-Jacobi approach or heuristic optimization algorithms, such as simulated annealing[34] or genetic algorithms.[35] These methods are efficient enough for strategic flight planning, but they are still not sufficiently fast for real-time applications that require update rates on the order of one second. Another drawback of these methods is that they model smooth trajectories suitable for future Flight Management System abilities, which are not (currently) practical for use in UAS DAA systems.

The Generic Resolution Advisor and Conflict Evaluator (GRACE), the subject of this paper, was inspired by the ideas of force field[28–30] and complexity theory,[36] and it leverages the computational efficiency of grid-based methods.[37,38] GRACE provides the benefits of flexibility, robustness, and good computational performance, which make it more suitable for evolving requirements of research and modeling of future DAA systems. It was therefore adopted in 2013 to provide the core alerting and guidance functions of NASA's Java Architecture for DAA Extensibility and Modeling (JADEM).[8]

It should be noted that JADEM was expressly designed to evaluate DAA concepts using different algorithms. For example, the Detect-and-Avoid Alerting Logic for Unmanned Systems (DAIDALUS),[39] released to open source in 2015, is being integrated into JADEM as an alternate alerting and guidance module.

III. Generic Resolution Advisor and Conflict Evaluator (GRACE)

As implemented in JADEM, GRACE is an aircraft-centric algorithm that works to predict and prevent collision threats to a single unmanned aircraft (the "ownship") from all other aircraft ("intruders").

Using the taxonomy defined in Ref. 19, GRACE can be characterized as:

Dimensions:	Horizontal and Vertical planes (3D)
Detection:	Explicit conflict detection threshold (but user-defined)
Resolution:	Optimized
Maneuvers:	Turns, Vertical maneuvers, and Speed changes
Multiple:	Global

GRACE is a combination of two loosely coupled algorithms. The Generic Conflict Evaluator (GCE) provides a customizable implementation of conflict detection functionality. The Generic Resolution Advisor (GRA) is a fast general purpose conflict resolver.

GRACE does not make any assumptions regarding temporal or spatial scales, performance capabilities of aircraft, or its sensor and communication systems. This flexibility is achieved by using customizable separation standards in GCE and a cost function in GRA, and by reliance on an external trajectory predictor for aircraft flight modeling.[8] In particular, this allows GRACE to be used for rotorcraft or other unconventional UAS.

III.A. Generic Conflict Evaluator (GCE)

GCE is an efficient deterministic algorithm for assessing intruder threats based on user-defined "separation standards" specified by logical conditions that can include almost any combination of the following (along with corresponding thresholds and filtering conditions):[8,40]

- horizontal separation;

- vertical separation;
- tau-separation based on "simple tau" or "modified tau";
- horizontal miss distance;
- time to Closest Point of Approach (CPA);
- time to first loss (violation) of separation.

In this paper, the term "violation" (of separation) will be used to indicate that a logical condition that defines a separation standard is violated.

GCE can support several different "levels" of threat detection, each level being defined by its own user-specified separation standard. For instance, a separation standard for the highest severity threat can be set to NMAC as defined in TCAS ($500 \times 100\,ft$).

III.A.1. Dynamic Grid Mapping and Threat Detection

GCE can be used in two different roles:

1. as initial conflict detection to find the threats that may require resolution; in this role it is called once for each call to GRACE;
2. in GRA to assess whether candidate maneuvers will result in a conflict-free trajectory; in this role GCE can be called many times (once per candidate maneuver); for this reason, its computational performance is critical.

GCE computation time is essentially linear with the number of intruders. Therefore a key performance challenge is to efficiently identify aircraft that can potentially create threats prior to making any geometrical computations, which can be time-consuming. This needs to be done for arbitrary aircraft positions, which can deviate from the nominal route (based on intent) or from previously proposed resolutions.

GCE uses a fast algorithm that avoids distance calculations. The algorithm is based on an airspace model consisting of identical discrete elements, or cells (the model is referred to as the "grid map"). The grid map provides a quick way to find intruders close to any arbitrary ownship state by "mapping" all intruder trajectories predicted within a specified "look-ahead time" to a sequence of 2D grids, one grid per each time step.[8] This can be seen as a deterministic variant of the method proposed in Ref. 37.

The GCE algorithm involves three main operations:

1. Predicting intruder trajectories at discrete time steps up to a specified look-ahead time.
2. Mapping intruders by locating them in grid cells using simple comparisons between the intruder coordinates and cell boundaries. Note that the grid map is stored efficiently as a time sequence of hash maps populated only with "occupied" cells.
3. Detecting threats from mapped intruders. This uses simple comparisons between cell boundaries and a bounding box with a side 2δ around the ownship's position to identify potential intruders, where δ is a horizontal distance deduced from the separation standards (Fig. 1). Intruders found within this bounding box are passed to the state-based threat detection logic, which uses specified separation standards to evaluate threats. A threat is considered "detected" if it is found to violate separation standards at least at one timestep.

If more than one threat is "detected", GCE returns the first threat with the earliest conflict start time, which is defined as the time of the first state that violated the separation standard.

Note that the first operation, which is expensive, is only done once, the resulting grid mapping subsequently being used in conflict resolution. This dramatically reduces GRACE computation time.

III.B. Generic Resolution Advisor

The Generic Resolution Advisor (GRA) is the conflict resolution component of GRACE. It relies on the output from GCE (including the highest priority declared threat and mapping intruders to grid cells). The main output of GRA is a resolution that includes a recommended avoidance maneuver and a corresponding ownship predicted trajectory.

GRA uses fast analytical transformations of linearized ownship trajectories before calling GCE for rigorous re-evaluation of candidate resolutions. It relies on user-defined cost functions for selecting the best maneuver. This promotes high computational efficiency without sacrificing flexibility or quality of proposed resolutions.

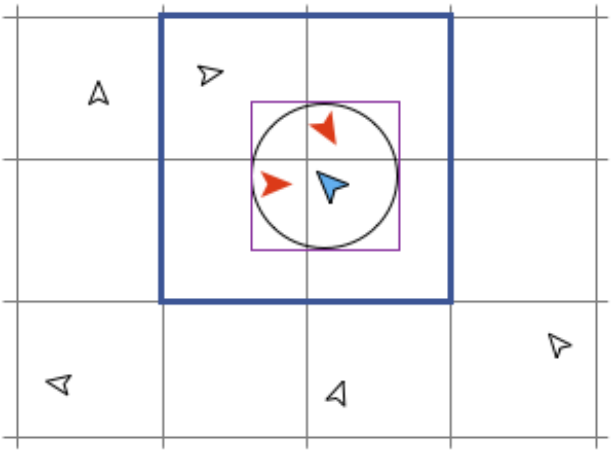

Figure 1. Grid-based threat detection

III.B.1. GRA Algorithm at a Glance

GRA obtains a resolution by trying to find a simple Standard Maneuver that would result in a conflict-free solution.

The following six Standard Maneuvers (in the remainder of the paper we refer to them simply as "maneuvers") can be used:

1. Turn Right;
2. Turn Left;
3. Vertical Up: increased vertical speed / flight path angle (faster climb or slower descent);
4. Vertical Down: reduced vertical speed / flight path angle (slower climb or faster descent);
5. Speed Down (decelerate);
6. Speed Up (accelerate).

Finding the best maneuver starts from a local transformation of the CPA, for the highest priority threat, into a Trajectory Change Point (TCP) such that it satisfies the following conditions:

1. it should not violate any separation criteria;
2. it should constitute a CPA for the transformed trajectory.

The first condition is self-explanatory. The second condition is needed to ensure that after passing the TCP an aircraft could return to its initial route without the risk of encountering the same threat. This is important to improve the stability of proposed resolutions and to help pilots determine when they can safely execute a "recapture" maneuver returning to mission flight plan. Effectively, the TCP defines the end of maneuver in GRA.

GRA starts by finding a linear trajectory calculated by extrapolating the state at CPA backwards to the desired start time (see section III.B.2 for details). Each of maneuvers listed above is applied in user-defined increments (steps), such as 5 degrees for heading change and 1 degree for change in flight path angle. At every step:

1. a new TCP is obtained by perturbing the linearized trajectory in the direction dictated by the applied maneuver and analytically estimating the CPA along the new trajectory;
2. the candidate TCP is evaluated to determine whether it is "locally conflict-free" (i.e. does not violate separation criteria);
3. if the answer is "yes", a candidate solution is re-evaluated by calling GCE, which checks for conflicts with all intruders in grid map to verify that the solution is "globally conflict-free" within the specified lookahead time; this involves generating a new ownship trajectory with the TCP added as a new constraint;
4. if re-evaluation did not confirm that the solution is "globally conflict-free", these operations are repeated, in the next step, for the same maneuver until the predefined limit in change of control variable (heading, flight path angle, or speed) is reached.

If the new solution is "better" than the previous "best" candidate, it replaces it as the new "best".

Note that first two operations use fast analytic calculations. The third operation takes advantage of GCE optimization provided by grid map, but still requires computationally intensive calls to Trajectory Predictor for generating a new ownship trajectory. However these calls are made typically only once for each maneuver rather than at every step, because in most cases reevaluation confirms that solution is conflict-free and immediately terminates iterations. This helps greatly reduce computation time.

This sequence of operations is shown in Fig. 2, which is further clarified in the remainder of this section.

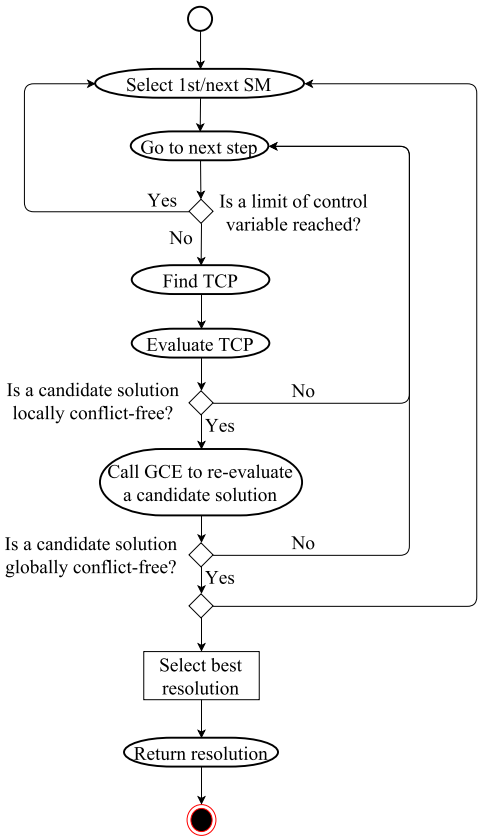

Figure 2. Generic Resolution Advisor algorithm

III.B.2. Fast Linearized Analytic Solution for Finding TCP

If a CPA for the original ownship trajectory (black solid curve in Fig. 3) and a predicted intruder trajectory (red curve) is known from the output of GCE, then the CPA for any perturbed ownship trajectory (blue curve) can be quickly estimated from an analytical solution based on linearization of predicted ownship and intruder trajectories near the CPA as illustrated in Fig. 3.

As a first step, the states of ownship and intruder at CPA (shown as arrows) are extrapolated backward to a maneuver start time that may include an anticipated delay in maneuver execution (referred to herein as "algorithm delay"), as shown by dashed black and red lines, respectively. This approximation is justified because the accurate representation of trajectory around a CPA is most important for evaluation of candidate resolutions. It also simplifies modeling of manuevers by applying them to the extrapolated ownship state rather than to its actual initial state. Therefore, the perturbed ownship trajectory is approximated by a straight line, shown as a dashed blue line in Fig. 3. Then, a candidate TCP can be defined as the CPA for the perturbed trajectory, shown as an arrow on the blue line. Note that this approach is based on the predicted trajectories from GCE, therefore it does not require any assumptions regarding the availability of intent data for ownship and intruders.

The CPA is found from the time to minimal distance between ownship and intruder positions on their linearized trajectories t_{CPA}, assuming that they are moving with constant speeds (see Ref. 8)

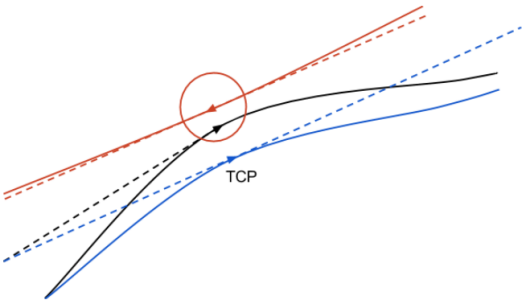

Figure 3. Linearization of the ownship and intruder trajectories near TCP

$$t_{CPA} = \max\left\{0, -\frac{\sum_k \Delta x_k \Delta v_k}{\sum_k (\Delta v_k)^2)}\right\} \tag{1}$$

where k represents a cartesian component and the summation occurs over all three components, with

$\Delta x_k = (x_I - x_O)_k$, $\Delta v_k = (v_I - v_O)_k$ - intruder's coordinates and velocities relative to ownship,

$(x_O)_k$, $(x_I)_k$ - initial ownship's and intruder's positions,

$(v_O)_k$, $(v_I)_k$ - ownship's and intruder's velocities.

Although the Eq. 1 for time-to-CPA is the same as in Ref. 40, it uses positions and velocities from the states extrapolated backward from CPA for original trajectories predicted by GCE and not from the observed initial states. This ensures more accurate TCP if a CPA for perturbed trajectory does not deviate too much from the original CPA.

III.B.3. Fast CPA Evaluation Using State-based Threat Detector

To determine whether a candidate solution is "locally conflict-free", the horizontal and vertical separations at CPA are checked against their threshold values for corresponding separation standard.

These separations can be easily found from ownship and intruder positions at CPA given by:

$$(x_O)_k + (v_O)_k \cdot t_{CPA} \tag{2}$$

$$(x_I)_k + (v_I)_k \cdot t_{CPA} \tag{3}$$

III.B.4. Re-evaluation of Solutions

Once a fast check has determined that a solution is "locally conflict-free," quick evaluation is repeated at the next step to account for possible uncertainty. If the solution is still conflict-free or a limit of control variable (e.g. heading or flight path angle) is reached, the solution is re-evaluated using a trajectory predictor of sufficient fidelity to account for aircraft performance and to check for conflicts with other intruders. This step involves recalculation of the ownship's predicted trajectory starting with actual rather than extrapolated initial state and using constraints modified by adding the new TCP. This trajectory then is rigorously evaluated for conflicts with all intruders as described in section III.A. Note that intruder trajectories do not need to be re-calculated since they are not perturbed.

III.B.5. Limits of Control Variables

GRA provides configurable "operational limits" of control variables for maneuvers. For instance, users can limit the maximum heading change for turns by 90 degrees, and the maximum change in flight path angle for vertical maneuvers by 5 degrees.

In addition to operational limits, GRA estimates "dynamic limits" of control variables, based on aircraft performance.

GRA stops incrementing the control variable for each maneuver if it reaches either its dynamic or operational limit.

III.B.6. Selection of Resolution

GRA can run in two user configurable "control modes". In the so called "UseFirst" control mode, GRA returns the first conflict-free solution it finds. In this case GRA works faster, guarantees that it will find a conflict-free solution if it exists, and provides solutions that are almost always stable but not necessarily optimal.

On the other hand, if a conflict-free resolution does not exist, or if GRA is configured for the so called "UseBest" control mode, GRA will find the best resolution using the following rules:

1. if the best solution was not defined yet, the first candidate solution becomes the best solution;
2. if a candidate solution is conflict-free and the old best solution is not conflict-free, the candidate solution becomes the new best solution;
3. if a candidate solution and the old best solution are both conflict-free or both not conflict-free, and the cost of candidate solution is lower than the cost of the old best solution, the candidate solution becomes the new best solution;
4. in all other cases the best solution does not change.

Note that, even though a TCP estimated from linearized trajectories is approximate, a candidate resolution is always based on the predicted ownship trajectory, generated using all known information about the ownship flight constraints (including the TCP) and aircraft performance.

GRA can use any externally defined cost function to select the best resolution maneuver. The cost function can rank maneuver types by an order of preference (e.g. to support right of way rules) and penalize or suppress specific maneuver types, too aggressive maneuvers, too frequent maneuver type changes within an encounter, and maneuvers that would result in too small separation at predicted CPA.[8]

IV. Evaluation of GRACE Performance

The most obvious potential uses of GRACE are as a fast conflict detection algorithm (GCE), as an automatic guidance algorithm for UAS (GRA), or as a model of such guidance in simulations of UAS.

GRACE provided the core algorithms for JADEM's alerting and guidance tools that have been evaluated and refined in a number of HITL studies and flight tests.[8] In all these studies GRACE was used to provide alerting to pilots. GRACE was also used to provide directive guidance in earlier studies and was used to compute bands and recovery guidance for the most recent studies.

GRACE also has been used for modeling UAS DAA systems in a number of parametric/factorial studies[41,42] and NAS-wide simulations.[13,43,44]

This section describes the methodology and sample results of such studies obtained using two approaches,

1. *Parametric*: uses 180 encounter geometries to test the performance of the algorithm under demanding conditions regardless of likelihood of occurrence in real life.
2. *NAS-wide*: uses projected UAS mission profiles developed under prior work[45] and recorded VFR traffic data[46] for a more realistic encounter model.

Three different scenario configurations were used in the parametric evaluation, namely,

1. *Baseline*: uses "perfect" surveillance data without simulated sensor errors and executes GRACE resolution maneuvers immediately; the mission profile is recaptured at GRACE recommended times following each maneuver;
2. *Noisy Surveillance*: similar to the baseline case but uses airborne radar model paired with a tracker;[8] this approximates a real-world fully automated concept;
3. *Pilot Delay*: similar to the baseline case but uses a model of pilot delayed action; this essentially represents a pilot-in-the-loop concept of operation.

In contrast, a single configuration was used for the NAS-wide case with "perfect" surveillance data, automatic execution of resolution maneuvers, and mission recapture with no delays.

The remainder of this section provides a summary of the metrics used in evaluations of GRACE performance, followed by results for the parametric and then for the NAS-wide evaluation.

IV.A. Performance Metrics

JADEM fast-time simulator calculates a number of metrics that can be used to evaluate system performance. The metrics most relevant to GRACE are listed below.

Conflicts	number of conflicts (encounters)
Resolutions	total number of resolutions proposed by GRE
Changes Per Encounter	number of changes in resolution type, such as a right turn followed by a left turn or by a vertical maneuver, divided by the number of encounters (conflicts)
Predicted Violations	number of times when GRE predicted a violation for any separation standard within a specified look-ahead time
Actual Violations	number of times when GRE detected a violation; this indicates that an avoidance maneuver failed to resolve a conflict
Failure Rate	a ratio of number of failures to resolve a conflict to number of detected conflicts in %
Predicted NMAC	number of times when GRE predicted a NMAC within a specified look-ahead time
Actual NMAC	number of times when GRE detected an NMAC; this indicates a failure of DAA system to prevent an imminent NMAC
$\mathbf{S}_{\mathrm{NMAC}}$	$S_{\mathrm{NMAC}} = \max(\frac{R_{\mathrm{NMAC}}}{R_{\mathrm{CPA}}}, \frac{Z_{\mathrm{NMAC}}}{Z_{\mathrm{CPA}}}) \times 100$, where S is a simple measure of severity, R is the range, and Z is the vertical separation. S_{NMAC} exceeds 100% in the case of an **Actual NMAC**, and lower values of S_{NMAC} correspond to larger separations at CPA relative to the NMAC zone.

IV.B. Parametric Evaluation of GRACE Performance

GRACE performance was evaluated for perfect and noisy non-cooperative sensors with $8nmi$ detection range.[8] A theoretical omnidirectional sensor was used to study the effect of pilot delay on the performance of GRACE, while onboard radar and tracker models were used to evaluate the effect of noise.

All test cases in this sub-section were generated for a 50-minute ownship flight following a multi-turn mission plan typically used in the HITL simulations described in Ref. 8.

Intruders cross the ownship's trajectory at five different points, which may occur before, after, or within turns. Encounters are created at each of these points for four headings, three ground speeds, and three vertical speeds (level, climb, descent) for a total of 180 encounters, all of which are designed to result in NMAC. Each of these encounters was processed separately and independently from others, and a summary of statistics for all encounters was generated. This is equivalent to repeating the same ownship flight for 180 different intruders.

GRACE was configured to provide guidance for preventing violation (loss of DAA Well Clear) with $0.66nmi$ miss distance threshold, $450ft$ vertical separation threshold, and $35sec$ modified tau threshold with tau computed using $0.66nmi$ distance modifier.[7,8] If loss of Well Clear could not be prevented, GRACE works to avoid NMAC and maximize separation at CPA. In all these tests GRACE is called every second with two minutes prediction horizon (look-ahead time), which is typical for DAA applications.

Table 1 summarizes the results for three sensor models, namely,

1. Omnidirectional: a theoretical perfect sensor that can detect all intruders within a geometric cylinder with a range of $8\,nmi$, a height of $\pm 5000\,ft$;
2. Directional Perfect: a tracker and sensor model of airborne radar,[8] using truth ownship and intruder states without sensor noise and navigation errors;
3. Directional Noisy: the same directional tracker and sensor model of airborne radar, which uses perturbed ownship and intruder states with added sensor noise and navigation errors.

With unlimited look-ahead time and sensor Field-of-Regard, any algorithm can be reasonably expected to avoid all violations of DWC. Table 1 demonstrates that with an $8nmi$ sensor range and a $2min$ look-ahead time, GRACE was still able to prevent most violations. Moreover, GRACE avoided NMAC in all cases when Well Clear violations could not be prevented. Note that, in the table, the number of predicted violations

exceeds the number of encounters for two reasons: first, return to mission plan may generate a secondary conflict, and second, for the directional cases, multiple resolutions may be commanded for each encounter (this is particularly evident for the noisy case).

Table 1. GRACE performance in simulated encounters using different sensors

Sensor	Predicted Violations	Predicted NMAC	Changes per Encounter	Actual Violations	Failure Rate (%)	S_{NMAC} (%)	
						Mean	Max
Omnidirectional	209	174	0.056	1	0.6	15	15
Directional perfect	246	149	0.006	9	5.1	34.3	39.5
Directional noisy	380	4	1.11	13	7.3	31.8	46.8

In the omnidirectional sensor case, only one conflict was not resolved. Analysis of this conflict showed that it was an artifact of JADEM recapture logic combined with a mismatch in handling turns. JADEM's flight simulator always executes turns in flyover mode (i.e. after crossing a waypoint), whereas GRACE was configured in this study to use flyby mode. This discrepancy induced a secondary conflict on recapture after the primary conflict was successfully resolved. In essence, this is a problem that needs to be addressed in JADEM.

The failure rate appears significantly higher in the case of directional perfect and noisy sensors. Closer examination revealed that all these failures could be attributed to late surveillance detections with intruders being detected when they already violated Well Clear. Moreover, in all these cases the detected trajectories were already diverging after they passed CPA, so no maneuvers were needed to improve the situation. These late detections are an artifact of sensor performance, which was not a focus of this study. For both perfect and noisy sensors GRACE was able to resolve all conflicts that were not detected too late. The number of changed resolutions per encounter for perfect sensors was very low (less than 0.06). For noisy sensors the number of changed resolutions per encounter increased to 1.1, which is attributed to large vertical errors of non-cooperative radar sensor.

For concepts that require a pilot to evaluate and execute DAA guidance, a delay is incurred before the maneuver is finally flown by the UAS control system. DAA algorithms should ensure that recommended maneuvers remain valid at that time. GRACE allows for delayed response by introducing a delay parameter (the aforementioned algorithm delay) and computing guidance starting at a point on ownship's trajectory corresponding to this delay (keeping the intervening segment "frozen" as it were; see section III.B.2).

It should be noted, however, that pilot response delays are not known in advance and vary from one pilot to another and from one encounter to another. Therefore, it is important to know how sensitive GRACE performance is to a mismatch between algorithm delay and actual response time.

Table 2 illustrates the effect of algorithm delays when the total pilot response time is 10 seconds. These delays are chosen based on results of previous HITL studies. For all algorithm delays varying between 5 and 15 seconds GRACE resolutions avoided all NMACs and prevented almost all Well Clear violations.

Table 2. GRACE performance with pilots in the loop for 10-second total response time

Algorithm Delay	Predicted Violations	Predicted NMAC	Changes per Encounter	Actual Violations	Failure Rate (%)	S_{NMAC} (%)	
						Mean	Max
5 seconds	2349	1849	0.73	3	1.7	13.4	13.8
10 seconds	2346	1849	0.64	0	0	–	–
15 seconds	2316	1849	0.60	1	0.6	15.6	15.6

Table 3 illustrates the effect of pilot response time given a conservative 15-second algorithm delay. For all pilot delays GRACE ensured that all NMACs could be avoided with comfortable safety margin (S_{NMAC} below 20%). The number of encounters that resulted in Well Clear violations increased with pilot delays as expected, but remained below 4% of total number of encounters even for total pilot delay as high as 20 seconds. The number of changed resolutions per encounter was below 1.3 for all combinations of algorithm and pilot delays.

These results clearly indicate that GRACE remains robust even when pilot delays are large and differ from delays anticipated by the algorithm.

Table 3. GRACE performance with pilots in the loop for 15-second algorithm delay

Total Pilot Delay	Predicted Violations	Predicted NMAC	Changes per Encounter	Actual Violations	Failure Rate (%)	S_{NMAC} (%)	
						Mean	Max
10 seconds	2316	1849	0.64	1	0.6	15.6	15.6
15 seconds	3419	2636	1.00	6	3.3	8	12.5
20 seconds	4686	3463	1.27	7	3.9	9.9	14.7

IV.C. Effectiveness in NAS-wide Mitigated Studies

The effectiveness of GRACE in mitigating collision threats was validated in a 24-hour NAS-wide fast-time study of simulated UAS traffic and recorded radar Visual Flight Rules (VFR) traffic. Instrument Flight Rules (IFR) traffic was not included, since separation of IFR traffic will presumably be maintained by Air Traffic Control. Eighteen UAS mission profiles developed under prior work were used in this study to simulate a variety of UAS aircraft conducting an assortment of possible future UAS missions, including: point-to-point transport, regional mapping/monitoring, and patrol.[13,45] The 24-hour scenario included more than 17,000 UAS flights with about 10,000 total flight hours, mostly in transitional class E airspace in proximity to VFR traffic.[45] It was assumed that all unmanned aircraft were equipped with cooperative ADS-B and MODE-C sensors and a non-cooperative directional sensor with 8-nmi range typical for onboard radars. VFR traffic was modeled as a mix of flights with and without cooperative sensors. The study leveraged JADEM's NAS-wide simulation capability and used a kinematic performance model for flying UAS missions (honoring commanded resolutions) and for evaluating resolution candidates in GRACE. UAS-to-UAS encounters were not considered in this study, and algorithm performance was evaluated in aggregate over all airspaces. Note that, while the study used perfect surveillance sensors, the VFR data is itself inherently noisy.

The study compares two simulations: "unmitigated" and "mitigated." The unmitigated simulation, for which GRACE resolutions are neither commanded nor executed, represents the baseline scenario for UAS without active DAA systems. In contrast, the mitigated simulation commands and executes GRACE resolutions. Moreover, JADEM's flight simulator "recaptures" the UAS nominal flight plan after successful avoidance maneuvers. This recapture can in turn result in secondary conflicts.

This study did not model communication failures, latency, or delayed pilot responses. Work to include more realistic models of pilot behavior with random delays is currently underway.

Table 4 compares the results of the two simulations and shows that mitigation reduced the number of predicted violations by a factor of eight and the number of actual violations by a factor of five. GRACE failed to prevent actual violations for only 2.5% of conflicts with predicted violations. More importantly, mitigation reduced the number of predicted NMAC events by a factor of 33, and eliminated all actual NMACs. The frequency of changed resolutions averaged at the level of 0.4 per encounter.

Changes in resolution types and failures to resolve conflicts can both be attributed to late detections, with intruders first detected when they are too close to ownship. The situation is further complicated by unknown intruder intent, with GRACE having to rely on extrapolated (dead-reckoned) intruder trajectories.

Table 4. Effect of mitigation on a full-day simulated UAS traffic over the NAS

Simulation	Predicted Violations	Actual Violations	Predicted NMAC	Actual NMAC	S_{NMAC} (%)	
					Mean	Max
Unmitigated	115409	1894	2220	44	29.4	546.3
Mitigated	14394	359	68	0	20.2	92.2

These results show that GRACE, used as an automatic guidance algorithm, was able to prevent almost all violations of separation. In cases where this was not possible because of late detection or unexpected intruder maneuver, GRACE was still able to avoid the NMAC.

V. Conclusions

Safe integration of UAS into the NAS requires development and validation of DAA systems as a means to comply with the FAA-mandated "see-and-avoid" requirement for human pilots. Despite the diversity of

algorithms that could potentially provide DAA functions, NASA recognized the need for a fast and flexible "generic" alerting and resolution algorithm that could help reduce the complexity of DAA system modeling. To fill that need, the Generic Resolution Advisor and Conflict Evaluator (GRACE) was implemented in NASA's Java Architecture for DAA Extensibility and Modeling to provide alerting, directive guidance, trial planning capabilities, bands, and DWC recovery guidance.

GRACE was designed without any assumptions regarding performance capabilities of an aircraft or its sensor and communication systems. This made it suitable for various applications and DAA guidance concepts. In particular, GRACE can be used in prototyping various decision support tools for ground pilots. Furthermore, GRACE makes no assumptions about degree of autonomy. This allows it to be used in fully autonomous UAS DAA operations, in remotely piloted unmanned aircraft, and potentially even in manned flights.

The new algorithm was validated and used in a number of human-in-the-loop experiments, in flight tests with live aircraft, in parametric studies with diverse encounter geometries, and in full-day NAS-wide simulations. All these tests demonstrated the ability of GRACE to reduce the frequency and severity of Losses of Well Clear and to prevent NMAC in interactions with other aircraft flying by VFR.

Current and future work includes using GRACE to improve robustness of DAA resolutions, taking into account uncertainties of intruder positions and intent.

Acknowledgments

The authors want to express gratitude to Eric Mueller and Heinz Erzberger for fruitful discussions and encouragement, Gilbert Wu for his work on integrating the sensor and tracker models and for contributing a description to said models, and the whole DAA team at NASA and our partners from Honeywell for all their help and support.

References

[1]Code of Federal Regulation, General Operating and Flight Rules, title 14, sec. 91.113.

[2]"Sense and Avoid (SAA) for Unmanned Aircraft Systems (UAS)," FAA Workshop, Oct. 2009.

[3]Angelov, P., *Sense and Avoid in UAS. Research and Applications*, Wiley, 2012, p. 38.

[4]Desilles, A., Zidani, H., and Cruck, E., "Collision analysis for an UAV," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Sept. 2012.

[5]Zeitlin, A., *Progress on Requirements and Standards for Sense & Avoid*, The MITRE Corporation, Aug. 2010.

[6]RTCA DO-185B "MOPS for TCAS II," Version 7.1, Volume I, June 19, 2008.

[7]Cook, S., Brooks, D., Cole, R., Hackenburg, D., and Raska, V., "Defining Well Clear for Unmanned Aircraft Systems," Paper presented to the American Institute of Aeronautics and Astronautics (AIAA 2015-0481).

[8]Abramson, M., Refai, M., Santiago, C., "The Generic Resolution Advisor and Conflict Evaluator (GRACE) for Unmanned Aircraft Detect-And-Avoid Systems," 2017, (In Press), NASA/TM-2017-219507.

[9]Rorie, R.C. and Fern, L., "UAS measured response: The effect of GCS control mode interfaces on pilot ability to comply with atc clearances," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 2014, Vol. 58, No. 1, pp. 64-68.

[10]Mueller, E.R., Isaacson, D., and Stevens, D., "Air Traffic Controller Acceptability of Unmanned Aircraft System Detect and Avoid Thresholds," 2015, NASA TM-2015-219392.

[11]Santiago, C., and Mueller, R., "Pilot Evaluation of a UAS Detect-and-Avoid System's Effectiveness in Remaining Well Clear," *10th USA/Europe ATM R & D Seminar (ATM2015)*, Lisbon, Portugal, 2015.

[12]Gong, C., Minghong G. Wu, and Santiago, C., "UAS Integration in the NAS Project: Flight Test 3 Data Analysis of JADEM-Autoresolver Detect and Avoid System," 2012, NASA/TM-2012-216051.

[13]Lee, S. M., Park, C., Thipphavong, D. P., Isaacson, D. R., and Santiago, C., "Evaluating Alerting and Guidance Performance of a UAS Detect-And-Avoid System," *NASA-TM-2016-219067*, Feb. 2016.

[14]Monk, K.J. and Roberts, Z., "UAS Pilot Evaluations of Suggestive Guidance on Detect-and-Avoid Displays," *Proceedings of the Human Factors and Ergonomics Society 60th Annual Meeting*, Washington, DC, 2016.

[15]Mueller, E. R., Santiago, C., and Watza, S., "Piloted Well Clear Performance Evaluation of Detect-and-Avoid Systems with Suggestive Guidance," 2016, NASA TM-2016-219396.

[16]Murphy, J., Hayes, P., Kim, S., Bridges, W., and Marston, M., "Flight Test Overview for UAS Integration in the NAS Project," *AIAA Atmospheric Flight Mechanics Conference (AIAA-2016-1756)*, San Diego, CA, 2016.

[17]Monk, K.J. and Roberts, Z., "Maintain and Regain Well Clear: Maneuver Guidance Designs for Pilots Performing the Detect-and-Avoid Task," *Proceedings of the International Conference on Applied Human Factors and Ergonomics*, Los Angeles, CA (In Press), 2016.

[18]Rorie, R.C., Fern, L., Monk, K., Santiago, C., Shively, R.J., and Roberts, Z., "Validation of minimum display requirements for a UAS detect and avoid system," *AIAA Aviation Conference*, Denver, CO (In Press), 2017

[19]Kuchar, J. K. and Yang, L. C. "A Review of Conflict Detection and Resolution Modeling Methods." *IEEE Transactions on Intelligent Transportation Systems*, 1(4), 2000.

[20]Harman, W. H., "TCAS: A System for Preventing Midair Collisions," *Lincoln Laboratory Journal*, 1989, vol. 2, no. 3.

[21]Albaker, B. M. and Rahim, N. A., "Unmanned Aircraft Collision Avoidance System Using Cooperative Agent-Based Negotiation Approach," *Int. J. Simulation, Syst. Sci. Technol.*, April 2010.

[22]Hill, J. C., Johnson, F. R., Archibald, J. K., Frost, R. L., and Stirling, W. C., "A Cooperative Multi-agent Approach to Free Flight," *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multi Agent Systems*, July 2005.

[23]Vanek, B., Peni, T., Zarandy, A., Bokor, J., Zsedrovits, T., and Roska, T., "Performance Characteristics of a Complete Vision Only Sense and Avoid System," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Sept. 2012.

[24]Chen, W.-Z., Wong, L., Kay, J., , and Raska, V. M., "Autonomous Sense and Avoid (SAA) for Unmanned Air Systems (UAS)," MP-SCI-202-28.doc available at ftp://ftp.rta.nato.int/PubFullText/RTO/MP/RTO-MP-SCI-202/, 2011.

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

[26]Isaacson, D. and Erzberger, H. "Design of a Conflict Detection Algorithm for the Center/TRACON Automation System," *Proc. 16th Digital Avionics Systems Conf.*, Irvine, CA, 1997,

[27]Erzberger, H., "The Automated Airspace Concept," *Proceedings of the Fourth USA/Europe Air Traffic Management R & D Seminar*, Dec. 2001.

[28]Kelly, W. and Eby, M., "Advances in Force Field Conflict Resolution Algorithms," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Aug. 2000.

[29]Hu, J., Prandini, M., Nilim, A., and Sastry, S., "Optimal Coordinated Maneuvers for Three-Dimensional Aircraft Conflict Resolution," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Aug. 2001.

[30]Kosecka, J., Tomlin, C., Pappas, G., and Sastry, S. "Generation of Conflict Resolution Maneuvers for Air Traffic Management," *1997 International Conf. on Robotics and Intelligent Systems*, Sept. 1997.

[31]Dougui, N., Delahaye, D., Peuchmorel, S., and Mongeau, M. "A light propagation model for aircraft trajectory planning," *Journal of Global Optimization*, 56(3), 2013.

[32]Roussos, G., Chaloulos, G., Kyriakopoulos, K., Lygeros, J. "Control of multiple non-holonomic air vehicles under wind uncertainty using model predictive control and decentralized navigation functions," *IEEE Conference on Decision and Control. IEEE*, 2008

[33]Roussos, G., Kyriakopoulos, K. "Towards constant velocity navigation and collision avoidance for autonomous non-holonomic aircraft-like vehicles." *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. Proceedings of the 48th IEEE Conference*, Shanghai, 2009

[34]Rodionova O., Delahaye D., Sridhar B., Ng H. K. "Deconflicting Wind-Optimal Aircraft Trajectories in North Atlantic Oceanic Airspace." *Advanced Aircraft Efficiency in a Global Transport System (AEGATS '16)*, Paris, France, 2016

[35]Delahaye, D., Peyronne, C., Mongeau, M., and Peuchmorel, S. "Aircraft conflict resolution by genetic algorithm and B-spline approximation." *2nd ENRI International Workshop on ATM/CNS*, 2011

[36]Sawhill, B., "Modeling Systemic Phenomena in the National Airspace System," *NASA Aeronautics Airspace Systems Program Technical Interchange Meeting*, March 2011.

[37]Von Viebahn, H. and Schiefele, J., "A Method for Detecting and Avoiding Flight Hazards," *Proceedings of the SPIE Meeting on Enhanced Synthetic Vision*, April 1997.

[38]Watkins, O. and Lygeros, J., "Stochastic Reachability for Discrete Time Systems: An Application to Aircraft Collision Avoidance," *IEEE Conference on Decision and Control*, 2003.

[39]Muñoz, C., Narkawicz, A., Hagen, G., Upchurch, J., Dutle, A., Consiglio, M., and Chamberlain, J., "DAIDALUS: Detect and Avoid Alerting Logic for Unmanned Systems," *Proceedings of the 34th Digital Avionics Systems Conference (DASC 2015)*, Prague, Czech Republic, 2015

[40]Muñoz, C., Narkawicz, A., and Chamberlain, J., "A TCAS-II Resolution Advisory Detection Algorithm," *AIAA Guidance, Navigation, and Control Conference*, 2013.

[41]Cone, A., Thipphavong, D.P., Lee, S.M., and Santiago, C., "UAS Well Clear Recovery against Non-Cooperative Intruders using Vertical Maneuvers," *17th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference , (In Press), 2017*

[42]*Thipphavong, D. and Cone, A.,"Ensuring Interoperability between UAS Detect-and-Avoid and Manned Aircraft Collision Avoidance,"* 12th USA/Europe ATM R & D Seminar (ATM2017)*, (In Press), 2017*

[43]*Johnson, M., Mueller, E., and Santiago, C., "Characteristics of a Well Clear Definition and Alerting Criteria for Encounters between UAS and Manned Aircraft in Class E Airspace,"* 10th USA/Europe ATM R & D Seminar (ATM2015)*, Lisbon, Portugal, 23-26 June 2015.*

[44]*Thipphavong, D.P., Johnson, M.A., Refai, M.S., and Snow, J.W., "Downstream Effects of Separation Assurance on Encounters between UAS and Manned Aircraft,"* Journal of Air Transportation (AIAA)*, (In Press), 2017*

[45]*Ayyalasomayajula, S., Wieland, F., Trani, A., and Hinze, N., "Unmanned Aircraft System Demand Generation and Airspace Performance Impact Prediction,"* Proceedings of the 32nd IEEE Digital Avionics Systems Conference, *IEEE. Syracuse, NY, October 2013.*

[46]*Park, C., Lee, H., and Musaffar, B., "Radar Data Tracking Using Minimum Spanning Tree-Based Clustering Algorithm,"* 11th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference (AIAA-2011-6825), Virginia Beach, VA, Sept. 2011.