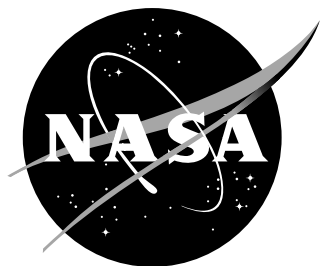


NASA/TM-2017-219507



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

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

*Mohamad Refai  
Crown Consulting Inc, Moffett Field, California*

*Confesor Santiago  
NASA Ames Research Center, Moffett Field, California*

---

May 2017

## NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI Program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI Program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Report Server, thus providing one of the largest collection of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- **TECHNICAL PUBLICATION.** Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.
- **TECHNICAL MEMORANDUM.** Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- **CONTRACTOR REPORT.** Scientific and technical findings by NASA-sponsored contractors and grantees.

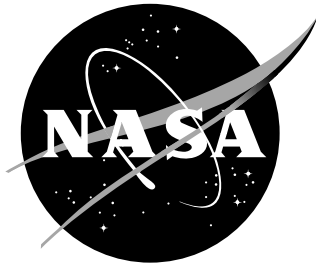
- **CONFERENCE PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.
- **SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- **TECHNICAL TRANSLATION.** English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include creating custom thesauri, building customized databases, and organizing and publishing research results.

For more information about the NASA STI Program, see the following:

- Access the NASA STI program home page at [\*http://www.sti.nasa.gov\*](http://www.sti.nasa.gov)
- E-mail your question via the Internet to [\*help@sti.nasa.gov\*](mailto:help@sti.nasa.gov)
- Fax your question to the NASA STI Help Desk at 443-757-5803
- Phone the NASA STI Help Desk at 443-757-5802
- Write to:  
NASA STI Help Desk  
NASA Center for AeroSpace Information  
7115 Standard Drive  
Hanover, MD 21076-1320

NASA/TM-2017-219507



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

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

*Mohamad Refai  
Crown Consulting Inc, Moffett Field, California*

*Confesor Santiago  
NASA Ames Research Center, Moffett Field, California*

National Aeronautics and  
Space Administration

Ames Research Center  
Moffett Field, California 94035-1000

---

May 2017

The use of trademarks or names of manufacturers in this report is for accurate reporting and does not constitute an official endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.

Available from:

NASA Center for AeroSpace Information  
7115 Standard Drive  
Hanover, MD 21076-1320  
443-757-5802

## Abstract

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” since it was designed without any assumptions regarding temporal or spatial scales, aircraft performance, or its sensor and communication systems. Therefore, GRACE was adopted as a core component of the Java Architecture for Detect-And-Avoid (DAA) Extensibility and Modeling, developed by NASA as a research and modeling tool for Unmanned Aerial Systems Integration in the National Airspace System (NAS). GRACE has been used 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
ATC	Air-Traffic Control
BADA	Base of Aircraft Data
CPA	Closest Point to Approach
DAA	Detect-and-Avoid
DAIDALUS	Detect and Avoid Alerting Logic for Unmanned Systems
DGM	Dynamic Grid Map
DWC	DAA Well Clear
FMS	Flight Management System
GCE	Generic Conflict Evaluator
GCS	Ground Control Station
GRA	Generic Resolution Advisor
GRACE	Generic Resolution Advisor and Conflict Evaluator
HITL	Human-in-the-Loop
HMD	Horizontal Miss Distance
JADEM	Java Architecture for DAA Extensibility and Modeling
LoWC	Loss of Well Clear
MATG	Multimodal Adaptable Trajectory Generator
MOPS	Minimum Operational Performance Standards
NAS	National Airspace
NMAC	Near Mid-Air Collisions
SC-228	RTCA Special Committee 228 working to develop UAS MOPS
TCAS	Traffic Alert and Collision Avoidance System
TCAS RA	TCAS Resolution Advisories
TCP	Trajectory Change Point
TSAFE	Tactical Separation Assured Flight Environment
UA	Unmanned Aircraft
UAS	Unmanned Aerial Systems
VFR	Visual Flight Rules
WCR	Well Clear Recovery

# 1 Introduction

Detect-and-Avoid (DAA) capability is a key enabler for safe integration of Unmanned Aerial Systems (UAS) into the National Airspace System (NAS). One of the core challenges of such capability is to meet FAA requirements to “see and avoid” and when passing other aircraft to remain “well clear” as mandated for manned aircraft in 14 CFR §91.113 [1], “Rules of the Road”. As specified by the FAA both “see and avoid” and “well clear” implicitly rely on pilot judgement and visual acuity, hence they are inherently subjective. 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” (henceforth 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. Note that DWC does not replace controller required separation but rather supplements it.

To support DAA capability, the DAA system should provide the following functions [2, 5]:

1. *Detect*: Use one or more onboard sensors to detect traffic in the vicinity of the unmanned aircraft.
2. *Track*: Use (noisy) detection results to (optimally) estimate traffic positions and velocities.
3. *Evaluate*: Assess collision risk of tracked traffic.
4. *Prioritize*: Assess threat priorities (urgency levels).
5. *Declare*: Alert pilot to avoidance action required.
6. *Determine*: Decide what action to take.
7. *Command*: Communicate the action for execution.
8. *Execute*: Execute the commanded action.

Requirements for DAA systems are currently being developed by RTCA Special Committee 228 (SC-228) as a means for UAS to meet the aforementioned “see and avoid” rules. SC-228 has approved the draft DAA minimum operational performance standards (MOPS) in December 2016. At the time of this paper, the DAA MOPS is awaiting review and approval by the RTCA Program Management Committee. In the development of the MOPS, DAA systems were envisioned as a means to maintain a low level of risk of Near Mid-Air Collisions (NMAC), defined as a violation of 500 ft horizontal and 100 ft vertical separation threshold [6]. To accomplish this goal, SC-228 investigated a number of objective definitions of Well Clear [7]. It was therefore essential to quantify the levels of risk achieved by proposed DAA concepts and definitions of DWC. To that end, NASA 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. 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.

Integration of diverse DAA algorithms poses a significant challenge in DAA architectures. Furthermore, the different separation standards used and the complexity of interactions between them can introduce difficult and subtle interoperability issues and can make evaluation of a DAA architecture more problematic.

For these reasons, NASA recognized the need for a flexible “generic” alerting and resolution algo-

rithm that could help reduce the complexity of DAA system modeling. The subject of this paper is the Generic Resolution Advisor and Conflict Evaluator (GRACE), the algorithm that was developed to fill that need. In addition to its core alerting and resolution features, GRACE powers the recently introduced “OmniBands” and “Well Clear Recovery” (WCR) guidance concepts as implemented in JADEM (see Section 6, subsections 6.3 and 6.4).

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 [10–14]. 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. These simulations confirmed the ability of GRACE to support the alerting and guidance features proposed for future DAA systems. The goal of this paper is to describe GRACE with sufficient algorithmic detail to be of interest to UAS researchers and to developers of practical DAA systems for manufacture and deployment.

The paper is organized as follows. The next section provides an overview of previous work related to development of DAA algorithms. Section 3 briefly describes JADEM and its use in NASA research. GRACE features and algorithmic details are described in Sections 4, 4.1, and 4.2. Section 5 describes the cost functions used by GRACE. Section 6 highlights the use of GRACE in alerting and guidance models supported by JADEM DAA. Section 7 provides sample results from an evaluation of GRACE. Finally, concluding remarks are presented in Section 8.

## 2 Background

A number of algorithms that could in principle support conflict detection and avoidance capabilities of DAA have been proposed (see Ref. 4, 15 and the references therein). Several algorithms were inspired by the Traffic Alert and Collision Avoidance System (TCAS) [16] 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 [17, 18], hence said systems cannot be used with non-cooperative aircraft.

Some algorithms were specifically designed for certain types of onboard sensors, such as radars [17] or optical sensors [19]. Other algorithms, such as Jointly Optimal Collision Avoidance, are in principle sensor-agnostic [20], but tuned for specific Aircraft Response Model. Another next generation algorithm, the Airborne Collision Avoidance System (ACAS), proposed to replace TCAS [21], 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 SC-228. 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 (ATC), include the Profile Selector En Route (PFS-E) component of the Center / TRACON Automation System [22], and Autoresolver and TSAFE developed for NASA’s Automated Airspace Concept [23]. However for use in DAA, these algorithms would require non-trivial modifications to accommodate smaller look-ahead-times and spatial separation standards, introduction of time separation, lack of trajectory intent information, and more frequent update rates.

More recent algorithms, “inspired by nature,” include “force field” methods [24–26], the closely related “light propagation” model [27], 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” [28,29]. These algorithms demonstrated their effectiveness in guaranteed collision avoidance, but they can lead to overcosts and large deviations from nominal routes [31].

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 [30] or genetic algorithms [31]. These methods are efficient enough for strategic flight planning, but they are still not sufficiently fast for real-time applications with update rate on the order of seconds. Another drawback of these methods is that they model smooth trajectories suitable for future FMS abilities but 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 [24–26] and complexity theory [32] and leveraged computational efficiency of grid-based methods [33, 34]. GRACE provided the benefits of flexibility, robustness, and good computational performance, which made it more suitable for evolving requirements of research and modeling of future DAA systems. It was therefore adopted in 2013 for providing the core alerting and guidance functions of NASA’s Java Architecture for DAA Extensibility and Modeling (JADEM). GRACE also provided fast conflict detection logic for Autoresolver Adapted for DAA (Autoresolver-AD), which was integrated in JADEM a year later.

Note that concurrent to the development of JADEM, NASA was also developing the Detect and Avoid Alerting Logic for Unmanned Systems (DAIDALUS) [35]. DAIDALUS is the latest product of years of research on formal Verification and Validation methods for the safe integration of advanced air traffic concepts and algorithms in the National Airspace System. DAIDALUS was first released as open source in 2015. In contrast to GRACE, which was designed as a general purpose conflict detection and resolution algorithm, DAIDALUS was specifically tailored to DAA alerting and guidance requirements as defined in the draft MOPS, and selected by RTCA SC-228 as the reference implementation of aforementioned requirements. DAIDALUS is currently being integrated into JADEM as an alternate alerting and guidance module.

It should be noted that JADEM was expressly designed to host different DAA algorithms. This provides several important benefits:

- allows to choose an algorithm best suited to each experiment’s goals, requirements, and constraints;
- simplifies comparison between algorithms, helping identify and address their weaknesses;
- provides an extra level of verification, increasing the overall confidence in JADEM’s simulation results;
- helps clearly differentiate between common DAA performance characteristics and algorithm/implementation-specific behaviors.

The next section briefly describes JADEM and its components.



### 3 Java Architecture for DAA Extensibility and Modeling (JADEM)

JADEM was developed to enable various research activities in support of DAA MOPS development, verification, and validation; these activities include

1. NAS-wide closed-loop simulations;
2. parametric studies of DAA system performance;
3. HITL simulations;
4. live flight tests.

To that end JADEM was designed as a portable software system, composed of Detect-And-Avoid (DAA) module and additional supporting software (such as test drivers, validation logic, pilot models, tracking and sensor models, TCAS, and NAS-wide simulation capability). Figure 1 shows how DAA can be used in a Live, Virtual, and Constructive (LVC) architecture. DAA is discussed in more detail in the remainder of this section.

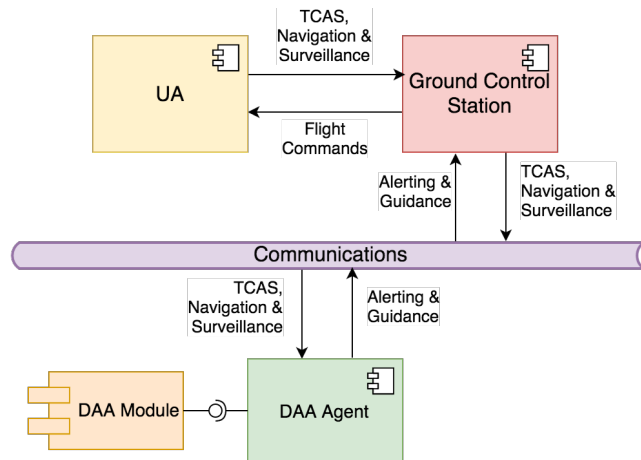


Figure 1. DAA use in LVC Architecture

In Fig. 1, the aircraft publishes TCAS, navigation, and surveillance data from its onboard systems and receives flight commands from the Ground Control Station (GCS). The GCS receives the data from onboard systems as well as DAA alerting and guidance, which the pilot uses to command flight maneuvers. The DAA Agent handles communication with the rest of the system and delegates DAA processing to JADEM's DAA module.

In addition to the models implemented in JADEM, the DAA module provides simplified perfect and noisy sensor models, an extended Kalman Filter with source selection tracker, GRACE, bands and WCR models, and supports integration of other resolution algorithms. The DAA module also supports trajectory prediction generically and JADEM provides a fast and robust kinematic implementation, the Multimodal Adaptable Trajectory Generator (MATG). MATG can handle any constraints (position, heading, speed, altitude, vertical speed, and time) in any combination for nominal and off-nominal conditions. It can be easily adapted for real aircraft behavior derived from track data, for aircraft performance models from the Base of Aircraft Data (BADA) (see

Ref. 36 and references therein), and for modeling unconventional aircraft not currently supported by BADA.

## 4 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. 15, GRACE can be characterized as:

Dimensions:	Horizontal and Vertical planes
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: Generic Conflict Evaluator (GCE) and Generic Resolution Advisor (GRA). GCE provides a customizable implementation of the *evaluate* function of DAA. GRA is a fast general purpose conflict resolver that recommends a resolution for use in the *determine* function of DAA.

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 cost function in GRA, and by reliance on an external trajectory predictor for modeling aircraft flight mechanics thus allowing users to select the fidelity level suitable for experimental needs. In particular, GRACE can be used for rotorcraft or other unconventional UAS.

A cost function used by GRA can be easily extended to include additional terms accounting for weather conditions, fuel consumption, or any other factors affecting the quality of resolutions.

### 4.1 Generic Conflict Evaluator (GCE)

GCE is an efficient deterministic algorithm for assessing intruder threats based on user-defined separation standards (described in the next section). It is a core module used in DAA *evaluation* and *guidance* functions as implemented in JADEM.

#### 4.1.1 Separation Standards

The following logical condition is typically used to indicate a threat for DAA systems

$$(HMD \leq HMD^*) \wedge (0 \leq \tau_{mod} \leq T_{mod}) \wedge (|\Delta z| \leq Z_{THR}) = true \quad (1)$$

Here *HMD* is the Horizontal Miss Distance defined as the minimum horizontal distance between the

ownship and intruder positions estimated by extrapolating along their respective headings.  $\tau_{mod}$  is “modified tau” defined by Eq. (3).  $|\Delta z|$  is the vertical separation between ownship and intruder states.  $HMD^*$ ,  $T_{mod}$ , and  $Z_{THR}$  are specified threshold values.

For a given set of thresholds, the left hand side of Eq. (1) defines what will henceforth be referred to as the separation standard; it is also used to define DWC. In this paper, the term “violation” (of separation) will be used, in a broad sense, to indicate that Eq. (1) holds for any given separation standard, and the more specific term “Loss of Well Clear” (LoWC) will be used to indicate a violation of the DWC.

Separation standards supported by GCE can include almost any combination of the following (along with corresponding thresholds and filtering conditions):

- horizontal separation;
- vertical separation;
- tau-separation;
- Horizontal Miss Distance (HMD);
- time to Closest Point of Approach (CPA);
- time to first loss (violation) of separation.

Tau-separation can be defined as “simple tau”

$$\tau = -\frac{r}{\dot{r}} \quad (2)$$

or “modified tau”

$$\tau_{mod} = \begin{cases} -(r_{xy}^2 - DMOD^2)/(r_{xy} \cdot \dot{r}_{xy}) & \text{if } \dot{r}_{xy} < 0 \text{ and } r_{xy} > DMOD \\ 0 & \text{if } r_{xy} \leq DMOD \\ \infty & \text{if } \dot{r}_{xy} \geq 0 \text{ and } r_{xy} > DMOD \end{cases} \quad (3)$$

Here  $r$  is the distance between the ownship and intruder positions known also as “slant range”,  $\dot{r}$  is the rate of change of  $r$ ,  $r_{xy}$  is the horizontal distance between the ownship and intruder positions,  $\dot{r}_{xy}$  is the rate of change of  $r_{xy}$ ,  $DMOD$  is the distance modification.  $DMOD$  values and  $\tau$  thresholds can be constant or altitude-dependent as defined, for example, by TCAS sensitivity levels [37]. Note that the last case in Eq. 3 corresponds to diverging encounter geometries and, per Eq. (1), does not indicate a violation.

In addition, a “vertical tau”, known also as “time to co-altitude”, can be defined:

$$\tau_z = -\frac{z_I - z_O}{\dot{z}_I - \dot{z}_O} \quad (4)$$

where  $z_O$ ,  $z_I$ ,  $\dot{z}_I$ , and  $\dot{z}_O$  are ownship and intruder altitudes and altitude change rates respectively.

GCE also supports the Air Force “acorn” Well Clear definition and several simplified Well Clear definitions based only on initial states of ownship and intruder or on extrapolated (dead-reckoned)

ownship trajectories as opposed to trajectories adhering to flight constraints [7]. These definitions can be useful for researching interoperability with different evaluation logics, such as the one used by TCAS [38].

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 ( $500 \times 100 \text{ ft}$ ).

#### 4.1.2 Dynamic Grid Mapping and Threat Detection

GCE can be used in two different roles:

1. in the *evaluate* function to identify the threats that may trigger the *determine* function; in this role it is called once for each call to the DAA system;
2. in the *determine* function 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 while straightforward, can be time-consuming. This needs to be done for arbitrary aircraft positions, which can deviate from the nominal route (based on intent) or 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 “Dynamic Grid Map” or DGM). DGM 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”. This can be seen as a deterministic variant of the method proposed in Ref. 33. Figure 2 illustrates the main idea of this method.

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 DGM cells (Fig. 2) using simple comparisons between the intruder coordinates and cell boundaries. Note that DGM is stored efficiently as a time sequence of hash maps, which are only populated with “occupied” cells.
3. Detecting threats from mapped intruders. This uses simple comparisons between cell boundaries and a bounding box with a side  $2\delta$  around the ownship’s position to identify potential intruders, where  $\delta$  is a horizontal distance deduced from the separation standards (Fig. 3). 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 performs the *prioritize* function by returning the first threat with the earliest conflict start time, which is defined as the time of the first state that violated the separation standard.

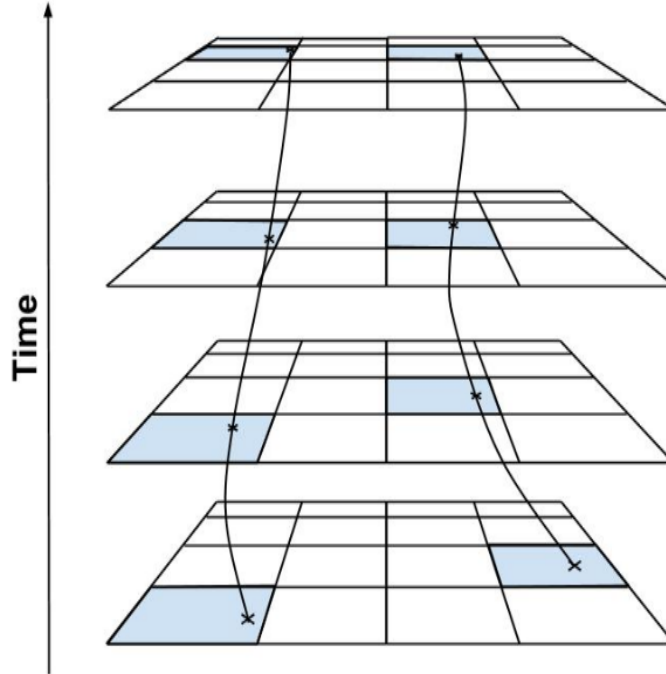


Figure 2. Dynamic grid mapping the intruder trajectories

Note that the first operation, which is expensive, is only done once, the resulting DGM subsequently being used in the *determine* function. This dramatically reduces DAA computation time.

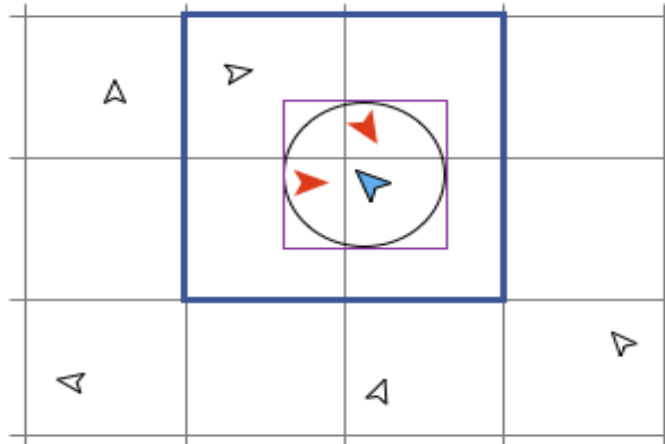


Figure 3. Grid-based threat detection

## 4.2 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 DGM. 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.

#### 4.2.1 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 (see Fig. 4):

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).

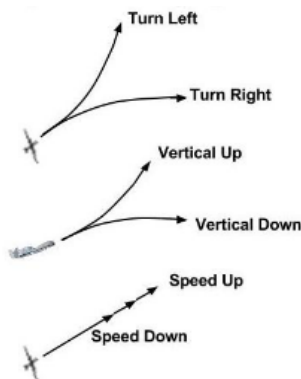


Figure 4. Standard Maneuvers in Generic Resolution Advisor

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 4.2.2 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 DGM 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 DGM, 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. 5, which is further clarified in the remainder of this section.

#### 4.2.2 Fast Linearized Analytic Solution for Finding TCP

If a CPA for the original ownship trajectory (black solid curve in Fig. 6) 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. 6.

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 maneuvers 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. 6. 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.

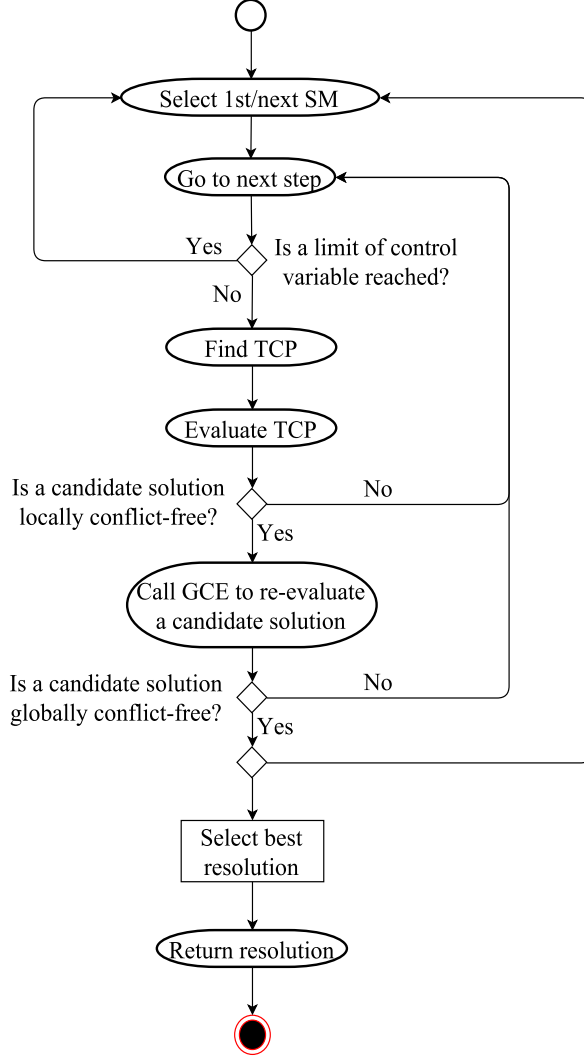


Figure 5. Generic Resolution Advisor algorithm

The CPA is found from the time to minimal distance between ownship and intruder positions on their linearized trajectories, assuming that they are moving with constant speeds (see Ref. 38). The distance between the ownship (O) and the intruder (I) at time  $t$  for linearized trajectories is given by

$$r(t) = \sqrt{\sum_k (\Delta x_k + \Delta v_k t)^2} \quad (5)$$

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

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



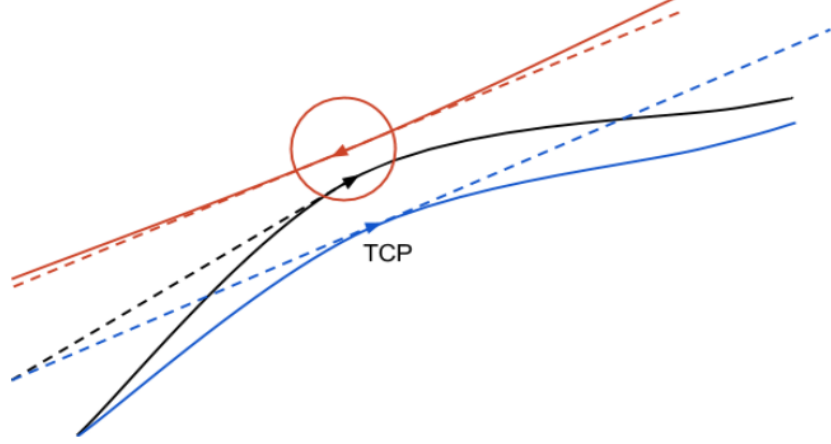


Figure 6. Linearization of the ownship and intruder trajectories near TCP

$(x_I)_k$  - initial intruder's position,

$(v_O)_k$  - ownship's velocity,

$(v_I)_k$  - intruder's velocity.

The distance between ownship and intruder positions at CPA is

$$r(t_{CPA}) = r_{CPA} = \min\{r(t)\} \Leftrightarrow dr(t_{CPA})/dt = 0 \quad (6)$$

and can be obtained from

$$\dot{r} \equiv dr/dt = \frac{\sum_k \Delta x_k \Delta v_k + \sum_k (\Delta v_k)^2 t}{r} = 0 \quad (7)$$

Hence, the time to CPA

$$t_{CPA} = \max \left\{ 0, -\frac{\sum_k \Delta x_k \Delta v_k}{\sum_k (\Delta v_k)^2} \right\} \quad (8)$$

Although this procedure gives the same equation for time to CPA as Ref. 38, 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.

#### 4.2.3 Fast CPA Evaluation Using State-based Threat Detector

The ownship and intruder positions at CPA are simply:

$$(x_O)_i + (v_O)_i \cdot t_{CPA} \quad (9)$$

$$(x_I)_i + (v_I)_i \cdot t_{CPA} \tag{10}$$

Only the horizontal and vertical separation need to be checked to determine whether a corresponding candidate solution will be “locally conflict-free” since  $dr/dt$  used in equations (2) and (3) is equal to zero at CPA according to (6).

#### 4.2.4 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, such as MATG, to account for aircraft performance and to check for conflicts with other intruders. This step involves recalculation of the ownship 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’ trajectories within a specified look-ahead time as described in a section 4.1. Note that intruder trajectories do not need to be re-calculated since they are not perturbed.

#### 4.2.5 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.

#### 4.2.6 Selection of Candidate Resolutions

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 solutions 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.

## 5 Cost Factors

GRA can use any externally defined cost function to select the best resolution maneuver. However, the choice of cost function may have a significant impact on the performance of GRACE.

In JADEM’s cost model the total cost of any maneuver is represented as a weighted sum of individual costs

$$C = \sum_i W_i F_i \tag{11}$$

where  $W_i$  are the “weights” of different “cost factors”  $F_i$ .

The JADEM cost model includes the following terms:

- *maneuver rank cost*, which ranks maneuver types by an order of preference (e.g. to support right of way rules);
- *maneuver type cost*, which is used to penalize or suppress specific maneuver types;
- *maneuver strength cost*, which penalizes maneuvers that are too aggressive, such as sharp turns or steep climbs and descents;
- *maneuver change cost*, which penalizes maneuver type changes within an encounter;
- *holding cost*, which favors shorter maneuvers to reduce the deviations from original flight plan and the impact of uncertainties;
- *collision threat cost*, which increases rapidly, dominating all other costs, as the separation at predicted CPA (for all intruders and within the look-ahead time) decreases.

Other costs (fuel-saving, time-saving, etc.) can be easily added if needed.

The “weights” of all cost factors, except the collision threat cost, are customizable and can be used to fine-tune the behavior of DAA algorithms. Usually the default cost factors work well for GRACE, whether it is used for directive guidance or for OmniBands and WCR.

### 5.1 Maneuver Rank Cost

Each maneuver type is assigned a rank cost to achieve a preferred order:

rank ( $k$ )	SM
0	Right Turn
1	Left Turn
2	Increased vertical speed
3	Reduced vertical speed
4	Slow down
5	Speed up

The rank cost factor is defined as:

$$F_{rank} = k \quad (12)$$

A maneuver with lower rank has lower cost and is therefore preferred over a maneuver with higher rank. For instance, all other things being equal, right turns are preferred over left turns, which in turn are preferred over all other maneuvers. By adjusting  $W_{rank}$  the users can control the significance of this ranking or disable it entirely.

## 5.2 Maneuver Type Cost

Operational considerations might necessitate penalizing or suppressing individual maneuver types on an encounter by encounter basis. For instance, “Vertical Up” maneuvers should be suppressed if the UAS receives a TCAS Resolution Advisory to descend. Another example would be penalizing “Vertical Down” maneuvers for low flying aircraft.

The simplest way to accommodate such requirements is to assign each maneuver type its own cost, which in JADEM is given by

$$F_M = 1 \quad (13)$$

The weights  $W_M$  can be adjusted dynamically (to large values) to penalize or suppress maneuvers as needed.

## 5.3 Maneuver Strength Cost

It is usually undesirable to use extreme maneuvers (very sharp turns, steep climbs, etc.) if it is possible to avoid conflicts without them. For example, if a threat can be avoided using a 10 degree left turn or using a 50 degree right turn, then the left turn should be chosen. This is accounted for by the Maneuver Strength (MS) cost

$$F_{MS} = \frac{p}{p_{lim}} \quad (14)$$

where  $p$  is a maneuver-specific control parameter, and  $p_{lim}$  is its limiting value. For heading maneuvers the control parameter is the difference between the target and the initial headings; for vertical maneuvers this can be a change in flight path angle or vertical speed (load factor). The values of  $p_{lim}$  are based on user-defined operational limits for the corresponding control variables.

## 5.4 Maneuver Change Cost

The Maneuver Change (MC) cost penalizes frequent changes of maneuver types. This is needed to ensure stability of resolution advisories issued by GRACE, which is essential to avoid confusing pilots with unreasonably frequent changes of proposed resolutions.

The following Maneuver Change (MC) cost factor can help alleviate this problem:

$$F_{MC} = 1 - \delta_{k,k'} \quad (15)$$

Here  $k, k'$  are the ranks of last and previous maneuvers recommended by GRA, and  $\delta_{k,k'}$  is the Kronecker delta, which is defined as 1 for  $k = k'$  (the same maneuver), and 0 otherwise (changed maneuver).

## 5.5 Holding Cost

The maneuver Holding (H) cost can be used to minimize the deviations from original flight plan if desired. This can be done by defining the following cost factor:

$$F_H = T_H/T_{lim} \quad (16)$$

Here  $T_H = t_E - t_S$  is the maneuver holding time (duration), defined as the difference between maneuver start time  $t_S$  and maneuver end time  $t_E$ , and  $T_{lim}$  is the look-ahead time.

While Holding and Maneuver costs appear similar (compare Eqs. (16) and (14)), they serve different roles. Figure 7 illustrates that the Maneuver Strength and Holding costs do not always act in the same direction. The dashed line in this figure shows the original ownship route after the arrow indicating a predicted CPA. The end points (TCP) for right and left turns are indicated by  $TCP_{RIGHT}$  and  $TCP_{LEFT}$ . The right turn in this case would have a lower Rank cost and Maneuver Strength cost, but its duration, and hence its Holding cost, would be higher. In this case, GRACE would likely recommend the left turn. This would be consistent with pilot intuition to reduce the deviation from original route and to avoid crossing ahead of the intruder.

## 5.6 Collision Threat Cost

The most safety-critical cost is the cost of approaching the NMAC threshold. This cost should become very high once the separation at CPA becomes lower than the NMAC limit, and it should rapidly decrease as a function of the distance relative to this limit. A suitable simple function to express this dependency, the curl of Agnesi, is used in NMAC (Collision) threat cost factor in the following way

$$F_{NMAC} = \sum_i \frac{1}{1 + (\max\{r_{h,i}/\delta_h, r_{v,i}/\delta_v\})^2} \quad (17)$$

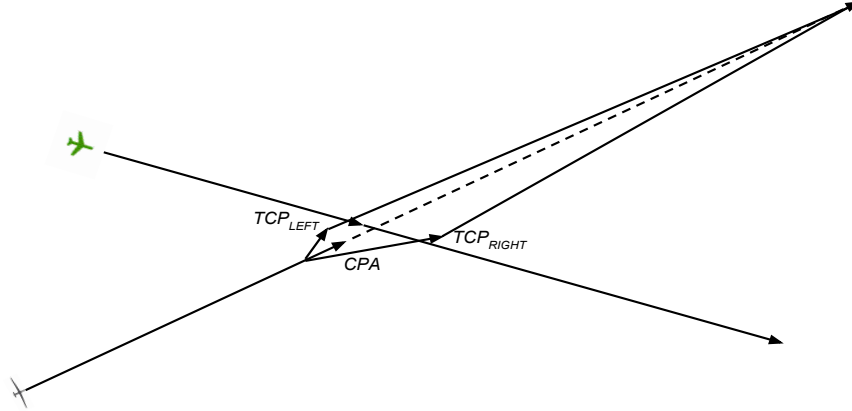


Figure 7. Right and left turns with competing costs

Here  $r_{h,i}$  and  $r_{v,i}$  are the predicted horizontal and vertical separations between the ownship and  $i$ -th intruder at CPA,  $\delta_h$  and  $\delta_v$  are the minimal horizontal and vertical separations, and summation is taken over all intruders with predicted CPA within a specified lookahead time. The values of  $\delta_h$  and  $\delta_v$  are defined as 500 ft and 100 ft respectively, which corresponds to NMAC thresholds, but in general they can be considered as horizontal and vertical uncertainty parameters that can increase with trajectory prediction errors.

$W_{NMAC}$  in the JADEM cost model is not user-defined. It is hard-coded to a large value to ensure that a collision threat cost always dominates over all other costs when CPA distance between the ownship and intruders becomes comparable with NMAC threshold  $\delta_h \times \delta_v$ .

## 6 GRACE Applications

GRACE provides a unique combination of computational performance and flexibility that makes it suitable for use in diverse applications. The most obvious potential uses 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 has been used for modeling UAS DAA systems in a number of parametric/factorial studies [39, 40] and NAS-wide simulations [10, 41, 42]. The methodology of such studies is described in section 7.

The focus of this section is on applications of GRACE for alerting and guidance tools supported by JADEM. These tools have been evaluated and refined in the following HITL studies and flight tests:

- Full Mission HITL simulation (2013), which examined the effect of different command and control interfaces on UAS pilots' ability to get in-the-loop to respond to ATC clearances [43].
- Part Task 4 study (PT4, 2014), which examined UAS pilots' subjective assessments of four DAA display configurations with either basic or advanced levels of information [44].
- Interactive HITL simulation (IHITL, 2014), which was conducted to investigate appropriate procedures for UAS DAA systems and optimal timing for pilots requesting or executing

separation maneuvers [13].

- Part Task 5 study (PT5, 2015), which examined UAS pilots’ subjective assessments of four DAA display configurations with varying forms of maneuver guidance [14, 45].
- Flight Test 3 (FT3, 2015), which validated CPA prediction accuracy and DAA alerting logic and trajectory model (including maneuvers) in realistic flight conditions, and evaluated TCAS interoperability with a DAA system to inform the final SC-228 DAA MOPS [46].
- Flight Test 4 (FT4, 2016), which extended FT3 for multi-intruder encounters, improved guidance stability for noisy surveillance, and added WCR guidance for Regain Well Clear and support for TCAS 71 messages.
- Mini HITL study (2016) that examined maneuver guidance designs for pilots performing Maintain and Regain Well Clear tasks [47].
- Part Task 6 study (PT6, 2016), which investigated TCAS/DAA interoperability to support the final MOPS for displays, alerting, and guidance [48].

In all these studies, GRACE was used to provide alerting for UAS pilots. In Full Mission HITL, PT4, and IHITL studies, GRACE was also used to give directive guidance to pilots. In later HITL studies and flight tests, GRACE was used to compute OmniBands and WCR.

The following sub-sections describe these applications in more detail.

### 6.1 Alerting System for UAS Pilots

GRACE was extensively used in several HITL studies to evaluate different proposals on alerting systems for ground pilots. Figure 8 shows one of the alerting structures that have been evaluated. In this table,  $DMOD$ ,  $HMD^*$ ,  $T_{mod}$ , and  $Z_{THR}$  were defined in Eqs. (1) and (3) and  $t_{CPA}$  is the estimated time to CPA, defined in Eq. (8).





Symbol	Name	Buffered Well Clear Criteria	Alerting Time Threshold	Aural Alert
	Warning Alert	$DMOD = HMD^* = 0.75$ nmi $Z_{THR} = 450$ ft $T_{mod} = 35$ s	25 s ( $t_{CPA} \sim 60$ s)	“Traffic, Maneuver Now”
	Corrective Alert	$DMOD = HMD^* = 0.75$ nmi $Z_{THR} = 450$ ft $T_{mod} = 35$ s	55 s ( $t_{CPA} \sim 90$ s)	“Traffic, Avoid”
	Preventive Alert	$DMOD = HMD^* = 1.0$ nmi $Z_{THR} = 700$ ft $T_{mod} = 35$ s	55 s ( $t_{CPA} \sim 90$ s)	“Traffic, Monitor”
	Traffic	In surveillance field of regard	–	–

Figure 8. An example of alerting structure

In all these studies GCE was used to generate all alert levels. The visual symbols in Fig. 8 were used in the pilot display. In some studies GRA was also used in conjunction with GCE to evaluate different guidance tools and to study TCAS RA interoperability requirements as described in the following sub-sections.

## 6.2 Directive Guidance

The simplest use for GRA is to provide directive guidance to ground pilots. In this case, maneuvers proposed by the algorithm are presented to the pilot at the GCS display. The pilot can accept the proposed maneuvers or choose different ones.

According to post-trial questionnaires in PT4 study, all pilots accepted between 75% and 100% of collision avoidance maneuvers recommended by GRACE.

## 6.3 OmniBands

As an alternative to traditional directive guidance, a new “OmniBands” tool had been recently proposed for evaluation in the HITL study. The OmniBands can be seen as an “omni-directional *determine* function” that defines alerted and conflict-free headings and altitudes for all intruders within a specified look-ahead time. In Fig. 9 the heading OmniBands are shown as red, yellow, and green sectors at the center indicating the ranges of target headings (the “bands”) that would result in high severity conflicts, low severity conflicts, and no conflicts respectively. A pilot is expected to use the OmniBands information for choosing a target heading, preferably within one of “green” sectors. The altitude OmniBands can be used in a similar way (they are presented as the bars of different colors as seen on the right side of Fig. 9). Note that the appearance of the OmniBands in the figure is display specific.

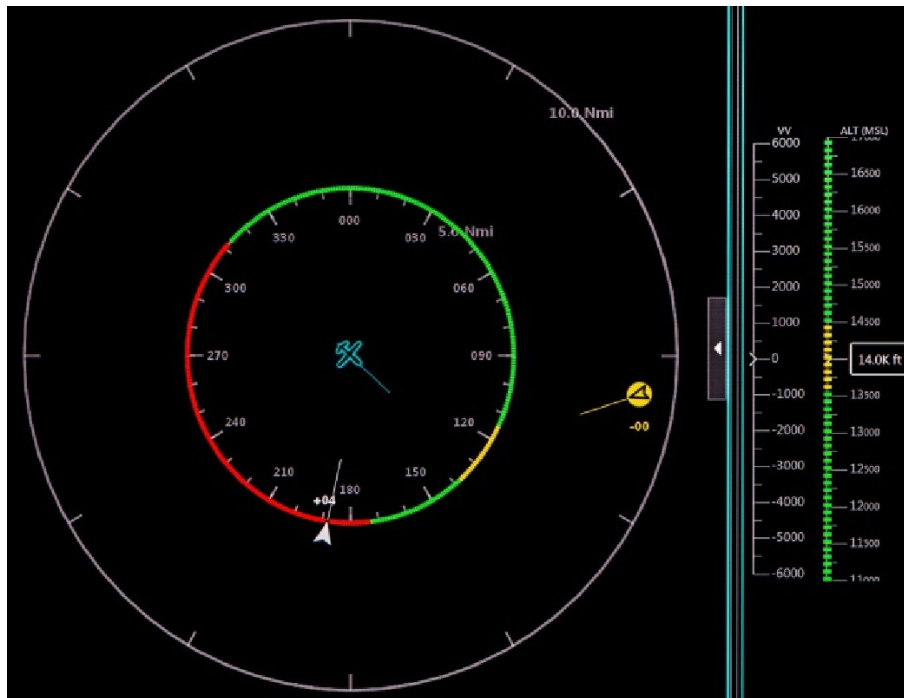


Figure 9. A concept of OmniBands tool

Support for OmniBands is more complex than directive guidance since the intruders may or may not be in a predicted conflict. In this case the *evaluate* function based on the current ownship’s position



and intent is irrelevant. The brute-force approach, evaluating threats for all target headings, would not provide the required realtime performance.

Therefore, a faster algorithm for OmniBands was introduced in JADEM. It relies on the ability of GRA to quickly find candidate conflict-free resolutions, such as right and left turns for heading maneuvers, for any intruder representing a potential threat. These conflict-free resolutions define the bounds of an alerted heading band for this threat.

An intruder represents a potential threat if it can be reached (“intercepted”) from current ownship state. This can be determined efficiently by chasing an intruder’s predicted CPA without generating the entire intercepting trajectory.

Therefore, the algorithm for finding heading bands does the following,

1. for each separation standard (each corresponding to a severity level)
  - (a) determine potential threats by finding the intruders that can be intercepted;
  - (b) find bands for each of these threats from GRA maneuvers in both directions;
2. assemble the OmniBands for all intruders and severity levels by combining the bands for individual intruders from previous step.

The algorithm performs linearly with the number of potential threats (intercepted intruders) and has near-constant performance for all other intruders because the intercept search is very fast.

In principle, a similar approach could be used for altitude bands as well. However, since the altitude bands had to be defined only at a few discrete altitude levels, a simpler “almost-brute-force” algorithm provided adequate performance. The algorithm simply calls GCE for each of discrete altitude levels to determine, which of these levels present high and low severity threats.

In addition to meeting the performance requirements for OmniBands calculations, the use of GRACE also ensures that all heading and altitude bands are “realistic” as based on ownship performance parameters for rates of turn, climb, and descent (see Section 4.2.6).

## 6.4 Well Clear Recovery

OmniBands guidance improved pilot performance in avoiding LoWC and was well received by the pilots, but it was found to have one major problem. When Loss of Well Clear (LoWC) was imminent, the OmniBands became “saturated” meaning that they included no “green” intervals at all. Therefore, the pilots were left without any guidance in situations when they needed it the most.

To remedy this drawback, the concept of Well Clear Recovery (WCR) was proposed and implemented in JADEM. WCR guidance tells a pilot the safest direction to maneuver to restore Well Clear.

WCR guidance bases its decisions on the maneuver suggested by GRA in “UseBest” control mode (see Section 4.2.6) and can be presented as a “wedge” on the pilot’s display. If the best maneuver is a left or right turn, it is shown as a green sector (Fig. 10). The vertical maneuver can be displayed as a sliding bar above or below the current ownship altitude in altitude band intervals.



Figure 10. Well Clear Recovery guidance

The low and high bounds of the wedge are determined from the values of the relevant control variable (e.g. heading or altitude) selected by GRA with certain corrections as follows:

- the low bound is the lowest value of the control variable needed for a timely recovery from LoWC, and the high bound is the GRA maneuver limit;
- both bounds are snapped to a specified grid;
- in all cases the difference between high and low bounds cannot be smaller than a configurable minimal wedge width.

This is very similar to using GRA as directive guidance, except that a pilot can choose any value within a wedge.

## 6.5 TCAS Interoperability

It is expected that the UAS flying in Class A and in transitional Class E airspace will be equipped with TCAS system. Therefore, the DAA system must ensure interoperability with TCAS guidance in encounters with TCAS-equipped intruders. JADEM was extensively used to validate the proposed DAA MOPS requirements for TCAS interoperability in recent HITL studies and flight tests.

GRACE flexibility was critical for success of these experiments since it made it easy to disable the vertical maneuvers on Corrective TCAS RA or to suppress the individual maneuvers that were not consistent with Preventive TCAS RA (see Section 5.2).

## 7 Evaluation of GRACE Performance

This section provides sample results of GRACE performance 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 [49] and recorded VFR traffic data [50] 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 (see section 7.1); 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.

In the remainder of this section the sensor and tracker models used to generate noisy data are briefly described, followed by a summary of the metrics used in evaluations of GRACE performance. Results for the parametric evaluation are then presented, followed by results for the NAS-wide evaluation.

### 7.1 Sensor and Tracker Models

As mentioned above, sensor and tracker models were used to provide noisy detection of traffic. The sensor models, developed by Honeywell under contract to NASA, include ADS-B, Mode-S, Mode-C, and airborne radar sensors.

The ADS-B model is used to detect intruders equipped with ADS-B out. Mode S and Mode C models detect intruders equipped with Mode S or Mode C transponders. The Radar model is used to detect intruders not equipped with any transponder.

Roughly speaking, ADS-B provides very accurate horizontal position and altitude, Mode S and Mode C provide noisy horizontal position but accurate altitude, and radar provides fairly accurate horizontal position but noisy altitude. Note that, in addition, the radar model detects intruders probabilistically, meaning that intruders within its nominal field of regard have a finite probability of not being detected and vice versa. Table 1 summarizes the radar characteristics used in section 7.3.

In addition to the four sensor models, a navigation model is provided, which creates “noisy” ownship states with position and velocity error.

The tracker used in this study is a proprietary fusion tracker developed by Honeywell [51].

Table 1. Airborne Radar Parameters

Field of View			Position Error			Detection Probability
Range	Bearing	Elevation	Range	Bearing	Elevation	
13 <i>nmi</i>	135°	20°	10 <i>m</i>	0.4°	0.4°	0.95%

## 7.2 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.

<b>Conflicts</b>	number of conflicts (encounters)
<b>Resolutions</b>	total number of resolutions proposed by GRE
<b>Changes Per Encounter</b>	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)
<b>Predicted Violations</b>	number of times when GRE predicted a violation (as defined in 4.1.1) for any separation standard within a specified look-ahead time
<b>Actual Violations</b>	number of times when GRE detected a violation; this indicates that an avoidance maneuver failed to resolve a conflict
<b>Failure Rate</b>	a ratio of number of failures to resolve a conflict to number of detected conflicts in %
<b>Predicted NMAC</b>	number of times when GRE predicted a NMAC within a specified look-ahead time
<b>Actual NMAC</b>	number of times when GRE detected an NMAC; this indicates a failure of DAA system to prevent an imminent NMAC
$S_{\text{NMAC}}$	$S_{\text{NMAC}} = \max\left(\frac{R_{\text{NMAC}}}{R_{\text{CPA}}}, \frac{Z_{\text{NMAC}}}{Z_{\text{CPA}}}\right) \times 100$ , where $S$ is a simple measure of severity, $R$ is the range, and $Z$ is the vertical separation. $S_{\text{NMAC}}$ exceeds 100% in case of <b>Actual NMAC</b> and lower values of $S_{\text{NMAC}}$ correspond to larger separations at CPA relative to NMAC zone

## 7.3 Parametric Evaluation of GRACE Performance

JADEM provides a way to automatically generate a diverse set of encounter geometries for a given ownship flight. This capability was used to evaluate performance of GRACE as a guidance algorithm. JADEM models automatic guidance by immediately executing GRACE recommended maneuvers. To model pilots in the loop, JADEM introduces configurable delays in evaluation and execution of GRACE resolutions. In both cases JADEM tries to “recapture” the earliest reachable waypoint after the end of a GRACE maneuver.

GRACE performance was evaluated for perfect and noisy non-cooperative sensors with 8*nmi* detection range. 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 departing from cartesian coordinates  $X = Y = 0$  and following the multi-turn mission plan shown in Fig. 11. This mission plan was typically used in HITL simulations based on Oakland Air Route Traffic Control Center, sector 40/41.

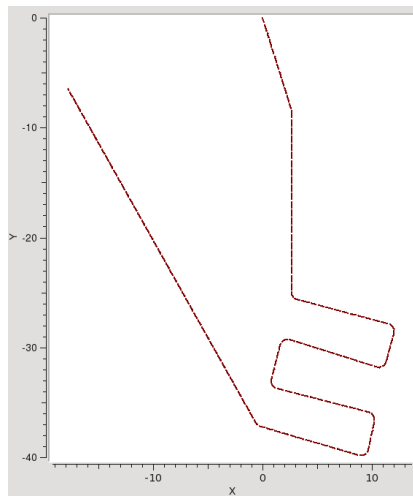


Figure 11. The ownship flight

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 LoWC with  $DMOD = HMD = 0.66nmi$ ,  $Z_{THR} = 450ft$ , and  $modTau = 35sec$ . If LoWC 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 2 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  $8nmi$ , a height of  $\pm 5000ft$ ;
2. Directional Perfect: a tracker and sensor model of airborne radar, configured as defined in Table 1, but using the 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 the perturbed ownship and intruder states with added sensor noise and navigation errors as defined in Table 1.

With unlimited look-ahead time and sensor Field-of-Regard any algorithm can be reasonably expected to avoid all LoWC events. Table 2 demonstrates that with an  $8nmi$  sensor range and a  $2min$

look-ahead time, GRACE was still able to prevent most LoWC events. Moreover, GRACE avoided NMAC in all cases when LoWC 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 2. 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 case of the omnidirectional sensor 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 were already in LoWC state. 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 4.2.2).

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 3 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 LoWC events.

Table 4 illustrates the effect of pilot response time given a conservative 15-second algorithm delay.

Table 3. 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 (%)	$S_{\text{NMAC}}$ (%)	
						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

For all pilot delays GRACE ensured that all NMACs could be avoided with comfortable safety margin ( $S_{\text{NMAC}}$  below 20%). The number of encounters that resulted in LoWC 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 4. 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 (%)	$S_{\text{NMAC}}$ (%)	
						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

## 7.4 Effectiveness in NAS-wide Mitigated Studies

The effectiveness of GRACE in mitigating collision threats was validated using 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 [10, 49]. 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 [49]. 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 (MATG) 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 5 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 5. 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 clearly show that GRACE, used as an automatic guidance algorithm, could prevent almost all violations of separation, and if this was not possible because of late detection or unexpected intruder maneuver, GRACE could still avoid the NMAC.

## 8 Conclusions

Safe integration of UAS into the NAS required 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, OmniBands, and Well Clear Recovery guidance.

GRACE was designed without any assumptions regarding performance capabilities of 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 capability, in remotely piloted 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. Sweet, D. N., Aronson, J. S., Roth, K., and Blake, M., “Fast-Time Simulation System for Analysis of Advanced Air Transportation Concepts,” *AIAA Modeling and Simulation Technologies Conf*, 2002, AIAA-2002-4593.
9. “NASA Technology Development Project Plan,” *Unmanned Aircraft Systems (UAS) Integration in the National Airspace System (NAS)*, Jan. 2013.
10. Lee, S. M., Park, C., Thippavong, 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.

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. 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.
13. 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.
14. 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.
15. 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.
16. Harman, W. H., "TCAS: A System for Preventing Midair Collisions," *Lincoln Laboratory Journal*, 1989, vol. 2, no. 3.
17. 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.
18. 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.
19. 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.
20. 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.
21. 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.
22. 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,
23. Erzberger, H., "The Automated Airspace Concept," *Proceedings of the Fourth USA/Europe Air Traffic Management R & D Seminar*, Dec. 2001.
24. Kelly, W. and Eby, M., "Advances in Force Field Conflict Resolution Algorithms," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Aug. 2000.

25. 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.
26. 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.
27. Dougui, N., Delahaye, D., Peuchmorel, S., and Mongeau, M. “A light propagation model for aircraft trajectory planning,” *Journal of Global Optimization*, 56(3), 2013.
28. 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
29. Roussos, G., Kyriakopoulos, K. “Towards constant velocity navigation and collision avoidance for autonomous nonholonomic aircraft-like vehicles.” *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. Proceedings of the 48th IEEE Conference*, Shanghai, 2009
30. 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
31. 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
32. Sawhill, B., “Modeling Systemic Phenomena in the National Airspace System,” *NASA Aeronautics Airspace Systems Program Technical Interchange Meeting*, March 2011.
33. 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.
34. Watkins, O. and Lygeros, J., “Stochastic Reachability for Discrete Time Systems: An Application to Aircraft Collision Avoidance,” *IEEE Conference on Decision and Control*, 2003.
35. 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
36. Abramson, M. and Ali, K., “Integrating the Base of Aircraft Data (BADA) in CTAS Trajectory Synthesizer,” 2012, NASA/TM-2012-216051.
37. Consiglio, M., Chamberlain, J., Muñoz, C., and Hoffer, K., “Concept of integration for UAS operations in the NAS,” *28th International Congress of the Aeronautical Sciences*, 2012, Table 1: TCAS Sensitivity Level Definition and Alarm Thresholds.
38. Muñoz, C., Narkawicz, A., and Chamberlain, J., “A TCAS-II Resolution Advisory Detection Algorithm,” *AIAA Guidance, Navigation, and Control Conference*, 2013.

39. Cone, A., Thippavong, 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
40. Thippavong, 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
41. 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.
42. Thippavong, 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
43. 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.*
44. Monk, K.J., Fern, L., Rorie, R.C., and Shively, R.J., "Effects of display location and information level on UAS pilot assessments of a detect and avoid system," *Proceedings of the Human Factors and Ergonomics Society 59th Annual Meeting, Los Angeles, CA, 2015.*
45. 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.*
46. 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.
47. 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.*
48. 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*
49. 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.*
50. 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.
51. Bageshwar, Vibhor L and Euteneuer, Eric A, "Multi-Intruder Aircraft, Multi-Sensor Tracking System," *2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC), 2015.*