

Small UAV Automatic Ground Collision Avoidance System Design Considerations and Flight Test Results

Paul Sorokowski
412th Test Wing, Edwards, California

Mark Skoog
Armstrong Flight Research Center, Edwards, California

Capt. Scott Burrows
412th Test Wing, Edwards, California

SarahKatie Thomas
412th Test Wing, Edwards, California



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 Reports Server, thus providing one of the largest collections 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 counter-part of peer-reviewed formal professional papers but has 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.

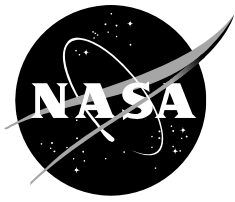
- **CONFERENCE PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.
- **SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with 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 organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

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

- Access the NASA STI program home page at <http://www.sti.nasa.gov>
- E-mail your question to help@sti.nasa.gov
- Phone the NASA STI Information Desk at 757-864-9658
- Write to:
NASA STI Program
Mail Stop 148
NASA Langley Research Center
Hampton, VA 23681-2199

NASA/TM—2015–218732



Small UAV Automatic Ground Collision Avoidance System Design Considerations and Flight Test Results

*Paul Sorokowski
412th Test Wing, Edwards, California*

*Mark Skoog
Armstrong Flight Research Center, Edwards, California*

*Capt. Scott Burrows
412th Test Wing, Edwards, California*

*SarahKatie Thomas
412th Test Wing, Edwards, California*

National Aeronautics and
Space Administration

*Armstrong Flight Research Center
Edwards, CA 93523-0273*

June 2015

Trade names and trademarks are used in this report for identification only. Their usage does not constitute an official endorsement, either expressed or implied, by the National Aeronautics and Space Administration.

Available from:

NASA STI Program
Mail Stop 148
NASA Langley Research Center
Hampton, VA 23681-2199

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
703-605-6000

This report is also available in electronic form at <http://www.sti.nasa.gov/> and <http://ntrs.nasa.gov>.

Table of Contents

Abstract.....	1
Nomenclature.....	1
Introduction.....	2
Research Objectives.....	4
Design Philosophy and Top-Level Requirements.....	5
Test Aircraft Requirements.....	5
Ground Control Van Requirements	6
Host Computer Requirements.....	6
Avoidance Maneuver Performance Requirements.....	7
Trajectory Prediction Requirements	8
Digital Elevation Model Data Requirements	9
Flight-Test Analysis Support Requirements	9
Flight-Testing as Part of the Design Process	9
Research System Overview	10
Dryden Remotely Operated Integrated Drone (DROID) Test Aircraft.....	10
Ground Control Van	12
Physical Components of SUAV Auto GCAS	14
Software Components of SUAV Auto GCAS	16
Generic “Sense Own-State” Module.....	17
SUAV Auto GCAS “Sense Own-State” Module.....	20
Generic “Sense Terrain” Module	20
SUAV Auto GCAS “Sense Terrain” Module	24
Generic “Common Interface” Module	24
SUAV Auto GCAS “Common Interface” Module	26
Signal Conditioning	27
Exception Handling	28
Connectivity and Throughput Checks.....	28
Generic “Predict Avoidance Trajectories” Module	28
SUAV Auto GCAS “Predict Avoidance Trajectories” Module.....	32
Generic “Identify Collision Threats” Module.....	36
Man-Made Vertical Obstructions.....	39
Natural Vertical Obstructions (Trees).....	40

SUAV Auto GCAS “Identify Collision Threats” Module	40
Generic “Determine Need to Avoid” Module.....	43
SUAV Auto GCAS “Determine Need to Avoid” Module.....	45
Nuisance Potential	45
Generic “Avoid” Module	46
SUAV Auto GCAS “Avoid” Module	47
Pilot-Vehicle Interface	47
SUAV Mode States	49
System Ground Testing.....	50
Flight-Testing.....	50
Test Objectives.....	51
Test Sites.....	51
Ground Survey at the GCAS Valley Test Site	54
Test Methods.....	56
Pilot-Activated Recovery System	56
Collision Avoidance Testing.....	56
Mishap Scenarios	59
Nuisance Evaluation Techniques	59
Test Preparation	62
SUAV Auto GCAS Avoidance Maneuver Simulation	62
Avoidance Maneuver Visualization.....	63
Test Cards	65
Data Recording Limitations	67
Test Results.....	68
Controlled Flight Into Terrain Protection	68
Nuisance Potential	70
Available Reaction Time	70
UAV Mission Segments.....	72
Avoidance Maneuver Termination	74
Trajectory Prediction Accuracy	79
Scan Pattern Functionality	82
Avoidance Maneuver Summary.....	86
Concluding Remarks.....	89

Summary of Recommendations	91
Appendix A: Parameter List	93
Appendix B: Analysis Techniques.....	97
Google Earth Interface.....	97
Excel® Test Planning Tool.....	97
Excel® - Google Earth Interface.....	99
Available Reaction Time	101
Three Segment Extrapolation Methods.....	104
Delay Segment.....	104
Delayed Avoidance Maneuver.....	105
Post-Termination: Mountainous Terrain.....	106
Post-Termination: Smooth Terrain	106
Comparison to Digital Terrain Truth Data.....	107
Determination of Trigger Post Locations.....	108
Determination of Worst-Case Mismatch Locations.....	110
Trajectory Normalization Analysis.....	112
Finding Distances.....	114
Nondimensionalizing Distances and Plotting on a Representative Trajectory Prediction Algorithm	116
Termination Logic Cross-Checks	117
Appendix C: Flight Log	122
Appendix D: Open Discrepancy Reports.....	124
References.....	127

Abstract

The National Aeronautics and Space Administration (NASA) Armstrong Flight Research Center Small Unmanned Aerial Vehicle (SUAV) Automatic Ground Collision Avoidance System (Auto GCAS) project demonstrated several important collision avoidance technologies. First, the SUAV Auto GCAS design included capabilities to take advantage of terrain avoidance maneuvers flying turns to either side as well as straight over terrain. Second, the design also included innovative digital elevation model (DEM) scanning methods. The combination of multi-trajectory options and new scanning methods demonstrated the ability to reduce the nuisance potential of the SUAV while maintaining robust terrain avoidance. Third, the Auto GCAS algorithms were hosted on the processor inside a smartphone, providing a lightweight hardware configuration for use in either the ground control station or on board the test aircraft. Finally, compression of DEM data for the entire Earth and successful hosting of that data on the smartphone was demonstrated. The SUAV Auto GCAS project demonstrated that together these methods and technologies have the potential to dramatically reduce the number of controlled flight into terrain mishaps across a wide range of aviation platforms with similar capabilities including UAVs, general aviation aircraft, helicopters, and model aircraft.

Nomenclature

ACAT	Automatic Collision Avoidance Technology
AFRC	Armstrong Flight Research Center
AGL	above ground level
ART	available reaction time
Auto GCAS	Automatic Ground Collision Avoidance System
CFIT	controlled flight into terrain
CDTM	Compressed Digital Terrain Maps
CONUS	contiguous United States
COTS	commercial off-the-shelf
DEM	digital elevation model
DR	Discrepancy Report
DROID	Dryden Remotely Operated Integrated Drone
DSOC	Defense Safety Oversight Council
DTED	Digital Terrain Elevation Data
EGI	embedded GPS/INS
EAFB	Edwards Air Force Base
FRRP	Fighter Risk Reduction Project
GCAS	Ground Collision Avoidance System
GCO	ground control operator
GEDACS	Global Elevation Data Adaptive Compression System
GPS	global positioning system
HIL	hardware-in-the-loop
HOTAS	Hands On Throttle-And-Stick
HUD	head-up display
HWY	highway
INS	inertial navigation system
KIAS	knots indicated airspeed
KTAS	knots true airspeed
MSL	mean sea level

NASA	National Aeronautics and Space Administration
NAV	buffer to account for GPS navigation accuracy
NED	National Elevation Dataset
NGA	National Geospatial-Intelligence Agency
PARS	Pilot Activated Recovery System
PCC	Piccolo II autopilot command center
PVI	pilot-vehicle interface
RPV	remotely piloted vehicle
SD	secure digital
SRTM	Shuttle Radar Topography Mission
SUAV	small unmanned aerial vehicle
TCB	terrain clearance buffer
TM	telemetry
TPA	trajectory prediction algorithm
UAS	unmanned aircraft system
UAV	unmanned aerial vehicle
UI	user interface
UIO	user interface operator
WGS84	World Geodetic System 1984
WP	waypoint
α	half angle
β	half width
B	base width

Introduction

Automatic Ground Collision Avoidance Systems (Auto GCAS) have been demonstrated in research and development flight-testing on multiple projects for over 25 years. Most early Auto GCAS development took place during the Advanced Fighter Technology Integration (AFTI) F-16 airplane (General Dynamics, now Lockheed Martin, Bethesda, Maryland) project in the late 1990s (ref. 1). The F-16 Fighter Risk Reduction Project (FRRP) was completed in 2010 and matured Auto GCAS technology to the point where it was ready to be integrated with production F-16 hardware and software (ref. 2). A production version of Auto GCAS was recently flight-tested on United States Air Force (USAF) F-16 airplanes with deployment to operational squadrons beginning in 2014 (ref. 3). A derivative implementation of Auto GCAS has also been flight-tested on the F-22 airplane (Lockheed Martin, Bethesda, Maryland) (ref. 4). An Auto GCAS implementation has also been planned for the F-35 airplane (Lockheed Martin, Bethesda, Maryland) after initial operational capability. All of these projects were components of the broader Automatic Collision Avoidance Technology (ACAT) program under the guidance of the Defense Safety Oversight Council (DSOC).

The DSOC has concluded that the comprehensive 75-percent mishap reduction rate mandated by the Secretary of Defense (refs. 5 and 6) cannot be achieved without the widespread implementation of Auto GCAS. Manual terrain avoidance systems warn the pilot and rely on pilot reactions to avoid terrain. Such systems have already been implemented on various platforms with mixed success. However, Controlled Flight Into Terrain (CFIT) mishaps continue to occur on many platforms within the Department of Defense (DoD). Controlled Flight Into Terrain mishaps also continue to result in an alarming number of general aviation fatalities every year.

The primary objections to Auto GCAS have been perceived nuisance potential and perceived cost. Nuisance potential is a valid concern but specific design requirements to minimize nuisance activations can be an integral part of any system. Cost is also a valid concern, but on many platforms the cost of fleet-wide implementation of Auto GCAS can be recovered by the prevention of a single CFIT mishap (ref. 7).

The DSOC has been highly motivated to find low-cost Auto GCAS alternatives with applications across widely varying aircraft types. As a result, the DSOC funded a limited-scope design and flight-test effort to transition Auto GCAS technologies from the previous F-16 system to a small unmanned aerial vehicle (SUAV). An overall goal of the SUAV Auto GCAS project was to enhance the proven technologies within Auto GCAS whenever practical, while at the same time exploring techniques to achieve significant cost savings.

This report has two primary goals:

- Communicate the design concepts and flight-test results from the SUAV Auto GCAS project; and
- Compare SUAV Auto GCAS design concepts to those used on other successful Auto GCAS projects (primarily the F-16) with the intent of helping future projects identify tradeoffs and determine which concepts can be tailored to best serve a given platform.

One of the most important lessons learned from recent Auto GCAS projects has been that there are many design approaches that will work. However, there will always be important tradeoffs that must be understood to improve the likelihood that a given design will be successful.

This report identifies a few of the most significant tradeoffs and design choices. Design concepts that work on a low-speed UAV may not be best for a high-speed fighter, and vice versa. As Auto GCAS is applied to a broader spectrum of platforms, such as transports, general aviation, and helicopters, it is expected that design concepts will continue to evolve. In spite of differences in aircraft type, it is also expected that a modular Auto GCAS architecture will be applicable regardless of platform.

The SUAV Auto GCAS project successfully demonstrated many important collision avoidance technologies. The foremost of these demonstrations included:

- Auto GCAS testing with multiple avoidance maneuvers including turns to either side;
- Testing of digital terrain scanning techniques determined directly from the predicted trajectory;
- In-flight-testing of highly compressed digital elevation maps;
- In-flight-testing of digital elevation maps that had been customized to reflect tighter tolerances in some areas, and relaxed tolerances in other areas;
- In-flight-testing of Auto GCAS on a UAV; and
- Auto GCAS algorithms were hosted on a smartphone during flight tests.

Since one of the motivations for this report is to assist design choices on future projects, recommendations are introduced as important design aspects are described. The relevant SUAV Auto GCAS flight-test results may not be described in detail until later in the report.

The specific implementation techniques used on the SUAV project were accomplished to provide an inexpensive test bed for the new Auto GCAS technologies, and were not intended to be a direct path to a production UAV implementation. Therefore, any production implementation of Auto GCAS on a UAV would still need to be customized for the intended platform.

For example, an appropriate level of redundancy would be essential for production UAV implementation. This SUAV project was essentially single-thread, which was practical for this research test

bed because a safety pilot was always ready to take control if the Auto GCAS algorithm commanded an inappropriate aircraft response. A production UAV implementation of Auto GCAS would require the level of redundancy necessary to meet the requirements of that system.

Using the smartphone to host the Auto GCAS algorithm was an implementation technique that was specific to this SUAV test bed demonstration. The smartphone posed specific challenges in terms of the need to use the Java™ (Oracle Corporation, Redwood Shores, California) programming language, the need to address timing aspects for real-time versus background tasks, and the need to utilize particular verification and validation techniques. This report does not address the smartphone-specific challenges. Although smartphones continue to be used for follow-on Auto GCAS projects, those are also considered test bed applications. A production application would require consideration of the host computer reliability as well as the abovementioned redundancy aspects.

Note: This report uses wording consistent with the terminology used on the SUAV Auto GCAS and F-16 Auto GCAS projects at the time. An activation on SUAV Auto GCAS is called an “avoidance maneuver;” an activation on F-16 Auto GCAS is called a “flyup” maneuver.

Research Objectives

The original DSOC customer direction was to:

- “Ensure interoperability between Auto GCAS functions and a small Unmanned Aircraft System (UAS);
- Tailor F-16 Auto-GCAS algorithms to the Dryden Remotely Operated Integrated Drone (DROID) UAV platform;
- Improve the Auto GCAS design where necessary to support UAS-class aircraft;
- Implement Auto-GCAS algorithms on a palm-sized personal computer;
- Demonstrate portability of Auto GCAS algorithms;
- Flight-test auto GCAS on a small UAS; and
- Provide regression level-of-effort for future platform integration.”

The intent of the last listed item was to minimize the effort required to implement Auto GCAS onto future platforms. Most of the items were blended as sub-objectives into the final project objectives. Additional discussions with the customer resulted in the following top-level project objectives, as documented in reference 8.

- Demonstrate the portability of the Auto GCAS algorithms;
- Demonstrate and evaluate the Auto GCAS integrated with a UAS platform mimicking a generic medium-to-large UAV hosted in a ground-based implementation;
- Demonstrate and evaluate the Auto GCAS integrated with a UAS platform mimicking a generic medium-to-large UAV hosted in an on-aircraft-based implementation; and
- Identify attributes of the Auto GCAS design which are not easily ported to other aerial vehicles.

These top-level project objectives provided more explicit focus on generic UAVs as a long-term application for this technology demonstration. Generic UAVs were selected for two reasons. First, UAVs are prime candidates for incorporation of an Auto GCAS. Second, the National Aeronautics and Space Administration (NASA) Armstrong Flight Research Center (AFRC) owned and operated a broad range of UAVs, including an MQ-9 Reaper (General Atomics, San Diego, California) and an RQ-4 Global Hawk (Northrup Grumman Corporation, Falls Church, Virginia). The NASA UAVs provided logical paths forward for future Auto GCAS research projects.

The first three top-level objectives were met. The fourth top-level objective was indirectly met; no Auto GCAS attributes were identified that would be difficult to port onto other aerial vehicles.

Design Philosophy and Top-Level Requirements

An Auto GCAS Requirements Guide has been established by the Air Force Research Laboratory (ref. 9). This guide provides a step-by-step process to help any Auto GCAS project develop specific requirements based on the mission and maneuvering capability of that platform. The Auto GCAS Requirements Guide also enables Auto GCAS developers to make independent decisions regarding each design aspect while considering the lessons learned over several decades of Auto GCAS development on multiple past projects. To date, this Requirements Guide has been used to influence design decisions on the following Auto GCAS projects:

- F-16 Auto GCAS Fighter Risk Reduction Program;
- Small UAV Auto GCAS;
- F-16 production Auto GCAS;
- F-35 production manual and Auto GCAS; and
- General Aviation Ground Proximity Warning System (a California Polytechnic State University, San Luis Obispo, student design project).

The initial SUAV Auto GCAS project requirements were intentionally kept flexible in order to maximize the potential return on investment for this research project. As trade studies were accomplished, detailed design requirements for each Auto GCAS subsystem were refined within design documentation (ref. 10). The fundamental top-level project requirements for the SUAV project are summarized in the following sections so that related projects can identify similarities and differences.

Test Aircraft Requirements

It was decided early on that the test aircraft would be a Dryden Remotely Operated Integrated Drone (DROID) already owned and operated at AFRC. The project team developed the following fundamental aircraft capability requirements that became integral to the project:

- “Provide for cost effective system integration;
- Dramatically different platform than an F-16 (in terms of performance, maneuvering capability, sophistication, commercial off-the-shelf (COTS) components availability, and the fact that it was unmanned);
- Maneuvering authority sufficient to mimic a generic medium-to-large UAV (the DROID’s slower top speed was judged acceptable for the purpose of this testing);
- Pre-existing onboard autopilot capable of executing the “generic UAV” terrain avoidance maneuvers based on commands from the host computer. This included auto-throttle capability.
- Aircraft modifiable to install host computer;
- Onboard GPS/INS;
- Autonomous control from the ground control van (waypoint tracking);
- Direct control from a ground cockpit (similar to an RPV);
- Direct line-of-sight control by a safety pilot (to be located near the prominent terrain features); with more maneuvering authority than the generic UAV imitation; and
- Approximate 1-hr mission length (constrained by fuel and batteries).”

See figure 1 for a comparison of the DROID test aircraft and the F-16 used during the Auto GCAS FRRP.



Figure 1. Comparison of F-16 and DROID.

Ground Control Van Requirements

The ground control van needed to be capable of supporting all of the test aircraft requirements listed above. Another fundamental requirement for the ground control van was that it could be successfully deployed to remote desert locations.

Host Computer Requirements

It was originally intended that the Auto GCAS algorithms be hosted in their entirety on a palm-sized personal computer. The intent was that the host computer could be installed on the SUAV without excessive gross weight impacts.

During development it was determined that this requirement could be met using a smartphone based on the Android™ platform (Google Inc., Mountain View, California). That decision was made in order to demonstrate a low-cost alternative and to maximize portability for future applications on other aircraft, especially within general aviation.

The project team also developed the fundamental host computer requirements which were:

- Commonly available computational device;
- Provide a growth path for follow-on projects;
- Capability to interface with on-board autopilot and other system components in either “on-aircraft” or “on-ground” configurations;
- Sufficient storage for all Auto GCAS algorithms including the necessary Digital Elevation Model (DEM) data and flight-test data;
- Sufficient throughput for Auto GCAS algorithms to execute within an acceptable frame time while also recording the designated flight test data; and

- Develop a method to allow an operator to interface with the host computer to control Auto GCAS setup and options from the ground control van.

From project inception, it was understood that these fundamental requirements would be met through ongoing design tradeoffs. It was already known that the smartphone storage capacity (32 GB) was more than sufficient to store all of the Auto GCAS algorithms and DEM data for the entire Earth at a resolution acceptable for aircraft with predominant missions at high altitude, while also including local maps at the higher resolution needed for an SUAV test mission. The primary design constraint related to storage capacity was that Auto GCAS data were only recorded at the initiation of each avoidance maneuver. By comparison, the F-16 had recorded Auto GCAS data throughout the flight, which was much more useful for assessing nuisance potential in situations in which flyups did not activate.

It was known that the smartphone throughput capacity was sufficient to complete the basic Auto GCAS functions at a rate that was considered adequate. In addition, the requirement could also be met by including a forecast for time delay as part of the Auto GCAS algorithm. Therefore, compensation for a slower frame rate could be achieved by increasing the time delay within the avoidance maneuver prediction portion of the algorithm. In other words, avoidance maneuvers would happen sooner if the frame rate were slower. The primary design constraint related to throughput capacity was the rate at which flight-test data could be recorded without interfering with the basic Auto GCAS functions.

It was also known that the on-ground configuration would increase the risk of loss of communication between the aircraft and the ground control van, as compared to the on-aircraft configuration. The on-ground configuration was implemented to evaluate Auto GCAS applications to future platforms that could not accept weight increase. The on-aircraft configuration was recognized as the more robust implementation and would be preferred for any future platform that could accept a small weight increase. Both configurations were designed and flight-tested in order to obtain a better understanding of the real-world performance.

One of the original customer requirements was to “Tailor F-16 Auto-GCAS algorithms to the Dryden Remotely Operated Integrated Drone (DROID) UAV platform.” Part of the intent behind that requirement was to start with the actual F-16 Auto GCAS C++ code and adapt that code as needed. However, after initiating SUAV Auto GCAS design work it became clear that the available documentation was insufficient to understand the F-16 approach to modularity and was insufficient for third-party tailoring of the algorithm. Once aware of those difficulties, the customer agreed that project goals could be achieved without using the F-16 code as a direct basis.

It also became apparent that using C++ code was not the best choice for implementation on a smartphone. The C++ code had seemed to be the logical choice because of the F-16 reference model and because most of the project programmers were already very familiar with C++ code. The C++ code required a wrapper in order to run on the smartphone, but that interpreter did not include some of the required capability, such as trigonometric functions. Therefore it was decided to use Java™ because it was native to the smartphone and completely avoided the problems associated with trying to use the C++ code. The skill of the main SUAV Auto GCAS programmer facilitated a very smooth transition to Java™.

Avoidance Maneuver Performance Requirements

The initial concept for this project was to re-host the previous F-16 flyup maneuver and adapt it to the maneuvering capability of the generic medium-to-large UAV. The F-16 flyup maneuver can be most simply described as “roll toward wings-level and pull at 5 g until clear of terrain.” The F-16 flyup maneuver was basically a maneuver flying straight over terrain after reaching wings-level. That type of maneuver worked reasonably well for the F-16, given the large amounts of kinetic energy that could be turned into altitude.

Not surprisingly, the simulated medium-to-large UAV had much more limited climb capability than did the F-16. Early design studies indicated that if a straight avoidance maneuver analogous to the F-16 flyup was used with medium-to-large UAV climb performance, the avoidance needed to be initiated many thousands of feet prior to reaching tall terrain features. Avoidance maneuvers induced by distant mountains would be considered an extreme nuisance if the operator intended to turn away from the mountains long before reaching them.

Based on that result, the design philosophy changed to incorporate the option for two turning avoidance maneuvers in addition to the straight maneuver. All three avoidance maneuvers were pre-defined to mimic typical medium-to-large UAV performance using the following autopilot command targets as requirements:

- Climb and descent rate: 1000 fpm;
- Bank angle: 40 deg;
- Roll rate: 50 deg/s; and
- Airspeed: 60 knots indicated airspeed (KIAS) (the middle of the DROID speed range).

These values were not selected to precisely model a specific UAV, but were selected to approximate the general capabilities of medium-to-large UAVs. The DROID was capable of much higher maneuvering levels in each axis.

The target airspeed of 60 KIAS was chosen somewhat higher than the airspeed for optimal turn radius of the DROID. This was primarily done to be more representative of the higher airspeeds typical for larger UAVs.

Trajectory Prediction Requirements

It was recognized early on that the existing simulation of the DROID was not very representative of the way the actual airplane flew. Updating the DROID simulation was considered out of scope for the SUAV Auto GCAS project, so the DROID simulation could not be used to help design the initial avoidance maneuver trajectory predictions.

Since the avoidance maneuvers were constrained to three basic autopilot commands (climb, bank, speed), a high-fidelity simulation model of the DROID was not necessary. All of the information necessary for trajectory predictions was obtained directly from preliminary flight-testing of the three avoidance maneuvers (left, straight, right). That process is described in more detail in the “SUAV Auto GCAS ‘Predict Avoidance Trajectories’ Module” section below.

The fundamental requirements for the trajectory prediction were to:

- Use a trajectory prediction divided into three basic phases: a time-delay phase, a capture phase, and a steady-state phase. These phases are discussed in more detail in the “Generic ‘Predict Avoidance Trajectories’ Module” section below.
- Use an adjustable time-delay phase to compensate for communications lag between components, and for changes to the basic frame rate.
- Use adjustable parameters for the capture phase to reflect actual aircraft responses, such as roll onset rate and g-onset rate.
- Use values during the steady-state phase that are sustainable by the aircraft and onboard autopilot.
- Allow parameters to be functions of other variables so they can be tuned for any flight condition dependencies observed during flight-testing.

In order to obtain the flight-test data necessary to tune the trajectory prediction, an additional fundamental requirement was to:

- Implement a method to initiate avoidance maneuvers based on a manual command, independent of terrain proximity.

Digital Elevation Model Data Requirements

Many military Auto GCAS implementations use no active terrain sensors, so no radio frequency emissions were required. This differs from many terrain-following systems that often relied on forward-looking radar or radar altimeters. The only sensors used for Auto GCAS typically had already been installed for other purposes.

The main SUAV Auto GCAS sensor was a standard blended GPS/INS as already installed on the DROID. Similar blended GPS/INS units had already become common on a wide variety of aircraft types. A GPS/INS that is already installed for navigation applications does not need to be considered part of the cost for an Auto GCAS. The main Auto GCAS utilization of GPS/INS data was to accurately locate the aircraft longitude, latitude, and altitude over a matrix of DEM data.

The intention from the beginning of this research project was that the project expand the concepts for using DEM data beyond the original F-16 implementation (as discussed in references 2 and 3). In particular, the intent was to demonstrate and evaluate the DEM data compression and re-rasterization methodology developed by NASA in support of early Auto GCAS tradeoff studies for another fighter aircraft. The fundamental requirements for using DEM data were to:

- Demonstrate the use of DEM source data from the National Elevation Dataset (NED). A description of the NED is contained in the section “Generic ‘Sense Terrain’ Module” below. Data from the NED had not been used on previous Auto GCAS projects.
- Demonstrate the use of existing NASA DEM data compression techniques to store DEM data for the entire Earth with vertical and horizontal accuracy adequate to support air-superiority fighter missions.
- Demonstrate the use of existing NASA DEM data lossless compression techniques to maintain high vertical and horizontal accuracy in the specific test areas to be used by the test aircraft.
- Demonstrate that the compressed DEM data can be used in a real-time aviation environment.
- Use a basic resolution for the re-rasterized DEM data of 3 arc-s (equivalent to the F-16).
- Develop appropriate scan techniques for lateral escape terrain avoidance maneuvers.

Flight-Test Analysis Support Requirements

In addition to the system requirements for the overall SUAV Auto GCAS design (including data recording), two additional features were implemented to support flight-test analysis. These included:

- Use of an off-the-shelf laser altimeter to provide an alternate source for AGL data; and
- Use an off-the-shelf high-definition tail camera to capture clear detail of terrain proximity during flight tests.

Flight-Testing as Part of the Design Process

Several DROID characterization flights and portions of the first SUAV project flights were used as inherent elements of the SUAV Auto GCAS design process. Therefore, the design impacts obtained from those flights are included in the following system overview descriptions for each module. The remaining

SUAV project flights were more focused on the intended Auto GCAS development and evaluation and are discussed in the “Test Results” section below.

Research System Overview

The SUAV Auto GCAS research project consisted of three primary elements: the DROID, the ground control van, and the Android™ platform smartphone hosting the Auto GCAS Algorithms. Additional background on the overall design can be found in the SUAV System Design Document (ref. 10).

Dryden Remotely Operated Integrated Drone (DROID) Test Aircraft

The DROID test aircraft was part of a fleet of four similar aircraft owned by NASA AFRC and operated by the AFRC Model Shop. The DROIDS have been used for a variety of purposes including pilot training, as research project test beds, and as part of student outreach programs (ref. 11).

The DROID was based on the Bruce Tharpe Engineering (Rogue River, Oregon) Modified Super Flyin’ King kit aircraft. The basic specifications were:

- Wing span: 9 ft 8 in;
- Total operating weight: 58 lb;
- Power rating: 11 HP;
- Maximum maneuvering speed (V_{max}): 80 KIAS; and
- Flight duration: 1.1 hr.

The SUAV Auto GCAS project used DROID 2 (fig. 2) which included the following project-specific modifications:

- Custom mounts for the Android™ platform smartphone;
- Laser altimeter (built by Latitude Engineering, Tucson, Arizona);
- High-definition video camera mounted on the tail;
- Extra batteries;
- Empennage stabilization support modification (wires were run from the vertical tail to the horizontal tail, as can be seen in fig. 2).



Figure 2. DROID research test aircraft.

During the SUAV Auto GCAS project, a pilot with direct line of site was always in control of the DROID during takeoff and landing. Once airborne, that pilot normally transferred control to a ground control operator (GCO) stationed inside the ground control van. Another aircraft control option was to operate the DROID in a manner similar to a remotely piloted vehicle (RPV) from a small cockpit inside the ground control van. A safety pilot using a handheld radio controller (fig. 3) had the responsibility of taking control if the Auto GCAS algorithms initiated unwarranted maneuvers.

On a few test missions during which the avoidance maneuvers were accomplished close to the ground control van, takeoff and landing was accomplished by the safety pilot. However, on most test missions the avoidance maneuvers were accomplished well away from the ground control van, so the safety pilot (with the handheld radio controller) needed to be much closer to the test location. In those situations another takeoff and landing pilot used a separate handheld controller that was wired directly to the ground station inside the ground control van. Once airborne, the takeoff and landing pilot typically transferred control to the GCO, who controlled the aircraft using standard DROID control options (mainly waypoints) to put the aircraft into holding patterns and set up the test maneuvers.

All of those control options were processed by the onboard Piccolo II autopilot (Cloud Cap Technology, Hood River, Oregon). The Piccolo II autopilot, shown in figure 4, measures 5.6 by 1.8 by 2.4 in and weighs 7.7 oz. The Piccolo II autopilot was used as an essential component, not just for basic aircraft control, but also as the primary source of input data for the Auto GCAS algorithm, the mechanism for responding to avoidance maneuver commands, and as a primary instrumentation source for post-flight data analyses. The Piccolo II autopilot communicated with a portable ground station located inside the ground control van using a 900-MHz, 1-W data link.



Figure 3. Safety pilot handheld controller.



Figure 4. Piccolo II autopilot.

Ground Control Van

The primary mission control for all SUAV Auto GCAS test missions was from the ground control van (fig. 5). A safety pilot (or the takeoff and landing pilot) was always ready to take direct control of the DROID for any reason, including loss of the communications link from the ground control van.



Figure 5. DROID and ground control van.

The interior of the ground control van included the following standard elements:

- A COTS portable ground station (fig. 6) provided the necessary link to the onboard Piccolo II autopilot through antennas set up on the ground control van.
- A laptop computer with COTS Piccolo command center (PCC) (Cloud Cap Technology, Hood River, Oregon) software installed (fig. 7) enabled the GCO to control the DROID. Control inputs were accomplished by a combination of keyboard and mouse commands.
- A small ground cockpit (fig. 8) provided the option for a research pilot to control the DROID as an RPV. The ground cockpit was assembled from a combination of COTS and custom components.
- Ultra-high frequency (UHF) and very high frequency (VHF) radio (for voice communications).



150043

Figure 6. Portable ground station.



150044

Figure 7. Piccolo command center (installed on laptop).



150045

Figure 8. Ground cockpit.

Additional ground control van elements installed in support of the SUAV Auto GCAS project included:

- A laptop computer with Auto GCAS user interface (UI) software installed; and
- Wind station equipment and laptop display.

The ground cockpit is shown in figure 8. The interior setup within the ground control van is shown in figures 9 (looking aft) and 10 (looking forward).



Figure 9. Ground control van interior (looking aft).



Figure 10. Ground control van interior (looking forward).

Physical Components of SUAV Auto GCAS

A simplified diagram of the essential Auto GCAS components for the smartphone-on-ground configuration is shown in figure 11(a). The components shown in blue were standard for the DROID and

ground control van. Once the DROID was airborne, the GCO normally entered control commands into the PCC laptop. Control commands typically directed the DROID to a series of waypoints in a predefined pattern.

The portable ground station provided a two-way command and control link between the PCC laptop and the Piccolo II autopilot on board the DROID.

The components shown in gold in figure 11(a) were unique to Auto GCAS. The UI laptop provided interaction with the Auto GCAS algorithms hosted on the smartphone. The UI laptop also provided the operator with the ability to set up the Auto GCAS mode states and algorithm options, along with displays for monitoring Auto GCAS status. The UI laptop implemented Auto GCAS avoidance maneuver commands through the portable ground station to the Piccolo II autopilot. The portable ground station basically acted as a pure feed-through in both directions. The UI laptop also provided a way to record data independent of the smartphone or the on-board Piccolo II autopilot.

A simplified diagram of the essential Auto GCAS components for the smartphone-on-aircraft configuration is shown in figure 11(b).

In addition to the smartphone installed on the DROID, a Gumstix® (Gumstix, Inc., San Jose, California) personal computer was needed to provide an interface between the USB connection on the smartphone and the RS-232 connection on the Piccolo II autopilot. The UI Auto GCAS avoidance maneuver processing and Piccolo UI software were ported to the Gumstix® personal computer, and a messaging interface was added to both components. In this configuration, all Auto GCAS avoidance maneuver decisions were made on board the DROID without the need for aircraft state data to be received on the ground and avoidance maneuver commands to be sent back up to the aircraft. The UI laptop still communicated with the smartphone through the portable ground station and Piccolo II autopilot to set up Auto GCAS mode state and algorithm options, and also provided the UI operator with displays by which to monitor Auto GCAS status.

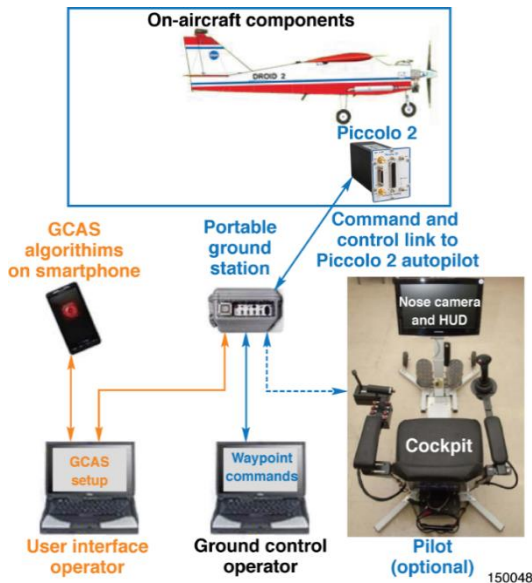


Figure 11(a). Phone-on-ground.

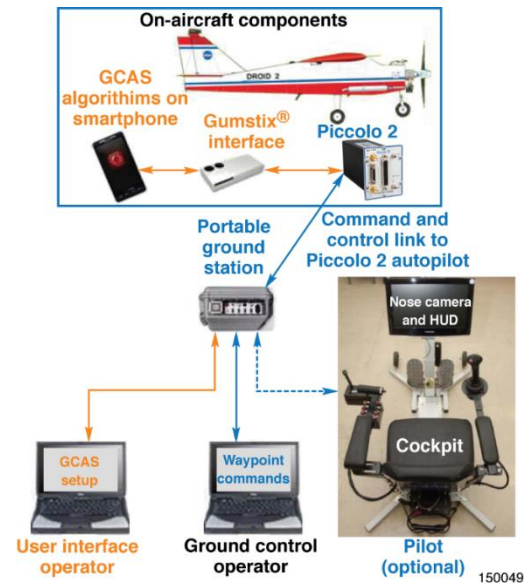


Figure 11(b). Phone-on-aircraft.

Figure 11. System diagrams.

Software Components of SUAV Auto GCAS

One of the key concepts for the overall ACAT program was to develop a modular architecture that could be applied to a variety of aircraft platforms. The F-16 FRRP Auto GCAS project began that process, and the SUAV Auto GCAS project further developed the modularity of each software component. A top-level diagram of the SUAV Auto GCAS modular architecture is shown in figure 12.

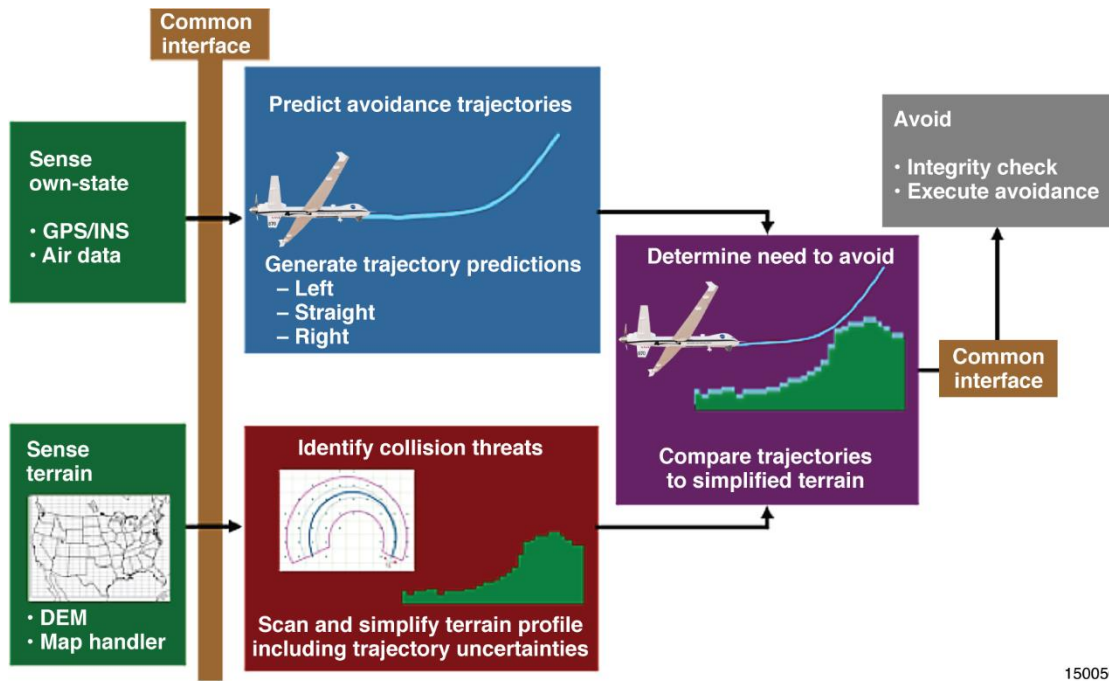


Figure 12. Top-level modular architecture.

A general description of the modular architecture is provided in this paragraph and more detailed functionality of each of the blocks is described in the following sections. The “Sense Own-State” module provided basic aircraft state information to the “Predict Avoidance Trajectories” module, which calculated the three-dimensional flightpaths for each of the three options for an avoidance maneuver (left, straight, and right). The “Sense Terrain” module generated a local map of digital terrain data from a much larger DEM. That process included decompression of the highly compressed data stored within the DEM. The “Identify Collision Threats” module scanned the digital terrain near each of the three trajectory predictions and converted the results into two-dimensional terrain profiles. The scanning process included uncertainties to account for the fact that the actual avoidance maneuver might not follow the predicted trajectory exactly. The “Determine Need to Avoid” module compared the three predicted trajectories with the associated two-dimensional terrain profiles. When the last of those three trajectories intersected with its terrain profile, an avoidance maneuver in that direction was communicated to the “Avoid” module, which then sent the appropriate commands to the autopilot. The “Common Interface” modules represent integrity checks that were accomplished at various points in the algorithm.

Due to the limited budget and shift to using Java™ as the programming language for the SUAV Auto GCAS code, the resulting implementation may not be as portable to other platforms as was originally intended. The basic functionality shown in figure 12 was implemented, but the code itself may not be as

easy to tailor to future applications as desired. The concurrent iGCAS project has developed a manual warning system for general aviation use, hosted on a smartphone, and has further matured the modular architecture and portability.

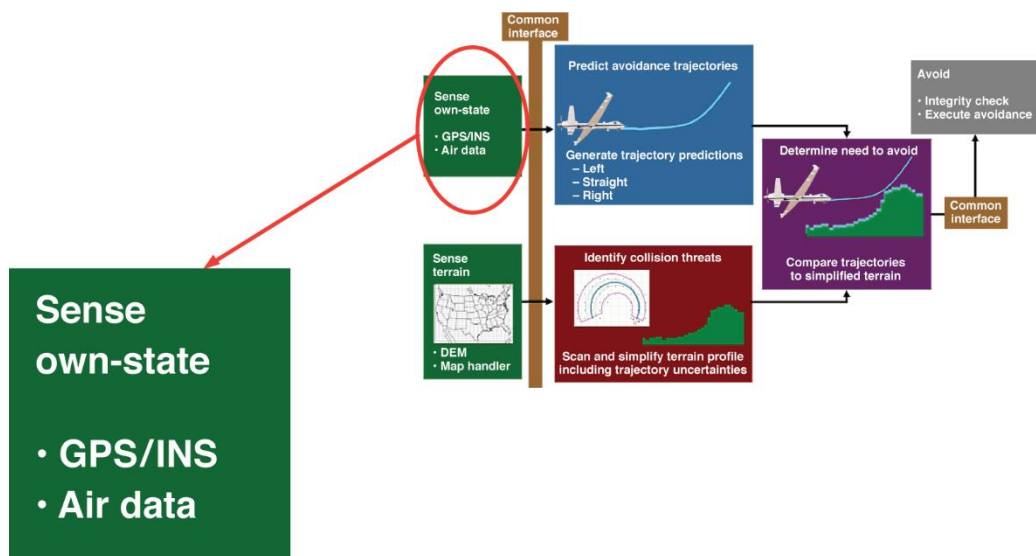
Some of the modules shown in figure 12 are fairly independent of the specific platform; other modules must be tailored for the characteristics of the intended platform. However, even the modules that require tailoring can use common design techniques and can build on the prior experience of similar platforms (fighters, UAVs, helicopters, transports, et cetera).

The following sections describe each module at a generic level and then more specifically as applied on the SUAV Auto GCAS project. Some of the generic module descriptions also provide an overview of techniques used on the F-16 FRRP (ref. 2) to show that Auto GCAS can be successfully implemented using varied methods.

The SUAV-specific modules were generally coded within the smartphone software, but a few elements (such as signal conditioning) were coded within the UI software. Future collision avoidance projects may choose to implement signal conditioning as part of the “Sense Own-State” module.

Generic “Sense Own-State” Module

The generic “Sense Own-State” module was a basic part of the modular architecture. Typically, the primary sources for these state values were outputs from a GPS/INS and an air data system, as represented by figure 13. The main purpose for this module was to provide enough information about the current state of the aircraft to support trajectory estimations far enough into the future to be useful for avoiding collisions.



150051

Figure 13. “Sense Own-State” module.

For high-performance aircraft, that extrapolation into the future could comprise several miles but might only cover 10 or so seconds. For lower-performance aircraft, that extrapolation could also need to be several miles (in order to ensure clearance over tall terrain features) but might cover more than 60 s. Those extrapolation techniques are covered in the discussion of the “Predict Avoidance Trajectories” module. This section limits discussion of the “Sense Own-State” module to the source data used to accurately locate the aircraft prior to the extrapolation.

It may be surprising that Auto GCAS does not need a dedicated sensor for altitude above ground level (AGL). Auto GCAS algorithms only rely on the GPS/INS altitude relative to a digital terrain map. Therefore, Auto GCAS does not require a radar altimeter or laser altimeter. This is a significant advantage for military aircraft, which need to minimize emissions that can reveal aircraft position to the enemy. Any aircraft needing Auto GCAS can also avoid the extra cost associated with installing a dedicated AGL sensor.

Since many modern aircraft have access to high-quality position information, this module was much less of a challenge than it had been in past decades. Embedded GPS/INS (EGI) capability has become standard on fighters and UAVs and is even becoming common on general aviation aircraft.

The primary GPS/INS outputs of latitude, longitude, and altitude are essential for any type of collision avoidance system. The accuracy of those outputs has evolved such that navigation uncertainty is no longer a significant impediment to collision avoidance systems.

Uncertainties in latitude and longitude can now be encompassed by horizontal errors of 50 ft or less. However, the remaining navigation uncertainty is still accounted for, as discussed in the “Generic ‘Identify Collision Threats’ Module” section below.

The vertical accuracy of many GPS/INS altitude outputs has also improved dramatically. Wide Area Augmentation System (WAAS) GPS/INS units can provide vertical accuracies within 40 ft. However, global positioning system/ inertial navigation system units can output altitude values in more than one reference frame. The most common are mean sea level (MSL) and World Geodetic System - 1984 (WGS84). For example, the F-16 used an MSL-based reference frame, whereas the Piccolo II autopilot on the SUAV Auto GCAS project used a WGS84 reference frame. Either altitude reference frame can work for collision avoidance systems, but care must be taken to ensure that all modules use the same reference frame. A factor for the vertical accuracy of the GPS/INS contributed to the generalized “built-in” buffer as discussed in the section “Generic ‘Determine Need to Avoid’ Module,” below.

Any production implementation of Auto GCAS on a low-performance aircraft (such as a UAV or many general aviation aircraft) will probably require some sort of density altitude state to use as an input to the trajectory prediction. The impact of non-standard day performance on the trajectory prediction could be very significant, particularly for operations near the service ceiling. In practice, a sensor for outside air temperature (OAT) may be sufficient, but each low-performance aircraft may have unique requirements.

Global positioning system / inertial navigation system units normally provide additional aircraft state information that is useful for collision avoidance systems. Euler angles, angular rates, and linear accelerations may all be important inputs to a collision avoidance system, depending on the platform. More of those terms would be needed for a high-performance aircraft such as a fighter; fewer terms may be sufficient for a lower-performance aircraft such as a UAV or transport.

Air data systems are another fundamental source of aircraft state information that might be needed as input data for collision avoidance systems. Some form of airspeed is probably important because it directly affects the maneuvering capability of any platform. The overall pitch and roll authority of a given platform will be a function of airspeed, as will the initial dynamic response to an avoidance maneuver command. Other air data terms such as angle of attack and sideslip may also be candidates for inputs to a collision avoidance system for some platforms, especially if improved-accuracy wind estimates are needed.

Many current-generation GPS/INS systems also provide a wind estimate by comparing inertial and air data sources. The need for wind as an input to a collision avoidance system will depend on the platform.

High-speed aircraft (such as fighters or commercial transports) may not need a wind estimate. Low-speed aircraft such as UAVs or general aviation aircraft may have a more significant need for a wind estimate.

As a rule of thumb, if the expected winds that may be encountered are less than 15 percent of the slowest airspeed for a given platform, wind estimates will probably not be needed and the remaining wind effects can be accounted for as part of other uncertainties. If the expected winds that may be encountered are greater than 30 percent of the slowest airspeed, it is likely that some sort of wind estimate will be needed. Those rule-of-thumb numbers are based on the F-16, which did not need wind estimates, and the SUAV, which could have benefited from improved wind estimates. Any wind expectations between 15 percent and 30 percent of the slowest airspeed on a given platform would need to be more thoroughly evaluated to determine whether or not a wind estimate is appropriate.

The preceding discussion of wind estimates only applies at the state of the aircraft leading up to avoidance maneuver initiation. Any wind changes that occur between the initiation of an avoidance maneuver and the termination of that maneuver will be more difficult to properly address. Most wind estimates depend on the aircraft being in relatively stable flight, and the accuracy of those estimates tends to degrade during maneuvering. Most wind estimates assume zero sideslip, but some sort of sideslip state sensor may help improve wind calculations. Wind shear as a function of altitude or horizontal position could have a significant impact on the repeatability of the overall collision avoidance algorithm.

Optimizing the use of wind estimates for collision avoidance systems is a significant technical area that still needs to be developed. The SUAV Auto GCAS project began that development, but more work is needed. Additional discussion of the SUAV experience with winds is described throughout this report.

Recommendation 1 (R1): Future Auto GCAS projects should consider applying resources to develop improved integration of in-flight wind estimates with the Auto GCAS algorithm.

Regardless of the input source being used, consideration should be given to what type of signal conditioning needs to be applied. The noise level inherent to individual parameters can dictate how much smoothing (if any) is needed. The most important outputs of a GPS/INS (latitude, longitude, and altitude) have most likely gone through a Kalman filter of some type, so a certain amount of smoothing can be expected in those source parameters. Other outputs, such as longitudinal acceleration and roll rate, may have high residual noise levels and an appropriately-designed smoothing algorithm may be warranted. The “SUAV Auto GCAS ‘Common Interface’ Module” section below contains additional discussion of signal conditioning as applied on the SUAV project.

A related consideration is the sample rate for each source of state data. Clearly, the sample rate needs to be fast enough to allow the ensuing trajectory predictions to keep up with ongoing aircraft motions. High-performance (or high-speed) aircraft will require a higher sample rate than low-performance / low-speed aircraft. If the computational capability of the intended platform is limited enough to force a low sample rate, the collision avoidance algorithm can compensate for that by increasing the time-delay phase of the trajectory prediction (discussed further in the “Predict Avoidance Trajectories” module sections below). However, there is a limit, and at some point that extra time delay could contribute to early avoidance maneuver activations and could be considered a nuisance. This is one of the motivations for a clearly-defined nuisance criterion on any collision avoidance project (as discussed in the “Available Reaction Time” section below).

SUAV Auto GCAS “Sense Own-State” Module

This section provides some insight into the choices that were made for the input parameters that fed the SUAV Auto GCAS algorithm. A summary of the fundamental input parameters follows:

Parameter	Source
• Latitude	Piccolo II GPS/INS
• Longitude	Piccolo II GPS/INS
• Altitude (WGS84)	Piccolo II GPS/INS
• Calibrated airspeed	Piccolo II air data system
• Bank angle	Piccolo II GPS/INS
• Dive angle (derived)	Piccolo II GPS/INS (based on climb rate and an estimate for true airspeed)
• True heading	Piccolo II GPS/INS

A summary of the secondary input parameters follows:

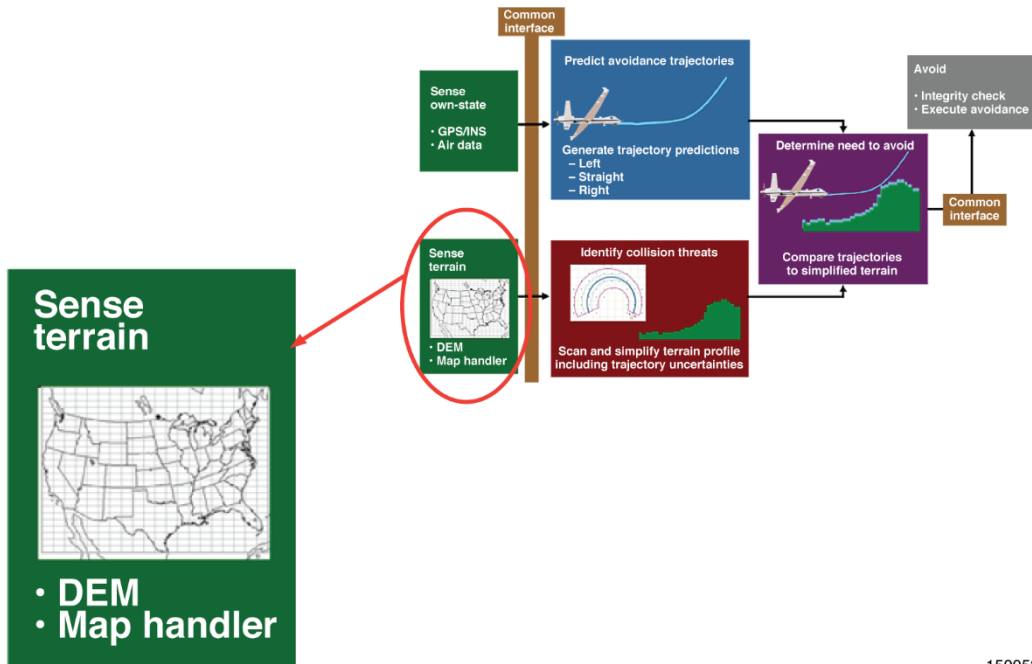
Parameter	Source
• Load factor	Piccolo II GPS/INS
• Roll rate	Piccolo II GPS/INS
• Wind speed (derived)	Piccolo II GPS/INS/air data
• Wind bearing (derived)	Piccolo II GPS/INS/air data

The fundamental input parameters were essential to proper operation of the SUAV Auto GCAS algorithm. The secondary input parameters were implemented in order to assess their importance and to evaluate the need for those parameters in similar applications.

None of the fundamental input parameters required special signal conditioning. Those parameters were adequate as they were output from the Piccolo II GPS/INS or air data system. During the early SUAV Auto GCAS design process it was decided that the only input parameters that required signal conditioning were the secondary input parameters roll rate and the wind values. The “SUAV Auto GCAS ‘Common Interface’ Module” section below contains additional discussion of signal conditioning for those parameters.

Generic “Sense Terrain” Module

Although this module was implemented as part of the SUAV Auto GCAS project, the resulting methods have much broader applications to any Auto GCAS project. In particular, very efficient techniques have been developed to utilize DEM at the core of this module. As represented in figure 14, the basic components of this module include DEM that cover a large enough region to support mission requirements, and a map handler to convert the very large regional map into local maps for the immediate area around the aircraft.



150052

Figure 14. "Sense Terrain" module.

An overall guiding principle was to provide a wide variety of users with the "best available" DEM for input to the Auto GCAS algorithms. In this context, "best available" meant high resolution and accuracy as well as the DEM source being widely accessible (lacking the classification restrictions of some military sources or the cost implications of some private sources).

After considerable study it was determined that the best widely-accessible DEM source for Auto GCAS applications was the NED produced by the United States Geological Survey (USGS). The NED provides horizontal resolutions of 1/3 arc-s (32 ft or better), and has an advertised vertical accuracy of 8 ft root mean square error (RMSE). The NED-based DEM are only available for the contiguous United States (CONUS) and parts of Alaska.

The second-best widely-accessible DEM source for Auto GCAS applications was the database resulting from the Shuttle Radar Topography Mission (SRTM), produced by the National Geospatial-Intelligence Agency (NGA) and NASA. The publicly available SRTM database provides horizontal resolutions of 1 arc-s (98 ft) for the United States and 3 arc-s over most of the Earth's landmass. This SRTM database has an advertised vertical accuracy of less than 32 ft for most areas within coverage. The SRTM database covers approximately 80 percent of the land mass of the Earth (between 60 deg north latitude and 56 deg south latitude). The NGA publishes the SRTM database in the standard Digital Terrain Elevation Data (DTED) format, which is a format different than that of the NED.

Several additional DEM sources provide coverage of the remaining areas of the Earth, but often at lower resolutions and degraded vertical accuracy. One of those sources has been labeled "Legacy" DTED to distinguish from SRTM DTED. Legacy DTED is also available from the NGA and has been a common source for terrain data prior to SRTM. Although Legacy DTED was a useful source in its time, it contains many discontinuities across longitudinal-latitude boundaries and localized artifacts that can result in vertical errors of hundreds of feet (in some cases there exists over a thousand feet of vertical error).

The goal of the SUAV Auto GCAS “Sense Terrain” module was to combine the best DEM sources for a given area into a single contiguous dataset for the entire Earth. Therefore, the NED was used for the CONUS, the SRTM database was used for most of the remaining land mass of the Earth, and other sources were used for landmasses not covered by the NED or SRTM.

A related goal was to demonstrate advanced DEM compression and re-rasterization concepts in a practical flight-test environment. Part of that goal included demonstrating how the levels of DEM accuracy could be customized in different areas based on the mission needs of the aircraft.

As an example, an air-superiority fighter mission does not require extensive operations at low altitudes. Therefore an Auto GCAS could be developed without the need for high-accuracy DEM over most of the surface of the Earth, over which the aircraft would primarily be in transit at high altitudes. However, to avoid nuisance Auto GCAS activations, a higher level of DEM accuracy would still be required for the terrain within the airspace around typical operating bases.

It was also recognized that Auto GCAS would become much more practical for many platforms by completely avoiding the need to load a specific regional DEM for each mission. In order to achieve all of the goals listed above, NASA developed numerical techniques to generate Compressed Digital Terrain Maps (CDTM) while maintaining minimal loss of accuracy.

The generic example CDTM installed for SUAV Auto GCAS flight-testing achieved a compression ratio of approximately 2500 to 1. That level of compression enabled DEM for the entire Earth to be reduced in size from 400 GB to 170 MB. This was primarily accomplished using a combination of two numerical methods that were well-suited to compress large amounts of terrain elevation data. These two methods were known as “tip-tilt” and “semi-regular tree networks.”

The combined mathematical methods nested a tip-tilted facet (sloped tile) fitting method within a recursive subdividing algorithm. The method began with a single very large facet of 1 deg x 1 deg (60 nm x 60 nm at the Equator) and used linear regression to fit the sloped tile to the terrain data underneath it. If the specified worst-case vertical or horizontal accuracy were exceeded, that facet was subdivided. The overarching recursive process continued subdividing the area until the maximum error between all of the facets and terrain data underneath them satisfied the targeted error tolerance over the entire cell. With this approach, a large global regular network of 1 deg x 1 deg cells (divided along major latitudes and longitudes) was produced as a database, where each 1 deg x 1 deg cell contained a semi-regular network of free-edged tip-tilted facets of varying size covering the entire cell.

An example of the tip-tilt concept is shown in figure 15. The enormous advantage of the tip-tilt technique is that it more closely matches natural terrain, as opposed to typical DEM flat, rectangular polygons arranged in a fixed grid (fig. 16). As can be seen, the tip-tilt facets tend to have errors at the corners. These corner errors can appear large compared to neighboring facets, but they are actually much smaller overall than similar errors that are induced along the edges of the flat rectangular DEM polygons.

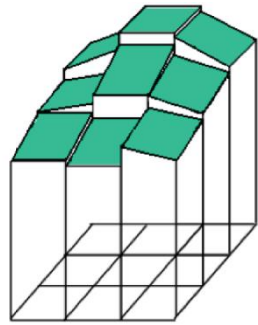


Figure 15. Example of tip-tilt concept.

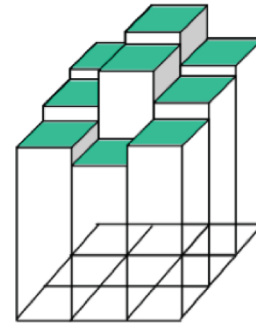


Figure 16. Example of typical DEM data.

The semi-regular tree compression technique allowed CDTM to be generated using facets that did not need to be arranged in a consistent “n x n” grid. Therefore, a single facet could represent many square miles over the Great Plains of the central USA. Many more facets were needed to cover the Rocky Mountains, but the sloped facets still fit those natural features much better than did flat rectangular polygons. The result was a dataset that was much smaller in size than would otherwise have been obtainable.

The main product of the NASA CDTM development effort was a very flexible software package that can customize a CDTM to meet the requirements of any Auto GCAS project. That product is called the Global Elevation Data Adaptive Compression System (GEDACS). NASA is in the process of applying for a patent for the GEDACS process (ref. 12) and the software will be available for public use via NASA license. Figure 17 shows an example GEDACS output representing the combined application of the tip-tilt technique and the use of semi-regular tree networks.

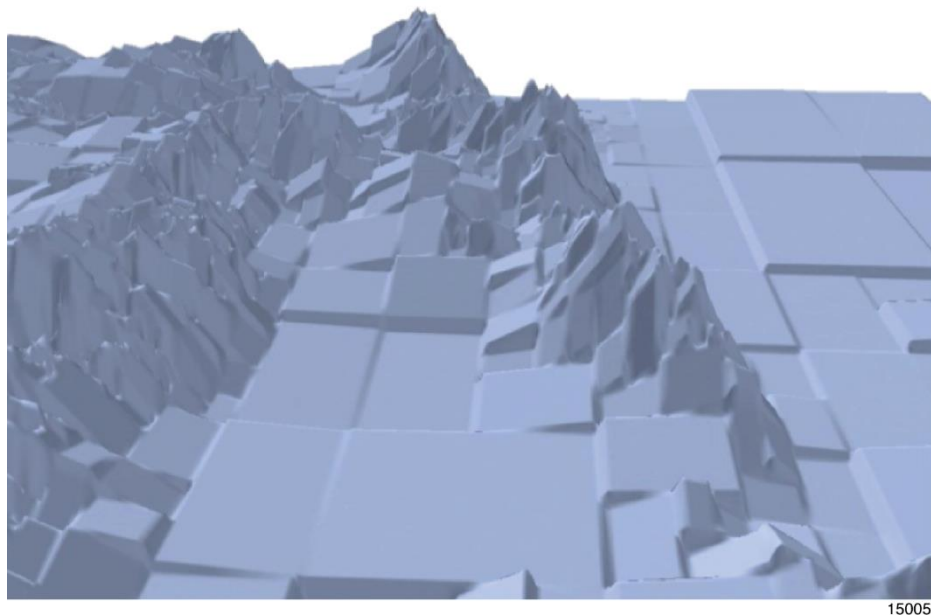


Figure 17. Example of GEDACS CDTM product.

The GEDACS was also designed to provide improved vertical accuracy in particular areas where a given platform requires low-altitude operations, such as near airfields or along low-altitude training routes. The GEDACS provides plenty of flexibility to meet the individual requirements of a given platform while minimizing storage space.

In the case of the generic example of worldwide CDTM used for SUAV Auto GCAS, the worst case for either vertical or horizontal accuracy was specified to be 500 ft anywhere on the Earth. A vertical accuracy of 500 ft would probably provide adequate Auto GCAS functionality for an aircraft that does not need to spend much time at low altitudes other than for takeoff and landing. Therefore, the SUAV Auto GCAS CDTM with worldwide coverage could be applicable to high-altitude platforms including many UAVs, airline transports, and even fighter aircraft that do not have a requirement for low-altitude strike missions.

Once the generic example of worldwide CDTM had been generated, the SUAV aircraft was used as a testbed to demonstrate that this CDTM data could be accessed and used in a real-time flight-test environment. A key part of that demonstration included the ability to re-rasterize the CDTM in real time so it could be accessed by Auto GCAS using a typical rectangular grid in the local area around the aircraft. That re-rasterization logic was not a trivial design effort. It required interpretation of data that could represent many square miles or a few hundred square feet, followed by conversion of those data into a rectangular grid.

Since any aircraft equipped with Auto GCAS will constantly be moving from one DEM area to another, a map handler was developed. The map handler was designed to add new local maps as the aircraft moves forward, and to remove local maps covering areas behind the aircraft (as those maps are no longer needed).

SUAV Auto GCAS “Sense Terrain” Module

In addition to the generic example of worldwide CDTM, local areas were also customized for SUAV flight-test operations. These higher-accuracy portions of the CDTM encompassed most of the Edwards Air Force Base (EAFB) restricted area, including much of the extreme terrain in the southern Sierra Mountains, and were considered nearly “lossless” compared to the NED source data. The large EAFB restricted area also included the three SUAV test areas: the northern part of Rosamond Lakebed, the North Base runway area at EAFB, and the GCAS Valley (near Fremont Peak). Only the GCAS Valley test area is discussed in detail in this report. The other two test areas were primarily used for functional checkout flights and preliminary flight tests.

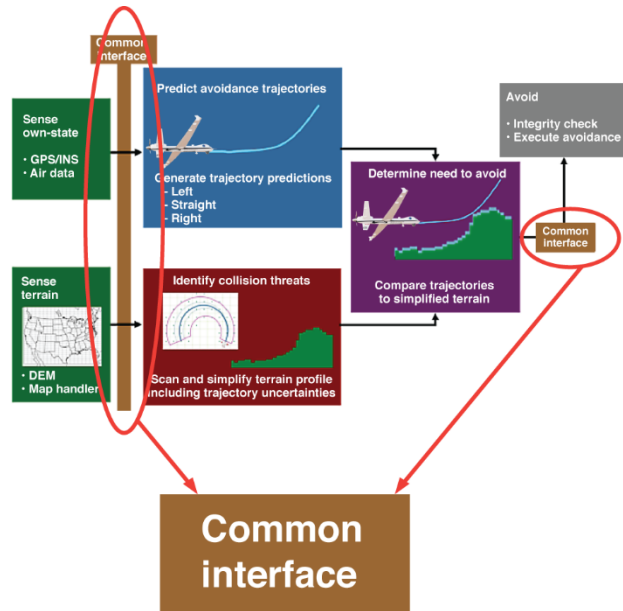
The EAFB restricted area map was generated by GEDACS using a worst-case allowable threshold of 20 ft. That small error value resulted in GEDACS selecting the minimum facet size throughout the restricted area (the minimum facet size had been specified as approximately 3.5 arc-s on each side). Using these techniques helped maintain the CDTM vertical accuracy in the flight-test areas at approximately 10 ft or better compared to the NED source data.

In order to obtain an independent source for the key terrain features in the GCAS Valley area, representatives from NGA stationed at EAFB were requested to conduct a ground survey. The results of that survey are discussed and compared with the customized CDTM data in the “Ground Survey at GCAS Valley Test Site” section of this report, below.

Generic “Common Interface” Module

The primary function of the generic “Common Interface” module (fig. 18) was to communicate all of the input parameters needed by the Auto GCAS algorithm. This module also acted like an “input voting

plane.” The intent was to accomplish integrity management checks on the incoming data to ensure that appropriate values were fed to the Auto GCAS algorithms. However, those inputs were typically only single-string, so there were no redundant inputs to compare and voting logic could not be applied. Therefore, the main goal was to ensure that the incoming data were not stale or otherwise corrupted.



150056

Figure 18. “Common Interface” module.

One of the general techniques used was heartbeat checks. Heartbeat checks have been typically applied between hardware components. The sending component includes a heartbeat signal as part of its output data structure, and the receiving component checks to confirm that the heartbeat changes in the expected way. If the received heartbeat does not change, a FAIL state is set to inhibit Auto GCAS avoidance maneuvers.

Another type of check used was called a “reasonableness check,” and was used to ensure the incoming data were within a reasonable range of values. For example, altitude inputs can be checked for values that are considered excessively high or excessively low for the expected operations of an aircraft. If an altitude input for an SUAV had a value of 50,000 ft, those data most likely would be corrupt and should not be used in the avoidance maneuver decision logic.

Care must be taken to not over-design reasonableness checks. It is very easy for designers to come up with a wide range of creative checks that may not function as intended with real-world data. Noisy input data can be particularly troublesome if reasonableness checks are applied that use differentiation and therefore amplify the noise. Noise amplification can lead to inappropriately setting FAIL states and inhibit avoidance maneuvers when there is nothing that is actually wrong with the input data stream.

It is also advisable to incorporate integrity management checks that are self-recovering. If the conditions that induced an invalid check are no longer true, the FAIL state should be removed automatically and allow Auto GCAS avoidance maneuvers to resume as needed.

The F-16 Auto GCAS implementation also used a number of interlocks. Interlocks prevented avoidance maneuver commands from being initiated when outside of the intended design envelope (that is, too slow,

gear down, aerial refueling door open, excessive uncertainty in the navigation system accuracy, excessive angle of attack, relevant failure modes, et cetera).

Developmental flight-test experience with the F-16 Auto GCAS has indicated the need for another monitor to check for corrupted digital terrain data. Although several methods might be used to accomplish this check, the goal would be to ensure that the DEM loaded into the aircraft has not been corrupted on the data cartridge since the time it was downloaded from the original source data (that is, via a mission planning process). The F-16 Auto GCAS designers implemented a checksum for groups of digital terrain data within the overall dataset. If the expected checksum for a given group did not match the checksum from the source data, Auto GCAS avoidance maneuvers were inhibited. After implementation, this method worked very well to isolate and protect against nuisance activations in areas with corrupted terrain data.

Recommendation 2 (R2): Future Auto GCAS projects should consider implementing a monitor to isolate and protect against corrupted digital terrain data.

Recent experience during operational testing of the F-16 Auto GCAS indicated the need for another type of integrity management monitor. The F-16 Auto GCAS implementation depended upon a flyup request bit that was communicated from the single-thread avionics system to the quad-redundant flight control system. No problems were observed during several hundred flights of developmental testing. However, on one sortie during operational testing it was determined that an unusual multiplexer (mux) bus contention problem induced by one of the remote terminals on the mux bus could lead to a false flyup request bit.

Recommendation 3 (R3): Future Auto GCAS projects should consider implementing a monitor to protect against false requests for avoidance maneuvers.

Although Recommendations 2 and 3 were primarily based on experience with Auto GCAS as implemented on the F-16, similar hypothetical situations could also have occurred with the SUAV implementation. The resultant lessons learned will still have important significance for future Auto GCAS projects.

SUAV Auto GCAS “Common Interface” Module

The SUAV Auto GCAS implemented integrity management checks in several different ways. These methods were heavily influenced by the hardware configuration (refer to figs. 11(a) and 11(b)).

The smartphone hosted all of the Auto GCAS-specific algorithms. The smartphone also hosted a limited set of integrity management checks that were only applied to incoming data from the UI (either the laptop computer or the Gumstix® personal computer). After initial power-up, the smartphone was ready and waiting for incoming data from the UI. The actions taken by the smartphone depended upon the mode state, as discussed in the “SUAV Mode States” section below.

The UI hosted the integrity management checks. When the smartphone was on the ground, these checks were hosted on the UI laptop computer. When the smartphone was on board the aircraft, these checks were hosted on the Gumstix® personal computer.

Regardless of the location of the smartphone, the UI integrity management checks were applied for two sets of incoming data. One set of checks was applied to incoming data from the smartphone. The other set of checks was applied to incoming data from the Piccolo II autopilot (as communicated through the Piccolo II autopilot ground station when the smartphone was on the ground).

The three basic types of SUAV Auto GCAS integrity management checks were:

- Signal conditioning,
- Exception handling, and
- Connectivity and throughput checks.

Each type of integrity management check is discussed in the following sections.

Signal Conditioning

Preliminary flight-test data indicated that the noise levels of the Piccolo-II-autopilot-calculated winds and sensed roll rate could skew the performance of the Auto GCAS algorithm if not smoothed. Therefore, an attempt was made to apply moving averages to the relevant outputs from the Piccolo II autopilot.

Those moving averages were applied within the UI software to avoid adding computational load to the smartphone software. When the smartphone was on the ground, the UI smoothing was hosted on the UI laptop computer. When the smartphone was on board the aircraft, the UI smoothing was hosted on the Gumstix® personal computer.

The implementation of the various smoothing algorithms was considered a low priority during the design process for this flight-test demonstration project. The end result was that a moving average was not applied to the wind values as input to the Auto GCAS algorithm, only to some of the wind parameters available in the post-flight data. Therefore, the Auto GCAS avoidance maneuver decisions were made solely on the instantaneous wind values provided as output from the Piccolo II autopilot.

A moving average smoothing algorithm was implemented on the roll rate input as intended. An error was found in the roll rate averaging affecting the data that were used to derive algorithm parameters. The software problem was corrected; however, given a lack of funding and time to re-evaluate and correct the affected algorithm coefficients, the decision was made to simply zero out the roll rate input to the Auto GCAS algorithm because of its status as a secondary parameter.

Based on early flight-test results it was decided that the load factor output from the Piccolo II autopilot did not require smoothing. The load factor noise was typically associated with engine vibration and was at a high enough frequency not to interfere with the Auto GCAS. However, later flights were conducted in a more turbulent environment, in which the load factor variations due to turbulence could easily exceed ± 0.5 g. Because the maximum Auto GCAS command capability was established to emulate a medium-to-large UAV with a 40-deg bank angle limit, the maximum sustained load factor was approximately 1.3 g. That limit placed the entire normal maneuvering envelope of the test aircraft roughly between 1.0 and 1.3 g, which was overshadowed by the turbulence response of ± 0.5 g or greater.

Because the Auto GCAS avoidance maneuver decisions were made solely on the instantaneous load factor values output from the Piccolo II autopilot, the timing of the initiation was particularly susceptible to turbulence effects. In hindsight, it would have been better to also include a smoothing algorithm for the load factor before it was input to the Auto GCAS algorithm in order to minimize the potential for inappropriate activations due to turbulence. Another option would have been to remove the load factor as an input to Auto GCAS because of the poor signal-to-noise ratio.

The various ways to implement and optimize Auto GCAS-related smoothing algorithms is one of the remaining technical issues that warrant additional research to provide future projects with a more solid foundation. Each platform will undoubtedly require some unique smoothing concepts, but there may also be generic smoothing concepts that could apply across multiple parameters on multiple platforms.

Recommendation 4 (R4): Future Auto GCAS projects should pay special attention to the input signal conditioning necessary for that particular implementation.

Exception Handling

The primary purpose for exception handling checks was to identify suspect data and automatically adjust those data as needed to avoid negative impacts to the Auto GCAS algorithm. Several types of exception handling checks were hosted in the UI software, including:

- Divide-by-zero checks;
- Valid-range checks; and
- Off-map checks.

In general, if these checks identified suspect data, the result led to a FAIL state. These exception handling checks functioned adequately with no negative impact to the flight-testing process.

Connectivity and Throughput Checks

The primary purpose for connectivity and throughput checks was to confirm that the communication between components was proceeding as required. These checks were hosted in the UI software and in the smartphone software. In general, if the continuity and throughput checks determined that communication was not proceeding as required, the result led to a FAIL state. Examples of connectivity and throughput checks included:

- Stale data checks, and
- Packet counting checks.

These checks led to FAIL states on a number of flight tests. The checks correctly identified communication problems between components, but the ensuing action taken after setting the FAIL state evolved over time. Additional discussion is provided in the “SUAV Mode States” section below.

The connectivity check “timeout value” was changed as a result of flight-testing. The original timeout value was selected to be 0.5 s to ensure that the FAIL state would be entered before the aircraft had traveled far enough to miss an entire re-rasterized terrain tile (a 3-arc-s rectangle). When flight-test telemetry dropouts were inducing excessive FAIL states, the timeout value was increased from 0.5 s to 5.0 s to provide an opportunity for the telemetry to improve before aborting a flight-test run. The 5-s timeout value was acceptable in flight-testing because a safety pilot was always in position to take control. A timeout of 5 s would probably be too large for a production implementation of Auto GCAS.

Generic “Predict Avoidance Trajectories” Module

This module requires the greatest amount of customization for the dynamic response of each specific platform. However, that customization does not require as much effort as might be expected. Figure 19 shows the overall concept for this module. The solid blue line within the “Predict Avoidance Trajectories” module on figure 19 represents the predicted trajectory if an avoidance maneuver were initiated at that instant (that is, the blue line represents Altitude versus Range from the initiation point).

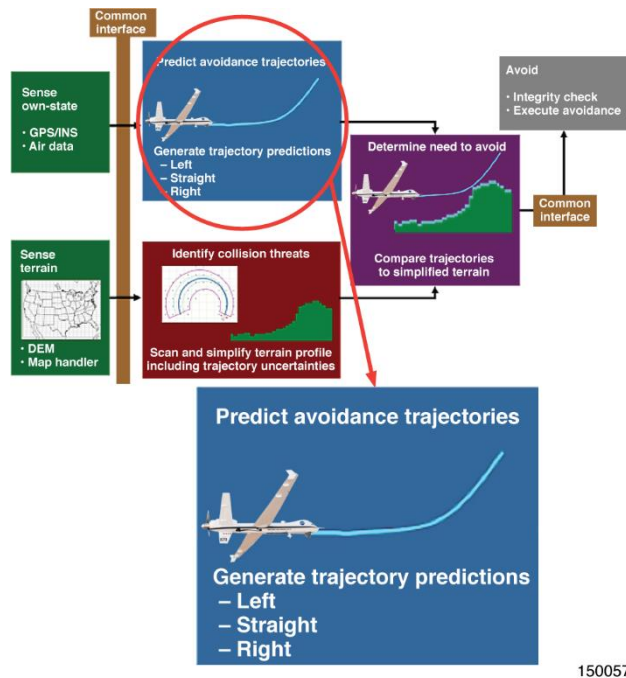


Figure 19. “Predict Avoidance Trajectories” module.

The first basic decision that must be made for every new Auto GCAS design is to define the specific avoidance maneuvers that will be implemented. The selected avoidance maneuvers must be appropriate to the type of mission that the platform conducts and will be directly influenced by the maneuvering capability of the aircraft.

An important Auto GCAS design philosophy is to choose avoidance maneuvers that are as aggressive as a pilot would be in a situation that requires immediate action to avoid hitting the ground. Therefore, Auto GCAS avoidance maneuvers should utilize a large portion of the maneuvering capability that is available at a given set of flight conditions. Avoidance maneuvers that do not utilize a large portion of the maneuvering capability could lead to nuisance activations. However, avoidance maneuvers that might exceed aircraft structural limits or cause the aircraft to run out of airspeed before the avoidance maneuver is complete should not be used unless given careful consideration.

A common design philosophy that was discussed early in the design process on several Auto GCAS projects can be paraphrased as, “It’s acceptable to exceed structural limits during an Auto GCAS avoidance maneuver if that helps an aircraft avoid impact with the ground.” However, any avoidance maneuver designed using that philosophy could not be flight-tested without risk of exceeding those structural limits. Great care would be needed to avoid conditions that might lead to an over-g; despite the best efforts of the test team, an over-g might still occur unless those boundaries are very well-defined and the aircraft response is especially predictable and controllable.

Another common design philosophy that was discussed on several Auto GCAS projects but has not yet been implemented successfully can be paraphrased as, “Even if an aircraft might run out of airspeed during an Auto GCAS avoidance maneuver, go ahead and let it try, since that would be better than hitting the ground.” That design philosophy might work reasonably well at some conditions, but the ability to generate useful trajectory predictions becomes less reliable at the high angle of attack associated with low airspeed maneuvering. The overall result of trajectory prediction uncertainty could lead to excessive nuisance activations. The design team on each Auto GCAS platform will need to decide, “How slow is too slow?”

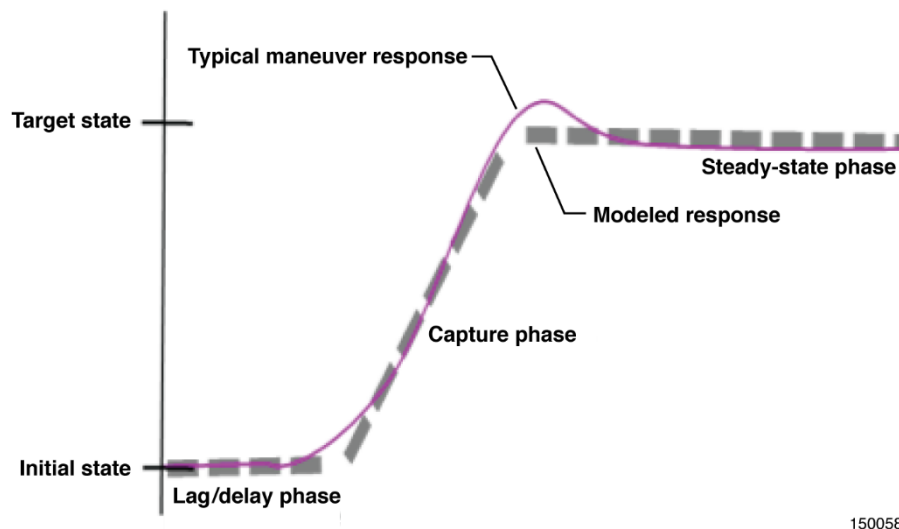
In the case of the high-performance F-16, a single basic flyup maneuver was used. As described earlier, a simplified description of the F-16 flyup maneuver was “roll toward wings-level and pull at 5 g until clear of terrain.” That simple description covers the essential F-16 flyup maneuver although there were several important modifications dependent on steep dives, high roll rates, et cetera. The F-16 typically had enough kinetic energy to convert into potential energy to allow successful flyup maneuvers straight over all but the tallest terrain features.

In the case of a relatively low-performance aircraft like the DROID (or, rather, the medium-to-large UAV it was constrained to emulate), it was not practical to expect successful avoidance maneuvers directly over typical terrain features. In particular, given the very limited sustained climb performance of medium-to-large UAVs (not much more than 1000 fpm), avoidance maneuvers that needed to fly over large hills or mountains would need to be initiated many thousands of feet prior to that feature. Those “straight” avoidance maneuvers would be considered an extreme nuisance by the pilot of a medium-to-large-UAV who intended to turn away from a terrain feature long before reaching it. Therefore it was necessary to include trajectory predictions for turning maneuvers to either side.

Recommendation 5 (R5): Future Auto GCAS projects should consider incorporating multiple trajectory predictions to provide more than one option and to reduce nuisance potential.

A full six-degrees-of-freedom simulation is not required to model the trajectory predictions for this module. High-fidelity aerodynamic, thrust, and flight control models are not required, although a simulation using those models can provide a very helpful starting point if available. If a high-fidelity simulation is not available, the necessary parameters can be determined directly from a few simple flight tests.

For the purpose of Auto GCAS, it is only necessary to model the particular response that is expected during an avoidance maneuver. Since the avoidance maneuver itself is usually very simple, the corresponding models needed in order to predict the avoidance maneuver trajectory can also be very simple, as sketched in figure 20.



150058

Figure 20. Simple kinematic state model.

This methodology models the magnitude and orientation of the lift vector during the avoidance maneuver. The magnitude of the lift vector can be modeled by approximating the change in either the load factor or the climb rate from the initial state during the transition (the capture phase) to a steady-state target, and can continue into the steady-state phase as needed. Similarly, the orientation of the lift vector can be modeled based on the initial bank angle and flightpath angle; the change in that orientation can be approximated during the transition to the target attitudes, and the resulting orientation can continue into the steady-state phase.

As an example, a fundamental portion of most Auto GCAS avoidance maneuvers is a bank angle capture. Given the initial bank angle of the aircraft, a lag/delay phase is used to account for any length of time in which no significant change in bank angle occurs (as observed on simulated or flight-test avoidance maneuvers). The lag/delay phase can account for the basic dynamics of the overall system and can also be used to add a term to account for Auto GCAS implementations that need to use a lower frame rate. As an example, the SUAV roll rate lag term was 0.2 s as determined by inspecting flight-test time histories from early characterization flights. Using data from the same SUAV characterization flights, the airspeed lag term was selected to be 0.24 s and the vertical velocity lag term 0.3 s.

A simple roll rate onset term can be used during the capture phase, until the bank angle approaches the target value. Individual aircraft dynamics may result in overshoots of the target bank angle, but, if small enough, that overshoot may not need to be modeled as part of the trajectory prediction. As the target bank angle is being approached, the roll rate model is attenuated and bank angle is maintained during the steady-state phase until the end of the avoidance maneuver. The bank angle lag/delay term and the roll rate onset term may need to be scheduled as a function of flight conditions. It is unlikely that those terms would remain constant across a wide airspeed range.

Similar simple kinematic models can be used for the other axes that represent an avoidance maneuver. A higher-performance aircraft (such as the F-16) may use load factor lag/delay and g-onset as pitch axis terms. A lower-performance aircraft, such as a UAV, may use climb rate instead of load factor. An aircraft with an auto-throttle may also use thrust command or airspeed command as part of the simple kinematic model. Without an auto-throttle, it is likely that airspeed changes during the avoidance maneuver would need to be modeled.

Each axis of the kinematic model is combined into simplified equations of motion to assemble the overall trajectory prediction that will be used in the modules that follow. A more detailed description of the overall process is contained in reference 9.

Given the simple nature of the kinematic model, it can be easily tuned as conditions are observed that don't match the original model. Excessively complicated trajectory prediction models can actually be harder to tune, take a lot of processor power, and may not provide much improvement in performance. "Keeping it simple" is a worthy trajectory prediction goal for any Auto GCAS project.

Since no aircraft model is perfect, the trajectory prediction should try to account for any uncertainties that may not match the modeled trajectory. Those uncertainties can be due to un-modeled effects of gross weight, off-standard day performance, local winds, et cetera. However, those uncertainties are usually secondary or tertiary effects that can be captured with a few extra simple terms that encompass the worst-case effects, and should not require complicated additions to the model. Compensation for these uncertainties can be modeled either by specific terms or generalized terms.

An example of a specific uncertainty term is the use of on-board local wind calculations to capture the expected variation due to the magnitude of the headwind, tailwind, or crosswind. Naturally, the ability to account for local winds is only as good as the quality of the onboard calculation. An alternative is to assume

a worst-case magnitude and direction, and include a single term to capture that effect as part of the trajectory prediction. That worst-case assumption could be set up differently for left or right trajectory predictions (that is, crosswinds from opposite sides).

Generalized uncertainty terms can also capture the overall range of uncertainty for a given platform. Monte Carlo simulations are one way to define that overall range of uncertainty. Another way is to use the variations observed across a number of flight-test avoidance maneuvers. Those generalized prediction variations can be applied as a horizontal uncertainty, a vertical uncertainty, or both. These generalized variations can be represented by a standard deviation and can be either positive or negative, so it is recommended to account for those variations as part of a root-sum-square implementation in the “Determine Need to Avoid” module, as discussed in the section “Generic ‘Determine Need to Avoid’ Module,” below.

The F-16 Auto GCAS implementation did not include an auto-throttle, so most trajectory predictions were based on the assumption that the throttle was not moved. Some F-16 trajectory predictions for flyups that were initiated at low airspeeds were based on the assumption that the pilot was aware of the low-speed condition (which was also annunciated on the head-up display), and would increase throttle as needed.

The F-16 trajectory predictions were developed by Lockheed Martin Aeronautics using high-fidelity simulations of the F-16. However, each F-16 Auto GCAS flight test project also verified the accuracy of the F-16 trajectory predictions using a flight-test technique that took advantage of the Pilot Activated Recovery System (PARS). The F-16 PARS was already part of the overall system design to provide the pilot with an option to initiate an automated recovery in spatial disorientation situations (one of the factors that has led to CFIT mishaps).

Since the F-16 PARS maneuver was designed to be identical to an Auto GCAS flyup maneuver, the use of PARS as a flight-test technique was also very effective as a quick way to obtain trajectory data. The PARS allowed the test aircraft to be established at a wide range of initial flight conditions, and then an Auto GCAS flyup maneuver to be initiated without the need to be in close proximity to terrain. Although Auto GCAS maneuvers could be initiated farther from terrain by using a flight-test altitude buffer, that test technique still required maneuvering the test aircraft into a particular orientation over a specific piece of terrain. The PARS maneuvers allowed the same trajectory data to be obtained independent of the location relative to terrain and were, therefore, much more efficient.

SUAV Auto GCAS “Predict Avoidance Trajectories” Module

In the case of the SUAV project, a low-fidelity simulation was available, but it was clear that it did not represent the actual flight characteristics of the DROID. Even if a high-fidelity, high-confidence simulation is available, it is recommended that a small number of flight tests be accomplished to obtain data directly from a representative set of planned avoidance maneuvers.

All of the necessary SUAV trajectory prediction parameters were determined from six preliminary flights totaling 4.0 flight hours. This was accomplished by executing a small matrix of avoidance maneuvers initiated by the user interface operator (UIO) (similar to PARS) to capture the range of variations that would be expected during the planned flight-test missions. The SUAV avoidance maneuvers were defined to take advantage of the standard capability of the Piccolo II autopilot. The basic autopilot functions used were:

- Bank angle command,
- Vertical rate command, and
- Indicated airspeed command.

The SUAV avoidance maneuvers were defined as:

- Straight:
 - Bank to wings-level;
 - Capture 1000 fpm climb; and
 - Capture 60 KIAS.
- Left and right turns:
 - Bank to 40 deg left or right,
 - Capture 800 fpm climb, and
 - Capture 60 KIAS.

The 40-deg bank target and the 1000-fpm climb rate target were selected to represent reasonable values for medium-to-large UAVs. The DROID was capable of a much higher bank and climb rate. An 800-fpm climb rate target was selected for the left- and right-turning avoidance maneuvers to mimic reduced aircraft performance when some of the excess power was used for turning instead of climbing (800 fpm is roughly 1000 fpm times the cosine of the 40-deg bank used for turning avoidance maneuvers).

During the preliminary flights, it was observed that avoidance maneuvers tended to achieve a slightly lower climb rate than the value commanded by the autopilot. Although this bias could have been explained by a number of different factors, it was easy to account for the bias in the trajectory prediction. A simple multiplier of 90 percent was incorporated in the trajectory predictions, resulting in 900 fpm instead of 1000 fpm for the straight prediction, and 720 fpm instead of 800 fpm for the turning predictions. The result was that avoidance maneuvers were initiated slightly earlier but better reflected the actual climb performance of the aircraft.

The DROID tended to change from one climb rate to another very quickly. Therefore the dynamic “capture” phase (when the aircraft was pulling up at more than 1 g) was over very quickly and the trajectory prediction reflected that short dynamic phase followed by a long steady-state phase. In contrast, an F-16 flyup could remain at 5 g for an extended period (more than 5 s after initiating from a steep dive) and the airplane was often clear of terrain before transitioning to a steady-state climb rate.

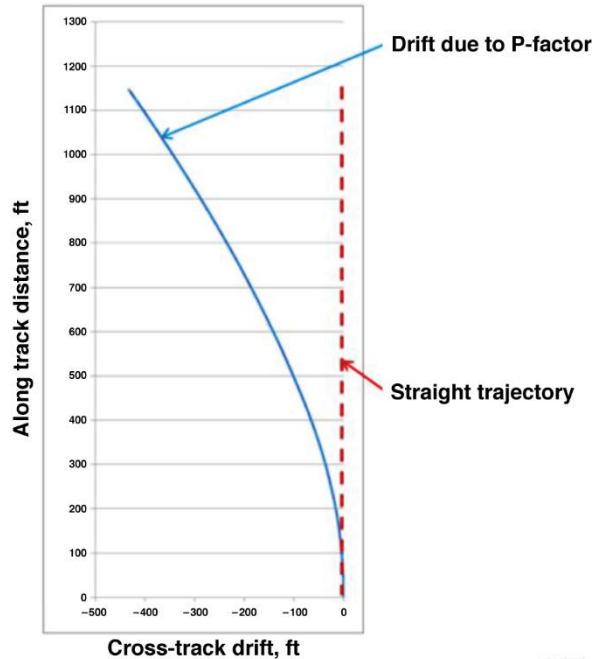
Another practical effect was observed on the preliminary characterization flights when PARS-type runs were accomplished. Although the autopilot was given a target of 40 deg for turning avoidance maneuvers, in practice the actual bank angle of the DROID tended to oscillate somewhat below 40 deg. Therefore, the trajectory prediction bank angle was changed to 37 deg to provide a better match with the actual flight-test data.

The 60-KIAS target airspeed was selected to be near the middle of the airspeed range of the DROID (45-80 KIAS) and simulated an optimum climb speed. The SUAV implementation had an advantage over the F-16 since an auto-throttle was an inherent capability of the Piccolo II autopilot. Adding an auto-throttle to the F-16 would have been prohibitive due to the additional cost.

It is expected that a generic production UAV Auto GCAS implementation would use avoidance maneuver definitions similar to those of an SUAV but would also use more optimal values customized for the particular platform. For example, some UAV Auto GCAS designers might choose to use a bank angle target scheduled as a function of gross weight and airspeed, a higher climb rate target (where available), and would probably select a higher target airspeed.

During the first few test flights for obtaining trajectory prediction data, it became very apparent that the DROID P-factor effect was much more significant than had been predicted by the low-fidelity simulation. The P-factor affects propeller-driven aircraft and is particularly apparent at lower airspeeds when such an

aircraft is climbing under high power. The P-factor effect was especially noticeable with the high-powered DROID. The main impact of the P-factor effect was that when the “straight” avoidance maneuver was commanded, the DROID didn’t fly straight - it ended up in a wings-level, climbing turn which drifted to the left at over 3 deg/s heading rate (at the slower DROID airspeeds of 40-45 KIAS). A graphical representation of drift experienced due to P-factor effect is shown in figure 21.



150059

Figure 21. Heading drift during “straight” avoidance maneuver.

A similar heading drift was present in turning maneuvers (i.e.; 40-deg left bank avoidance maneuvers had a tighter radius than expected, whereas 40-deg right bank avoidance maneuvers had a wider radius than expected). The P-factor effect was significant enough that it was included as part of the trajectory prediction.

One option for accommodating P-factor effect would have been to adjust the autopilot bank target to compensate for the heading drift (by using a biased bank angle target opposing the P-factor effect). That option would have required significant regression testing in the hardware-in-the-loop (HIL) simulation environment at a time when there was insufficient funding for such testing. In addition, it is likely that a single bank angle adjustment for the P-factor effect would not have worked well under all conditions. Therefore the decision was made to include the heading drift directly as part of the trajectory prediction.

Part of the goal of using this approach was to demonstrate the way in which specific aircraft characteristics such as P-factor effect can be readily incorporated as part of the trajectory prediction design process. The guiding philosophy was to model any unusual aspects in the trajectory prediction instead of changing the autopilot.

Figure 22 shows an example of the three trajectory predictions that were used as part of the SUAV Auto GCAS algorithm. The example shows that the straight trajectory prediction isn’t actually straight, but includes an allowance for the effects of the P-factor. The example also shows that the left trajectory prediction has a slightly tighter radius than the right trajectory prediction to account for the P-factor effect.

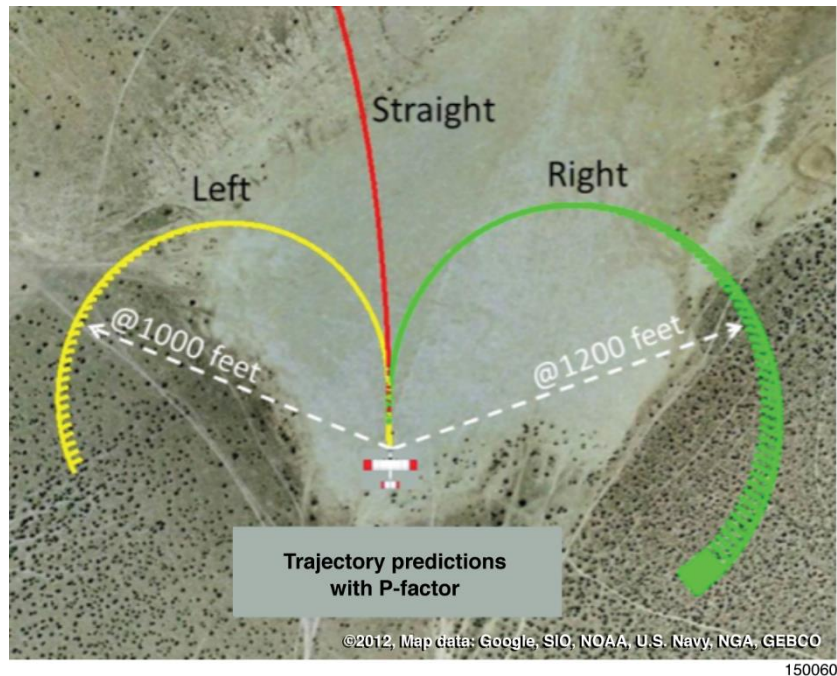


Figure 22. SUAV three-trajectory example.

Based on data from the preliminary flights, P-factor also impacted the climb rate, depending on the direction of the turn (whether the path of the turn was with or against the P-factor effect). The climb rate for the left trajectory predictions was reduced by 200 fpm to reflect this effect. The climb rate for right trajectory predictions was increased by roughly 200 fpm.

Additionally, the preliminary flights indicated that the P-factor effect was stronger when the airspeed was being increased at the beginning of climbing maneuvers. The P-factor effect was much less significant when the airspeed was decelerating or when the target airspeed was almost achieved. Therefore, the P-factor effect was modeled to be roughly half as effective when the airspeed had increased to within 5 kn of the 60 KIAS target.

As mentioned in previous sections, the SUAV trajectory prediction also used instantaneous winds as sensed by the Piccolo II autopilot to adjust the trajectory prediction. Headwinds shortened the trajectory prediction in the direction of flight. Crosswinds shortened the upwind side while extending the downwind side. Tailwinds extended the trajectory prediction in the direction of flight.

Another option worth considering on any Auto GCAS project is to assume the worst-case winds as an inherent part of the trajectory prediction. This approach completely avoids reliance on real-time wind calculations. The simplest method is to always assume a worst-case tailwind when calculating all three trajectory predictions. This method can be further modified by also assuming a worst-case crosswind component for the left and the right trajectory predictions. The general effect of these methods is to cause avoidance maneuvers to be initiated earlier than they would initiate without accommodated wind effect. If the nuisance potential of the earlier initiations can be tolerated, this “worst-case wind” approach may be a viable option for many platforms.

Also as mentioned in previous sections, the DROID was susceptible to turbulence. Vertical gusts (either from changes in local winds or flying through the up- or downdrafts created by thermals) often resulted in load factor variations in excess of +/- 0.5 g. On occasion these gusts resulted in load factor variations well

above +/- 1 g, even when the autopilot was simply attempting to hold level flight. The result was that approximately half of the accomplished PARS-type runs showed an un-conservative trajectory prediction (that is, the actual aircraft trajectory was closer to the ground than was indicated by the prediction).

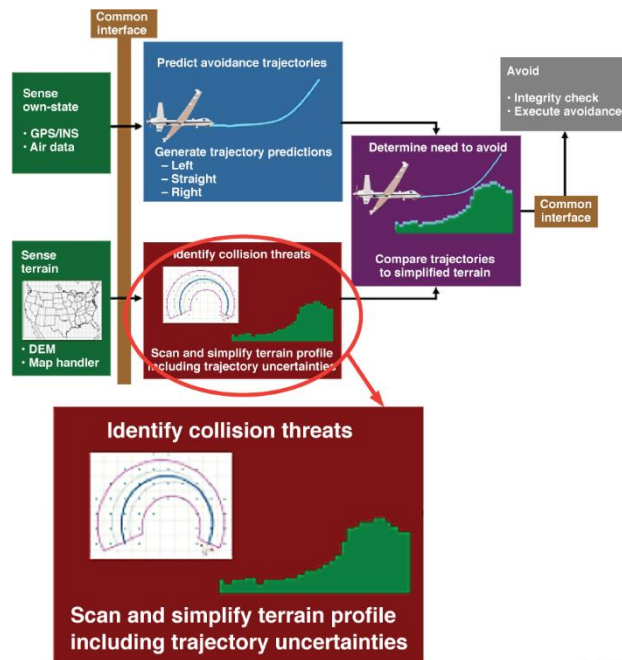
In the context of the Auto GCAS trajectory prediction, the influence of vertical gusts was obtained indirectly through the measured climb rate. Given the observed effect of vertical gusts on the DROID, a negative 200 fpm “bias” was applied to the measured climb rate input to the trajectory predictions, but with no change to the overall trajectory. Therefore, a 1000-fpm measured descent rate resulted in an initial condition feeding the trajectory prediction that appeared to be from a 1200-fpm descent rate. A 1-g level condition resulted in an initial condition that appeared to be a 200-fpm descent rate. Climbing maneuvers were also affected by the bias. The bias generally caused avoidance maneuvers to initiate slightly earlier and provided a little more pad relative to the ground.

The SUAV Auto GCAS algorithms did not include trajectory predictions that were influenced by the effects of density altitude. On a hot day, or at altitudes well above the typical test band for the DROID, density altitude effects could require significant changes in the trajectory prediction.

Recommendation 6 (R6): Future Auto GCAS projects for performance-limited aircraft may need to consider including density altitude as an input to the trajectory prediction.

Generic “Identify Collision Threats” Module

For any Auto GCAS application, the function of this module can be described most simply as “scan the digital terrain near the predicted aircraft flightpath.” Figure 23 shows the overall concept of this module. The main purpose of presenting F-16 scan pattern concepts is to show that there are options other than the methods used on the SUAV in the following section. Although many of the concepts will be similar for any aircraft, the specific methods and values chosen can be different depending on the mission type and the maneuvering capability of the relevant aircraft.



150061

Figure 23. “Identify Collision Threats” module.

The specifics of the SUAV process are described in the “SUAV Auto GCAS ‘Identify Collision Threats’ Module” section below.

In the case of the F-16, the “Identify Collision Threats” module needed to scan only the digital terrain underneath a trajectory that would encompass the possible variations during a “roll toward wings-level and pull” flyup. Therefore, the F-16 scan region was relatively straight. The F-16 digital terrain scan area needed to be wide enough to capture all of the terrain that might be overflowed throughout the entire flyup maneuver.

The F-16 digital terrain scan also needed to start from initial conditions that could be very dynamic, such as a 7-g descending turn with bank angle past 90 deg, steep dives, or a turn reversal with a high roll rate. Therefore the F-16 scan region needed to encompass the trajectory of any flyup maneuver that was initiated from these types of dynamic initial conditions.

Figure 24 shows the basic variations in scan pattern shape for the F-16. The scan pattern could be fairly narrow for straight-and-level flight. The scan pattern became wider in the direction of the turn. The scan pattern was more circular in the case of a steep dive, because the resulting flyup could go in any direction depending on the specific conditions at flyup initiation.

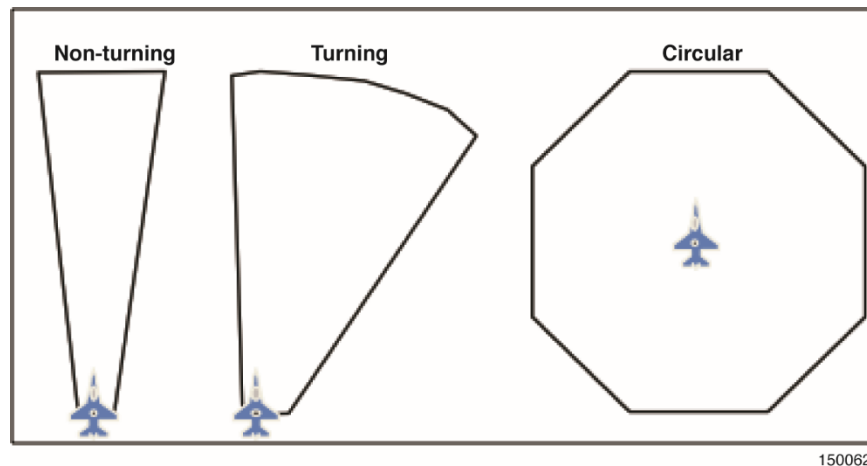


Figure 24. Basic F-16 scan pattern shapes.

The fundamental tradeoff with the F-16 was that a “too-wide” scan pattern could lead to increased nuisance potential because the digital terrain that was being scanned wasn’t actually a threat, causing unnecessary flyup activations. A “too-narrow” scan pattern could lead to reduced collision protection if the flyup maneuver flew over digital terrain that was not scanned. All Auto GCAS projects will need to find an optimal solution to that tradeoff.

The general philosophy behind the F-16 scan width was to ensure that at least two DEM posts were captured by the scan at the base of the scan pattern (near the F-16). This approach was necessary because the F-16 scan method looked for individual posts, not the rectangular areas surrounding each post. Given a DEM post spacing of 3 arc-s (roughly 300 ft between posts), a minimum scan width of greater than 600 ft at the base of the scan pattern ensured that at least two DEM posts were captured. Figure 25 shows how the base width and half angle both influence the overall shape of the scan pattern. The half angle represented how much the scan pattern needed to grow as a function of distance from the initial point.

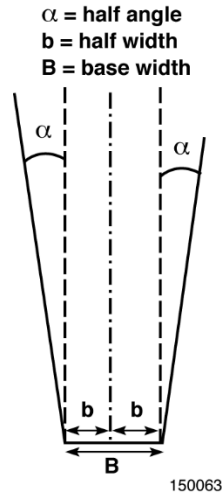


Figure 25. Basic F-16 scan pattern terms.

The chosen F-16 philosophy for scan width worked very well for normal F-16 missions above 500 ft AGL. However, for more aggressive missions below 500 feet AGL the wider scan widths would have resulted in excessive nuisance flyup activations. Therefore the F-16 design utilized a scan width narrower than 300 ft for low-level missions in order to reduce nuisance activations but at the expense of decreased collision protection. The tradeoff was considered acceptable because pilots flying those types of low-level missions were typically very aware of ground proximity and it was not a requirement to provide 100-percent collision protection. In those cases it was considered much more important to not interfere with the mission.

Another factor that determined scan width was the uncertainty in the navigation solution. Since modern embedded GPS/INS components provide excellent navigation accuracy, the uncertainty factor is no longer as significant as it was for older navigation systems. However, it is still possible for the GPS/INS location to be off by dozens of feet, and that possibility can be accounted for by increasing the width at the base of the scan pattern accordingly.

Another type of uncertainty that must be represented is called the track uncertainty. During an avoidance maneuver, the airplane will probably not track along the centerline of the trajectory prediction. That uncertainty grows with time out in front of the point at which the avoidance maneuver would need to be initiated. This is why the basic F-16 scan patterns shown in figure 24 get wider as a function of distance from the current location of the airplane. The normal F-16 scan patterns got wider at a half angle of 3.5 deg compared to the base width, as shown in figure 25.

The magnitude of the track uncertainty can be obtained either from simulation runs or from flight-test maneuvers. The goal is to try to cover all track variations that may occur without making the half angle so large that it increases the nuisance potential.

The length of the F-16 scan shapes was set at 18,000 ft. That scan length was primarily driven by the worst-case radius of the circular scan that would be needed to recover from a near-vertical supersonic dive. The 18,000-ft scan length also accounted for the tallest rising terrain that was likely to be a factor for the F-16 when using the non-circular scan shapes.

Once all of the DEM posts within a scan pattern have been identified, that information must be converted into a form that can be compared with the trajectory prediction. A simple method was used on both the F-16 and the SUAV to convert the three-dimensional array of DEM posts within the scan pattern

into a two-dimensional matrix. At a series of fixed distances from the airplane (called “range bins”) along the centerline of the scan pattern, the highest DEM post at that distance is used to generate a simplified two-dimensional matrix, as shown in figure 26.

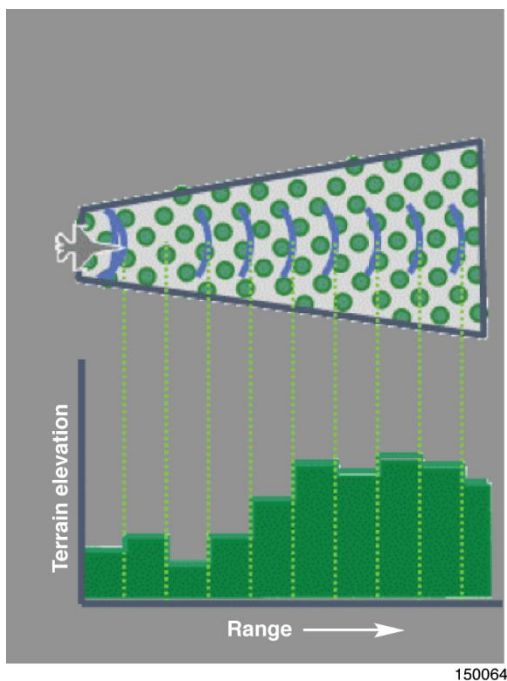


Figure 26. Converting F-16 terrain posts into a two-dimensional matrix.

Man-Made Vertical Obstructions

Many digital terrain data sources do not include information for man-made vertical obstructions such as radio towers and tall buildings. For example, sources such as NED and Legacy DTED provide “bare-earth” data without including man-made structures. One exception is SRTM data, which include a portion of some structures. The SRTM methodology was based on radar sensors, it included portions of natural or man-made objects that were reflected to the radar receiver on the Space Shuttle. The SRTM data tended to clip the maximum height off of tall, thin obstructions such as radio towers because each post was averaged over a 60- by 60-m area.

Neither the F-16 nor the SUAV designs included dedicated data for vertical obstructions or obstacles as a core part of the Auto GCAS algorithm. Since the F-16 used SRTM, approximately 50-70 percent of obstacle height was included in the DTED (for any obstacles that were in place in the year 2000 when the Space Shuttle mission was accomplished). A deliberate decision was made to not include obstacles in the F-16 Auto GCAS because very few historical F-16 mishaps involved obstacles. In addition, the F-16 already had a separate obstacle warning system. The SUAV Auto GCAS was a low-budget research project; the addition of vertical obstructions to the algorithm was considered out of scope.

Recommendation 7 (R7): The developers on any Auto GCAS project should consider the addition of vertical obstructions as part of the algorithm.

It is recommended that whatever vertical obstruction database is used be kept separate from the digital terrain data until just before the combination is converted into the simplified two-dimensional matrix. This approach will allow the vertical obstruction data to be updated independent of the terrain data as new

buildings and towers are constructed. Future Auto GCAS projects will undoubtedly report their experience with vertical obstruction implementations. It is also likely that obstacle sensors will be developed that could help supplement a DEM-based Auto GCAS.

Natural Vertical Obstructions (Trees)

Most digital terrain sources do not include information on tree height. However, the SRTM database does include any portion of tree height that was reflected to the radar receiver and processed into the resulting DTED. It is unlikely that SRTM data captured individual trees, but informal tests have been conducted showing that, for more thickly forested areas, approximately 30-70 percent of tree height has been reflected in SRTM data, depending on the type of tree and the density of the forest.

In the near future it may be very practical to include a tree height database that can provide a representative buffer for thickly forested areas without adding nuisance potential over areas that are known to be barren of trees. Until then, generalized tree buffer terms have been used, as discussed in the section “Generic ‘Determine Need to Avoid’ Module” below.

SUAV Auto GCAS “Identify Collision Threats” Module

The overall SUAV scan concept was very similar to the F-16. The fundamental steps were:

- Define a scan pattern that encompasses all DEM posts that might be overflowed during the corresponding avoidance maneuver (while also keeping the scan pattern small enough to minimize nuisance potential);
- Include an allowance for navigation and track uncertainty;
- Identify the height of all DEM posts in (or near) the scan pattern; and
- Convert the highest DEM posts into a two-dimensional matrix for later comparison with the trajectory prediction.

The fact that this Auto GCAS application was for an SUAV provided a number of benefits and constraints compared to the F-16. The primary benefits provided by the SUAV were a smaller range of airspeeds and much less dynamic maneuvering at initial conditions. The primary constraint was due to the lower climb performance. As previously discussed, the lower climb performance of the SUAV required the use of turning scan patterns in addition to a straight scan pattern. Due to the slower airspeed of the SUAV, turns could be accomplished in a much tighter radius than that of the F-16. Therefore, those turning scan patterns could be added while scanning far fewer DEM posts than similar turning scan patterns would require for an F-16. These factors enabled simpler and less variable SUAV scan patterns than those required for the F-16.

The total length of the SUAV straight scan pattern was defined to be 18,000 ft (the same as that of the F-16). The long scan length was selected because of the limited SUAV climb performance, and provided an option to climb straight over large, distant terrain obstacles. The 18,000-ft scan length was also used in termination logic to determine when an avoidance maneuver was clear of terrain.

From a practical standpoint, the 18,000-ft scan length would not be needed in most situations because the algorithm would almost always choose one of the turning avoidance maneuvers. One hypothetical situation that could result in selection of a straight avoidance maneuver would be a ridge crossing at low altitude in which elevated terrain existed on both sides of a saddle. Another hypothetical situation in which a straight avoidance maneuver might be selected would be in a very narrow “V” canyon within which turning would not be an option but where distant, tall terrain features also existed at the end of the canyon.

Figure 27 shows a simplified example of a left turning scan pattern for the SUAV. The solid dark blue line represents the centerline of the trajectory prediction for a given set of initial conditions. The purple lines represent the outer boundary of the scan pattern including the horizontal navigation and track uncertainties. The horizontal navigation uncertainty remained at a constant value of 50 ft throughout the scan pattern. The purple lines in figure 27 are a simplified representation of the outer boundaries of the scan area (as described below).

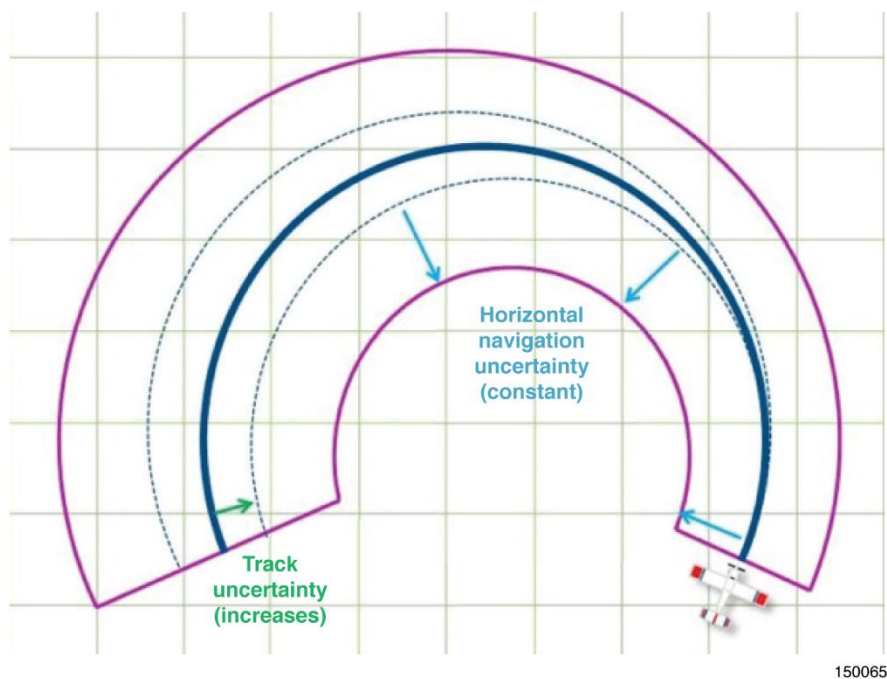


Figure 27. SUAV turning scan pattern example.

The dashed blue lines in figure 27 represent the horizontal track uncertainty and increase with distance from the initial location. The growth angle for the SUAV track uncertainty was defined as 5 deg for the straight scan and 10 deg for turning scans. Those angles were determined from analysis of early flight-test runs and included an allowance for changing winds after the avoidance maneuver initiation. Basically, the scan angles were adjusted to encompass the variations from run to run. The same technique could be applied to simulation runs.

The right turning scan pattern used the same values for horizontal navigation and track uncertainties as did the left scan. However, the resulting shape of the right scans was different due to the effect of P-factor on the trajectory prediction centerline (as discussed above).

The total length of the SUAV turning scan patterns was enough to execute an avoidance maneuver through at least 225 deg of turn. A turn of at least 225 deg was sufficient to execute a teardrop-shaped maneuver that got the aircraft pointing back toward the inbound flightpath. The teardrop shape can be visualized by extrapolating linearly backward for the inbound segment and forward for the outbound segment. Those lines converge at the cusp of the teardrop shape. The centerline for trajectory predictions at higher airspeeds had a wider turn radius, so the actual length of those scan patterns was correspondingly longer.

Once the SUAV Auto GCAS algorithm determined the size and shape of a given scan pattern based on initial conditions, the DEM posts near that pattern were scanned to determine the height of each post. That

process required the re-rasterization of CDTM data. For the SUAV Auto GCAS project it was decided to re-rasterize the CDTM at the same 3 arc-s post spacing that was used by the F-16. That was a somewhat arbitrary decision that was made solely to provide a consistent reference for comparison. The CDTM re-rasterization logic allows for selection of any post spacing.

The SUAV project could have re-rasterized the CDTM into more tightly spaced posts to reflect the tighter turn radius compared to an F-16. The SUAV project could also have re-rasterized the CDTM into less tightly spaced posts than those of the F-16 to demonstrate that lower-resolution data could have been used effectively. Lower-resolution data might have been necessary if the overall Auto GCAS algorithm had been constrained by the processor speed on the Android™ platform smartphone; that was not the case for the SUAV project, but could be the situation encountered by higher-air-speed aircraft that still need the option for multi-trajectory avoidance maneuvers.

Another SUAV design aspect that was different than that of the F-16 was the way in which digital terrain data were determined to be within the scan pattern. The F-16 technique only used terrain data when individual terrain posts were inside the scan pattern. This technique helped minimize nuisance potential but had the unfortunate side effect of missing terrain entirely if the scan pattern was narrower than the post spacing (the scan pattern was the narrowest near the aircraft). The SUAV technique included all terrain data when any part of the rectangle surrounding a post was within the scan pattern.

Figure 28 shows how DEM posts were scanned for the SUAV project. The scan pattern is the same one shown in figure 27. The small green diamonds represent the 3-arc-s DEM posts as re-rasterized from the CDTM. Some of the DEM posts identified during the scan process were actually outside the scan pattern. The fine green rectangles in the background of figure 28 represent the individual polygons corresponding to each DEM post. If any portion of those rectangular polygons was within the scan pattern, the corresponding DEM post was included. Therefore, all of the green diamonds shown in figure 28 represent DEM posts that would have been used by the algorithm, even if outside the purple scan lines. This method ensured that nearby posts were always included in the scan pattern.

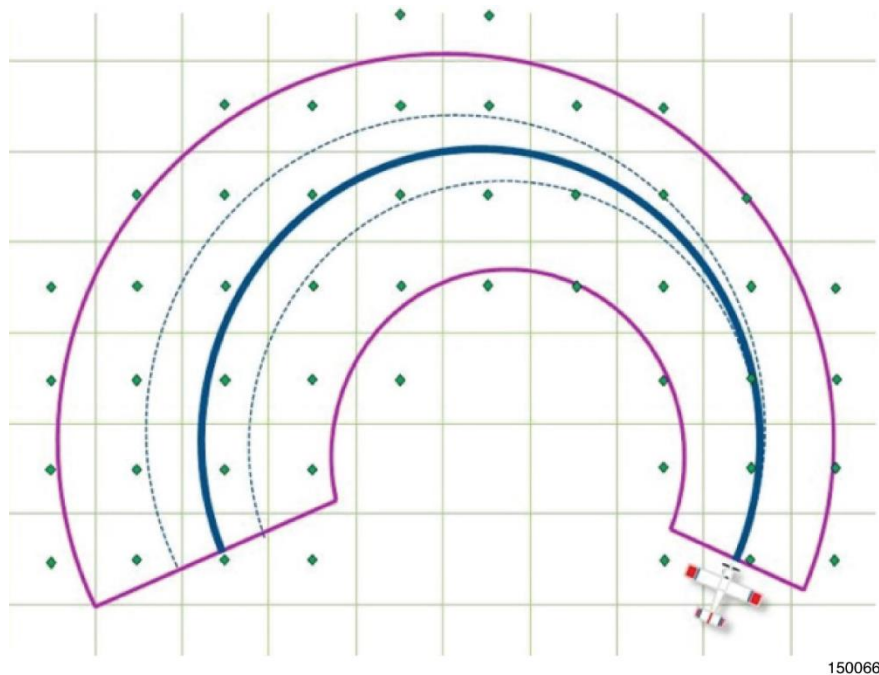


Figure 28. SUAV DEM post scanning.

Once the height of each DEM post near the scan pattern was determined, those posts were converted into a two-dimensional matrix similar to the process used for the F-16. However, the range along the trajectory prediction centerline was broken into variable length bins, instead of the constant length bins used for the F-16.

The variable bin lengths were implemented on the SUAV as a way to reduce the computational workload for the processor on the smartphone; there would be fewer bins further away from the aircraft. The SUAV variable bin length was related to the scan width. The bin length was smaller near the aircraft (roughly equivalent to the horizontal navigation uncertainty) and larger at a distance. The highest DEM post within each bin was used to assemble the overall 2-D profile.

Circular bin shapes were another method that was used to reduce the computational workload on the SUAV smartphone. The F-16 used narrow, rectangular bin shapes in order to result in a higher resolution two-dimensional terrain profile as needed for that aircraft. Because of the lower airspeeds and tighter turn radius of the DROID, it was acceptable for the two-dimensional terrain profile to be at a lower resolution than the F-16 (i.e., fewer bins over the length of the scan).

Each SUAV bin was represented by a circle of increasing radius. The center of each scan circle was located to provide overlap between neighboring bins. Using a circular scan resulted in a very simple equation to determine if a given DEM rectangle was within the scan radius for each bin. The purple lines in figure 28 are a simplified representation of the outer boundary of all of the scan circles. In theory, the circular scans might “miss” a small portion of a DEM rectangle if that portion fit between the overlapping circles. However, this was not considered a practical problem because any relevant DEM rectangle “missed” at one time frame would be identified in a later frame as the aircraft progressed forward.

Generic “Determine Need to Avoid” Module

The main function of this module was to decide when an avoidance maneuver needed to be initiated. This determination was accomplished by comparing the trajectory prediction (output from the “Predict Avoidance Trajectories” module) with the two-dimensional simplified terrain profile (output from the “Identify Collision Threats” module). When the predicted aircraft trajectory got too close to the terrain, the “Determine Need to Avoid” module sent out an avoidance maneuver command to be executed by the autopilot. Figure 29 shows the overall concept for this module.

Once the DEM posts within the scan pattern were identified and converted into a two-dimensional matrix, a generalized “built-in” buffer was applied to account for any remaining uncertainties. A primary reason for adding a generalized altitude buffer was to account for suspected errors in the DEM data. The magnitude of those errors depended on the resolution of the source data and the compression and re-rasterization techniques that were used. As a point of reference, the F-16 and the SUAV each used an altitude buffer of 30 ft to account for possible errors in the DEM.

Another reason to add a generalized altitude buffer was to account for known variations in the trajectory prediction that had not already been included in the algorithm (such as the effects of gross weight, off-standard day conditions, et cetera).

In order to provide an allowance for vertical variations in the trajectory prediction, the SUAV used an altitude buffer of 10 ft (the F-16 used 15 ft for a similar altitude buffer). Another reason to add a generalized altitude buffer was to account for vertical navigation uncertainty. The F-16 EGI output an estimate for vertical uncertainty; that value was increased by a one- or two-sigma multiplier (depending on the mode

selected) to determine the value to be used for this portion of the altitude buffer. There was no similar vertical uncertainty output available from the SUAV Piccolo II autopilot.

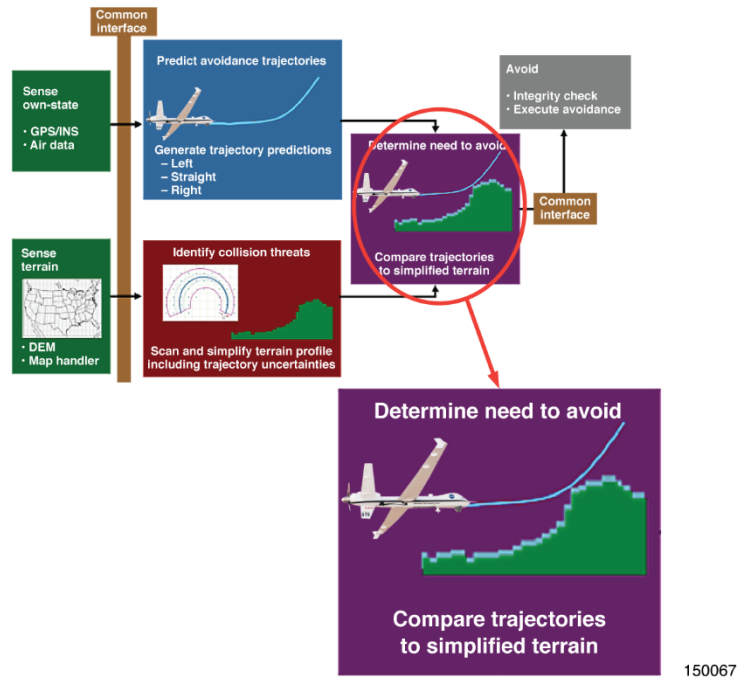


Figure 29. “Determine Need to Avoid” module.

If each of these altitude buffers were directly added to the two-dimensional DEM profile, the result could be nuisance-prone because that method would assume that all of the errors were in the same positive direction at the same time. Both the F-16 and the SUAV used a combination of each component, utilizing a root-sum-square method. The equation for combining the constituent buffers as used on the SUAV project is provided in equation (1):

$$\text{Total} = \text{NAV} + (\text{DEM}^2 + \text{TPA}^2)^{1/2} \quad (1)$$

where NAV is used as the buffer to account for GPS navigation accuracy , DEM is used as the digital terrain buffer, and TPA is used as the trajectory prediction buffer.

Auto GCAS developers should also consider the need for an additional altitude buffer. The F-16 used a tree buffer of 70 ft that was applied across the entire two-dimensional DEM profile. Obviously, many trees are shorter or taller than 70 ft. After a number of informal studies, the 70-ft value was settled upon as a way to provide some improved protection from tree impacts without excessively increasing nuisance potential. The SUAV did not use a tree buffer, because it was known that the flight-testing for this research project would be accomplished only over barren desert hillsides.

Although the concept for this module is very simple, the implementation for future Auto GCAS projects will require careful consideration to determine the answer to the question “how close is too close?” Determination of the minimum approach to terrain will require consideration of the essential mission requirements for the relevant aircraft. Even if the relevant aircraft only has a high-altitude mission requirement, it will be appropriate for any built-in altitude pad to be defined at the lowest number practical that still provides adequate collision protection.

SUAV Auto GCAS “Determine Need to Avoid” Module

Given that the SUAV Auto GCAS algorithm was based on three possible trajectory predictions, the project-specific implementation of this module required three sets of “Determine Need to Avoid” logic. Each of the three trajectory predictions (left, straight, and right) was compared with the corresponding two-dimensional terrain profile underneath that trajectory.

When any of the three trajectory predictions intersected with the corresponding terrain profile (including any altitude buffers) that direction was no longer considered a valid terrain avoidance option. When two of the directions were no longer valid options, the aircraft was allowed to continue since the third direction still provided a valid way to avoid terrain. An avoidance maneuver was only initiated when the last trajectory prediction intersected with its buffered terrain profile.

The end result of this process is represented by the example shown in figure 30. The black polygons show the flightpath of the test aircraft leading up to an avoidance maneuver initiation. The red “straight” trajectory and the yellow “left” trajectory would have intersected the terrain well before the green “right” trajectory and therefore were not viable terrain avoidance options. The green “right” trajectory was the only remaining terrain avoidance option. When that trajectory prediction intersected the buffered terrain profile, an avoidance maneuver was initiated to the right. The philosophy behind this trajectory selection was only to avoid the imminent terrain threat (given the assumption that the pilot had lost awareness of terrain proximity for any reason). There was no attempt to help the pilot resume the original flight plan. If the pilot was lost in clouds or flying at night, the algorithm would help protect against additional terrain threats, but eventually the pilot would need to climb to a safe altitude or take some other corrective action.

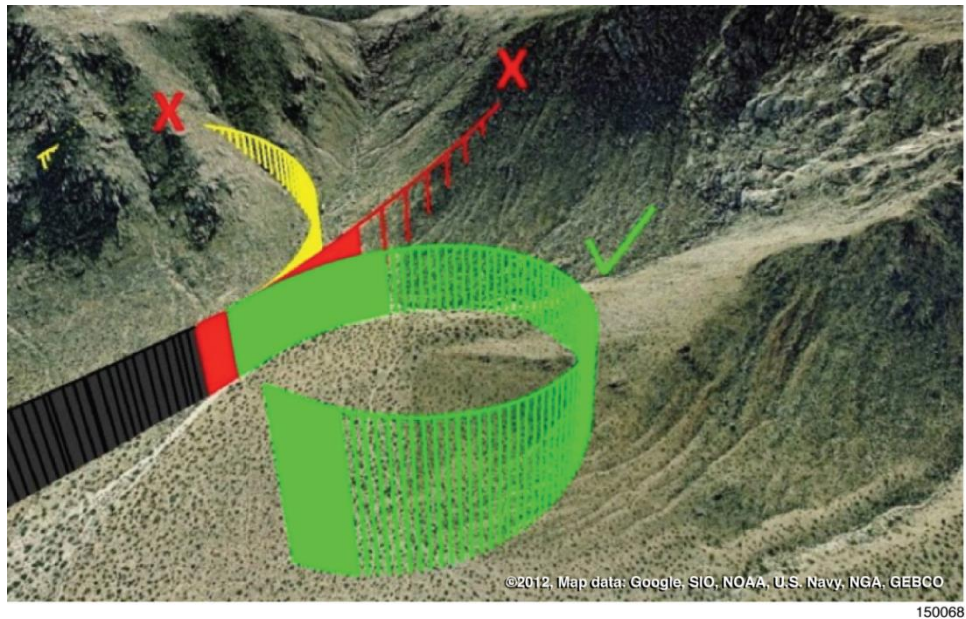


Figure 30. Example of “Avoid to the Right.”

Nuisance Potential

Previous ground collision avoidance projects have concluded that “time to avoid” is a much better metric than “distance to avoid.” An avoidance maneuver at high airspeeds needs to be initiated at a greater

distance from terrain than does a similar avoidance maneuver at low airspeeds. Using “time to avoid” automatically accounts for variations in airspeed and acts as a normalizing metric. Using “time to avoid” also provides a better indication of nuisance potential. The overall concept of “time to avoid” has been implemented in several ways. A “time-to-flyup” calculation was often used within the Auto GCAS algorithm to trigger the avoidance maneuver when that value reached zero. Also, an available reaction time (ART) calculation has been used as part of post-flight data analyses to provide an assessment of how well the algorithm achieved design goals (see the “Available Reaction Time” section below).

One of the original F-16 Auto GCAS projects in the late 1990s conducted flight tests to evaluate nuisance potential in terms of time (ref. 1). It was concluded that any flyup initiated more than 1.5 s earlier than necessary to barely avoid hitting the ground would be considered a nuisance by a pilot who was accomplishing a normal F-16 low-level mission and was also aware of ground proximity. The Auto GCAS FRRP F-16 (ref. 2) confirmed that almost all flyup maneuvers occurred within that 1.5-s criteria. The primary exceptions were at very steep dives, where the original nuisance evaluation did not apply.

Although the 1.5-s value was determined specifically for an F-16 conducting low-level missions, that value has also proven useful as a point of reference and design goal for other projects. If any Auto GCAS design can be implemented consistent with that 1.5-s value, it will probably not be considered a nuisance. Individual Auto GCAS projects can probably relax that 1.5-s design goal, but should only do so when it is certain that the resulting nuisance potential will be considered acceptable by the pilot.

For an autonomous or remotely-piloted vehicle it might be difficult to develop a specific design criterion similar to the 1.5-s value of the F-16. Test pilots were able to provide an “anxiety rating” when recovering the F-16 in close proximity to the ground that would be difficult to replicate for UAVs. It might be appropriate to develop a UAV criterion in a high-fidelity simulation, since the “sight picture” for the operator would not be much different than that seen from a ground control station.

Generic “Avoid” Module

The main function of this module (fig. 31) was to translate the avoidance maneuver command from the Auto GCAS algorithm into an avoidance maneuver that could be executed by the aircraft autopilot or flight control system. On any Auto GCAS project, the first step in this process should be to verify that the avoidance maneuver command was valid. In practice, that first step has been implemented as simple “heartbeat” checks to ensure that the avoidance maneuver command was coming from a system component that was still functional. More creative integrity checks may be possible, but care must be taken not to build in so many unwarranted checks that the system will not work when it is needed.

For the F-16 Auto GCAS, the flyup command was a single bit that basically communicated “Initiate a Flyup NOW.” The F-16 flight control system had already been pre-programmed to translate that flyup command into the various components to result in a flyup maneuver that can be summarized as “roll toward wings-level at roughly 150 deg/s, and pull as much as 5 g until clear of terrain.” The commanded rates and peaks for each individual axis were developed as part of previous simulation and flight-testing over a number of F-16 Auto GCAS projects.

The F-16 implementation of this module included some “heartbeat” checks to verify that a flyup command was being sent from a healthy avionics system (single-string) to the flight control system (quad-redundant).

The F-16 implementation also used a number of interlocks, which prevented flyup commands from being initiated when outside of the intended design envelope (that is, too slow, gear down, aerial refueling

door open, excessive uncertainty in the navigation system accuracy, excessive angle of attack, relevant failure modes, et cetera).

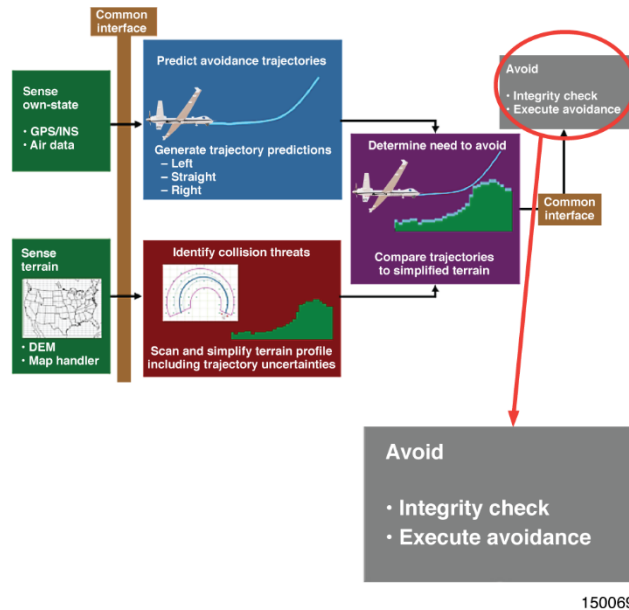


Figure 31. “Avoid” module.

SUAV Auto GCAS “Avoid” Module

For the SUAV Auto GCAS project, this module was slightly more complicated than that of the F-16, because the Piccolo II autopilot needed to execute three possible avoidance maneuvers (left, straight, and right). Each of those avoidance maneuvers had been tuned to reflect the desired response, as discussed in the “SUAV Auto GCAS ‘Predict Avoidance Trajectories’ Module” section above.

The SUAV Auto GCAS implementation was also different from the F-16 implementation in that the SUAV did not accomplish any integrity management checks within this module. All of the SUAV Auto GCAS integrity checks were accomplished as part of the “input voting plane,” as discussed in the “Generic ‘Common Interface’ Module” section above. This method was necessary because the SUAV Auto GCAS project did not have the option of inserting additional integrity checks into the proprietary Piccolo II autopilot. This implementation was less desirable than the F-16 implementation because there existed less protection against communication errors. As discussed above, this module could also be made more robust by incorporation of a monitor to protect against false requests for avoidance maneuvers.

Pilot-Vehicle Interface

For any Auto GCAS implementation intended for a production aircraft, considerable attention must be paid toward ensuring that the pilot has adequate system information, presented in a logical manner, while not being inundated with information to the point at which it interferes with other essential operations. The intended level of pilot interaction with Auto GCAS must also be determined.

The general pilot-vehicle interface (PVI) philosophy used for the F-16 was to minimize Auto GCAS displays and audio annunciations. The intent was for Auto GCAS to execute in the background, remaining unseen until needed.

In the original version of F-16 PVI, the pilot normally had no warning of an imminent flyup. When a flyup initiated, the word “FLYUP” was displayed in the head-up display (HUD) and an audio tone was announced. Pilots judged that those annunciations were sufficient to communicate why the aircraft was suddenly performing an uncommanded maneuver.

Operational pilots who had flown flight-test versions of Auto GCAS requested the option to display chevrons on the HUD that appeared approximately 3 s prior to a flyup and merged into a break-X at flyup. Some F-16 test pilots felt that the HUD chevrons could be a distraction and wanted the default to be “no chevrons.” The operational F-16 pilots found that the HUD chevrons provided helpful information when flying low-level missions. As of the writing of this report, the production F-16 version of Auto GCAS implements the chevron display as the default, but pilots were given the option to disable the chevrons. This implementation may be re-evaluated after Auto GCAS experience has been obtained in operational units.

At low airspeeds, the “FLYUP” indication on the HUD was replaced by an “AIRSPEED” indication. This indication was intended as a prompt for the pilot to add power, because insufficient airspeed could prevent the flyup from completing. The prompt was necessary because the F-16 did not have auto-throttle capability, which would allow Auto GCAS to make power changes.

The F-16 pilots were also provided with several Auto GCAS Mode settings. Most F-16 missions were conducted with Auto GCAS in the “NORM” setting. Low-level missions were conducted with Auto GCAS in the pilot-selectable “MIN” setting, which minimized nuisance potential by reducing built-in buffers while still providing good CFIT protection.

Most importantly, F-16 pilots were allowed to disable Auto GCAS at any point during a mission, and also to stop Auto GCAS flyups already in progress by using a paddle switch on the sidestick. There had been considerable debate regarding providing those two pilot controls; the concern was that pilots might disable Auto GCAS inappropriately, rendering it unable to prevent CFIT mishaps. However, the final decision was to provide the pilot with the methods for disabling Auto GCAS in case unforeseen problems were encountered in operational units. Once Auto GCAS is judged sufficiently mature, one or both of those disabling controls might be removed if the pilot community concurs.

Even after considerable evolution of F-16 PVI, it was still considered marginal by the developmental flight-test community (see reference 3 for details). The PVI for the F-22 “Line in the Sky” implementation of Auto GCAS was also considered marginal by the developmental flight-test community (see reference 4 for details). A common thread between these PVI design philosophies was the decision to minimize costs by avoiding significant changes to audio tones and other aspects of PVI. The short-term cost savings achieved by both projects may result in long-term impacts to operational users. The operational communities will provide the final judgment on the respective PVI implementations.

Pilot-vehicle interface was also considered a low priority for the SUAV Auto GCAS project, since the project was a limited-scope flight-test demonstration. Although the cockpit in the ground control van provided some capability for a pilot to manually fly the DROID, there was insufficient budget to optimize the interface for Auto GCAS. One impact was that the pilot did not have the ability to terminate an avoidance maneuver already in progress. Also, the setups for test cards could be flown by the ground cockpit pilot, but when an avoidance maneuver occurred and terminated, control was returned to the GCO and not directly to the ground cockpit pilot. Methods were under consideration to address this topic, but were not implemented, during the course of the project.

A few Auto GCAS displays were provided on a computer-generated HUD overlaid on the video from the nose camera, but these were designed more as flight-test aids than production implementations. Any

implementation of Auto GCAS on UAVs will need to pay special attention to incorporating the PVI that may be appropriate to that particular system.

Recommendation 8 (R8): Future Auto GCAS projects should carefully assess the tradeoffs between short-term PVI cost savings against the potential for longer-term impacts on the user.

SUAV Mode States

The SUAV Auto GCAS project was a flight-test demonstration, therefore mode states were not designed for a production implementation. However, some consideration was given to basic mode state logic. The basic SUAV Auto GCAS modes states were IDLE, STANDBY, ON, and FAIL.

The IDLE mode state was designed to prevent Auto GCAS modules from being executed. The IDLE mode state was also the default mode state during start-up, requiring a manual command from the UI for transition to another mode state.

The STANDBY mode state allowed the Auto GCAS modules to execute, but prevented avoidance maneuvers from being initiated. The STANDBY mode state proved useful on both the F-16 and the SUAV because it allowed Auto GCAS data to be recorded during checkout flights and nuisance evaluations without concern that an unexpected avoidance maneuver might occur.

The ON mode state allowed automatic avoidance maneuvers to initiate. The ON mode state was the typical mode state used during Auto GCAS flight-testing.

The FAIL mode state was entered when connectivity checks indicated a loss of communication between components, or when an error in the data was detected. The loss-of-communication check was disruptive on early flights when the smartphone was in the ground control van and telemetry dropouts occurred as the DROID was flown close to the hills. The dropouts could have been caused by multi-path interference with the telemetry signal. The frequency of telemetry dropouts was dramatically decreased when a directional antenna assembly was developed for use with the ground control van.

Telemetry interference due to terrain proximity could influence many Auto GCAS projects on UAVs if the algorithm is hosted on the ground. Improved aircraft antenna design (such as mounting antennas on both the top and the bottom of the aircraft) may help, but multi-path reflections off of the surrounding terrain and line-of-sight problems with the ground antenna may result in loss of communication just when it is needed most.

Recommendation 9 (R9): Future Auto GCAS projects involving flight-testing of UAVs should pay particular attention to telemetry and control links when operating in close proximity to terrain.

When the Auto GCAS algorithm was hosted with the smartphone on the DROID, FAIL states due to telemetry dropout were eliminated and the overall system was much more robust. However, a few FAIL states still occurred, due to on-board communication interruptions.

In the original SUAV implementation, when a FAIL state was annunciated Auto GCAS would not initiate an avoidance maneuver and any avoidance maneuvers in progress were immediately terminated. By comparison, the F-16 had already implemented a better FAIL state mechanization that allowed avoidance maneuvers to continue after a failure. The original SUAV mechanization proved to be suboptimal because the aircraft would stop flying away from terrain, forcing the safety pilot to take control. The FAIL state logic was re-designed for the last two flights so that an avoidance maneuver would continue if the FAIL

condition occurred after an avoidance maneuver initiation. In this case, the avoidance maneuver would continue until the safety pilot took control (or communication with the ground control van was restored).

Recommendation 10 (R10): Future Auto GCAS projects on UAVs should consider a mode state implementation that allows the avoidance to continue even after reaching a FAIL state.

The obvious problem with that re-designed mechanization would be the situation in which a hard FAIL condition occurs and there is no way for the algorithm to know when it is clear of terrain and safe to terminate the avoidance maneuver. During SUAV Auto GCAS flight-testing, this condition was not a problem because the safety pilot was always in position to take control. In a production implementation, timeout logic might be sufficient to allow the aircraft to avoid the terrain that triggered the avoidance maneuver in the first place and allow the UAV to fly back the way it came until the FAIL state no longer exists.

The SUAV Auto GCAS project was a flight-test demonstration, and FAIL states needed to be reset manually by the UIO. The F-16 implementation allowed for self-recovery from failure states.

Recommendation 11 (R11): Future Auto GCAS projects on UAVs should consider a self-recovering mode state implementation to resume CFIT protection as soon as the cause for the FAIL state no longer exists.

System Ground Testing

Prior to flight-testing, a rigorous set of verification and validation ground tests was accomplished for every new software-hardware configuration. The primary intent of these tests was to confirm that the software was coded as expected and that the hardware interfaces allowed for proper communication between components. Some ground tests were also accomplished to verify that there was no electromagnetic interference between components.

The hardware configuration for these ground tests was very similar to the flight-test configuration. All of the essential hardware components as shown in figures 11(a) and 11(b) were connected in an HIL simulation environment. The HIL setup included a smartphone running the Auto GCAS algorithms connected to a UI laptop computer (or a Gumstix[®] personal computer if simulating a “smartphone on aircraft” configuration). The portable ground station provided the necessary interface between the UI laptop computer, GCO laptop computer, ground cockpit, and the Piccolo II autopilot.

The software configuration of each set of ground tests was frozen to match the following phase of flight-testing. The most significant difference between ground testing and flight-testing was that the DROID dynamics were simulated in the HIL simulation environment hosted within the AFRC Simulation Laboratory. The DROID simulation was not very representative of the actual test aircraft, but was still adequate to accomplish ground testing.

Flight-Testing

The standard set of parameters as recorded throughout each test flight for the SUAV Auto GCAS project is given in appendix A. The specialized data analysis techniques that were used are presented in appendix B. A flight log for all SUAV Auto GCAS flights is presented in appendix C. The flight log is divided into two sections. The first section documents the preliminary flights that were used to obtain basic

aircraft response data and accomplish the initial Auto GCAS functionality checks. The second section documents the primary Auto GCAS flight tests. Appendix D contains a summary of the open NASA discrepancy reports (DR) that existed at the end of the SUAV Auto GCAS project. The main intent for this list is to allow the staff on similar future projects to decide if each item warrants improvement.

Test Objectives

Prior to formal SUAV Auto GCAS flight-testing, a few flights were conducted with the objective of obtaining basic DROID response information during maneuvers that were similar to Auto GCAS avoidance maneuvers. These characterization flights were accomplished using normal DROID flight procedures and normal DROID autopilot functionality. The data obtained from these flights were not directly applicable to an Auto GCAS avoidance maneuver because each axis of the maneuver had to be initiated by separate keystrokes (instead of simultaneously). However, the limited data obtained from these flights were sufficient to provide a decent starting point for the Auto GCAS algorithm.

The SUAV Auto GCAS flight-test objectives were to:

- Accomplish basic functionality checks;
- Verify Auto GCAS component function; and
- Evaluate fully integrated Auto GCAS: Evaluate:
 - collision avoidance capability; and
 - nuisance potential (limited).

The test flights tended to be conducted in groups based on the location of the test site and whether the smartphone was on the ground or on the test aircraft.

Test Sites

Three test sites, each site serving a particular purpose, were used for all of the SUAV Auto GCAS flight tests. Figure 32 provides an aerial map view of the three test sites.

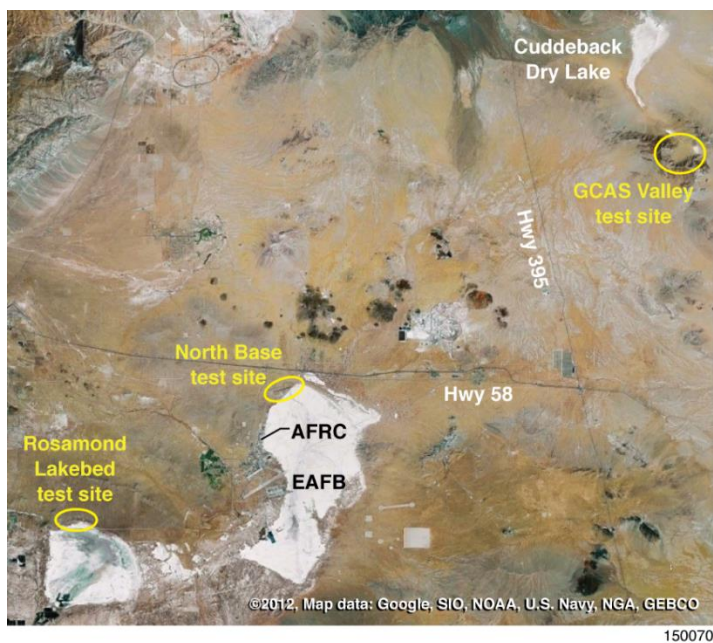


Figure 32. Test site overview.

The coordinates for each test site are provided to enable readers to find those locations on Google Earth (Google Inc., Mountain View, California). In each case, the coordinates represent the approximate position of the ground control van as shown in the figure for that site.

The North Base test site lies at $34^{\circ} 59.400'$ north latitude and $117^{\circ} 51.890'$ west longitude. The North Base test site was the closest location to AFRC and therefore required the least amount of logistical effort for test team deployment. This site was also selected because it was where normal DROID operations had previously been accomplished. The North Base test site was one of the locations within EAFB airspace that had been approved for SUAV operations (as coordinated through the normal flight-test scheduling process).

The North Base test site had the advantage of an existing hard-surface runway (fig. 33). From an Auto GCAS point of view, the North Base test site was limited to smooth-terrain test points because the entire area is quite flat.



Figure 33. North Base test site.

The first few characterization flights and functional check-flights were accomplished at the North Base test site location. The Auto GCAS component functions were also verified at this test site. One of the North Base test site flights included the first few smooth-terrain Auto GCAS test points.

The Rosamond Lakebed test site lies at $34^{\circ} 52.132'$ north latitude and $118^{\circ} 5.200'$ west longitude. The Rosamond Lakebed test site offered the slight advantage of small terrain features that could provide Auto GCAS test options other than those provided by the predominantly smooth terrain of the North Base test site. The main advantage of the Rosamond Lakebed test site was that the logistics of test team deployment were much simpler than what would have been required to deploy the test team to a remote desert location.

There is a small hill (approximately 75 ft tall) on the north side of the lakebed. That hill was used as the location for the first few Auto GCAS test points over terrain that was not smooth (fig. 34). The safety pilot was located at the top of that hill for those test points.

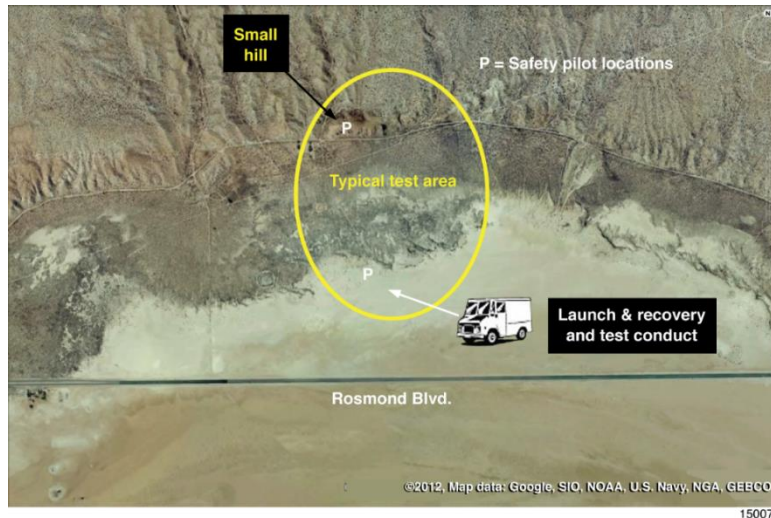


Figure 34. Rosamond Lakebed test site.

The Rosamond Lakebed test site offered no hard surface runway, but the lakebed surface was normally dry enough to support DROID takeoff and landing operations. This test site was another of the locations within EAFB airspace that had been approved for SUAV operations (as coordinated through the normal flight-test scheduling process).

The GCAS Valley test site lies at 35° 12.208' north latitude and 117° 25.116' west longitude. The GCAS Valley test site was selected because it provided a very useful variety of terrain features, and because one of the small lakebeds within it provided an adequate takeoff and landing surface for the DROID. The specific test locations within the GCAS Valley test site (fig. 35) were selected in order to provide adequate areas in which to position the safety pilot and ground observers.

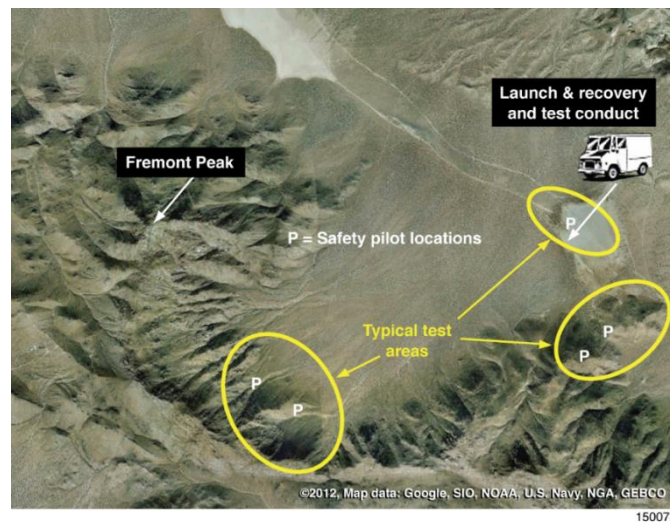


Figure 35. GCAS Valley test site.

The GCAS Valley test site required the most logistical effort for test team deployment. This test site was only reached after almost two hours of driving time from AFRC; approximately half of that time spent driving on partially-maintained dirt roads. This test site was located directly underneath the Black Mountain low-altitude supersonic corridor, and required special scheduling to avoid airspace conflicts with other flight-test operations. On each test day, a Notice to Airmen (NOTAM) was issued to alert other pilots to the DROID test activity in that area.

The test area closest to Fremont Peak was about 1.5 miles from the lakebed used for takeoff and landing. It required an additional 30-45 minutes to move the safety pilot and ground observers to their assigned stations via rough dirt roads.

Despite the logistical challenges, this test site was well-suited for Auto GCAS flight testing on the DROID class of aircraft. Almost all SUAV Auto GCAS flight-testing was accomplished at the GCAS Valley test site. This test site included smooth terrain; a small hill (approximately 350 ft taller than the surrounding terrain) with safety pilot access to the top of that hill; moderate terrain (approximately 700 ft taller than the surrounding terrain) including a small box canyon with safety pilot access on the side of that canyon; and a small saddle area for nuisance testing.

Ground Survey at the GCAS Valley Test Site

An important consideration for the GCAS Valley test site was whether or not the terrain features were actually at the same longitude, latitude, and altitude positions as those indicated by the DEM within the NED. The answer was not considered a flight-safety issue because the safety pilot remained the primary protection against ground impact. However, a technical problem could have arisen if the actual terrain in the GCAS Valley were significantly different than that as shown by the NED, so a ground survey for key terrain features within the GCAS Valley was conducted.

Ground surveys have been a common procedure on past Auto GCAS flight-test projects. On the F-16 Auto GCAS projects in the 1990s and portions of the F-16 FRRP, ground surveys were accomplished by project personnel hiking to the relevant terrain features with handheld GPS units. This same technique was also used for the SUAV Auto GCAS project in the GCAS Valley for the valley floor and the accessible sides of the hills (only where the terrain was not too steep to walk).

One of the test sites used by the F-16 FRRP included terrain that was much too steep for simple hiking surveys. At that test site the professional surveyors at the NGA were requested to accomplish a more formal survey. These NGA surveyors were already stationed at EAFB and conducted surveys like this as a normal part of their support for base activities. For inaccessible terrain locations these surveyors used standard triangulation techniques to obtain high accuracy.

The NGA was also requested to support the SUAV Auto GCAS project by surveying the more inaccessible portions of the GCAS Valley. From an Auto GCAS perspective it was not necessary to obtain a ground survey in a rectangular grid for the entire area; it was only necessary to survey the ridgelines at the planned test sites within the GCAS Valley. The ridgelines were the most likely areas to trigger an avoidance maneuver, and were also the areas where it was most likely for the test aircraft to get closest to terrain (during or shortly after the avoidance maneuver).

Figure 36 shows the results of the NGA survey for a portion of the small box canyon area. The bottom cusp of each white balloon represents the survey altitude at that longitude and latitude. The white line below each balloon represents the difference between the survey altitude and the Google Earth representation.



Figure 36. NGA survey at primary test area.

The Google Earth graphics were also based on NED data, and figure 36 is considered a good depiction of the altitude errors that would be inherent to the Auto GCAS algorithm that was also based on NED data. Figure 37 provides an overall summary of the difference between the NGA survey and the NED data at the closest NED post to the survey coordinates.

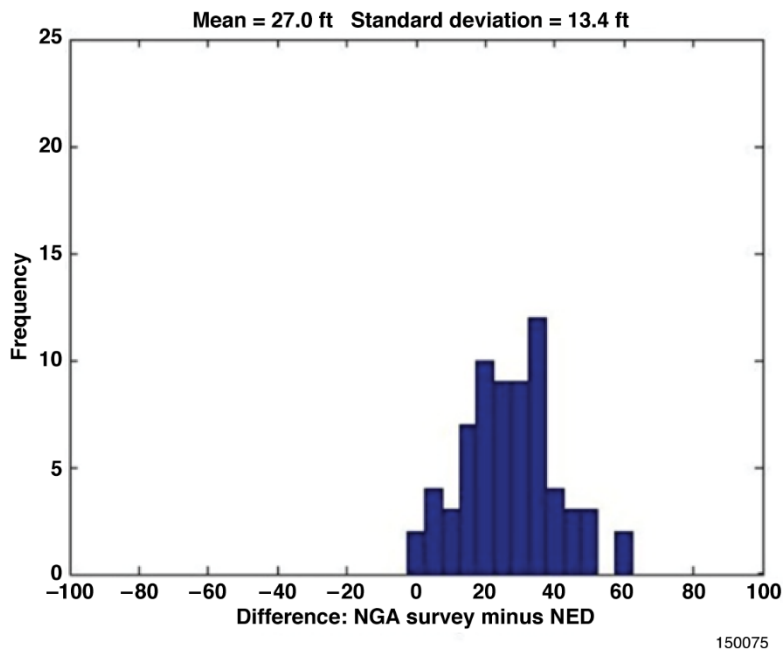


Figure 37. NGA survey histogram.

The histogram data shown in figure 37 indicate that the survey data were higher than the nearest NED post by about 27 ft on average. One survey point was approximately 60 ft higher than the corresponding NED post. There were no cases in which the NGA survey data were below the NED data.

The overall conclusion from the NGA survey was that the NED data were sufficiently accurate to provide a viable source for the Auto GCAS algorithm. The 27-ft average error was within the tolerance provided by the DEM uncertainty buffer of 30 ft. The scan pattern methodology and the selection of DEM posts outside the scan pattern (fig. 28) provided sufficient additional buffering.

Test Methods

The primary test methods used were:

- PARS (to obtain trajectory data);
- Collision avoidance testing;
- Mishap scenarios; and
- Nuisance evaluations.

Each method is described below.

Pilot-Activated Recovery System

The SUAV flight-test technique used was based on the F-16 PARS maneuver. The goal was to initiate a maneuver that replicated an avoidance maneuver as closely as was practical. For the SUAV Auto GCAS project, PARS maneuvers were initiated by the GCO instead of the pilot, but the intent was the same.

The PARS maneuvers provided a significant flight-test advantage over Auto GCAS -initiated avoidance maneuvers when the only intent was to obtain trajectory data. Auto GCAS avoidance maneuvers require that the test aircraft be maneuvered into a particular set of flight conditions over a specific terrain feature. The PARS maneuvers could be initiated as soon as the aircraft was set up on the target test conditions, regardless of terrain. The extra maneuvering time required to set up for an Auto GCAS maneuver significantly impacted test efficiency and effectiveness. On average, SUAV PARS test runs were executed at twice the rate of Auto GCAS test runs.

Collision Avoidance Testing

The vast majority of SUAV Auto GCAS flight-testing was collision avoidance testing. These tests always required the most planning and test team coordination, because the test aircraft was being intentionally aimed at terrain.

The single most important guiding principle for collision avoidance tests was, “Do not rely on Auto GCAS to prevent a collision.” Any collision avoidance system under test could reveal unforeseen problems at the most inopportune time. Discovering those problems when close to terrain could easily result in a mishap, so extra care and control was necessary to maintain test safety.

In a piloted test airplane like the F-16, this was accomplished by placing all responsibility for avoiding collisions on the test pilot. In the case of the SUAV project, all responsibility for avoiding collisions was placed on the safety pilot (located on the ground near the test area). The essential point was that the F-16 test pilot and the SUAV safety pilot were briefed to take control as needed, regardless of how much confidence had been developed in the Auto GCAS.

Another important guiding principle for collision avoidance tests was, “Add a flight test altitude buffer for each new set of flight conditions.” Both the F-16 and the SUAV used a terrain clearance buffer (TCB) that was implemented as a flight-test aid. The TCB caused avoidance maneuver initiations to occur at a higher altitude than would normally occur with the basic Auto GCAS algorithm.

The TCB value was specified on the test cards for each run and could be set by the test pilot on the F-16 project and the UIO on the SUAV project. The TCB value specified for a given run was dependent upon the target test conditions. In general, steeper dives and more dynamic maneuvers led to higher TCB values. Once Auto GCAS had been evaluated at higher TCB settings, the same test conditions could be repeated at lower TCB settings.

Some SUAV collision avoidance flight tests were conducted with a TCB value of zero. Those runs used whatever buffer was built into the Auto GCAS algorithm, but there was no additional flight-test buffer. Tests with zero TCB were only accomplished once the test team was satisfied with the system performance established using higher TCB values. When testing with zero TCB the responsibility for avoiding collisions remained with the safety pilot.

A primary goal of the collision avoidance tests was to thoroughly “wring-out” the Auto GCAS implementation under controlled conditions prior to conducting nuisance evaluations, which tend to be less scripted. On the F-16 project, this goal was accomplished during incremental phases because the emphasis was on reaching the nuisance evaluations phase as quickly as practical. It was critical to show that the F-16 Auto GCAS would not be nuisance-prone under operationally relevant scenarios. On the SUAV project, the main emphasis was on demonstrating the portability of the Auto GCAS algorithms, so there was much less emphasis on nuisance evaluations. Therefore, nuisance evaluations were only conducted on the last two flights of the SUAV project (for more background see the “Nuisance Evaluation Techniques” section below).

On the SUAV project, the general collision avoidance test procedure for a given test run began with the GCO commanding the test aircraft into a circular holding pattern (see fig. 38). The holding pattern was typically a few hundred feet above the intended test point altitude and at least 500 ft above the valley floor. For the example shown in figure 38, the holding pattern was clockwise. Before the test aircraft left the holding pattern, the GCO commanded the test aircraft to descend to the intended test point altitude. After the test conductor confirmed that the test team was ready to test, the GCO commanded the test aircraft to leave the holding pattern and enter the test pattern.

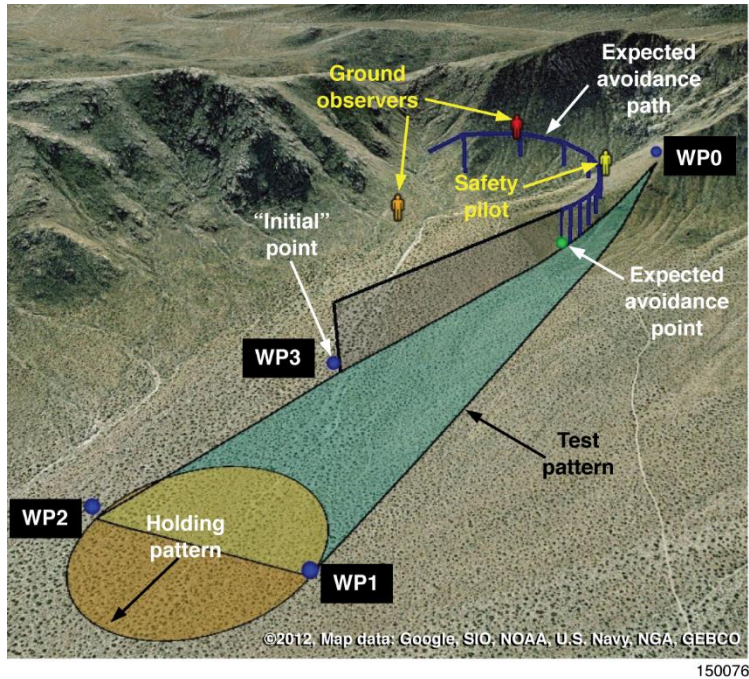


Figure 38. Example collision avoidance pattern showing waypoints (WP).

A sequence of four waypoints was used to establish the test pattern. Waypoints 1 and 2 consisted of longitude, latitude, and altitude coordinates located at the tangent points of the holding pattern that remained fixed throughout a test mission. Waypoint 3 was located at the “Initial” point for the final run-in to an expected avoidance maneuver (represented in figure 38 by the black line between the “initial” point and the expected avoidance point at the green dot). The coordinates for waypoint 3 varied, depending upon the setup needed for a particular test point. Between waypoints 2 and 3, the test aircraft was usually commanded to accelerate or decelerate to the test airspeed. Waypoint 0 was the “aimpoint” for each avoidance maneuver. Waypoint 0 established the heading for the final run-in line from waypoint 3. The coordinates for waypoint 0 varied depending on the aimpoint needed for a particular test point.

The expected avoidance point was defined on each test card. The coordinates for that point assumed that the test aircraft flew exactly along the planned run-in line, but that was not always true because of crosswinds or other setup problems. Those coordinates were also used for a countdown display in the ground control van. The dark blue line in figure 38 represents the expected avoidance path. That path was shown on the test cards to provide the test team with an idea of what to expect for a given test run.

The primary information communicated on the test cards was whether a given run was expected to execute an avoidance maneuver left, straight, or right. However, the actual test event did not always occur as expected. In particular, if a given test run was aimed along the crease in a ridgeline, the avoidance could go either left or right, depending on small setup variations. Those subtle Auto GCAS algorithm decisions were intentional parts of the test. The test team was always ready for the test aircraft to turn in unplanned directions. The safety pilot always took control if the aircraft appeared to be headed too close to terrain or if it was heading outside the field of view (such as over a ridgeline or into the glare of the sun).

The test cards also provided the test team with some idea of the specific path the test aircraft might take relative to the safety pilot and ground observers. The ground observers were located in key positions relative to nearby terrain features to assist the safety pilot when the test aircraft was flying close to those terrain features the proximity of which was not easy to judge from the safety pilot perspective (see example

in figure 38). If the aircraft remained clear of terrain, the ground observers made periodic “clear” calls on the radio. If the ground observers judged that the aircraft was getting too close to terrain, they called for an “abort” on the radio and the safety pilot took immediate control.

When a test run went according to plan and the avoidance maneuver terminated normally, the test conductor called “test complete” on the radio. That call was another signal for the safety pilot to take control. Therefore, the safety pilot took control on every collision avoidance test run, even if everything went according to plan. This procedure was an intentional one, the purpose of which was to eliminate any uncertainty as to who was flying the aircraft and when on every collision avoidance test run.

After taking control on every collision avoidance test run, the safety pilot would first and foremost ensure terrain clearance. Once satisfied that the aircraft was well clear of terrain, the safety pilot then maneuvered the aircraft back toward the holding pattern. When the test conductor determined that the test aircraft was in a good position, control was transferred from the safety pilot to the GCO. The GCO maintained the aircraft in the holding pattern until the test conductor coordinated the setup for the next test card. The test team became very proficient at executing these test procedures and averaged one collision avoidance test run every 5 min.

Mishap Scenarios

A few collision avoidance test runs were configured to replicate three scenarios that have occurred in real-world UAV and general aviation mishaps. The basic flight conditions for each of these mishaps were obtained from the USAF Safety Center (for the UAVs) and the National Transportation Safety Board (for the Cessna).

A top-level description of those mishap scenarios is:

- During an approach to landing, an MQ-9 was destroyed after hitting some power lines short of the runway after a steep descent. The key mishap flight conditions replicated were the bank angle and descent rate over smooth terrain. The test airspeed was approximately 10 kn less than for the mishap aircraft.
- During combat support operations in mountainous portions of Afghanistan, an MQ-1 Predator (General Atomics, San Diego, California) was destroyed when the aircraft entered a box canyon while attempting to cross from one mountain valley into another valley. The key mishap flight conditions replicated were the bank angle, dive angle (near level flight), and ingress to a terrain feature that resembled a box canyon. Once again, the test airspeed was approximately 10 kn less than for the mishap aircraft.
- During a nighttime cross-country flight over rising mountainous terrain, a Cessna Turbo 182T (Cessna Aircraft Company, Wichita, Kansas) was destroyed with two fatalities when the established climb rate was insufficient to clear a ridgeline. The key mishap flight conditions replicated were the climb rate on a wings-level approach to rising terrain. The exact airspeed at impact for the mishap aircraft was not known but was likely about 20-30 kn faster than the test aircraft.

Nuisance Evaluation Techniques

The primary test technique used for nuisance evaluations was to mimic UAV mission segments that could occur during normal operations while also within reasonably close proximity to terrain. The primary intent of those tests was to evaluate whether Auto GCAS avoidance maneuvers would be considered a nuisance if they occurred during those mission segments.

Most normal UAV operations are conducted many thousands of feet above terrain, where the risk of CFIT is very low. However, a number of situations exist within which the risk of CFIT is higher.

Any operations in mountainous regions could place the UAV near its performance ceiling, forcing the UAV to attempt ridge crossings in the passes between mountain peaks in order to get from one valley to another (as in the MQ-1 mishap). Examples of this scenario include combat operations in mountainous regions such as Afghanistan, but could also include peacetime missions in support of firefighting operations over mountainous areas within the CONUS. In addition, even if the UAV is not flying near its performance ceiling, air traffic control limitations could place a ceiling on operations that constrains the UAV to an altitude lower than surrounding mountain peaks.

Two UAV mission segments were selected for SUAV Auto GCAS nuisance evaluations. These were called “ridge crossings” and “valley patrols.” In order to obtain a more qualitative evaluation, both of these segments were planned for a pilot manually flying those mission segments. A system test of Auto GCAS could also have been flown under GCO control using the normal technique using waypoints for both mission segments.

Ridge crossing tests were constrained by the requirement that the safety pilot keep the test aircraft in sight at all times. The test aircraft could not cross a ridge above the safety pilot and then continue across that ridge beyond where the safety pilot would lose visual contact. Therefore, a more practical location was selected that allowed the safety pilot to be positioned where the test aircraft would remain in sight throughout the test run. The only location at GCAS Valley with reasonable dirt road access was a small saddle area. This saddle allowed limited ridge-crossing testing, but the terrain was only slightly higher on both sides of the saddle (fig. 39).

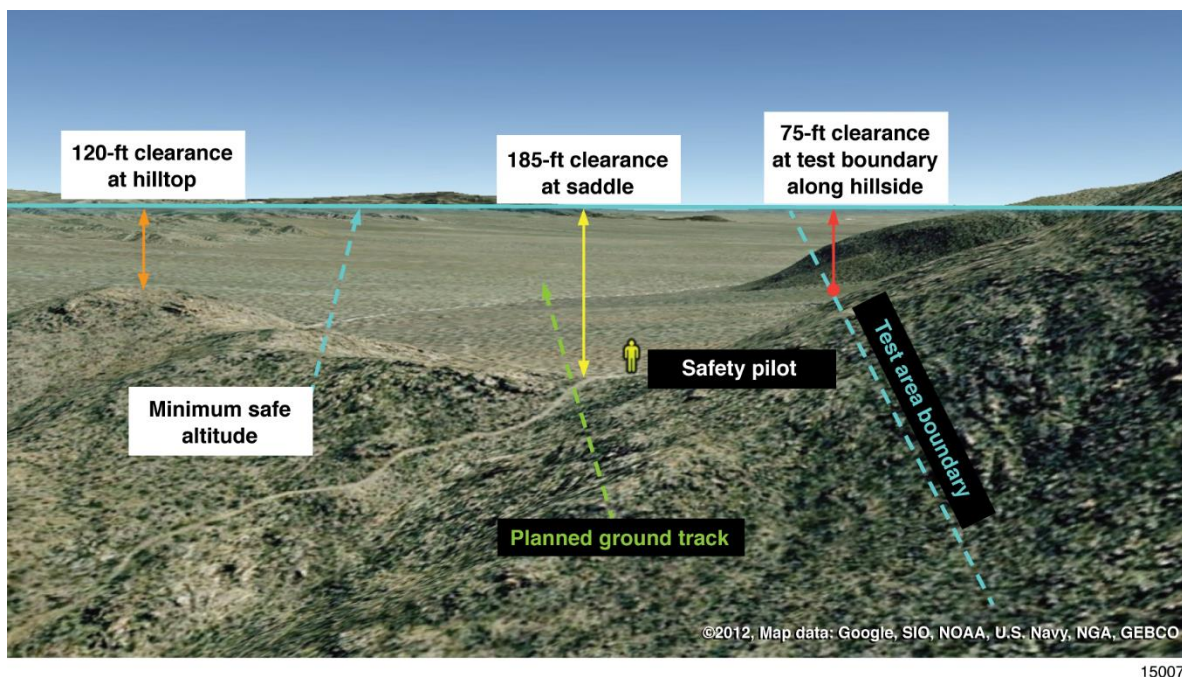


Figure 39. Ridge crossing test location.

The test cards for ridge crossings listed a minimum safe altitude that would keep the test aircraft clear of all terrain within the designated test area (as shown in figure 39). The minimum safe altitude was based on MSL altitude as provided by the Piccolo II autopilot GPS/INS. The pilot was directed to stay above that

minimum safe altitude while also staying within the test area, a large display of each of which was visible directly in front of the pilot. This method was used to attempt to simulate typical mission constraints that might be placed on a UAV operator.

If the pilot flew directly over the saddle while also above the minimum safe altitude, there would be at least 185 ft of clearance. The intent was to show that the saddle could be crossed at a reasonably low altitude without the Auto GCAS algorithm inducing a nuisance avoidance maneuver.

If the pilot strayed close to the boundary of the test area, the clearance would have been approximately 75 ft. The safety pilot was in a good position to take control if a run became too close to terrain on that side of the saddle.

Valley patrol tests were intended to represent a mission segment in which a UAV was tasked to loiter within a mountain valley while conducting normal surveillance operations. Similar to the ridge crossings, the pilot was directed to stay above a minimum safe altitude while also staying within the test area (fig. 40).

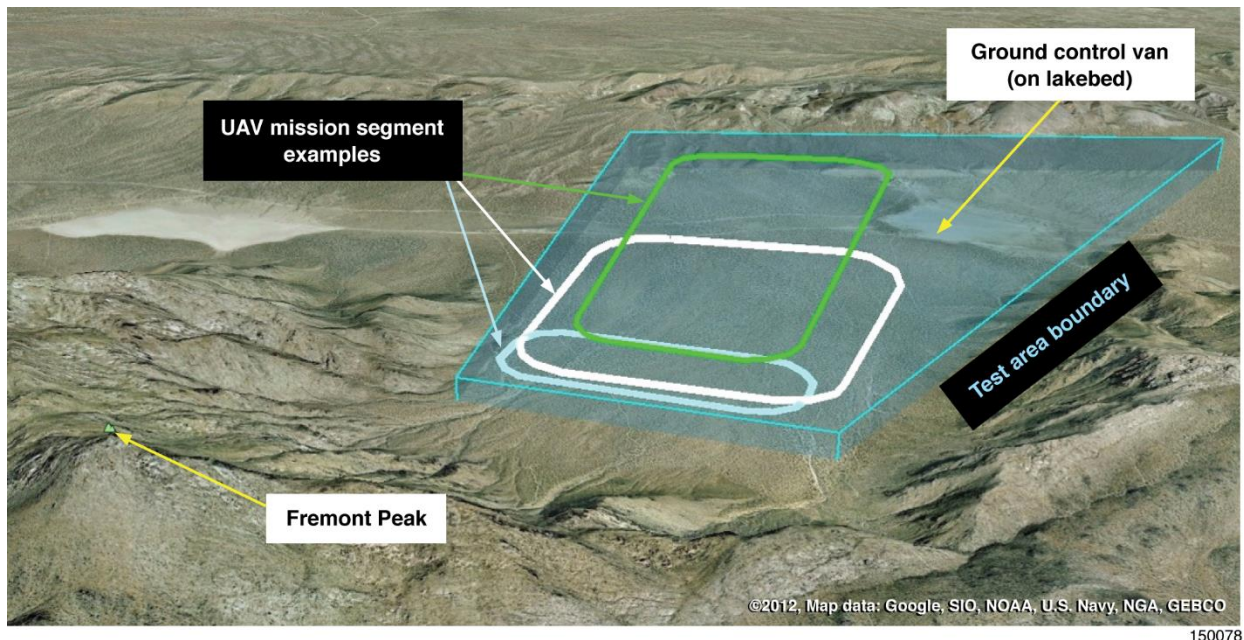


Figure 40. Valley patrol test area.

Given those mission constraints, there would be at least 150 ft of clearance anywhere within the test area. The examples in figure 40 show typical racetrack patterns, but the pilot was allowed to follow any pattern as long as the test boundary and altitude constraints were observed. The intent was to show that missions of this type could be conducted at reasonably low altitude without the Auto GCAS algorithm inducing a nuisance avoidance maneuver.

Valley patrol tests were planned so that the safety pilot and ground observers could be located where they could monitor the test area boundaries and abort the run if the test aircraft strayed too far outside those boundaries.

Ridge crossings and valley patrols were planned to be accomplished either with the normal three-trajectory Auto GCAS algorithm (allowing left, straight, or right avoidance maneuvers) or with the

multi-trajectory option disabled (allowing only straight avoidance maneuvers). The intent was to provide a comparison of the nuisance potential of those differing design approaches.

Test Preparation

It may be possible to accomplish Auto GCAS flight-testing by selecting a likely terrain feature and aiming the aircraft at that feature. However, more useful test results and safer test conduct can be achieved when the test team knows what to expect for each test run. A powerful, integrated flight-test planning tool was developed in support of the SUAV Auto GCAS project. That tool consisted of two primary components:

- The SUAV Auto GCAS avoidance maneuver simulation, and
- The avoidance maneuver visualization using Google Earth.

The combination of these two components considerably enhanced flight-test planning. The avoidance maneuver visualization component also greatly improved flight-test data analysis, as discussed in the “Test Results” section below.

SUAV Auto GCAS Avoidance Maneuver Simulation

A customized simulation of SUAV Auto GCAS avoidance maneuvers had been developed early in the project as a design tool. That simulation enabled the user to establish aircraft setup conditions relative to terrain features and initiate the simulation. The results then provided an indication of where and when the avoidance maneuver was expected to occur. At a more fundamental level, the simulation also indicated whether an avoidance maneuver would occur, because some setups would not necessarily lead to an avoidance maneuver. For SUAV Auto GCAS, the simulation also predicted whether the avoidance maneuver would be executed to the left, straight, or right.

One of the original purposes for this simulation was to provide software programmers with a functional reference model. The simulation provided test cases that could be used to compare with results from the Java™ software hosted on the smartphone.

The SUAV Auto GCAS simulation used the same modular architecture and the same simplified kinematic models for the avoidance maneuver dynamics as those described in the “SUAV Auto GCAS Software Components” section above. The simulation also used a subset (limited to the SUAV Auto GCAS test sites) of the same DEM data that the smartphone would use.

The use of a simplified Auto GCAS simulation for this purpose has a history that goes back to early F-16 Auto GCAS flight-testing in the 1990s. At that time, the simulation was successfully used for flight-test planning and to determine the abort altitudes required to observe flight-test safety procedures. The simplified kinematic models in that F-16 flyup simulation were initially generated based on hundreds of high-fidelity six-degrees-of-freedom simulation runs provided by Lockheed Martin. Those kinematic models were later verified using flight-test data. The F-16 flyup simulation was also used for trajectory reconstruction in over a dozen F-16 CFIT mishap investigations.

In the 1990s, the utility of the output from the F-16 flyup simulation was of course constrained by the current state of the art for computer graphic visualization tools. It was possible to feed the predicted trajectory from the simulation (or the actual trajectory from flight-testing) into three-dimensional computer graphics software to generate views of the flightpath of a run relative to the intended terrain, but that process was labor-intensive and each run could take hours to generate from a single perspective. More recent

computer graphics tools have helped overcome that constraint for the SUAV Auto GCAS project as described in the following section.

Avoidance Maneuver Visualization

Various figures in this report have already provided examples of the SUAV Auto GCAS visualization tools (figs. 22, 30, and 38). The tremendous utility of these tools was enabled by linking the output from the SUAV Auto GCAS simulation to provide inputs into Google Earth. This linking was accomplished by using MATLAB® (The MathWorks, Natick, Massachusetts) to feed the outputs from the simulation into standard Google Earth graphic functions. The result was keyhole markup language, or KML, files that could be viewed easily in Google Earth. (Keyhole Corp., Mountain View, California, was a technology company purchased by Google in 2005).

Examples of the types of information output from the simulation into Google Earth included:

- The planned flightpath leading up to the avoidance maneuver;
- The predicted flightpath during the avoidance maneuver;
- The scan pattern used at avoidance maneuver initiation;
- The highest terrain “threat” at a given range from the aircraft;
- The height of the combined TCB and built-in buffers above the highest terrain at a given range; and
- Alternative trajectories that were not selected.

The DEM data were also represented by placing Google Earth polygons around each terrain post, but these data were not output directly from the simulation.

An example of how Google Earth was used to help visualize the setup for a particular flight-test point is shown as a top view in figure 41. This figure shows a top view of the terrain near the scan pattern used by the Auto GCAS algorithm at the instant at which the right avoidance maneuver was predicted by the simulation. The left and straight trajectory predictions were also calculated by the simulation, but were already rejected as invalid avoidance options prior to this point in time.

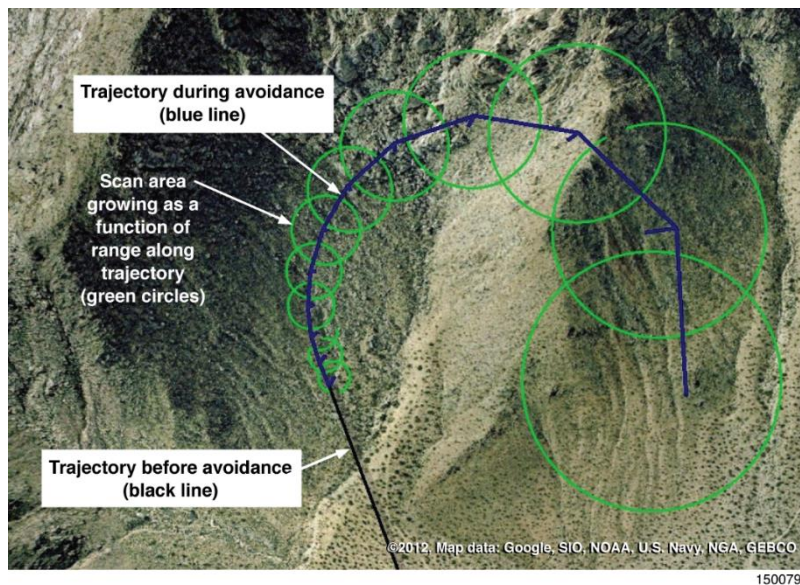


Figure 41. Circular terrain scanning example 1.

The green circles in figure 41 illustrate how the expanding scan pattern can be simplified by combining the navigation uncertainty and track uncertainty (as described in the “SUAV Auto GCAS ‘Identify Collision Threats’ Module” section above) into expanding circles centered at increasing ranges along the predicted trajectory.

Although this circular scan technique may theoretically “miss” terrain features located near the intersection of these circles, missed terrain features have not been a problem in practice. If a terrain feature happened to fall into the “crevice” between two scan circles, that terrain feature was always detected by the scan during a following frame. Additional conservatism was built into the scanning technique because the individual DEM posts were represented by rectangular polygons as used by the Auto GCAS algorithm. DEM posts well outside the scan pattern were detected whenever a tiny corner of the rectangular polygon was within any of the scan circles.

The way in which the circular scanning technique identifies the highest terrain post in the vicinity is illustrated in figure 42 by a perspective view of the same trajectory prediction as that shown in figure 41.

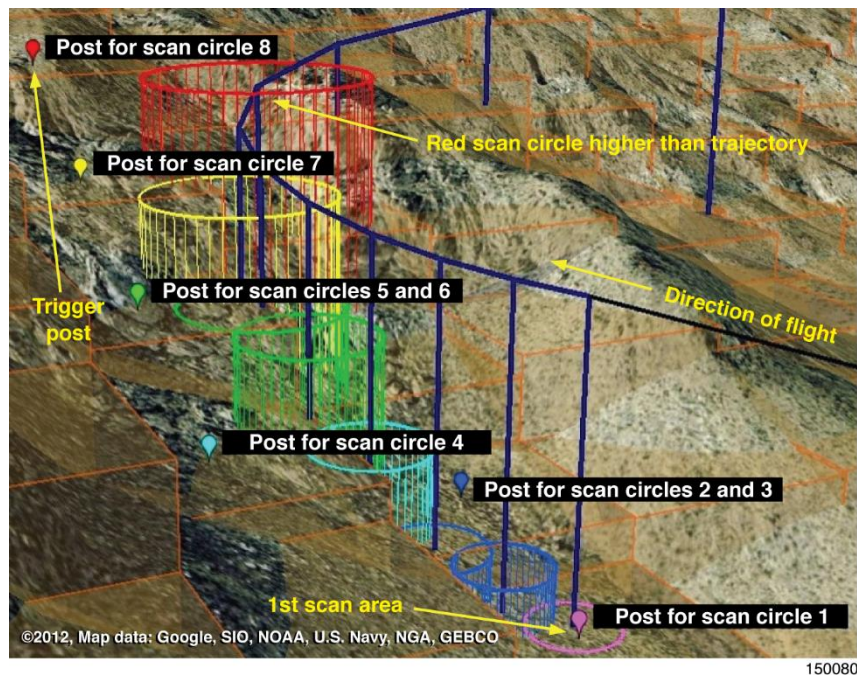


Figure 42. Circular terrain scanning example 2.

The orange polygons in figure 42 represent the way in which the scanning algorithm interprets each terrain post. The relevant terrain posts are represented by the multi-colored balloons in figure 42. The terrain post height is at the cusp underneath each balloon, which is the same height as the corresponding flat, horizontal, orange polygon. Each post has been re-rasterized from the CDTM data.

The smallest circle in figure 42 represents the first terrain scanning area and was closest to the location where an avoidance maneuver was predicted to initiate. The only orange polygon within the bounds of that smallest circle was the one corresponding to the first balloon. Therefore, the height of the circle is set at the same height as the balloon.

Scan circles 2 and 3 (in blue) overlap more than one orange polygon. The highest orange polygon was the one established by the post represented by the dark blue balloon. Therefore, scan circles 2 and 3 are set at the same height as the dark blue balloon.

That pattern continues with the light blue scan circle 4 set at the same height as the light blue balloon. The two green scan circles 5 and 6 are set at the same height as the green balloon, and so on for the yellow scan circle 7 and the red scan circle 8.

The dark blue curving line in figure 42 represents the trajectory prediction for that run. The dark blue vertical lines help visualize the height of the trajectory above the ground at the center of each scan circle. It can be seen that the dark blue trajectory line is above all of the scan circles up to and including the yellow scan circle. Therefore, the terrain represented by all of those circles would not have been high enough to trigger the avoidance maneuver. However, the red scan circle is above the dark blue trajectory prediction, and therefore the terrain post represented by the red balloon is the one that triggered this particular avoidance maneuver.

Also note that the red scan circle in figure 42 is barely touching the orange polygon corresponding to the red balloon. This occurrence was very common. One frame earlier, the trajectory prediction would have been slightly farther from the red post, the red scan circle would have been at a much lower height, and therefore an avoidance maneuver would not have been initiated at that point in time.

All of the scan circles in figure 42 represent no added built-in buffer or TCB. If the simulation of this run had used a higher value for the TCB, the tops of all of the scan circles would have been higher, but the yellow scan circle would probably have been high enough to trigger the avoidance sooner.

Once the information shown in these Google Earth representations was understood, the predicted trajectory for a given simulation run could be assessed very quickly to see if it would yield the desired results. If the resulting avoidance maneuver went in the wrong direction, or was triggered by the wrong piece of terrain, the initial conditions were adjusted until the desired results were obtained.

During early SUAV flight tests, the Google Earth visualization tools were used to identify initial points and aimpoints that would maximize the likelihood that the avoidance maneuver would occur in a predictable direction. The avoidance maneuver direction was not always intuitive because the P-factor effect made the trajectory prediction asymmetric. Test day winds also influenced the avoidance maneuver direction selected by the Auto GCAS algorithm.

As the test team became more experienced with test run setup and at judging the rate at which the test aircraft approached terrain, the visualization tools were used to identify initial points and aimpoints that challenged the Auto GCAS algorithm by making the optimal avoidance maneuver direction more ambiguous, such as aiming directly at the crease in a ridgeline.

The end result of the process was a test matrix that efficiently captured as many setup variations as were practical for a short flight-test project. After the test matrix was established and approved by project management, the Auto GCAS simulation was also used to generate test cards for each individual run in the test matrix.

Test Cards

An example test card is shown in figure 43.

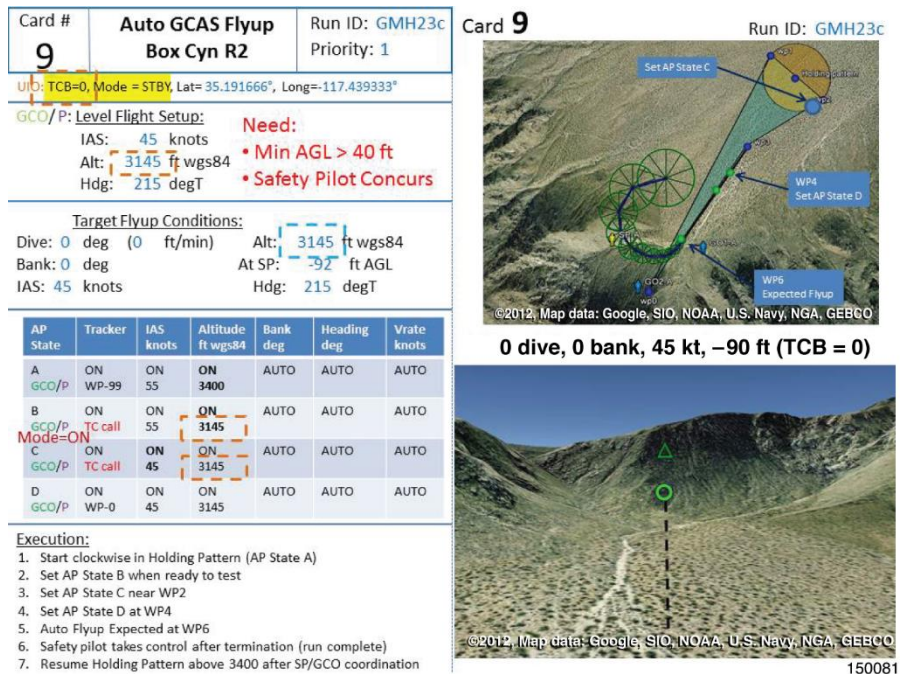


Figure 43. Example test card.

The left-hand side of the test card was primarily intended for the test conductor and the test team in the ground control van. In addition to various test-unique SUAV information, the test card also showed basic setup information that would be useful for any Auto GCAS test, such as the:

- TCB value and coordinates for initiation of the expected avoidance maneuver;
- Specified level flight setup (versus diving);
- Target conditions for the avoidance maneuver;
- Sequence of autopilot states; and
- Test execution sequence.

The upper right-hand side of the test card showed a top-down Google Earth view of the planned test run and was used by all team members to visualize the expected results. In particular, this portion of the test card was intended to help the safety pilot and the ground observers know where the test aircraft would be approaching terrain, and the expected direction of the avoidance maneuver. During test missions, the same waypoints used to generate the test pattern on each individual test card were also replicated on a large high-definition display visible to all members of the test team in the ground control van.

The lower right-hand side of the test card was primarily intended for those runs during which a test pilot was flying the aircraft from the ground cockpit. This Google Earth image provided the test pilot with a view along the planned run-in line (the dashed black line). The green circle showed the pilot an aimpoint for runs at zero TCB. The green triangle showed the pilot an aimpoint for runs with a TCB of 200 ft.

Much of the information shown on the test card was pre-established in pre-configured PCC files for each individual run. Therefore, when the test conductor called for a particular test card number, it was easy to configure the PCC for that run. The test conductor called out each autopilot state at the appropriate point in the test pattern and the GCO commanded the PCC to that state.

The expected coordinates for the avoidance maneuver initiation were entered into a countdown timer calculation on the UIO laptop, and the results displayed to the test conductor. That countdown timer used a simple equation based on the current distance to the initiation point and the current velocity of the test aircraft. The test conductor provided a radio call at 10 s out and then a countdown from 5 s. This method worked reasonably well as long as the test aircraft was somewhat stabilized on test conditions and the winds weren't very high. Even on runs when the countdown timer did not exactly predict the timing of the avoidance initiation, the 5-s countdown helped everyone on the test team know that an avoidance maneuver was imminent.

Data Recording Limitations

The SUAV Auto GCAS project experienced advantages and disadvantages associated with using the Android™ platform smartphone as the primary source of post-flight data. One advantage was that the smartphone provided the ability to record data that was internal to the Auto GCAS algorithm (similar to a data pump on a mux bus). It was not necessary to develop and pay for a separate flight-test instrumentation system for that type of internal test-data recording. Another advantage was that the data were recorded on the same computer that was hosting the Auto GCAS algorithm, avoiding the potential for time synchronization problems that could have occurred with data being recorded on an independent device after being fed through other system components.

The disadvantages have been discussed in the “Host Computer Requirements” section above. In the context of total smartphone storage capacity, the design constraint was that Auto GCAS data were only recorded at the initiation of each avoidance maneuver. The F-16 recorded Auto GCAS data throughout the flight, which was much more useful for assessing nuisance potential in situations in which flyups did not activate. In the context of overall smartphone throughput capacity, the primary design constraint was the rate at which flight-test data could be recorded without interfering with the basic Auto GCAS functions.

The combination of these two design constraints led the team to decide to only record trajectory prediction and scan pattern data at the initiation of an avoidance maneuver. Both of those data types consisted of a moderately-sized array of data. Attempting to record the trajectory prediction and scan pattern arrays at every time frame during the run-in could have stressed both the throughput capacity and the storage capacity of the smartphone. If the throughput capacity were stressed, the Auto GCAS algorithm might not function properly.

The F-16 Auto GCAS was not limited in the same way with respect to recording of trajectory prediction and scan pattern data. The F-16 used an ethernet from the computer hosting the Auto GCAS algorithm to the data recording device. That ethernet path provided sufficient high-capacity throughput to record all of the trajectory prediction and scan pattern data throughout the entire test flight. This capability was a tremendous advantage over the SUAV Auto GCAS implementation because it enabled post-flight data review of maneuvers that came close to a flyup but did not actually initiate a flyup.

Recommendation 12 (R12): Future Auto GCAS test projects using a smartphone as the data recording device should consider implementing a recording method that provides data throughout the flight, not just when avoidance maneuvers occur.

It may not be necessary to record trajectory prediction and scan pattern data throughout the entire test flight. It may be possible to record those data at a slower rate (such as every other frame) or only when the predicted time to the avoidance maneuver drops below a predefined value (such as 5 s). The extra ability to analyze post-flight data would probably be worth the additional design effort.

In order to accommodate the extra throughput that is required, some portion of the standard set of data recorded throughout the test mission may need to be sacrificed. Some of those “always recorded” parameters may need to be deleted, or some may need to be recorded at a slower rate. The standard set of parameters as recorded throughout each test flight for the SUAV Auto GCAS project is given in appendix A.

Test Results

One participant in the SUAV Auto GCAS test project described the results by stating, “It worked surprisingly well.” That somewhat cryptic description is actually a good overall summary, because the system did work better than expected given that the entire project was conceived as a low-budget demonstration effort.

There was never the intent that this demonstration effort directly apply to a production implementation. However, there was always a vision that the results might be good enough to form a basis for follow-on efforts. The results validated the potential for the new technology concepts and opened the door to several follow-on efforts. The overall assessment of SUAV Auto GCAS is divided into the following areas:

- Excellent CFIT protection,
- Adequate nuisance potential, and
- Outstanding modular technologies.

Controlled Flight Into Terrain Protection

There were a total of 61 Auto GCAS events that were included in the post-flight analysis process. This includes almost all test runs on which an avoidance maneuver was initiated by Auto GCAS, but does not include PARS-type avoidance maneuvers. These events do not include another three dozen runs from flights 7, 8, 9, and 10, when the telemetry reception was especially poor (before the directional ground antenna was developed).

Valid initiations occurred on 52 of the 61 runs. The nine invalid initiations were induced by residual telemetry problems caused by incorrect setup or aiming of the directional antenna assembly.

The primary measure of CFIT protection was, “How close did the aircraft get to the rocks?” This question was addressed using the minimum AGL value reached during a recovery. All AGL values, including minimum AGL, were obtained by comparing to the NED truth source as described in the “Comparison to Digital Terrain Truth Data” section of appendix B.

Of the 52 valid initiations, 42 resulted in a usable minimum AGL value. Eight cases did not obtain a usable minimum AGL value because the safety pilot diverted the flightpath before reaching the minimum AGL location. In those cases, the safety pilot took control because the ability to judge terrain clearance was degrading (due to a combination of distance and other terrain in the background), not because of a problem with the trajectory of the Auto GCAS avoidance maneuver. Two of the unusable minimum AGL cases were due to FAIL conditions that were modified in later software updates.

Figure 44 shows “how close the aircraft got to the rocks” using a histogram for the 42 runs that resulted in a usable minimum AGL value.

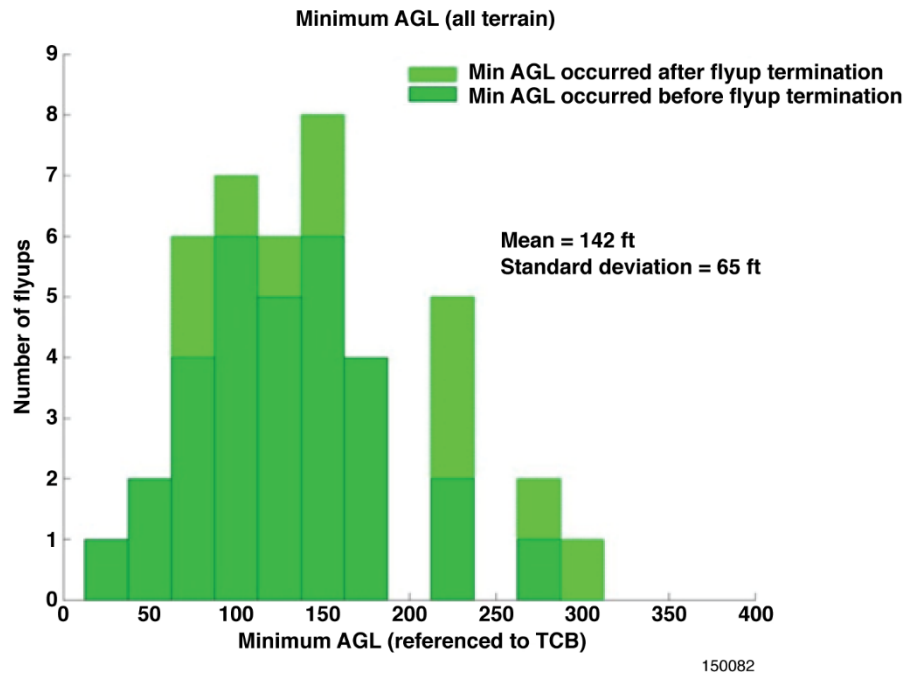


Figure 44. Minimum AGL (mountainous and smooth terrain).

Because the test runs were accomplished in a buildup manner, they tended to progress through TCB settings of 200, 100, and 0 ft. In order to show all of the data on the same plot, the TCB was subtracted from the actual minimum AGL value obtained from each run. This effectively shows “how close the aircraft would have come to the rocks if TCB had been set to zero.”

Most of the runs in figure 44 show that the aircraft cleared the buffered “rocks” by 100 ft or more, with a mean of 142 ft. None of the 42 runs penetrated the TCB. This shows that the combination of the scanning techniques and the built-in buffer of 40 ft worked quite well to minimize the potential for terrain impact.

One run would have been within 25 ft of the “rocks” if the TCB had been set to zero (the test run had a TCB setting of 100 ft). That particular run was one during which the actual trajectory during the avoidance maneuver went far outside the scan pattern at initiation. It is suspected that this deviation was caused by a significant change in winds after the avoidance was initiated. The effects of non-steady winds were discussed above, and warrant additional consideration on future Auto GCAS systems. This case shows the way in which non-steady wind effects could reduce CFIT protection if future Auto GCAS algorithms do not account for those effects.

The data in figure 44 represent all of the usable minimum AGL data regardless of the type of terrain along the flightpath. Almost all of those runs were over mountainous terrain. Only a few runs were conducted over smooth terrain (a dry lakebed) because that terrain was not considered sufficiently challenging to the system. The two runs over smooth terrain that provided usable minimum AGL data are shown in figure 45. As expected, these runs recovered at a minimum AGL that was just above the built-in buffer of 40 ft.

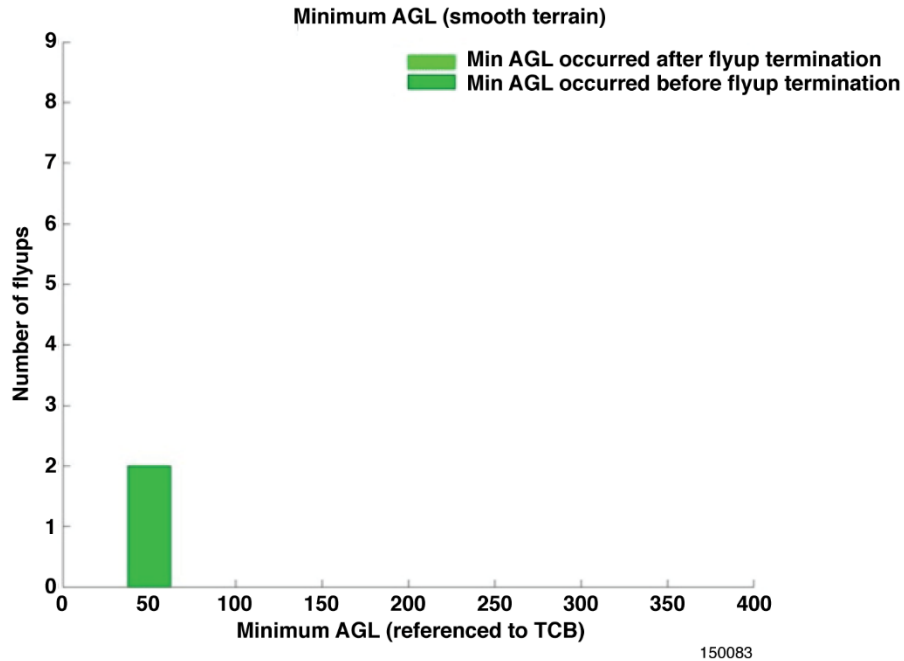


Figure 45. Minimum AGL (smooth terrain).

The data in figures 44 and 45 indicate that the SUAV Auto GCAS demonstrated excellent CFIT protection given the basic design that was implemented for this demonstration effort.

Nuisance Potential

Providing a measure of nuisance potential is more problematic than providing a measure for CFIT protection. Nuisance potential of a generic UAV is a more subjective measure that will almost always depend on the perspective of the individual UAV operator as well as on the mission of a particular UAV.

Most Auto GCAS nuisance evaluations have been accomplished by conducting a series of operationally relevant tasks and asking the pilot to qualitatively evaluate the nuisance potential of any Auto GCAS avoidance maneuvers encountered. That type of subjective evaluation provides a valid overall indication of nuisance potential but does not establish a quantitative boundary for when an avoidance maneuver would be considered a nuisance.

Available Reaction Time

As discussed above, nuisance criterion flight tests were conducted on an F-16 in the 1990s. This study was the only known analytical flight-test study specifically planned to develop a criterion for Auto GCAS nuisance potential. The F-16 study was accomplished by allowing pilots to initiate avoidance maneuvers at progressively lower altitudes until each pilot reached a comfort threshold for that maneuver. The end result was a quantitative nuisance criterion for typical low-altitude maneuvering for F-16 missions.

The F-16 nuisance criterion indicated that any Auto GCAS avoidance maneuver initiated more than 1.5 s earlier than necessary to avoid hitting the ground, would be considered a nuisance by an F-16 pilot aware of ground proximity. That time-based metric was considered much more applicable than a distance-based metric. The term “available reaction time” (ART) has been used as one way to describe that time-based metric.

A similar analytical study to develop an Auto GCAS nuisance criterion for UAVs has not been conducted. It is suspected that the operator of a UAV would accept avoidance maneuver activations at higher ART values than the 1.5-s threshold for F-16 pilots. An Auto GCAS nuisance criterion for a UAV would probably depend on a number of additional relevant factors, such as:

- The size of the UAV;
- The maneuverability of the UAV;
- Whether the UAV is remotely piloted or autonomous; and
- Whether direct line-of-sight control or a satellite link were being used (which determines the amount of delay in the two-way control link).

Since many of these factors will be platform-dependent, it is probably not appropriate to define a generic nuisance criterion that will work for all UAVs. However, it should be practical and necessary to develop a nuisance criterion that makes sense for each platform by using simulators combined with in-flight pilot experience. It will be essential for Auto GCAS designers on future UAV projects to have some kind of guidance regarding nuisance potential so that design tradeoffs can be made.

Recommendation 13 (R13): Future Auto GCAS projects on UAVs should develop a nuisance criterion specific to that project.

An indicator of the nuisance potential as experienced on the SUAV Auto GCAS project can be obtained by inspection of ART. Available reaction time was defined as the amount of time after initiation of the Auto GCAS maneuver that the same maneuver could have been delayed and barely avoided terrain. The premise of the ART calculation was to determine the amount of time available for a pilot to react if an Auto GCAS maneuver were not initiated. A negative ART indicates that the avoidance maneuver would not prevent the aircraft from flying into the terrain, or, in this case, the terrain plus TCB. A formal study has not yet been accomplished to quantify the ART nuisance boundary for UAVs similar to the test aircraft. The ART was not determined from a qualitative pilot opinion but was based on an extrapolation of the actual aircraft states combined with the trajectory of the actual avoidance maneuver for a given test run. The method for determining ART values on the SUAV Auto GCAS project is described in appendix B.

A total of 43 runs provided usable ART values. A number of runs did not provide usable ART values, usually because the safety pilot took control and ART could not be determined from the limited data available for those runs.

The ART values obtained during SUAV Auto GCAS tests are shown in figure 46. Even though these values cannot be compared to a UAV nuisance criterion, they still provide some helpful insight into the overall conservatism of the SUAV Auto GCAS algorithm and the consistency from run to run.

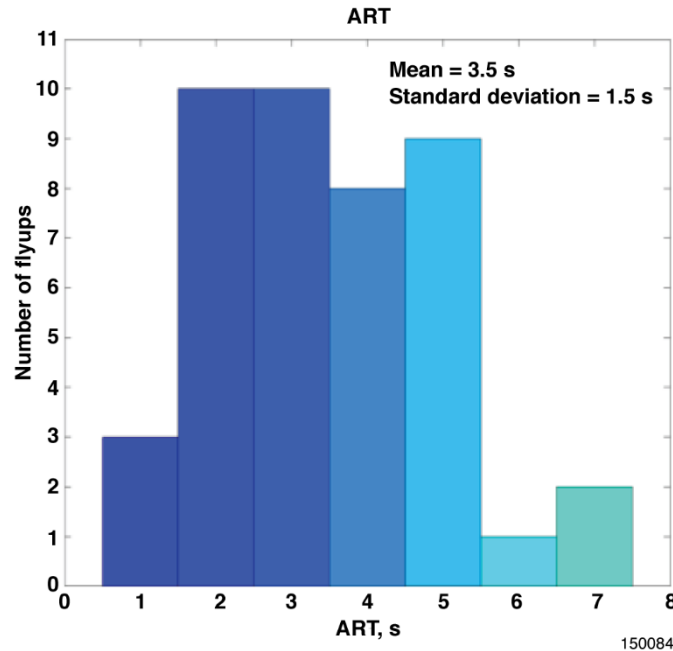


Figure 46. Available reaction time.

As expected, the ART values shown in figure 46 tend to be higher than the 1.5-s criterion of the F-16. The mean ART value of 3.5 s might still be quite adequate for a UAV. The highest ART value (approximately 7 s) might be considered a nuisance for UAVs.

UAV Mission Segments

Since the DROID was not a production platform having a well-defined mission, operationally relevant tasks could not be selected from an existing list. However, as previously discussed, ridge crossings and valley patrols were the two mission segments identified with some operational relevance to UAV missions. Only a few of each task type were conducted near the end of flight-testing because these were considered lower-priority tests for this project.

All of the ridge crossing and valley patrol tests were conducted with the TCB set to zero. Higher TCB settings were not used because the intent was to accomplish these runs as if they were portions of an operational mission in which no flight-test buffers would be added. A buildup approach was used by starting new runs approximately 200 ft above the test card minimum safe altitude that would be used on the final run. This approach provided the pilot in the ground cockpit the chance to become familiarized with the visual cues available on the cockpit video display and to get an idea of the overall setup for each run.

Seven ridge crossings were accomplished on flight 20, including two buildup runs that were 200 ft higher than the final runs. The ridge was crossed in both directions (roughly west to east and then east to west). Five runs were accomplished with the Multi-Trajectory option set to ON, providing the Auto GCAS algorithm with the option to use left, straight, or right avoidance maneuvers. All five of those runs crossed the ridge without any indication of the need for an avoidance maneuver. The aircraft crossed the ridge as close as 196 ft AGL.

Two additional runs were accomplished with the Multi-Trajectory option set to OFF. This setting forced the Auto GCAS algorithm to only use the straight trajectory for avoidance maneuvers. One of these runs crossed the ridge at approximately 175 ft AGL, but without an avoidance maneuver.

The second run with the Multi-Trajectory option set to OFF showed an indication of a straight avoidance maneuver, but the aircraft did not react. This miss was quickly traced to a setup problem with the Gumstix® personal computer. This setup problem did not invalidate the other ridge crossing runs because the main goal was to determine whether nuisance activations would occur. However, this last run with the Multi-Trajectory option set to OFF, did indicate a possible nuisance activation. A pilot assessment would be needed to determine whether that particular avoidance would have been considered a nuisance. If the aircraft had reacted to the initiation, it is likely that the maneuver would have been very short, and the pilot may not have considered it a nuisance given the low altitude.

An example visualization aid for the ridge crossing with the lowest AGL is shown in figure 47. This figure shows the flightpath followed by the DROID, with the superimposed wingspan of an MQ-9 (the white lines between the red and green “wingtips”) to represent a medium-to-large UAV. This figure provides a good perspective of how closely an MQ-9 could have crossed that ridge without triggering an avoidance maneuver.

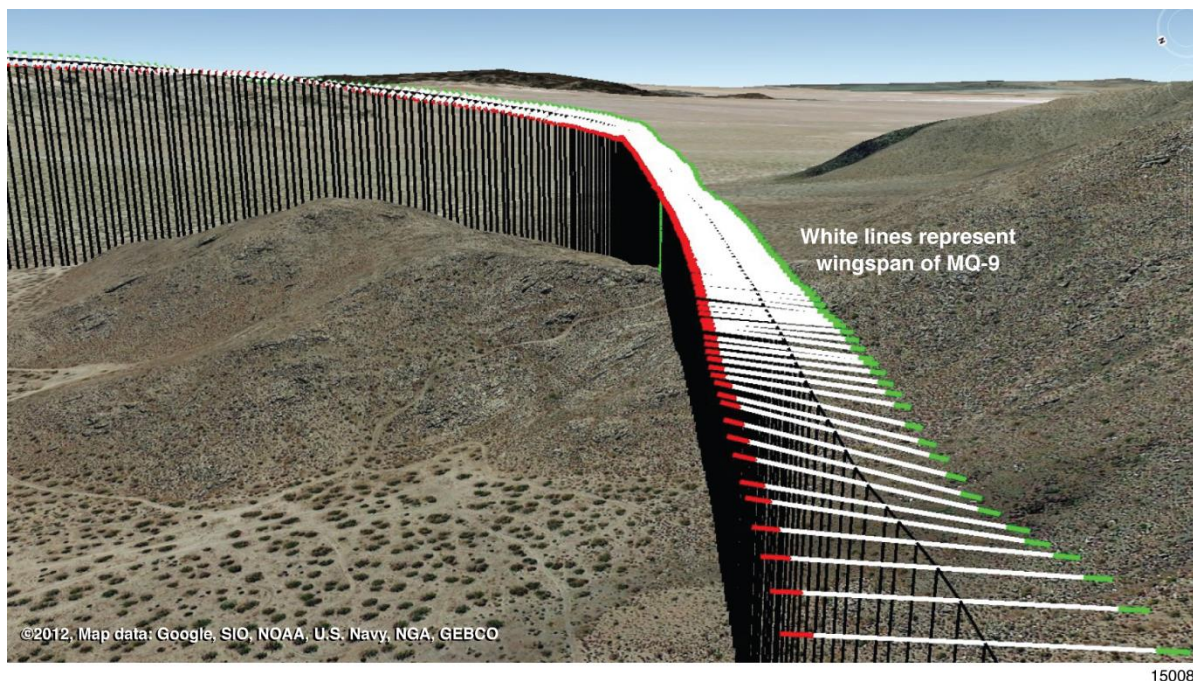


Figure 47. Ridge crossing test.

A few valley patrol tasks were attempted on flight 21. Part of flight 21 had been dedicated to the final collision avoidance runs with the smartphone installed on the test aircraft. Since it was already late in the day, the decision was made to only attempt valley patrol tasks with the Multi-Trajectory option set to OFF. The previous flight had provided confidence that tasks executed with the Multi-Trajectory option set to ON would be unlikely to show any avoidance maneuvers at the planned altitudes (at least 150 ft AGL with no ridge crossings).

The first valley patrol pass was conducted in a counterclockwise direction as a buildup run at 200 ft above the minimum safe altitude shown on the test cards. There were no avoidance maneuver indications

on this pass. While attempting to set up for a lower-altitude pass (at the minimum safe altitude), a straight avoidance maneuver occurred. This avoidance maneuver occurred well away from the safety pilot and terrain clearance could not be properly judged from that distance. In this situation the safety pilot took control and maneuvered the aircraft back toward the holding pattern.

The third valley patrol pass was intended to be a clockwise pattern at 200 ft above the minimum safe altitude. However, during the setup another straight avoidance maneuver occurred. This maneuver occurred even further away from the safety pilot than had the avoidance on the first valley patrol pass, so the test team decided not to attempt any more nuisance evaluations on that flight.

The overall observations made during the valley patrol tests confirmed the general expectations. When the Auto GCAS algorithm was constrained to only use the straight trajectory, the occurrence of avoidance maneuvers was not intuitive and could happen at surprising locations. This occurrence was partially because of the very low climb performance that was being modeled (in order to replicate a medium-to-large UAV). The low climb performance led to avoidance maneuver initiations even when terrain features were several thousand feet away from the test aircraft. The occurrence was made even less intuitive because of the way the P-factor effect was modeled within the Auto GCAS algorithm. As modeled, the P-factor effect caused the “straight” trajectory prediction to be a long, curving flightpath to the left. The pilot could not simply look straight ahead to see possible terrain conflicts because the trajectory as adjusted for the P-factor effect could trigger from terrain well to the side and was not easy to visualize.

Another factor that diluted these nuisance evaluations was the setup of the ground cockpit. In an ideal setup, control would be returned directly to the pilot immediately after an avoidance maneuver had completed, but because of the way in which the ground cockpit was set up during these tests, when an avoidance maneuver was complete, control reverted to the GCO. The pilot could resume control by hitting a button on the ground cockpit, but it was not always obvious when that button was active. This setup could have been easily improved, but with cost and schedule impacts to the ground control station that were determined to be out of scope.

Avoidance Maneuver Termination

An Auto GCAS nuisance evaluation can also be influenced by the timely termination of avoidance maneuvers. If an avoidance maneuver lasts too long, it will undoubtedly be considered a nuisance. Conversely, even if a pilot feels an avoidance maneuver may not have been necessary, it may not be viewed as a nuisance as long as the maneuver was very short in duration (pilots have described this occurrence as a “speedbump”).

During the F-16 Auto GCAS FRRP (ref. 2), a quantitative criterion was used to assess flyup maneuver duration. That criterion was originally developed during F-16 Auto GCAS flight-testing in the 1990s. The resulting criterion specified that a flyup must terminate when the flightpath had cleared the terrain immediately ahead of the aircraft by less than 5 deg of overshoot. This flightpath-based criterion was applicable to the F-16 since it only used straight flyup maneuvers, and the aircraft was usually near wings-level when it cleared significant terrain features. It now appears that the production F-16 version of Auto GCAS will be able to meet that criterion, and related F-16 pilot comments have been very favorable. However, a similar criterion does not yet exist for the timely termination of UAV avoidance maneuvers.

Recommendation 14 (R14): Future Auto GCAS projects on UAVs should develop a termination timeliness criterion specific to that project.

The SUAV Auto GCAS mechanization added another dimension to a termination assessment because of the options for turning avoidance maneuvers. Since most of the SUAV avoidances were turning

maneuvers, the most relevant angular measure was heading change. In addition, UAV pilots may be able to tolerate more angular termination overshoot in terms of heading change than F-16 pilots in terms of flightpath angle.

A termination analysis was accomplished for every run that continued all the way through normal termination (without the safety pilot taking control). Of the 52 valid initiations, 28 continued all the way through normal termination without being interrupted by the safety pilot or experiencing telemetry dropout. On each run, an ideal termination heading was calculated using a simple extrapolation of the tangent line at each point along the turning avoidance maneuver. The ideal heading was defined as the point at which that tangent line first became clear of terrain. That ideal heading calculation included three frames of persistence, for consistency with the on-board termination calculation. The ideal heading was compared with the actual heading at termination to obtain a delta heading.

The specific methodology is described in appendix B and an example is shown in figure 48. The semi-horizontal black and red lines show the trajectory of the test aircraft prior to and after the avoidance maneuver initiation, respectively. The vertical black and red lines show the update rate at which the algorithm was calculated in that segment. The green lines show tangential extrapolations at key points along the trajectory. Those extrapolations included a vertical component that was based on the climb rate at that point. This example shows how straight tangent lines were projected from the turning maneuver trajectory until all terrain within approximately 3 nm was cleared for three sequential frames. The distance of 3 nm was selected to be consistent with the normal Auto GCAS algorithm, but could be tuned for specific platforms. The projected tangent lines slope up in order to include the climb rate established at that point in the avoidance maneuver.

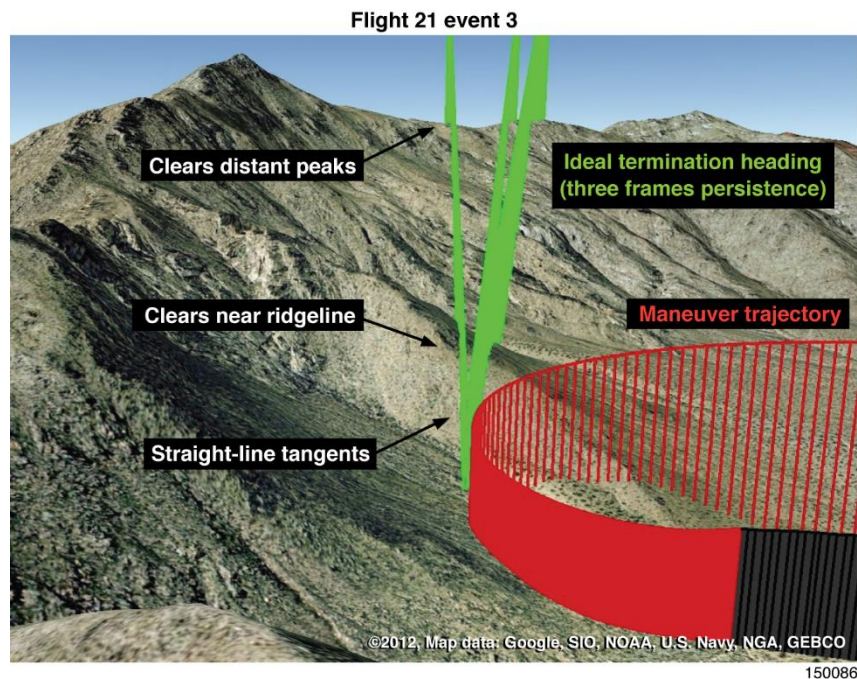


Figure 48. Ideal termination heading.

Despite the lack of a formally-developed termination criterion for UAVs, a reference of some kind was still needed to help interpret the test results. It was assumed that a UAV pilot would tolerate quite a bit more angular overshoot than the 5-deg flightpath overshoot that worked well for the F-16 pilots. Therefore

a “Good” termination was defined to be no sooner than the ideal heading, but less than 15 deg past that ideal heading. The results are shown in figure 49.

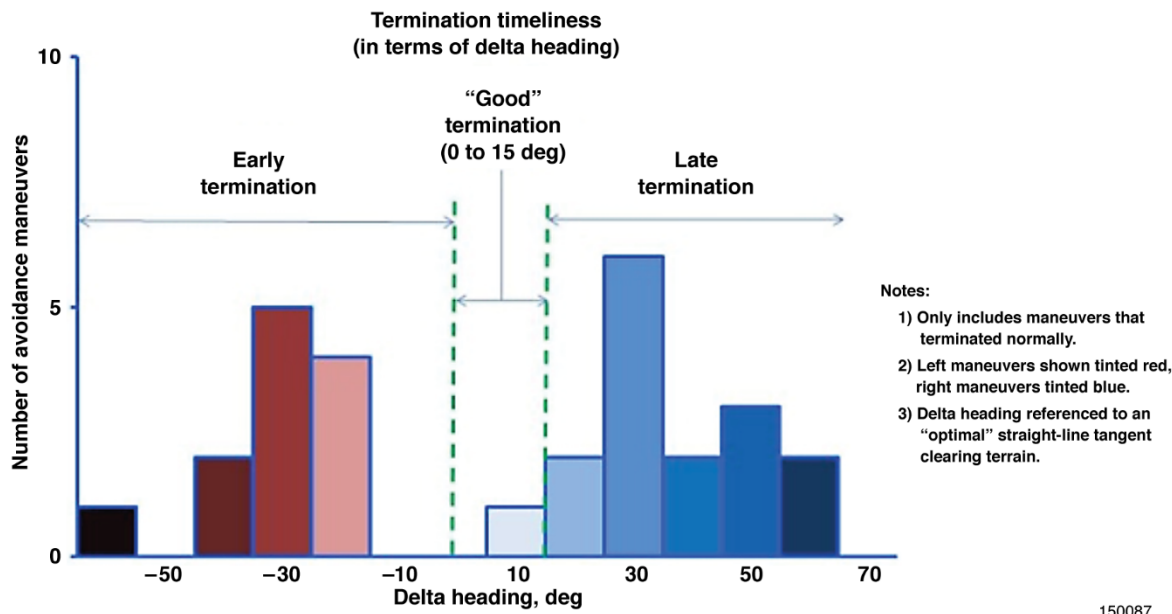


Figure 49. Termination timeliness: heading.

The data in figure 49 show that there was only one avoidance maneuver that would have been considered “Good” using the 0- to 15-deg delta heading criteria. The remaining maneuvers were considerably late or early. This undesirable result can be attributed to the way in which the P-factor effect was used as part of the termination logic.

When the DROID P-factor effect was initially quantified, the decision was made to incorporate that characteristic as part of the trajectory prediction algorithm instead of taking the extra time to try to design a set of autopilot commands to compensate for the P-factor effect. This approach also influenced the termination logic because the straight trajectory (which is a curving trajectory due to the P-factor effect) was used to determine when the aircraft was clear of terrain.

An example of how the P-factor effect induced late terminations is shown in figure 50. The semi-horizontal black and red lines show the trajectory of the test aircraft prior to and after the initiation of the avoidance maneuver (as in figure 48). The green line shows the tangential extrapolation of the ideal termination (three sequential frames clear of terrain). The transition from the red line to the blue line shows the point at which the flight-test maneuver was terminated. The blue line represents the straight trajectory (as influenced by the P-factor effect) that was used by the Auto GCAS algorithm to determine when the avoidance was terminated. Even though the ideal green tangent line for this maneuver was clear of both the near ridgeline and the more distant peaks, that maneuver did not terminate until the blue straight trajectory (as influenced by the P-factor effect) was clear of the terrain. The specific features of the terrain in the test area tended to amplify this effect, but the overall result tended to be a significant delay in termination that would most likely be considered a nuisance, the maneuver lasting much longer than necessary.

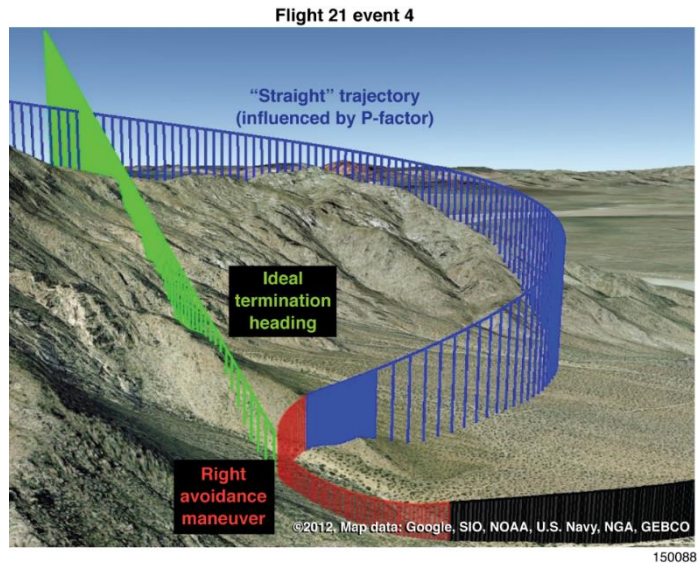


Figure 50. Late termination example.

An example of how the P-factor effect induced early terminations is shown in figure 51. The semi-horizontal red line shows the trajectory of the test aircraft during the avoidance maneuver. The transition from the red line to the blue line shows the point at which the flight-test maneuver was terminated. The orange line shows the tangential extrapolation of the trajectory at that point. The green line shows the tangential extrapolation of the ideal termination (three sequential frames clear of terrain). In this case, the blue curving trajectory (as influenced by the P-factor effect) caused the termination logic to see the gap in terrain before the aircraft was pointed toward that gap, causing the avoidance maneuver to terminate while the aircraft was still pointed at terrain (orange line). In several cases the aircraft did not continue in the turn, which triggered the almost-immediate initiation of additional avoidance maneuvers. Although not a direct safety hazard, when additional avoidance maneuvers occur immediately after a previous avoidance maneuver has terminated, the result creates an impression of reduced confidence in the Auto GCAS algorithm, and would probably be considered a nuisance as compared to a single, longer avoidance maneuver.

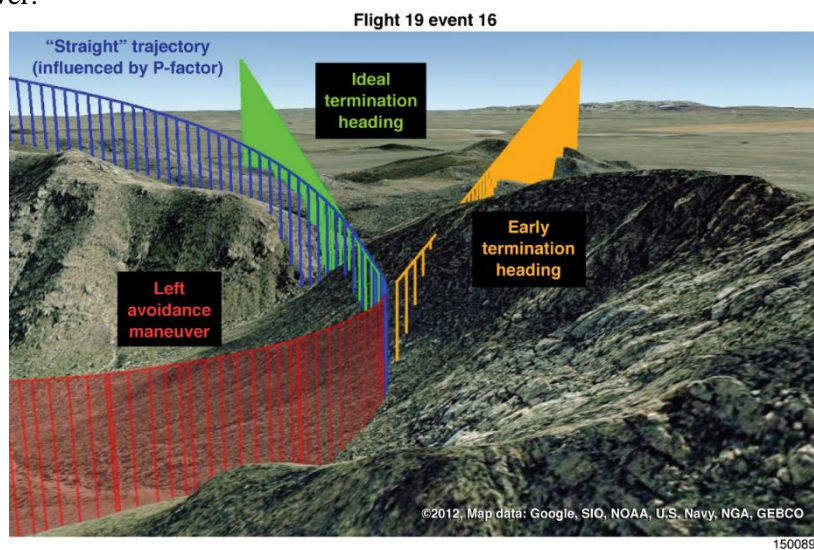


Figure 51. Early termination example.

Given the overall undesirable results, it may not be warranted to include the P-factor effect in the termination logic even if the P-factor effect is considered in the development of the trajectory prediction. This rationale is based on the principle that an aircraft should be considered clear of terrain when it can roll out of the turn and continue in a straight climb without intersecting the digital terrain.

Recommendation 15 (R15): Future Auto GCAS projects on propeller aircraft should determine if the use of the P-factor effect in the termination logic is warranted on that platform.

Direct pilot evaluations of the termination characteristics were not obtained because nuisance evaluations were only accomplished on two flights and no avoidance maneuvers were initiated when the Multi-Trajectory option was set to ON. In addition, the ground control van mechanization did not return control directly to the pilot, so any qualitative termination evaluation would have been skewed when compared to a more representative mechanization. Pilots flew some of the collision avoidance test points, but the focus was on getting the system to function as desired, and not on a thorough evaluation of termination timeliness.

Termination timeliness was analyzed in terms of delta time in addition to the delta heading analysis. A “Good” termination was defined to be no earlier than the time at which the ideal heading was achieved, and less than 1.5 s beyond that time (1.5 s correlates to the 15-deg delta heading based on a nominal turn rate of 10 deg/s during these avoidance maneuvers). The results are shown in figure 52.

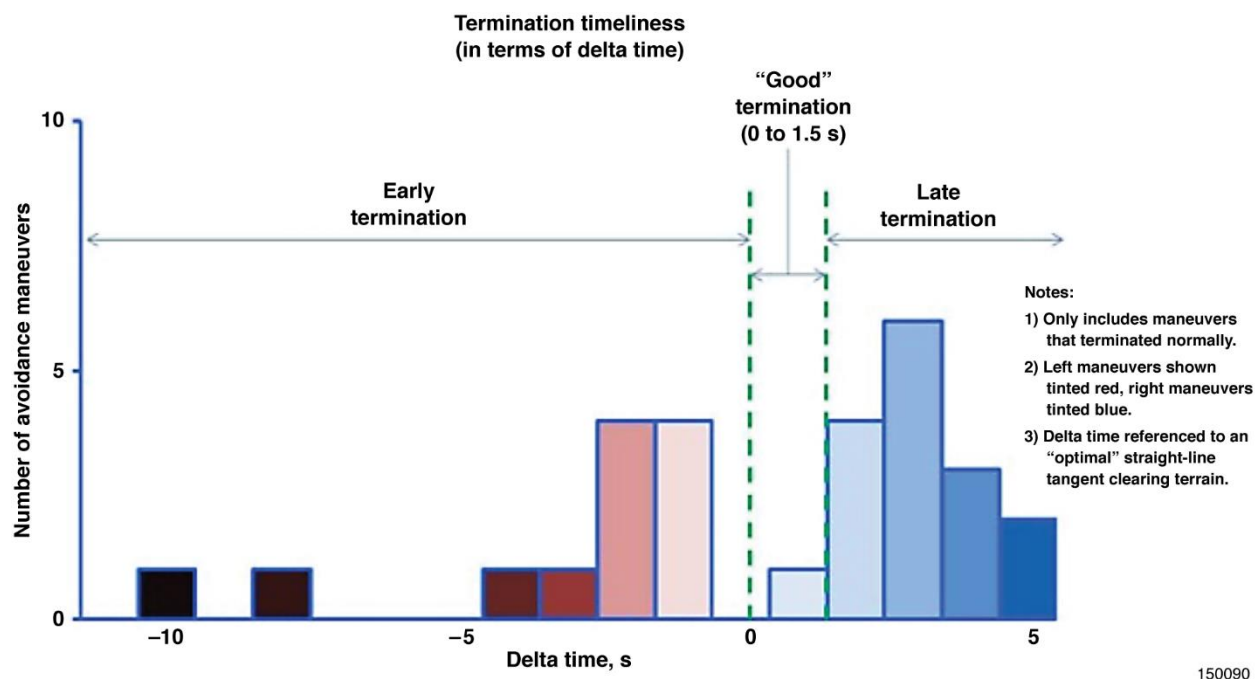


Figure 52. Termination timeliness: time.

Using the delta time criterion also results in a single maneuver that would have been considered “Good.” The remaining maneuvers reflect the undesirable effects of the P-factor. The data depicted in figure 52 indicate that termination timeliness was generally between 1-5 s early for left turns, and 1-5 s late for right turns. Two outliers are shown terminating early by approximately 8 and 10 s. These two outliers show that the method for calculating the ideal heading may be susceptible to using instantaneous values for the extrapolated flightpath. Future projects implementing similar analysis techniques should consider using a

five-point moving average instead of the instantaneous values (which may be more susceptible to gusting winds).

The overall conclusion from this termination analysis is that the designers for each platform will need to carefully consider the logic to be used for when an avoidance maneuver should be terminated and control returned to the pilot.

Recommendation 16 (R16): Future Auto GCAS projects should consider implementing termination logic that returns control to the pilot when well clear of terrain in the immediate vicinity, but should not be overly conservative for distant peaks.

Trajectory Prediction Accuracy

The accuracy of the trajectory prediction can be assessed by answering two fundamental questions:

- How well did the trajectory prediction at initiation compare to the flightpath of the actual avoidance maneuver?
- Did the flightpath of the actual avoidance maneuver stay within the scan pattern used at initiation?

These questions are addressed by utilizing the analyzed data from individual test runs as in the example in figure 53 to create the summary data that are depicted in figure 54. A more complete description of the technique that was used is presented in appendix B.

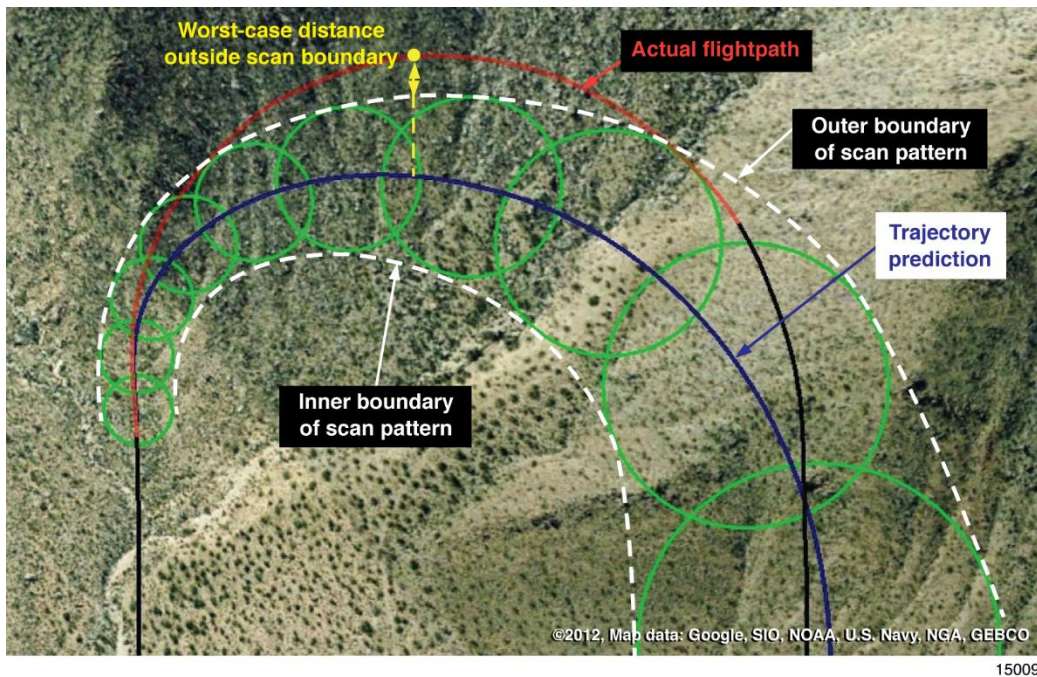
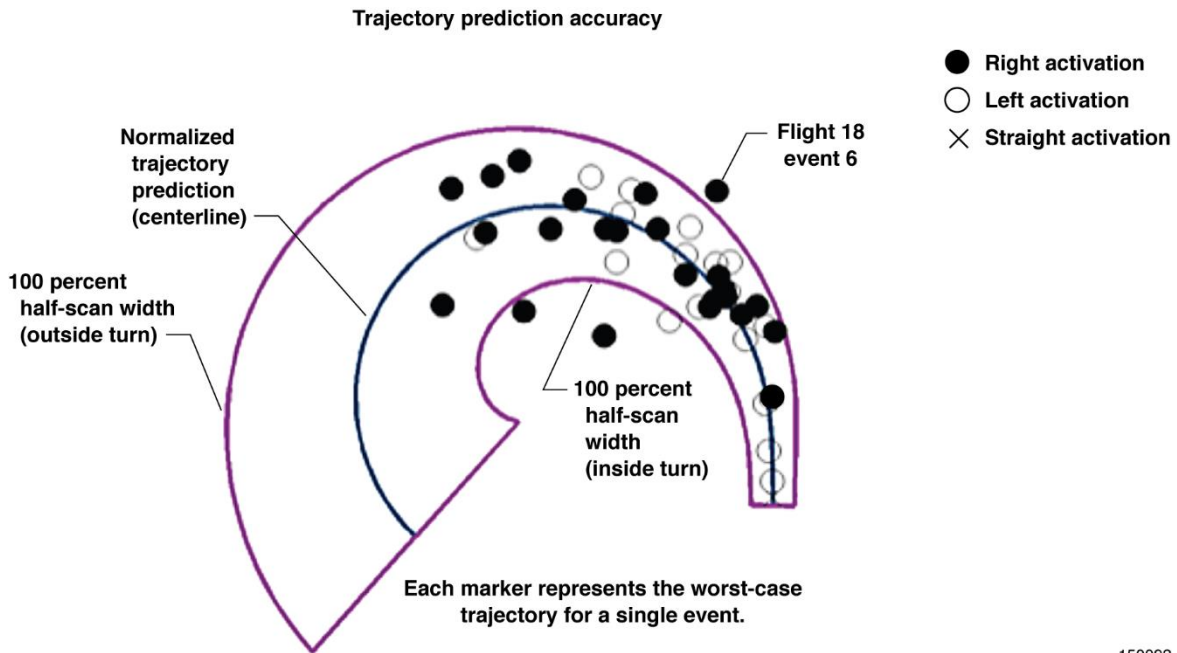


Figure 53. Worst-case trajectory example.



150092

Figure 54. Trajectory prediction accuracy.

Figure 53 shows the worst-case example from all test runs. On this run (flight 18, event 6) the actual flightpath went the furthest outside of the scan pattern on the side closest to terrain. The blue line shows the trajectory prediction at initiation. The green circles show the scan pattern at initiation, with the dashed white line representing the outer edges of that scan pattern. The red line shows the actual flightpath during the avoidance maneuver. The yellow dot identifies the point where the actual flightpath went the furthest outside the scan pattern.

The situation shown in figure 53 is undesirable because the aircraft could have flown close to terrain features that might not have been detected by the scan pattern. In this case, the resulting minimum AGL was 125 ft, but that test point had been executed with the TCB set to 100 ft. If the TCB had been set to zero, terrain clearance would have been approximately 25 ft. This was the lowest minimum AGL for any test run, and it was considerably lower than the more typical clearance at 100 ft or greater.

In a production Auto GCAS implementation this characteristic could degrade CFIT protection. Limited analysis indicates that this worst-case example was probably induced by changing winds during the avoidance maneuver. The onboard wind calculations from the Piccolo II autopilot showed a significant shift in wind magnitude and direction during the maneuver. However, those onboard wind calculations become more suspect when the aircraft is not in straight-and-level flight. If confirmed, this characteristic warrants additional effort to account for changing winds through increased buffers, a wider scan pattern, or some other method.

Recommendation 17 (R17): Future Auto GCAS projects should determine if changing winds during the recovery should be addressed in the design for that platform.

A case-by-case analysis of each test run is useful, but it is even more helpful to look at summary data that encompass all of the relevant test results. The individual run analysis technique presented in figure 53 was applied to all relevant test runs to obtain the summary data depicted in figure 54. Each symbol in

figure 54 represents a single test run at the point along the actual flightpath that was furthest away from the trajectory prediction.

Although the scan pattern size and shape varied based on airspeed, wind, and other factors, the data were normalized to a generic scan pattern by using a percentage deviation from the centerline trajectory prediction. At any given distance along the centerline, 100 percent was located on the scan pattern line (perpendicular to the centerline) and 50 percent was located halfway between the centerline and the scan pattern line.

The normalized data depicted by figure 54 also include test runs that resulted in right avoidance maneuvers (shown with closed symbols) even though the generic scan pattern shown is for left maneuvers (shown with open symbols). The data for right maneuvers was included by flipping the x-axis for those test runs. This method maintained the relative position of each dot in the sense that any symbols below the centerline in figure 54 were inside the turn, whereas any symbols above the centerline were outside the turn. The specific normalization analysis techniques are described in appendix B.

Figure 54 addresses the two fundamental questions about the accuracy of the trajectory prediction. The summary data given in figure 54 show that most of the test runs flew reasonably close to the centerline of the trajectory prediction, and most of the worst-case locations were within the scan pattern. Another way of showing the same trend is presented in figure 55.

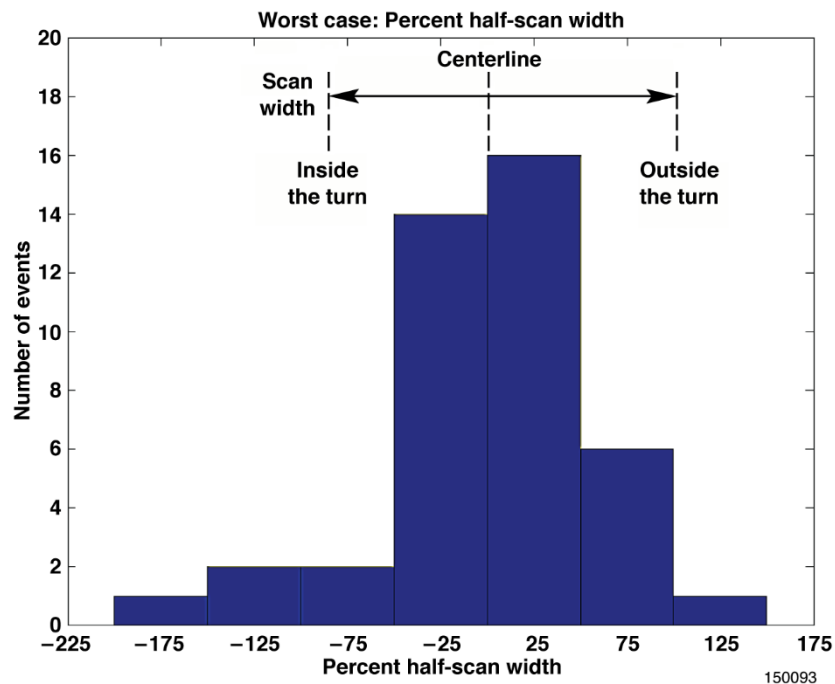


Figure 55. Trajectory prediction histogram: across the scan pattern.

Figure 55 shows the same data from figure 54 represented by a histogram indicating the worst-case position as a percentage of the scan width relative to the centerline. Almost half of the worst-case positions were less than 40 percent of the scan pattern width. Most runs stayed with the scan pattern for the entire avoidance maneuver. Only a few runs strayed outside the scan pattern (as indicated on the histogram for values greater than 100 percent).

Only one case went beyond the scan pattern on the outside of the turn (closer to terrain). That run was at 132 percent relative to the outer scan pattern line and occurred during event 6 of flight 18 (fig. 53).

Three cases went beyond the scan pattern on the inside of the turn (farther from terrain). These three runs ranged from 117 percent to 197 percent relative to the inner scan pattern line. Similar to event 6 of flight 18, it is suspected that variable winds during these maneuvers contributed to flying beyond the scan pattern. These cases do not necessarily indicate degraded CFIT protection, but there could be increased nuisance potential if this situation is not adequately addressed in a production implementation. The only situation in which these three cases might result in degraded CFIT protection would be if the aircraft were descending into terrain that happened to enter the scan pattern inside of the predicted avoidance maneuver (at the same time as when the other avoidance directions were unavailable). That situation may be possible but is probably unlikely.

Figure 56 shows the same data from figure 54, represented by a histogram indicating where the worst-case locations occurred relative to the distance (range from initiation) along the centerline. This histogram shows that most of the worst-case locations were within the first third of the total scan length. The same trend can be seen in figure 54. This does not imply that the scan length was longer than necessary. The avoidance maneuvers for these particular test situations were completed relatively quickly, and therefore any trajectory errors did not have time to grow larger. Different test situations could lead to trajectories that last longer and therefore require the longer scan length.

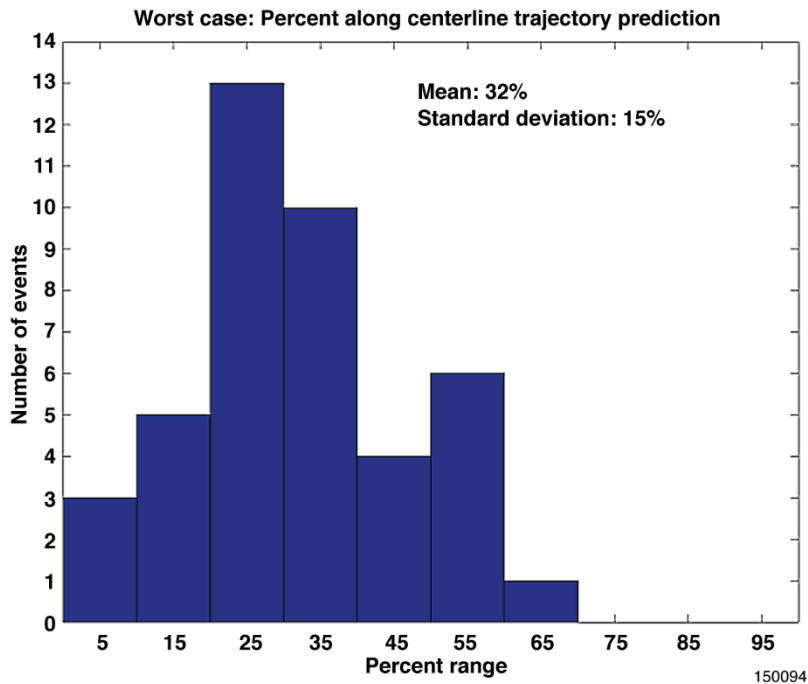


Figure 56. Trajectory prediction histogram: along the centerline.

Scan Pattern Functionality

The functionality of the scan pattern can be assessed by answering two fundamental questions:

- 1. What was the location of the terrain post that triggered the avoidance maneuver (relative to the scan pattern)?

- 2. Did the flightpath of the actual avoidance maneuver stay within the scan pattern used at initiation? (Note: this is the same question that was used to assess the trajectory prediction accuracy; the trajectory prediction accuracy and scan pattern functionality modular components are inter-related.)

These questions are addressed by utilizing the analyzed data from individual test runs similar to the example presented in figure 57 to create the summary data that are depicted in figure 58. The individual run in figure 57 is the same run that was shown in figure 53. Additional data is presented in figure 54. The terrain post that triggered the avoidance maneuver is represented by the orange balloon in figure 57. The method used to determine the specific location of that trigger post for a particular run is described in appendix B. In the SUAV Auto GCAS algorithm, the terrain posts were not used directly, but rectangular areas around each post were created at the same height as that post. The rectangular area is represented by the orange box in figure 57.

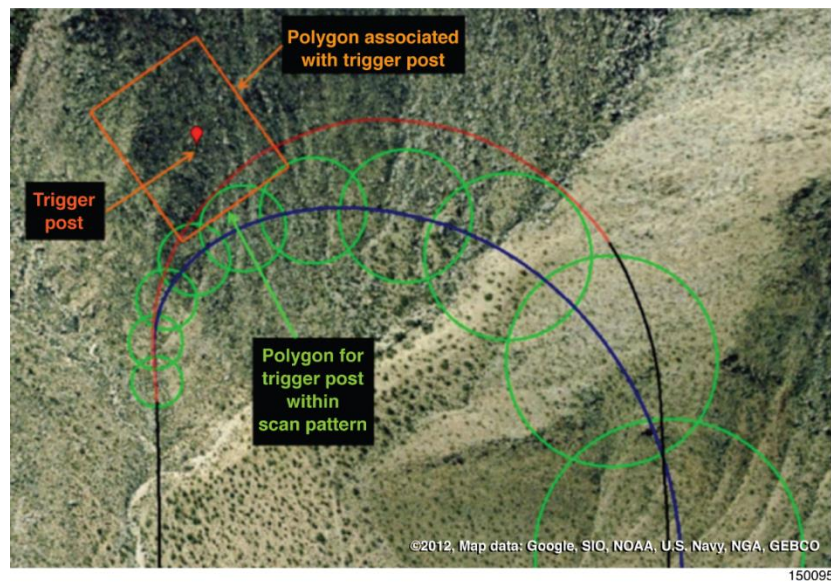


Figure 57. Scan pattern functionality.

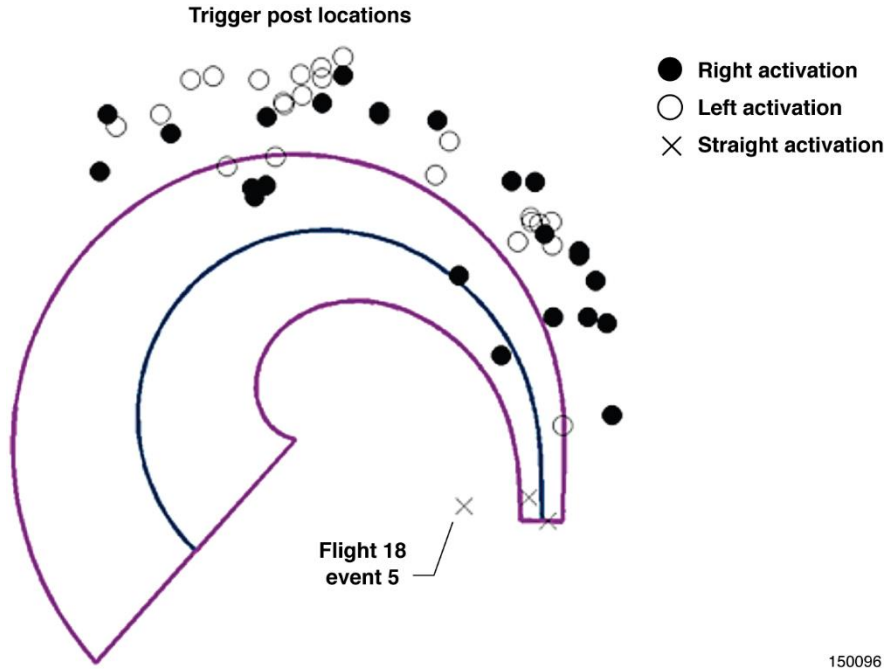
An avoidance maneuver was triggered as soon as one of the rectangular terrain polygons was within one of the green scanning circles at a height above the trajectory prediction (including any built-in or flight-test altitude buffers added to the terrain height).

In figure 57 the orange terrain polygon was within three of the green scan circles. In this case, the buffered terrain height was above the trajectory prediction at all three of those scan circles, so the avoidance maneuver was triggered by whichever was first in the internal sequence of checks. One frame earlier (approximately one-fifth of one second), the path of the trajectory prediction was far enough away from that particular terrain polygon/post so that an avoidance maneuver was not triggered.

The main point to be gleaned from figure 57 is that the individual terrain post that triggered this flyup maneuver was outside the scan pattern. As will be seen in figure 58, this result was fairly common, and the terrain scanning approach was implemented as a conscious part of the design process to ensure that terrain posts were not missed.

The summary data presented in figure 58 were constructed using the same type of normalization process as applied to those data depicted in figure 54. As can be seen, almost all of the trigger posts were outside

the scan pattern (on the side closer to terrain). A small number of trigger posts were within the scan pattern, and one was outside the scan pattern on the inner side of the turn. These occurrences were generally when the test aircraft was in a dive, so that the scan pattern or trajectory prediction descended down onto the highest terrain post, as compared to the more typical case in which the test aircraft flew in level flight toward the trigger post.



150096

Figure 58. Trigger post locations.

The mean for trigger post locations was at 175 percent relative to the scan width for the outer scan line. The trigger posts were usually outside the scan pattern because of the scan technique that included any portion of a terrain rectangle within the scan pattern (fig. 28). There was no significant difference in trigger post location for left or right avoidance maneuvers.

The trigger post with the greatest percentage outside the scan pattern was at 366 percent relative to the scan width for the inner scan line. That seemingly extreme case was from event 5 of flight 18, which was a straight activation from a diving maneuver over smooth terrain. That straight activation reflected the general trend expected from diving maneuvers over smooth terrain: the straight trajectory prediction climbs at 1000 fpm whereas the left and right predictions climb at 800 fpm. Over smooth terrain the left and right trajectories will be lower and intersect with smooth digital terrain before the straight trajectory, resulting in a straight trajectory as the last viable option.

In addition, the same trend resulted in trigger posts quite close to the initiation point because the straight trajectory prediction curved upwards quite rapidly compared to the essentially flat digital terrain. When the trigger post occurred close to the initiation, the very narrow scan pattern detected the tiny corner of a terrain rectangle generated from a post that was well outside the scan pattern compared to the scan width.

Figure 59 shows the same data from figure 58, represented by a histogram indicating the trigger post locations as a percentage of the scan width relative to the centerline. The data in figure 59 resulted in a mean value of 175 percent scan width and a standard deviation of 71 percent. The straight activation at

366 percent was an outlier and was not included in this histogram, but the value was included in the calculation of the mean.

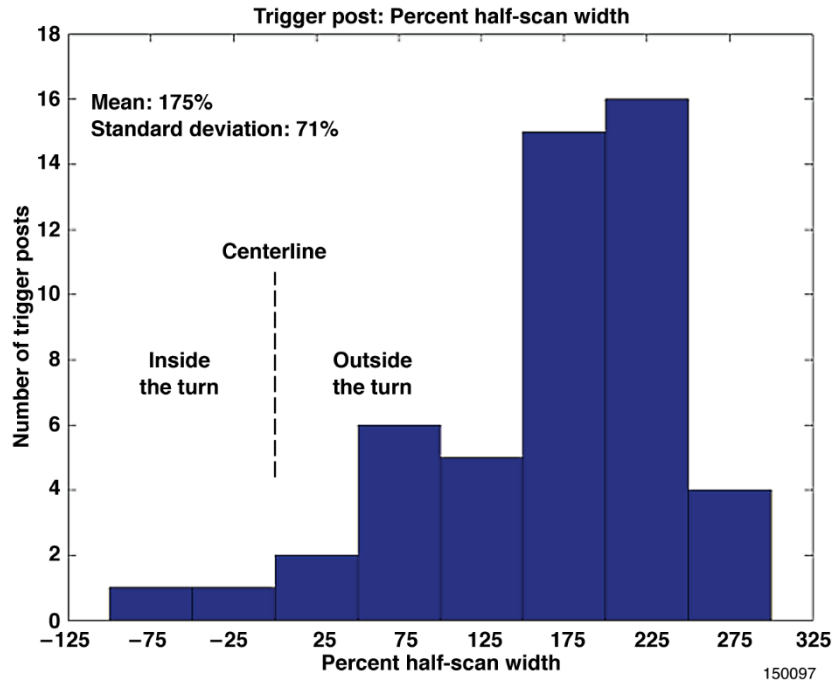


Figure 59. Trigger post location histogram: across the scan pattern.

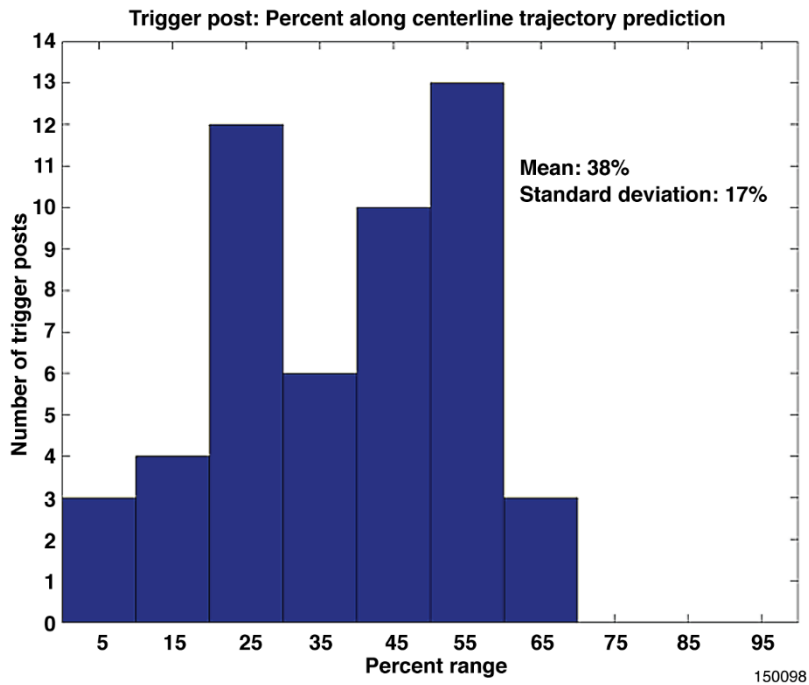


Figure 60. Trigger post location histogram: along the centerline.

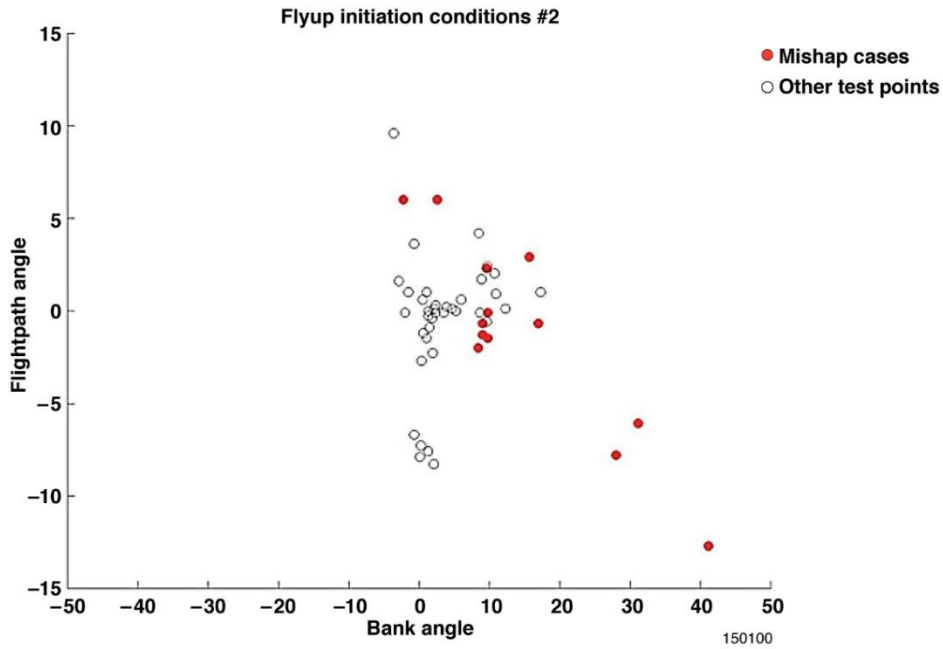


Figure 62. Initiation flightpath versus bank angle.

Figure 62 shows the bank angle variations at initiation. Most of the runs were near wings-level, but a few were initiated at approximately 30-40 deg bank. The three mishap cases targeted 0 deg, 15 deg, and 25 deg of bank. Although one test card sequence targeted negative bank angle, that series was only attempted on flight 17 and the resulting data were not usable due to poor telemetry on that flight.

Almost all of the avoidance maneuvers were to the left or the right, as shown in figure 63. The data in figure 63 include all valid initiations, even if the safety pilot took control later.

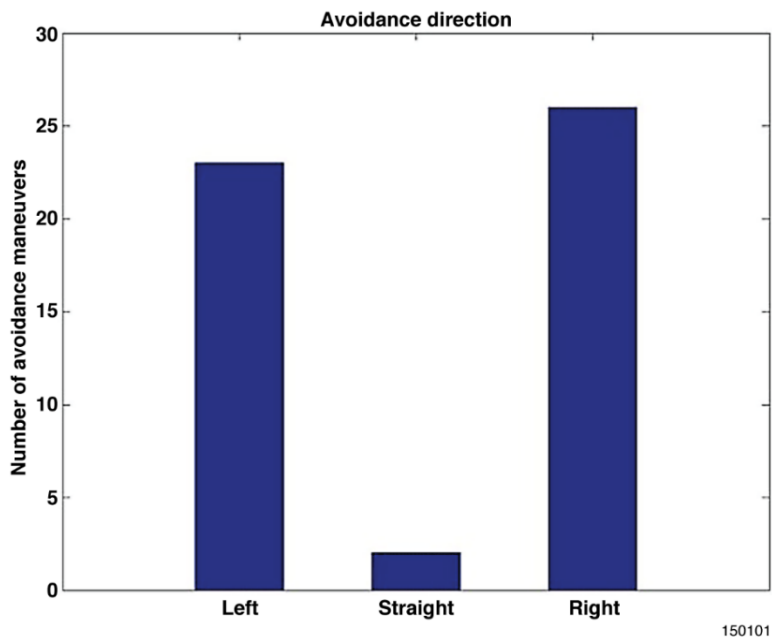


Figure 63. Avoidance maneuver direction.

Only two test setups resulted in straight avoidance maneuvers with usable data. Straight avoidance maneuvers were only expected to occur over smooth terrain, or possibly during ridge crossings. A number of smooth-terrain test points were attempted, but most of those occurred on missions with poor telemetry and resulted in unusable data. The Auto GCAS algorithm initiated one straight avoidance maneuver during a ridge crossing on flight 20, but the Gumstix[®] personal computer was not set up correctly on that mission and the autopilot did not respond to the commanded maneuver.

The longest avoidance maneuver was approximately 15 s and the shortest approximately 1 s, as shown in figure 64. The duration was a simple calculation that began when the Auto GCAS algorithm initiated the avoidance maneuver and ended when the algorithm determined the maneuver was complete. The data in figure 64 do not include runs in which the safety pilot took control prior to normal termination.

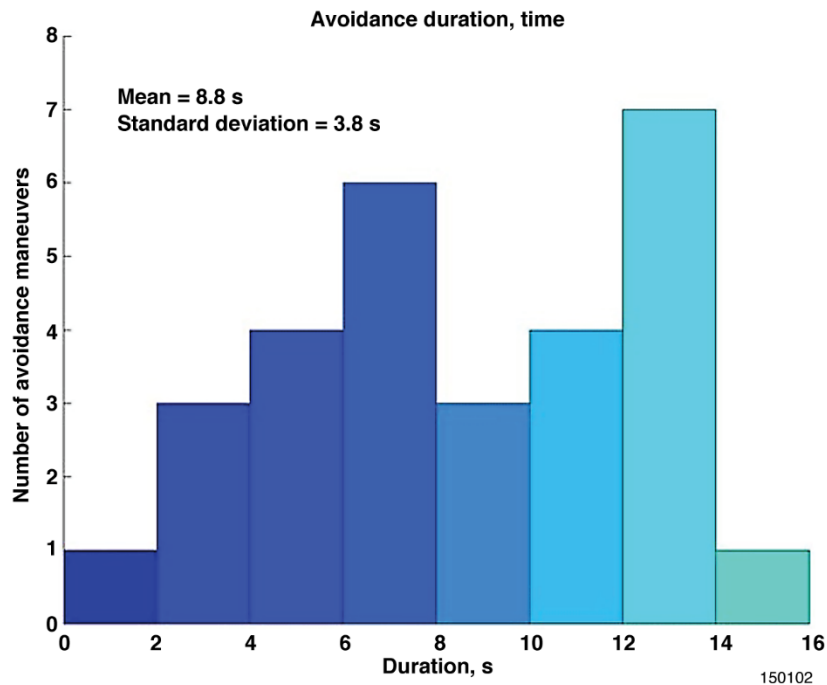


Figure 64. Avoidance maneuver duration.

There was considerable variation in avoidance maneuver duration with no discernible pattern. This variation was not surprising, because the duration was solely driven by the amount of time needed for the aircraft to clear the terrain. Many of the test setups were intentionally targeted at a bowl-shaped canyon in order to stress the system. That bowl-shaped canyon tended to force the aircraft to “go back toward the way it had come,” resulting in quite a few avoidance maneuvers each lasting longer than 10 s. The duration of avoidance maneuvers was also influenced by the way the P-factor effect was modeled in the termination logic.

The amount of heading change that occurred during avoidance maneuvers is shown in figure 65. The data in figure 65 only include turning maneuvers, and only include runs that continued to normal termination before the safety pilot took control.

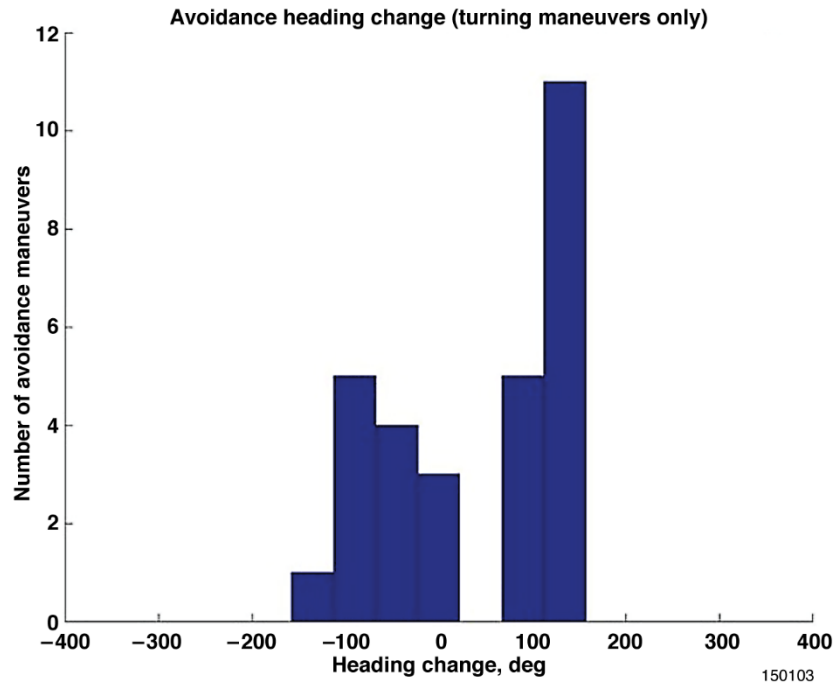


Figure 65. Avoidance maneuver heading change.

The effect of the P-factor bias can be seen in figure 65. Left turns had a mean heading change of approximately 80 deg, whereas right turns had a mean heading change of approximately 110 deg. Left turns tended to have less heading change and right turns more heading change. It was not unusual for avoidance maneuvers to continue for 90 deg heading change or more.

Concluding Remarks

The small unmanned aerial vehicle (SUAV) Automatic Ground Collision Avoidance System (GCAS) project successfully demonstrated many important collision avoidance technologies. Foremost among these demonstrations were:

- Auto GCAS testing with multiple avoidance maneuvers including turns to either side;
- Testing of digital terrain scanning techniques determined directly from the predicted trajectory;
- In-flight testing of highly compressed digital elevation models;
- In-flight testing of digital elevation models that had been customized to reflect tighter tolerances in some areas and relaxed tolerances in other areas;
- In-flight testing of Auto GCAS on an unmanned aerial vehicle; and
- Hosting Auto GCAS algorithms on a smartphone during flight tests.

Additional noteworthy accomplishments included:

- Design and implementation of a flight-test user interface that enabled ground operators to control the Auto GCAS algorithms on the smartphone (either with the smartphone on the ground or on board the test aircraft);
- Rapid design and implementation of a directional antenna system to avoid multi-path noise when in the vicinity of terrain, which greatly improved communications between the test aircraft and the ground control van;

- Derivation of Dryden Remotely Operated Integrated Drone trajectory prediction models purely from very limited flight-testing of the specific avoidance maneuvers on the test aircraft, without relying on any type of simulation;
- Innovative use of Google Earth to enable pre-flight visualizations of Auto GCAS setups, enhance test-card generation, and greatly improve post-flight data analysis of test maneuvers; and
- Auto GCAS algorithms coded into Java™ (Oracle Corporation, Redwood Shores, California) (the native language for the smartphone).

Although the SUAV Auto GCAS implementation was not intended for production, the test results are positive enough to provide a solid basis for scaling onto many production UAV platforms. The same basic SUAV Auto GCAS concepts should also be adaptable to similar aircraft types having relatively low airspeeds and low maneuverability (typical of general aviation).

Some of the concepts have demonstrated the potential to provide significant capabilities for Auto GCAS implementations on higher-air-speed, higher-maneuverability aircraft such as fighters, transports, business jets, and airliners. These concepts include the use of highly-compressed DEMs, multi-trajectory avoidance options, and terrain scanning techniques.

Auto GCAS follow-on projects could include:

- Implementation of GCAS algorithms on a smartphone or tablet for general aviation platforms. General aviation applications could be automated when a digital autopilot is available, but significant improvements to protection against controlled flight into terrain (CFIT) might also be obtained when a digital autopilot is not available by using the same GCAS algorithms as part of a ground proximity warning system utilizing manual pilot reactions. Any automatic implementation of GCAS using a smartphone or tablet would need to pay particular attention to the redundancy and reliability aspects in order to avoid violation of the “do no harm” principle. General aviation smartphone or tablet implementations should be evaluated first in a simulation environment and then in flight-testing.
- Flight-test demonstration of GCAS algorithms for helicopter platforms. It is expected that helicopters could also benefit from automatic GCAS or a GCAS-based ground proximity warning system. Because of the additional avoidance maneuver options available to a helicopter (stop and hover, reverse direction, et cetera) additional development would be required to properly evaluate GCAS tradeoffs.
- Analytical studies and flight-test demonstrations to evaluate the effectiveness of new Auto GCAS concepts for fighter platforms. Even though a very capable production version of Auto GCAS is nearing deployment to the F-16 fleet, variations of Auto GCAS are still in development for the F-22 and F-35. Some of the techniques implemented on the SUAV Auto GCAS could provide significant improvements to the tradeoff between CFIT protection and nuisance potential for any of those airplanes.
- Analytical studies and flight-test demonstrations to evaluate the effectiveness of new Auto GCAS concepts for transport platforms. Transport platforms have limited maneuverability compared to fighter platforms, but also have much higher airspeed envelopes compared to an SUAV. This characteristic might lead to the need for adaptations of existing GCAS technologies that would be appropriate to explore in a flight-test development environment.

Summary of Recommendations

Recommendation 1 (R1): Future Auto GCAS projects should consider applying resources to develop improved integration of in-flight wind estimates with the Auto GCAS algorithm (page 19).

Recommendation 2 (R2): Future Auto GCAS projects should consider implementing a monitor to isolate and protect against corrupted digital terrain data (page 26).

Recommendation 3 (R3): Future Auto GCAS projects should consider implementing a monitor to protect against false requests for avoidance maneuvers (page 26).

Recommendation 4 (R4): Future Auto GCAS projects should pay special attention to the input signal conditioning necessary for that particular implementation (page 28).

Recommendation 5 (R5): Future Auto GCAS projects should consider incorporating multiple trajectory predictions to provide more than one option and to reduce nuisance potential (page 30).

Recommendation 6 (R6): Future Auto GCAS projects for performance-limited aircraft may need to consider including density altitude as an input to the trajectory prediction (page 36).

Recommendation 7 (R7): The developers on any Auto GCAS project should consider the addition of vertical obstructions as part of the algorithm (page 39).

Recommendation 8 (R8): Future Auto GCAS projects should carefully assess the tradeoffs between short-term PVI cost savings against the potential for longer-term impacts on the user (page 49).

Recommendation 9 (R9): Future Auto GCAS projects involving flight-testing of UAVs should pay particular attention to telemetry and control links when operating in close proximity to terrain (page 49).

Recommendation 10 (R10): Future Auto GCAS projects on UAVs should consider a mode state implementation that allows the avoidance to continue even after reaching a FAIL state (page 50).

Recommendation 11 (R11): Future Auto GCAS projects on UAVs should consider a self-recovering mode state implementation to resume CFIT protection as soon as the cause for the FAIL state no longer exists (page 50).

Recommendation 12 (R12): Future Auto GCAS test projects using a smartphone as the data recording device should consider implementing a recording method that provides data throughout the flight, not just when avoidance maneuvers occur (page 67).

Recommendation 13 (R13): Future Auto GCAS projects on UAVs should develop a nuisance criterion specific to that project (page 71).

Recommendation 14 (R14): Future Auto GCAS projects on UAVs should develop a termination timeliness criterion specific to that project (page 74).

Recommendation 15 (R15): Future Auto GCAS projects on propeller aircraft should determine if the use of the P-factor effect in the termination logic is warranted on that platform (page 78).

Recommendation 16 (R16): Future Auto GCAS projects should consider implementing termination logic that returns control to the pilot when well clear of terrain in the immediate vicinity, but should not be overly conservative for distant peaks (page 79).

Recommendation 17 (R17): Future Auto GCAS projects should determine if changing winds during the recovery should be addressed in the design for that platform (page 80).

Appendix A: Parameter List

The main purpose in presenting the parameter list is to provide future project teams with a starting point for consideration when they are developing their own project-specific parameter list. Table A1 provides a complete list of the parameters that were available to the Auto GCAS test team. All of the parameters that were used for post-flight analysis were recorded on the smartphone and downloaded after the flight. The post-flight analysis parameters were recorded on the smartphone regardless of its location (whether in the ground control van or in the DROID). Additional telemetered data were available in the ground control van, but those data were not used for post-flight analyses and so are not included in table A1. Additional data were also recorded on the user interface (UI) laptop computer, but those data were not used for post-flight analyses and so have not been included in table A1.

Three sources provided the data that were recorded on the smartphone: the Piccolo II autopilot; the UI laptop computer (by way of uplinked telemetry or the Gumstix[®] personal computer); and the Auto GCAS algorithm on the smartphone. Selected parameters from the Piccolo II autopilot included basic aircraft state information such as latitude, longitude, airspeed, altitude, angular rates, linear accelerations, et cetera, along with autopilot status and target values and the winds as estimated by the standard Piccolo II autopilot software. The UI parameters consisted primarily of state values that could be set by the UI Operator: Terrain Clearance Buffer; horizontal and vertical uncertainties (for the built-in buffers); the flag to switch between multiple avoidance trajectories and the straight-only trajectory, and others. The data from the Auto GCAS algorithm on the smartphone included Auto GCAS modes, time to flyup for each of the three trajectories, and minimum approach to terrain for each of the three trajectories, along with some Auto GCAS status words.

A number of parameters were not directly available from any of the sources but could be calculated using the available parameters. Those calculated parameters are listed near the end of table A1.

The parameter names in table A1 are listed in two columns: “Smartphone parameters,” and “AUTO GCAS maneuver-specific parameters.” Most of the parameter names are the same in both of these columns. In a few cases the parameter existed in only one form, indicated in table A1 by the entry “N/A.” in the relevant parameter column. The difference only applied to the type of post-flight analysis being conducted.

Table A1. Small Unmanned Aerial Vehicle Automatic Ground Control Avoidance System parameters.

Smartphone parameters	AUTO GCAS maneuver-specific parameters	Units	Description	Source	Data rate
time_system	time_system	ms	Time since start-up	Piccolo II	5 Hz
flt	N/A	N/A	Flight number	Algorithm	5 Hz
exe	exe	N/A	Execute flag (1 or 0)	Smartphone	5 Hz
latitude	latitude	deg	GPS latitude	Piccolo II	5 Hz
longitude	longitude	deg	GPS longitude	Piccolo II	5 Hz
altitude_gps_wgs84	altitude_gps_wgs84	ft	GPS altitude above the WGS84 ellipsoid	Piccolo II	5 Hz
ktas	ktas	kn	Calculated true airspeed	Piccolo II	5 Hz
kias	kias	kn	Indicated airspeed (raw)	Piccolo II	5 Hz
bankAngle	bankAngle	deg	Bank (roll) angle	Piccolo II	5 Hz
climbRate	climbRate	ft/s	Climb rate	Piccolo II	5 Hz
rollRate	rollRate	deg/s	Roll rate, as read from Piccolo II 3-axis gyroscope	Piccolo II	5 Hz
heading	heading	deg	True heading	Piccolo II	5 Hz

windVelNorth	windVelNorth	ft/s	Wind velocity, north (parameter is positive with wind from south)	Piccolo II	5 Hz
windVelEast	windVelEast	ft/s	Wind velocity, east (parameter is positive with wind from west)	Piccolo II	5 Hz
horizUncertainty	horizUncertainty	ft	Horizontal uncertainty	UI laptop	5 Hz
vertUncertainty	vertUncertainty	ft	Vertical uncertainty	UI laptop	5 Hz
TCB	TCB	ft	Terrain Clearance Buffer	UI laptop	5 Hz
rollRateLag	rollRateLag	N/A	Roll rate lag	UI laptop	5 Hz
airspeedLag	airspeedLag	N/A	Airspeed lag	UI laptop	5 Hz
climbRateLag	climbRateLag	N/A	Climb rate lag	UI laptop	5 Hz
multiAvoidReq	multiAvoidReq	N/A	Multi-avoid request	UI laptop	5 Hz
acDataValid	acDataValid	N/A	AC data valid flag	UI laptop	5 Hz
AGCASmodeReq	AGCASmodeReq	N/A	AGCAS mode request	UI laptop	5 Hz
AGCASmode	AGCASmode	N/A	AGCAS mode as reported by smartphone	Algorithm	5 Hz
apCmd	apCmd	N/A	AGCAS autopilot command request	Smartphone	5 Hz
strTime2Flyup	strTime2Flyup	s	AGCAS time to flyup, straight	Smartphone	5 Hz
leftTime2Flyup	leftTime2Flyup	s	AGCAS time to flyup, left	Smartphone	5 Hz
rightTime2Flyup	rightTime2Flyup	s	AGCAS time to flyup, right	Smartphone	5 Hz
minApprTerrainStr	minApprTerrainStr	ft	AGCAS Minimum Approach to Terrain, straight	Smartphone	5 Hz
minApprTerrainLeft	minApprTerrainLeft	ft	AGCAS Minimum Approach to Terrain, left	Smartphone	5 Hz
minApprTerrainRight	minApprTerrainRight	ft	AGCAS Minimum Approach to Terrain, right	Smartphone	5 Hz
errorCode	errorCode	N/A	AGCAS error code	Algorithm	5 Hz
warningCode	warningCode	N/A	AGCAS warning code	Smartphone	5 Hz
infoCode	infoCode	N/A	AGCAS info code	Smartphone	5 Hz
lmRefLatitude	lmRefLatitude	deg	Local map reference, latitude	Smartphone	5 Hz
lmRefLongitude	lmRefLongitude	deg	Local map reference, longitude	Smartphone	5 Hz
time_gps_hours	time_gps_hours	hr	GPS time	Piccolo II	5 Hz
time_gps_minutes	time_gps_minutes	min	GPS time	Piccolo II	5 Hz
time_gps_seconds	time_gps_seconds	s	GPS time	Piccolo II	5 Hz
altitude_baro	altitude_baro	ft	Barometric altitude, MSL	Piccolo II	5 Hz
laserAlt	laserAlt	ft	Laser altitude, AGL	Piccolo II	5 Hz
RPM	RPM	rpm	Revolutions per minute	Piccolo II	5 Hz
mag_hdg_deg	mag_hdg_deg	deg	Magnetic heading	Piccolo II	5 Hz
pitch_deg	pitch_deg	deg	Pitch	Piccolo II	5 Hz
yaw_deg	yaw_deg	deg	Yaw	Piccolo II	5 Hz
pitch_rate_dps	pitch_rate_dps	deg/s	Pitch rate	Piccolo II	5 Hz
yaw_rate_dps	yaw_rate_dps	deg/s	Yaw rate	Piccolo II	5 Hz
xaccel_g	xaccel_g	g	Acceleration, x-direction	Piccolo II	5 Hz
yaccel_g	yaccel_g	g	Acceleration, y-direction	Piccolo II	5 Hz
zaccel_g	zaccel_g	g	Acceleration, z-direction	Piccolo II	5 Hz
LoopTarget0_kts	LoopTarget0_kts	kn	Indicated airspeed loop target	Piccolo II	5 Hz
LoopTarget1_ft	LoopTarget1_ft	ft	Altitude loop target	Piccolo II	5 Hz
LoopTarget2_deg	LoopTarget2_deg	deg	Bank loop target	Piccolo II	5 Hz
LoopTarget3_deg	LoopTarget3_deg	deg	Flaps loop target	Piccolo II	5 Hz
LoopTarget4_deg	LoopTarget4_deg	deg	Heading loop target	Piccolo II	5 Hz
LoopTarget5_deg	LoopTarget5_fpm	ft/min	VRate loop target	Piccolo II	5 Hz

AP_Global	AP_Global	N/A	Autopilot global on/off flag. Indicates whether or not SUAV was flown by ground control operator(s), or whether or not a safety pilot had control.	Piccolo II	5 Hz
TrackerStatus	TrackerStatus	N/A	Tracker status	Piccolo II	5 Hz
LoopStatus0	LoopStatus0	N/A	Indicated airspeed loop status	Piccolo II	5 Hz
LoopStatus1	LoopStatus1	N/A	Altitude loop status	Piccolo II	5 Hz
LoopStatus2	LoopStatus2	N/A	Bank loop status	Piccolo II	5 Hz
LoopStatus3	LoopStatus3	N/A	Flaps loop status	Piccolo II	5 Hz
LoopStatus4	LoopStatus4	N/A	Heading loop status	Piccolo II	5 Hz
LoopStatus5	LoopStatus5	N/A	VRate loop status	Piccolo II	5 Hz
windVelNAve	windVelNAve	ft/s	10 s average of parameter windVelNorth	UI laptop	5 Hz
windVEIEAve	windVelEAve	ft/s	10 s average of parameter windVelEast	UI laptop	5 Hz
triggerPressed	triggerPressed	N/A	Trigger pressed	UI laptop	5 Hz
PARSstate	PARSstate	N/A	Pilot Activated Recovery Switch state	UI laptop	5 Hz
PARSengaged	PARSengaged	N/A	Pilot Activated Recovery Switch engaged flag	UI laptop	5 Hz
flyupActive	flyupActive	N/A	Flyup active flag. Boolean 1 (flyup active) or 0 (flyup not active). Same as parameter flyup_active.	UI laptop	5 Hz
noTMcount	noTMcount	N/A	Telemetry failure flag	Smartphone	5 Hz
flyupHold	flyupHold	N/A	Flyup hold flag	Smartphone	5 Hz
flyupHoldCount	flyupHoldCount	N/A	Flyup hold count	Smartphone	5 Hz
Vnorth	N/A	ft/s	North velocity (north-east-down frame)	Piccolo II	5 Hz
Veast	N/A	ft/s	East velocity (north-east-down frame)	Piccolo II	5 Hz
Vdown	N/A	ft/s	Down velocity (north-east-down frame)	Piccolo II	5 Hz
Clock_ms	Clock_ms	ms	Same as parameter time_system	Calculated	N/A
Hours	Hours	hr	Same as parameter time_gps_hours	Calculated	N/A
Minutes	Minutes	min	Same as parameter time_gps_minutes	Calculated	N/A
Seconds	Seconds	s	Same as parameter time_gps_seconds	Calculated	N/A
time_sec	time_sec	s	Time, starting at zero	Calculated	N/A
time_sec_since_midnight	time_sec_since_midnight	s	Time since midnight	Calculated	N/A
time_sec2	time_sec2	s	Time since midnight, starting at zero	Calculated	N/A
time_irig	time_irig	s	Time, IRIG style	Calculated	N/A
time_delta	time_delta	s	Change in time between data points	Calculated	N/A
time_delta2	time_delta2	s	Change in time between data points, starting at zero	Calculated	N/A
LeftRPM	LeftRPM	rpm	Same as parameter RPM	Calculated	N/A
latitude_deg	latitude_deg	deg	Same as parameter latitude	Calculated	N/A
longitude_deg	longitude_deg	deg	Same as parameter longitude	Calculated	N/A

height_ft	height_ft	ft	Same as parameter altitude_gps_wgs84	Calculated	N/A
direction_deg	direction_deg	deg	Same as parameter heading	Calculated	N/A
alt_ft	alt_ft	ft	Same as parameter altitude_baro	Calculated	N/A
tas_kts	tas_kts	kn	Same as parameter ktas	Calculated	N/A
roll_rate_dps	roll_rate_dps	deg/s	Same as parameter rollRate	Calculated	N/A
roll_deg	roll_deg	deg	Same as parameter bankAngle	Calculated	N/A
agl_ft	agl_ft	ft	Same as parameter laserAlt	Calculated	N/A
windsouth_fps	windsouth_fps	ft/s	Same as parameter windVelNorth	Calculated	N/A
windwest_fps	windwest_fps	ft/s	Same as parameter windVelEast	Calculated	N/A
pars_engage	pars_engage	N/A	Same as parameter PARSengaged	Calculated	N/A
ias_kts	ias_kts	kn	Same as parameter kias	Calculated	N/A
climb_rate_fps	climb_rate_fps	ft/s	Same as parameter climbRate	Calculated	N/A
climb_rate_fpm	climb_rate_fpm	ft/min	Parameter climbRate, multiplied by 60.0	Calculated	N/A
flyup_active	flyup_active	N/A	Flyup active flag as set by UI. Boolean 1 (flyup active) or 0 (flyup not active). Same as parameter flyupActive	Calculated	N/A
altitude_gps_msl	altitude_gps_msl	ft	GPS altitude, above mean sea level	Calculated	N/A
N/A	type	N/A	Text string used in determining where flyups occur in data	Calculated	N/A
N/A	event	N/A	Event number	Calculated	N/A
N/A	alt_AGL_ft	ft	Altitude, AGL (used for comparison with laser altimeter data)	Calculated	N/A
N/A	alt_NED_wgs84_ft	ft	Altitude, NED, using WGS84 ellipsoid	Calculated	N/A
N/A	alt_NED_wgs84_m	m	Altitude, NED, using WGS84 ellipsoid	Calculated	N/A
N/A	GPSVelNorth	ft/s	GPS velocity, north	Calculated	N/A
N/A	GPSVelEast	ft/s	GPS velocity, east	Calculated	N/A
N/A	GPSVelDown	ft/s	GPS velocity, down	Calculated	N/A
N/A	deltaDistN	ft	Change in distance, north	Calculated	N/A
N/A	deltaDistE	ft	Change in distance, east	Calculated	N/A
N/A	deltaDistD	ft	Change in distance, down	Calculated	N/A
N/A	dt_sec	s	Differential time	Calculated	N/A
N/A	GPSVel_fps	ft/s	GPS velocity, total	Calculated	N/A
N/A	GPSVel_kts	kn	GPS velocity, total	Calculated	N/A
N/A	GPSAccelNorth_fpsps	ft/s ²	GPS acceleration, north	Calculated	N/A
N/A	GPSAccelEast_fpsps	ft/s ²	GPS acceleration, east	Calculated	N/A
N/A	GPSAccelDown_fpsps	ft/s ²	GPS acceleration, down	Calculated	N/A
N/A	ART	s	Available reaction time	Calculated	N/A
N/A	FPA	deg	Flightpath angle	Calculated	N/A
N/A	deltaTermAz	deg	Number of degrees turned until termination occurs	Calculated	N/A
N/A	deltaTermTime	N/A	Time until termination occurs	Calculated	N/A

Appendix B: Analysis Techniques

Most of the analyses presented in this report consist of straightforward presentations of the data and require no further elaboration. A few of the analysis methods were unique, however, warranting some additional background description to help future project teams adapt those methods to their own needs. Those unique analysis techniques were:

- Google Earth interface;
- Available reaction time;
- Determination of trigger post locations;
- Determination of worst-case mismatch locations;
- Trajectory normalization analysis; and
- Termination logic cross-checks.

Google Earth Interface

One of the tools used by the Small Unmanned Aerial Vehicle (SUAV) team for planning test missions and analyzing flight-test data was Google Earth. Google Earth provided a markup language known as keyhole markup language, or KML. (Keyhole Corp., Mountain View, California, was a technology company purchased by Google in 2005). Keyhole markup language allows users to plot locations, lines, and polygons within Google Earth for visualization purposes. This same markup language enables Google Earth to render images of three-dimensional buildings, text, and other visualization aids. For the purpose of analyzing SUAV flight test data, KML proved to be an extremely useful data visualization tool for several reasons:

- Raw flight-test data (SUAV latitude, longitude, and elevation) were plotted using Google Earth, which readily allowed users to see where the Dryden Remotely Operated Integrated Drone (DROID) airplane flew relative to the nearby terrain features.
- Algorithm data were also plotted using Google Earth. Algorithm data include trajectory predictions, scan patterns, and representative digital elevation model (DEM) data. Plotting algorithm data using Google Earth allowed users to determine the effectiveness of the collision avoidance algorithms.
- The Auto GCAS team was able to use Google Earth to visualize Auto GCAS termination predictions. Such visualizations would have been much less useful and far more difficult without Google Earth.
- The team was able to use Google Earth to visualize an available reaction time (ART) by extending the actual avoidance trajectory to where the representative DEM data were located. The ART was useful for determining a degree of “nuisance potential,” as discussed above within the main body of this report.

During the analysis of the SUAV flight-test data, various software routines were implemented to write KML files for use with Google Earth. In order to process KML files for use with Google Earth, the analysis team used MATLAB[®] (The MathWorks, Natick, Massachusetts). The MATLAB utility was the tool of choice because it provided a simple interface with which users could easily write scripts which, when executed, would write KML files. Of particular interest was the MATLAB Mapping Toolbox[™], which contained several geodetic calculation functions. The geodetic calculation functions were used to calculate trajectories on the WGS84 ellipsoid mathematical model of the surface of the Earth.

Excel[®] Test Planning Tool

The SUAV Auto GCAS test point setups and test cards were produced using an Excel[®] (Microsoft Corporation, Redmond, Washington) worksheet, a sample of which is shown in figure B1, which

implemented both the collision avoidance algorithms (similar to those used on the smartphone) and the DEM data set as needed for the SUAV Auto GCAS flight-testing. The smartphone contained DEM data for the entire Earth, but the Excel® worksheet needed DEM data only for the local test areas.

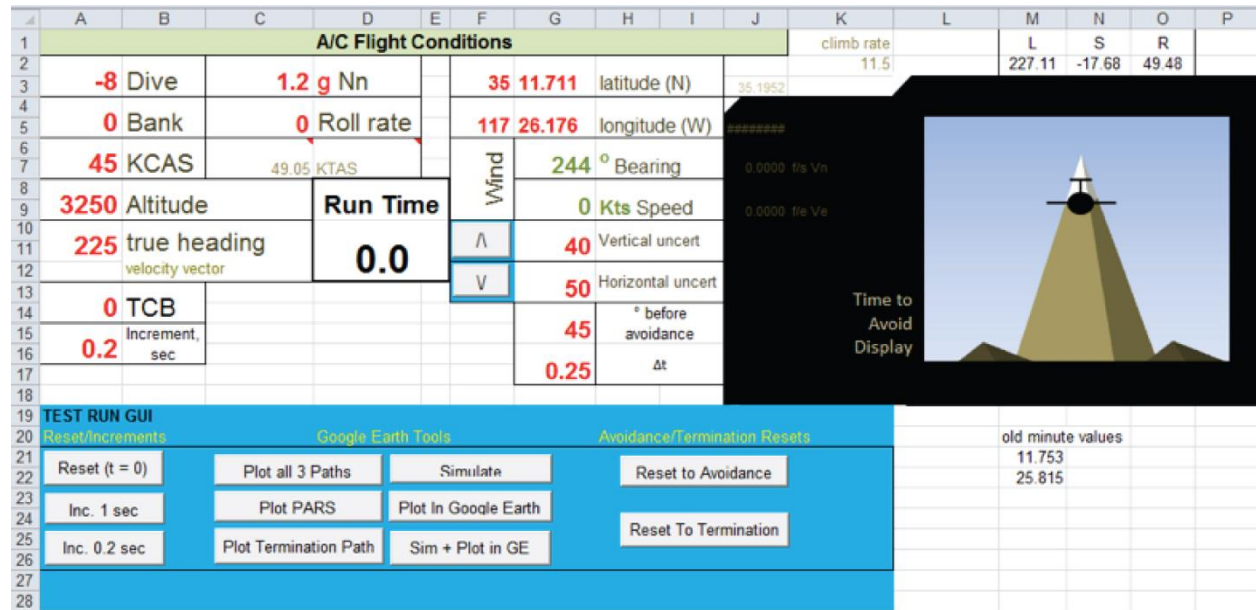


Figure B1. Excel® test planning application.

The Excel® worksheet provided a very useful, basic simulation of the end-to-end Auto GCAS algorithm. The Excel® methodology had been used successfully to support a number of F-16 airplane (Lockheed Martin, Bethesda, Maryland) Auto GCAS projects dating back to the early 1990s, and the Excel® implementation was also used successfully in support of this SUAV Auto GCAS project. Future related projects should take into consideration that any Auto GCAS simulation method will need to have functionality similar to the Excel® implementation described herein. The combination of the Excel® worksheet simulation and Google Earth was new for the SUAV Auto GCAS project. That combination of the two was particularly effective for evaluating and selecting potential flight-test setups. The ability to feed simulation results into Google Earth should be a consideration for future Auto GCAS projects.

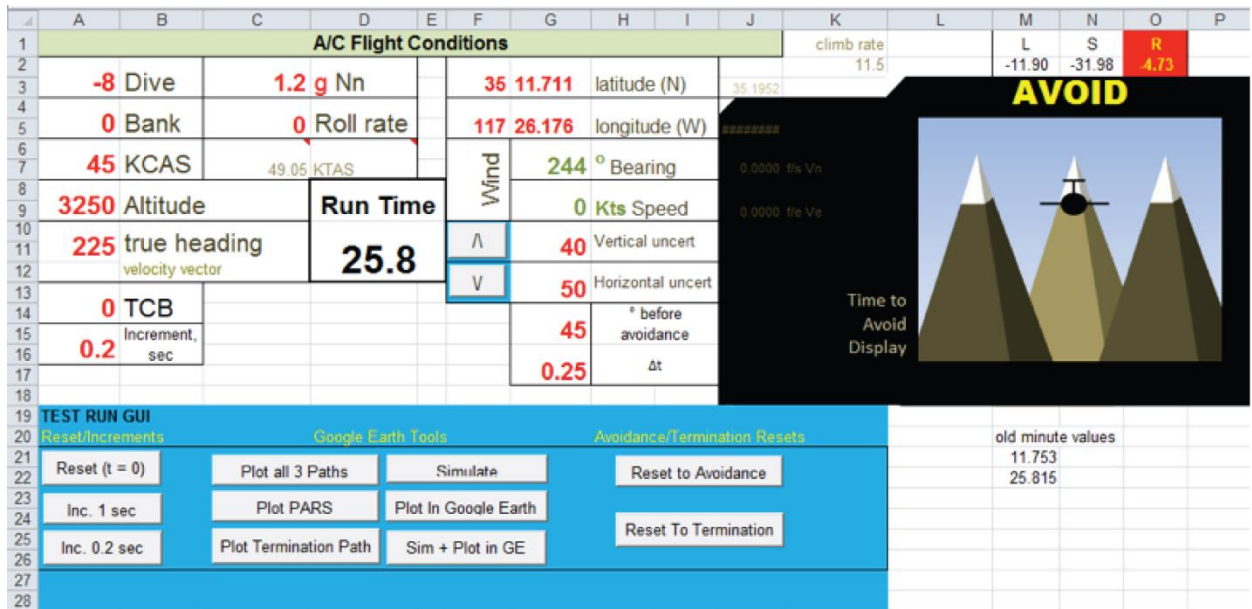
The user interface of the Excel® test planning application is shown in figure B1. The user could enter all of the flight-test point setup parameters (the red numbers shown in figure B1), and then use the graphical user interface buttons (the buttons within the blue fields, also shown in figure B1) to plot collision avoidance algorithm results using Google Earth.

The test planning process typically began by choosing a simple start location (latitude, longitude, and altitude) using Google Earth. A target terrain feature and a desired heading for approaching that terrain feature were visually selected. The Excel® - Google Earth combination was then used to refine the setup to obtain the desired conditions at collision avoidance initiation. The remaining parameters KCAS (knots calibrated airspeed); dive; bank; TCB (terrain clearance buffer); et cetera were entered and the “Run Time” along the desired heading was varied until the proximity to terrain indicated the need for an avoidance maneuver. Given all of the setup parameters, the results from the Excel® worksheet indicated the expected location for initiation of the avoidance maneuver and the direction of the avoidance (left, straight, or right). A considerable amount of additional useful information was also provided, such as the expected flightpath

and the closest approach to terrain. All of this information could be visualized using Google Earth as described below.

Excel® - Google Earth Interface

To implement the user interface buttons that are shown within the blue fields in figure B1, the analysis team used Microsoft Visual Basic for Applications (VBA). The VBA is a scripting interface that has been included with all builds of Microsoft Office® since version 4.0 (Microsoft Office® 2010 contains VBA version 14). The VBA allows Excel® users to facilitate automatic worksheet features and calculations that if performed manually would require vast amounts of time. In the case of the information shown in figure B1, the user would have to constantly enter a time in the “Run Time” field in order to determine exactly where and when a ground collision avoidance maneuver would occur. The analysis team inserted VBA code to automate the process of determining the location of a ground collision avoidance maneuver. The user simply clicked on the button labeled “Simulate,” and Excel® ran the calculations until an avoidance, as shown in figure B2, was found.



150105

Figure B2. Avoidance maneuver initiation example.

Figure B3 displays the results of clicking on the “Plot in Google Earth” command button that is shown in figures B1 and B2. In figure B3, the black line is the “pre-avoidance” flightpath (that is, the predicted flightpath until the point of an Auto GCAS maneuver). When the flightpath was expected to be straight (a 0 initial bank value in merged cells A4/A5 as shown in figure B2), the program drew a straight line in Google Earth to represent a straight pre-avoidance flightpath. That straight line began at the initial longitude-latitude-altitude coordinates and along the heading specified by the user, and ended at the location of the avoidance maneuver initiation.

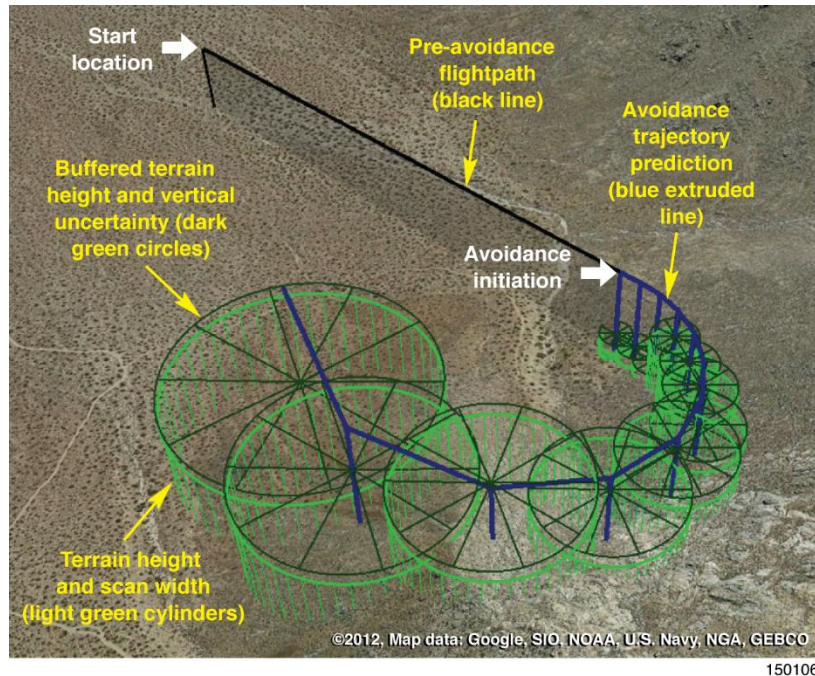


Figure B3. Result from clicking “Plot in Google Earth” button illustrated in figure B2.

If a turning flightpath was expected, the pre-avoidance flightpath displayed as curved in Google Earth. The length of that curved black line was based on the user-specified heading change, using a radius established by the user-specified bank angle and load factor. For example, in figure B2, merged cells G14/G15 were set to 45, which meant that the curved black line in Google Earth represented a turn through a heading change of 45 deg prior to the Auto GCAS avoidance maneuver initiation. In the example worksheet shown in figure B2, the load factor of 1.2 g entered in merged cells C2/C3 determined the radius of the curved flightpath. Visualizing the results in Google Earth enabled the user to determine where a turning flight-test maneuver needed to begin in order to result in an avoidance initiation at the desired location and flight conditions.

The light green cylinders in figure B3 represent the Auto GCAS algorithm scan width, which is a function of the range along the avoidance prediction path (the scan width circles increase in diameter with range from the initiation point). The scan width near the initiation point was based on the horizontal uncertainty (from merged cells G12/G13 in figure B2). The dark green discs (“wagon wheels”) in figure B3, which lie slightly above the light green cylinders, represent the sum of the terrain clearance buffer (TCB from merged cells A13/A14 in figure B2) and vertical uncertainty (from merged cells G10/G11 in figure B2). Whenever any portion of the buffered terrain (represented by the dark green discs in figure B3) intersected with the trajectory prediction (represented by the blue line in figure B3) the result would indicate the coordinates for the expected avoidance initiation (indicated by the white arrow in figure B3).

The “Plot In Google Earth” buttons shown in figures B1 and B2 allowed the user to display KML paths within Google Earth that represented the ground collision avoidance trajectory predictions. An Excel® - MATLAB® interface was required in order to use MATLAB® functions that were needed to produce KML files. The MATLAB® application and Excel® communicated through the Microsoft Component Object Model (COM). By implementing a COM interface, MATLAB® and Excel® were able to easily exchange data in real time.

In addition to pre-flight predictions using the Excel[®] tool, the analysis team also plotted flight-test data from the DROID within Google Earth. Figure B4 represents an actual avoidance maneuver taken from flight-test data. The black extrusion represents the actual DROID flightpath before and after the avoidance maneuver. The red extrusion represents the actual DROID flightpath during the avoidance maneuver. Terminology for the green cylinders and dark green “wagon wheels” remains the same as that used for figure B3. The orange cubes represent the DEM data for the GCAS valley test area. It can be seen in figure B4 that neither the left nor straight trajectory was selected (because the trajectory predictions were well below the buffered terrain within the scan patterns). Figure B4 also shows that the actual flightpath for the selected right avoidance maneuver closely followed the trajectory prediction for this example. Additional examples for Google Earth plots of flight-test data are shown in the main body of this report as well as below within this appendix.

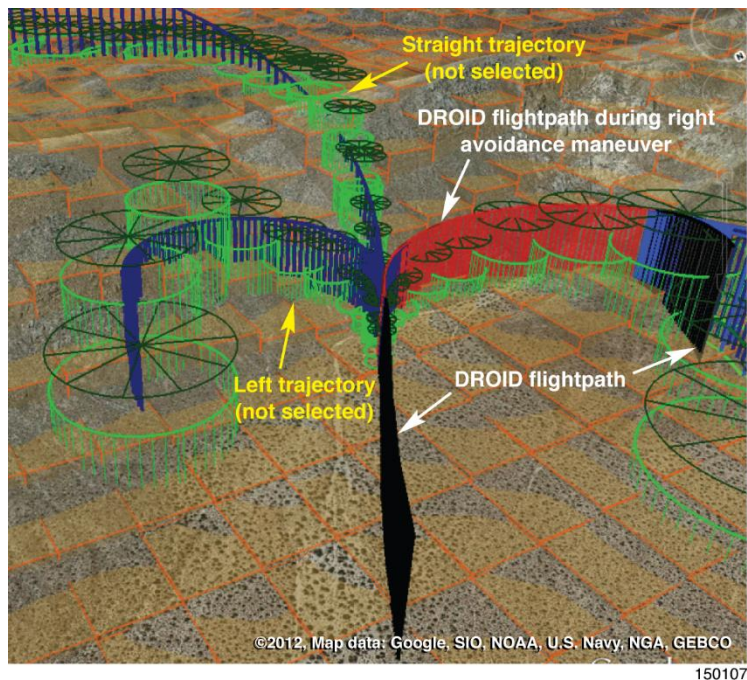


Figure B4. Example flight-test data.

Available Reaction Time

The premise of the “Available Reaction Time (ART)” calculation was to determine the amount of time within which a pilot would need to react were an Auto GCAS maneuver not initiated. Available reaction time was defined as the amount of time after initiation of the Auto GCAS maneuver within which the same maneuver could have been delayed while still avoiding terrain. The reason for calculating ART was to determine the “nuisance potential” of the system. A formal study has not yet been performed to quantify the ART nuisance boundary for UAVs similar to the DROID. For the purpose of this demonstration project, ARTs in excess of six seconds were considered possible nuisances, while those less than four seconds were considered probable non-nuisances. A negative ART indicates that the avoidance maneuver would not prevent the DROID from flying into the terrain, or in this case, the terrain plus TCB.

To calculate ART, the trajectory of the DROID after avoidance maneuver initiation was extrapolated as though the avoidance maneuver had been delayed. Figure B5 shows a setup maneuver, avoidance activation, and avoidance maneuver but without an extrapolated trajectory. The extrapolated trajectory consisted of three segments: the delay segment, the delayed avoidance maneuver segment, and the

post-termination maneuver segment. The delay segment, shown in green in figures B6, B7, and B8, estimated the DROID's path had no avoidance maneuver been initiated. The delayed avoidance maneuver segment, shown in dark red in the same three figures, used the same avoidance maneuver path that occurred in flight-testing, delayed by the selected time increment. The post-termination maneuver segment, shown in dark blue in these figures, attempted to estimate the trajectory as though the avoidance maneuver had continued after termination of the flight-test maneuver. The delay segment was progressively increased until any of the three segments intersected with the digital terrain. The ART was then defined at the value of the delay segment one frame before the extrapolation intersected with the digital terrain.

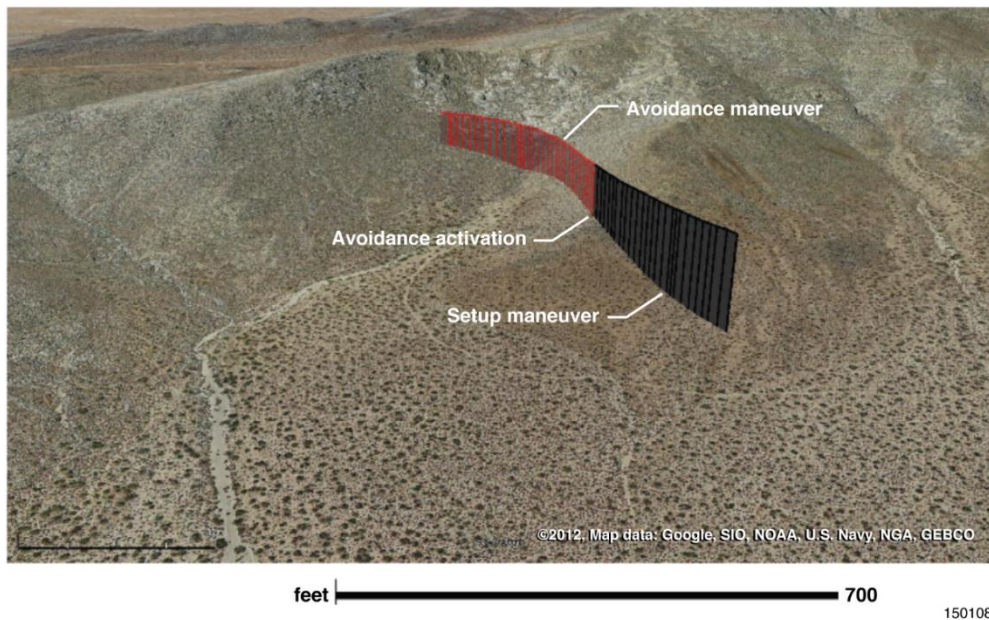


Figure B5. Actual trajectory.

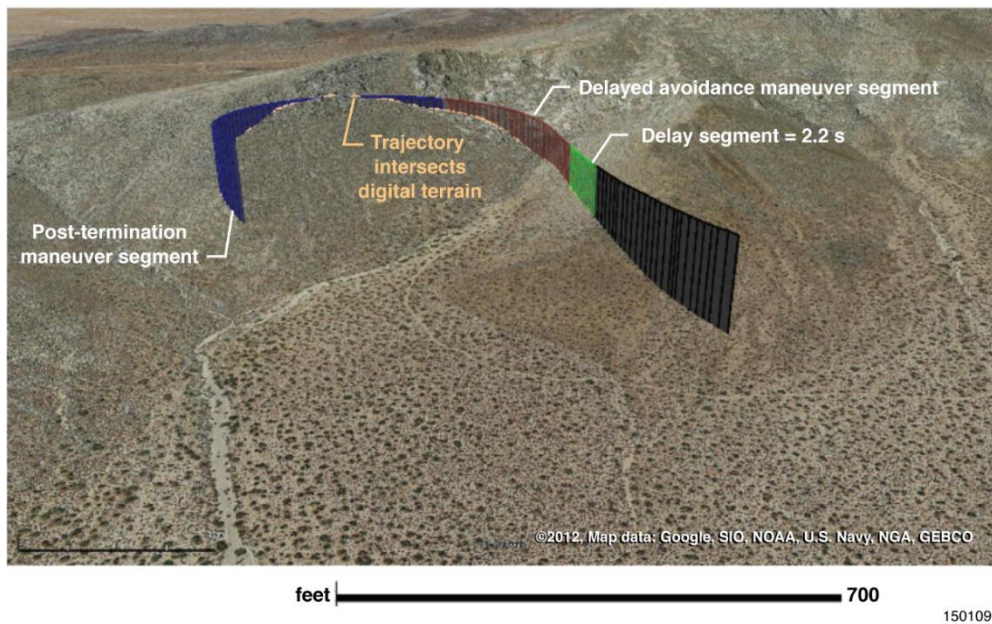


Figure B6. Three-segment extrapolation at intersection with terrain.

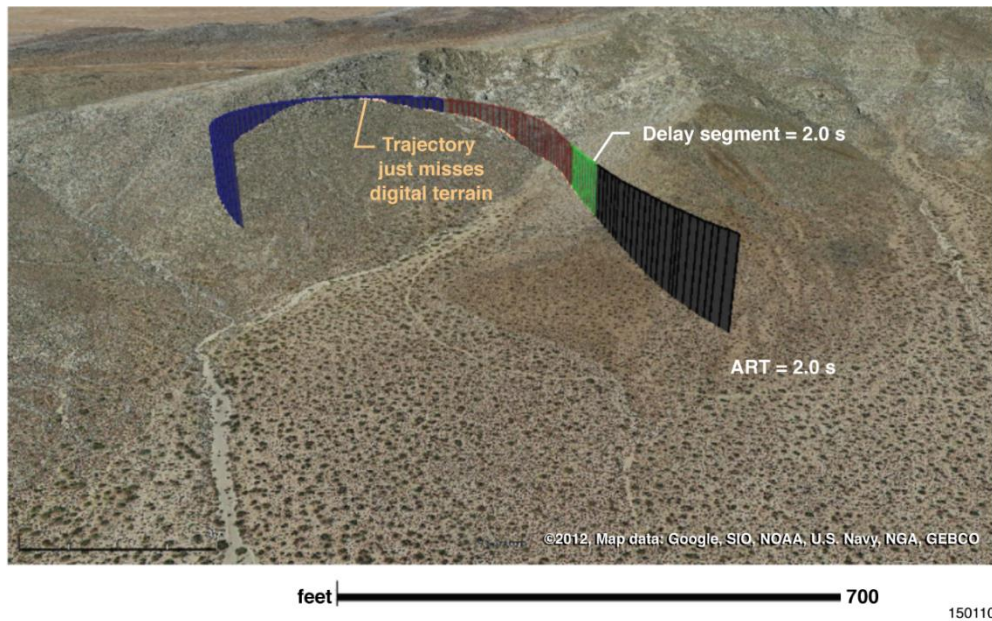


Figure B7. Three-segment extrapolation one frame before terrain intersection.

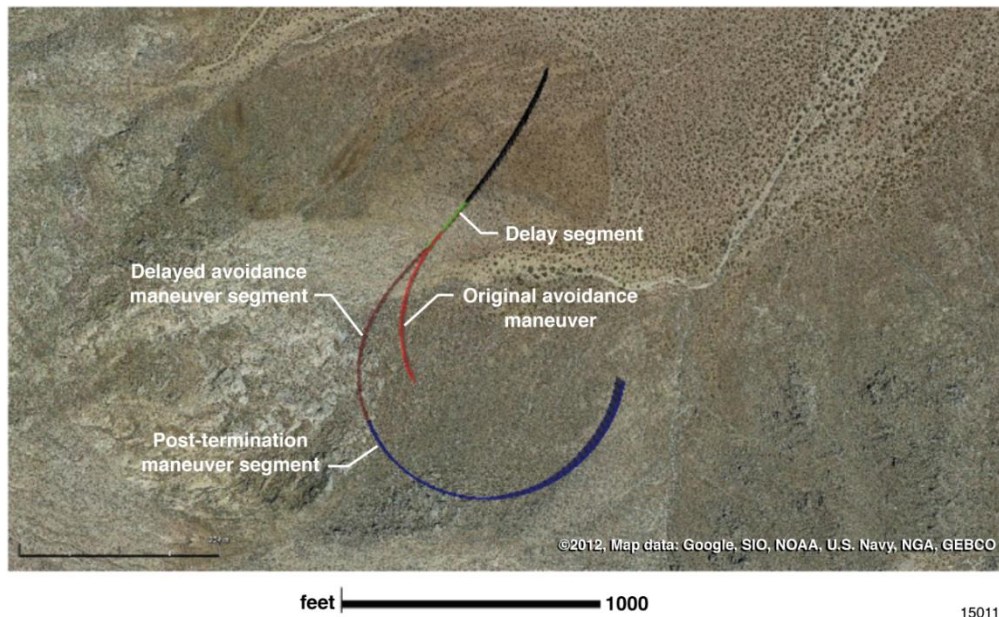


Figure B8. Three-segment extrapolation from above.

Figures B5, B6, and B7 show flight 19 event 16. Figure B5 shows the actual trajectory of the DROID, including the path during the setup maneuver prior to the avoidance activation and the actual avoidance maneuver. Figure B6 shows the three segments of extrapolation delayed 2.2 s, resulting in part of the trajectory intersecting with the digital terrain. Fig. B7 shows the three segments one frame earlier, when the extrapolation was delayed 2.0 s and the trajectory just missed the digital terrain. For this example, this technique resulted in an ART value of 2.0 s.

The delay segment treated the DROID as a point mass and used basic particle kinematics to extrapolate using the aircraft states at the original activation, ignoring acceleration in the z-axis. The delayed avoidance maneuver was a copy of the actual avoidance maneuver, rotated about the z-axis to align with the delay segment, so that the flightpath continued smoothly into the delayed avoidance maneuver. The flightpath after the delayed avoidance maneuver was calculated differently, depending on whether the avoidance maneuver was over smooth terrain or not.

Over smooth terrain, the important factor in determining whether the extrapolated flightpath would intersect with the digital terrain was the movement of the DROID in the vertical plane, because the digital terrain was of approximately the same height throughout the surrounding area. In mountainous areas, however, the movement of the DROID in the horizontal plane dominated the consideration of whether the extrapolated flightpath would intersect the digital terrain. For this reason, over smooth terrain, the actual flightpath of the DROID after avoidance maneuver termination was used. Over mountainous terrain the trajectory prediction (as calculated at activation) was used to estimate the flightpath after avoidance maneuver termination. The trajectory prediction was used for mountainous terrain instead of the actual flightpath because the safety pilot often took control shortly after normal termination (as planned for each test point). The trajectory prediction was considered the best estimate for the flightpath that would have occurred had the safety pilot not taken control. It was not necessary to use the trajectory prediction for smooth terrain because there were few of those events and the safety pilot did not take control until after the point at which the extrapolated trajectory intersected with digital terrain.

The best available digital terrain model was used to determine the intersection with the extrapolated flightpath. For these ART analyses, the one-third-arc-second National Elevation Dataset (NED) was used. The delay was incremented in time steps that were the same as the recorded data on the smartphone (roughly 0.2 s). The specific magnitude of each time step was dependent on the smartphone data time steps because the smartphone calculations were somewhat asynchronous and did not calculate at precise, regular intervals.

Three Segment Extrapolation Methods

As discussed above, the DROID trajectory was extrapolated in three segments: the delay segment, the delayed avoidance maneuver segment, and the post-termination maneuver segment. A top view of these three segments at the point where the ART value was determined for flight 19 event 16 (as viewed from above) is shown in figure B8. This figure shows that the three-segment extrapolation method creates a continuous trajectory.

Delay Segment

For the first segment of the extrapolated flightpath for ART calculations, the SUAV was treated, for simplicity, as a point mass. In general, ART calculations were based on particle kinematics, with the exception that vertical accelerations were ignored, and the DROID was assumed to experience a constant rate of climb or descent.

In the case of the ART analysis, vertical accelerations were not taken into account for two reasons. The first reason was the extremely low signal-to-noise ratio in the z-axis Piccolo II autopilot accelerometer data (see figure B9). The second reason was that the DROID often flew in near-equilibrium flight (lift equaling weight, and thrust equaling drag) as commanded by the autopilot. In addition, the small size of the DROID made it extremely susceptible to even slight changes in wind direction, wind speed, turbulence, and other atmospheric conditions. In calculating the ART for the F-16, the maneuvers were far “larger” in the sense of the scale of the F-16; therefore all accelerations (north/east/down) had to be taken into account.

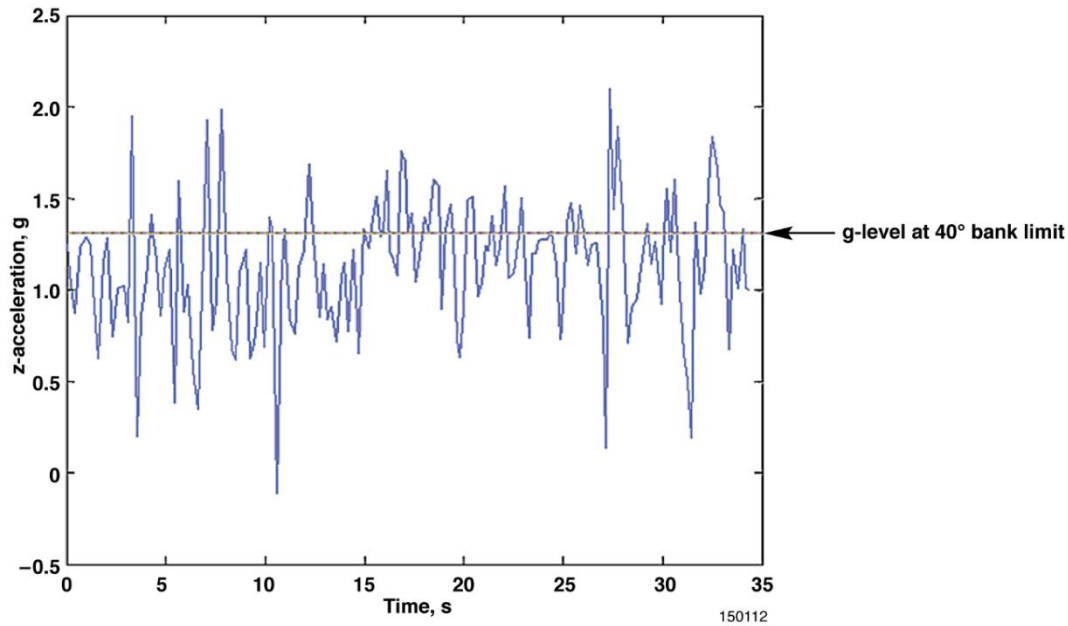


Figure B9. Impact of atmospheric turbulence on measured vertical acceleration.

In the case of the ART analysis as applied to the SUAV project, the noisy z-axis accelerometer data induced inappropriate and unlikely estimated flightpaths. The Piccolo II autopilot kinematics data (north and east velocities and accelerations) also failed to record; therefore, the analysis team generated the estimated flightpaths from known position and velocity data (latitudes, longitudes, and climb rate). Velocities were obtained from changes in latitude and longitude, and accelerations were further differentiated from changes in velocity. The technique of differentiation in order to obtain velocities and accelerations was not ideal because differentiation often amplified signal noise.

To counteract the effects of signal noise due to differentiation, the analysis team applied a generic smoothing routine to the north/east velocities and accelerations; vertical acceleration remained too noisy to be useful even after smoothing attempts. Figure B9 shows a typical sample of the measured z-acceleration. The upper limit of the normal maneuvering envelope of the DROID as configured (1.3 g given the preprogrammed bank limit of 40 deg) is also shown in figure B9. This clearly shows that the noise in the data captured was induced by atmospheric turbulence, not the normal maneuvering of the DROID.

Delayed Avoidance Maneuver

The delayed avoidance maneuver flightpath estimated what the DROID would have done had the same avoidance maneuver been executed after some time delay. Since the heading may have changed during the delay segment, the delayed avoidance maneuver was rotated to align with the heading at the end of that delay segment. A new initial heading and a new position (latitude, longitude, and altitude) were chosen to match the end of the delay segment flightpath, and from this position, the flightpath angles and changes in heading of the original avoidance maneuver were applied. Changing the initial heading but recreating the same change in heading effectively “rotated” the avoidance maneuver about the z-axis and maintained continuity in the horizontal flightpath between the delay segment and the delayed avoidance maneuver. Rotating the avoidance maneuver about the y-axis was not necessary because vertical acceleration was ignored and the flightpath angle did not change during the propagation.

Figures B6 and B8 show that the method of rotating the flightpath of the original avoidance maneuver and appending it to the end of the delay segment resulted in a continuous propagated flightpath. Ignoring the vertical acceleration did not induce a significant discontinuity in the vertical flightpath at the point where the delay segment met the delayed avoidance maneuver. This method worked because most flight-test maneuvers were near 1 g during the approach to the avoidance maneuver. If the maneuvering of the DROID at the initiation of the avoidance maneuver were more dynamic, a different propagation method would have been required.

Post-Termination: Mountainous Terrain

When the avoidance maneuver ended, if neither the safety pilot nor ground control gave instructions to the DROID, the DROID would stay in the climbing turn last commanded during the avoidance maneuver. In other words, the autopilot made no effort to return the DROID to level flight before terminating. Over mountainous terrain, it was the curvature of this climbing turn that most accurately determined whether the extrapolated flightpath would intersect with digital terrain. The actual post-termination flightpath could not be used because it was normal test procedure for the safety pilot to take control soon after the avoidance maneuver had terminated. Therefore it was determined that the best estimate for propagating the flightpath of the DROID after termination was based on the steady climbing turn calculated as part of the trajectory prediction from when the avoidance maneuver was initiated.

To make the estimation of flightpath as continuous as possible, the point along the trajectory prediction where the heading most closely matched the heading at the end of the delayed avoidance maneuver was used as the beginning of the post-termination segment. This flightpath was then translated in three dimensions to meet the end of the delayed avoidance maneuver. This method may result in some discontinuity in flightpath angle, but since the slope of the flightpath of the DROID was so much less than the slope of the digital terrain, small offsets in extrapolated altitude were less significant than the latitude and longitude of the DROID during this segment. Also, this method guaranteed that the flightpath angle over the entire extrapolation was a flightpath angle sustainable by the DROID.

Flight 19 event 10 having flown over mountainous terrain, it was used to demonstrate this method for the post-termination segment. Figures B6 and B8 show that this method created a smooth transition (that is, revealed no obvious discontinuities) from the delayed avoidance maneuver to the post-termination segment.

Post-Termination: Smooth Terrain

Over smooth terrain, the most important factor in determining ART was the vertical flightpath shortly after the avoidance maneuver. In order to obtain the most accurate trajectory propagation, the actual flightpath after termination of the avoidance maneuver was used (instead of the trajectory prediction method that was used for mountainous terrain). This method worked over smooth terrain because there were a few seconds until the safety pilot took control, and that amount of time was sufficient to provide enough data for the propagation. The actual flightpath after termination, up until the safety pilot took control, was appended to the end of the delayed avoidance maneuver. That flightpath was rotated and translated similar to the way in which the delayed avoidance maneuver was appended to the delay segment. This ensured continuity in both heading and flightpath angle.

The post-termination segment of the extrapolated flightpath over smooth terrain was significantly less important to the ART calculation than that segment over mountainous terrain. Most avoidance maneuvers over smooth terrain terminated during a slight climb, which was sufficient to clear smooth terrain.

Comparison to Digital Terrain Truth Data

Once the longitude, latitude, and altitude of the DROID were extrapolated for the entire three-segment trajectory, the extrapolated trajectory was compared to “best source” data for the digital terrain. For this ART analysis the best source for truth data was considered to be the one-third-arc-second NED. At every point of longitude-latitude in the extrapolated flightpath (shown by solid colored vertical lines in Google Earth snapshots in figures B6, B7, B10, and B11) the height of the NED was found. The TCB for each event was then added to the NED as a way to normalize across every test run, since the Auto GCAS system interpreted the “ground” to be at digital terrain height + TCB. The NED altitude plus TCB is represented by a tan line in these figures. Because flight 19 event 16 had a TCB of 0 ft and Google Earth also used the one-third-arc-second NED, these tan lines are barely visible above the ground as it is displayed in Google Earth depiction of figures B6 and B7. Therefore, flight 19 event 7, with a TCB of 100 ft, was used to demonstrate this concept in figures B10 and B11. Figure B10 shows the delayed avoidance maneuver at the frame when the propagated trajectory was just clear of the buffered terrain (NED + TCB). That frame in the propagation was used to define the ART value. Figure B11 shows the delayed avoidance maneuver one frame later, when the propagated trajectory intersected with the NED + TCB. The vertical tan line shows the first point along the trajectory where the NED + TCB line was above the trajectory. The intersection in this event happened during the delayed avoidance maneuver segment, but it sometimes happened in the post-termination segment (as in the case of flight 19 event 16).

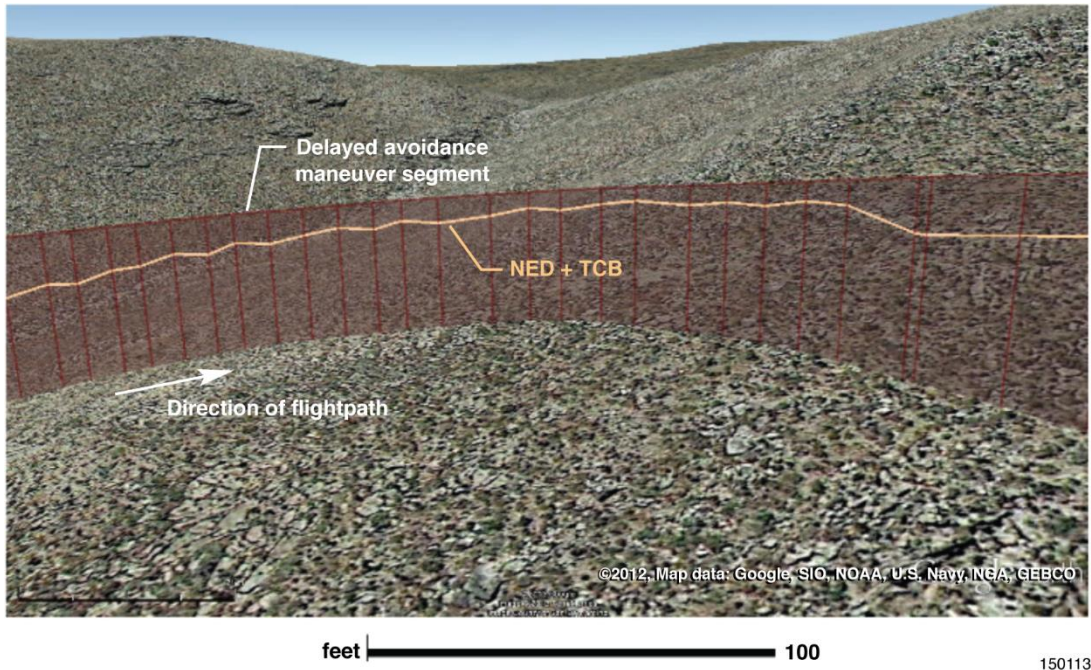


Figure B10. Extrapolated trajectory above NED + TCB (at the ART).

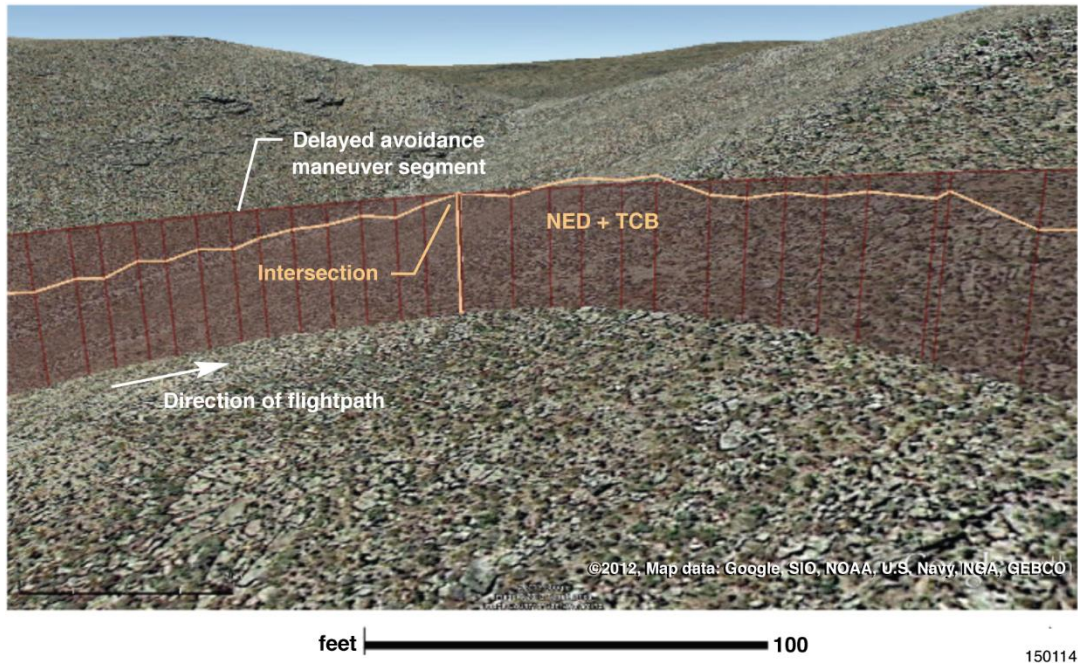


Figure B11. Extrapolated trajectory below NED + TCB (one frame after the ART).

Determination of Trigger Post Locations

As described in the “Generic ‘Sense Terrain’ Module” section in the main body of this report, the DEM was decompressed as a rectangular grid of posts with latitude, longitude, and altitude, and treated as a flat polygon surrounding that post at the same altitude. The trigger post analysis seeks to determine which specific post caused the avoidance maneuver.

The trajectory prediction and related scan circles in the direction of the avoidance were used as the basis for this analysis. In the case of flight 16 event 6, which is depicted in figures B12, B13, and B14, the avoidance was to the right, so the right trajectory prediction was analyzed. Within the Auto GCAS algorithm, the digital terrain was represented by scan circles at the altitude of the highest terrain polygon within the radius of each circle. That altitude was raised further to account for the flight-test buffer (the TCB) and the built-in buffer (vertical uncertainty) since the trajectory prediction was being compared to DEM + TCB + vertical uncertainty altitude.

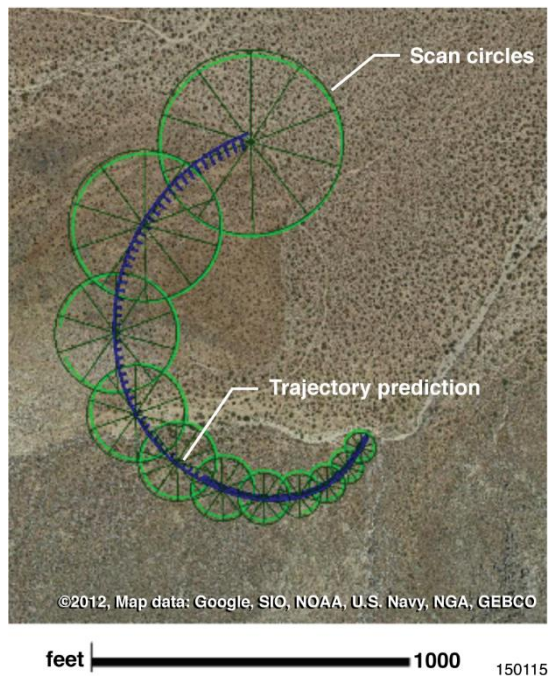


Figure B12. Trajectory prediction and scan circles viewed from above.

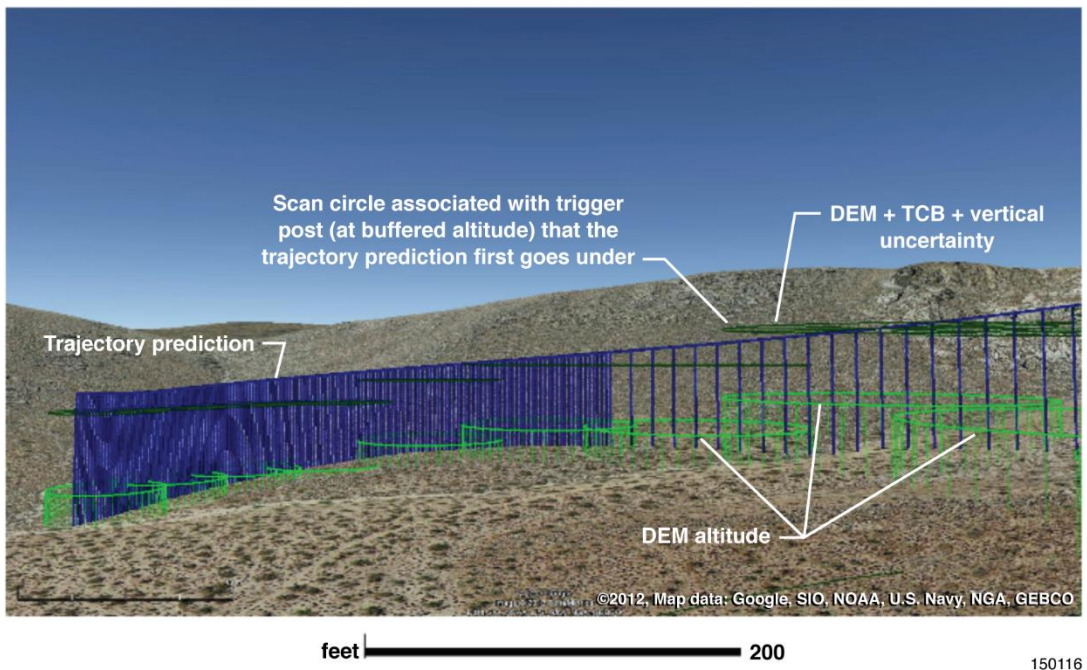


Figure B13. Trajectory prediction and scan circles viewed from side.

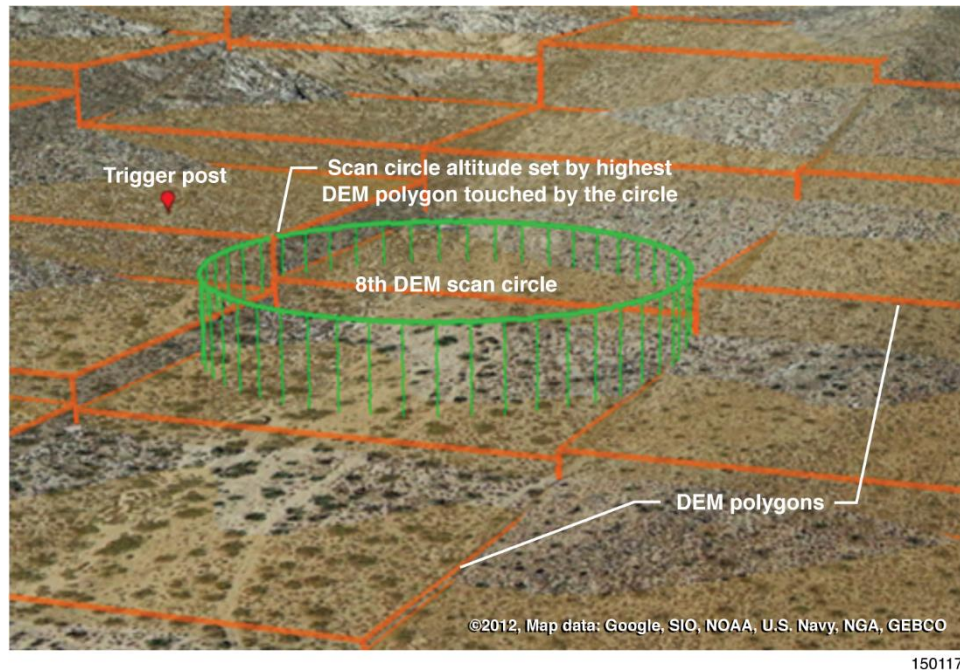


Figure B14. Trigger post location relative to scan circle.

In figure B13, the scan circles are shown in light green (at the DEM altitude) and dark green (at the DEM + TCB + vertical uncertainty altitude). The trajectory prediction is shown in blue. When the trajectory prediction was lower than one of the buffered scan circles the avoidance maneuver was triggered. The trigger post was associated with the polygon that gave this scan circle its height. In figure B13, it can be seen that the 8th light green scan circle (from the left) is paired with the first dark green buffered circle that the trajectory prediction passes underneath. Figure B14 shows this 8th scan circle and the surrounding DEM polygons. In this case the scan circle just touches the corner of the polygon associated with the trigger post. The practical significance of this implementation was that the DEM posts which triggered avoidance maneuvers tended to be outside of the scan circles and therefore created some additional horizontal buffer away from the actual terrain.

Determination of Worst-Case Mismatch Locations

In order to evaluate the accuracy of the trajectory prediction, the worst-case mismatch between the actual ground track compared to the predicted ground track was found for each event.

The actual ground track always included the automated avoidance maneuver. In some cases, the actual ground track also included the time after the automatic system terminated in order to determine when the DROID was at its minimum altitude above the ground (as long as the safety pilot did not take control). In figures B15 and B16, the actual ground track is shown by a red dotted line. The predicted ground track and half-scan width, as determined by the algorithm, are shown in blue and purple solid lines, respectively.

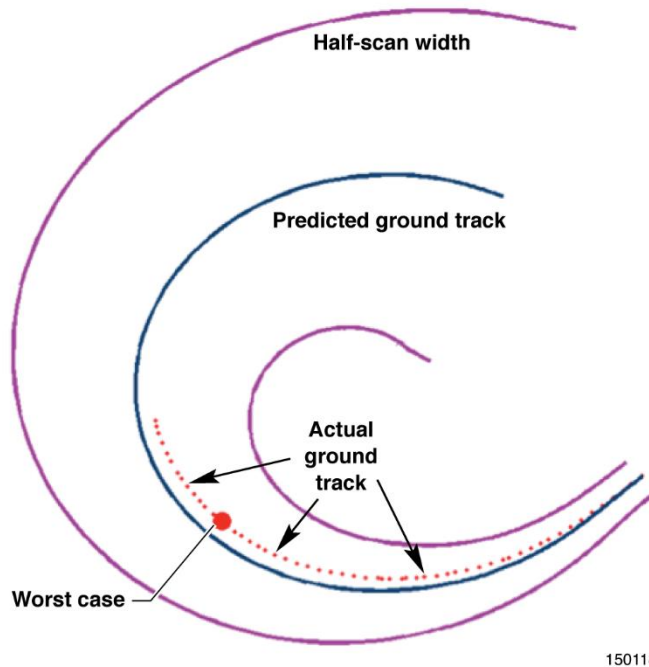


Figure B15. Worst-case point: inside of the scan.

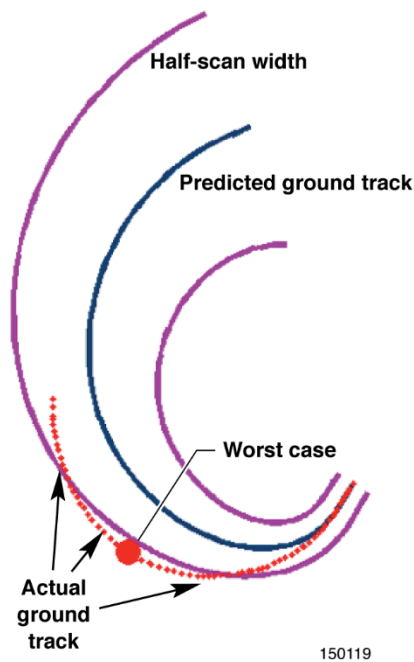


Figure B16. Worst-case point: outside of the scan.

In some cases, the ground track was never farther away from the predicted ground track than the half-scan width. The DROID stayed above the terrain included in the scan pattern. In such cases, the worst case was the point at which the actual ground track was farthest from the prediction. Figure B15 shows the worst-case analysis plot for such a case, flight 12 event 1. The red actual ground track is clearly inside the

purple scan, and the worst case is the point at which the red actual ground track is farthest from the blue predicted ground track.

In other cases, the actual ground track was farther away from the prediction than the half-scan width. In these cases, the DROID flew over terrain not necessarily accounted for by the algorithm. In such cases, the worst case was the point at which the flightpath was farthest outside the scan. Figure B16 shows the worst-case analysis plot for such a case, flight 18 event 6. The red actual ground track extrudes past the purple scan, and the worst case is the point at which the red actual ground track is farthest outside the purple scan.

The methods for finding the distance along the trajectory prediction and the half-scan width at each predicted ground track point are described in the following section.

Trajectory Normalization Analysis

In order to assess how well the overall Auto GCAS algorithm was working, a normalization analysis was used to compare key points from multiple test runs all on the same plot. This normalization method was used to show key points at the trigger posts and the location of the worst-case mismatch between the actual trajectory compared to the prediction as described in the appendix sections “Determination of Trigger Post Locations” and “Determination of Worst-Case Mismatch Locations.” The following discussion assumes a known latitude and longitude for each point in question.

The premise of the normalization analysis was that any desired point could be characterized by a distance along the trajectory prediction and distance perpendicular to the trajectory prediction. These distances could also be expressed as percentages of the total length and percentages of the corresponding width.

Scans for individual flight-test runs had different shapes and lengths depending on factors such as initial turn rate, airspeed, and wind. In Figures B17 through B22, comparisons of the trajectory prediction to the right were generated by changing the initial turn rate (bank), airspeed, and wind, respectively. A similar set of scans could be generated for a trajectory prediction to the left or straight. In each set of comparison figures, the trajectory prediction shown on the left-hand side of the page is based on the DROID initially traveling at 0 deg bank and 75 KTAS with no wind. Figure B17 compares the no-bank trajectory prediction to a trajectory prediction with an initial 40-deg left bank in figure B18. The DROID needs time to adjust from turning left to turning right, so the initial opposite bank makes the turn somewhat wider. Conversely, an initial bank in the same direction of the avoidance makes the turn tighter. Figure B19 compares the 75-KTAS trajectory prediction to a trajectory prediction with an initial speed of 60 KTAS in figure B20. The slower airspeed results in a tighter turn radius. Figure B21 compares the trajectory prediction with no initial crosswind to a trajectory prediction with a 20-kn crosswind from the right in figure B22. A crosswind of that magnitude from that direction tightens the turn radius considerably, while a crosswind in the opposite direction widens the turn radius.

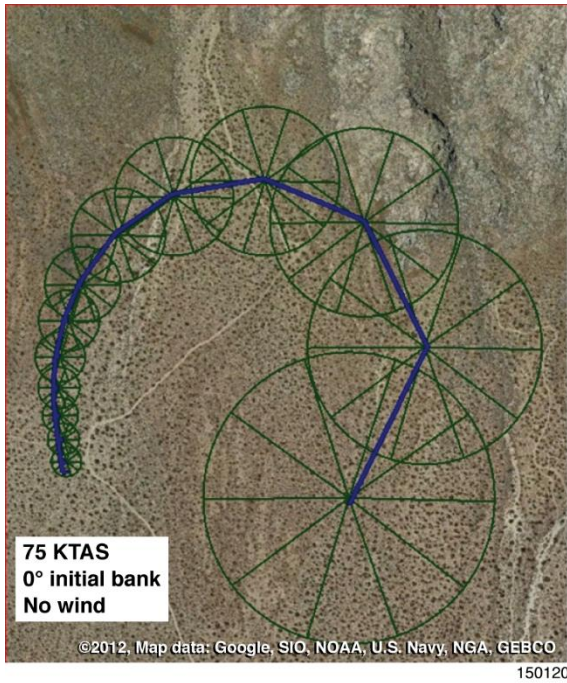


Figure B17. Trajectory prediction with no initial bank.

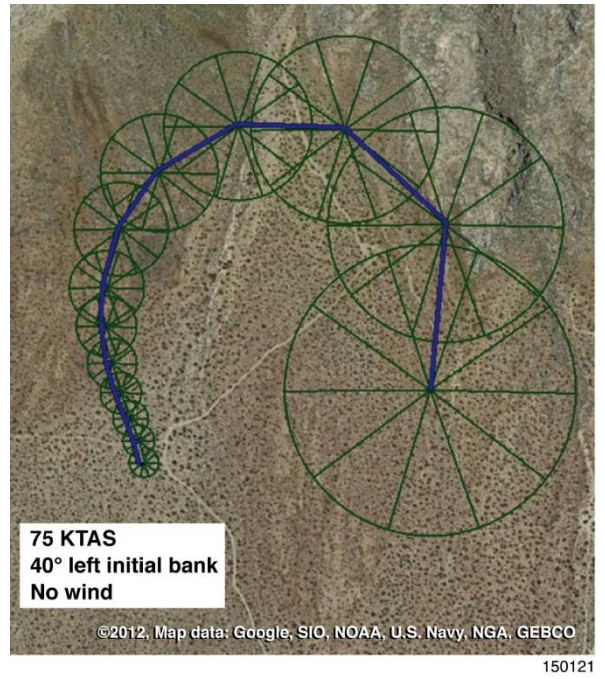


Figure B18. Trajectory prediction with initial 40-deg left bank.

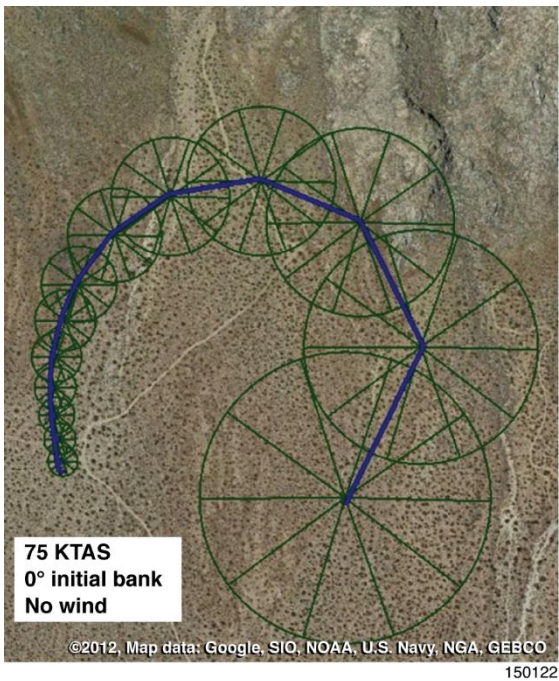


Figure B19. Trajectory prediction with 75-kn initial speed.

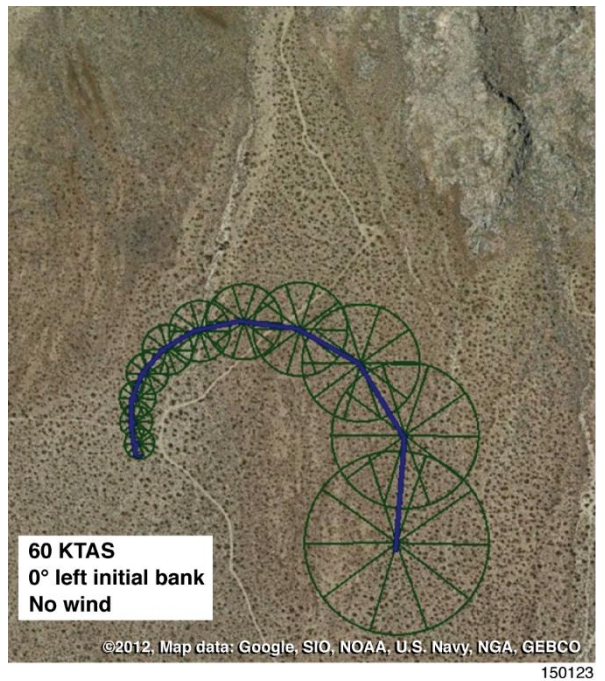


Figure B20. Trajectory prediction with 60-kn initial speed.

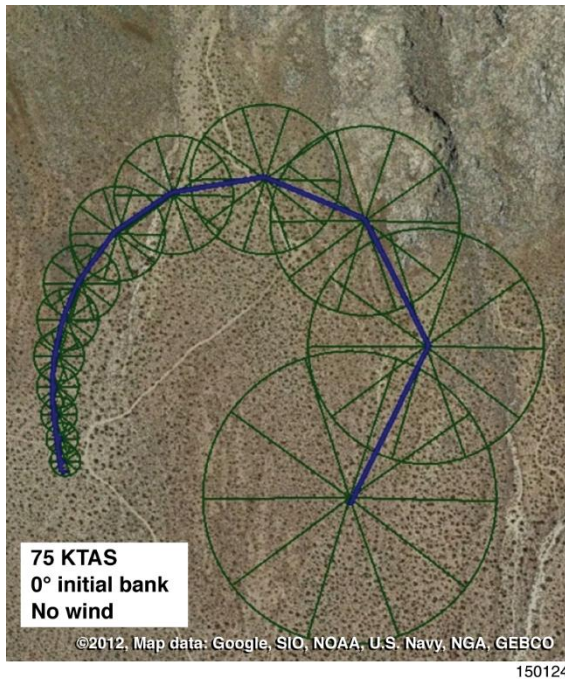


Figure B21. Trajectory prediction with no initial wind.

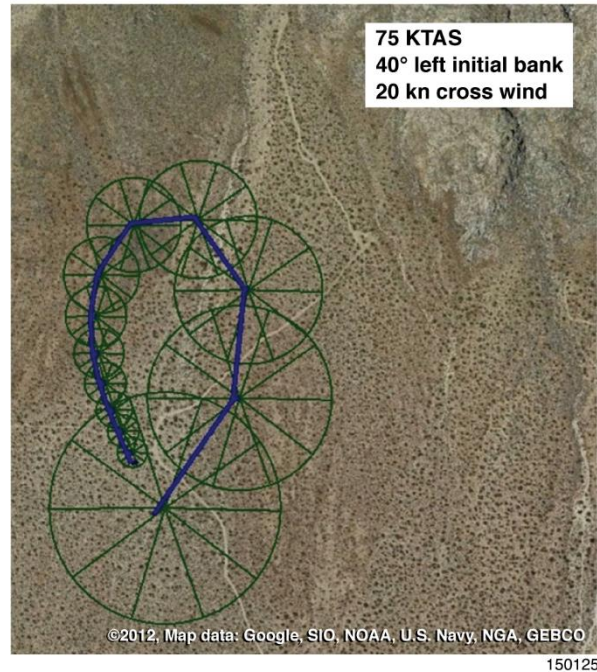


Figure B22. Trajectory prediction with 20-kn initial crosswind.

A single scan shape was selected to use as a reference for showing the key points from all of the flight-test runs. This reference scan shape was selected from flight 12 event 4, initiated at 73 KTAS, 0.5 deg bank, with a negligible crosswind. By normalizing the location of each point in question relative to the reference scan shape, those points could all be placed on one plot even if the original avoidance maneuver was to the left, straight, or right. The results shown in figures 54 and 58 of the main body of this report used the techniques discussed in the following paragraphs. The results of this analysis are also shown on histograms in figures 55, 56, 59, and 60 of the main body of this report, to provide an alternative summary.

Finding Distances

The trajectory prediction on the smartphone determined the centerline at multiple bin locations. The spacing between each bin was a function of how dynamic the expected avoidance maneuver would be at that location. The bins were spaced closer together during dynamic portions of the maneuver such as during g-onset or roll-rate onset. The bins were spaced farther apart if the expected avoidance maneuver at that location would be relatively stabilized.

Given the latitude and longitude of individual bin locations along the centerline of a trajectory prediction from flight-testing, the first steps were to find the total length of the centerline and the scan pattern width at each bin location. A visual representation of this method is shown in figure B23. Each distance point is shown as a blue dot. At the beginning of the trajectory (the most dynamic portion of the maneuver), the dots are so close together that they appear to form a solid line. The total distance is the sum of the straight line distances between neighboring distance points.

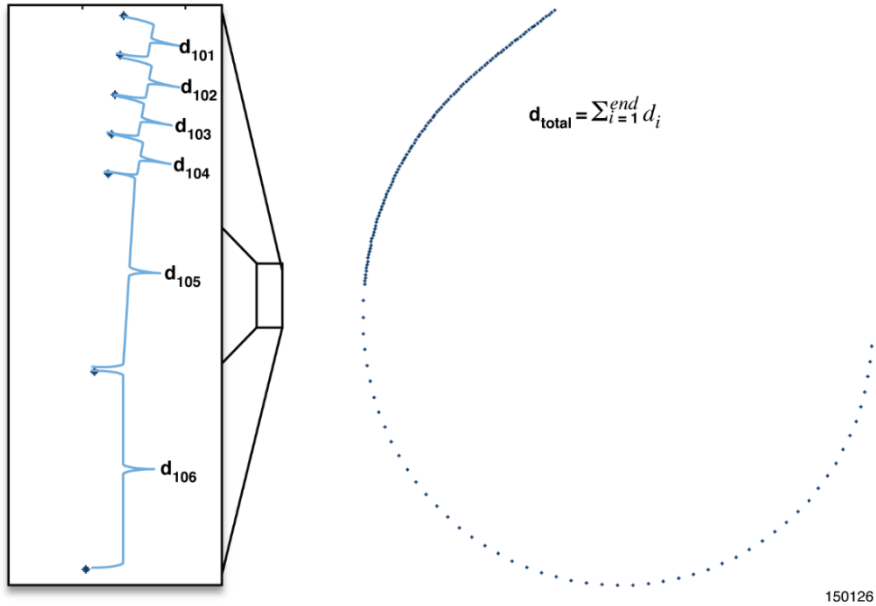


Figure B23. Method for finding total distance along trajectory prediction.

The scan pattern width could then be found at any distance point along the trajectory prediction using equation (B1). In order to match the values used during SUAV Auto GCAS flight test in equation (B1), *HorizontalUncertainty* = 50 ft, and *UncertaintyGrowthAngle* = 10° for turning scans and 5° for straight scans. The results of this method are shown in figure B24. By plotting the purple half-scan width on each side of the blue trajectory prediction, the entire scan is shown.

Half Scan Width

$$= \sqrt{\text{HorizontalUncertainty}^2 + (\text{DistanceAlongTPA} * \sin(\text{UncertaintyGrowthAngle}))^2}$$

(B1)

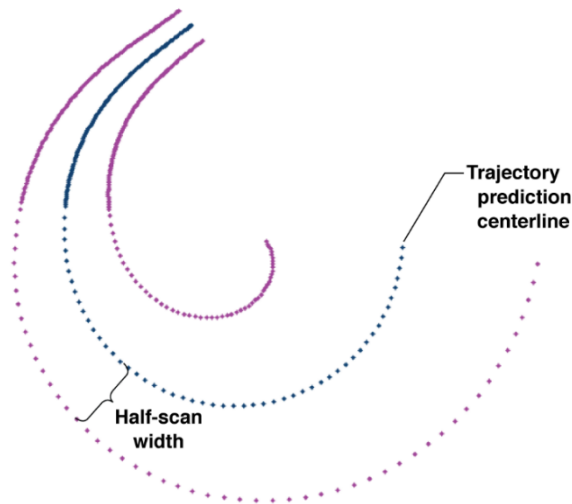


Figure B24. Half-scan width.

Once the total length of the centerline and the scan pattern width at each point along the entire length were known, the next step was to identify where key points of interest from a flight-test maneuver were located relative to the centerline. As mentioned at the beginning of this section, those key points could be trigger posts or they could be the location of the worst-case mismatch between the actual trajectory and the prediction. Given the latitude and longitude of a key point in question, distances along the centerline and perpendicular to the centerline were needed.

The two distance point locations along the centerline closest to the key point in question were found by using a distance function to find the distance from every point location along the centerline to the key point. The total distance along the centerline from maneuver initiation up until the distance point closest to the key point in question was determined by using the sum of the distances between neighboring distance point locations, as already described in the discussion above of figure B23.

To find the distance perpendicular to the centerline, distance functions were used to determine the perpendicular distance, as shown in figure B25. The two closest bin locations were mathematically connected by a straight line. A perpendicular to that line was determined which also intersected with the key point. The length of this line was used as the distance perpendicular to the centerline.

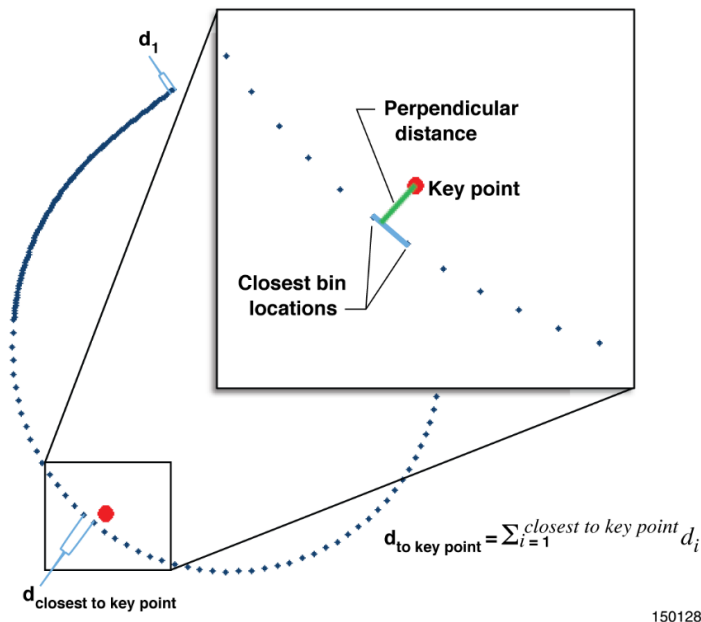
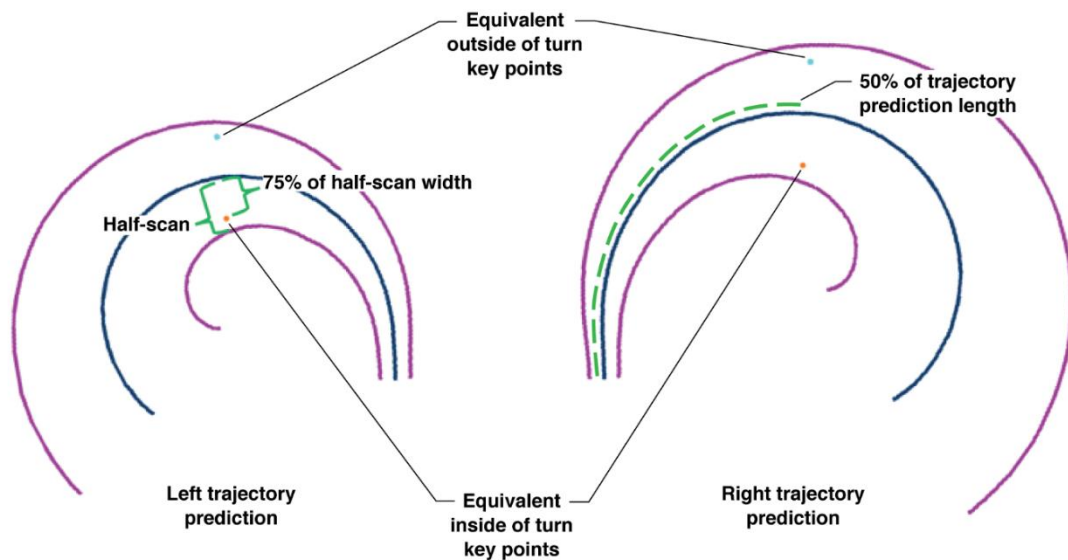


Figure B25. Method for finding perpendicular distance at a given point along the trajectory prediction.

Nondimensionalizing Distances and Plotting on a Representative Trajectory Prediction Algorithm

In order to make the distances nondimensional, the perpendicular distance was divided by the half-scan width at that location to convert into a percentage. Similarly, the distance along the trajectory prediction from maneuver initiation to the key point was converted into a percentage by dividing by the total length of the trajectory prediction. In addition, to transpose a point from a right or straight trajectory prediction onto the intended reference (left trajectory prediction), the key point location inside or outside of the trajectory prediction was used. The right trajectory prediction shown in figure B26 shows a point in orange that is 50 percent along the trajectory prediction length, and 75 percent of the half-scan width inside the curve of the trajectory prediction. The right trajectory prediction shown in figure B26 also shows a point in

blue that is 50 percent along the trajectory prediction length, and 75 percent of the half-scan width outside the curve of the trajectory prediction. The left trajectory prediction shown in figure B26 shows how the orange and blue points were transposed onto the reference left trajectory prediction.



150129

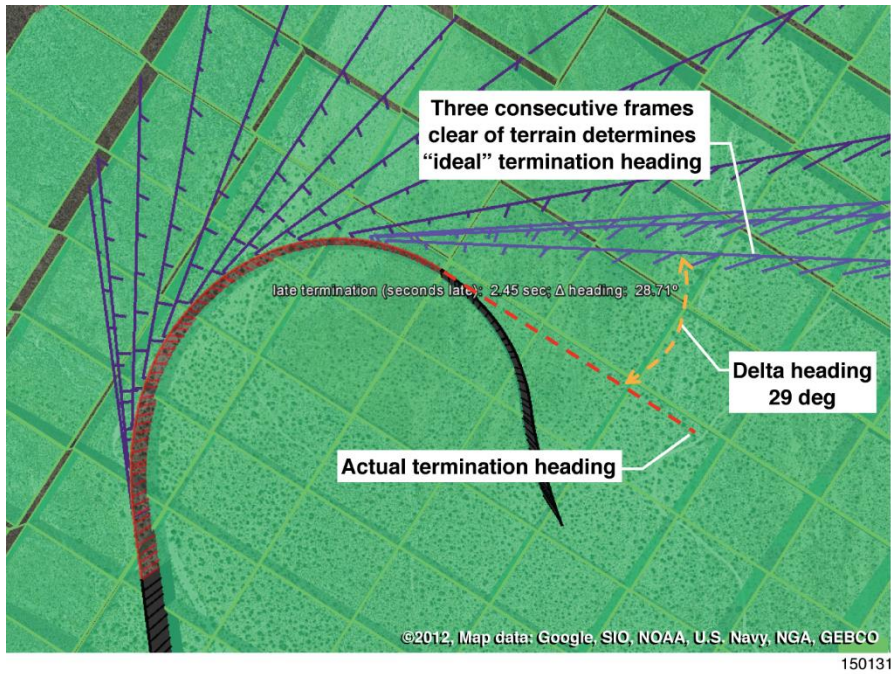
Figure B25. Example for non-dimensional key point locations.

Termination Logic Cross-Checks

When an SUAV avoidance maneuver was not terminated by the safety pilot, the Auto GCAS algorithm determined when the avoidance maneuver was no longer necessary and returned control to the ground control operator. In order to assess the timeliness of the avoidance maneuver termination, the flight-test results were compared with an ideal termination defined by generating a straight line tangent to the avoidance flightpath. A straight line tangent (including the current climb rate) was used as a way to identify that there was no obstructing terrain directly in front of the DROID. Although other methods could have been used, this was considered the simplest and would most directly correlate with the view of the pilot (in this case, the view provided by the forward-looking video camera). When the straight tangent line was projected to be clear of the DEM terrain for three consecutive time frames (approximately 0.2 s per frame), that third frame was considered the ideal termination.

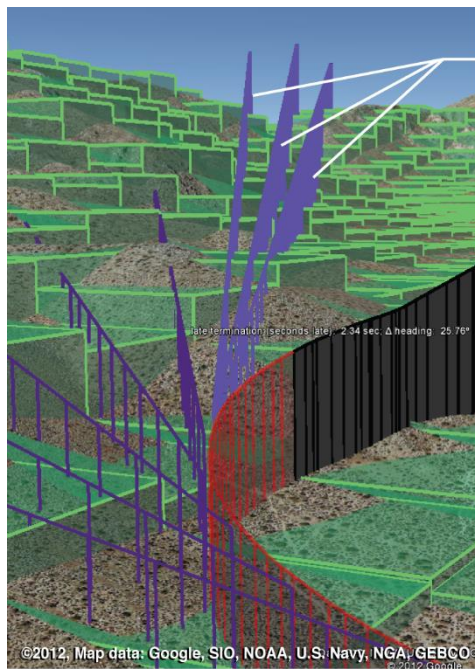
The purpose for these termination logic cross-checks was to determine whether the Auto GCAS algorithm terminated the maneuver earlier than it should have, at about the right time, or later than necessary. As a general result, when the DROID maneuvered left to avoid terrain, the software logic caused the avoidance maneuver to terminate earlier than it should have. For a right avoidance, the Auto GCAS maneuver tended to terminate later than necessary. The overall results are described in the main body of this report; the methods used are described below.

Figure B27 illustrates how the Auto GCAS algorithm determined when to terminate the avoidance maneuver using an example from flight-testing. The black line in figure B27 represents the actual flightpath of the DROID prior to the avoidance maneuver. The navy blue line in figure B27 represents the trajectory prediction at initiation (in this case, the Auto GCAS algorithm determined that the DROID should avoid terrain by executing a left turn). The red line in figure B27 represents the actual flightpath during the avoidance maneuver. To determine when an avoidance maneuver should be terminated, the Auto GCAS



150131

Figure B28. Tangential lines (violet) drawn from the avoidance flight path (red).



150132

Figure B29. Right-turning avoidance terminated late.

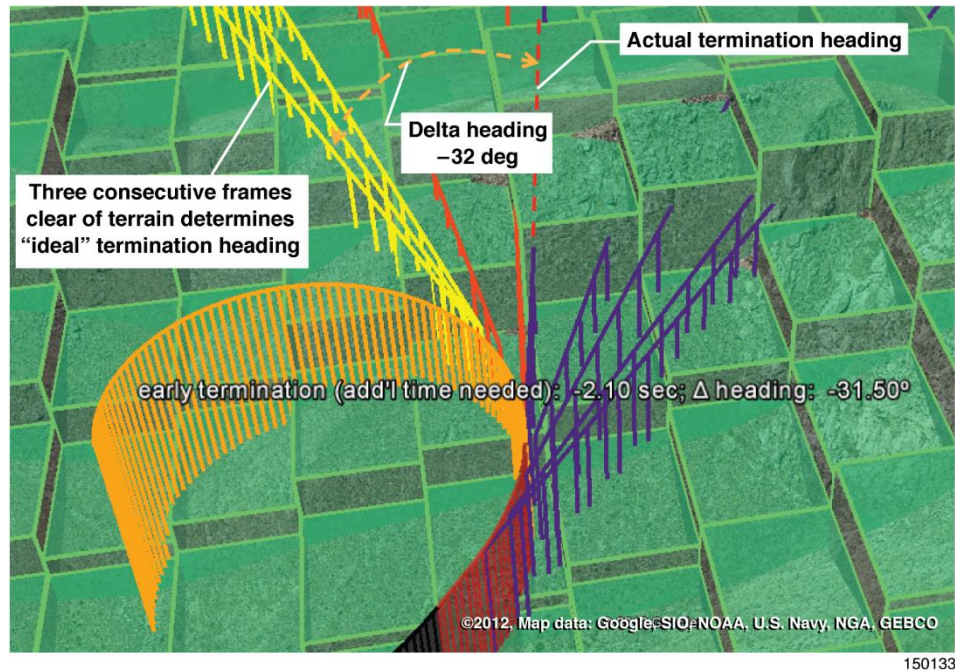


Figure B30. Left-turning avoidance terminated early.

Next, three arc-second buffered DEM terrain tiles (represented by the green polygons) were added. The buffered terrain added the TCB value to the DEM altitude but did not include any built-in buffers. Therefore, the results from any test run could be evaluated in the same manner as if the TCB were set to 0. The DEM terrain tiles were sized at three arc-seconds to be consistent with the resolution used by the Auto GCAS algorithm and to minimize computational time. This calculation being post-flight, the theoretical accuracy could have been improved using higher resolution DEM tiles (as fine as the one-third-arc-second resolution of the NED source data) but that increased accuracy was not considered necessary for this analysis.

The ideal termination heading was defined to be when the green DEM polygons did not block three consecutive tangential paths (as shown by the three lighter colored violet lines in figures B28 and B29). The determination of when the tangent lines no longer intersected with the DEM terrain was accomplished as a numerical calculation but is shown using Google Earth to help visualize the concepts.

Figures B28 and B29 illustrate termination calculations for similar right-turning avoidance maneuvers. Figure B28 shows the avoidance maneuver as viewed from almost directly above. Figure B29 shows the same maneuver from the perspective of a lower viewing angle. The darker violet lines represent the tangential paths that were blocked by terrain (drawn every five frames), and the three right-most lighter violet lines represent the three-frames-of-persistence clear of terrain (drawn every frame). In these cases, the right-turning avoidances illustrated in figures B28 and B29 indicate that the avoidance maneuver terminated approximately 26 to 29 deg later than necessary.

The short vertical lines along each tangential path represent a distance equivalent to one arc-second. Greater accuracy could have been achieved with smaller intervals for the tangential paths (that is, one-third arc second instead of one arc-second resolution) but computational time would have been increased significantly as a result.

Figure B30 illustrates termination calculations for a left-turning avoidance maneuver. The red flightpath in figure B30 once again represents the actual avoidance maneuver path up to the point at which the Auto GCAS algorithm terminated the maneuver. The violet straight lines (drawn every fifth frame) represent the tangential paths along the red actual avoidance path. The orange semicircle represents an extrapolated avoidance trajectory that the DROID would have taken had it continued the avoidance maneuver. The dark-orange straight lines (drawn every fifth frame) represent the tangential paths along the extrapolated orange semicircle. Finally, the three straight yellow lines (drawn at each frame) represent the consecutive tangential paths which do not intersect with the green DEM polygons.

To determine an extrapolated avoidance trajectory, several new elements were needed. Since the smartphone hosting the Auto GCAS algorithm did not use a consistent time interval for each frame, an average time interval was selected using the last five time increments in the red portion of the actual avoidance maneuver (see figure B31). Next, a constant radius was calculated for the extrapolated avoidance maneuver based on the arc between the last two points of the actual avoidance maneuver. This radius was calculated from equation (B2):

$$s = \theta \cdot R \tag{B2}$$

where θ is the heading change calculated from the last two points in the red avoidance path in figure B30, and s is the arc distance between those two points. In this example case, the left-turning avoidance maneuver shown in figure B30 terminated approximately 32 deg earlier than it should have. Although this extrapolation method worked reasonably well on most runs, it was also susceptible to noise in the source data, causing some uncertainty in the results. An alternative method could use an extrapolated radius based on the average arc over the previous several frames.

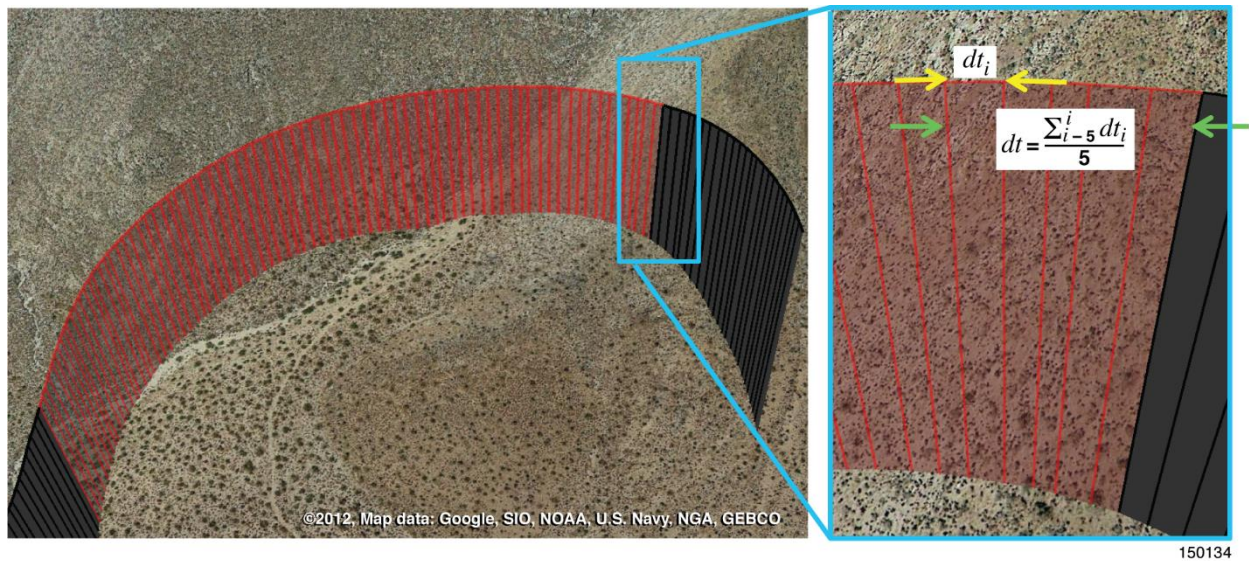


Figure B31. Differential time element illustration.

For the extrapolated runs the delta time and delta heading were based on the new incremental elements described above.

Appendix C: Flight Log

This appendix presents a summary of all flights that were related this project. Table C1 presents the first few flights that were accomplished on the DROID aircraft to obtain basic trajectory prediction data along with the SUAV system checkout flights. Table C2 presents all of the SUAV test flights that were accomplished to evaluate the overall system and obtain test data.

Table C1. Preliminary flights.

Flight number	Flight date	Duration, min	Smartphone location	Test site	Test types*	Notes
DROID #1	4/14/2011	47	N/A	North Base	10 sequential axis PARS runs**	Normal inputs to Piccolo II
DROID #2	4/14/2011	33	N/A	North Base	8 sequential axis PARS runs**	Normal inputs to Piccolo II
DROID #3	4/14/2011	10	N/A	North Base	No test runs**	Normal inputs to Piccolo II
DROID #4	4/14/2011	29	N/A	North Base	9 sequential axis PARS runs**	Normal inputs to Piccolo II
DROID #5	4/14/2011	35	N/A	North Base	13 sequential axis PARS runs**	Normal inputs to Piccolo II
SUAV #1	9/28/2011	44	Van	North Base	9 combined axis PARS runs**	UI inputs to Piccolo II
SUAV #2	9/28/2011	42	Van	North Base	7 Auto GCAS functional checks	None
SUAV #3	9/28/2011	51	Van	North Base	17 combined axis PARS runs**	UI inputs to Piccolo II
SUAV #4	10/7/2011	26	Van	Rosamond Lakebed	3 Auto GCAS runs	Intermittent RPM sensor
SUAV #5	10/7/2011	15	Van	Rosamond Lakebed	No test runs	Failed RPM sensor
SUAV #6	10/7/2011	27	Van	Rosamond Lakebed	3 Auto GCAS runs	Intermittent RPM sensor
SUAV #7	10/18/2011	53	Van	Rosamond Lakebed	12 Auto GCAS runs	TM dropouts near hill
SUAV #8	10/18/2011	51	Van	Rosamond Lakebed	2 Auto GCAS runs 11 combined axis PARS runs	None

Notes:

* The number of runs listed represents the runs attempted. Some runs were not completely successful for a variety of reasons.

** The six flights identified with double asterisks were accomplished to determine the parameters needed to define the trajectory predictions. Pilot Activated Recovery System (PARS) maneuvers were initiated in each axis to obtain the required data. PARS maneuvers were executed using the same command sequence intended for the corresponding flyup maneuvers.

Table C2. Primary test flights.

Flight number	Flight date	Duration, min	Smartphone location	Test site	Test types*	Notes
SUAV #9	10/21/2011	55	Van	GCAS valley	13 Auto GCAS runs	Top of small hill TM dropouts near hill
SUAV #10	10/21/2011	55	Van	GCAS valley	11 Auto GCAS runs	Top of small hill TM dropouts near hill
SUAV #11	11/7/2011	49	Van	GCAS valley	7 Auto GCAS runs	Base of small hill TM much improved
SUAV #12	11/7/2011	35	Van	GCAS valley	6 Auto GCAS runs	Medium hill
SUAV #13	3/5/2012	59	Aircraft	Rosamond Lakebed	6 Auto GCAS runs (no flyups) 1 nuisance test rehearsal	No flyups: traced to Piccolo II roll rate values
SUAV #14	3/15/2012	32	Aircraft	GCAS valley	5 Auto GCAS runs	Smartphone data did not record
SUAV #15	3/15/2012	2	Aircraft	GCAS valley	No test runs	Smartphone dislodged on takeoff
SUAV #16	3/15/2012	53	Aircraft	GCAS valley	9 Auto GCAS runs	FAIL states disrupted testing
SUAV #17	3/29/2012	59	Van	GCAS valley	13 Auto GCAS runs	Near Fremont Peak (Smartphone back in van to allow testing without disruptions due to FAIL states.)
SUAV #18	3/29/2012	63	Van	GCAS valley	11 Auto GCAS runs	Lakebed and medium hill Guest pilots.
SUAV #19	3/29/2012	54	Van	GCAS valley	12 Auto GCAS runs	Medium hill
SUAV #20	5/31/2012	40	Aircraft	GCAS valley	7 nuisance test ridge crossings	Small hill (Smartphone back in DROID with improved FAIL states.)
SUAV #21	5/31/2012	52	Aircraft	GCAS valley	12 Auto GCAS runs 3 nuisance test valley patrols	Medium hill

Note:

* The number of runs listed represents the runs attempted. Some runs were not completely successful for a variety of reasons.

Appendix D: Open Discrepancy Reports

Table D1 provides a list of each NASA Discrepancy Report (DR) that remained open at the end of the SUAV Auto GCAS project. An additional 29 DRs were written during the project but were closed by the Configuration Control Board, normally because the problem had been confirmed as fixed. The main intent of table D1 is to allow staff of similar future projects to decide whether each item warrants improvement. A paraphrased and shortened description for each DR is provided in the table.

Table D1. Open Discrepancy Reports.

DR #	Title	Short description and disposition	Auto GCAS component
11-125	TRANSITION TO IDLE KEEPS FLYUP ACTIVE	<ul style="list-style-type: none"> When a flyup is engaged, transitioning to IDLE keeps the flyup active. Workaround: Click on another mode to transition out of flyup mode 	User interface
11-126	INTERMITTENT CONNECTION BETWEEN PHONE AND USER INTERFACE	<ul style="list-style-type: none"> User interface timeout on AGCAS connection occurs at random times, causing a FAIL mode to be asserted. Frequency is very low, but random in nature. Workaround: Click on appropriate mode on user interface to resume normal function. 	User interface / smartphone software
11-130	INAPPROPRIATE FLYUP TERMINATION DUE TO P-FACTOR	<ul style="list-style-type: none"> Termination of some flyups is delayed longer than necessary. Data show that flyups to the right result in delayed termination; flyups to the left result in early termination. Recommendation: Future SUAV Auto GCAS projects should consider modifying the termination logic to use a straight trajectory that is unaffected by P-factor. 	Algorithm on smartphone
12-108	FAIL INDICATIONS	<ul style="list-style-type: none"> FAIL indications occurred shortly after flyup initiation on at least three test runs. These FAIL indications were somewhat disruptive to the normal test flow and degraded the intended test data. On later missions the FAIL "timeout" was changed from 0.5 to 5.0 s to ensure no test disruption, but 5.0 s may not be the optimal setting. Recommendation: Determine optimal FAIL timeout value for future Auto GCAS projects. 	
12-110	STREAMLINE CAPABILITY FOR PILOT CONTROL AFTER AGCAS FLYUP/ABORT	<ul style="list-style-type: none"> When the DROID pilot is commanding the from the pilot control station at the time a FLYUP is initiated, the pilot does not have a single-action command ability to regain control of the DROID. Recommendation: Mechanize system so that a single action command by the pilot will regain control after a FLYUP or ABORT. 	Ground control van and user interface

12-111	AUTOPILOT DID NOT RESPOND TO FLYUP COMMAND	<ul style="list-style-type: none"> • A right FLYUP command was sent but there was no response from the autopilot. This occurred only once on the three flights on March 29. • Recommendation: Additional research as needed to support future Auto GCAS development efforts. 	Smartphone to Piccolo II interface
12-112	ILLOGICAL FLYUP TRAJECTORIES IN RECORDED DATA	<ul style="list-style-type: none"> • The left, straight, and right trajectories recorded at FLYUP initiation were all identical. • Recommendation: Additional research as needed to support future Auto GCAS development efforts. 	Algorithm on smartphone
12-113	FLYUP WENT OUTSIDE SCAN PATTERN	<ul style="list-style-type: none"> • The actual flyup trajectory went outside the scan pattern by approximately 50 ft horizontally. The altitude approached 25 ft of the TCB, indicating that going outside the scan may have contributed to reduced terrain clearance. This run may have been influenced by wind changes after FLYUP initiation. • Recommendation: Additional research as needed to support future Auto GCAS development efforts. 	Algorithm on smartphone
12-114	INDECISIVE FLYUP TERMINATION	<ul style="list-style-type: none"> • Three additional short flyups occurred after termination of the first flyup (over total duration of 7 s). Cases with one or two additional flyups occurred on other flights. It would be appropriate for every flyup to terminate without subsequent flyups for at least several seconds. • Recommendation: Additional research as needed to support future Auto GCAS development efforts. 	Algorithm on smartphone
12-115	FLYUP WENT INSIDE SCAN PATTERN	<ul style="list-style-type: none"> • Flyup was as much as 130 ft inside the scan. Flyups with this characteristic pose an increased risk of nuisance potential but are unlikely to increase risk of terrain impact. • Recommendation: Additional research as needed to support future Auto GCAS development efforts. 	Algorithm on smartphone
12-116	POSSIBLE LATE FLYUPS	<ul style="list-style-type: none"> • The data show that the predicted trajectories were well below the buffered terrain for both left and right options. This result implies a large jump in the TPA compared to a single frame earlier. • Recommendation: Additional research as needed to support future Auto GCAS development efforts. 	Algorithm on smartphone
12-117	GPS DISCONTINUITIES	<ul style="list-style-type: none"> • GPS position discontinuities up to 132 ft occurred (in between frames) on several flights. This problem was later traced to a known problem in the Piccolo II software version used during flight-testing. • Recommendation: Update Piccolo II software before future tests using the DROID. 	Piccolo II software
12-118	AUTOPILOT EXCEEDS LIMITS	<ul style="list-style-type: none"> • The DROID was able to establish a descent rate over 2500 ft/min and a bank angle of almost 50 deg. These values were well in excess of expected Piccolo II limits of @ 1000 ft/min and 40 deg bank. This problem was later traced to large-amplitude rudder 	Ground control van to Piccolo II interface

		<p>inputs applied by the pilot in the ground cockpit (rudder inputs went directly to the control surfaces and were not limited by the Piccolo II).</p> <ul style="list-style-type: none"> • Recommendation: Reconfigure rudder pedal commands so Piccolo II limits will not be exceeded, or advise pilots on limited use of rudder pedals. 	
12-120	NO CAPABILITY FOR COCKPIT PILOT TO DISENGAGE FLYUPS	<ul style="list-style-type: none"> • The ground cockpit pilot could not disengage a flyup and regain control (as the design was then implemented). The main impact of not implementing this capability was the inability to properly conduct Auto GACS nuisance testing. • Recommendation: The ideal mechanization would allow a single HOTAS action to terminate a flyup in progress and not allow that flyup to resume. “New” flyups could occur unless the pilot chooses to continue holding the HOTAS. 	Ground control van to user interface
12-121	ABORT BUTTON FUNCTIONALITY	<ul style="list-style-type: none"> • The UI ABORT button sends a signal to terminate the current flyup but another flyup is almost immediately recalculated and performed. • Workaround: Instead of clicking on the UI ABORT button, change the AGCAS mode to STANDBY or IDLE until ready to resume avoidance maneuvers. 	User interface
12-122	AGCAS SOFTWARE RECORDED REDUNDANT LATITUDE AND LONGITUDE VALUES IN SMARTPHONE	<ul style="list-style-type: none"> • For eight flight-test events the Auto GCAS software recorded redundant trajectory prediction latitude and longitude data on the smartphone SD card. These cases are related to DR 12-112. • Recommendation: Additional research as needed to support future Auto GCAS development efforts. 	Algorithm on smartphone

References

1. Skoog, Mark, and Kevin E. Prosser, "Advanced Fighter Technology Integration/F-16 Automatic Ground Collision Avoidance System Evaluation," AFFTC-TR-99-28, December 2000. Available from AFRL/VAAI, Wright-Patterson AFB OH 45433-7542.
2. Sorokowski, Paul, et al., "Automatic Ground Collision Avoidance System Fighter Risk Reduction Project," AFFTC-TIM-10-05, December 2010. Available from AFRL/RBCC, Wright-Patterson AFB, OH 45433-7542.
3. Moore, Duane P., and Kyle W. Schlappi, "USAF F-16 Block 40/50 Test and Evaluation for the Automatic Ground Collision Avoidance System (Auto GCAS) and Pilot Activated Recovery System (PARS), 412TW-TR-13-04, October 2013. Available from AFLCMC/WWM, Wright-Patterson AFB, Ohio 45433-7424.
4. Webb, Charles H., and Ryan K. Owen, "F-22 Line in the Sky (LIS) Automatic Ground Collision Avoidance System (AGCAS) Test and Evaluation," 412TW-TR-13-42, January 2014. Available from Air Force Life Cycle Management Center, F-22 Division, Wright-Patterson AFB OH 45433-7017.
5. Rumsfeld, Donald, "Memorandum for Secretaries of the Military Departments, Subject: Reducing Preventable Accidents," U06916-03, May 19, 2003.
6. Gates, Robert M., "Memorandum for Secretaries of the Military Departments, Subject: Zero Preventable Accidents," May 30, 2007.
7. Defense Safety Oversight Council Aviation Safety Improvements Task Force Safety Technology Working Group, "Fighter/Attack Automatic Collision Avoidance Systems Business Case," May 2006.
8. Skoog, Mark, Jack Ryan, and Mike Marston, "ACAT/SUAV Objectives and Requirements Document," ACAT/SUAV-ORD-001, January 2011. Available from AFRC – Code Z.
9. "Automatic Collision Avoidance Technology / Fighter Risk Reduction Project Auto GCAS Requirements Guide," draft version 1.0, February 9, 2008. Available from AFRL/RQQ.
10. Gruca, Dena, and Jack Ryan, "ACAT/SUAV System Design Description (SDD)," ACAT/SUAV-SDD-001, January 2012. Available from AFRC – Code Z.
11. Marston, Michael L., "Dryden Remotely Operated Integrated Drone (DROID) Project," Operational Concept Document, Revision A, December 2009.
12. Preliminary patent information, "Global Elevation Data Adaptive Compression System (GEDACS)," NASA ID DRC-012-001.