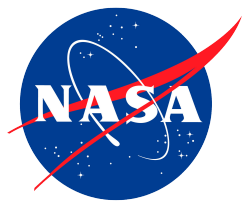
Formal Methods in the Development of Highly Assured Software for Unmanned Aircraft Systems

César Muñoz
cesar.a.munoz@nasa.gov
Safety-Critical Avionics Systems Branch
NASA, Hampton, VA, USA



Autonomous Control and Information Technology Seminar Series
NC A&T
April 9, 2021

A Numerical Challenge[1]

- Given the following sequence:
  - $a_0 = \frac{11}{2}$,
  - $a_1 = \frac{61}{11}$,
  - $a_{n+2} = 111 - \frac{1130 - \frac{3000}{a_n}}{a_{n+1}}$.
- Compute an approximation of a_{20}.

[1] Taken from *Arithmétique des ordinateurs*, J-M. Muller, 1989.

Why We Should Not Depend Upon Software[2]

"Software products - even software of modest size - are among the most complex artifacts that humans produce, and software development projects are among our most complex undertakings. They soak up however much time or money, however many people we throw at them."

"The results are only modestly reliable. After the most thorough and rigorous testing some bugs remain. We can never test all threads through the system with all possible inputs."

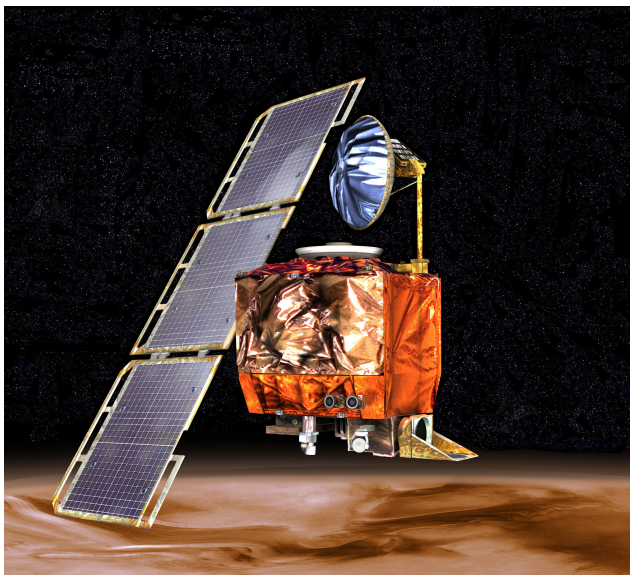
[2] *Digital Woes: Why We Should Not Depend Upon Software*, L. R. Wiener, 1993.

The Ariane 5 Disaster (1996)[3]

- Ariane 5's first test flight ended with the rocket self-destructing 37 seconds after launch.
- The accident cost approximately $370m.
- The cause of the accident was a malfunction in the control software triggered by a conversion from a 64-bit float into a 16-bit integer.

[3]Picture credits to DLR German Aerospace Center.

The Mars Climate Orbiter Lost (1998)[4]



- Communication with the Mars Climate Orbiter was lost as the spacecraft went into orbital insertion in Mars due to an error in the ground-based software.
- The estimated cost of the orbiter was \$125m.
- The error was due to a mismatch of non-SI unit of pound-force seconds instead of the SI unit of newton-seconds in two software modules.

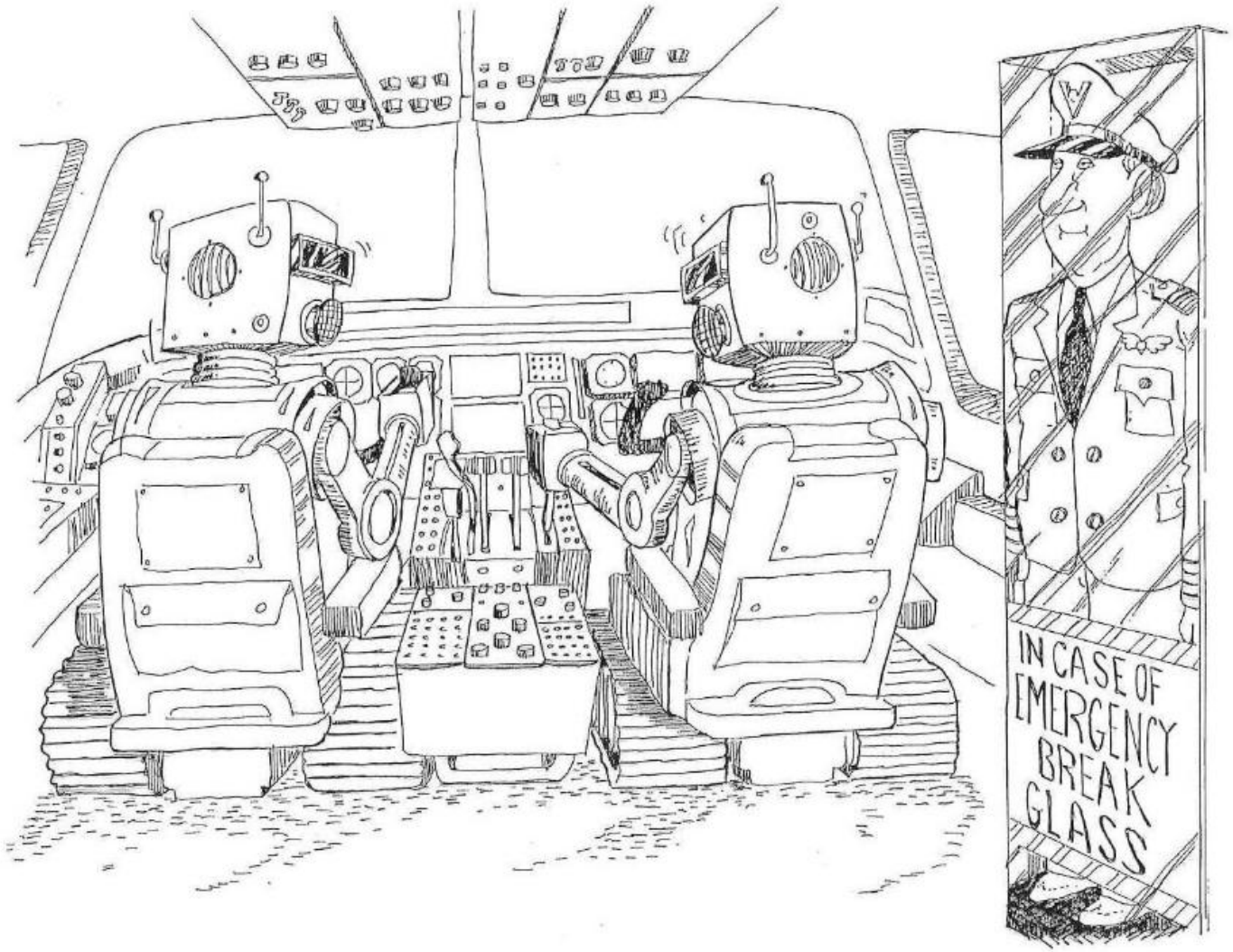
[4] Picture credits to JPL/NASA.

Safety in Air Transportation Systems

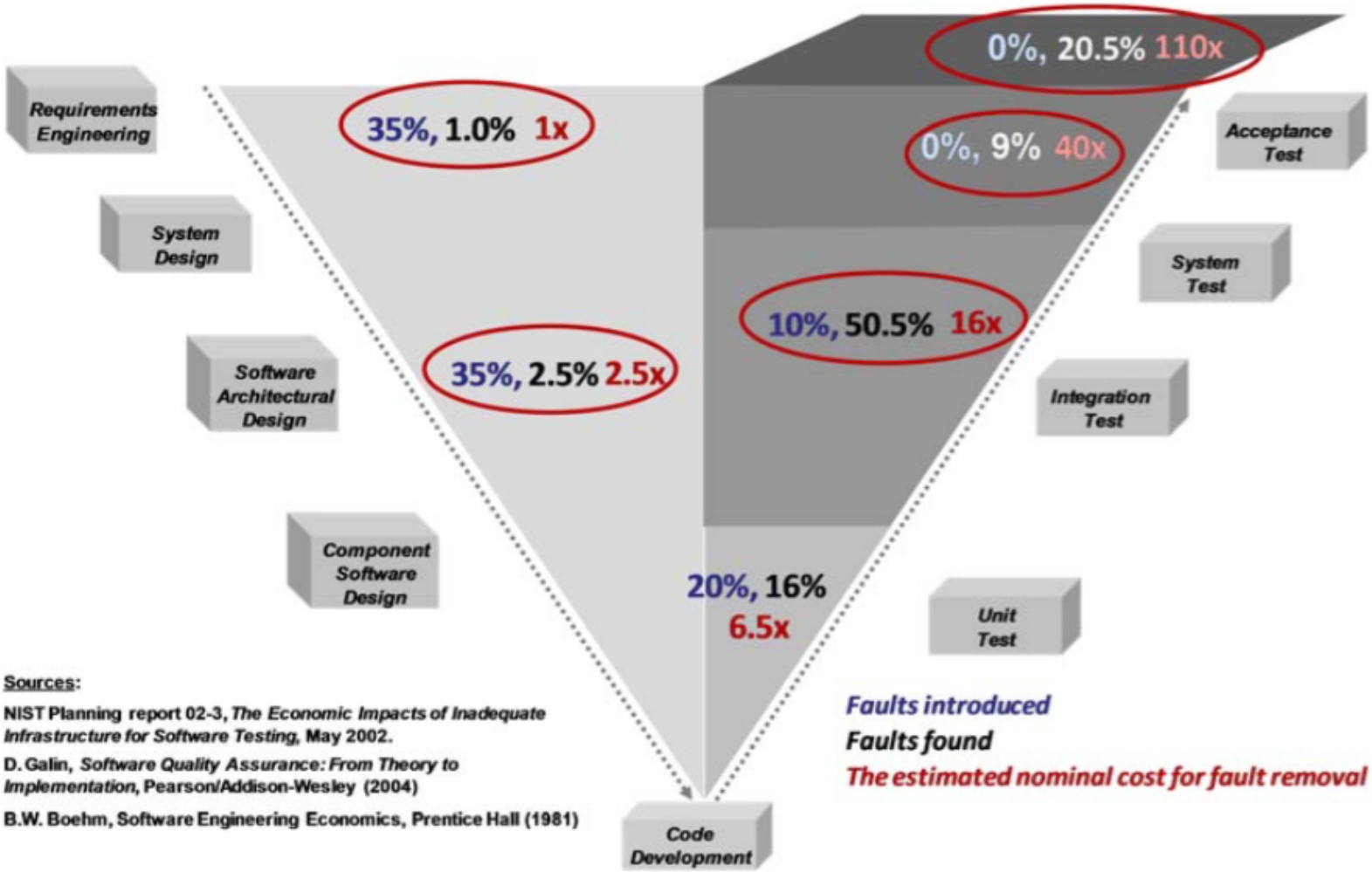
- Air transportation is the safest form of transport when measured by distance travelled.
- Standard software certification processes for avionics software, e.g., DO-178B and DO-178C, require safety-critical systems to be designed to less than one life loss per billion (10^9) hours of operation.
- New modes of air transportation, e.g., autonomous aircraft, and new computational models, e.g., deep-learning AI, create new validation, verification, and certification challenges.
- "*A Boeing 787 has 6.5 million lines behind its avionics and online support systems.The control software to run a U.S. military drone uses 3.5 million lines of code.*"[5]

[5] `https://www.visualcapitalist.com/millions-lines-of-code.`

The Cockpit of the Future



Error Leakage Rates Across Development Phases[6]



[6] Taken from [FGG+12].

Notations and Tools Used in DO-178B Projects[7]

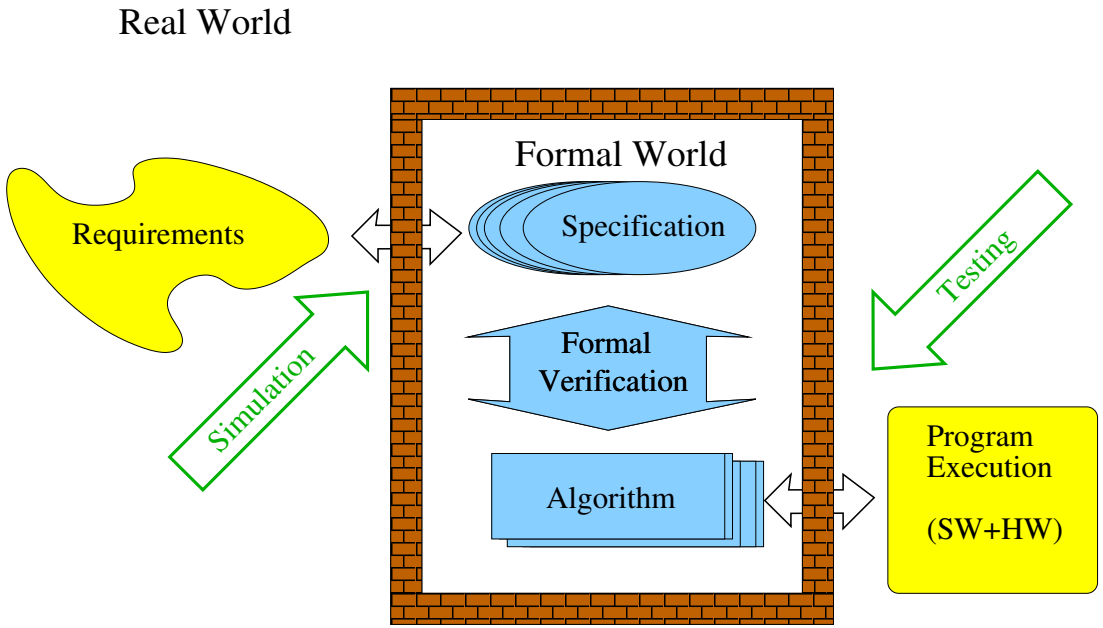
Notation	Sys Req	ICD	HL Sw Req	LL Sw Req	Hw Req
English	39	27	36	32	29
Tables/Diagrams	31	30	30	19	19
Exec. Models	7	1	8	8	1
...					

Tool	Sys Req	ICD	HL Sw Req	LL Sw Req	Hw Req
DOORS	23	13	22	18	12
Word Processor	19	20	18	17	16
Spreadsheet	5	4	5	4	3
...					

[7] Taken from [Fed09].

Formal Methods

- *Formal Methods* are mathematically rigorous techniques and tools for the specification, design, and verification of software and hardware systems.
- Formal specifications are well-formed statements in a mathematical logic.
- Formal verification uses a set of inferences rules that can be checked by a mechanical process.



State of the Art of Formal Methods for Systems Requirements

- Industry tools, e.g.,
  - Analysis of Semantic Specifications and Efficient generation of Requirements-based Tests (ASSERT) [SMD+17]
  - Specification and Analysis of Requirements (SpeAR) [GFH16]
  - Constrained Language Enhanced Approach to Requirements (CLEAR) [BMH+18].
- Benefits:
  - Formal, human-readable, requirements
  - Automated requirement analysis via model-checking and automated theorem proving
  - Automated generation of test cases, etc.
- However, these tools mostly support restricted logics, e.g.,lack of support for nonlinear arithmetic.

Formal Methods at NASA Langley Research Center (LaRC)

- Focus on next generation of air transportation systems, including autonomous unmanned aircraft systems (UAS).
- Use of expressive logics (Higher-Order Logic):
  - \+ Enable the specification and analysis of complex operational and functional requirements
  - \- Requires the use of interactive theorem proving. Hence, limited automation and steep learning curve
- The rest of this talk illustrates the use of FM at LaRC in the development of highly assured software for UAS.
  1. ADS-B's CPR Protocol
  2. Detect and Avoid Alerting Logic for UAS

I. Automatic Dependent Surveillance-Broadcast (ADS-B)

- ADS-B is a surveillance system based on global positioning systems that supports the next generation of air traffic management systems (NextGen).
- Aircraft equipped with ADS-B provide accurate surveillance information (3-D positions and velocities) to other aircraft in their vicinity and to ground stations.
- ADS-B is defined by an international standards committee (RTCA DO-260B/Eurocae ED-102A) [RTC09].
- Since 2020, ADS-B is mandatory in the US/Europe for most commercial aircraft. More than 40000 aircraft are now equipped with ADS-B.[8]

[8] `https://www.faa.gov/nextgen/equipadsb/`.

ADS-B Broadcast Message

Position Message

- Airborne message is 112 bits long, 56 of which is data frame,
- 35 bits of 56 are allocated for latitude and longitude,
- 1 bit for control, 17 for latitude, and 17 for longitude.
- Raw latitude/longitude encoded in 17 bits yields an accuracy of approximately 300m, but the requirement for NextGen is 5m!

Compact Position Reporting (CPR)

- CPR is an algorithmic component of ADS-B that compresses and recovers approximate location of an aircraft.
- CPR algorithm is based on the fact that higher order bits of raw positions are very unlikely to change over a short period of time.
- CPR uses a special coordinate system where lat/lon directions are divided into zones of approximately 360 nautical miles.

Where Is the Plane?[9]

Two ATC agencies 'blacklist' 787 over position-data flaw

10 DECEMBER, 2015 | SOURCE: FLIGHT DASHBOARD | BY: STEPHEN TRIMBLE | WASHINGTON DC

Most of the Boeing 787s delivered to date contain a software defect that, in at least five identified aircraft, have erroneously reported their location to controllers, prompting two air traffic management agencies to put the Dreamliner on a "blacklist" for certain services.

[9] https://www.flightglobal.com/news/articles/two-atc-agencies-blacklist-787-over-position-data-419916/.

Formal Analysis of CPR's Requirements [DMTM17]

- An ideal CPR algorithm, i.e., assuming infinite precision exact arithmetic, is not correct with respect to original requirements.
- However, an ideal CPR algorithm is correct for a tightened set of requirements.
- A straightforward single-precision floating-point (FP) implementation is unsound even under tightened requirements, e.g., encoding/decoding lat -77.368^o,lon 180^o has an error of about 1500 nmi.

Formal Analysis of CPR's Floating-Point Implementations [DMT+20]

- A double-precision floating-point implementation in C of CPR was formally verified with respect to tightened requirements.
- A 32-bit integer implementation of CPR in C has been formally verified and is being considered as reference implementation of CPR in RTCA DO-260B/Eurocae ED-102A.

Theorem 1

A position recovered by the proposed CPR implementation is never farther than $\sqrt{2} \times 2.2888 \times 10^{-5}$ *degrees from original position.*

II. Detect and Avoid of UAS

THE WALL STREET JOURNAL.

Home World U.S. Politics Economy Business Tech Markets Opinion Arts Life Real Estate

At Glencore, Mining Emperor Tries to Save His Realm

Volkswagen Helps Customers Identify Tainted Cars

Coca-Cola Urges FIFA Soccer Chief Blatter to Step Down

Sbarro Seeks New Life Beyond the Mall

BUSINESS

FAA: U.S. Airliner Nearly Collided With Drone in March

Incident Appears to be First Case of a Big U.S. Airliner Nearly Colliding With an Airborne Drone

By JACK NICAS

Updated May 9, 2014 7:56 p.m. ET

OFFBEAT

Pilot Says Drone Flew Past Jet Nearing J.F.K.

By PATRICK MCGEEHAN and JOSEPH GOLDSTEIN MARCH 5, 2013 12:17 PM 61 Comments

News

Quadcopter drone flew 'deliberately close' to UK passenger plane

The incident occured at Southend Airport with the pilot telling air traffic control that it was a "remote control helicopter [with a] very small engine"

James Vincent | @jjvincent | Monday 27 October 2014 12:36 GMT | 4 comments

See and Avoid

In the case of manned aircraft (14 CFR Part 91 [FAA09]):

- Pilots may not operate so close to another aircraft as to create a collision hazard.
- Vigilance is maintained by pilots so as to see and avoid other aircraft.
- Pilots may not pass over, under, or ahead of another aircraft unless well clear.

Detect and Avoid

- Detect and Avoid (DAA) was defined by the FAA as "the combination of UAS Self-Separation (SS) plus Collision Avoidance (CA) as a means of compliance with 14 CFR Part 91".
- DAA Requirements:
  1. Provide a geometric means to determine well-clear status.
  2. Interoperate with existing collision avoidance systems.
  3. Avoid undue concern for traffic aircraft.
  4. Enable self-separation capabilities.

Requirement 1: Geometric Determination of Well Clear

- In the case of manned aircraft, well-clear status is a subjective determination by the on-board pilot.
- In the case of Unmanned Aircraft System (UAS), determination of well-clear status shall be made by an on-board system.
- Hence, an unambiguous mean of determining well-clear status is needed for UAS.

FAA DO-365 Definition of Well Clear

- A *loss of well-clear* occurs when:
  1. Distance at time of closest point of approach is less than `HMD` and either horizontal range is less than `DMOD` or modified tau is less than `TAUMOD`, and
  2. relative altitude is less than `ZTHR` or time to co-altitude is less than `TCOA`,
- where
  - Distance modification threshold (`DMOD`) = 4000 ft.
  - Horizontal miss distance threshold (`HMD`) = 0.66 nmi.
  - Modified τ threshold (`TAUMOD`) = 35 s.
  - Time to co-altitude threshold (`TCOA`) = 0 s.

A Formal Definition of Well Clear

Two aircraft are in well-clear violation if and only if $\texttt{WCV}(\mathbf{s}, \mathbf{v})$ holds.

$$\begin{aligned} \texttt{WCV}(\mathbf{s}, s_z, \mathbf{v}, v_z) \equiv\ & \texttt{Horizontal\_WCV}(\mathbf{s}, \mathbf{v})\ \texttt{and} \\ & \texttt{Vertical\_WCV}(s_z, v_z), \end{aligned} \tag{1}$$

where

$$\begin{aligned} \texttt{Horizontal\_WCV}(\mathbf{s}, \mathbf{v}) \equiv\ & d_{\text{cpa}}(\mathbf{s}, \mathbf{v}) \leq \texttt{HMD}\ \texttt{and} \\ & (\|\mathbf{s}\| \leq \texttt{DMOD}\ \texttt{or}\ 0 \leq \tau_{\text{mod}}(\mathbf{s}, \mathbf{v}) \leq \texttt{TAUMOD}), \\ d_{\text{cpa}}(\mathbf{s}, \mathbf{v}) \equiv\ & \|\mathbf{s} + t_{\text{cpa}}(\mathbf{s}, \mathbf{v})\, \mathbf{v}\|, \\ \texttt{Vertical\_WCV}(s_z, v_z) \equiv\ & |s_z| \leq \texttt{ZTHR}\ \texttt{or}\ 0 \leq t_{\text{coa}}(s_z, v_z) \leq \texttt{TCOA}, \\ t_{\text{coa}}(s_z, v_z) \equiv\ & -\frac{s_z}{v_z}. \end{aligned}$$

Requirement 2: Existing Collision Avoidance Systems

- Traffic Alert and Collision Avoidance System (TCAS) is a family of systems designed to reduce the risk of mid-air collision between *cooperative* aircraft (i.e., transponder equipped).
- TCAS is mandated in most commercial aircraft.
- TCAS II, the current generation of TCAS, provides: Traffic Alerts (TAs) and Resolution Advisories (RAs).

Interoperability with TCAS II RA

For an appropriate choice of threshold values, i.e., `DMOD`, `HMD`, `ZTHR`, `TAUMOD`, and `TCOA`, the well-clear volume contains the TCAS II RA volume.

Theorem 2 (Inclusion)

For all $(\mathbf{s}, s_z), (\mathbf{v}, v_z)$,

$$\mathit{TCAS\_II\_RA}(\mathbf{s}, s_z, \mathbf{v}, v_z) \implies \mathit{WCV}(\mathbf{s}, s_z, \mathbf{v}, v_z).$$

Requirement 3: Undue Concern for Intruder

In a pair-wise encounter, the ownship and intruder make the same determination of their well-clear status.

Theorem 3 (Symmetry)

For all $(\mathbf{s}, s_z), (\mathbf{v}, v_z)$,

$$\mathit{WCV}(\mathbf{s}, s_z, \mathbf{v}, v_z) \iff \mathit{WCV}(-\mathbf{s}, -s_z, -\mathbf{v}, -v_z).$$

More Theorems

1. Extensibility: A well-clear volume with a given set of thresholds is completely included in another well-clear volume with larger thresholds.
2. Local Convexity: In a pair-wise, non-maneuvering encounter, there is at most one time interval where the aircraft are predicted to be in well-clear violation.
3. Convergence: In a pair-wise, non-maneuvering converging encounter, the time interval of violation includes the time of closest point of approach

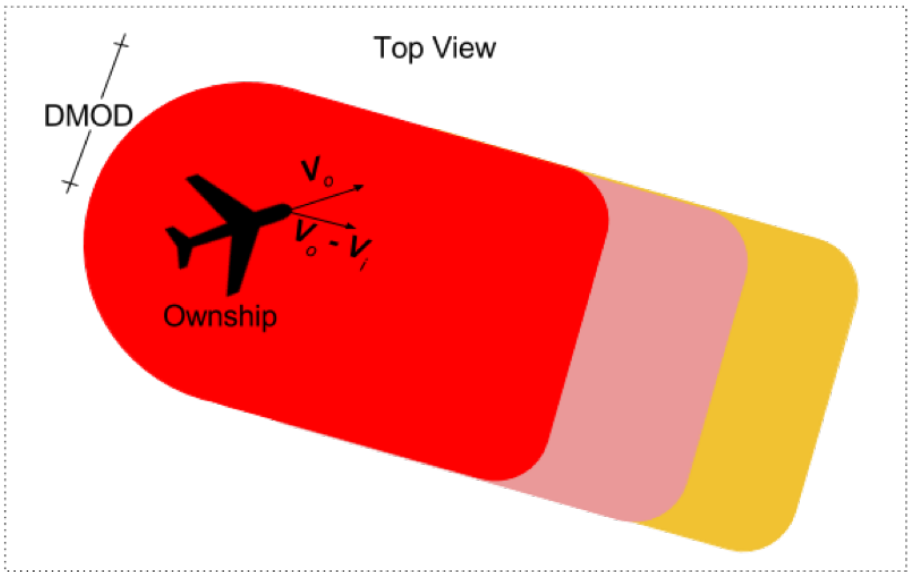
Requirement 4: Detect and Avoid Capabilities

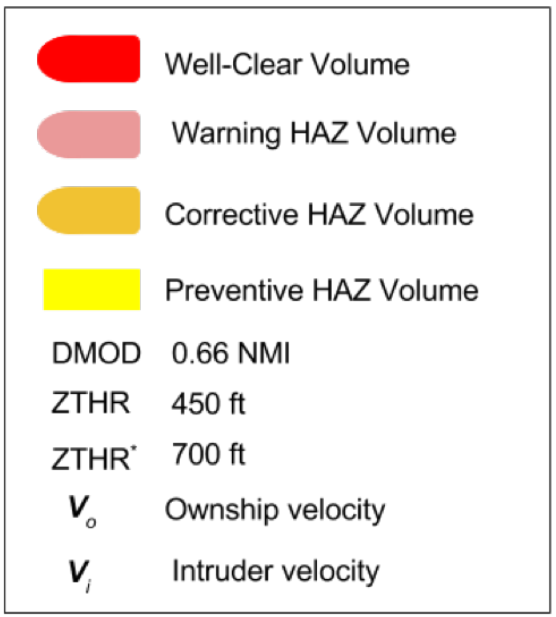
- Detection Logic: Predict whether ownship and intruder will lose well-clear within a lookahead time.
- Alerting Logic: Provide an alert level indicating severity of potential loss of well-clear status.
- Maneuver Guidance Logic: Provide guidance to maintain or recover well-clear status.

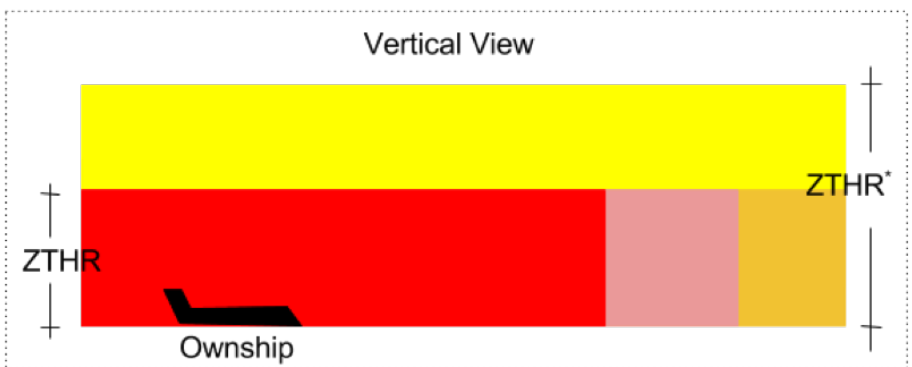
Alerting Logic

(Figure is notional)

Alerting logic is based on hazard volumes of increasing size, i.e.,







Operational Requirements for Alerting Logic

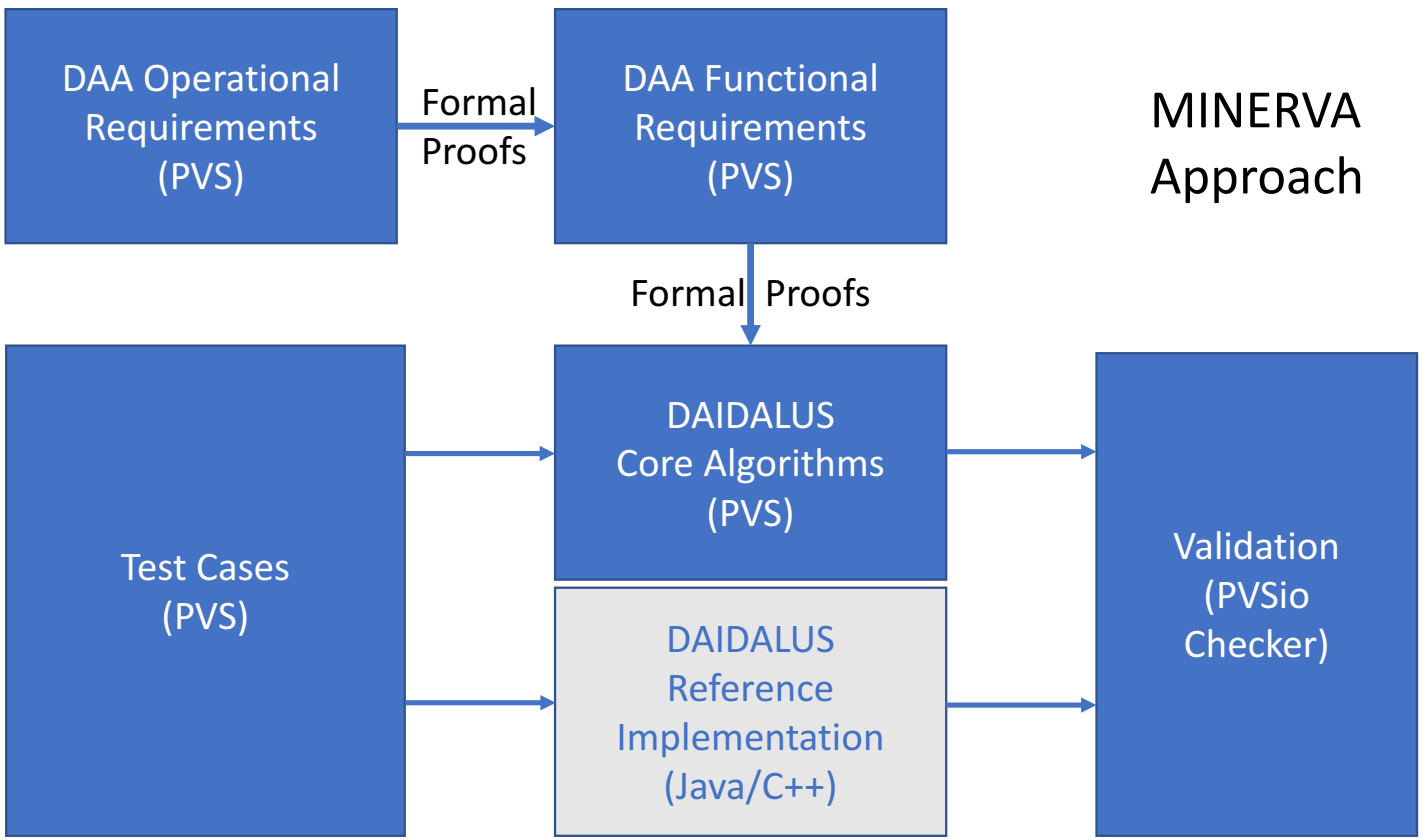
In a nominal, pair-wise, non-maneuvering encounter:

- Both aircraft simultaneously compute the same alert level
  $\Leftarrow$ *Well-clear symmetry*.
- Alerts progress according to severity level
  $\Leftarrow$ *Well-clear extensibility*.
- Once an alert is issued, it is continuously issued until threat disappears
  $\Leftarrow$ *Well-clear local convexity*.
- Once an alert is issued, it does not disappear before time of closest point of approach
  $\Leftarrow$ *Well-clear convergence*.

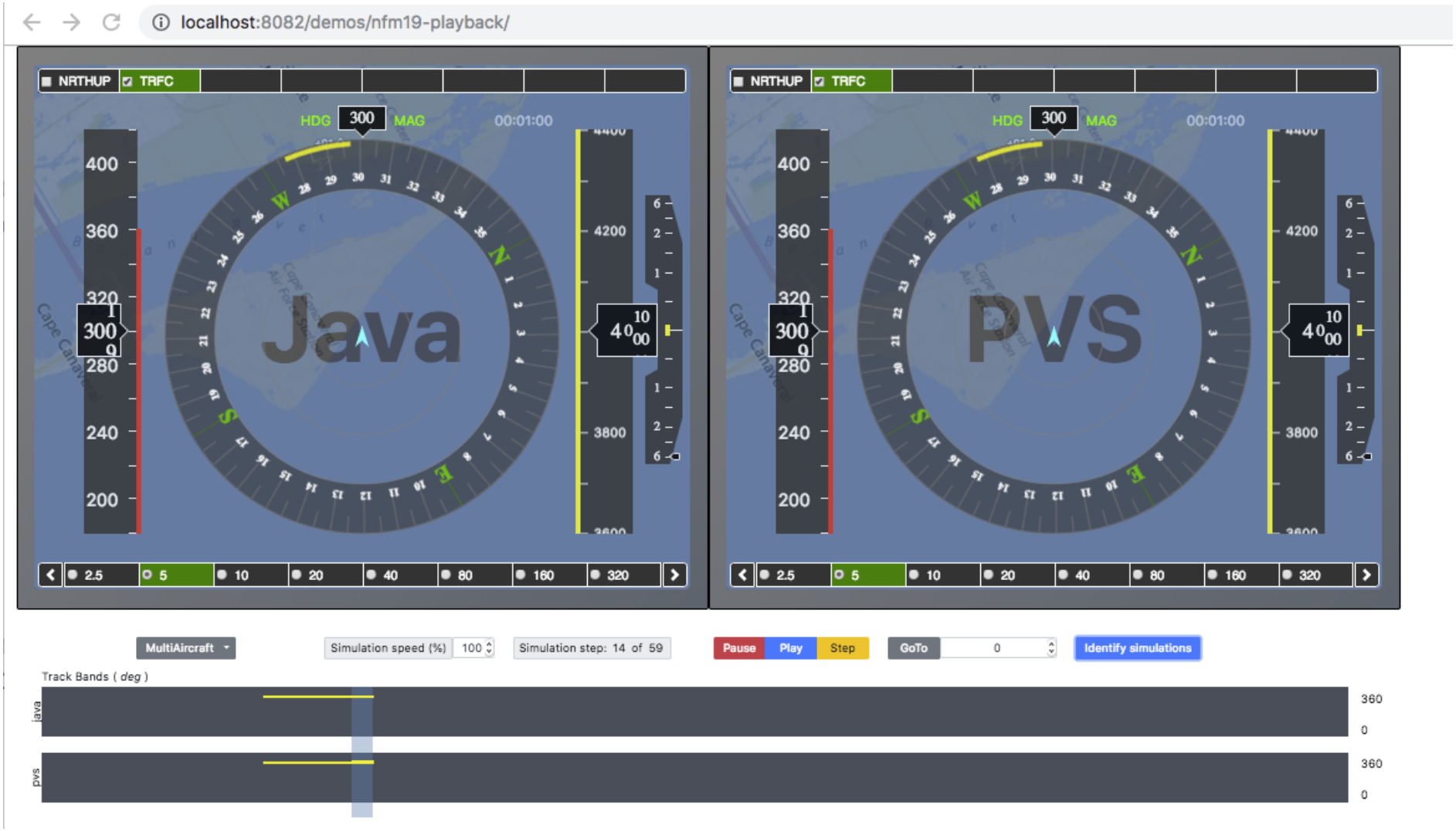
Detect and Avoid Alerting Logic for Unmanned Systems

- Reference implementation Minimum Operational Standard Requirements of Detect and Avoid for Unmanned Aircraft Systems (FAA DO-365).
- Specifications and software (Java and C++) released under NASA Open Source Agreement (`http://github.com/nasa/wellclear`).
- Further information: `https://shemesh.larc.nasa.gov/fm/DAIDALUS`.

Mirrored Implementation Numerically Evaluated against Rigorously Verified Algorithms (MINERVA)



Validation of DAIDALUS Using MINERVA



Concluding Remarks

- Rich specification language makes it possible to reason about a *family* of complex high-level properties and fine tune *specific* operational and functional requirements.
- Formal methods guarantee safe re-use of functional requirements.
- Formal methods enable powerful but practical model validation approaches, e.g., MINERVA.

No Silver Bullet

- Highly assured software is possible because of major initial development on theorem proving technology (non-linear arithmetic, model animation, etc.) and formal libraries (kinematics, conflict detection, and resolution, separation assurance, aircraft trajectories, etc.)
- Major advances are still needed on areas such as
  - Proof automation.
  - Readability and usability by non-experts.
  - Support for verification of non-deterministic behavior.

Final Thought[10]

"All models are wrong; the practical question is how wrong do they have to be to not be useful."

[10] *Empirical Model Building and Response Surfaces*, G. Box and N. Draper, 1987.

One Last Thing

```
class NumericalChallenge {

  static double a(int n) {
    if (n==0)
      return 11/2.0;
    if (n==1)
      return 61/11.0;
    return 111 - (1130 - 3000/a(n-2))/a(n-1);
  }

  public static void main(String[] argv) {
    for (int i=0;i<=20;i++)
      System.out.println("a("+i+") = "+a(i));
  }
}
```

Approximation of $a_0 \ldots a_{20}$

```
$ java NumericalChallenge
a(0) = 5.5
a(1) = 5.545454545454546
...
a(6) = 5.7491209211360 4
a(7) = 5.781810945409518
a(8) = 5.81131466923334
...
a(12) = 5.935956716634138
a(13) = 6.534421641135182
a(14) = 15.413043180845833
a(15) = 67.47239836474625
a(16) = 97.13715118465481
a(17) = 99.82469414672073
a(18) = 99.98953968869486
a(19) = 99.9993761416421
a(20) = 99.9999627595651 1
```

Why We Should Not Depend Upon Software

Theorem 4

- *For all* n, $a(n) = \frac{6^{n+1}+5^{n+1}}{6^n+5^n}$.
- $\lim_{n\to\infty} a^n = 6$.

$$a(20) \approx 6$$

Acknowledgements

- Formal Methods Team: Swee Balachandran (NIA), Ricky Butler (NASA), Aaron Dutle (NASA), Marco Feliú (NIA), Alwyn Goodloe (NASA), George Hagen (NASA), Paolo Masci (NIA), Mariano Moscato (NIA), César Muñoz (NASA), Ivan Perez (NIA), Laura Titolo (NIA),
- Former members: Heber Herencia, Anthony Narkawicz, Radu Siminiceanu, Andrew Smith, Jason Upchurch.

References I

Devesh Bhatt, Anitha Murugesean, Brendan Hall, Hao Ren, and Yoganada Jeppu.
The CLEAR way to transparent formal methods.
In *Proceedings of the 4th Workshop on Formal Integrated Development Environment (F-IDE 2018)*, Oxford, UK, July 2018.

Aaron Dutle, Mariano Moscato, Laura Titolo, César Muñoz, Gregory Anderson, and François Bobot.
Formal analysis of the Compact Position Reporting algorithm.
Formal Aspects of Computing, 2020.

References II


Aaron Dutle, Mariano Moscato, Laura Titolo, and César Muñoz.
A formal analysis of the Compact Position Reporting algorithm.
In Andrei Paskevich and Thomas Wies, editors, *Proceedings of the 9th International Conference on Verified Software: Theories, Tools, and Experiments (VSTTE 2017)*, volume 10712 of *Lecture Notes in Computer Science*, pages 19–34, Heidelberg, Germany, July 2017. Springer.


FAA Sponsored Sense and Avoid Workshop.
Sense and avoid (SAA) for Unmanned Aircraft Systems (UAS), October 2009.


Federal Aviation Administration.
Requirements engineering management findings report.
Final Report DOT/FAA/AR-08/34, Air Traffic Organization NextGen & Operations Planning Office of Research and Technology Development, Washington, DC 20591, May 2009.

References III


Peter H. Feiler, John B. Goodenough, Arie Gurfinkel, Charles B. Weinstock, and Lutz Wrage.
Reliability validation and improvement framework.
Special Report CMU/SEI-2012-SR-013, Software Engineering Institute, November 2012.


Kerianne H. Gross, Aaron W. Fifarek, and Jonathan A. Hoffman.
Incremental formal methods based design approach demonstrated on a coupled tanks control system.
In *2016 IEEE 17th International Symposium on High Assurance Systems Engineering (HASE)*, pages 181–188, Jan 2016.

References IV


RTCA SC-186.
Minimum Operational Performance Standards for 1090 MHz extended squitter Automatic Dependent Surveillance - Broadcast (ADS-B) and Traffic Information Services - Broadcast (TIS-B).
2009.

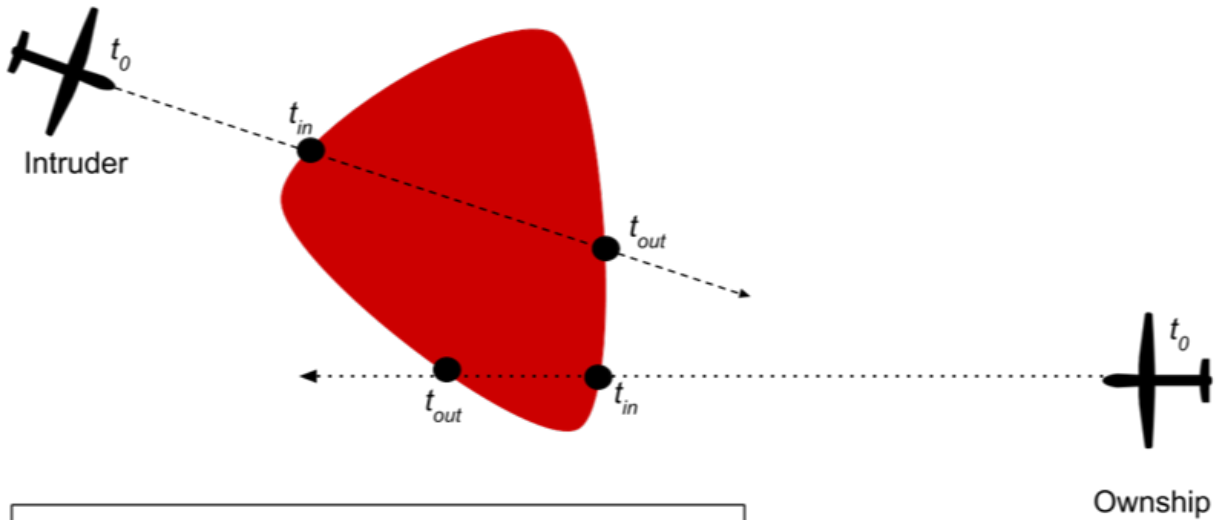

Kit Siu, Abha Moitra, Michael Durling, Andy Crapo, Meng Li, Han Yu, Heber Herencia-Zapana, Mauricio Castillo-Effen, Shiraj Sen, Craig McMillan, Daniel Russell, Sundeep Roy, and Panagiotis Manolios.
Flight critical software and systems development using ASSERT.
In *2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)*, number 978-1-5386-0365-9/17, pages 1–10, Sept 2017.

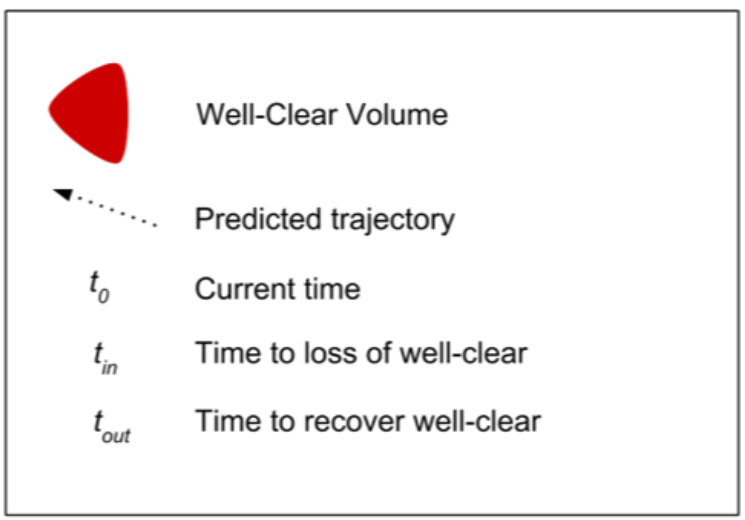
Backup Slides

Detection Logic

(Figure is notional)

Assuming non-maneuvering trajectories, compute time interval of loss of well-clear, i.e.,

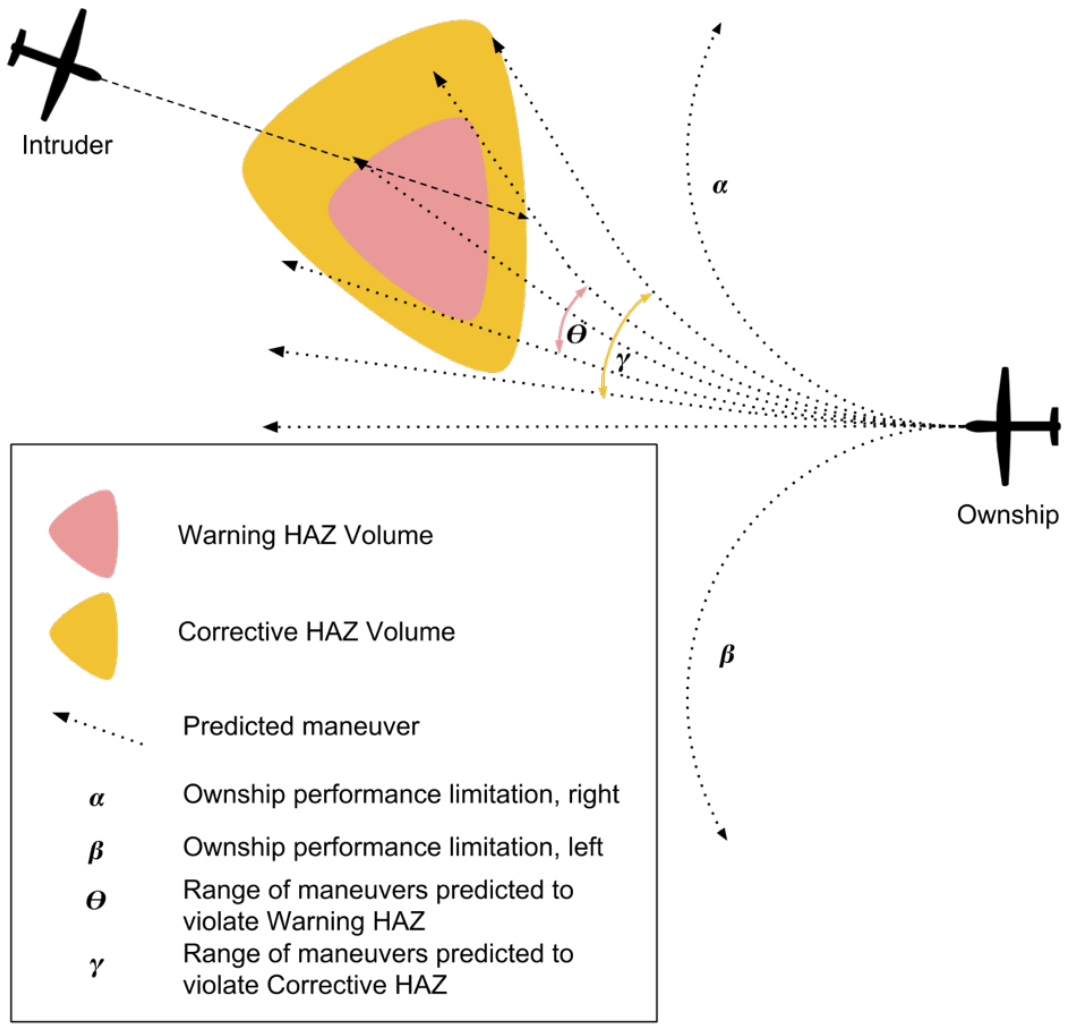




Maneuver Guidance Logic

(Figure is notional)

Compute range of maneuvers to maintain/recover from loss of well clear, i.e.,



Functional Specification of Detection Logic

The following algorithm returns the time interval of loss of well-clear within a lookahead time T.

$$
\begin{aligned}
&\texttt{detection\_WCV}(\mathbf{s}, s_z, \mathbf{v}, v_z, T) \equiv \\
&\quad \texttt{let } [t_1, t_2] = \texttt{detection\_VWCV}(s_z, v_z, T) \texttt{ in} \\
&\qquad \texttt{if } t_1 > t_2 \texttt{ then } [T, 0] \\
&\qquad \texttt{elsif } t_1 = t_2 \texttt{ and } \texttt{Horizontal\_WCV}(\mathbf{s} + t_1\mathbf{v}, \mathbf{v}) \texttt{ then } [t_1, t_1] \\
&\qquad \texttt{elsif } t_1 = t_2 \texttt{ then } [T, 0] \\
&\qquad \texttt{else let } [t_{\text{in}}, t_{\text{out}}] = \texttt{detection\_HWCV}(\mathbf{s} + t_1\mathbf{v}, \mathbf{v}, t_2 - t_1) \texttt{ in} \\
&\qquad\quad [t_{\text{in}} + t_1, t_{\text{out}} + t_1] \\
&\qquad \texttt{endif},
\end{aligned}
$$

where

$$
\begin{aligned}
\texttt{detection\_VWCV}(s_z, v_z, T) &\equiv \ldots \\
\texttt{detection\_HWCV}(\mathbf{s} + t_1\mathbf{v}, \mathbf{v}, t_2 - t_1) &\equiv \ldots
\end{aligned}
$$

Theorem 5 (Soundness and Completeness)

For all $(\mathbf{s}, s_z), (\mathbf{v}, v_z)$, $T > 0$, *and* $t \in [0, T]$,

$$\mathit{WCV}(\mathbf{s} + t\mathbf{v}, s_z + tv_z, \mathbf{v}, v_z) \iff$$
$$t \in \mathit{detection\_WCV}(\mathbf{s}, s_z, \mathbf{v}, v_z, T).$$

Independent Configurable Architecture for Reliable Operations of Unmanned Systems

- ICAROUS is an on-board software architecture for developing autonomous UAS applications.
- It integrates formally verified algorithms that enforce containment constraints and monitors distance/time boundaries to prevent violations, e.g.,
  - DAIDALUS: Detect and Avoid.
  - PolyCARP: Geofence containment.
- ICAROUS provides services such as path planning, stand-off distance and path conformance, tracking, merging and spacing.

ICAROUS

- Released under NASA Open Source Agreement (`http://github.com/nasa/icarous`).
- Further information: `https://shemesh.larc.nasa.gov/fm/ICAROUS`.

ICAROUS Distributed Architecture

Modular, highly configurable, and mission independent architecture that interoperates with NASA's UTM Services, AFRL's OpenAMASE, NASA's Plan Execution Interchange Language (PLEXIL), NASA's Core Flight Systems (cFS), MavLink, Ardupilot, etc.