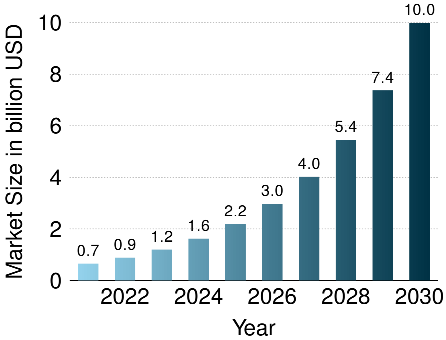
ADVANCING THE AI-BASED REALIZATION OF ACAS X TOWARDS REAL-WORLD APPLICATION

The 36th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)

DLR

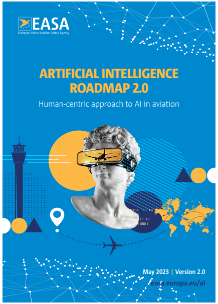
Motivation

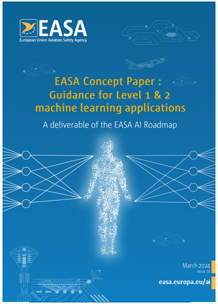
- AI already vital in many domains
- AI in aviation to reach $10 billion by 2030, CAGR >35 %
- Safety in other domains often tread lightly
- AI-based systems require unmet levels of safety



- AI will severely impact future aviation
- Safety is paramount

EASA Roadmap for Safe AI in Aviation





Learning Assurance

“All [. . .] actions [. . .] that error[s] [. . .] have been identified and corrected such that the AI/ML constituent [. . .] provides sufficient generalisation and robustness capabilities.”

- EASA AI Roadmap and Concept Papers
- Way towards safe Artificial Intelligence in aviation
- Emphasize a clear and transparent approach

Operational Design Domain

- Developed by SAE International
- Designed for autonomous systems
- Clearly defines environmental conditions
- Enforces boundaries of operation
- Required by EASA for all AI applications

OPERATIONAL DESIGN DOMAIN

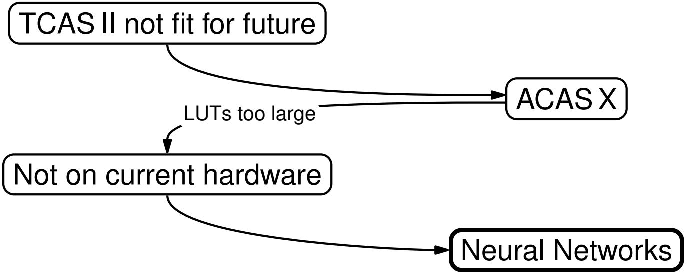
- Scenery
  - Geography = Above land
- Dynamic Elements
  - Intruder
    - . . .
- Environmental Conditions
  - Wind = 0 kn to 40 kn

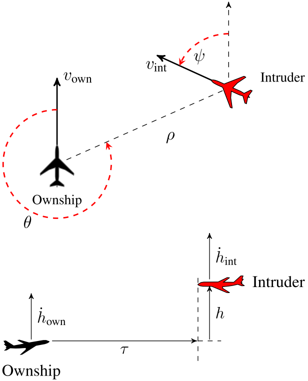
"Operating conditions under which a given driving automation system [. . .] is specifically designed to function, including [. . .] **environmental**, **geographical**, and time-of-day restrictions, and [. . .] **traffic** or **roadway** characteristics."

Collision Avoidance

- Collision Avoidance is crucial for safety
- TCAS II is the current standard

Problem **Solution**





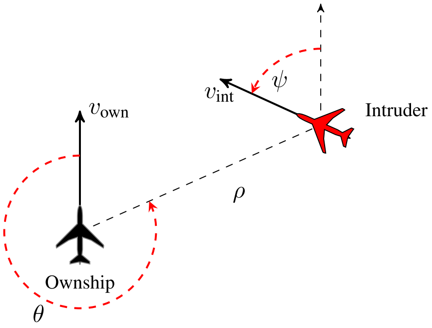
Images based on [8]

[8] Kyle D. Julian and Mykel J. Kochenderfer. “Guaranteeing Safety for Neural Network-Based Aircraft Collision Avoidance Systems”. In: *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*. IEEE, Sept. 2019

HCAS: A Horizontal Collision Avoidance System

Variable	Unit	Description
ρ	ft	Distance to intruder
θ	°	Bearing angle to intruder
ψ	°	Relative heading angle
v_{own}	$ft\,s^{-1}$	Ownship's true airspeed
v_{int}	$ft\,s^{-1}$	Intruder's true airspeed
τ	s	Time to closest point of approach
s_{adv}	–	Previous advisory

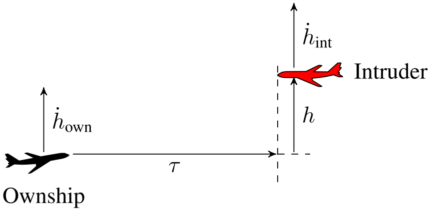
Advisory	Description
COC	clear of conflict
WL	weak left
WR	weak right
SL	strong left
SR	strong right



VCAS: A Vertical Collision Avoidance System

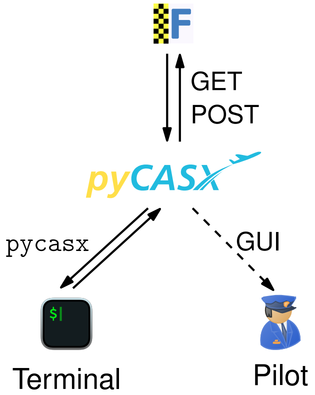
Variable	Unit	Description
h	ft	Altitude difference
$\dot{h}_{own}$	ft s^{-1}	Ownship's vertical rate
$\dot{h}_{int}$	ft s^{-1}	Intruder's vertical rate
τ	s	Time to closest point of approach
s_{adv}	–	Previous advisory

Advisory	Description
COC	clear of conflict
DNC	do not climb
DND	do not descend
DES1500	descend $\geq$ 1500 ft/min
CL1500	climb $\geq$ 1500 ft/min
SDES1500	strengthen descend to $\geq$ 1500 ft/min
SCL1500	strengthen climb to $\geq$ 1500 ft/min
SDES2500	strengthen descend to $\geq$ 2500 ft/min
SCL2500	strengthen climb to $\geq$ 2500 ft/min



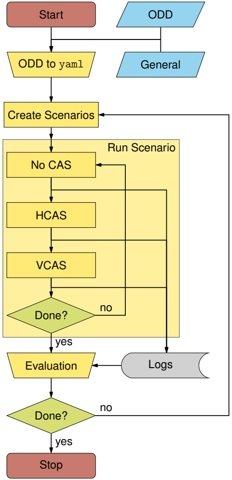
pyCASX – An Automated Testing Tool

- Open-source tool
- Designed to test Collision Avoidance Systems
- Focused on FlightGear
- Provides a suite of CLI tools
- Fully customizable via `Hydra`
- Easy to use REST interface

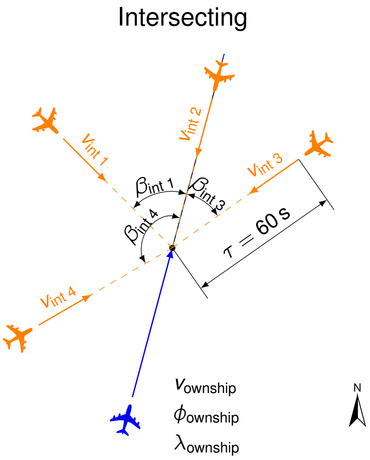


pyCASX – Process Flowchart

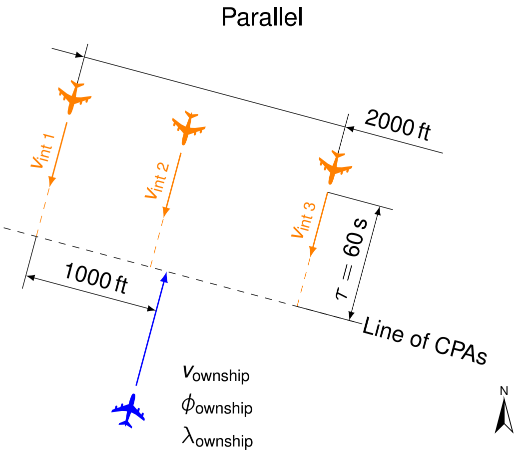
1. Convert the ODD to a `yaml` file
2. Create scenarios: `pycasx scenarios`
3. Copy into the correct folder: `pycasx copy`
4. Run scenarios in a loop: `pycasx run`
5. Evaluate the results



Scenarios

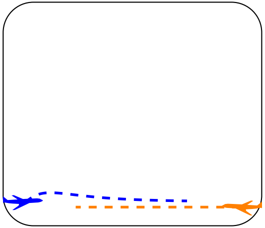


Intersecting
$v_{int\ 1}$
$v_{int\ 2}$
$v_{int\ 3}$
$v_{int\ 4}$
$\beta_{int\ 1}$
$\beta_{int\ 3}$
$\beta_{int\ 4}$
$\tau = 60\,s$
$v_{ownship}$
$\phi_{ownship}$
$\lambda_{ownship}$
N

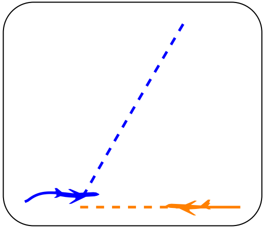


Parallel
2000 ft
$v_{int\ 1}$
$v_{int\ 2}$
$v_{int\ 3}$
1000 ft
$\tau = 60\,s$
Line of CPAs
$v_{ownship}$
$\phi_{ownship}$
$\lambda_{ownship}$
N

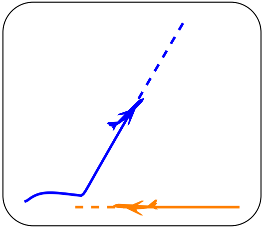
Evolution of a VCAS scenario



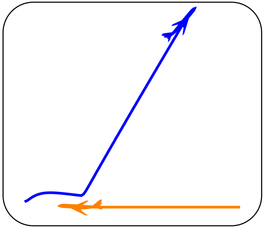
(a) Scenario starts at $t = 0\,\text{s}$.



(b) *CL1500* issued at $t = 30\,\text{s}$.

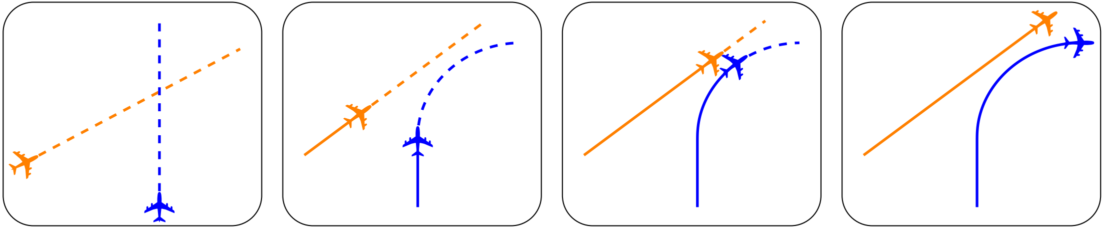


(c) CPA at $t = 57\,\text{s}$ with $h = -732\,\text{m}$.



(d) Scenario end at $t = 90\,\text{s}$.

Evolution of an HCAS scenario



(a) Scenario starts at $t = 0\,\mathrm{s}$.

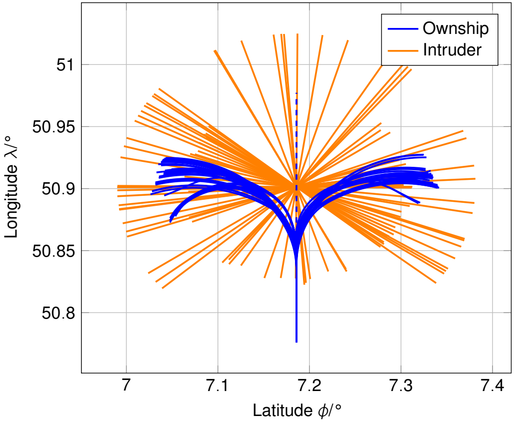
(b) *WR* issued at $t = 26\,\mathrm{s}$.

(c) CPA at $t = 60\,\mathrm{s}$ with $\rho = 7587\,\mathrm{m}$.

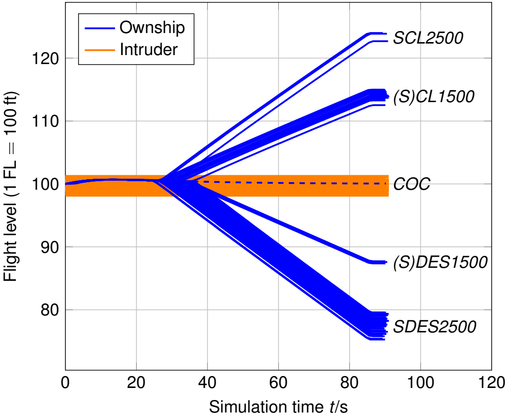
(d) Scenario ends at $t = 90\,\mathrm{s}$.

Results

HCAS



VCAS



Summary and Outlook

- Testing ODDs vital for safe AI
- `pyCASX` automatically tests defined ODDs
- Help with AI Engineering for Collision Avoidance
- Define more types of scenarios
- Test with CAS enabled on both aircraft
- pypi v1.0.0 | python 3.8 | 3.9 | 3.10 | 3.11 | pre-commit.ci passed | Python tests (pytest) passing | docs passing | REUSE compliant

- AI Engineering requires specific tools
- Automation is crucial for ODD testing

pyCASX

Contact

Johann Christensen
German Aerospace Center (DLR)
Institute for AI Safety and Security
Rathausallee 12 | 53757 Sankt Augustin | Germany
johann.christensen@dlr.de

References

[1] Precedence Research. *Artificial Intelligence in Aviation Market Size, Share, and Trends 2024 to 2034*. Research rep. 1748. Precedence Research, May 2022

[2] European Union Aviation Safety Agency (EASA). *Artificial Intelligence Roadmap 2.0*. Tech. rep. Version 2.0. Postfach 10 12 53, 50452 Cologne, Germany: European Union Aviation Safety Agency (EASA), May 2023

[3] European Union Aviation Safety Agency (EASA). *EASA Concept Paper: Guidance for Level 1 & 2 Machine Learning Applications*. Tech. rep. Version Issue 02. Postfach 10 12 53, 50452 Cologne, Germany: European Union Aviation Safety Agency (EASA), Apr. 19, 2024

[4] RTCA, Inc. *DO-185B*. Tech. rep. Washington, DC, USA: GlobalSpec, June 19, 2008

[5] RTCA, Inc. *DO-385*. Tech. rep. Washington, DC, USA: GlobalSpec, Oct. 2, 2018

[6] RTCA, Inc. *DO-386 Volume 1 & 2*. Tech. rep. Washington, DC, USA: GlobalSpec, Dec. 17, 2020

[7] Kyle D. Julian et al. “Policy compression for aircraft collision avoidance systems”. In: *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*. IEEE, Sept. 2016

[8] Kyle D. Julian and Mykel J. Kochenderfer. “Guaranteeing Safety for Neural Network-Based Aircraft Collision Avoidance Systems”. In: *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*. IEEE, Sept. 2019

[9] FlightGear developers & contributors. *FlightGear*. Version 2020.3.19. Oct. 18, 2023

[10] Omry Yadan. *Hydra - A framework for elegantly configuring complex applications*. GitHub. 2019

Imprint

Topic:	**Advancing the AI-Based Realization of ACAS X Towards Real-World Application**
Date:	October 28, 2024 to October 30, 2024
Author:	Johann Christensen
Institute:	Institute for AI Safety and Security
Credits:	All images “DLR (CC BY-NC-ND 3.0)” unless otherwise stated