“The Reliability and Vulnerability of Autonomous Systems”

Dr Darren Ansell C.Eng
UCLAN Space and Aerospace Engineering Lead

A Presentation at informal Meeting of Experts on lethal autonomous weapons systems (LAWS) – CCW
14 April 2015, United Nations, Geneva, Switzerland

uclan
University of Central Lancashire

Introduction

- How does autonomy affect the reliability and vulnerability of a weapon system?

- What type of resilience to programming and deployment errors can be built into autonomous systems or the regulations governing their use?

Questions to ask….

- What is the consequence of the autonomous system not operating when required?
- What is the consequence of the autonomous system operating when not required?
- What is the consequence of the autonomous system operating incorrectly?

Reliability

Adapted from DO-178C, Software Considerations in Airborne Systems and Equipment Certification:

Level	Aircraft	LAWS
A-Catastrophic	Failure may cause multiple fatalities, usually with loss of the airplane.	Failure may cause multiple unintended fatalities
B - Hazardous	Failure has a large negative impact on safety or performance, or reduces the ability of the crew to operate the aircraft due to physical distress or a higher workload, or causes serious or fatal injuries among the passengers.	Failure may cause serious or fatal injuries
C-Major	Failure significantly reduces the safety margin or significantly increases crew workload. May result in passenger discomfort (or even minor injuries).	Failure may cause minor injuries
D-Minor	Failure slightly reduces the safety margin or slightly increases crew workload. Examples might include causing passenger inconvenience or a routine flight plan change.	Failure increases operator/commander workload
E-No Effect	Failure has no impact on safety, aircraft operation, or crew workload.	Failure has no impact on safety

Determinism of AS Software

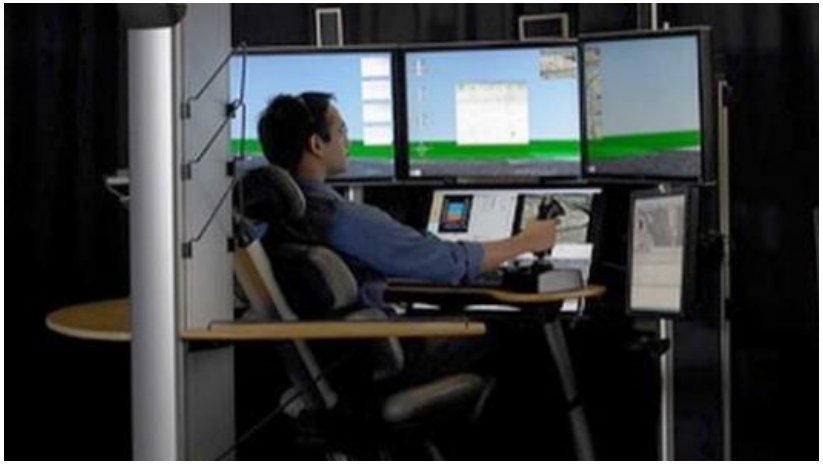
A **deterministic algorithm** is an algorithm which, given a particular input, will always produce the same output, with the underlying machine always passing through the same sequence of states.

A **nondeterministic algorithm** is an algorithm that, even for the same input, can exhibit different behaviours on different runs

Autonomy software interacts with a dynamic environment in a non-deterministic manner. Many AS algorithms are non-deterministic and this should limit their application in LAWS.

Testing of Autonomous Systems

Virtual (Synthetic) Environments



Good for systems integration testing, but fidelity of simulation and models varies. Often not validated.

Testing of Autonomous Systems

Active research areas:

e.g. 'Formal Methods'

"Mathematically based techniques for the specification, development and verification of software and hardware systems."

e.g. 'Model Checking'

Model checking is a technique for automatically verifying correctness properties of finite-state systems. Difficult for autonomous systems which can been seen as near infinite state machines.

In general, we verify the software always tries to achieve its goals/targets to the best of its knowledge/beliefs/ability. That the software aims for situations it believes to be good and avoids situations it believes to be bad.

Consequently, any guarantees made are about the autonomous system's decisions given certain inputs, not about its overall effects.

Resilience to Programming Errors

- False or missing software requirements
- Incorrect algorithms or code
- Inadequate testing
- Incorrect or unexpected usage of the autonomous systems software

A good approach is to adhere to DO-178C! Use certificated toolkits for deployment.

Hacking and Cyber-Attacks

- **Denial of service/taking over C2 of the actual platforms.**

Current UxV requirements deal with traditional information assurance aspects and not defence against offensive cyber attacks.

- **Use of commercial off-the-shelf (COTS) and open source products in control stations.**

The dependence on commercial hardware (processors, etc.) also exposes the system to the cyber vulnerabilities of the global supply chain.

- **Increasing desire to network platforms and control station locations.**

- **The ability to inexpensively deny GPS to ground and low-flying air systems is a well known threat.**

- **Signatures and Anti-Tamper**

Current platforms have little or no provisions for anti-tamper.

Understanding Vulnerabilities:Types of Cyber-Attacks

- Flaws

A flaw is unintended functionality. Common to AS!

- Features

Intended functionality which can be misused by an attacker to breach a system

- User Error

Failure to follow procedures, exposing system to risk

Advanced System Health Management & Cyber Resilience

- **Onboard systems** to detect and mitigate effects of a sub-system or component failure:

  E.g. Sensor readings are abnormal – ignore output and use alternatives, or device has failed, switch in alternative.
- **Onboard systems** to detect and mitigate a cyber attack:
  - E.g. Monitoring running processes on a device, isolation of infected devices. Looking for the symptoms elsewhere in the system. Similar to anti-virus.
- **Offboard systems** to monitor behaviour of the system

Summary

- Difficulty in thoroughly checking the behavior of AS software leaves it exposed to reliability and vulnerability problems.
- Adopt DO-178C style requirements in the regulatory requirements for the engineering of LAWS (not just airborne systems).
- Minimise use of COTS in Level A autonomous systems – difficult to certificate as manufacturers are reluctant to disclose design information.
- Use certificated toolkits for deployment of autonomous software.
- Challenge designs that do not have on and off-board system behavior monitoring.

Thank You!

dansell@uclan.ac.uk

INNOVATIVE THINKING
FOR THE REAL WORLD

uclan
University of Central Lancashire