Carnegie Mellon University
Software Engineering Institute

Assuring Cyber-Physical Systems – Scalable Formal Verification

January 2022

Dionisio (Dio) de Niz
dionisio@sei.cmu.edu

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM22-0079

ACPS Initiatives

Model-Based Engineering

- Practical formalized record of Intent
  - "Ambiguities resolved to permissible interpretations"[1]
  - Standard AADL
    - Practitioner, Research
- Analysis Automation
  - Theory to practice
  - Open source tool
- Integrating Code Generation
- Early Error Discovery

Formal Verification of CPS

- New Analysis for New Features
  - Multicore, Autonomy
- Scientific Expertise in CPS Domains
  - Timing, Control, Logic
- Inter-Analyses Verification
- Inter-Domain Optimization
  - Control + Scheduling
  - Scheduling + Logical Verification
- Scalable Analyses
  - Runtime Verification
  - Mixed-Trust Scheduling

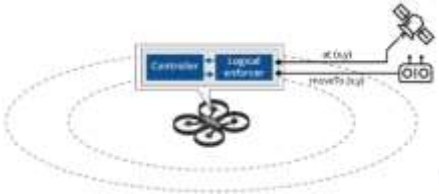
Scalable Assurance

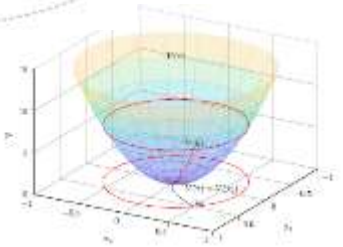
Kinetic effect of Cyber-Physical Systems
- Safety critical
- Requires strong assurance
  - Logic (value)
  - Timing (before crash)
  - Correct physical reaction

Strong assurance not possible at practical scale
- Multi-Criticality:
  - But not required for everything
- Artifact size
  - Too large for strong verification techniques

Cognitive Design Overload (large systems)
- Top-level requirements -> refinement cycles -> implementation
- Assurance at all levels of refinement

Rapid Certifiable Trust





Minimize what is verified

- Enforcer to prevent unverified code misbehavior
- Verify enforcer: physics, timing, logic
- Protect enforcers

Verified Physical Effects

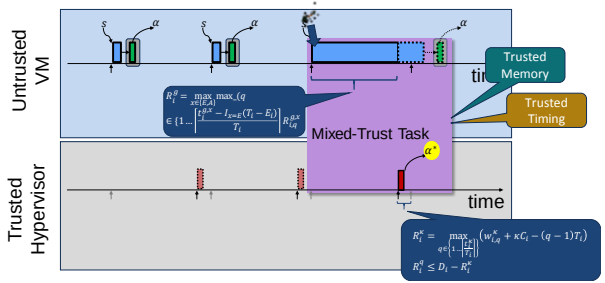
- Recoverable Set: $\varepsilon_{SC^j}(1)$ Safety Set: $\varepsilon_{SC^j}(\epsilon_s) \triangleq \epsilon_s\, \varepsilon_{SC^j}(1)$
  - Controlled System: $\dot{x} = f_\varphi(x) \triangleq f(x, \varphi(x))$
  - Lyapunov Function:

$$V_\varphi\colon \mathbb{R}^n \to \mathbb{R}, \mathcal{N}_{V_\varphi}(x_{eq}) \subseteq \mathcal{N}_\varphi(x_{eq}), V_\varphi(x_{eq}) = 0, \forall x \in \mathcal{N}_{V_\varphi}(x_{eq}) - \{x_{eq}\}\colon (i)\ \mathrm{V}_\varphi(\mathrm{x}) > 0,$$

$$\dot{V}_\varphi(x) = \frac{\partial V}{\partial x} \cdot f_\varphi(x) < 0, \quad \text{level set: } \varepsilon_\varphi(\epsilon) = \left\{ x \in \mathcal{N}_{V_\varphi}(x_{eq}) \,\middle|\, V_\varphi(x) \le \epsilon \right\}, \epsilon \le 1$$

Verified Timing

- Timing enforcer prevents late output (after crash)
- Mixed-trust task: unverified in VM + verified in HV
- Schedulability equations combined schedulers (HV+VM)



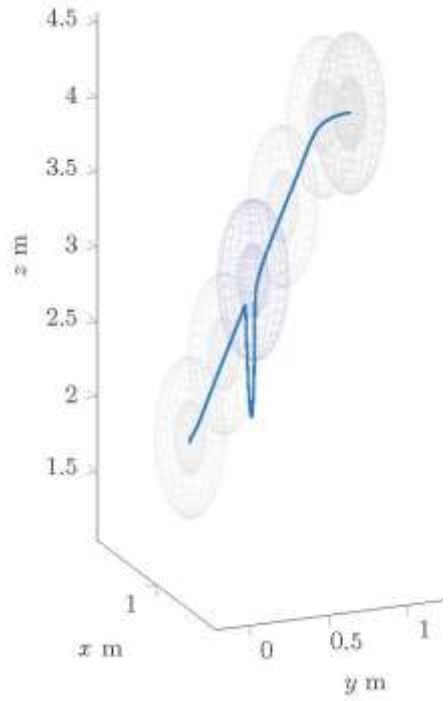
Verified Logic

- Verified Hypervisor
- Verified enforcement composition logic

Analysis of Mission Progress Enforcing Unsafe Behavior

6 DOF $\Rightarrow$ 12 state variables

$$\ddot{p}_x = -\cos\phi \sin\theta \frac{F}{m}$$
$$\ddot{p}_y = \sin\phi \frac{F}{m}$$
$$\ddot{p}_z = g - \cos\phi\cos\theta \frac{F}{m}$$
$$\ddot{\phi} = \frac{1}{J_x}\tau_\phi$$
$$\ddot{\theta} = \frac{1}{J_y}\tau_\theta$$
$$\ddot{\psi} = \frac{1}{J_z}\tau_\psi.$$



Linear design:

- linearize at equilibrium
- assume full state available
- LQ state feedback design
- reference points = equilibrium states

Drone Experiment