High-Assurance Runtime Verification

Toward a New Era of Flight
The committee did not individually prioritize these barriers. However, there is one critical, crosscutting challenge that must be overcome to unleash the full potential of advanced increasingly autonomous (IA) systems in civil aviation. This challenge may be described in terms of the question "How can we assure that advanced IA systems—especially those systems that rely on adaptive/nondeterministic software—will enhance rather than diminish the safety and reliability of the NAS?" There are four particularly challenging barriers that stand in the way of meeting this key challenge:

- Certification process
- Verification and validation
- Decision making by adaptive/nondeterministic systems
- Trust in adaptive/nondeterministic IA systems

V&V Challenges of IA Systems
The sophisticated algorithmic methods that enable increasingly autonomous (IA) systems do so by sacrificing predictability
- Machine learning (ML)
- Advanced planning algorithms

Traditional formal verification proves that a program text satisfies a specification
- ML programs evolve as they learn
- How to verify a moving target?
- What is the formal specification of an ML program?

Runtime Verification
- Roots in the monitoring system logs (1970s) and Simplex architecture (1990s)
- Runtime verification (RV) is a form of dynamic verification
- Given a specification $\phi$
- And a trace $\tau$ obtained from a system under observation (SUA)
- An RV monitor checks for language inclusion $\tau \in \mathcal{L}(\phi)$
  - Accept all traces admitting $\phi$
- RV Frameworks generate monitors from a specification
- Specification logics: linear temporal logics (LTL), regular expressions, etc.

High-Assurance RV
- Can RV safeguard a system that cannot be otherwise assured?
- Can we maintain safety and predictability through RV even if using ML?
  - For high-confidence applications, RV mechanisms must be assured to the same level as required for conventional safety-critical systems
- What evidence will we need in a safety case?
- NASA Langley has been investigating these questions
- Copilot RV Framework

RV on AI-Based Autonomous Systems
- If RV is to be used to ensure the safe deployment of AI-based autonomous systems, the monitor specifications must be consistent with human notions of safety
- How do we formalize commonsense and acceptable behavior
  - Do not behave erratically
  - Drive in a manner expected by other drivers
  - “I’m sorry Dave, I’m afraid I can’t do that”
- How do we specify when an objective function has gone too far?

Many Challenges Remain
- Existing automated verification tools are not well suited to analyze the complex math found in monitor specifications for aerospace applications
  - Floating point errors are a particular concern for monitors
- Composing assured RV with the SUO
  - Ideally monitors should only “watch”, but in reality compete for resources
- Noninterference theorems for RV
- Generating secure monitors from high-assurance RV frameworks
- Verifying the correctness and safety of the steering performed from any viable system state once a specification violation is detected
- Performing RV on vehicle swarms
- Applying RV to ML based autonomy

Some Challenges We have Addressed
- Getting the monitors right
  - Domain experts validate
  - Proofs of correctness with SMT solvers and model checkers
- Traceability from specification to monitor code
- Finding bugs in monitors
  - Static and dynamic analysis tools
- Proofs that monitor code generated by the Copilot framework satisfies the monitor specification
  - Frama-C WP tool