28th Annual INCOSE international symposium

Washington, DC, USA

July 7 - 12, 2018

Extending Formal Modeling for Resilient Systems Design

Azad M. Madni, Michael Sievers, Dan Erwin, Ayesha Madni, Edwin Ordoukhanian, Parisa Pouya
University of Southern California

Problem

- Systems and networks in the 21st century are required to be resilient in the face of uncertainty and systemic and external disruptions
- Predictability, flexibility and adaptability are essential for verifiable, resilient behavior of systems and system-of-systems networks
- For predictable system operation, system (model) has to be **verifiable** in terms of both static properties and dynamic behavior
- For flexibility, system (model) needs to be **modifiable** by an external agent
- For adaptability, system (model) needs to have the ability to **self-adjust** (i.e., self-restructure, self-reorganize, self-reconstitute)
- These requirements lead to the need for formal and probabilistic modeling to address tradeoffs between system (model) verifiability, flexibility and adaptability
- This recognition provided the impetus for our research

Research Overview

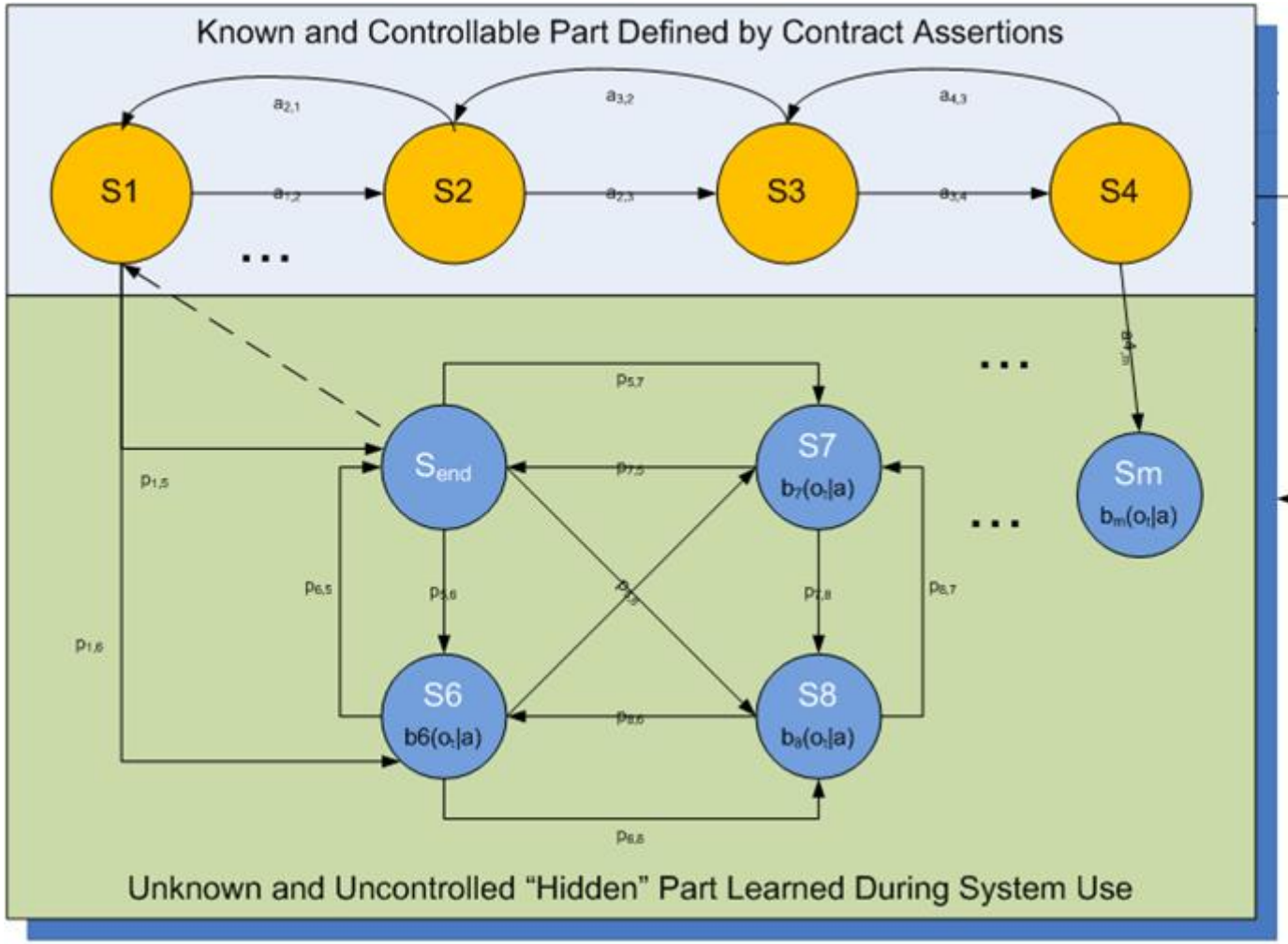
- Objective
  - develop a formal modeling approach for designing resilient systems
- Approach
  - based on Resilience Contract (RC), a formal, probabilistic construct
  - RC = Traditional Contract + flexible assumptions + Partially Observable Markov Decision Process + in-use learning
- Application
  - planning and decision making in multi-UAV swarm and spacecraft swarm
  - problem of interest to both DOD and civilian sector
- Sponsor
  - DOD Systems Engineering Research Center (SERC)

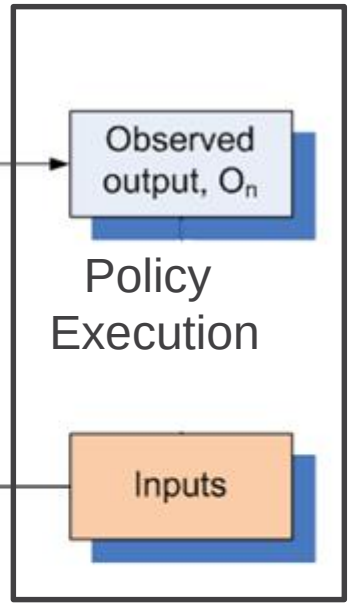
Resilience Contract

Resilience Contract

- A hybrid modeling construct for stochastic/probabilistic systems
  - partial observability, noisy sensors, uncertain environment
- Incorporates flexible assertions to allow for **uncertainty** in the knowledge of system state and state of the environment
  - flexible assertions: relax “assume-guarantee” in traditional contract
  - Partially Observable Markov Decision Process for uncertainty handling
- Has in-situ **learning** capability
  - developed at design time, trained during operational use (“learning”)
  - in-use learning (hidden states, transitions, emissions)
- Addresses key design **trade-offs**
  - correctness (V&V) vs flexibility/adaptability (resilience)
- Applications
  - multi-UAV swarms
  - system-of-systems (SoS) networks (e.g., self-driving cars)
  - closed-loop mission assurance

Resilience Contract (RC)





RC evaluates POMDP reward; typical responses:

- Keep going
- Stop
- Enforce trajectory to a safe state
- Notify support team

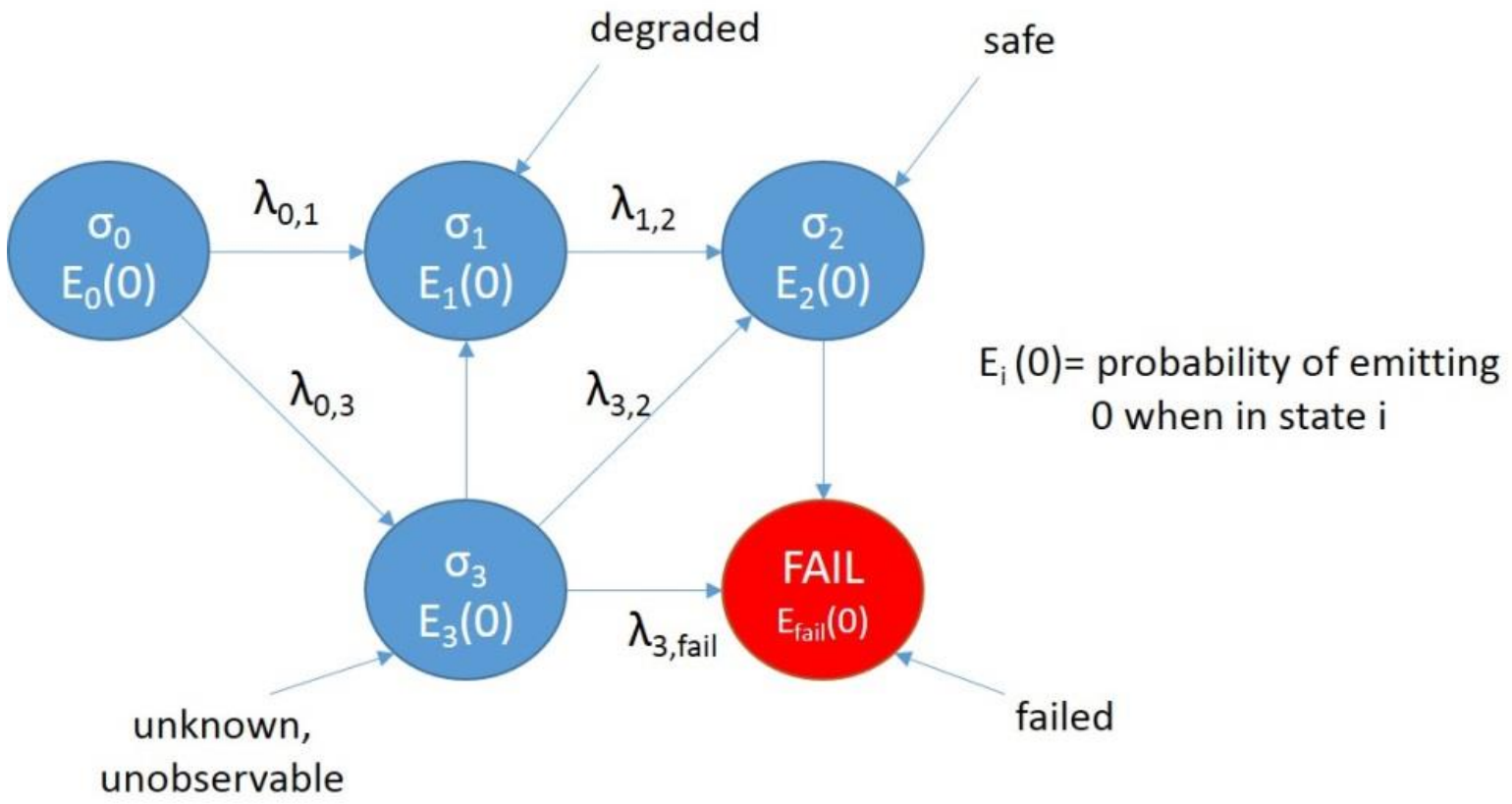
Rationale

- Engineered resilience is a “messy” problem
  - incompatible with invariant methods
  - requirements can be imprecise (especially initially)
  - actions can be unclear (especially initially)
  - system states can be ambiguous (partial observability, uncertainty)
- Want a formal methodology consistent with theorem-proving
  - key tradeoff: flexibility (messy problem) vs. correctness (V&V)
- RC Approach: probabilistic + formal modeling
  - relax assumptions and guarantees in traditional contract – enables dealing with messiness while being compatible with formal V&V
  - POMDP accounts for invariant knowledge and makes provision for in-use learned knowledge
  - POMDP is a means for decision making based on belief state and action policy
  - RC functions like a closed-loop control system – outcomes of actions are observed and used to determine next actions

Partially Observable Markov Decision Process (POMDP)

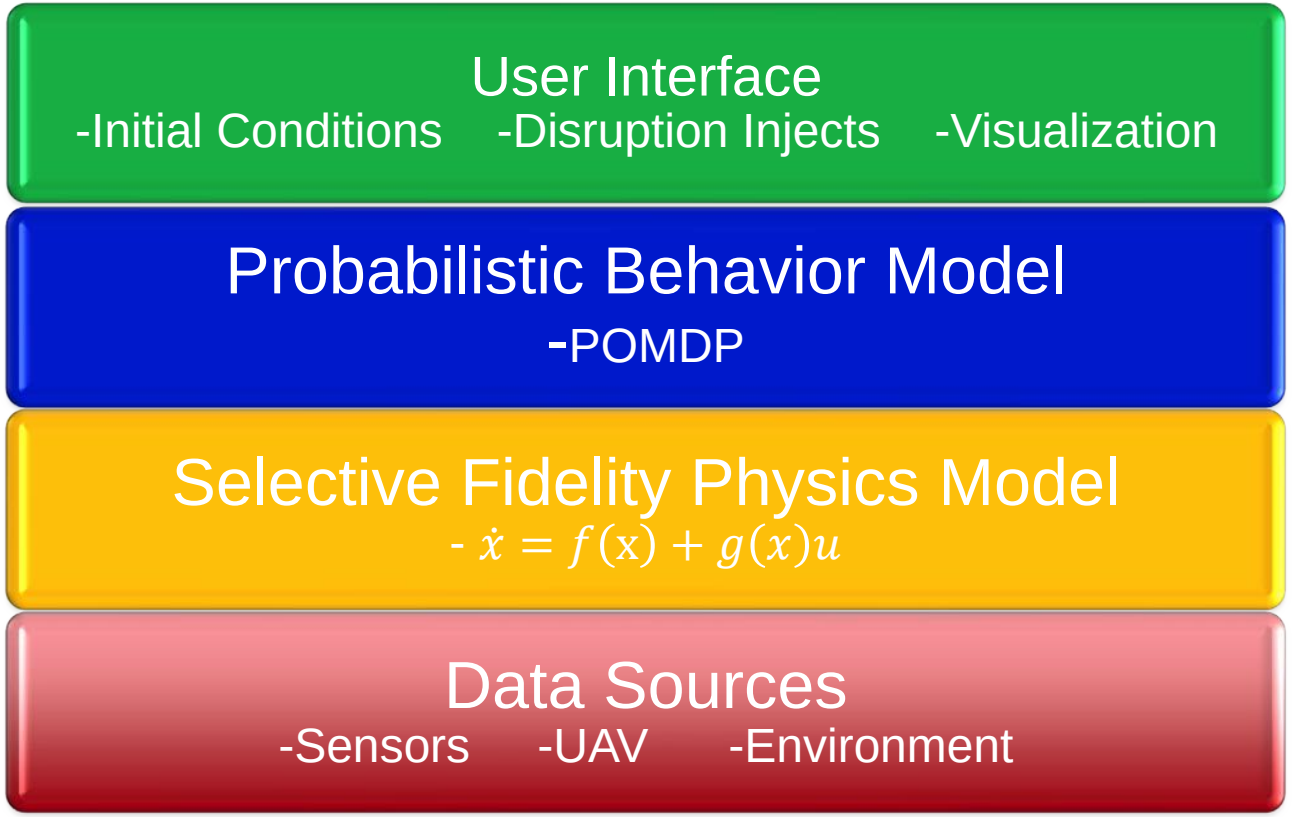
- Defined by:
  - set of: states S, actions A, observations O
  - transition model, reward model, observation model
- Rationale for Use
  - many real world problem environments are not fully observable
  - partial observability implies current state not necessarily known, system state may not be fully identifiable
  - agent cannot execute optimal policy with respect to what is known for that state (this is why heuristics become important)
  - Markov assumption invariably holds
- Markov assumption
  - optimal policy depends only on current state
  - applies to transition model

Exemplar POMDP Model



Enabling Technologies

Layered Architecture



POMDP Model Formulation

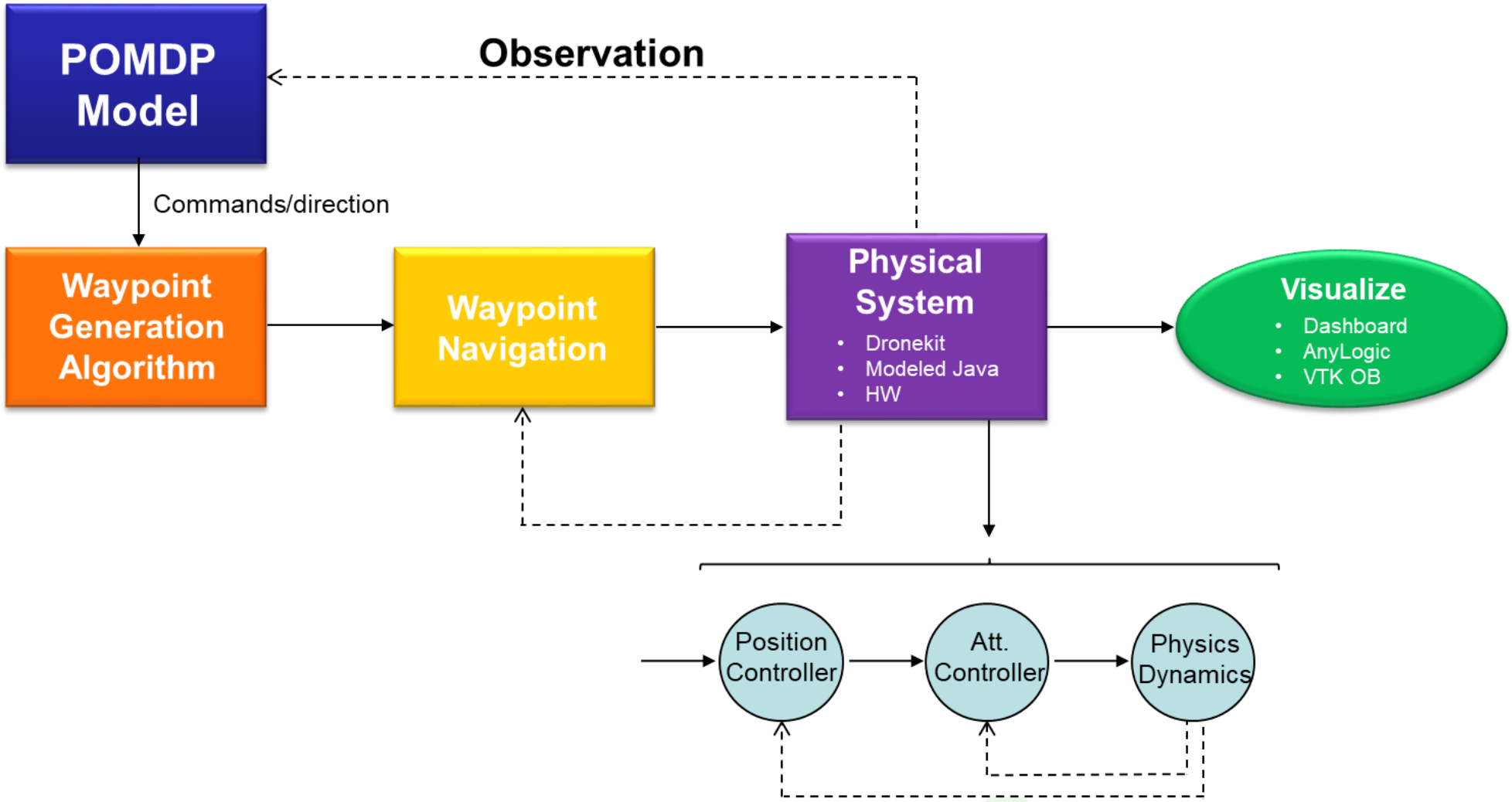
- Probabilistic Database:
  - Transition Matrix
  - Observation Matrix
  - Reward Matrix
  - Belief State Initialization
- Mathematical Formulas:
  - Belief State Update:

$$b(s') = \frac{p(o|s')\sum_{s\epsilon S} p(s'|a,s).b(s)}{\sum_{s'\epsilon S} p(o|s').p(s'|a,\ \ b)}$$

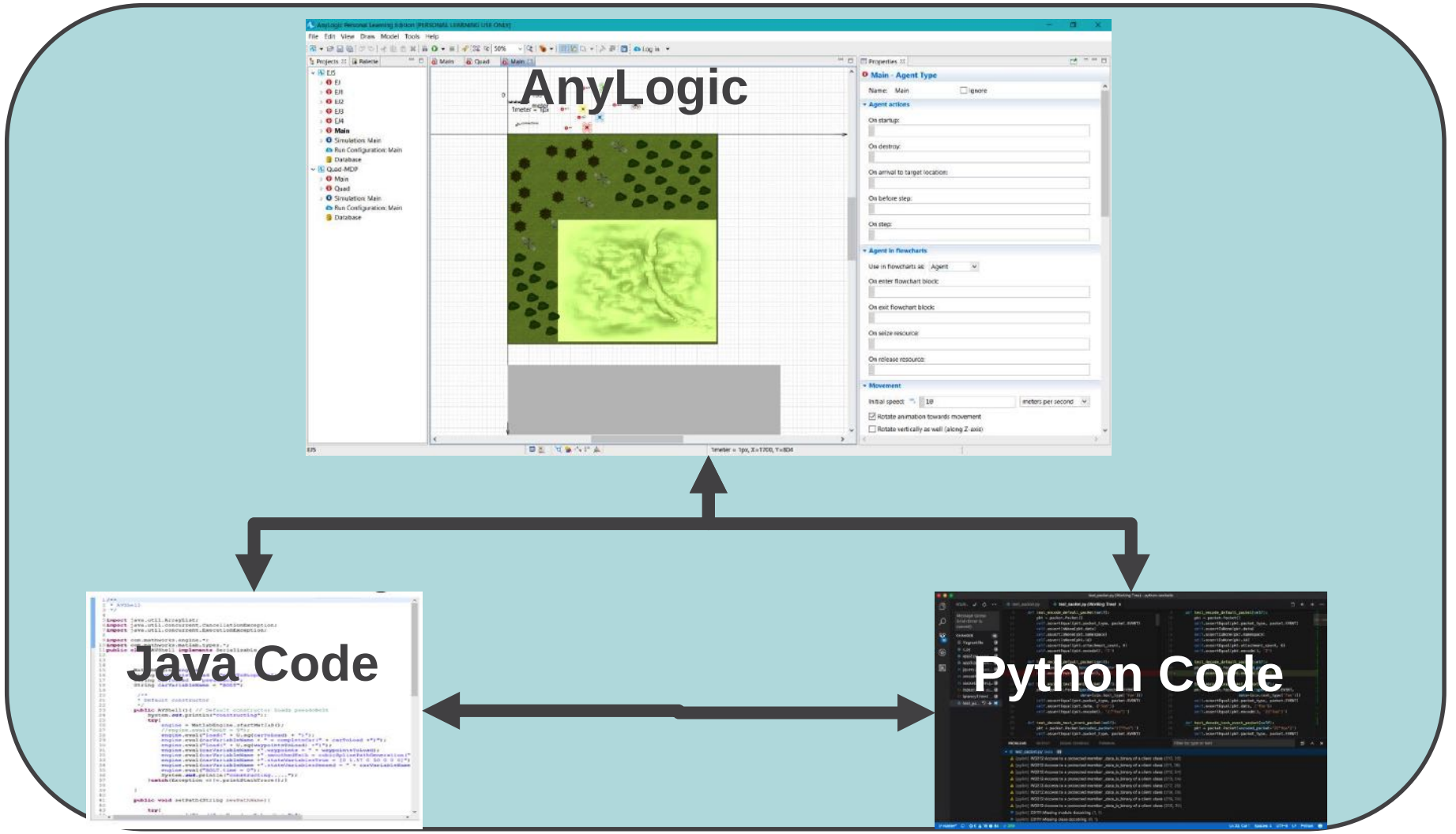
  - Immediate Reward or Expected Reward with Observation = o:

$$R(a)_o = \sum_{s\in S} b(s)_o * Reward\_mat(\text{s, a})$$

Integrated Model Representation



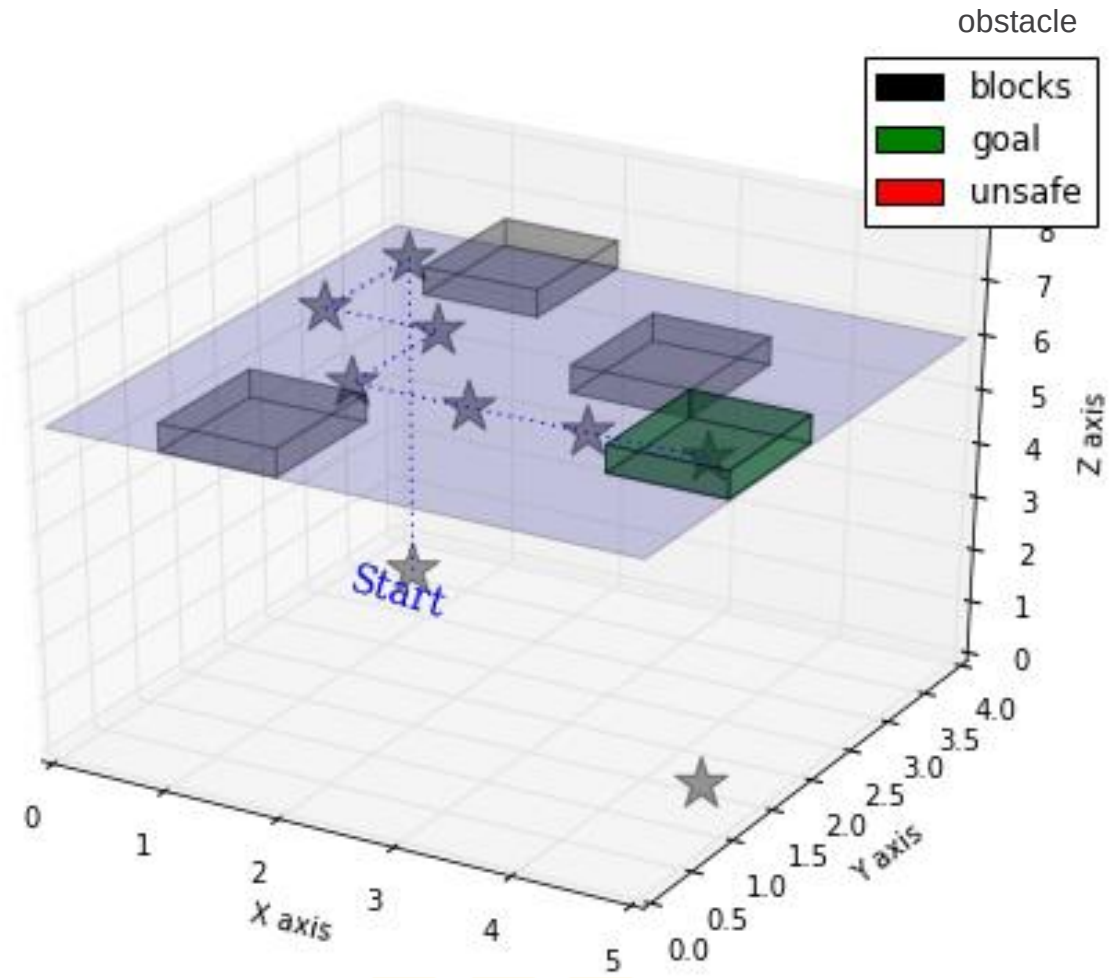
Technology Platform: Simulation Workflow



Hardware Testbed

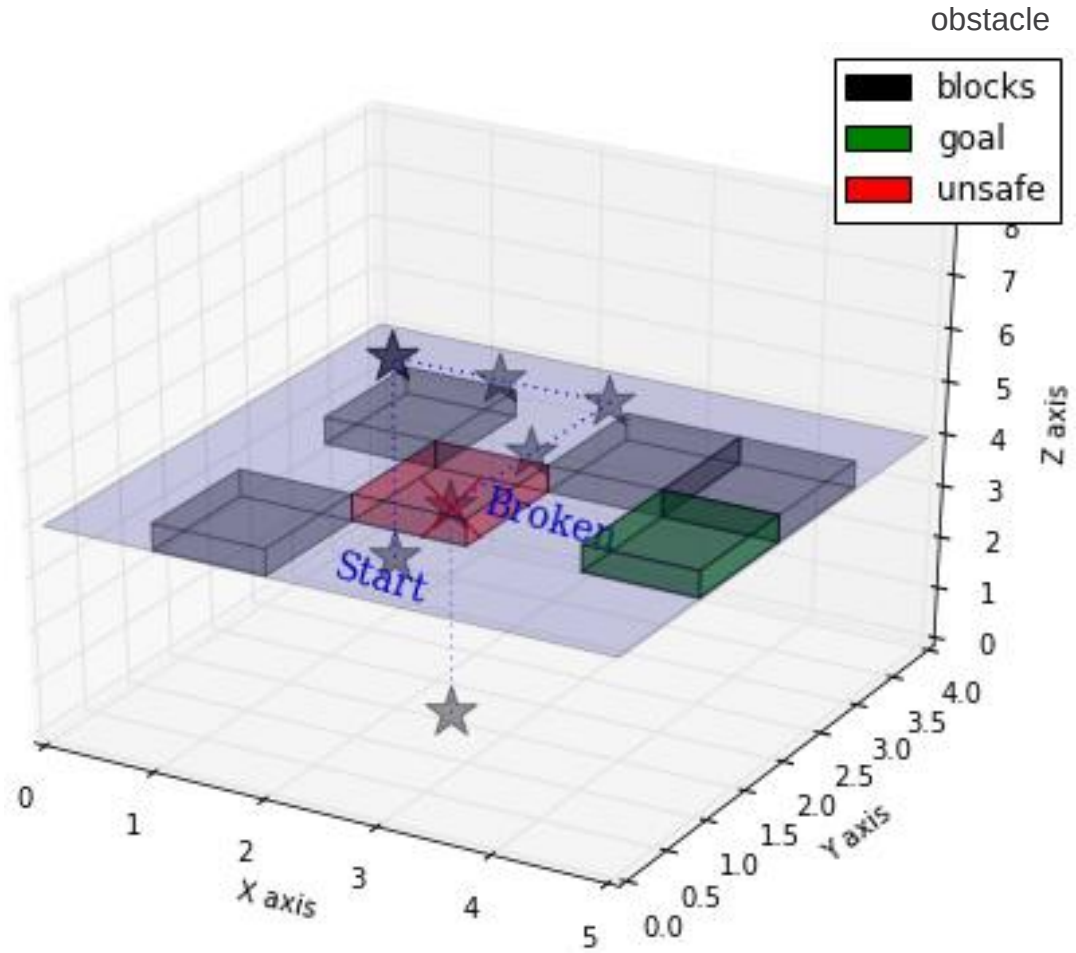
Illustrative Example

MDP Model:
QC Flight Example 1



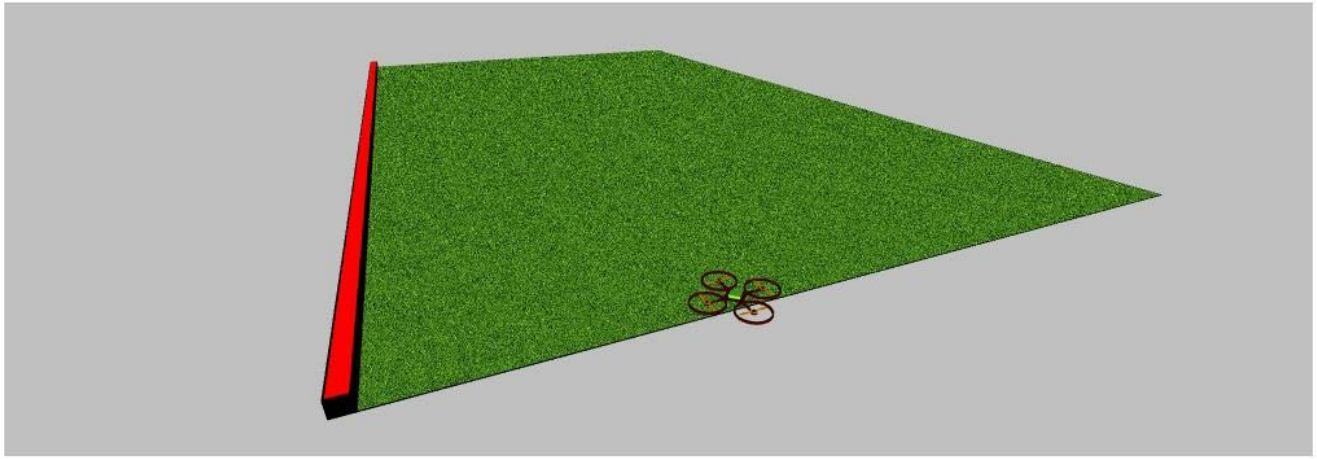
A QC navigates from start position to goal by observing and avoiding obstacles.

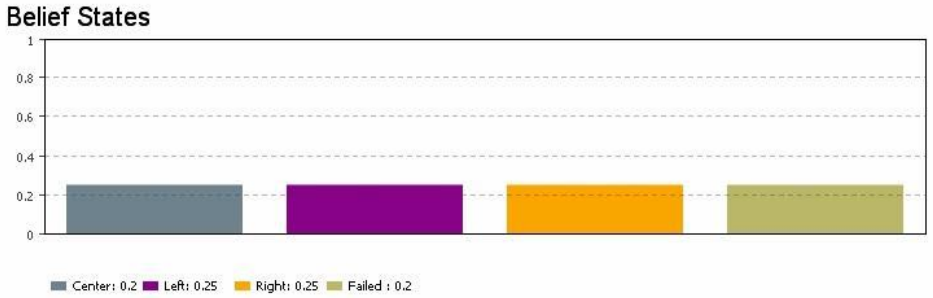
MDP Model:
QC Flight Example 2

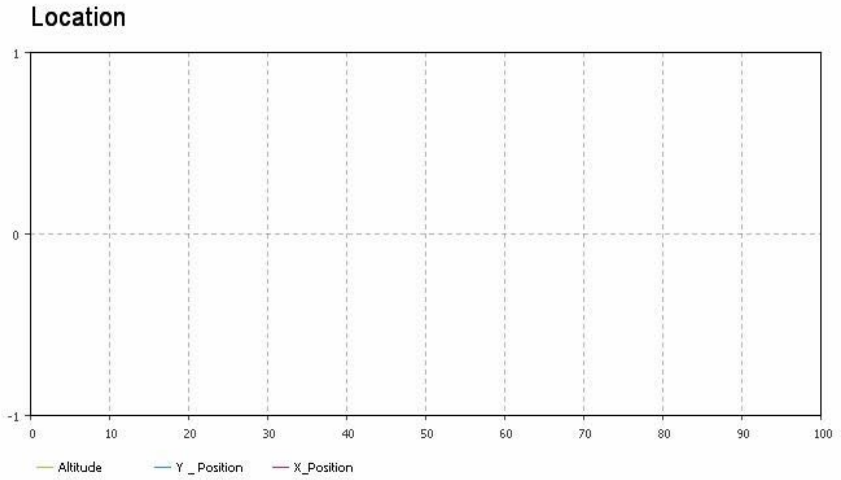


A QC detects a critical fault condition and lands

POMDP Waypoint Navigation





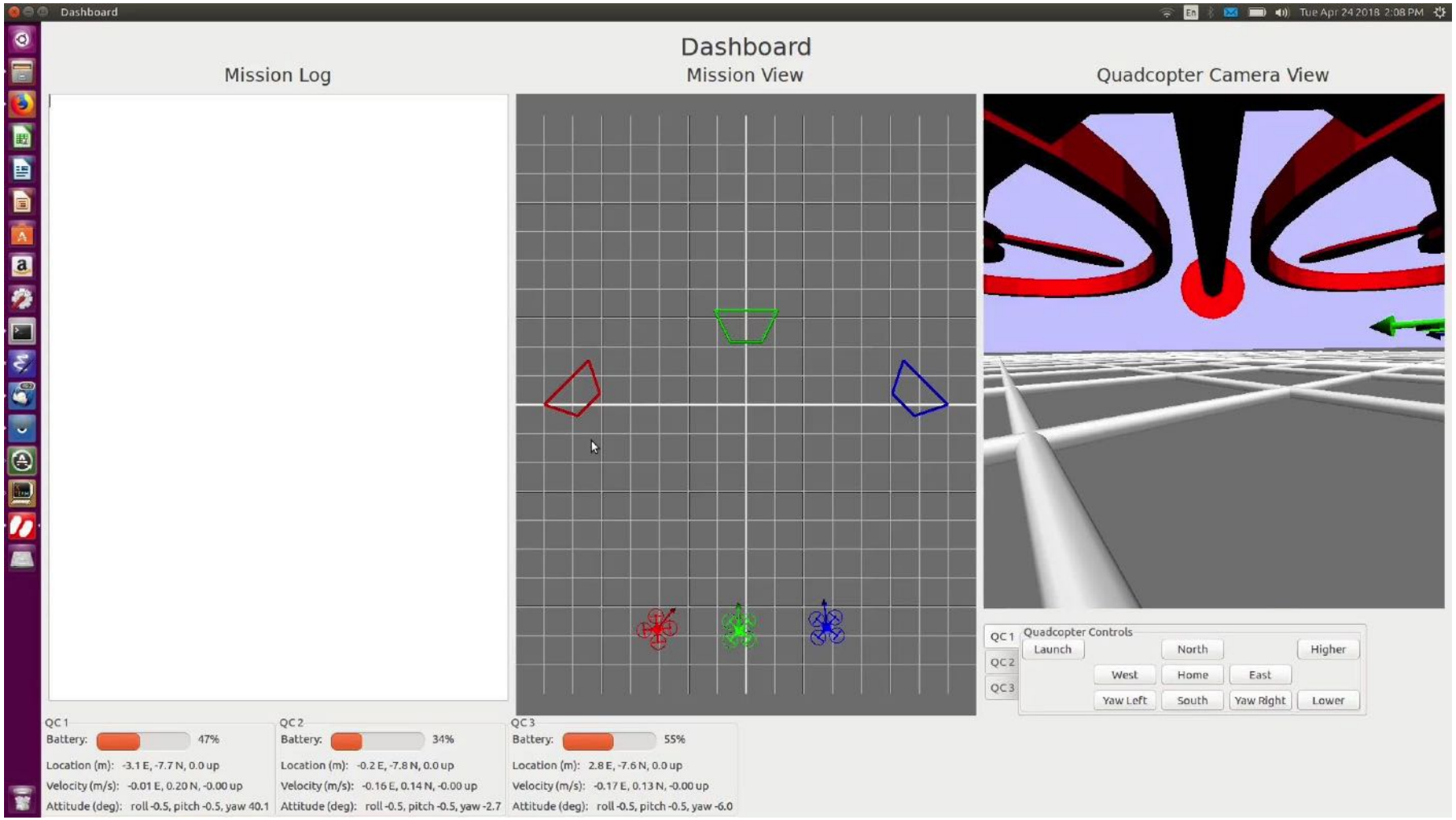


Multi-Vehicle Monitoring and Control Demo

Multi-UAV Monitoring and Control Dashboard

- Demonstration
  - customizable dashboard for monitoring and control of simulated/physical vehicles
- Underlying technologies
  - dronekit platform with visualization facilities
  - quadcopters (hardware) and quadcopter simulation models
- Key capabilities
  - simulated vehicles exhibit behavior of physical vehicle (real hardware)
  - same commands used to control simulated and physical vehicles (quadcopters)
  - can easily replace simulated vehicles with physical vehicles

Simulated QCs Working with Dashboard



Lessons Learned

Lessons Learned

- POMDP model able to perform simple actions with incomplete knowledge
- Just enough fidelity in physics models reduces computational complexity
  - less computation-intensive
  - adequate realism (selective fidelity) for seamless interaction with probabilistic model
- Language compatibility in technology platform essential for smooth integration
  - Java – Python integration
- Combination of contracts, hierarchical POMDP, and heuristics prevent state space explosion
  - provide required inputs from physics model to POMDP

Research Contributions

Research Contributions

- **Resilience Contract** – a hybrid modeling construct for stochastic systems
- **Experimentation testbed** - for system/SoS design, integration and evaluation
  - identify and resolve mismatches between probabilistic decision-making and physics modeling layers
  - e.g., vehicle physics model and POMDP model can run at different time scales
    - dynamic physics model runs every 0.01 seconds to assure requisite accuracy
    - POMDP model runs at a slower rate (issues high level commands)
    - right sampling rate for POMDP is determined experimentally
    - overall response time to action command needs to be minimized
- **Exemplar Demonstration** – multi-UAV monitoring and control

Summary

- Resilience – a key requirement of 21st century systems/networks to cope with disruptions
  - growing system and operational environment complexity
  - need for long-lived, adaptable and self-adaptive systems
- Current approaches – ad hoc, inadequate for V&V, do not scale
  - difficult to verify model correctness and validate behaviors
  - difficult to assess their long-term impact
- Innovative Approach – combines formal and probabilistic system modeling
  - resilience contract - combination of formal and probabilistic modeling
  - tradeoff between system model correctness (verifiability) and model flexibility (resilience)
- Demonstration – multi-UAV monitoring and control in testbed and actual environment
  - experimentation testbed – explore resilient design options
  - smart dashboard – monitoring and control of simulated and physical vehicles
  - simulated and physical vehicles
  - plan view and individual quadcopter view
- Way Ahead – continue development of the overall approach and prepare for transition

Way Forward: Near-Term

- Disruptions
  - Random injections
  - Random Fault Behavior
  - Random Duration
  - Random Severity
- Time
  - CPS require strong time semantics
  - Time-critical events
  - Hard, semi-hard constraints
- Learning
  - New (unseen before) states
  - Update transition and emissions
  - Update policy

28th Annual INCOSE international symposium

Washington, DC, USA

July 7 - 12, 2018

www.incose.org/symp2018