Formal Methods in Resilient Systems Design using a Flexible Contract Approach

Sponsor: OUSD(R&E) | CCDC

By
Dr. Azad Madni

11th Annual SERC Sponsor Research Review
November 19, 2019
FHI 360 CONFERENCE CENTER
1825 Connecticut Avenue NW, 8th Floor
Washington, DC 20009

www.sercuarc.org

Project Team

- Prof. Azad Madni, Principal Investigator
- Prof. Dan Erwin, Co-Investigator
- Dr. Ayesha Madni, Project Manager
- Edwin Ordoukhanian, RA, Hardware-Software Integration
- Parisa Pouya, RA, Probabilistic System Modeling
- Shatad Purohit, RA, Model Based Systems Engineering

Outline

- Background
- Research Objectives
- Accomplishments Summary
- Technical Approach
- Prototype Implementation
- Findings and Lessons Learned
- Technology Transition

Background

- 21st century DoD systems will continue to be complex, long-lived, likely to be extended / adapted to new missions over their lifetime, and with stringent physical and cybersecurity requirements
- These systems will need to be resilient when operating in dynamic, uncertain environments comprising hostile / deceptive actors
- A resilient system is one that is capable of safe operation in the face of systemic faults, failures, and unexpected disruptions
- Design of resilient DoD systems poses unique modeling challenges because of need to be correct, adaptable and continuously learning when operating in partially observable, dynamic environments
- Developing such a model will contribute to the body of knowledge in MBSE as well as complex systems modeling and simulation

Research Objectives

- Develop a formal modeling approach for designing resilient systems
- **Domain:** Autonomous Systems and System-of-Systems

Complicating Factors

- Partial observability
- Noisy sensors
- Failures and malfunctions
- Intelligent / deceptive adversary
- Changing goals or plans

Accomplishments Summary

- Developed innovative **closed-loop modeling** construct
  - resilience contract enables system model verification while affording flexibility for adaptation and reinforcement learning
- Developed **exemplar prototype** supported by rudimentary testbed
  - evaluated resilience techniques for multi-QC swarm operations
  - tested POMDP algorithms with fixed and dynamic obstacles
- Experimented with **POMDP algorithm**
  - navigation in presence of fixed and dynamic obstacles
  - with different n-step lookahead options
- Assembled a **transition package** comprising
  - installation and user guide
  - description of software modules and hardware specification
- **Transitioned prototype** to The Aerospace Corporation
  - for use on their MBSE initiatives and complement their MBSE/DE testbed

Technical Approach

Characterizations of System Resilience

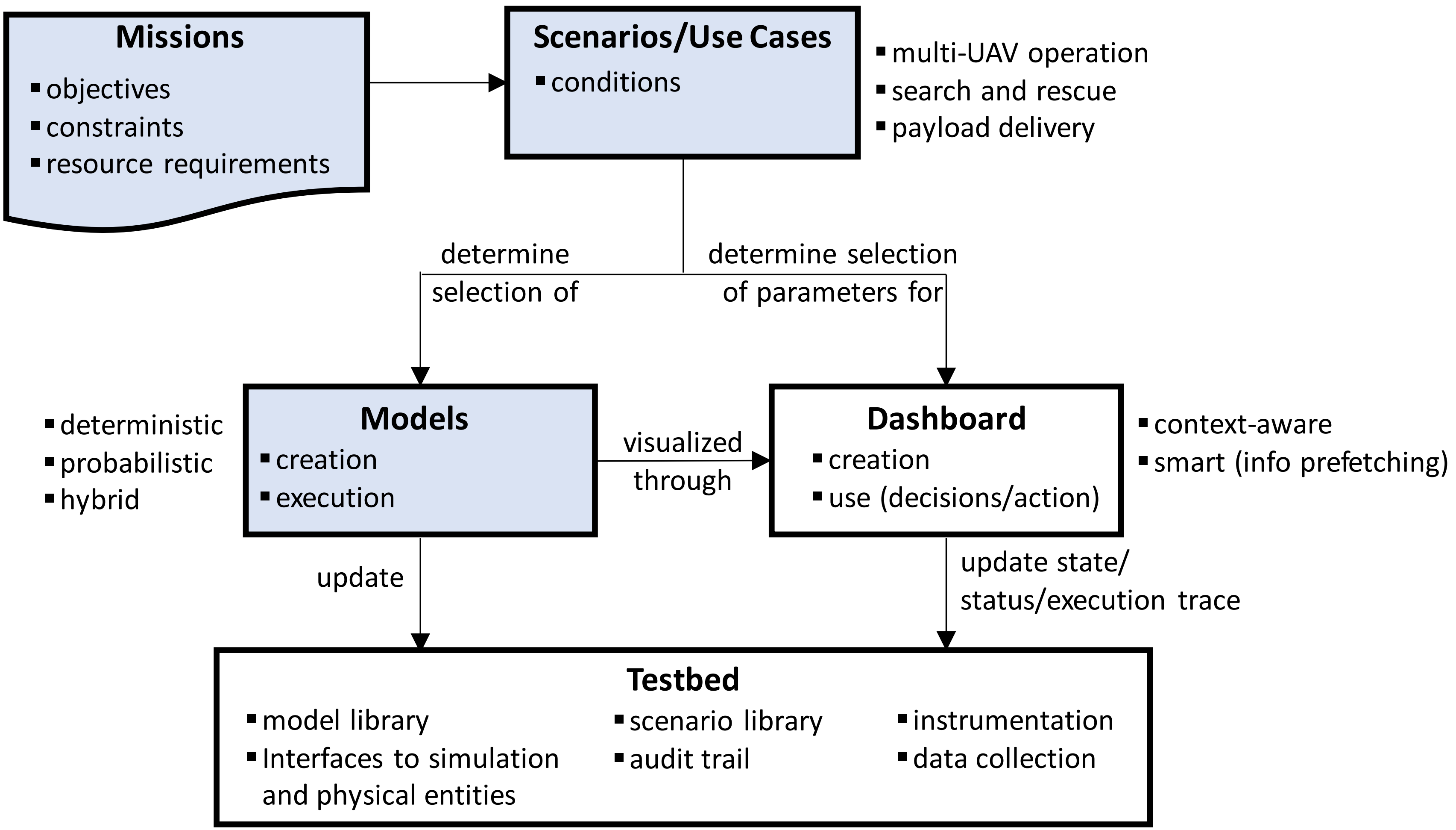
- **Recoverability:** Ability of system to rebound and return to equilibrium (fully/partially restore previous state)
- **Robustness:** Ability of system to absorb a disturbance within design envelope without any structural change
- **Dynamic Extensibility:** Ability of system to extend gracefully (i.e., add capacity/resources) in response to sudden increase in demand ("adaptive capacity")
- **Adaptability:** Ability of system to monitor problem context and adjust continually through dynamic reorganization/reconfiguration to circumvent or respond to disruptions

Not all characterizations lead to productive lines of inquiry for realizing resilient systems! **Dynamic Extensibility** and **Adaptability** do.

Modeling Requirements for Resilient Systems

- Verifiability (provable correctness)
- Flexibility (adapt to changing conditions)
- Bidirectional reasoning support (resilient response)
- Scalability and extensibility (no. of agents, interconnections)
- Provide useful outputs with partial information (not “data hungry”)
- Learn from new evidence (observations)

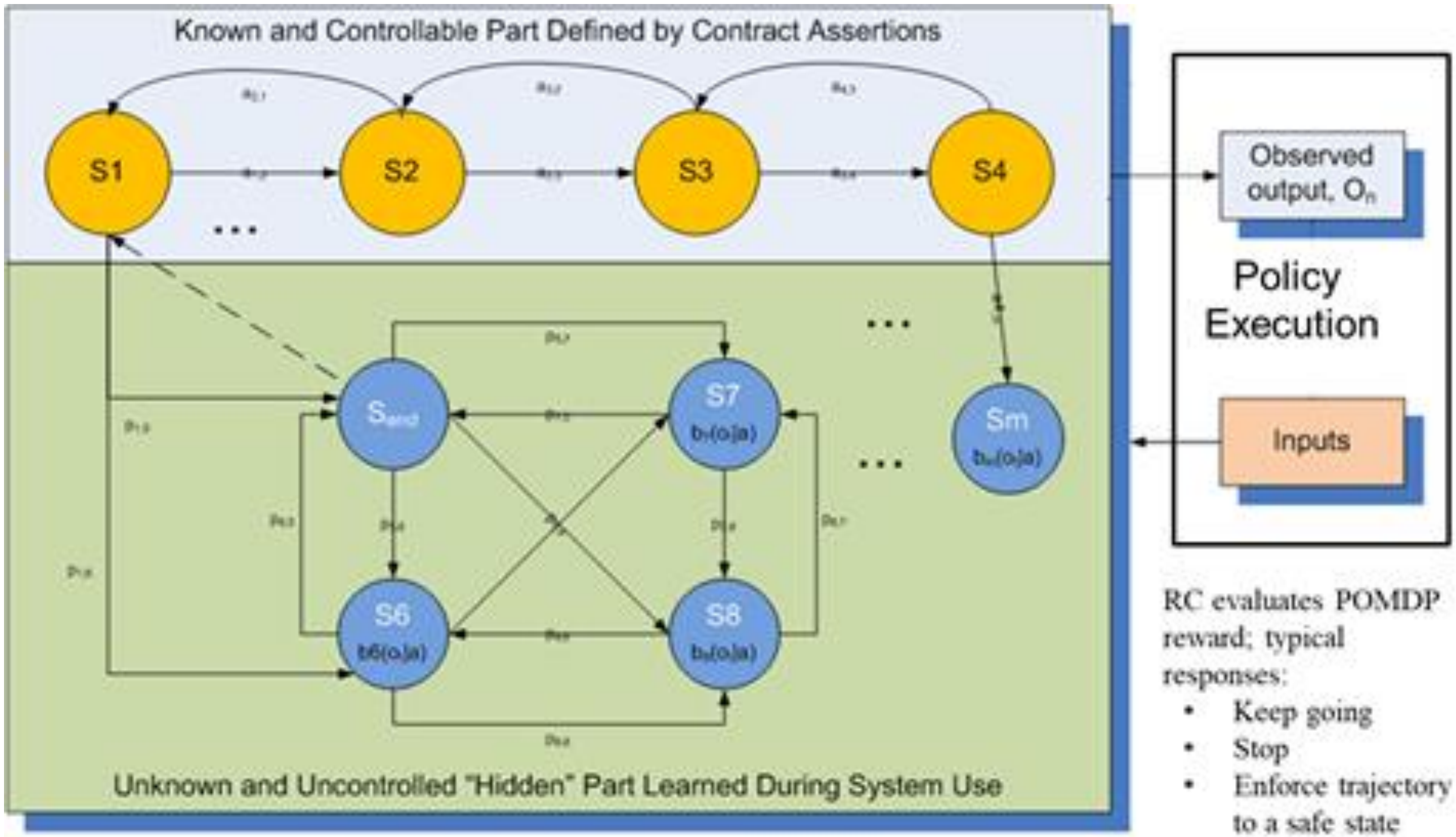
Conceptual Framework



Resilience Contract: Key Characteristics

- Probabilistic extension of traditional contract
  - — Relaxes "assert-guarantee" - replaces with "belief-reward" (flexibility)
  - — Partially Observable Markov Decision Process (uncertainty handling)
  - — In-use reinforcement learning (hidden states, transitions, emissions)
  - — Heuristics/pattern recognition (complexity reduction)
- Exhibits desired model characteristics
  - — **Verifiability:** key to safety and security
  - — **Flexibility:** key to adaptability and resilience
  - — **Learning:** key to performance improvement

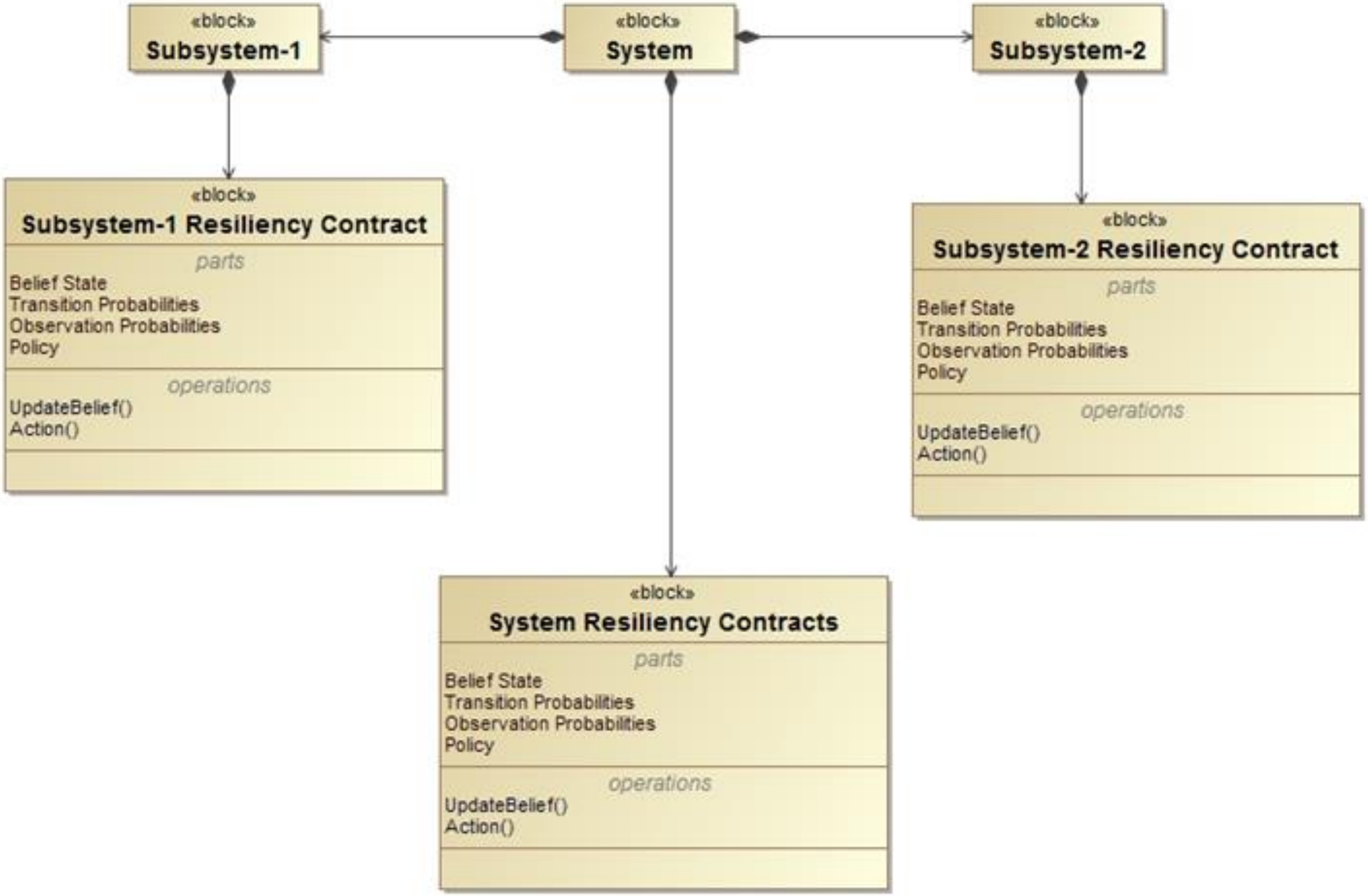
Resilience Contract (RC)



RC evaluates POMDP reward; typical responses:

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

Resiliency Model



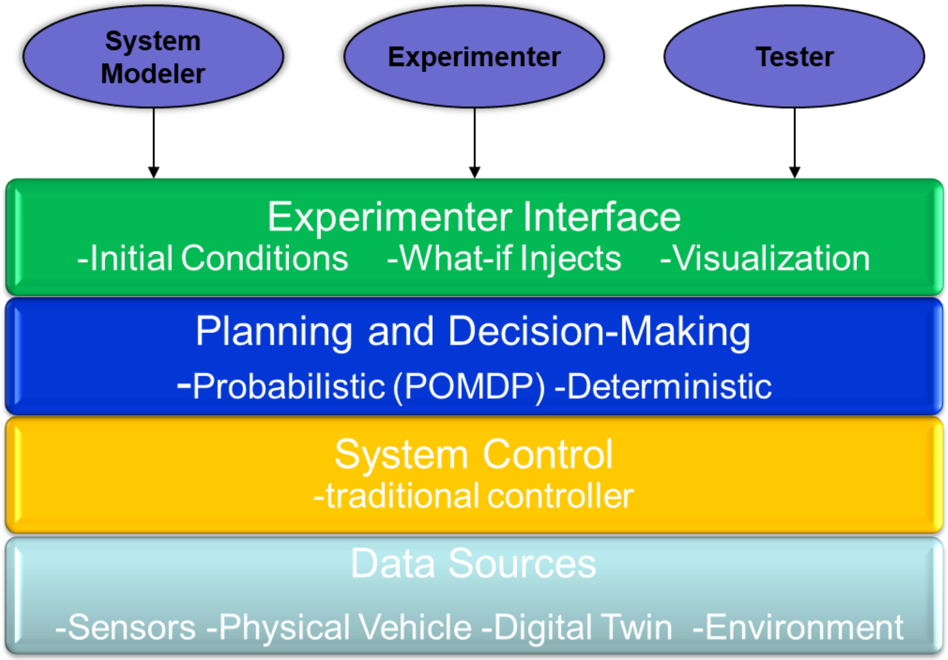
Reinforcement Learning

- Is key to incrementally updating an incomplete system and environment model with observations made by collection assets
- Requires real-time interaction with environment (observations)
- Take actions based on current knowledge of system states and real-time observations
- Sources of learning: sensors, networks, people

Testbed Overview

- Goal
  - —enable fundamental understanding of state-based modeling techniques, self-learning algorithms, and adaptation concepts
  - —support prototyping, evaluation and demonstration
- Prototyping Platform
  - — fly vehicle indoors in a laboratory or outdoors in the real-world
  - — large enough to carry onboard computer with suite of sensors (e.g. camera)
  - —onboard computer runs autopilot software as well as POMDP
  - — support open source software
- Evaluation Platform
  - —verify models (correctness analysis)
  - —explore concepts of operation (different assumptions, technologies)
  - —conduct simulation-based controlled experiments (e.g., probabilistic models)
- Demonstration Platform
  - —demonstrate a prototype UAV whose actions could be controlled by a decision-making algorithm such as POMDP

Testbed Architecture



- Developed concurrently with prototype system
- Currently supports system modeling, model verification, system behavior simulation, threat simulation
- Simulations runs on separate machines within a distributed, networked architecture

Prototype Testbed

- Multiple Quadcopters (QCs)
  - driven by Raspberry Pi and Navio Flight Controller
  - full IMU: 3-axis accelerometers, rate gyros, magnetometer
  - take inputs from laptop and/or remote controller
    - control values (throttle, roll-pitch-yaw)
    - perform autonomous flight
- Current Capabilities
  - run customized Python scripts to control QCs
    - Using dronekit framework and commands
  - perform semi-autonomous flights
    - Able to launch, take-off, hover, and perform limited waypoint navigation
  - smart dashboard to monitor status and control position of QCs
    - communicate with both simulated and physical vehicles

Testbed Hardware

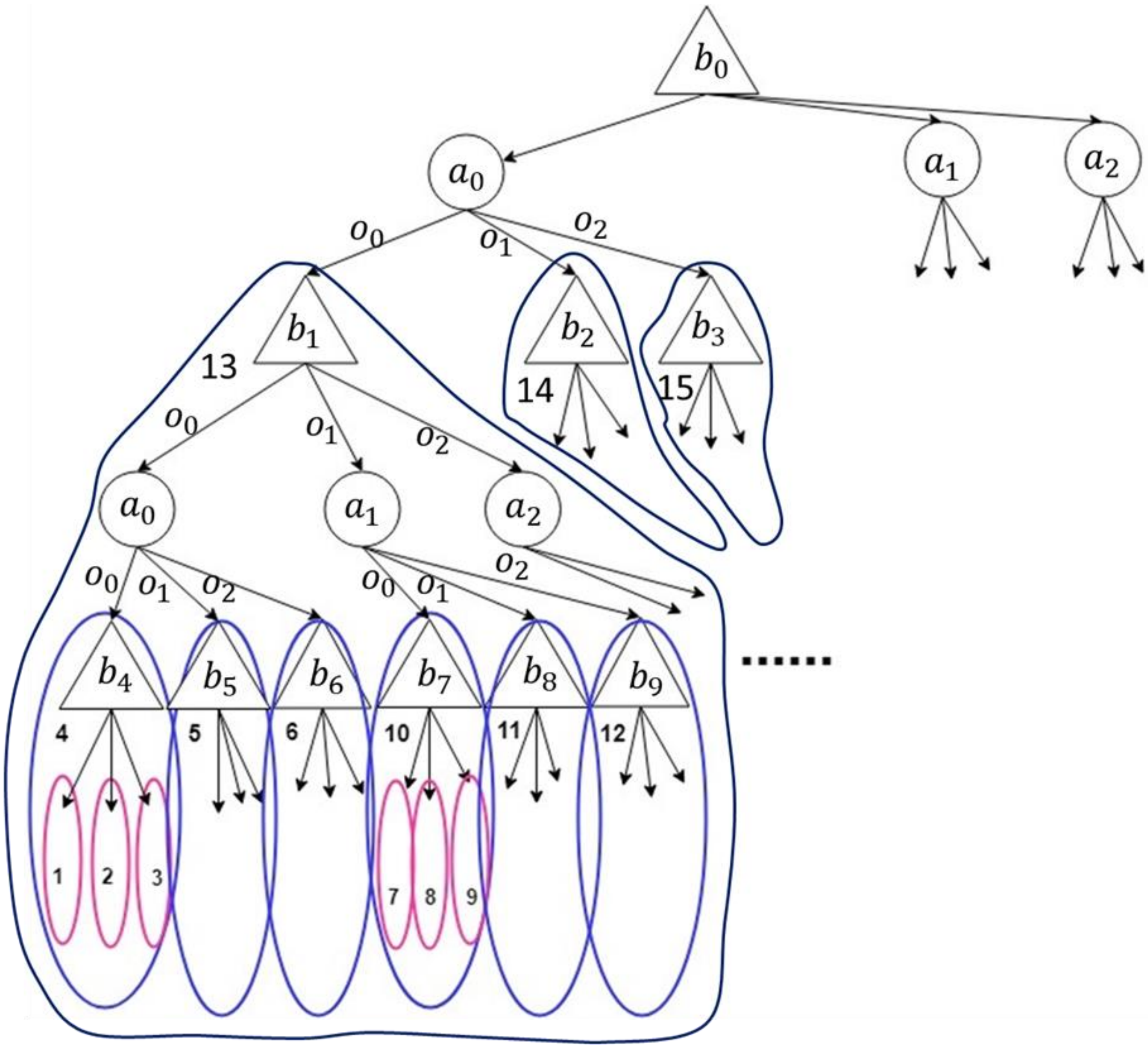
POMDP Solution Algorithm

Function $NStepLookAhead(b^t, N, \gamma)$

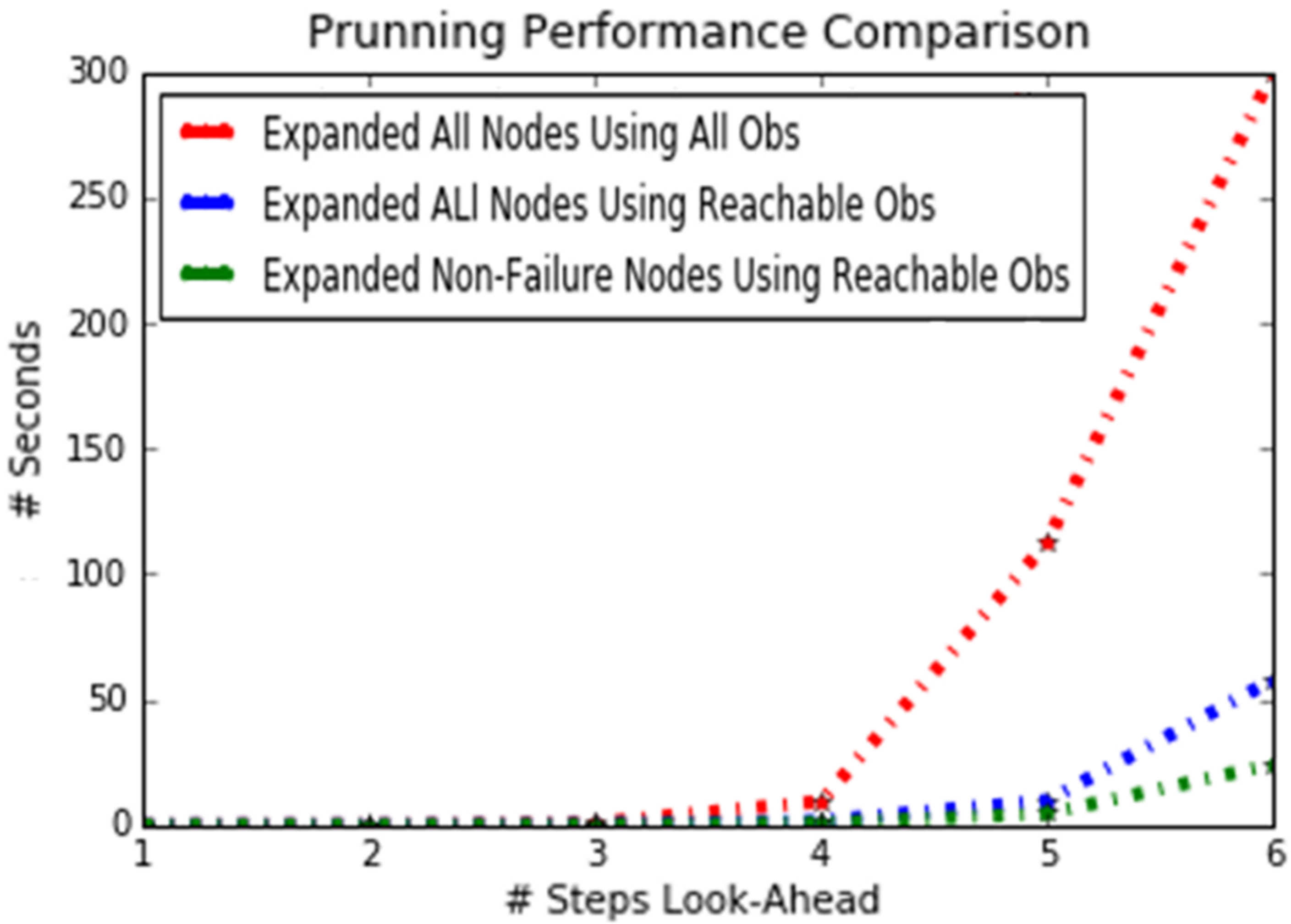
$value \leftarrow 0$
$If\ N\ equal\ to\ 1$:
  $For\ all\ a \in A$:
    $Calculate\ \Omega(b^t, a)$ # rechable observations for b^t and a
    $For\ all\ i \in \Omega(b^t, a)$:
      $b^{t+1} \leftarrow b^{t+1} | a, i$ # belief update
      $value \leftarrow value + (\sum_{s \in S} R(s) b^{t+1}(s)) * (\sum_{s \in S} b^t(s)\, p(i|s,a))$
    $EndFor$
  $EndFor$
  $Return\ value$
$If\ N > 1$:
  $For\ all\ a \in A$:
    $Calculate\ \Omega(b^t, a)$ # rechable observations for b^t and a
    $For\ all\ i \in \Omega(b^t, a)$:
      $b^{t+1} \leftarrow b^{t+1} | a, i$ # belief update
      $If\ b^{t+1}\ not\ Terminl$:
        $value \leftarrow value + \gamma\ NStepLookAhead(b^{t+1}, N-1, \gamma) + \left(\sum R(s) b^{t+1}(s)\right) * (\sum b^t(s)\, p(i|s,a))$
      $Else$:
        $value \leftarrow value + (\sum_{s \in S} R(s) b^{t+1}(s)) * (\sum_{s \in S} b^t(s)\, p(i|s,a))$
      $EndIf$
    $EndFor$
  $EndFor$
  $Return\ value$
$EndIf$

- N-Step Look-Ahead Online Algorithm
- Finds the optimal policy for the current belief state
- The belief state is updated at every time step
- The action that leads to the maximum long-term reward is considered the optimal policy for that belief state

N-Step Look-Ahead Visualization

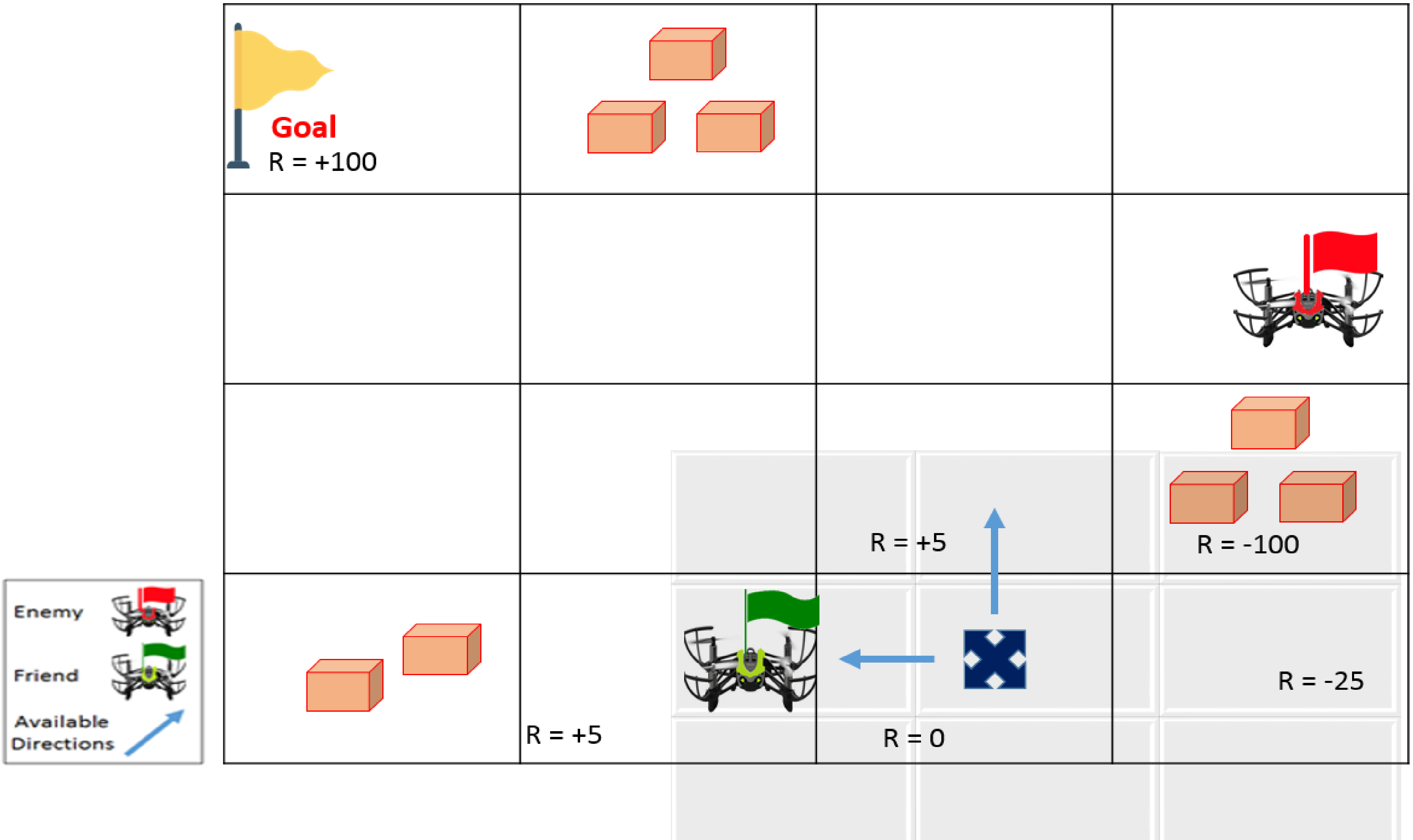


N-Step Look-Ahead: Pruning Performance



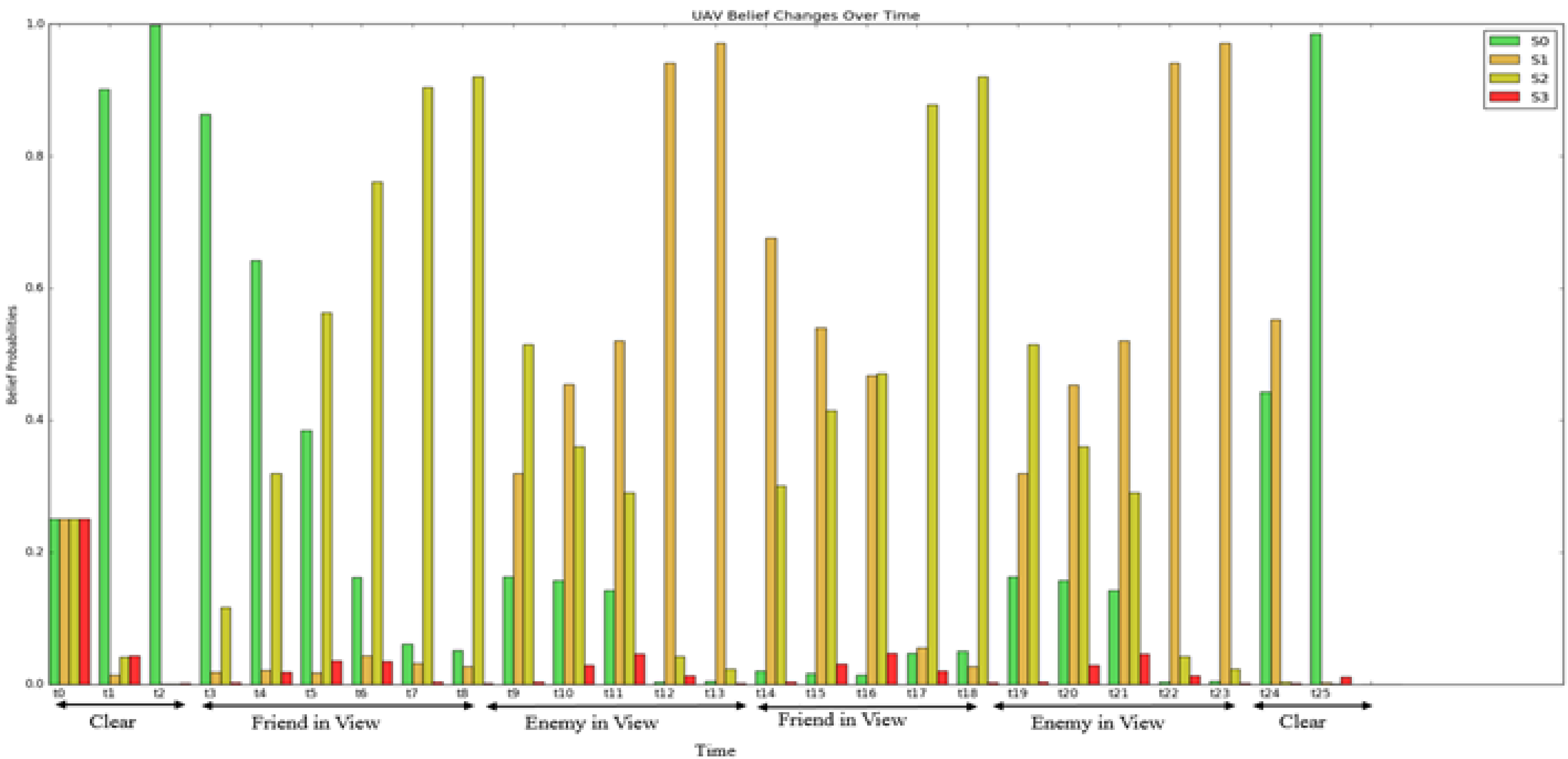
Illustrative Example: Navigation Through Hostile Environment

- **Goal:** Find safe, shortest path to pre-defined destination



Navigation with Dynamic Obstacles

- Exemplar Changes in Quadcopter Belief Vector



Experimentation with Resilience Contract

- **Experiment 1:** Performance of POMDP obstacle avoidance algorithm on testbed hardware (Raspberry Pi 3 QC flight computer)
  - POMDP ran on QC with no loss in performance while autopilot software was also running
    - POMDP guidance efficient enough - practical for real-time use on autonomous vehicles
- **Experiment 2:** Flying QC avoiding obstacles under POMDP control
  - developed and integrated a custom GPS driver into the Ardupilot software
  - able to fly quadcopter indoors in autopilot mode
  - excessive motor vibration prevented stable autonomous operation for long period to run obstacle avoidance algorithm
    - vehicle model issue, unrelated to POMDP

Technical Findings

- Key problem in implementing hybrid models
  - resolving mismatches between PDM and vehicle control layers
- Mismatch resolution
  - ensure that propagated commands from PDM layer to controller do not violate physical and regulatory constraints
  - propagate execution constraints from control layer to PDM layer for PDM layer to account for when issuing commands
  - incorporate heuristics (e.g., priorities, region of influence) to resolve conflicts and simplify computation
- POMDP and vehicle controller work on different time scales
  - dynamics model runs every 0.01 seconds (accuracy)
  - POMDP runs slower (high level decisions/commands)
  - waypoint navigation problem - minimize response time to action
  - ideal sampling period for POMDP determined experimentally

Findings and Lessons Learned

- POMDP model equivalent to a rule-based system for simple scenarios with full observability
- POMDP model states need to be defined and created based on various conditions that the system/SoS can potentially experience when interacting with its environment
- Ability to acquire new knowledge through reinforcement learning and expand the model as required makes POMDP modeling attractive for complex scenarios with partial observability
- POMDP value function and time horizon for estimating online policy are key parameters that influence system / SoS behaviors

Findings and Lessons Learned (cont'd)

- POMDP reward/value function should be designed to account for physical aspects of the vehicle
- POMDP model(s) should be designed to include both goal and failure states in the system state-space.
  - Based on the probabilities assigned to different states (including both failure and goal) and the changes in the beliefs over time, one can reason why an action is taken.
  - E.g. Th belief of failure reduces as the actions to avoid failure are taken.
- Concurrent development of testbed and system model facilitated experimentation and data collection
- Smart dashboard for monitoring and control of vehicles proved to be valuable for understanding and debugging vehicle behaviors

Prototype Transition

- Prototype combined with prototype from RT-183 to create a rudimentary modeling, simulation, execution monitoring and visualization testbed
- Transitioned integrated prototype to The Aerospace Corporation to complement and enhance their MBSE capabilities
  - Aerospace customers include NASA, NOAA, SMC, Air Force

Summary and Conclusions

- Resilience Contract (RC) is well-suited to modeling complex systems that operate in dynamic partially observable environments
  - —simultaneously addresses system model verification and system flexibility
  - — combines formal and probabilistic modeling with heuristics
- POMDP models can be constructed and solved using effective approximations with finite-step lookahead
- Prototype testbed had just enough capability for modeling and experimenting with different models, and for hardware-software integration
- Prototype transitioned along with dashboard created in RT-183 to The Aerospace Corporation

References

- Ordoukhanian, E., and Madni, A.M. Model Based Approach to Engineering Resilience in Multi-UAV System-of-Systems, MDPI *Systems,* special issue on *"Model-Based Systems Engineering,"* Feb 2019

- Madni, A.M., Sievers, M., Madni, A., Ordoukhanian, E., and Pouya, P. Extending Formal Modeling for Resilient System Design, *INSIGHT,* Vol. 21, Issue 3, pp. 34-41, October 2018

- Madni, A.M. and Sievers, M. Model-Based Systems Engineering: Motivation, Current Status, and Research Opportunities, *Systems Engineering*, Special 20th Anniversary Issue, Vol. 21, Issue 3, 2018.

- Madni, A.M. and Boehm, B. (eds), *"Engineered Resilient Systems: Challenges and Opportunities in the 21st Century,"* Procedia Computer Science 28 (2014), ISSN 1877-0509, Elsevier, 2014.

- Madni, A.M., and Sievers, M. Closed Loop Mission Assurance Based on Flexible Contracts: A Fourth Industrial Revolution Imperative, in *Systems Engineering in the Fourth Industrial Revolution: Big Data, Novel Technologies, and Modern Systems Engineering*, Kenett, R., Swarz, R.S., and Zonnenshaim, A. (Eds.), Wiley and Sons, expected Fall 2019

References (cont'd)

- Madni, A.M., Sievers, M., Erwin, D. Formal and Probabilistic Modeling in the Design of Resilient Systems and System-of-Systems, *AIAA Science and Technology Forum*, San Diego, California, January 7-11, 2019

- Sievers, M., Madni, A.M., and Pouya, P. Assuring Spacecraft Swarm Byzantine Resilience, *AIAA Science and Technology Forum*, San Diego, California, January 7-11, 2019

- Madni, A.M. Formal Methods in Resilient Systems Design Using a Flexible Contract Approach, *NDIA 21st Annual Systems Engineering Conference,* Tampa, Florida, October 22-24, 2018.

- Madni, A.M., Sievers, M., Ordoukhanian, E., and Pouya, P., and Madni, A. "Extending Formal Modeling for Resilient Systems," *2018 INCOSE International Symposium*, July 7-12, 2018.

- Madni, A.M. "Formal Methods for Intelligent Systems Design and Control," *AIAA SciTech Forum, 2018 AIAA Information Systems, AIAA InfoTech@Aerospace*, Kissimmee, Florida, January 8-12, 2018

Azad M. Madni

- Professor, Astronautical Engineering, University of Southern California
- Executive Director, Systems Architecting and Engineering Program
- Director, Distributed Autonomy and Intelligent Systems Laboratory
- Founder and CEO, Intelligent Systems Technology Inc.
- INCOSE Fellow, Pioneer and Founder
- Life Fellow, IEEE; Fellow, AAAS; Fellow, AIAA; Life Fellow, SDPS; Life Fellow, IETE
- Ph.D., M.S., B.S. in Engineering, UCLA; Graduate of Stanford's Executive Program
- **Research Interests**: Formal and Probabilistic System Modeling; Resilient Cyber-Physical-Human Systems; Interactive Storytelling in Virtual Worlds, Intelligent Systems Engineering
- **2019 Awards and Honors**
  - *2019 Presidential Award from Society of Modeling and Simulation International*
  - *2019 AIAA/ASEE Leland Atwood Award* for excellence in aerospace engineering
  - *2019 ASME CIE Leadership Award* for advancing use of computers in engineering
  - *2019 INCOSE Founders Award* for increasing global awareness of INCOSE
  - *2019 EC William B. Johnson International Inter-Professional Founders Award*
  - *2019 OCEC Prestigious Pioneering Educator Award*
- **Recent Books**
  - Madni, A.M., Boehm, B. et al. (eds.) *Disciplinary Convergence: Implications for Systems Engineering Research*, Springer, 2018.
  - Transdisciplinary Systems Engineering: Exploiting Convergence in a Hyper-Connected World (foreword by Norm Augustine) Springer, 2017
  - Tradeoff Decisions in System Design (foreword by John Slaughter), Springer, 2016
  - Madni, A.M. and Boehm, B. (eds), *"Engineered Resilient Systems: Challenges and Opportunities in the 21st Century,"* Procedia Computer Science 28 (2014), ISSN 1877-0509, Elsevier, 2014

Thank You