Combining Formal and Probabilistic Modeling in Resilient Systems Design

Azad M. Madni

University of Southern California

2019 CSER
conference on systems engineering research

Combining Formal and Probabilistic Modeling in Resilient Systems Design

Azad M. Madni
Michael Sievers
University of Southern California

April 3-4, 2019

2019 Conference on Systems Engineering Research
Washington, D.C.

This Photo by Unknown Author is licensed under CC BY

USC Viterbi
School of Engineering
Systems Architecting and Engineering

Outline

- 21st Century DoD Systems
- Engineered Resilience
- System Modeling
- Formal Probabilistic Approach
- Exemplar Problem
- Experimentation Testbed
- Initial Findings
- Summary

Copyright © 2014-2019 Azad M. Madni

21^{st} Century DoD Systems

conference on systems engineering research

- High complexity (hyper-connectivity, dependencies)
- Long-lived (> 20 years)
- Likely to be extended / adapted for over lifetime
- Stringent physical and cyber security
- Need dependability + adaptability + proactive error prevention
- Need to operate safely in dynamic, uncertain environments subject to disruptions
- To address these challenges, we need new models, methods and tools

Copyright © 2014-2019 Azad M. Madni

USC Viterbi
School of Engineering
Systems Architecting and Engineering

Engineered Resilience is a Difficult and Messy Problem…Why?

- **Requirements:** can be imprecise (especially initially)
- **Actions:** can be unclear (especially initially)
- **Environment:** can be unknown or partially known (partial observability, unknown hostile and/or deceptive actors)
- **System states:** can be ambiguous (uncertainty)

These characteristics are incompatible with traditional, invariant modeling methods

Copyright © 2014-2019 Azad M. Madni

USC Viterbi
School of Engineering
Systems Architecting and Engineering

Systems Modeling

- Primary means for engineering systems including resilient systems
- A fragmented area for engineering resilient systems
- Most serious problems result from the gap between requirements and models that need to reflect requirements
  - contribute to poor flow down of system requirements to software requirements
- Different aspects of system behavior represented by different models
  - need to harmonize different models

Copyright © 2014-2019 Azad M. Madni

System Modeling Requirements

- Verifiability (correctness)
- Flexibility (to adapt to changing conditions)
- Bidirectional reasoning support (resilience-related decisions)
- Scalability and extensibility (no. of agents, interconnections)
- Exploit partial information to generate value (not “data hungry”)
- Learn from new observations (system and environment states)

Copyright © 2014-2019 Azad M. Madni

Formal Probabilistic Approach

- Combines formal and probabilistic modeling with heuristics
  - enables flexibility (resilience)-verifiability (safety) tradeoffs
  - heuristics: help contain combinatorial explosion in state space
- Exploits reinforcement learning
  - learning of system states and environment with new evidence
  - system states model: Partially Observable Markov Decision Process
  - provides value even with partial information
- Employs a layered architecture
  - planning and decision making (top level) and control (bottom level)
  - decisions and information flow from top level to bottom level
  - execution constraints flow from bottom level to top level
  - global objectives have precedence over local goals (if conflict)
- Defines new construct: Resilience Contract
  - balance system verifiability and system flexibility requirements

Copyright © 2014-2019 Azad M. Madni

Exemplar Problem: Multi-UAV Operations

conference on systems engineering research

- UAVs used in missions in which environment is largely unknown and potential hostile with deceptive actors
- UAVs have to complete mission safely with original / descoped objectives
- UAVs can experience malfunctions and disruptions
- A mathematical model of environment is seldom available
- UAVs have collection assets to sense the environment
- UAVs can employ reinforcement learning to progressively learn the environment from sensed information
  - e.g., use RL to navigate through changing, partially observable environments

Copyright © 2014-2019 Azad M. Madni

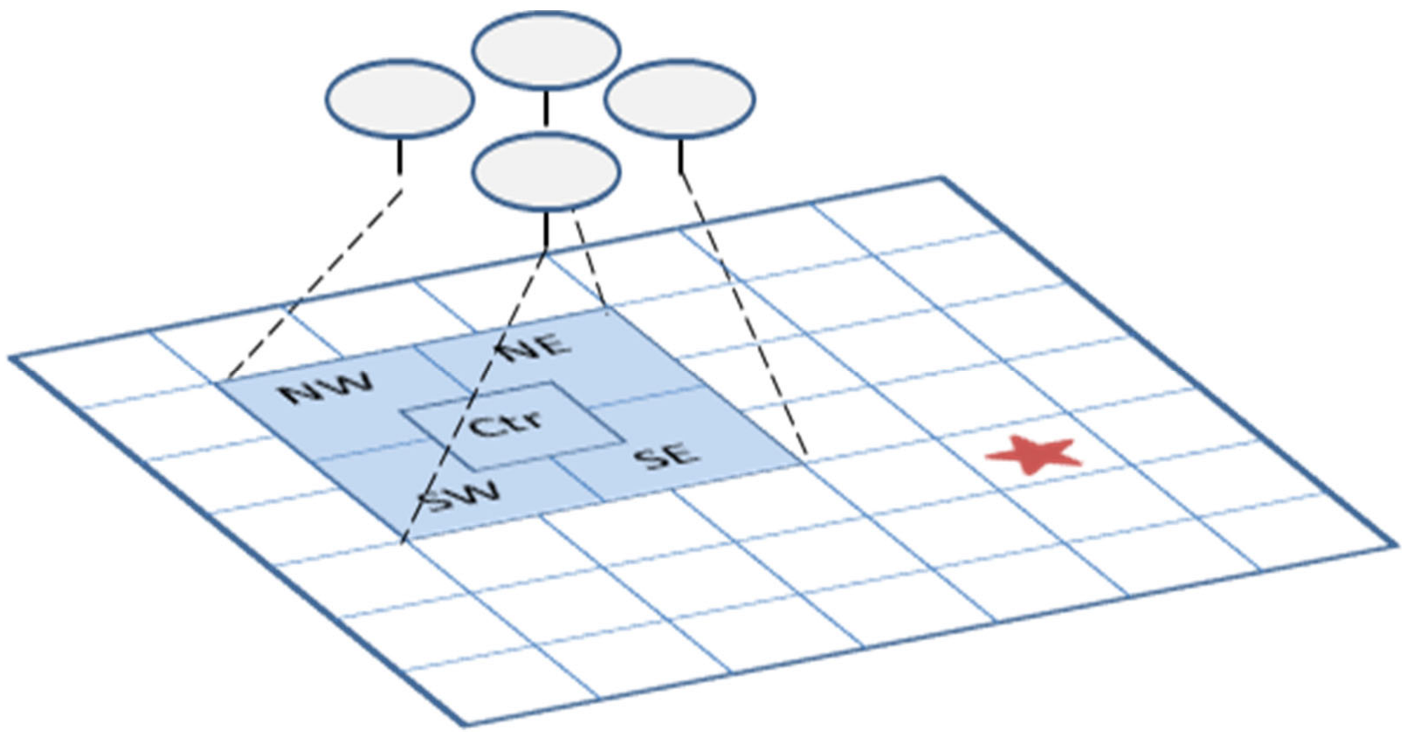
USC Viterbi
School of Engineering
Systems Architecting and Engineering

What Resilience Means for Problem Domain

- **Operate safely** in dynamic, uncertain environments
  - tolerate / survive systemic faults and failures
  - adjust / adapt to environmental disruptions
  - protect / defend against physical and cyber threats
  - reconfigure / restructure to minimize impact of disruptions (e.g., security breaches, loss of sensing node or comm link)
- **Accomplish goals** with incomplete information
  - e.g., navigate safely to destination with partial observability

Copyright © 2014-2019 Azad M. Madni

UAV position relative to a recon target (red star) and FOV (blue)



Copyright © 2014-2019 Azad M. Madni

USC Viterbi
School of Engineering
Systems Architecting and Engineering

Exemplar Contracts

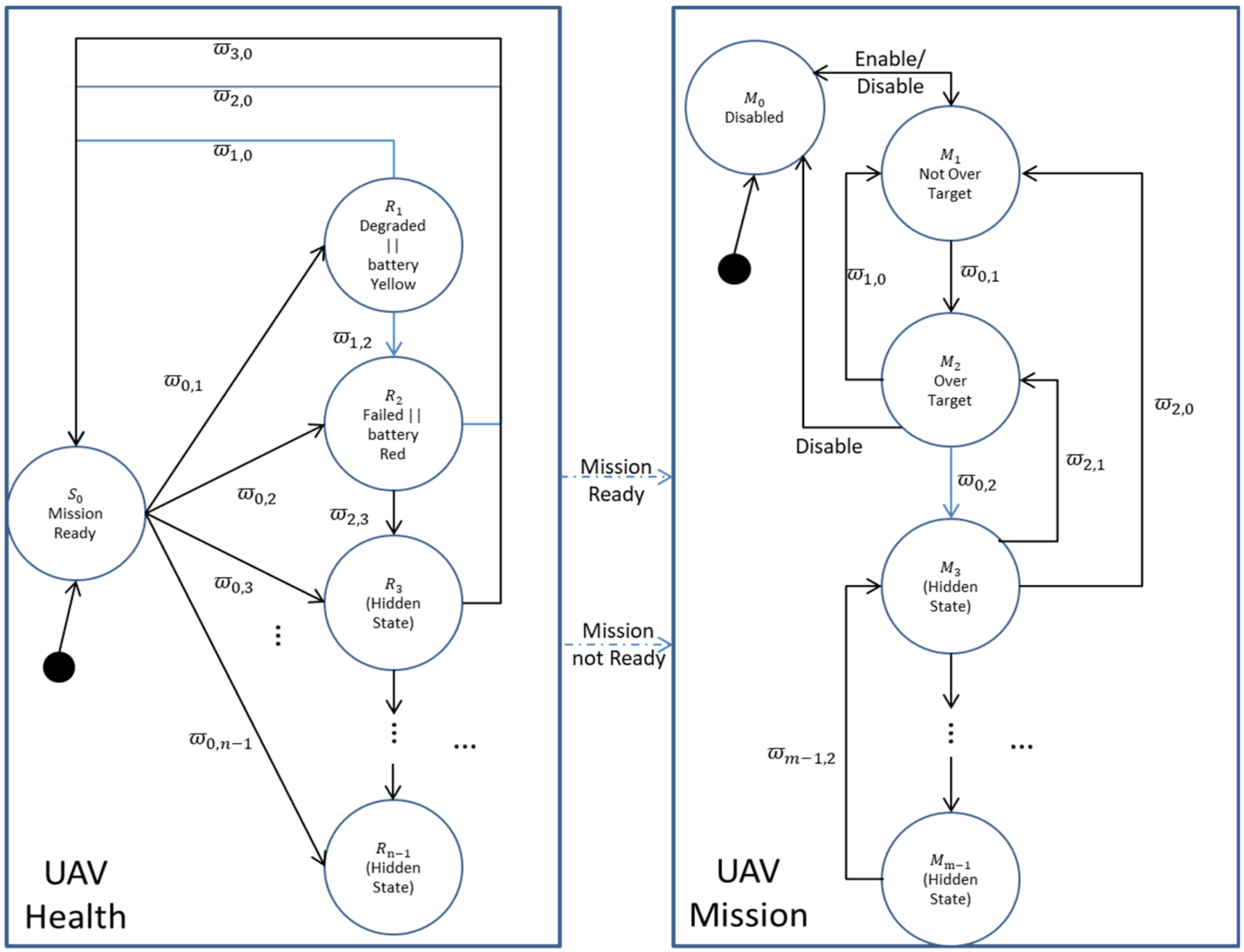
conference on systems engineering research

1. ¬overTarget && healthy && batteryGreen → move_to_target
2. ¬batteryRed && degraded || batteryYellow → move_to_base
3. batteryRed || failed → land
4. unknownHealth || unknownBattery → move_to_base
5. overTarget && CTR && healthy → takeImages & hover
6. overTarget && NW && healthy → takeImages & move SE
7. overTarget && NE && healthy → takeImages & move SW
8. overTarget && SW && healthy → takeImages & move NE
9. overTarget && SE && healthy → takeImages & move NW

Copyright © 2014-2019 Azad M. Madni

USC Viterbi
School of Engineering
Systems Architecting and Engineering

Simplified POMDP:
Health and Mission Models



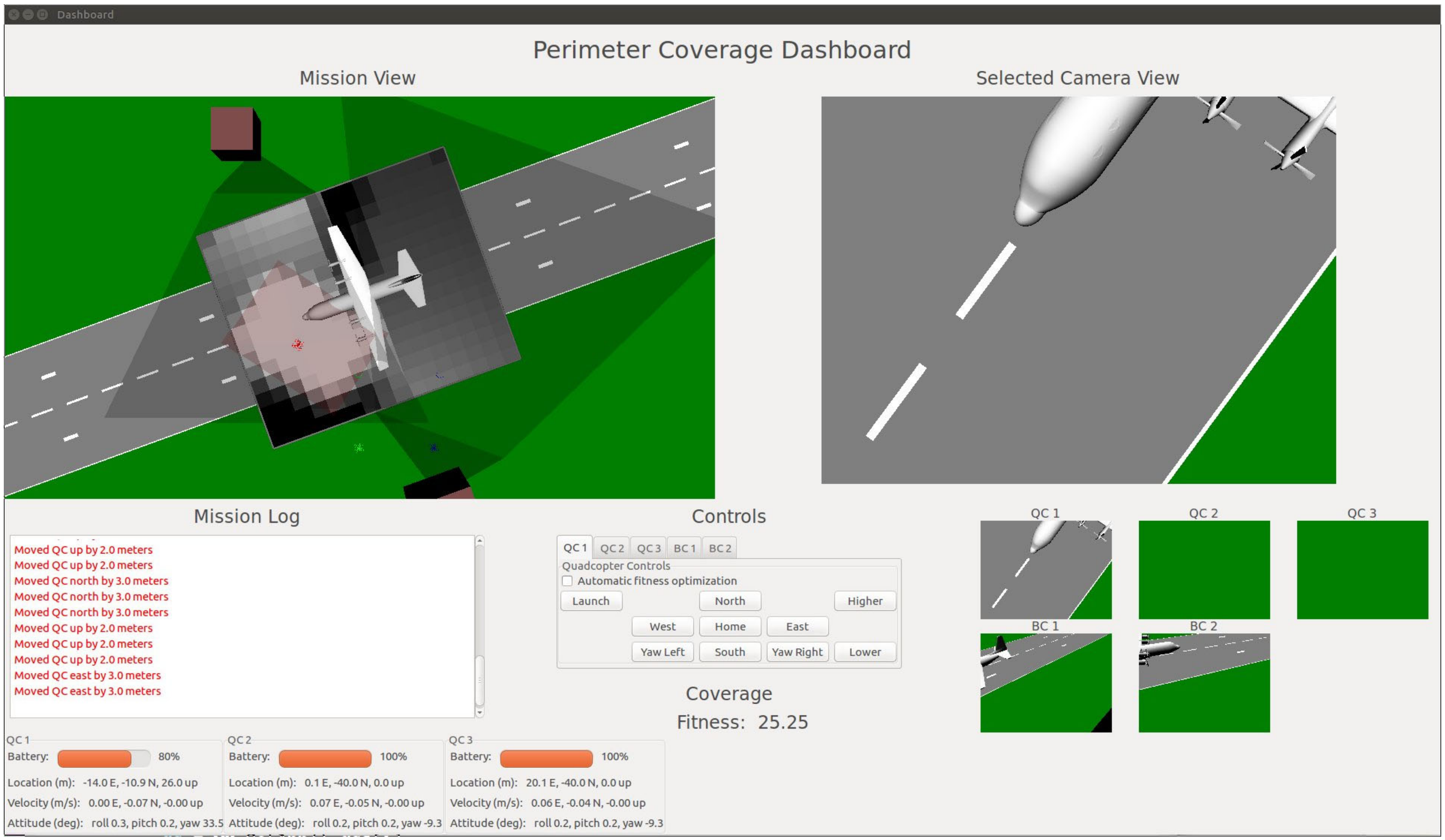
Copyright © 2014-2019 Azad M. Madni

Multi-UAV Dashboard Prototype

- Capabilities
  - customizable dashboard for monitoring and control of simulated or physical vehicles
- Underlying technologies
  - dronekit platform with visualization facilities
  - quadcopters (hardware) and quadcopter simulation models
  - quadcopter planning and decision-making model
  - quadcopter controller
- Key capabilities
  - simulated vehicles exhibit behavior of physical vehicle
  - same commands used to control vehicle models and physical vehicles (quadcopters)
  - can switch from simulated vehicles to physical vehicles

Dashboard Showing Camera View of Flying Quadcopter

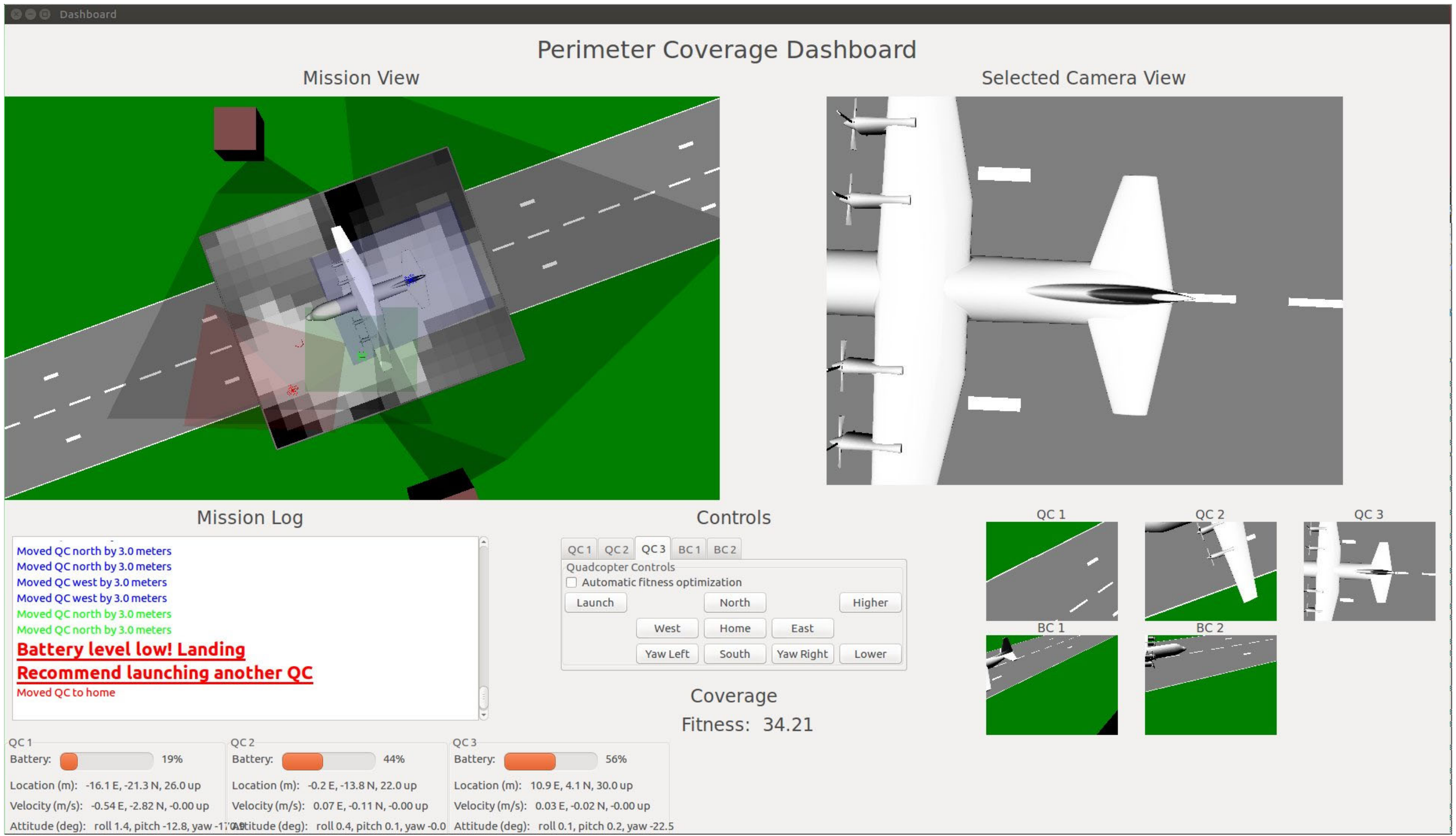
conference on systems engineering research



Copyright © 2014-2019 Azad M. Madni

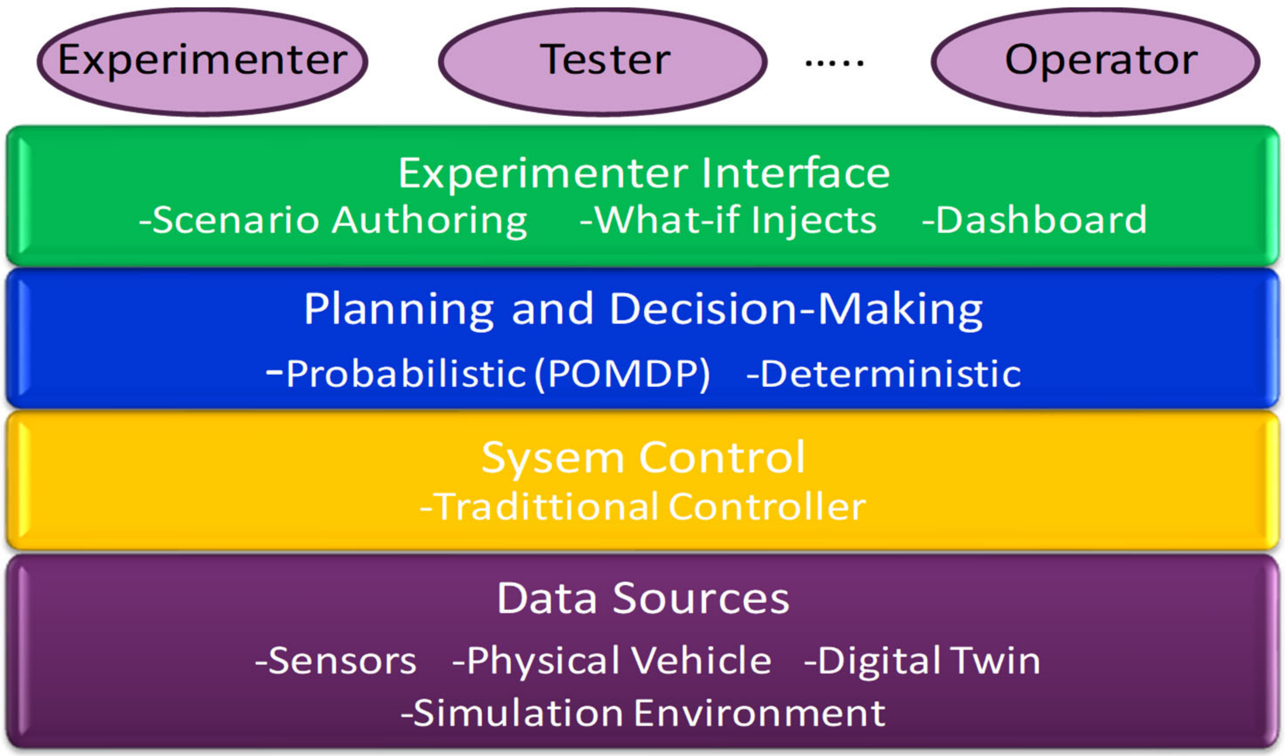
USC Viterbi
School of Engineering
Systems Architecting and Engineering

Dashboard Showing 3 Flying QCs With One Low on Battery and Landing



Copyright © 2014-2019 Azad M. Madni

Experimentation Testbed Architecture



Copyright © 2014-2019 Azad M. Madni

Prototype Testbed

conference on systems engineering research

- Quadcopters
  - driven by Raspberry Pi and Navio Flight Controller
  - full IMU: 3-axis accelerometers, rate gyros, magnetometer
  - inputs from laptop and/or remote controller
    - control values (throttle, roll-pitch-yaw)
    - perform autonomous flight
- Instrumented Testbed
  - layered architecture (UI, planning and DM, control, data sources)
  - customized Python scripts for vehicle control
    - dronekit framework and commands
  - semi-autonomous flights
    - launch, take-off, hover, and perform limited waypoint navigation
  - context-sensitive monitoring and control dashboard
    - monitor vehicle status and control vehicle
    - communicate with simulated vehicles and physical system

Prototype Testbed Hardware

Copyright © 2014-2019 Azad M. Madni

USC Viterbi School of Engineering
Systems Architecting and Engineering

Findings To-Date

- Key problem in implementing hybrid models
  - resolving mismatch between planning & decision-making layer and vehicle control layer
- 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 take into account when issuing commands
  - incorporate heuristics (e.g., priorities, region of influence) to resolve conflicts and simplify computation

Copyright © 2014-2019 Azad M. Madni

Findings To-Date (cont'd)

- 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 with goal of minimizing response time to action
    - ideal sampling period for POMDP determined experimentally
- Simultaneous creation of prototype and testbed - good strategy
  - introduced rigor in experimentation
  - compatible with introducing Digital Twin
  - currently: able to switch between simulation model and physical system
  - future: incorporate operational data from physical twin into Digital Twin
- Monitoring and execution dashboard – a key capability
  - facilitated understanding and debugging of vehicle behaviors

Copyright © 2014-2019 Azad M. Madni

USC Viterbi
School of Engineering
Systems Architecting and Engineering

Summary

- 21st Century Systems need to be safe, resilient and affordable
- Have to operate in uncertain, hostile and deceptive environment with partial observability
- System model verifiability is needed for system safety
- System model flexibility is needed for resilient behavior
- Such capabilities beyond traditional systems modeling capabilities
- Resilience Contract, a probabilistic approach based on POMDP, when coupled with heuristics and reinforcement learning, can satisfy safety, resilience and improved performance needs
- Our research is demonstrating viability of this approach with different CONOPS and resilience mechanisms

Copyright © 2014-2019 Azad M. Madni

Thank You

Copyright © 2014-2019 Azad M. Madni

USC Viterbi
School of Engineering
Systems Architecting and Engineering