START:UAVs — Software Techniques for Automated Resiliency and Trustworthiness in Uncrewed Aerial Vehicles

Kevin Angstadt† Jonathan Dorn† Kevin Leach† Aaron Paulos‡ Westley Weimer†

†University of Virginia ‡Raytheon BBN Technologies

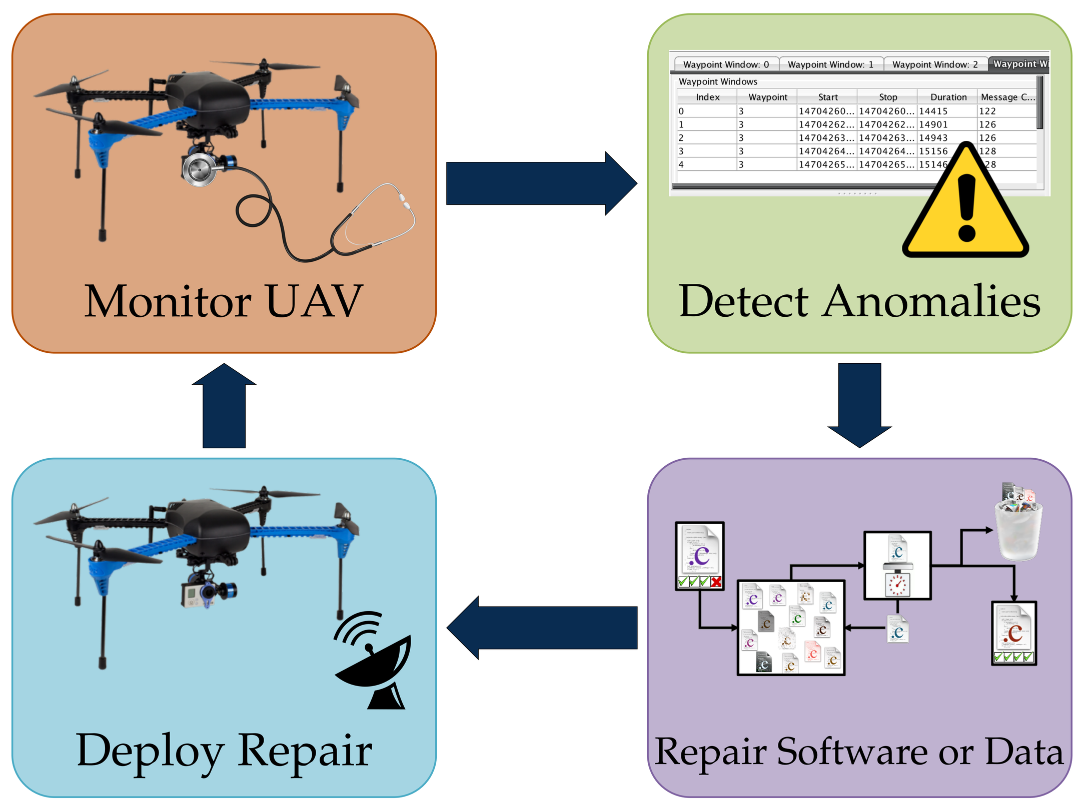
{angstadt, dorn, leach, weimer}@virginia.edu apaulos@bbn.com

University of Virginia
Department of Computer Science
Charlottesville, VA 22904

Problem

- Autonomous vehicles often operate in harsh environments
  - Adversaries may attempt to prevent successful mission completion
  - Communication with human operators may be infrequent or delayed
- Vehicles need to be *resilient* to unexpected software glitches and environmental factors
- How can we ensure that the human operator *trusts* the changes made to software to overcome glitches and the environment?

Our Approach: START

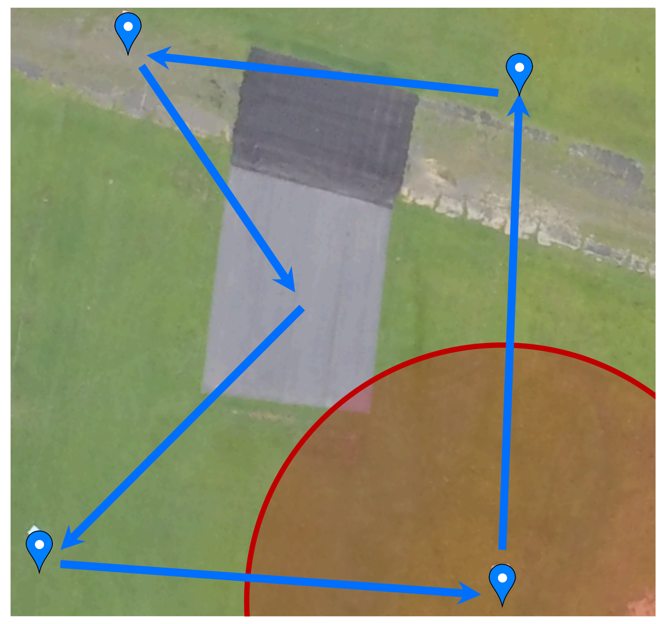


Case Study: MAVLink Vulnerability

- MAVLink is a lightweight communication protocol used by many small, uncrewed vehicles
  - Vehicle continuously broadcasts telemetry (sensor) data
  - Ground control station sends mission commands to vehicle, which are then executed
- Protocol is ***unencrypted***: Anyone can communicate with the vehicle
- Attacker can take control of vehicle using commodity hardware costing $25

Attack Scenario

- Uncrewed Aerial Vehicle (UAV) flies surveillance mission to observe fixed points on ground
- Attacker with directional antenna can communicate with UAV for some portion of flight path
  - Attacker attempts a *stealthy* attack to keep UAV from observing part of the surveillance region
  - Example: skip target, but don't crash
  - "Wind blew the UAV off course"



Ground Control

- Communicates with UAV via MAVLink
- Translates high-level commands from START system to MAVLink messages

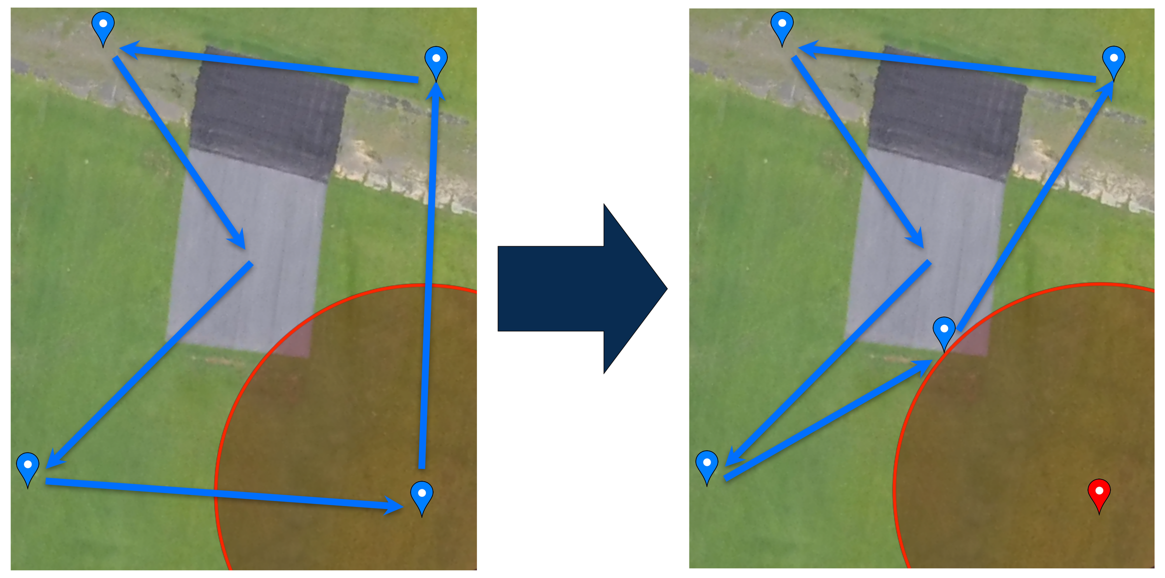
Repair Controller

- Deploys initial mission
- Develops new mission in response to attack
- Orchestrates mission deployment

Trust Assessment

- Compares telemetry to mission parameters
- Detects attack when telemetry diverges from expected values
- Validates new mission

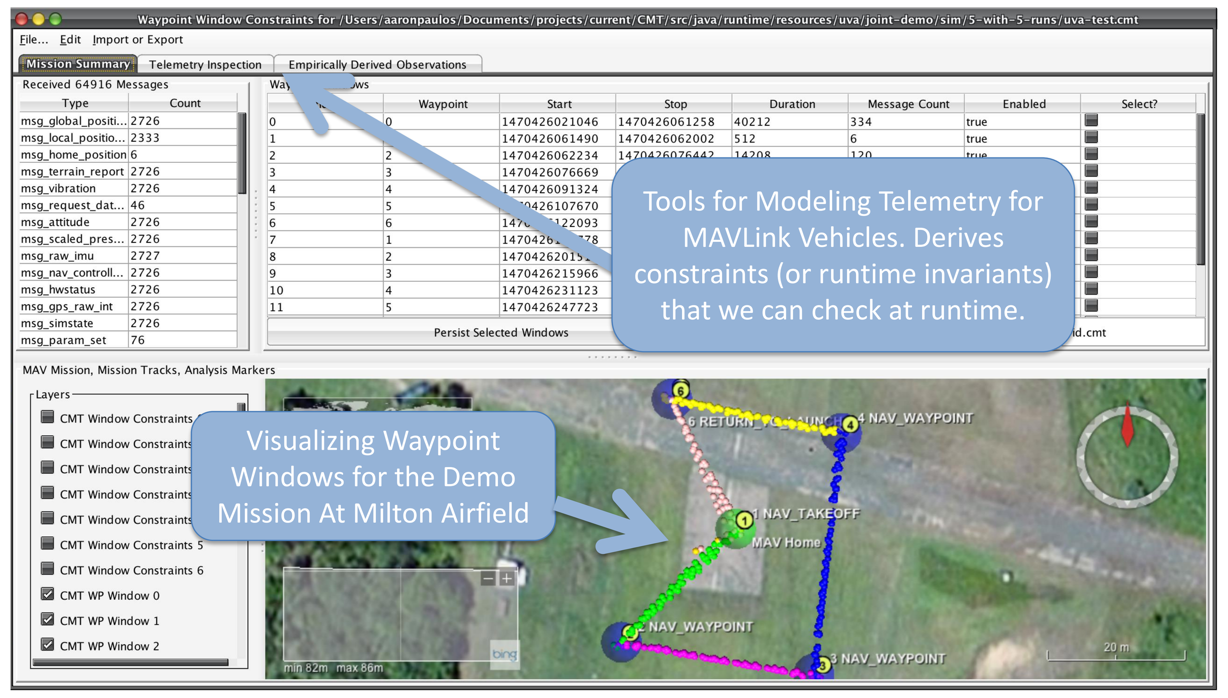
Flight Path Repair



Pre-Repair Post-Repair

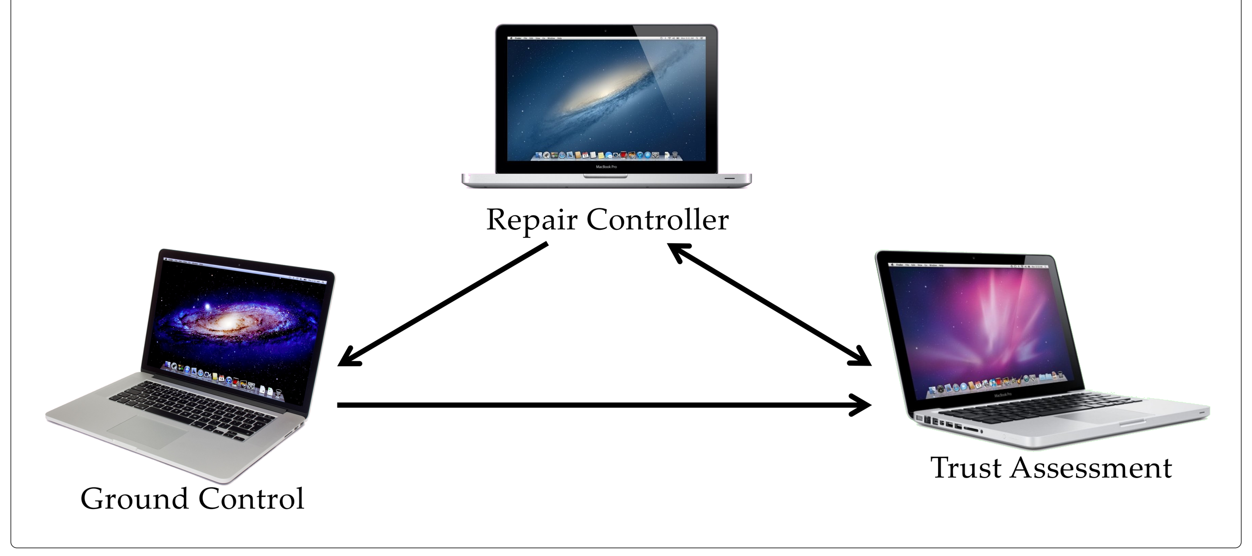
- Use *genetic algorithm* to generate new mission
  - Based on the notions of *mutation*, *crossover*, and *fitness* from evolutionary biology
- Favor solutions that:
  - Avoid the attack zone
  - Prefer fewer extra waypoints
  - Retain order of observation points
  - Get close to observation points
  - Have shorter paths

Continuous Measurement of Trust



- Collect telemetry data from known good missions (simulation and/or real flights)
- Build model from telemetry data
- Compare live telemetry data against model
  - Trust decreases as live mission diverges from model
  - Significant divergence from model triggers warning that attack is likely occurring

START System Components



Milton Field

All testing for this project was conducted at Milton Field, a WWII-era airfield owned by UVA

Results

- Air Force demonstration in August 2016
- System successfully detected attacks using MAVLink vulnerability
- Repaired missions avoided attack regions while still filming observation points

Future Directions

- Detect and repair faults in UAV control software
- Additional measurements of trust

Industry and Academic Collaborators

Raytheon BBN Technologies | Carnegie Mellon University | UCLA | The University of New Mexico