Persistent Surveillance with a Fleet of Drones

Patrick Washington phw@stanford.edu

Motivation
- Surveillance with drone fleet
- Drones have limited battery life
- Must periodically charge
- Need policy to make decisions

Applications
- Monitor forests, military bases
- Modify for taxis, warehouses
- Related to energy/load balancing

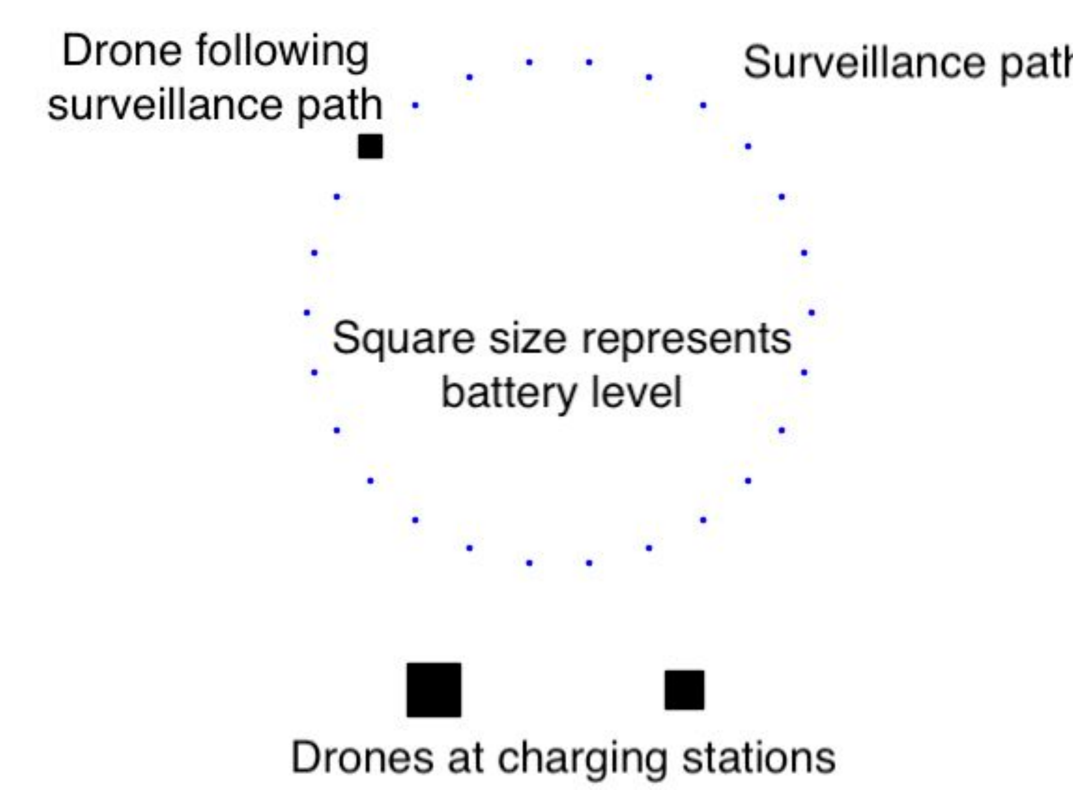
Scan for a sample simulation video

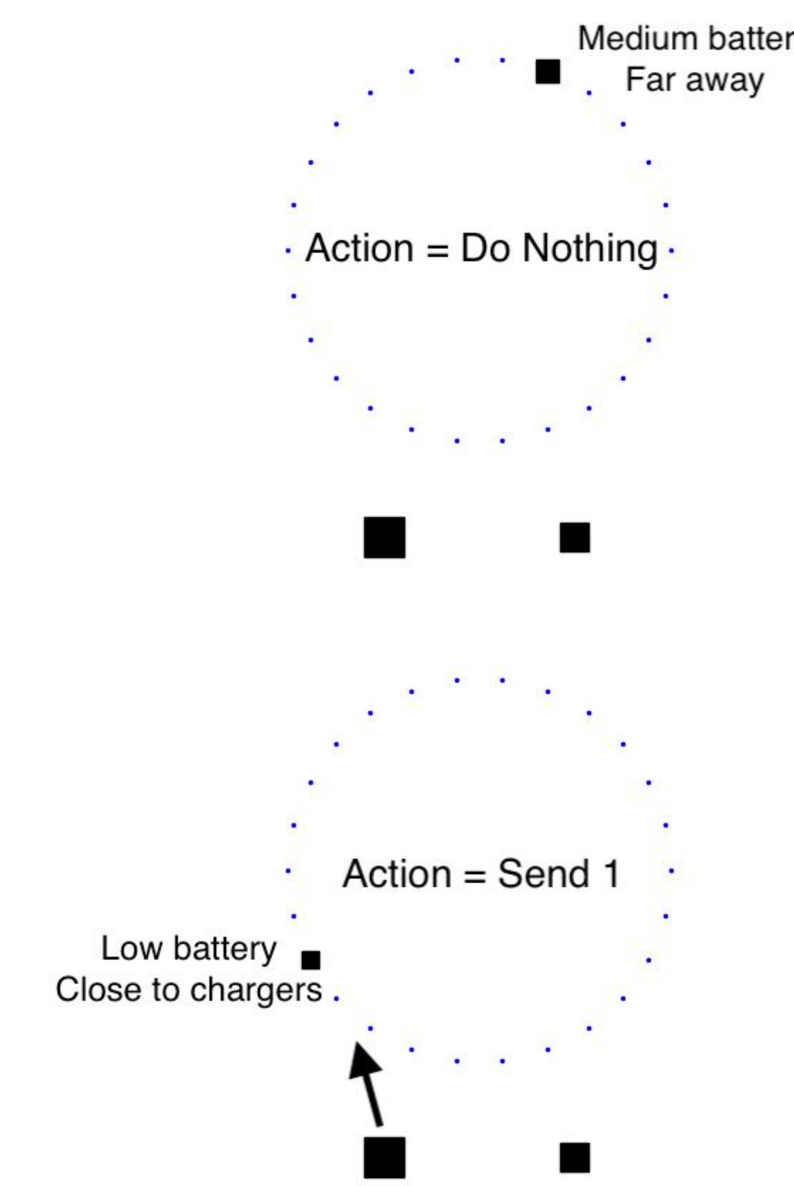
Problem Setup
- Batteries and positions evolve stochastically
- Surveillance station moves around a path
- Charging stations fixed on the ground
- Always at least one drone at surveillance

Challenges
- Size of state-action space scales poorly
- Previous approaches intractable
- States are cyclic, end state can be unclear

Approach
- Value iteration (VI) from full batteries
- States are battery levels and path index
- Monte Carlo for travel transitions
- Monte Carlo helps reduce state space
- Reward life, penalize death and replacement

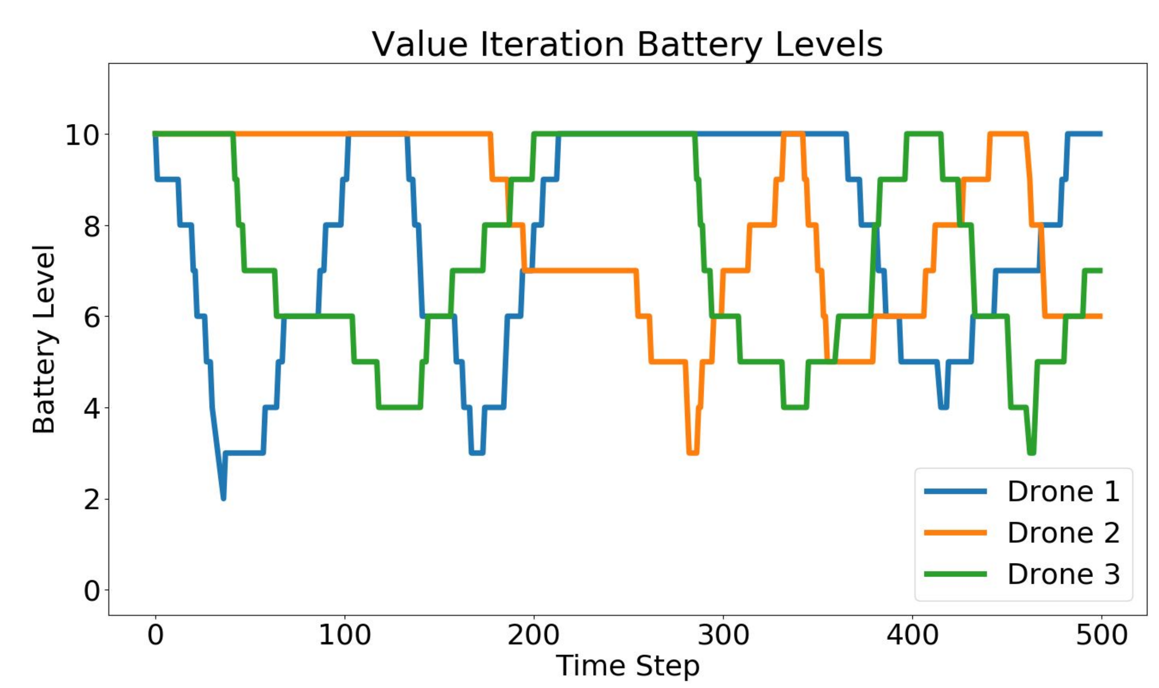
Input-Output Model
- Input: battery levels and path index
- Output: choose a replacement or do nothing

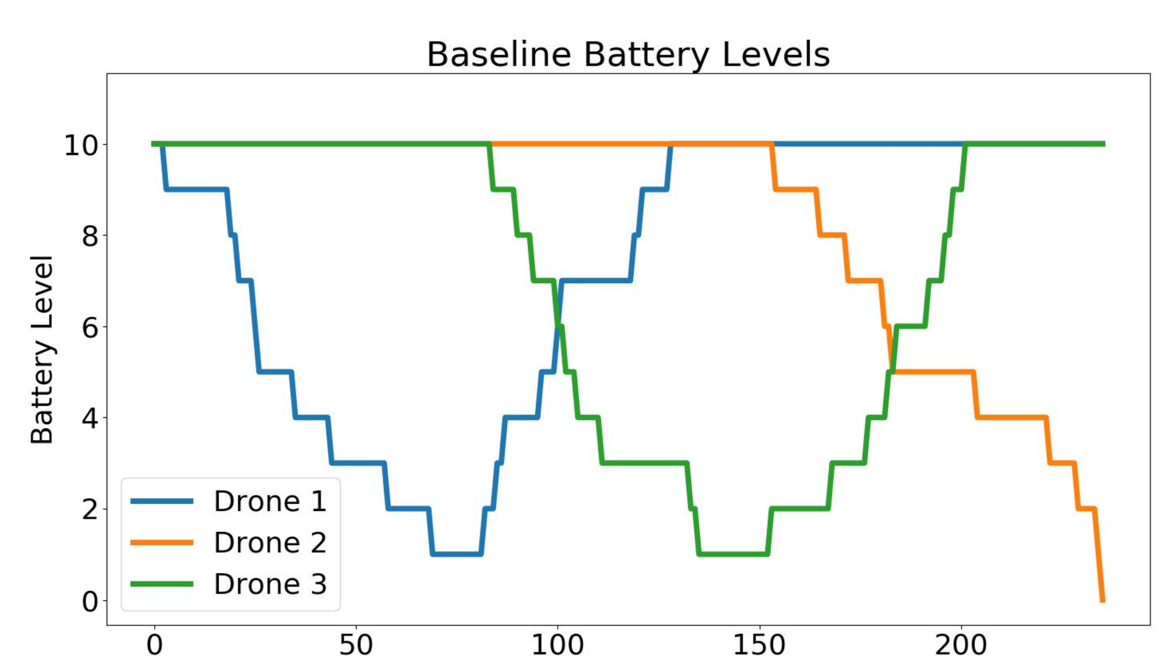


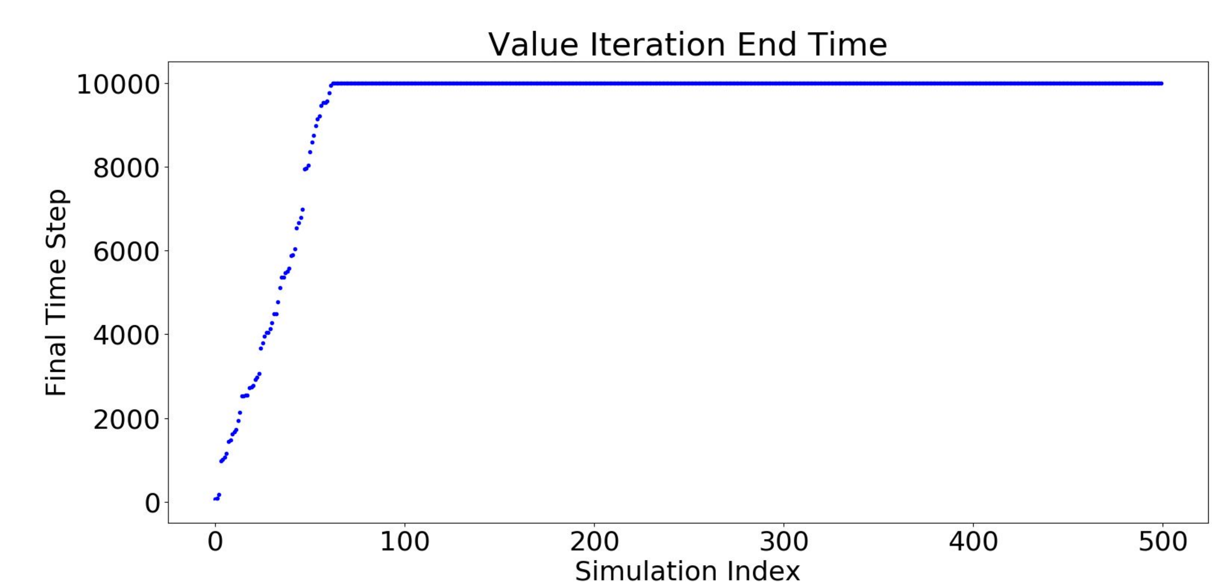


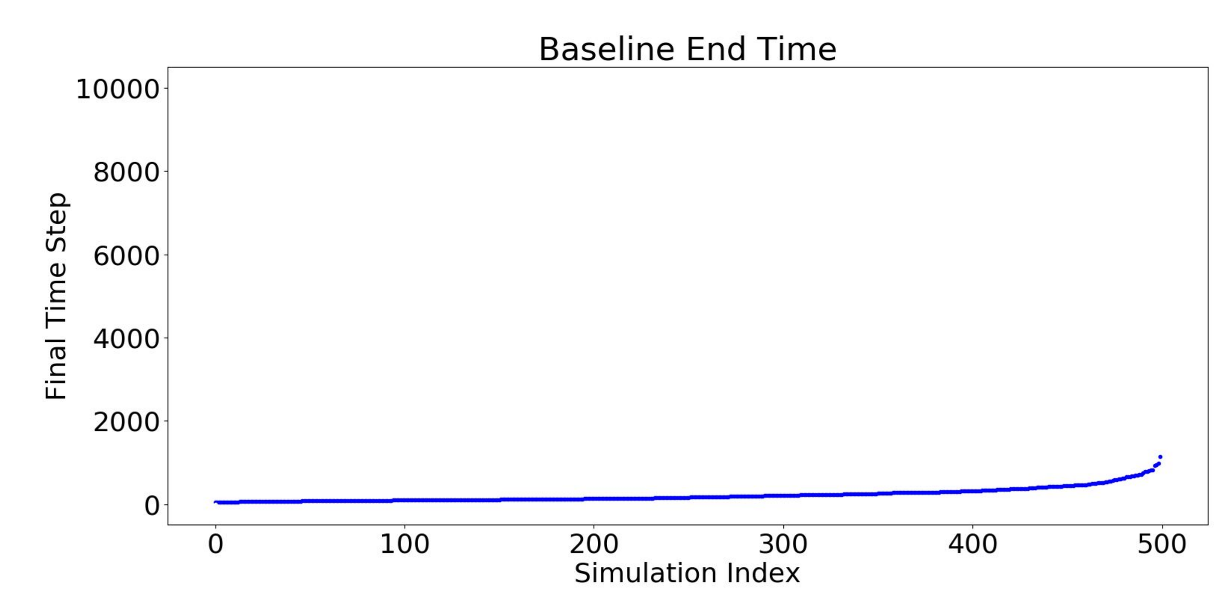
Results
- Simulated to 10,000 time steps
- Averaged over 500 simulations
- VI: 9355 steps until dead
- VI: reached limit in 87.6%
- Baseline: 219.6 steps until dead
- Baseline: reached limit in 0%

Baseline
- Find expected replacement time
- Find expected flight time left
- Replace when times are close
- Small threshold added for safety









Analysis
- VI policy avoids low charge
- Can fail after fast battery drain
- Caution allows room for error
- Death penalty affects risk-taking
- Can lead to extra replacements

Future Steps
- Policy does not generalize
- Solve for each set of parameters
- Use a function approximator
- Initialize with sample VI solutions
- Explore with Q learning or similar

- Extend to harder problems
- Multiple surveillance paths
- Autonomous taxi service
- Operate warehouse or factory