IOWA STATE UNIVERSITY

An Efficient Algorithm for Multiple-Pursuer-Multiple-Evader Pursuit/Evasion Game

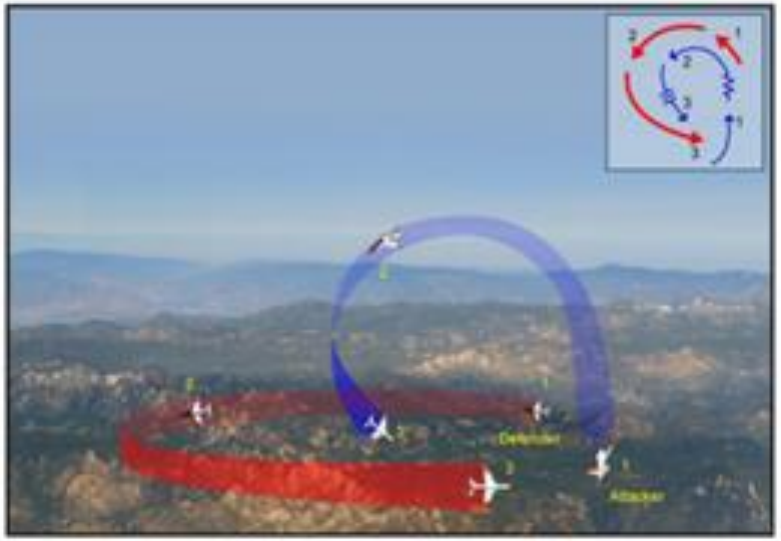
Josh Bertram bertram1@iastate.edu

Peng Wei pwei@gwu.edu

AIAA SciTech 2021

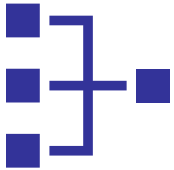
Introduction

- Pursuit Evasion problem: dogfighting
- Good challenge problem
- Good approaches exist for small teams (1v1, 2v2)
- Not many approaches for large scale encounters (4v4, 10v10)

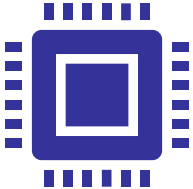


Example high yo-yo encounter from CNATRA training manual (publicly available)

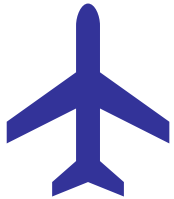
Our Approach



Formulate as Markov Decision Process (MDP)

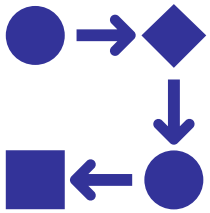


Use FastMDP algorithm to solve the MDP quickly (milliseconds)



Use forward projection to link aircraft dynamics

Markov Decision Processes (MDPs)



Framework for sequential decision making

Discrete time control system with feedback

Solution is a control law known as optimal policy

From Artificial Intelligence (AI) community

Used by ACAS-X (Partially Observable MDP)

Powerful, general approach

But, difficult to make tractable / efficient

Typical approaches are slow to solve MDP

But once solved, fast run-time execution

ACAS-X generates lookup table ~500MB - 1GB, for example

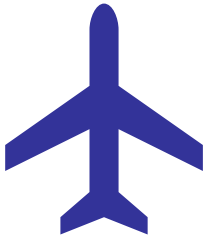
FastMDP Algorithm

Solves a "useful subclass" of MDPs in milliseconds

Applicable to aerospace and robotics problems

Adapts to rapidly changing environments

No training phase; solves on-line in real-time



Needs aircraft dynamics model and a Reward Function

Reward function controls behavior of agent

Provides positive and negative rewards

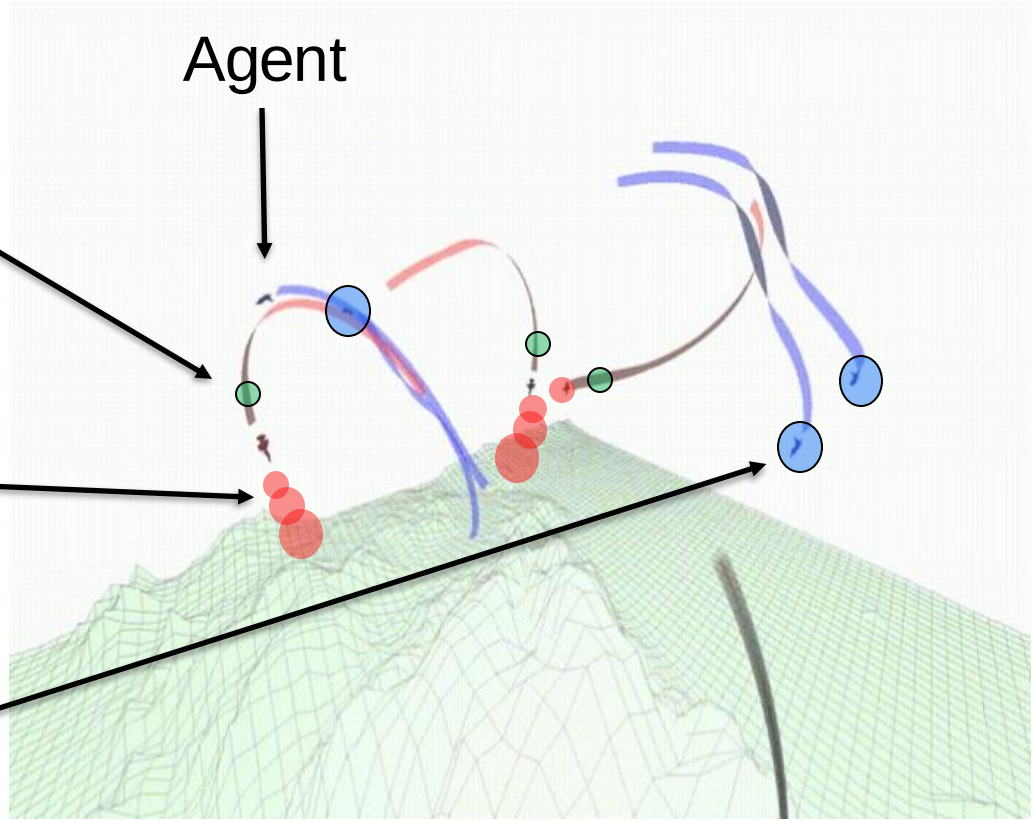
Built from "reward primitives"

Reward Function

Positive reward placed behind

Negative reward placed in front of enemy

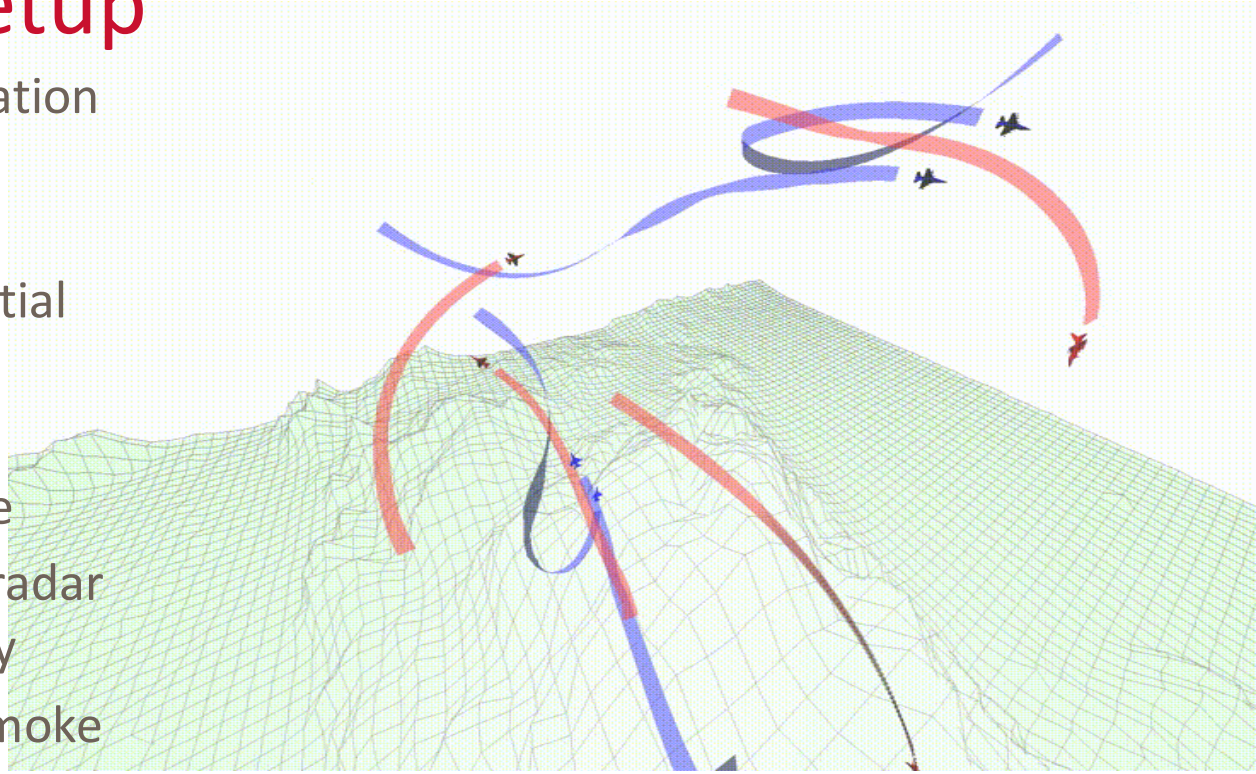
Negative reward placed around teammates



FastMDP Control Flow

Experimental Setup

- Simulation and visualization environment
- Monte Carlo testing of different team sizes, initial conditions, parameters
- Blue team has slight performance advantage
- Simple mechanism for radar lock, downing an enemy
- Hard deck, collisions, smoke trails, etc



Pursuit-Evasion Performance

Metrics:

- Probability of Win
  - Probability team wins encounter
- Probability of Survivability
  - Measure of number of team members left after encounter

Intuition:

- Both metrics should be high
- Blue should always win
- Should degrade as airspace is congested

Table 4 Probability of win P_{win} and Probability of survivability P_s of blue team as team size increases over 10 runs

Team Size	P_{win}	P_s
1v1	100%	100%
2v2	100%	100%
3v3	100%	100%
4v4	100%	100%
10v10	100%	99%
100v100	100%	97%

Computational Performance

Measures average time for each agent to perform one "cycle" of FastMDP:

- Formulate an MDP
- Solve MDP
- Forward project actions
- Select best action

Target performance:

- 100 ms or less (> 10 Hz)
- Good performance even up to 100 vs 100
- Laptop CPU hardware, Python implementation

Table 5 Processing time required for each agent on red or blue team as team size increases over 10 runs

Team Size	Mean (ms)
1v1	2.26
2v2	2.50
3v3	2.70
4v4	3.16
10v10	5.55
100v100	27.59

Conclusions

FastMDP algorithm scales well to 100 vs 100 encounters, shown to also perform well in this simulation.

Computationally efficient way to model complex, dynamic problems.

Can achieve high frame rates, suitable for real-time embedded applications.

Future work: Compare to other algorithms (ProNav, swarm algorithms), incorporate uncertainty.

Questions?

Table 6 Links to videos

Team Size	URL
1v1	https://youtu.be/zGWXxtJUwk8
2v2	https://youtu.be/Q9O50cqpVtA
3v3	https://youtu.be/6Zok4sj43C4
4v4	https://youtu.be/qhI6av3oJN4
10v10	https://youtu.be/6twTWNRurwo