Landing Throttleable Hybrid Rockets with Hierarchical Reinforcement Learning in a Simulated Environment

Project Team:
Francesco Alessandro Stefano Mikulis-Borsoi
Advisor: Dr. Marek Petrik
Co-Advisor: Paul Gesel

University of New Hampshire
College of Engineering and Physical Sciences

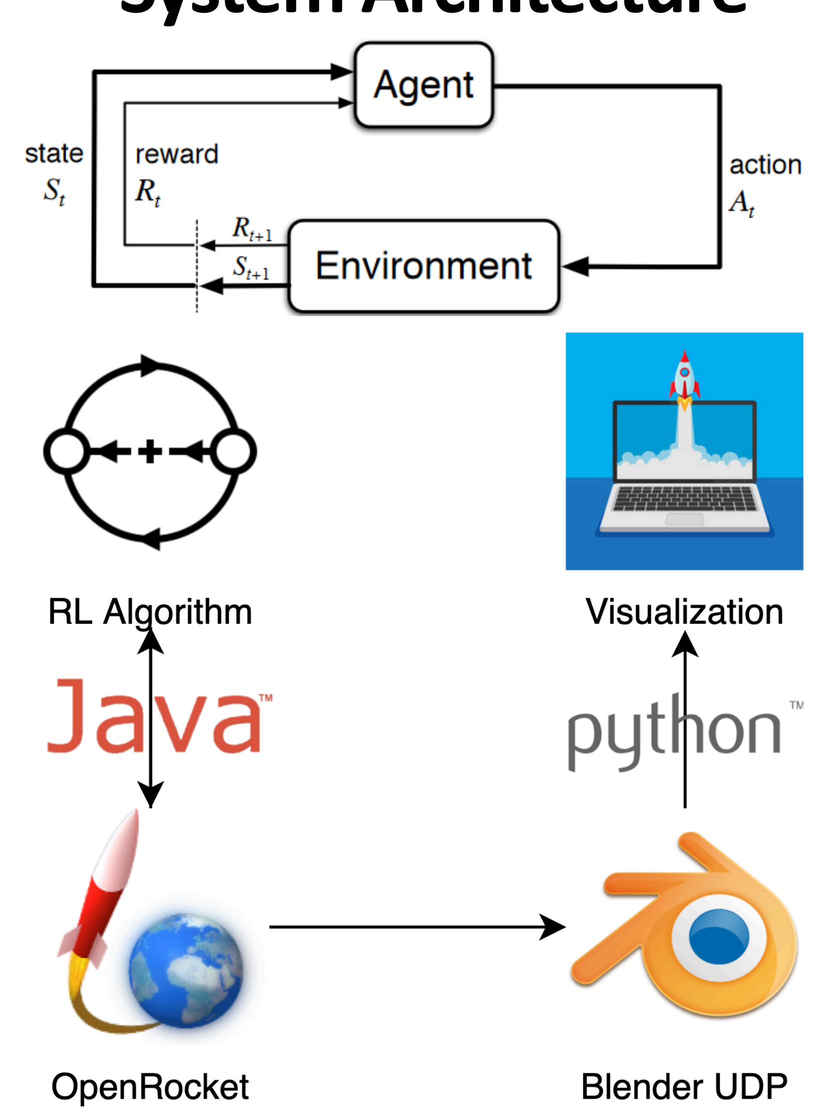
Introduction

- Markov Decision Processes model decision making in discrete, stochastic, sequential environments; their next state is independent of the past state given the present state
- Hierarchical reinforcement learning involves layers of MDPs, maintaining interpretability
- Difficult and interesting problem: continuous and partially observable state-space, non-linear dynamics and requirement of real-time control

Objectives

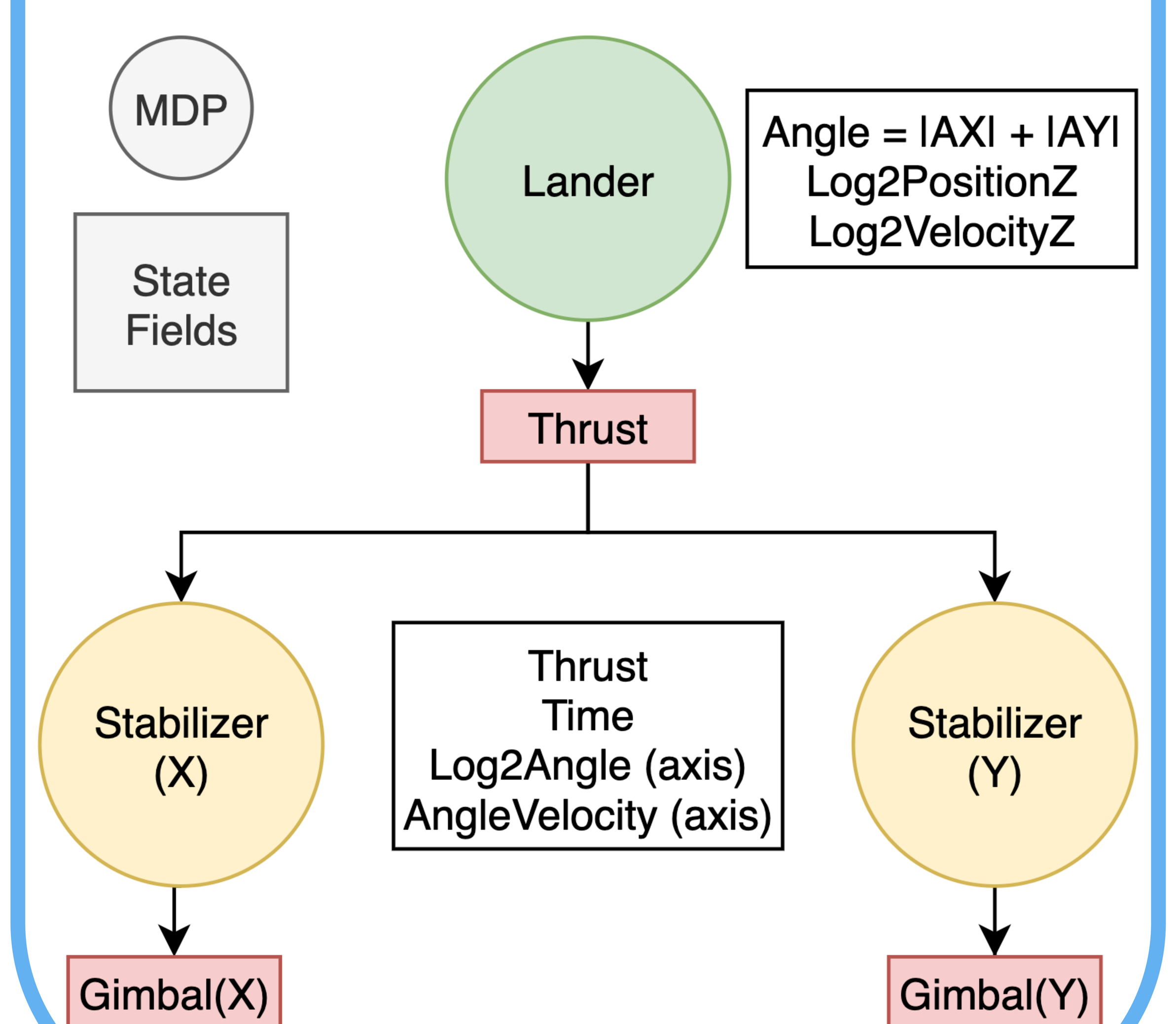
- Land a throttleable hybrid rocket in 3D with thrust vectoring from varying initial states
- The controller must execute with constrained CPU, RAM and imprecise sensor data
- Integrate 3D visualization for visual verification
- Land the rocket with **vertical velocity < 2 m/s** and **zenith angle < 8°** from **varying initial states** around 30 m at -10 m/s, within 8 seconds

System Architecture

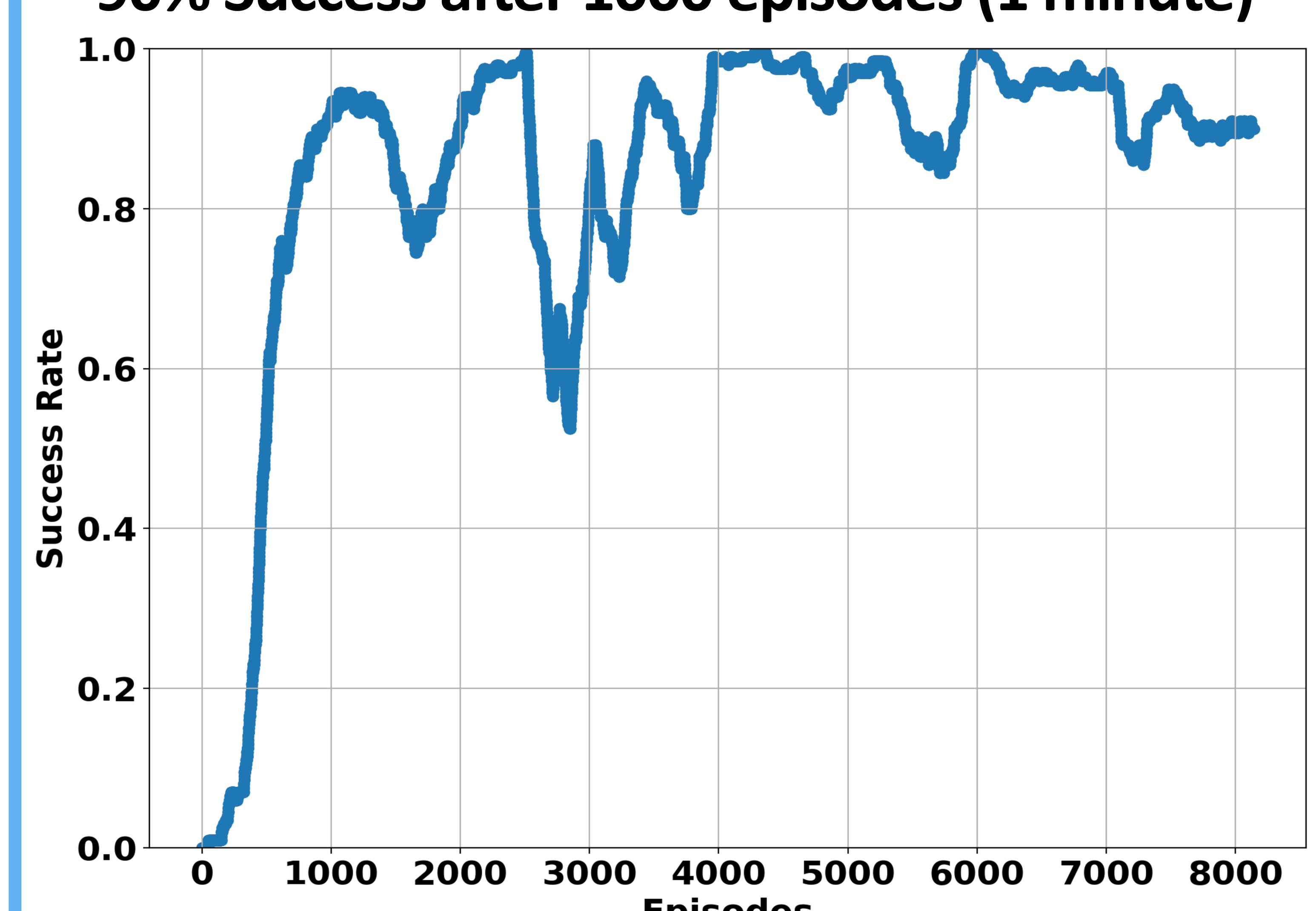


Hierarchical Structure

- The Lander MDP selects the thrust, feeding it to the state definition of the stabilizer.
- The stabilizer is an MDP whose inputs and outputs are axis independent. Ensuring this type of symmetry effectively reduces the size of state space by the square root (otherwise requiring each state field to have X and Y axis)



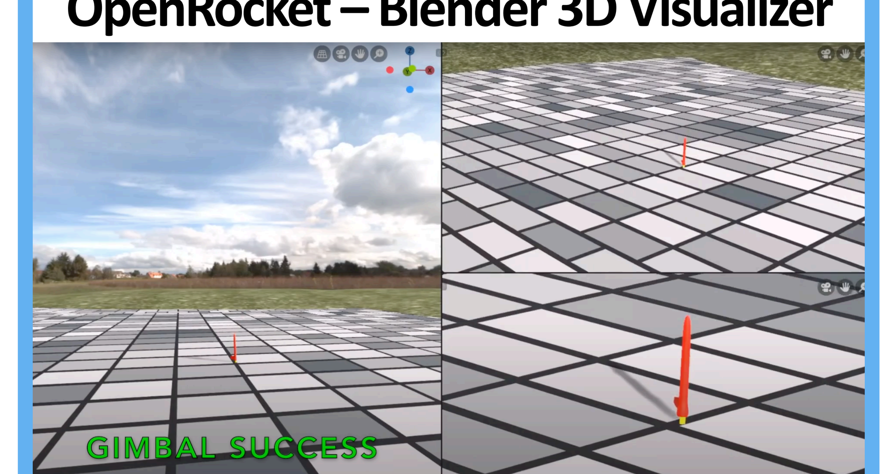
90% Success after 1000 episodes (1 minute)



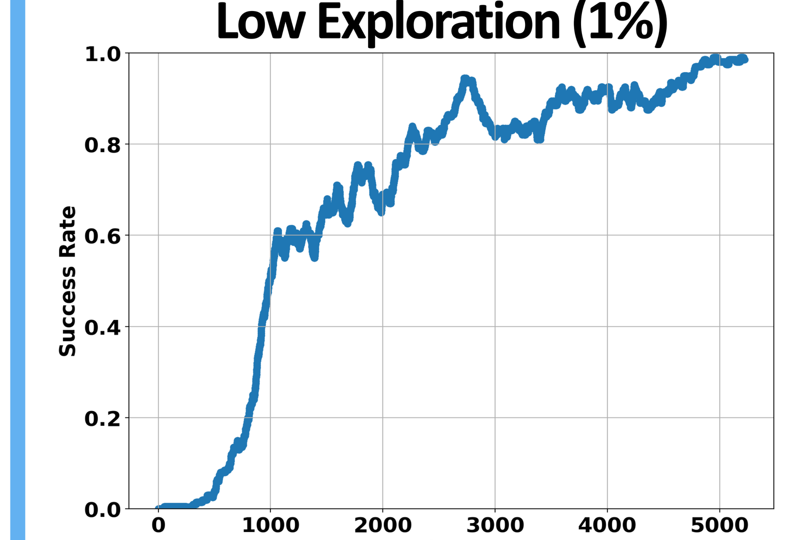
RL Contribution

- Developed a **standardized RL framework** for OpenRocket, which should encourage the community to test different MDP formulations, and specific reward functions
- **MDPs are defined with a schema** (with fields such as "stateDefinition"), and **custom formulas** can be used to calculate complex state fields, parsed with recursive descent
- Software development with extensibility in mind led to the creation of **common interfaces for different learning RL methods** (implemented MC, TD-0, SARSA) - where the rewards can be specified in the schema
- Crucial modifications to the source code allow for **plotting** the state and action **fields** of the custom MDPs (with **MDP-specific discretization**)
- The **non-hierarchical** version of this discretization definition **succeeded < 10%** of the time, even after an order of magnitude more training compared to the hierarchical problem formulation!

OpenRocket – Blender 3D Visualizer



Low Exploration (1%)



3D Visualizer

- Integrated OpenRocket with Blender via Python server leveraging UDP (local network capability)
- Created OpenRocket visualization extension
- Ability to replay scenes and view simulations

The Approach

- Demonstrates that hierarchical MDPs are effective for reducing the state space
- Leverages different discretization of the state space based on specific needs for the MDPs
- Lander: MC (terminalReward = -|velocityZ|)
  Stabilizer: TD-0 (reward = - angle^2 + 1)
- Splitting the MDPs allows for specific reward functions, with a meaningful single objectives

Current Status

- Over 169 commits to the fork of OpenRocket
- Over 10k lines of contributed code
- Developing an expansion to include a Reacher MDP that will guide the rocket to the landing pad

Next Steps

- Continue researching the implications of hierarchical RL in complex decision problems
- Develop a toolbox unifying the OpenRocket RL with the OpenAI-Gym framework
- Continue extending the customizations in OpenRocket to finalize the framework
- Release this problem as an RL benchmark