Test and Evaluation of Reinforcement Learning Via Robustness Testing and Explainable AI

Dr. Ali K. Raz
Assistant Professor
System Engineering and Operations Research
George Mason University
araz@gmu.edu

Dr. Kyle Williams
Autonomous Sensing and Control
Sandia National Labs
kwilli2@sandia.gov

Dr. Kris Ezra
Research Scientist
Center for Integrated Systems in Aerospace
Purdue University
kris@purdue.edu

Approved for Public Release – SAND2021-13202 C

Acknowledgement

This work was supported by the Laboratory Directed Research and Development program at Sandia National Laboratories, a multi-mission laboratory managed and operated by the National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This work describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

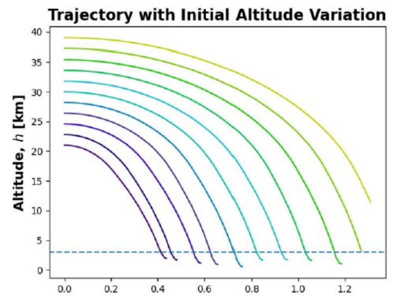
Sandia National Laboratories

Motivation and Contributions

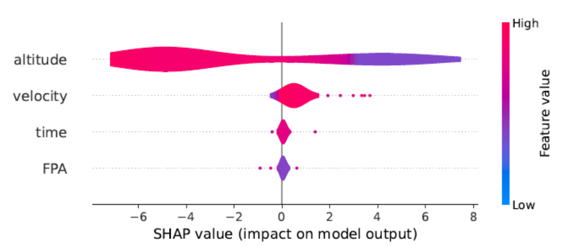
- **Motivation:**
  - Reinforcement Learning (RL) provides the ability to train an artificial intelligence (AI) agent to operate in dynamic uncertain environments
  - Impressive performance outcomes to learn nearly-optimal solutions in a variety of application domains
  - Limited testing and characterization of performance bounds of RL solutions
    - Impedes transition to real time systems
- **Contributions of this work:**
  - Develop a comprehensive Test and Evaluation Framework for RL
    - ❑ Robustness Testing of RL solutions
    - ❑ Understanding of RL decision making via Explainable AI
    - ❑ Validation of RL solutions
  - Demonstrate application of RL to high-speed aerospace vehicle mission
    - ❑ Investigate uncertanity in flights parameters such as angle of attack, velocity, altitude, and flight path angle

Test & Evaluation Framework for Reinforcement Learning

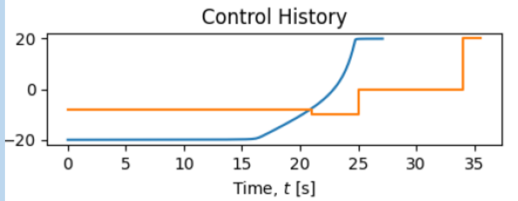
Robustness Testing



Explainable AI



Validation with known solutions

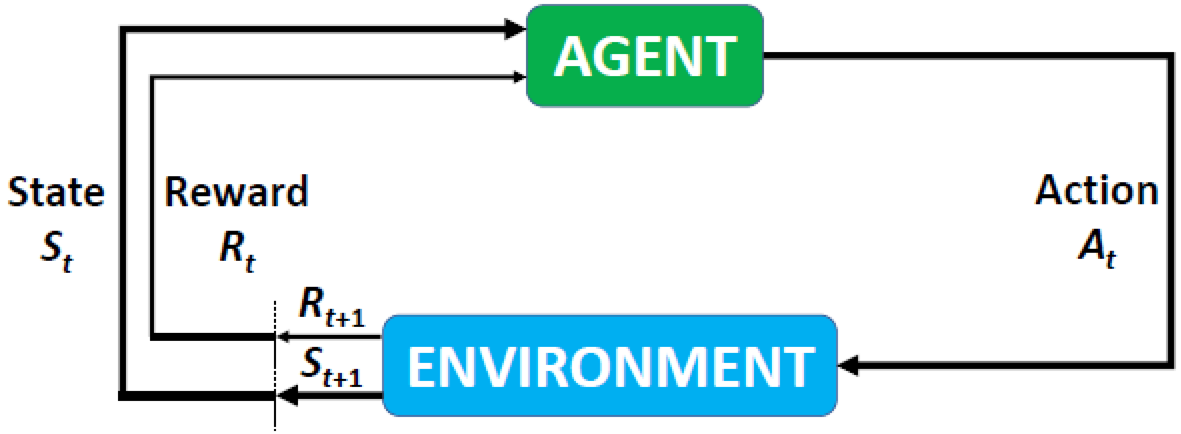


Brief Introduction to Reinforcement Learning

- What is RL?
  - A methodology to allow an agent to learn what actions to take in dynamic and uncertain environments and learn the optimal behavior
  - RL interacts with the simulation environment to achieve pre-defined goals
    - Achieving goals is rewarded
    - Learning occurs from exploration of environment and exploitation of reward

- Pieces of an RL problem:
  - State s_t of the environment
  - Actions, $a_t \in A$
  - Reward, $r_{t+1}(s_t, a_t)$ for action a_t at s_t
  - Policy, $\pi_t(s, a)$
    - Selecting action a_t at state s_t
    - Deterministic or Stochastic
  - Implemented via RL algorithms

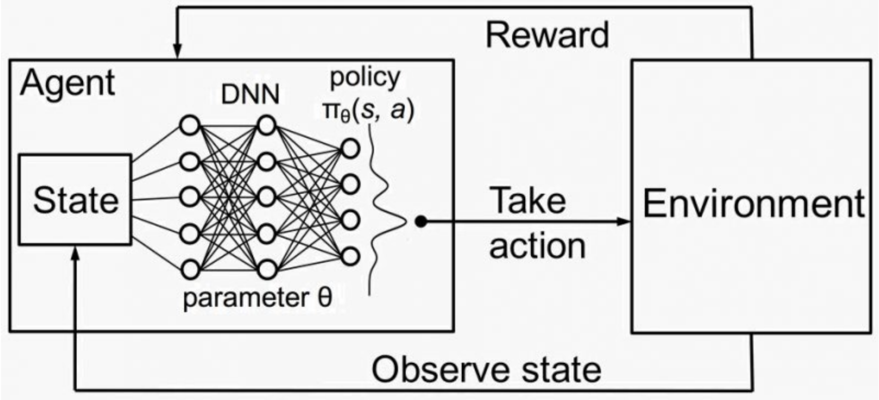
Reinforcement Learning



The Need for RL Test and Evaluation (T&E)

- Once trained, RL agent is essentially a Deep Neural Network (DNN)
  - ✓ Well-established performance outcomes
  - ❑ Limited characterization of performance bounds due to variations and uncertanities
  - ❑ Limited explanation of black-box decision-making logic

- Status-Quo of RL Testing:
  - ✓ Strong focus on RL implementation and comparing learning policies in different application domains
  - ✓ Selective demonstration of test cases, mostly based on Monte Carlo simulation and user selected variations
  - ❑ Limited evaluation of acceptable and unacceptable performance regions



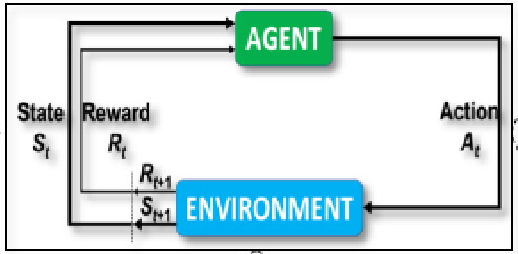
SE4AI
Example T&E Questions to Ask

- ❑ What is impact of variations in environment, observed states and action space on the RL performance?
- ❑ How does the input (i.e., observed state) influence RL decision making?
- ❑ How does RL respond to modeled (i.e., incl. in training) and unmodeled uncertanities?
- ❑ How does the array of RL solutions compare to other accepted solutions?

PROPOSED THREE PART T&E FRAMEWORK FOR RL

Robustness Testing

Purpose:
Sensitivity analysis of variations in action space, enviorment, and state observation
Methodology:
Design of Experiments and Statistical Analysis
Value:
Performance bounds and characterization of uncertanities



Compare to Known Solutions

Purpose:
Evaluate RL performance to known and accepted solutions
Methodology:
Problem space dependent; closed form mathematical solutions.
Value:
Validate RL performance and robustness testing results

Explainable AI (XAI)

Purpose:
Determine influential features of trained RL decision-making logic
Methodology:
Post-hoc XAI method: Shapely Additive Explanations
Value:
Explain which state vector values contribute to RL decision and why sensitivities are present in robustness test

Remainder of this briefing:

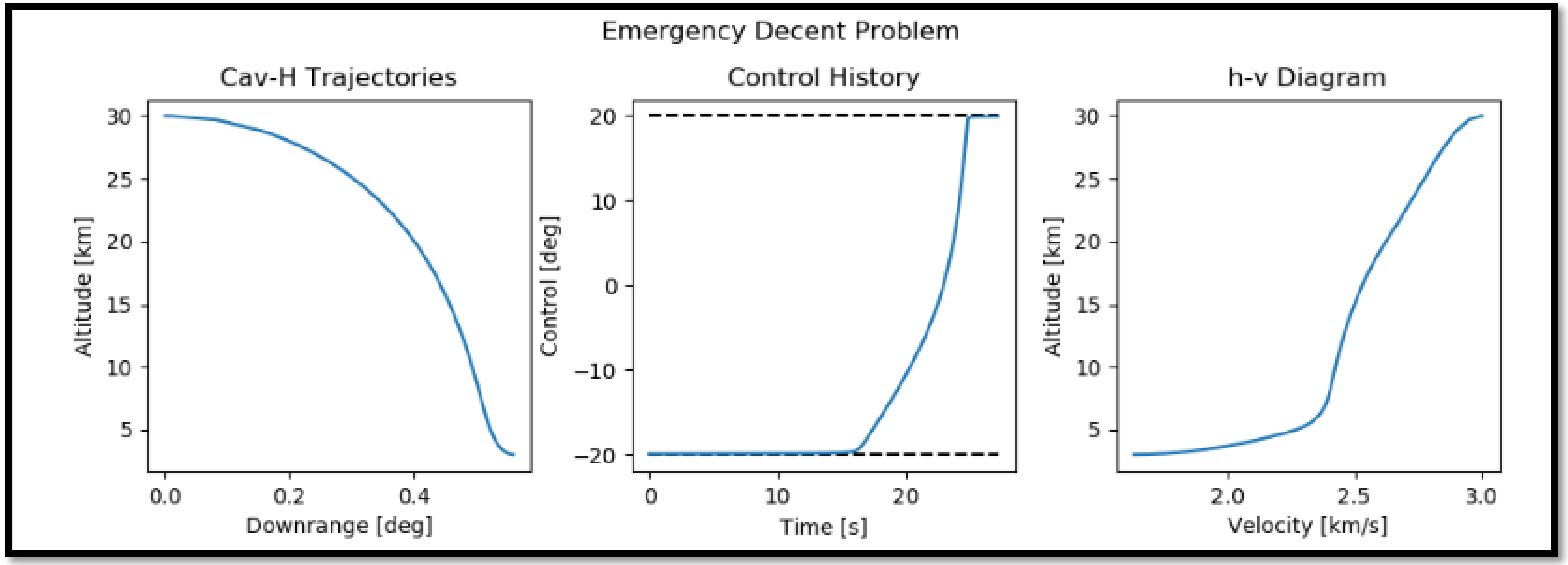
1. Formulate a high-speed aerospace mission suited for RL application
2. Apply the T&E Framework for analysis of RL-solution

Vehicle Model Parameters

- **States:**
  h: altitude, θ: downrange angle,
  v: velocity, γ: flight path angle
- **Control:** α: angle of attack
- **Dynamics:**
- $$\dot{x} = \begin{bmatrix} \dot{h} \\ \dot{\theta} \\ \dot{v} \\ \dot{\gamma} \end{bmatrix} = \begin{bmatrix} v \sin\gamma \\ \frac{v}{r}\cos\gamma \\ -\frac{D(\alpha)}{m} - \frac{\mu}{r^2}\sin\gamma \\ \frac{L(\alpha)}{mv} - \left(\frac{v}{r} - \frac{\mu}{vr^2}\right)\cos\gamma \end{bmatrix}$$
- **Objective:** $J = \min t_f = \int_0^{t_f} dt$
- **Initial Constraints:**
  $$\Psi_0 = 0 = \begin{bmatrix} h - 30\text{ km} \\ \theta \\ v - 3\text{ km/s} \\ \gamma \end{bmatrix}_{t=t_0}$$
- **Path Constraint:**
  $$|\alpha| \leq 20°$$
- **Terminal Constraints:**
  $$\Psi_f = 0 = \begin{bmatrix} h - 3\text{ km} \\ \gamma \end{bmatrix}_{t=t_f}$$

Emergency Descent Problem for an Untrusted High-Speed Vehicle

- The vehicle is at 30 km altitude and 3 km/s velocity needs to descend to level flight at a safe altitude (3 km) in minimum time
- Constraints must be satisfied at all times



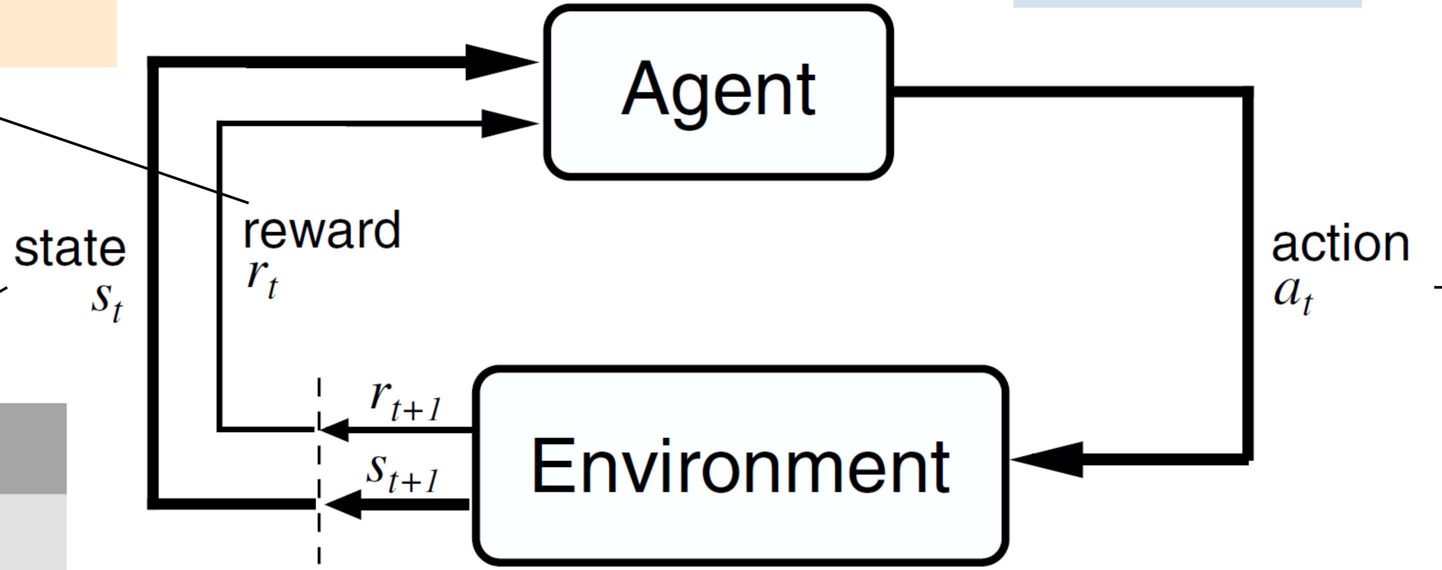
Reinforcement Learning Problem Formulation

Reward Function

- Designed to teach the RL agent an emergency descent problem
- Reward structure based on distance to target and FPA

RL Agent

- RL trained from SB3 Python package
- PPO
- RL training parameters (backup)



State

- Distance to target
- Altitude
- Velocity
- AoA
- FPA

Environment

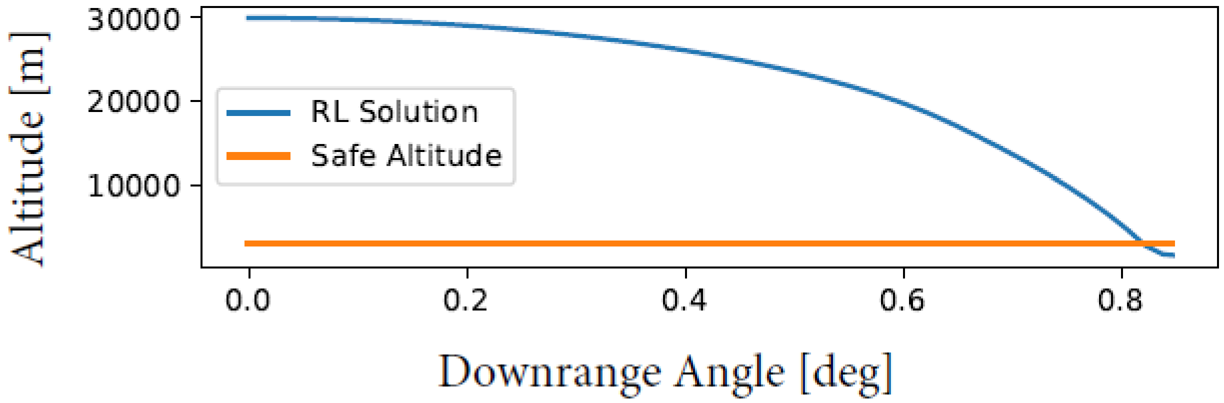
- Emergency Descent Problem Space
- Atmosphere, Sim clock, Scheduler, etc.

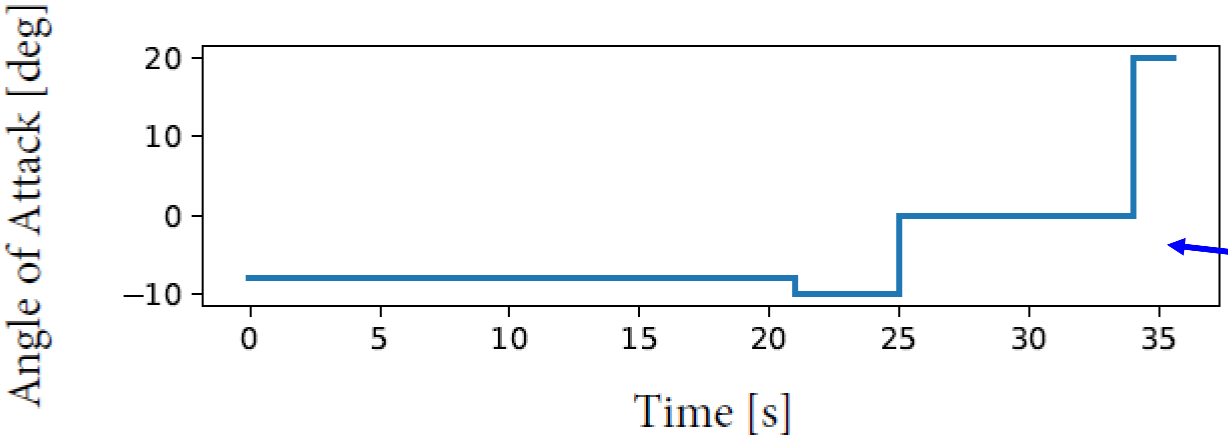
Action Space

- AoA Command
- ±20º in variable increments

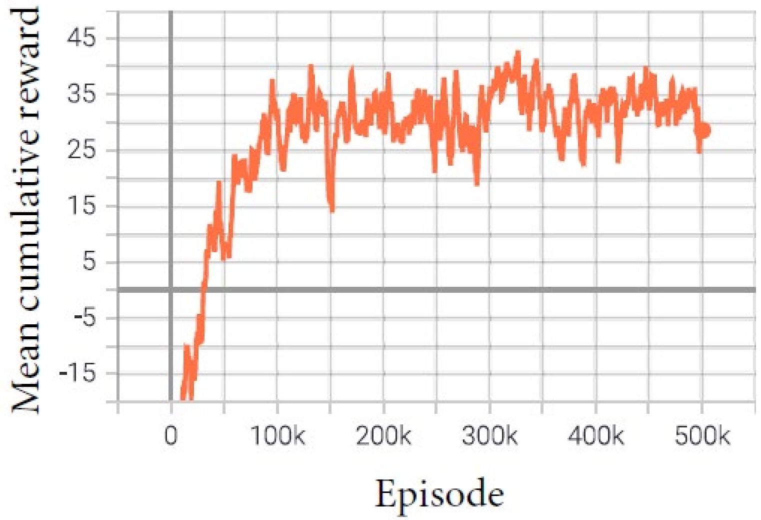
RL Results – Nominal Case (Vehicle Descent from 30km to 3km)

- RL agent trained to provide AoA commands to guide the vehicle to safe altitude
  - Training included randomly sampling vehicle initial conditions
  - Training completed after 500k episodes





Sufficient cumulative reward of +30 to train policy



AoA commands issued by the RL agent

Robustness Testing of RL Solutions

- Purpose: Identify sources of variation in RL problem space and quantify the impact on RL performance

General Sources of Variations in RL

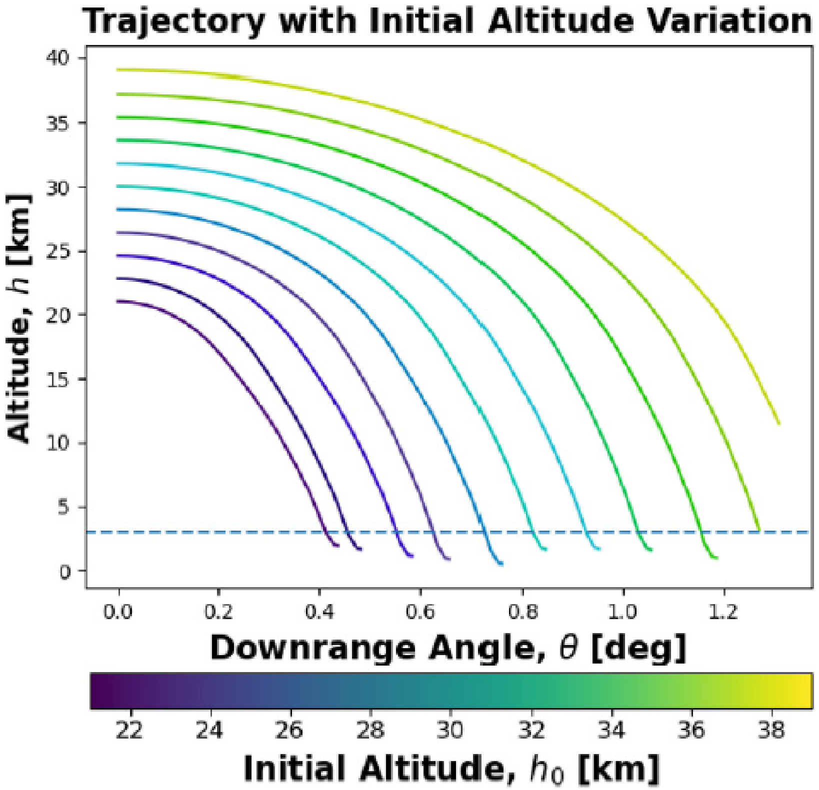
Source	Nature of Variation	Modeling Approach for RT
Environment	Initial Conditions	Latin Hypercube Sampling Monte Carlo Simulations Design of Experiments
Action Space	Tolerance and Sensitivity	Expected probability distribution with parameters (e.g., $\mathcal{N}(\mu, \sigma^2)$)
	Impulses and Hard Overs	Expected magnitude and time duration
State Space	Tolerance and Sensitivity	Expected probability distribution with parameters (e.g., $\mathcal{N}(\mu, \sigma^2)$)
	Impulses and Hard Overs	Expected magnitude and time duration

Derived Test Cases for High-Speed Vehicle RL Solution

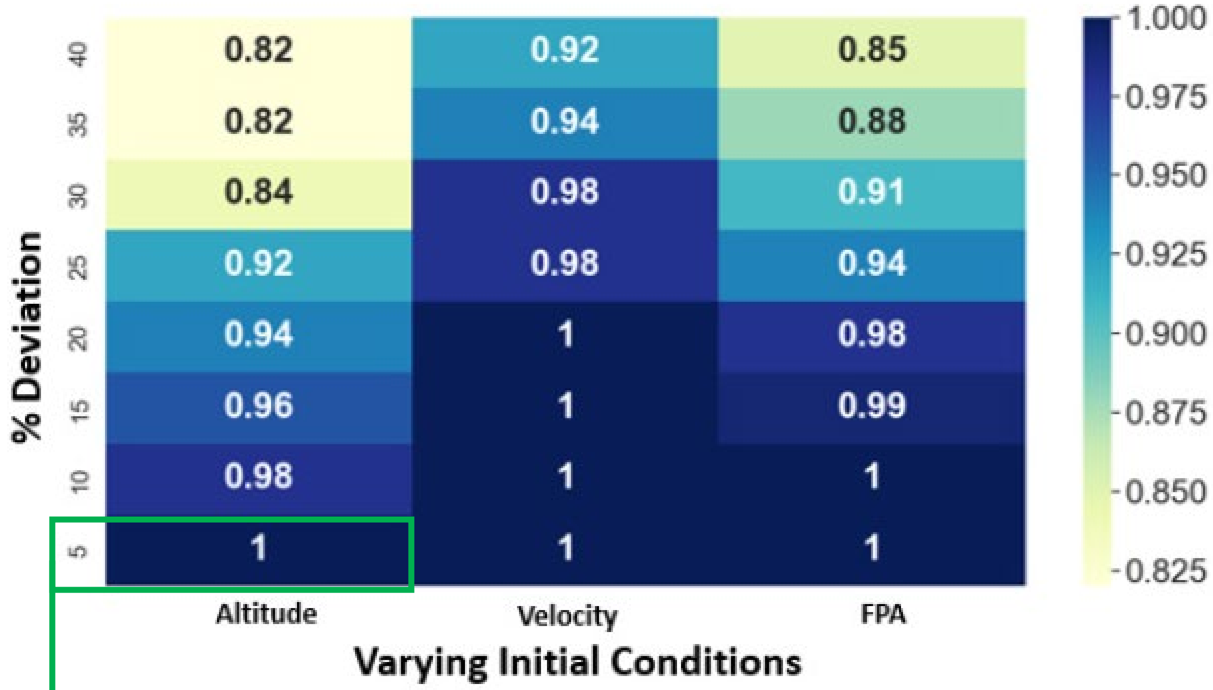
Test Cases	Objective
TC-1	Individually vary environment Initial Conditions (ICs) (i.e., altitude, velocity, FPA) to examine performance
TC-2	Quantify performance bounds on ICs variation with LHS
TC-3	Sensitivity to impulses on action space
TC-4	Sensitivity to random variation in action space
TC-5	Sensitivity to impulses on state space
TC-6	Sensitivity to random variations in state space

Robustness Testing Results (TC 1 & 2)

TC-1 Modeling Approach:
Exercise RL agent by randomly sampling ICs with pre-defined range; Results shown for 30% variations



TC-2 Modeling Approach:
Utilize Latin-Hypercube Sample to generate IC samples outside training bounds
Results shows successful trajectories per 50 samples



Safe target altitude not reached from higher altitudes

100% success only within 5% of altitude variation

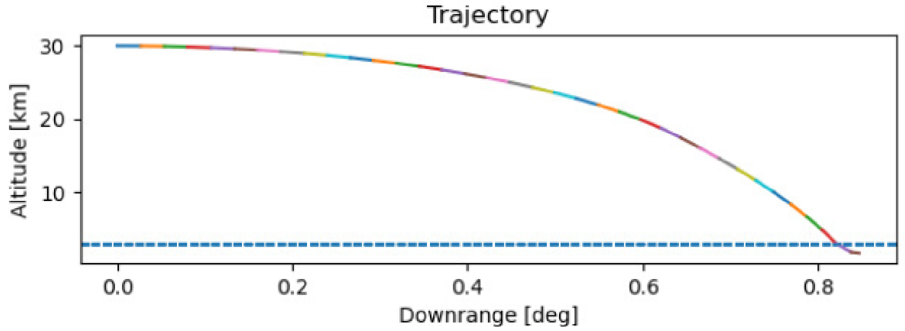
Robustness Testing Results (TC 3)

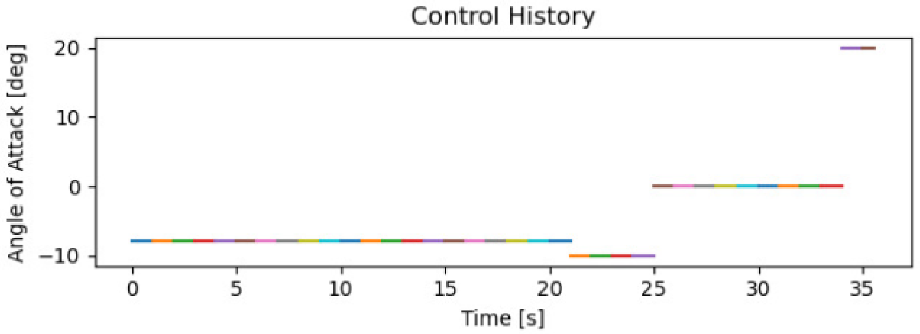
TC-3 Modeling Approach

Introduce random variation in action space, i.e., the AoA command randomly sampled within ±4 o

Nominal Case – Unperturbed

(Success)

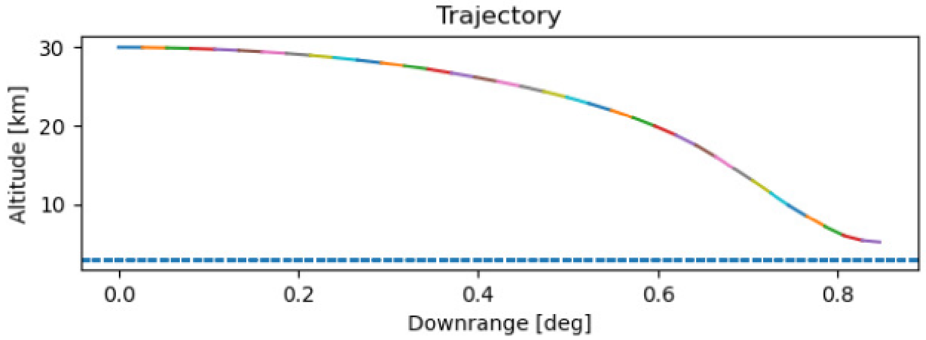


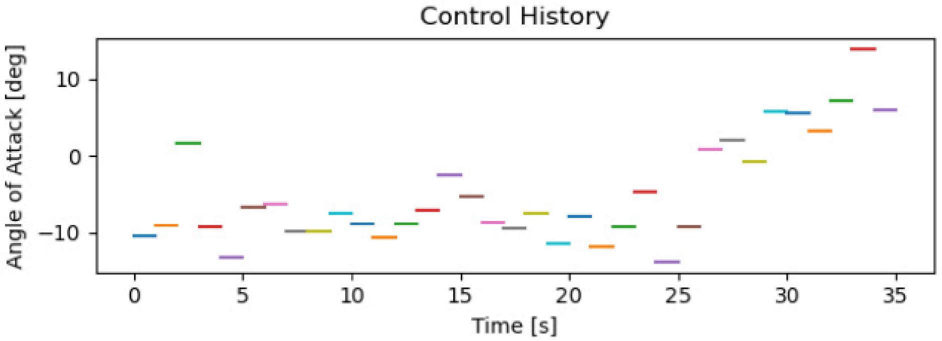


RL AoA command at 1Hz control frequency shown by different colors

TC-3 – Perturbed

(Failure observed after 1000 trials)





Robustness Testing characterizes performance bounds on RL
Helps in setting operational requirements for RL and derived requirements for lower-level control systems.

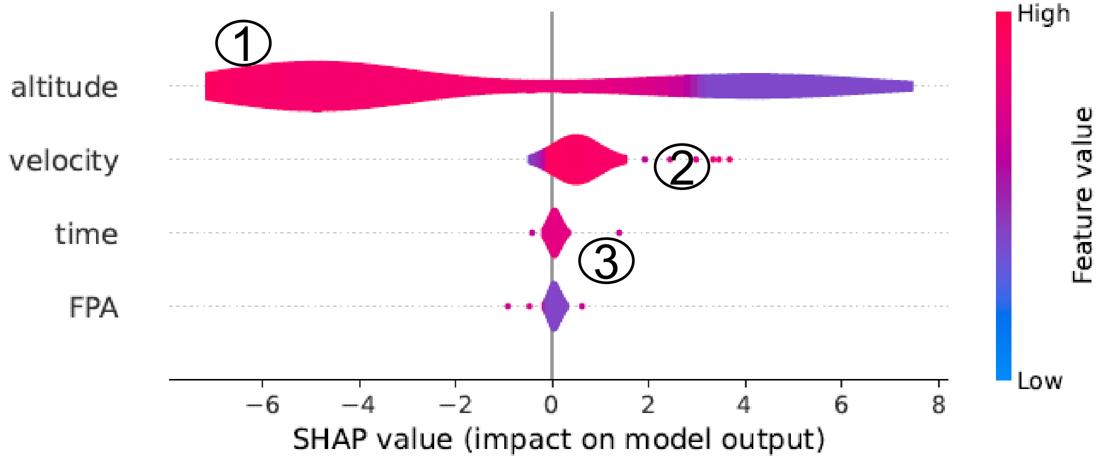
Examination Via Explainable AI (XAI) Techniques

Brief Introduction to Explainable AI

- Investigates trained Deep Neural Network (DNN) models with analytical techniques to extract decision making attributes
- SHapley Additive exPlanations (SHAP)
  - State of the art for reverse engineering the output of any predictive model
  - Yields importance of input features for a given prediction
  - Focuses on coalitions in cooperative game theory

SHAP Applied to RL Problem

- **Inputs:** Time, Altitude, Velocity, and Flight Path Angle
- **Output:** Angle of Attack (between -20˚ and 20˚)
- **Number of trajectories:** 1000
- **Objective:** Reach a particular target in a minimum time

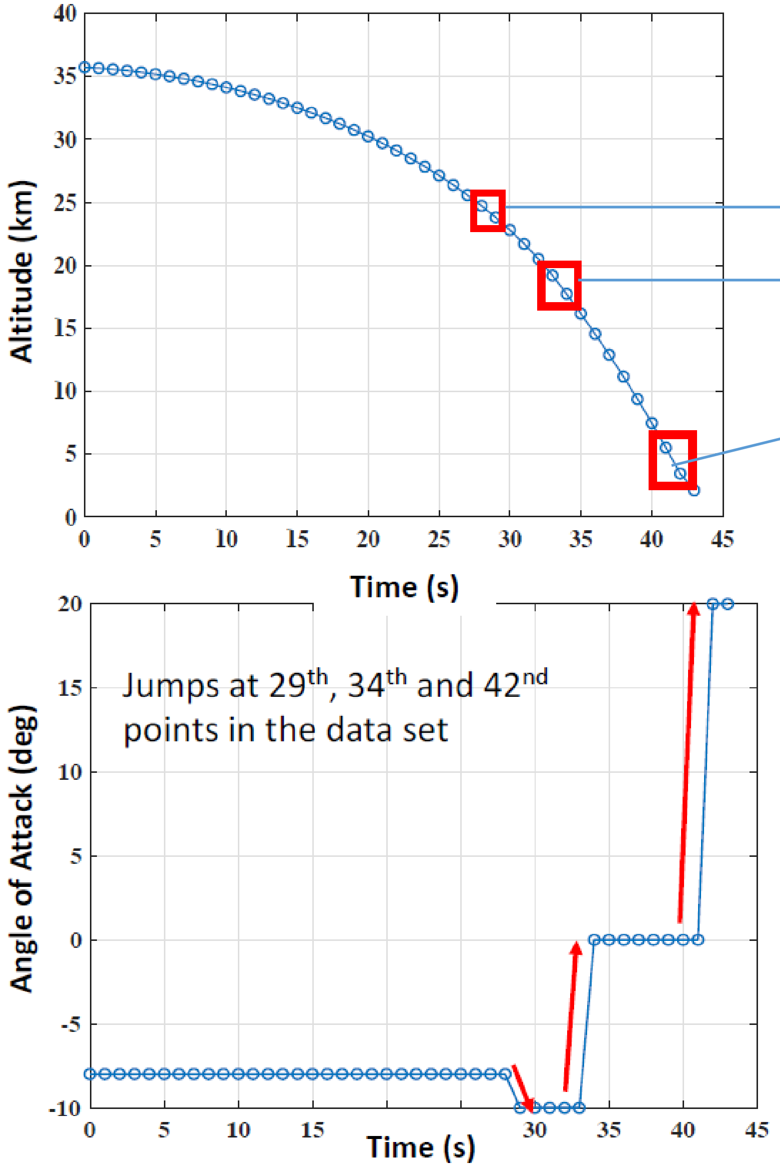


① Higher altitude values oppose a change in AoA whereas low altitude support it.

② Higher velocity values positively influence change in AoA

③ FPA and Time have least impact.

Real Time Analysis with SHAP

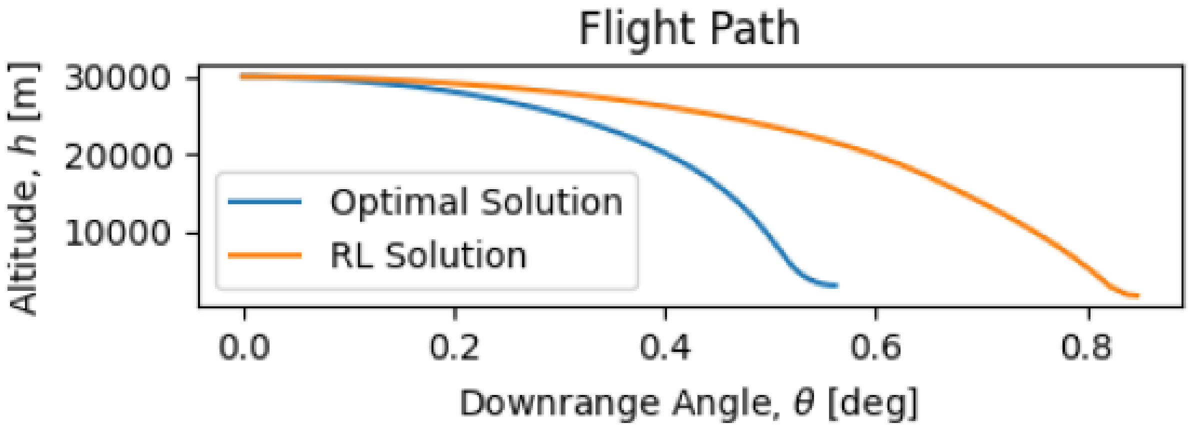


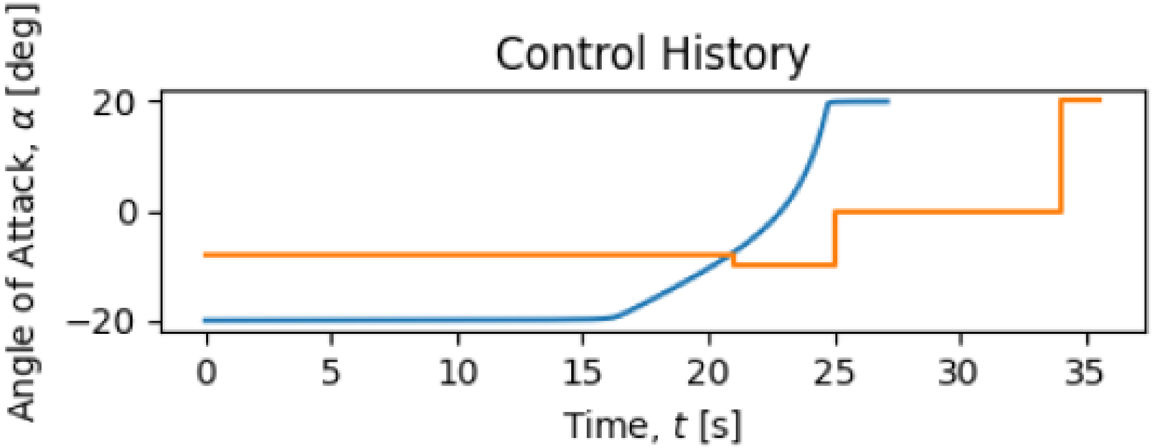
Data Point #	AoA (deg)	SHAP Values			
		Time	Altitude	Velocity	FPA
29	-8	0.3431	1.2848	3.6734	0.2792
30	-10	0.1581	1.8660	3.4531	0.1032
33	-10	0.1613	3.3503	1.9171	0.1517
34	0	0.1488	3.8376	1.4204	0.1736
41	0	0.1330	5.2111	0.1003	0.1362
42	20	0.1116	5.3742	-0.0201	0.1147

- Higher SHAP values for altitude and velocity correlate with the changes in AoA
  - As the vehicle descents to target altitude, higher AoA values are issued to prevent the vehicle from diving further
- Currently, investigating further interpretation of SHAP values and mapping SHAP to actual values

Sample Feasible Vehicle Trajectory and Control History Plot

Validation with Optimal Control Solutions





RL Results with PPO algorithm

Training Options:

Varying initial conditions

500K Episodes

Optimal Control solved by indirect methods using beluga package

RL agent approximates optimal solution

Potential differences due to:

- RL solution is discrete action space
- OP solution is continuous action space
- **Goal is not to exactly reproduce optimal trajectory**

Takeaways and Dissemination

Takeaways

- **RL is being actively developed for applications in real time systems**
  - From a System Engineering perspective, RL is a component that integrates with other system components
  - System Engineering approaches for test, evaluation, and validation are necessary to support RL transition in real systems
    - Performance evaluations that go beyond optimal leaning policy comparison and algorithm development

- **Three-part RL Test and Evaluation Framework proposed in this work provides:**
  - Robustness Testing of RL inspired by Systems Engineering for AI
  - Explainable AI to comprehend RL decision making
  - Validation of RL solutions with known solutions methods

Dissemination

- **Presentations:**
  - AIAA SciTech 2021: “Implementation of Hypersonic Motion Primitives for Reinforcement Learning Using Optimal Control Theory”
  - AIAA Defense 2021: “Reinforcement Learning Techniques for Aerospace Vehicle Missions Through Predator and Prey Models”
- **Publications:** ***(Currently in work)***
  - “Test and Evaluation Framework for Reinforcement Learning”; IEEE Aerospace Conference (Under review)
  - “Testing and Validation of Reinforcement Leaning in Aerospace Applications”, AIAA 2022 Conferences (ETC: Fall 2022)

Our hope is to continue to refine this T&E framework and provide a methodology and tool set for other RL researchers to quantify effectiveness and limitations of RL-based solutions

GMU-Purdue-Sandia Team Acknowledgement

PI and Co-PI

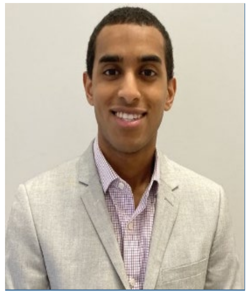
Dr. Ali Raz

Dr. Kris Ezra

Current Graduate and Undergraduate Students

Sean Nolan
PhD Student

Winston Levin
MS Student



Ahmad Mia
MS Student

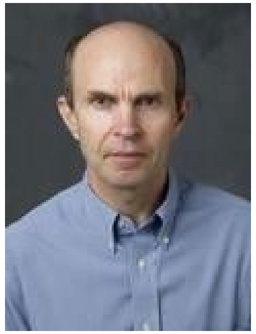
Lauren Risany
Ugrad / SNL Intern

Eli Sitchin
MS Student

TPOC

Dr. Kyle Williams

Technical Staff



Dr. Linas Mockus

Rob Campbell

Backup

Reinforcement Learning Training Configuration

RL Training Setup

$$\boldsymbol{x_0} = \begin{bmatrix} h_0 \\ \theta_0 \\ v_0 \\ \gamma_0 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 30,000 \text{ m} \\ 0^\circ \\ 3,000 \text{ m/s} \\ 0^\circ \end{bmatrix}, \begin{bmatrix} 5,000 \text{ m} \\ 0^\circ \\ 500 \text{ m/s} \\ 2.5^\circ \end{bmatrix}\right)$$

$$\bar{h} = \left|\frac{h - h_{target}}{h_0 - h_{target}}\right|$$

$$\bar{t} = t/t_{max}$$

$$\bar{\gamma} = |\gamma/5^\circ|$$

Reward

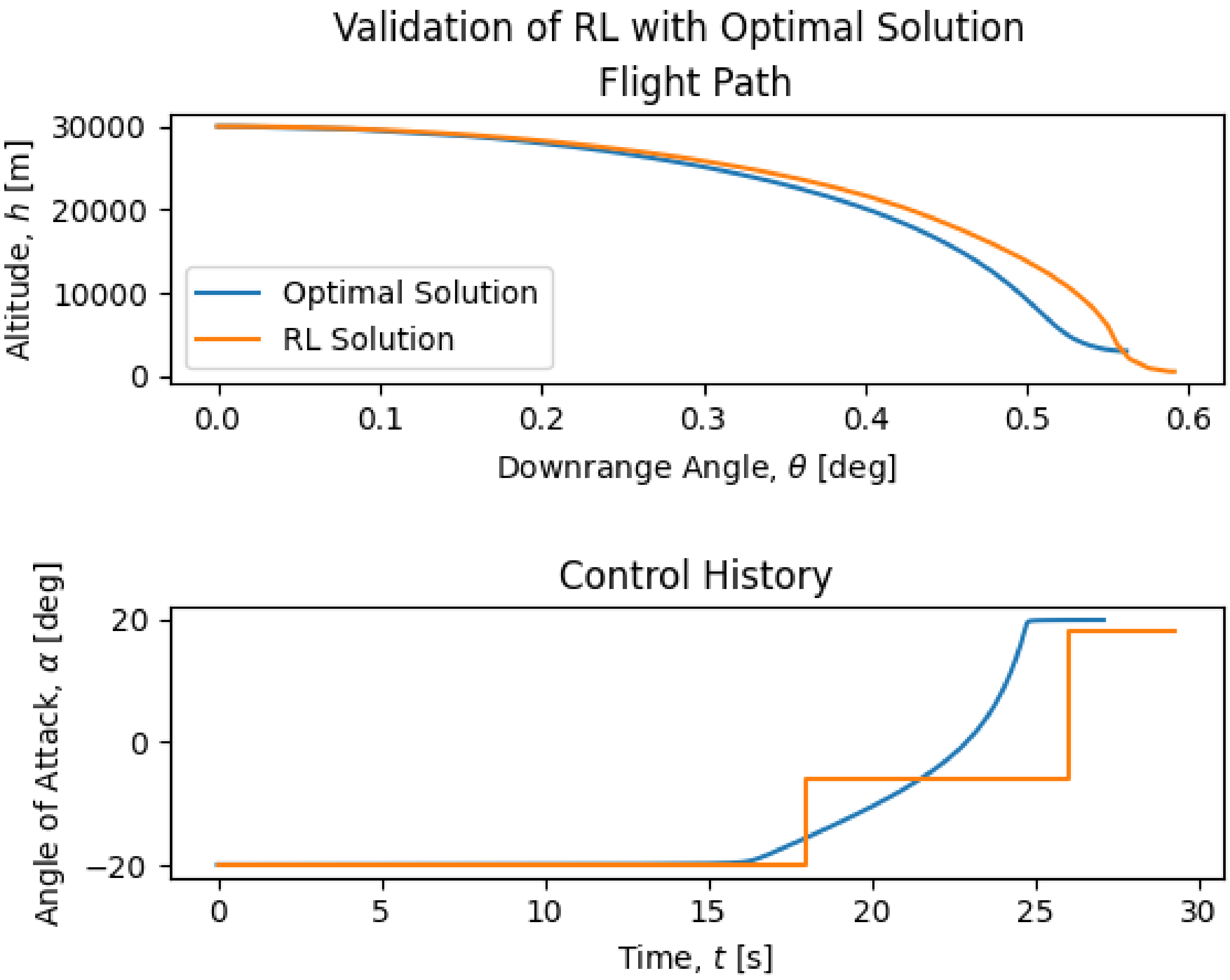
$$\begin{cases} 80(1-\bar{t}) + 20(1-\bar{\gamma}) & \text{if done, success} \\ -100\bar{h} + 16(1-\bar{t}) + 4(1-\bar{\gamma}) & \text{if done, } \neg \text{ success} \\ -\bar{h} & \text{otherwise} \end{cases}$$

RL Hyperparameters

Hyperparameter	Value
batch_size	64
n_steps	512
gamma	0.9700959605924205
learning_rate	0.00010751473610906142
ent_coeff	0.001489192868297319
clip_range	0.11696450376676784
n_epochs	20
gae_lambda	0.8862438354037199
max_grad_norm	4.652723150289042
vf_coef	0.6590547382769023
net_arch	medium
activation_fn	tanh

- The Optuna package for Python was used to optimize the hyperparameters governing the PPO training.
- Using 64 trials with 200,000 steps was sufficient to produce good hyperparameters for training. After optimizing the hyperparameters, the DNN was trained for 500,000 episodes.

VALIDATION WITH OPTIMAL TRAJECTORIES – SINGLE POINT TRAINING



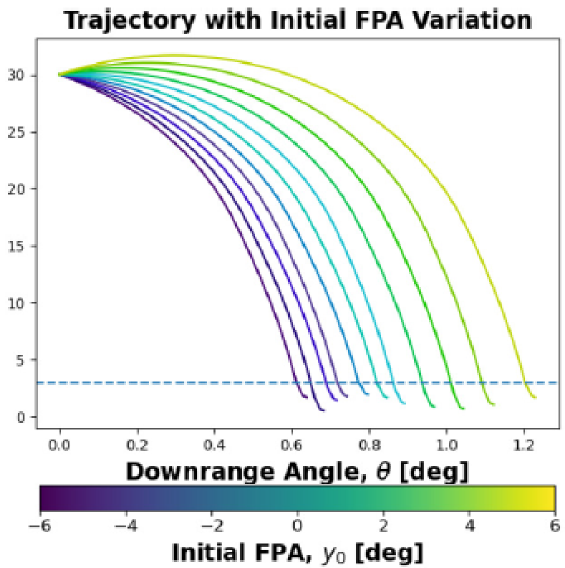
RL Results with PPO algorithm

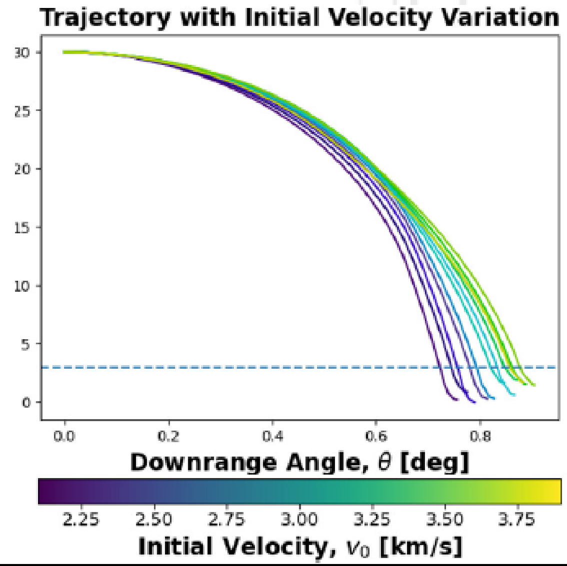
Control Options:

Training for nominal CASE

NO Variation in Training ICs

Back-up Test Case Results





RL unable to recover if an impulse was applied during the 25-30 second window; No impact was found for moderate impulses in 0-20 seconds