IEEE CSS TCAC Workshop at the 7th IEEE Conference on Control Technologies and Applications, August 15th, 2023

# Towards Trustworthy Autonomy:

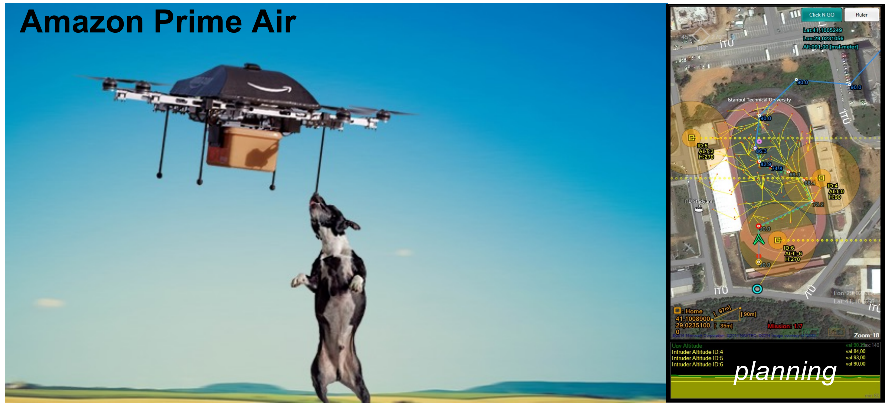
## *How AI Can Help Address Fundamental Learning, Adaptation and Decision-Making Challenges in Aerospace Controls*

**Prof. Gokhan Inalhan,**
**Autonomous Systems and AI**
**Cranfield University**

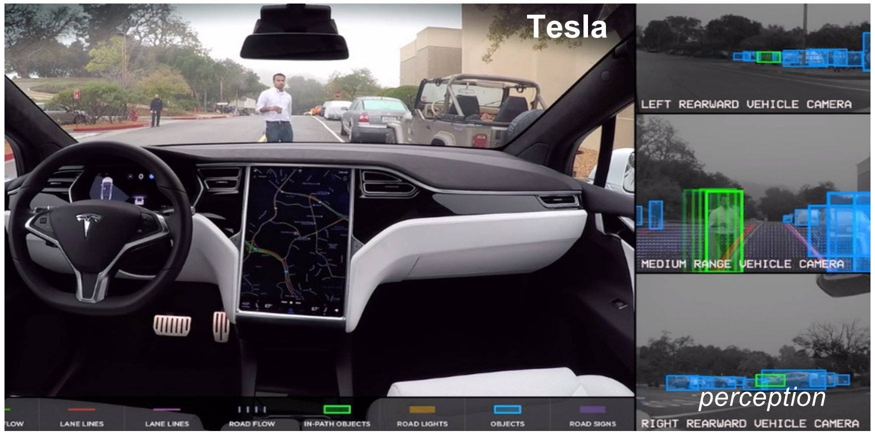
**August 15th, 2023**

www.cranfield.ac.uk

# Autonomous Systems and Artificial Intelligence in a Customer-Led World



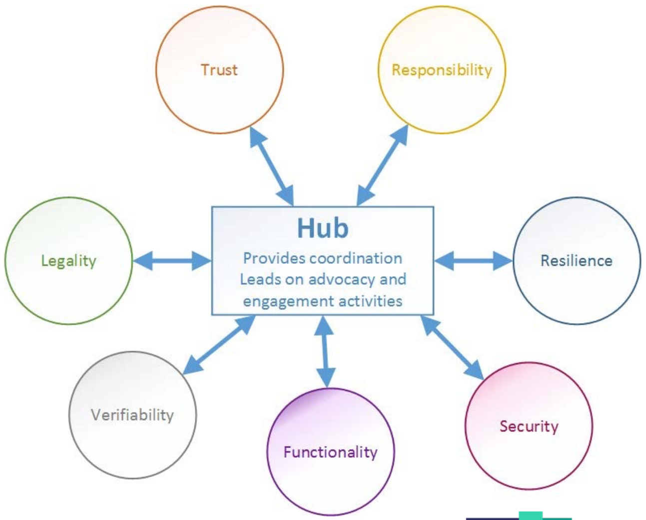



New technology solutions lead to new Business Models, and Autonomy and AI are key enablers

# EPSRC Trustworthy Autonomous Systems (TAS) Node on Security : The Control Challenge

- Autonomous Systems rely on the ability to conduct **run time adaptations of control decisions** over attacks or "perceived" attacks:
  - Adversaries
    - Physical
    - Information-plane
  - Information and dynamic environment uncertainties
  - Degraded performance
    - CNS and Infrastructure
    - Actuators
- How to do this in a "**trustworthy**" fashion in a "**learning-enabled context**"?
  - Safe
  - Secure
  - Reliable

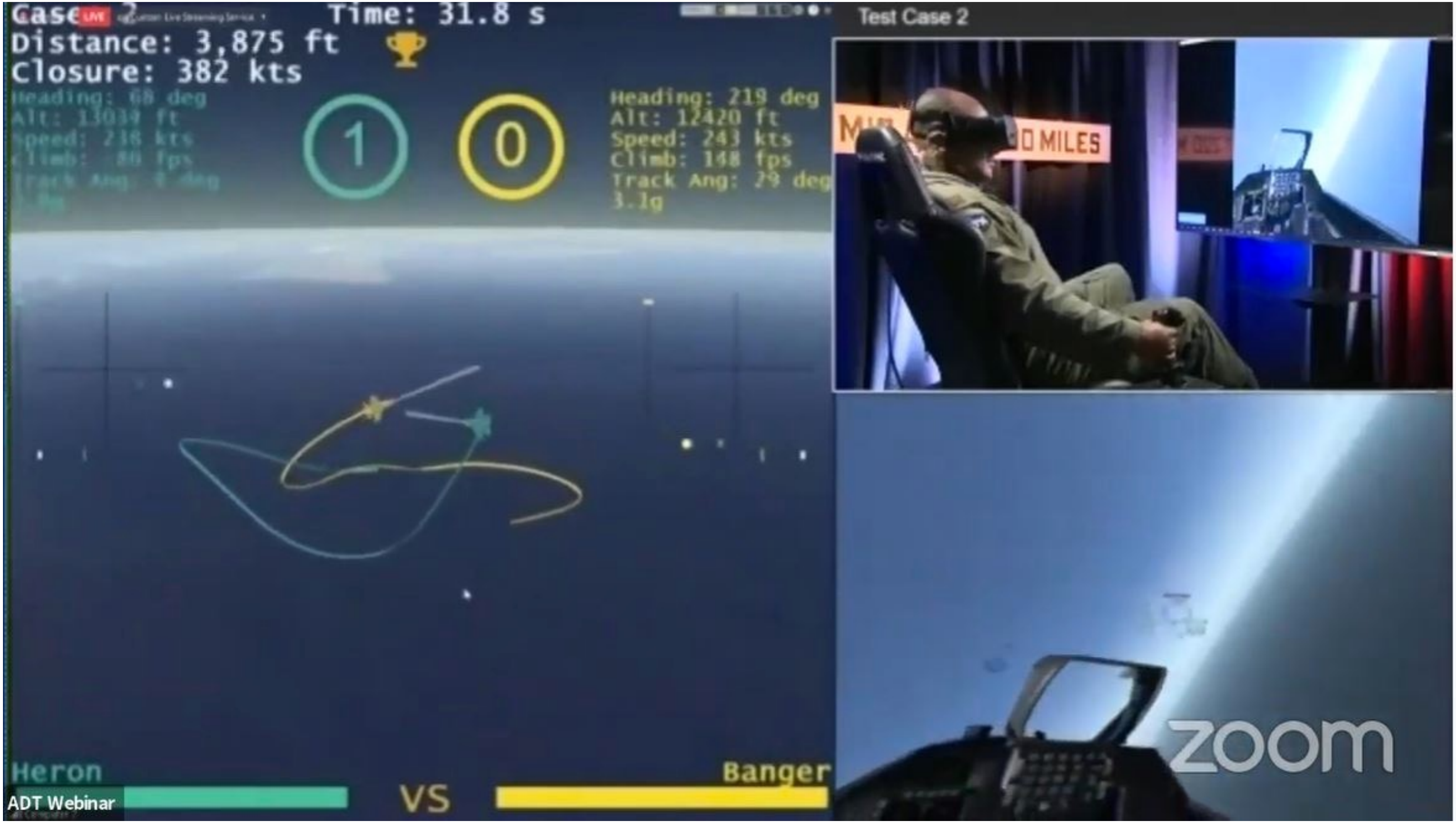



Engineering and Physical Sciences Research Council


Source : BAE Systems

DARPA Air Combat Evolution (ACE) program


REC
Case: 10 Time: 77.5 s
Distance: 1,144 ft
Closure: 62 kts
Heading: 144 deg
Track Ang: 111 deg
Flawless Victory
Lockheed
Case: 10 Time: 31.0 s
Distance: 2,374 ft
Closure: -58 kts
Aurora
Kills
Flawless Victory
Aurora
VS
Perspecta
Case: 10 Time: 102.3 s
Distance: 1,420 ft
Closure: 75 kts
GTRI
VS
SoarTech
Case: 10 Time: 87.5 s
Distance: 3,738 ft
Closure: 166 kts
Track Ang: 113 deg
EpiSci
VS
physicsAI
Test Case 10
Left Right
Heading (deg) 0 -178.5
Altitude (ft) 15487 10621
Speed (kts) 403 439
Fuel (%) 40 70
Separation Amount (ft)
Forward 5005
Lateral 2471
2471
4966
12:54 54
Thursday - January 30
34 MINUTES LEFT TO CONTROL ZONE UPDATE

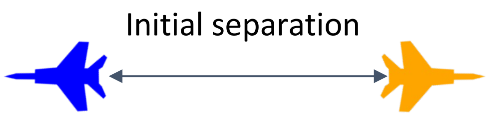

Case 2
Time: 31.8 s
Distance: 3,875 ft
Closure: 382 kts
Heading: 68 deg
Alt: 13039 ft
Speed: 238 kts
1
0
Heading: 219 deg
Alt: 12420 ft
Speed: 243 kts
Climb: 148 fps
Track Ang: 29 deg
3.1g
Heron
VS
Banger
ADT Webinar
Test Case 2
zoom

# Alpha Dog Results: Assumptions

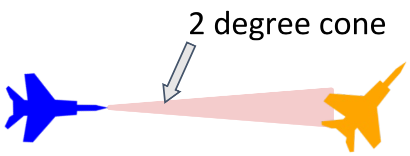
- Main assumptions:
  - Our own and **enemy's position**, velocity, attitude, angular rate is **known at 50 Hz**.
  - **No self preservation logic**, aircrafts can go through each other.
  - **Shooting is activated automatically** without any trigger command.
  - The combat starts with the aircraft's noses pointing opposite of each other.

| Property | Object Platform(s) |
|---|---|
| Time since start of engagement | N/A |
| Latitude/longitude/altitude | All |
| Location relative to scenario center (x/y/z) | All |
| Orientation (roll//pitch/heading) | All |
| Velocity (x/y/z) | All |
| Angular rate (roll/pitch/yaw rates) | All |
| Acceleration (x/y/z) | Friendly Only |
| Angular acceleration (roll/pitch/yaw second order rates) | Friendly Only |
| Angle of attack | Friendly Only |
| Sideslip angle | Friendly Only |
| Current control surface deflections | Friendly Only |
| Throttle position | Friendly Only |
| Fuel state | Friendly Only |
| Current thrust | Friendly Only |
| Distances to other aircraft | All |
| Angle of bearing to other aircraft | All |

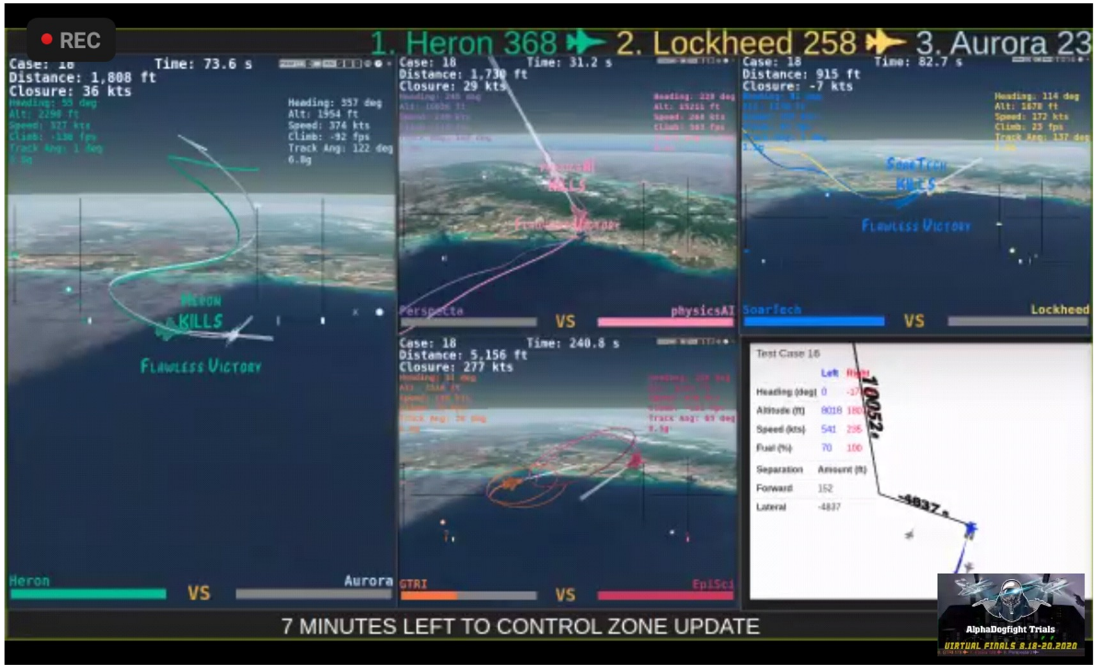
Start Condition:




Win Condition:



# Alpha Dog Results: Winner Heron System’s Falco




- Better stick and throttle control than others. Heron’s **precise cone direction control** made them winner.
- Control command are given at 10 Hz, whereas other teams used 50Hz.
- **102 differently configured agent** trained and the main agent trained against these, similar to the AlphaStar.
- **4 billion episodes were trained**, which took **5 weeks**.

# Alpha Dog Results: Fighter Pilot Comments

- In real life, the pilot can only see where the sensor can see, which gives an unfair advantage against the human operator.
- The AI agents did not have collision avoidance logic or self-preservation logic.
- Being in the shooting cone is considered a shot, but human pilots are trained to hold the target in plane and in range, and take into account the bullet time of flight.
- The human fighter pilots would lose to the human game player in the digital combat simulator(DCS).


C.W. Lemoine


- A match between Heron System's Falco and human DCS player is set.
- The result was 3 wins for Falco, 1 win for human and 1 draw.

# Our Concept

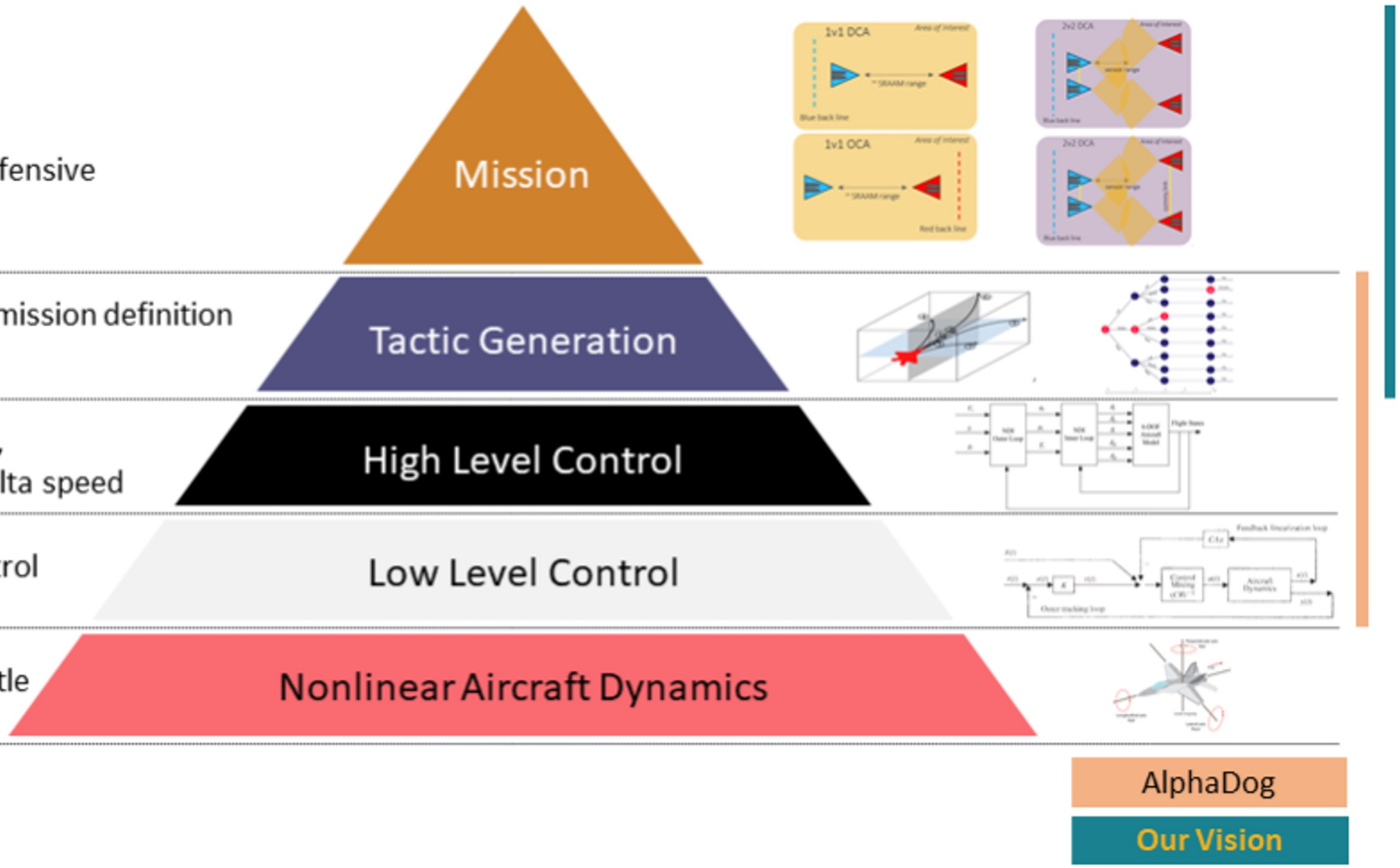
Define operational type;
VWR/BWR, defensive or offensive

Generate tactics based on mission definition and aircraft capability

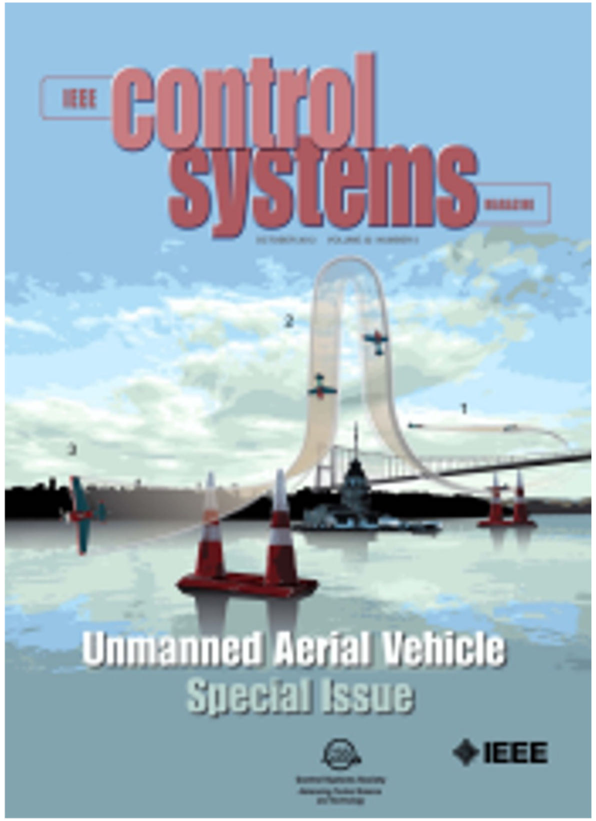
Command delta flight path,
delta heading angle and delta speed

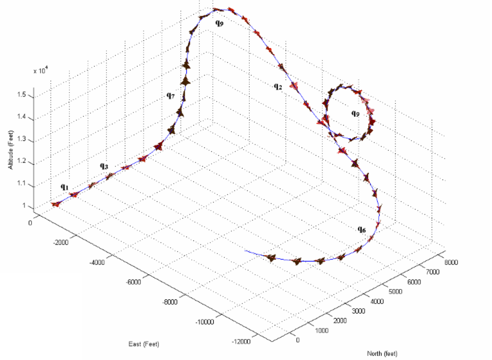
Body rate and throttle control

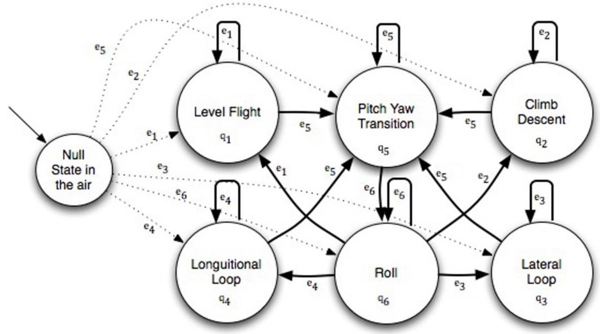
Directly use stick and throttle



# Mimicking the best fighter pilots… 2008

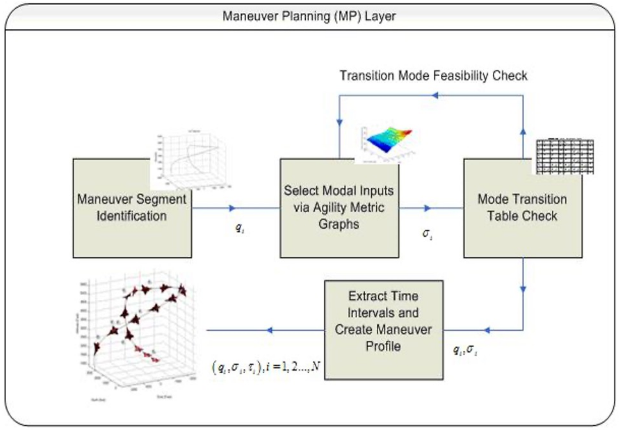





Maneuver Library Transformer

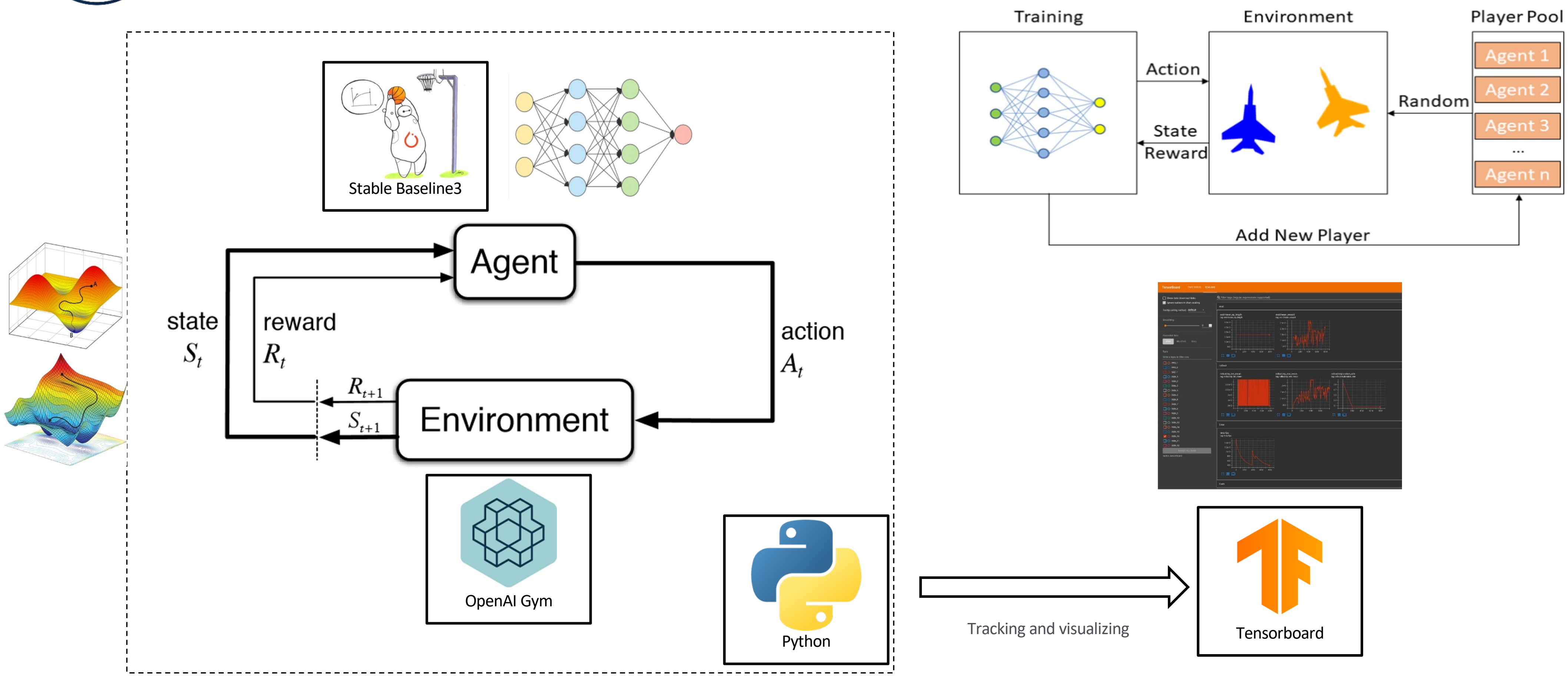
| | Mode | Constrained States $O$ | Driven Dynamics $D$ | Modal Inputs $M$ |
|---|---|---|---|---|
| $q_1$ | Level Flight | $e_p, h, \phi, \psi, \beta, P, Q, R$ | $n_p, \theta = \alpha$ | $V_T(t)$ |
| $q_2$ | Climb/Descent | $e_p, \phi, \psi, \beta, P, Q, R$ | $n_p, h, \theta$ | $V_T(t), \gamma = \theta - \alpha$ |
| $q_3$ | Lateral Loop | $r_{lat}, \beta$ | $\eta_{lat}, h, \alpha, \phi_w, P, Q, R$ | $V_T(t), \theta_w, \dot{\psi}_w$ |
| $q_4$ | Longitudinal Loop | $r_{lon}, \beta$ | $\eta_{lon}, e_p, \alpha, \phi'_w, P, Q, R$ | $V_T(t), \dot{\theta}'_w, \psi'_w$ |
| $q_5$ | Pitch-Yaw Transition | $n_p, e_p, h, V_T, \phi$ | $\theta, \theta_w, \psi, \psi_w$ | $P(t), Q(t), R(t)$ |
| $q_6$ | Roll | $n_p, e_p, h, \theta, \psi, V_T, \beta$ | $\alpha, \phi$ | $P(t), Q(t), R(t)$ |




N. Kemal Ure and Gokhan Inalhan, "*Autonomous Control of UCAVs Design of a multimodal control and flight planning framework for agile maneuvering*", IEEE Control Systems Magazine, 32-5, 74-95, 2012

# Autonomous Execution of Aircraft Supermaneuvers

# Reinforcement Learning Architecture

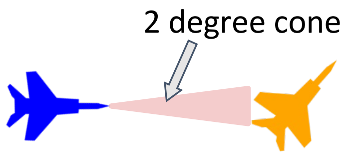


# Building Blocks of Tactics : Fundamental Transformer

## Reward Function

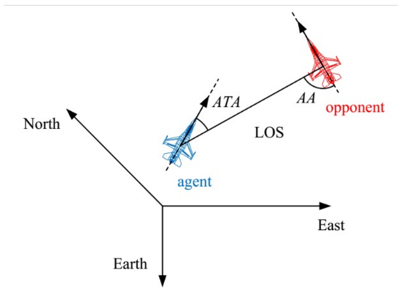
- Keep nose on target
- Stay behind
- Keep distance

Win Condition:



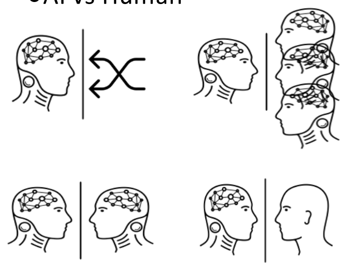

## Relative Initial State

- Orientation
- LOS vector
- Relative speed



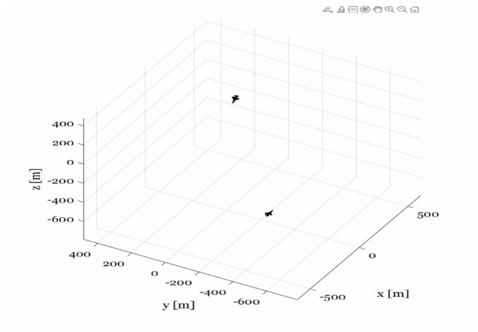

## Against Who

- AI vs Random agent
- AI vs AI
- AI vs Pool of AI
- AI vs Human

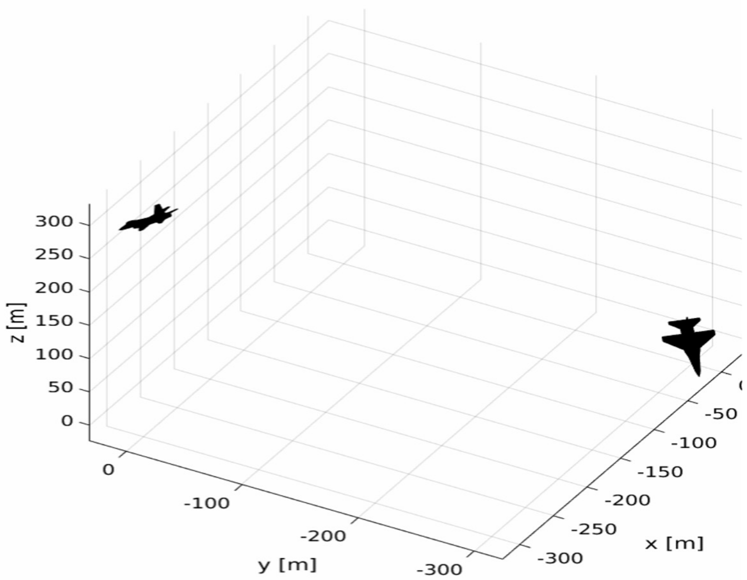


## Tactics

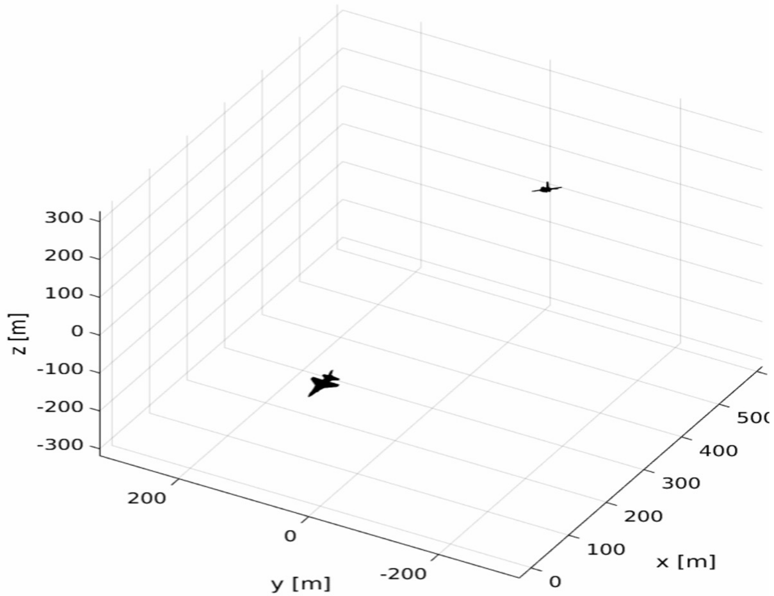
- Competency
- Sensitivity
- Capability



# Effect of Weapon of Engagement Zone

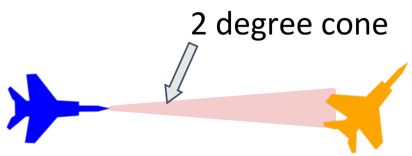



Weapon Of Engagement Zone: Full Cone

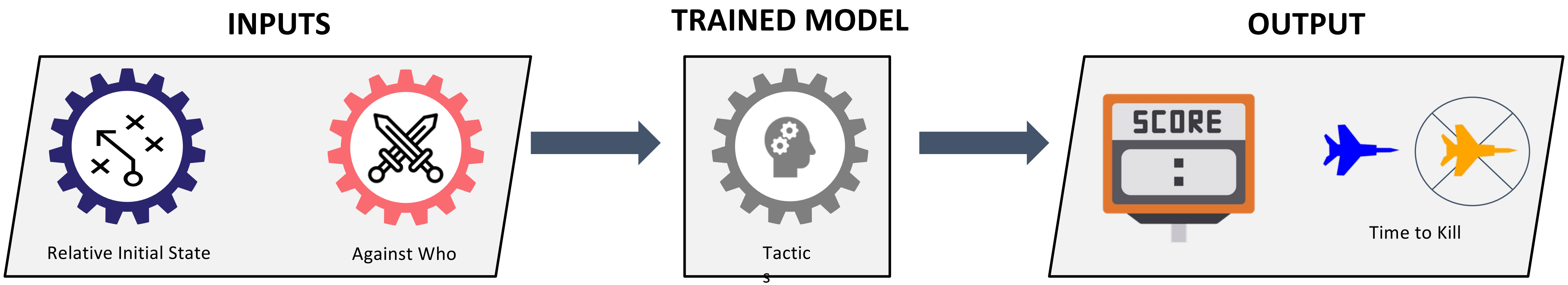


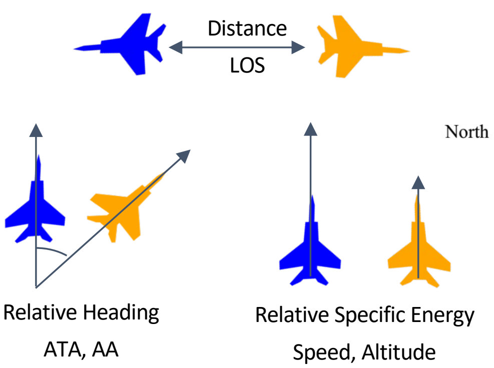

Weapon Of Engagement Zone: Back Cone

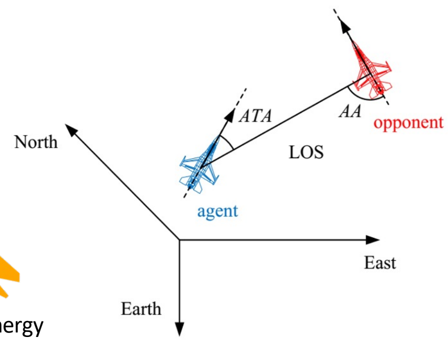
Win Condition:

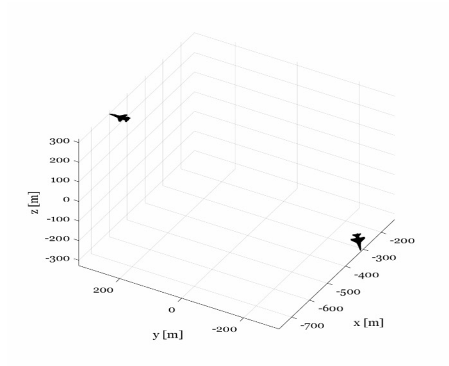


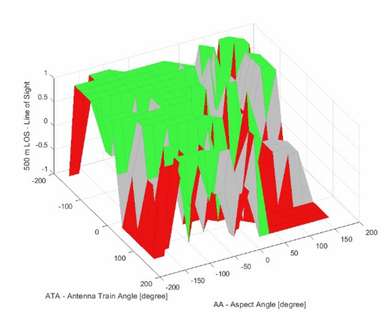
# Winning Assessment with Changing Combat Conditions



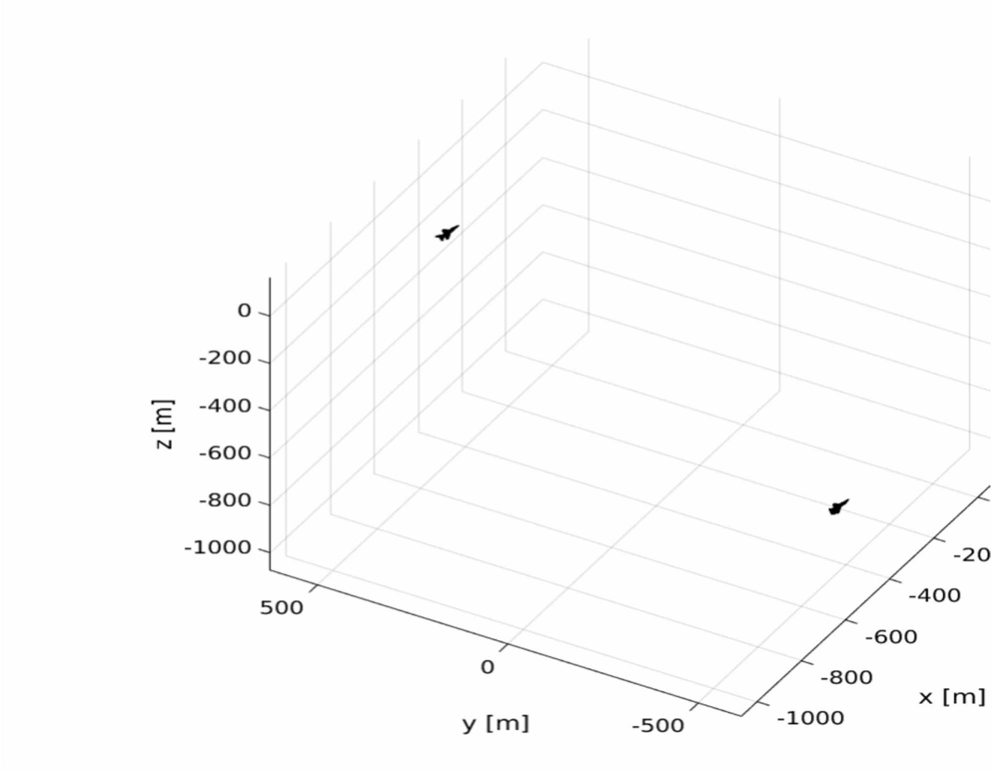



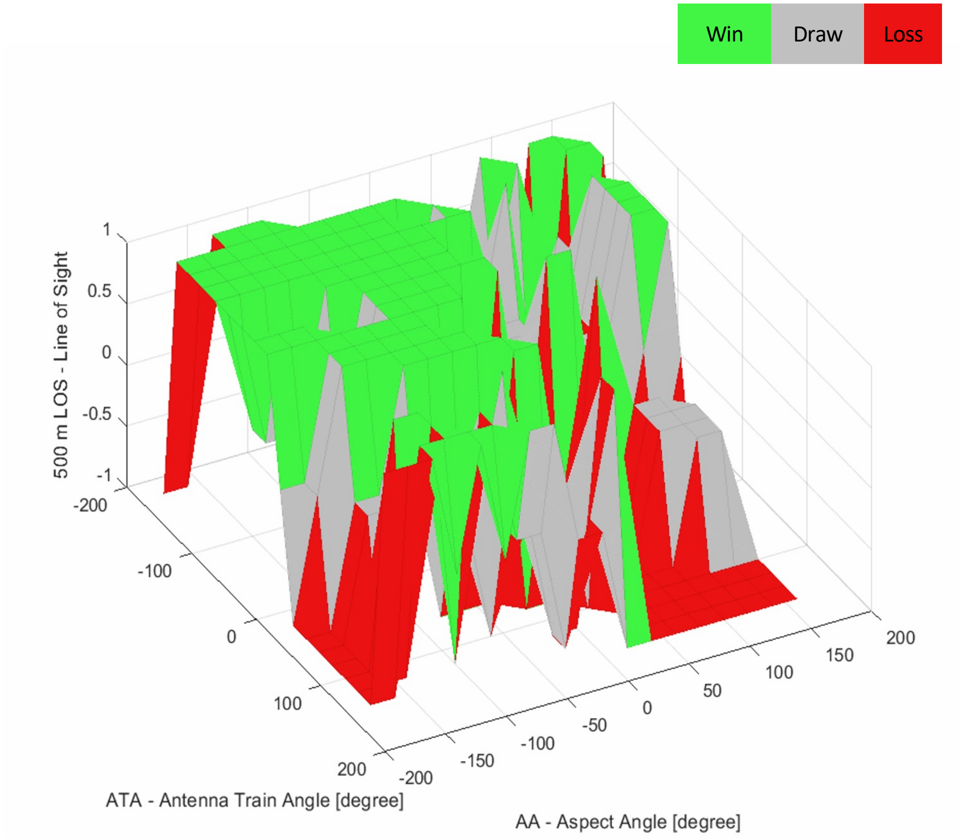



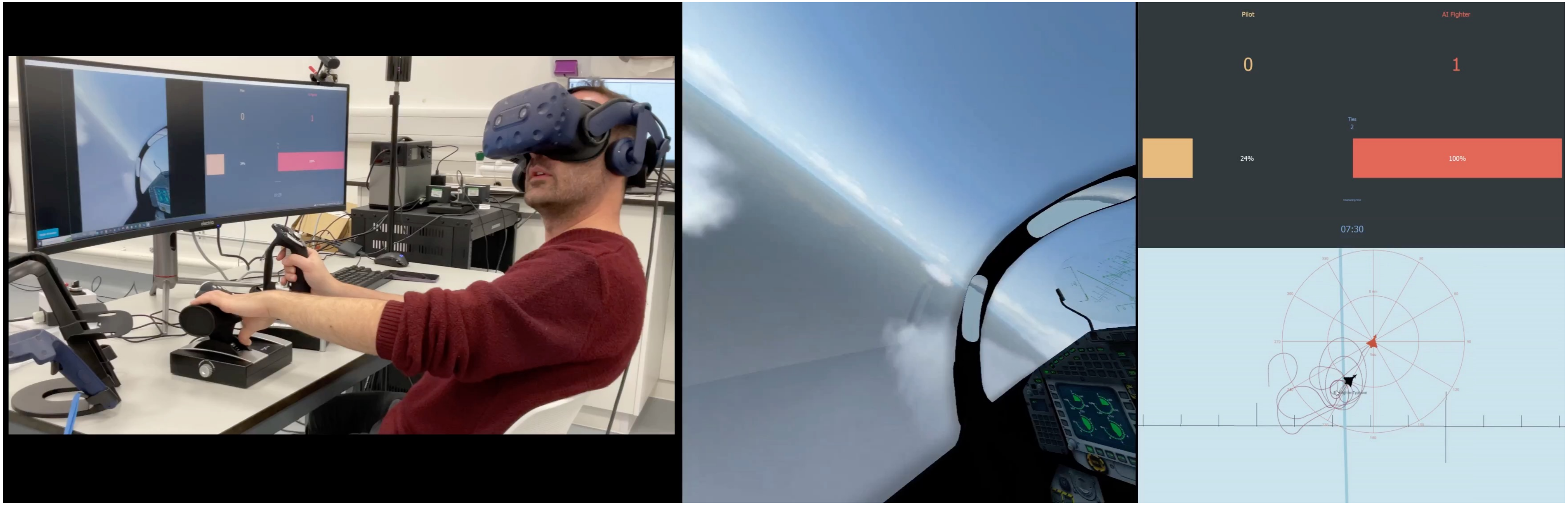



# Tactics and Engagement Decision



# Our VR-Based Combat Evaluation System



# Our Methodology: Stages

1

**Stage 1: Create Environment**

- Aircraft classes
- Multi-processor utilization
- Lightweight, fast training cycle
- Episode length, termination conditions

2

**Stage 2: Design Reward Functions**

- No numerical ambiguity
- Continuous, differentiable
- Episodic, and geometric rewards

3

**Stage 3: Initial Training**

- Curriculum learning, against random agent
- Start from small the set of initial condition
- Introduce different start location and orientation

4

**Stage 4: Self-Play Training**

- Incremental learning
- Update opponent model regularly to promote further learning

# Aircraft Model

- Discrete action sets with maneuver decomposition reduces 6DoF nonlinear dynamics to 3 DoF point mass model with distinct control input set
- Discretize action set
  - Combination of maximum and minimum delta velocity, delta path angle, and delta heading angle commands.
  - Total 27 discrete actions

$$V_{t+\delta t} = V_t + \frac{K_V}{s+K_V}\Delta V_c\Big|_{\delta t} = V_t + \Delta V_c \times \left(1 - e^{-K_V \delta t}\right)$$

$$\chi_{t+\delta t} = \chi_t + \frac{K_\chi}{s+K_\chi}\Delta \chi_c\Big|_{\delta t} = \chi_t + \Delta \chi_c \times \left(1 - e^{-K_\chi \delta t}\right)$$

$$\gamma_{t+\delta t} = \gamma_t + \frac{K_\gamma}{s+K_\gamma}\Delta \gamma_c\Big|_{\delta t} = \gamma_t + \Delta \gamma_c \times \left(1 - e^{-K_\gamma \delta t}\right)$$

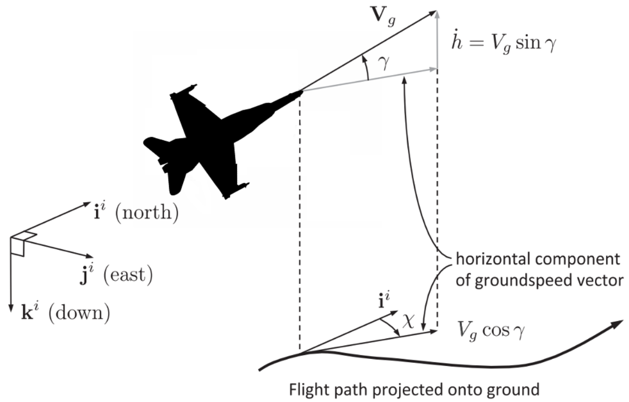
$$x_{t+\delta t} = x_t + \delta t V_{t+\delta t} \cos \chi_{t+\delta t} \cos \gamma_{t+\delta t}$$

$$y_{t+\delta t} = y_t + \delta t V_{t+\delta t} \sin \chi_{t+\delta t} \cos \gamma_{t+\delta t}$$

$$z_{t+\delta t} = z_t - \delta t V_{t+\delta t} \sin \gamma_{t+\delta t}$$

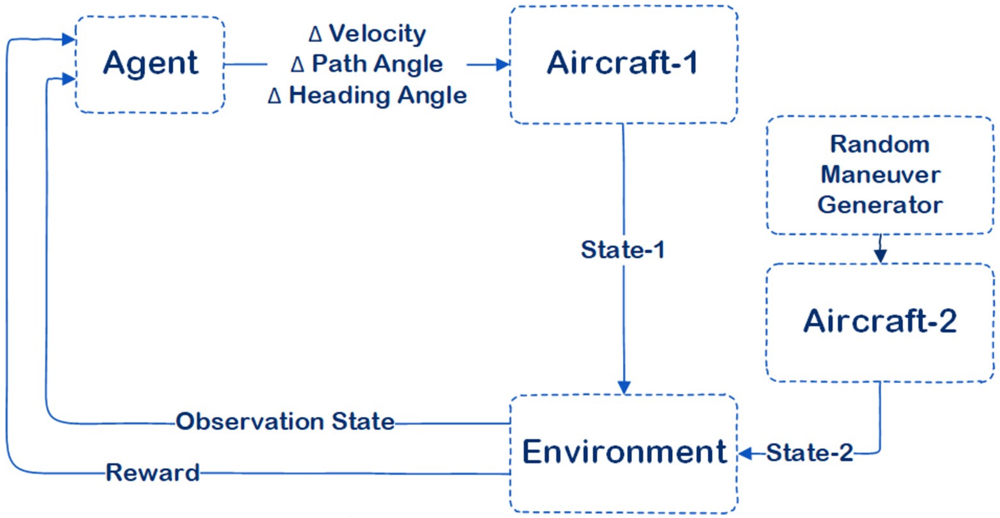
| No | Manoeuvre | Control Values | | |
|---|---|---|---|---|
| | | $\Delta V_c$ | $\Delta \chi_c$ | $\Delta \gamma_c$ |
| 1 | Right upward turn accelerate | 1 | 1 | 1 |
| 2 | Right turn accelerate | 1 | 1 | 0 |
| 3 | Right downward turn accelerate | 1 | 1 | -1 |
| 4 | Upward turn accelerate | 1 | 0 | 1 |
| 5 | Forward accelerate | 1 | 0 | 0 |
| ... | ... | ... | ... | ... |
| 27 | Left downward turn decelerate | -1 | -1 | -1 |

| Parameters | Symbol | Range or value |
|---|---|---|
| Velocity range | $V$ | [100 m/s, 250 m/s] |
| Heading angle range | $\chi$ | [-180°, 180°] |
| Path angle range | $\gamma$ | [-180°, 180°] |
| Delta velocity command | $\Delta V_c$ | [-10 m/s, 10 m/s] |
| Delta heading angle command | $\Delta \chi_c$ | [-20°, 20°] |
| Delta path angle command | $\Delta \gamma_c$ | [-20°, 20°] |
| Velocity gain | $K_V$ | 2 |
| Heading angle gain | $K_\chi$ | 0.6 |
| Path angle gain | $K_\gamma$ | 0.4 |

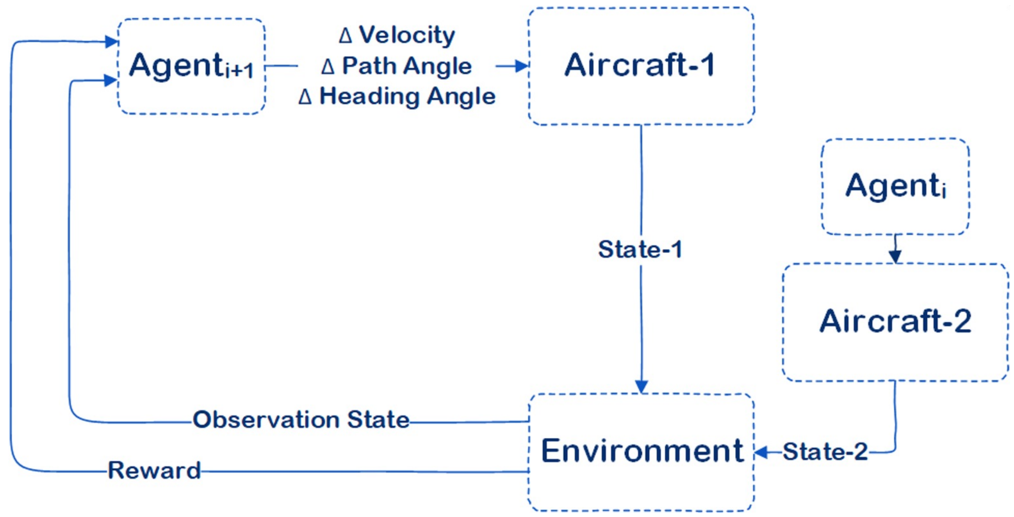


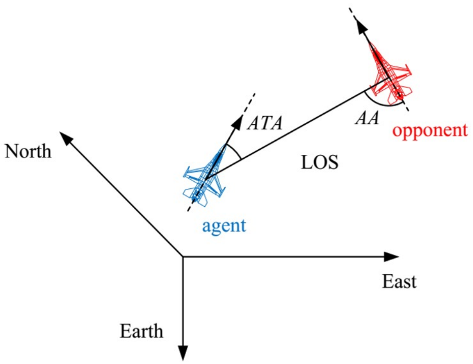
# Our Methodology: Training Architecture

**Agent1: Random Motion Agent**



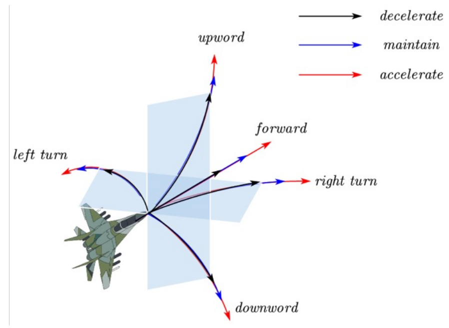

**Agent_i: Self-play**




### Observation Space

- Line of sight vector
- Own aspect angle (AA)
- Own antenna train angle (ATA)



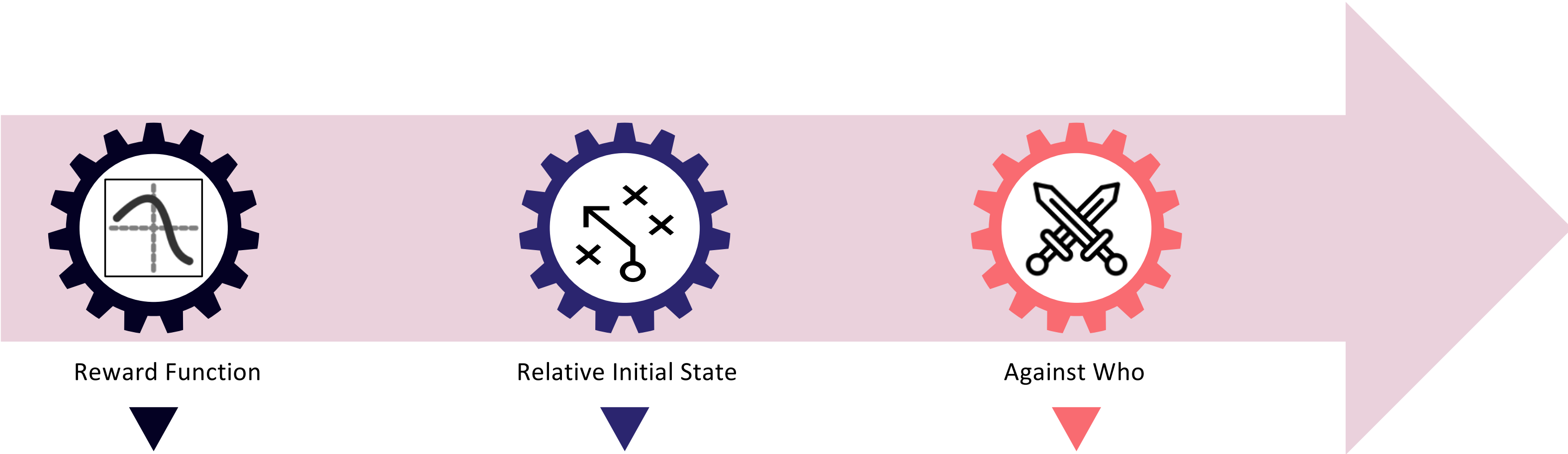

### Action Space

- Delta speed
- Delta flight path angle
- Delta heading angle

# Key Benefits of Our Methodology

| | |
|---|---|
| **Discrete Actions** | • Search through discrete action set using the maneuver decomposition.<br>• Results in a faster learning compare to the continuous action set. |
| **Generalization** | • Aircraft dynamics approximating almost any aircraft.<br>• Trained across wide range of initial conditions. |
| **Explainability** | • Discrete action set allows explainability by relating discrete actions to decomposed rewards in decision tree. |

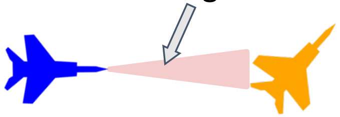
# Reward Functions: Building Blocks of Tactics



**Reward Function**

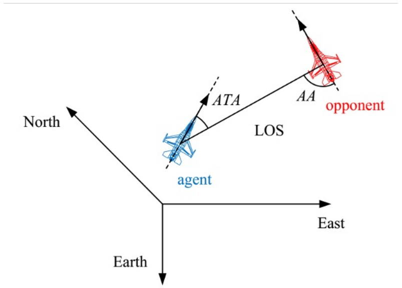
- Keep nose on target
- Stay behind
- Keep distance

Win Condition:



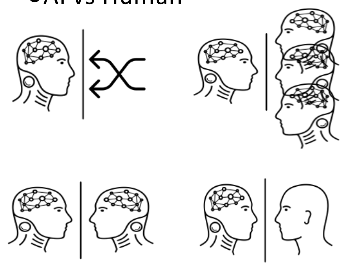

**Relative Initial State**

- Orientation
- LOS vector
- Relative speed



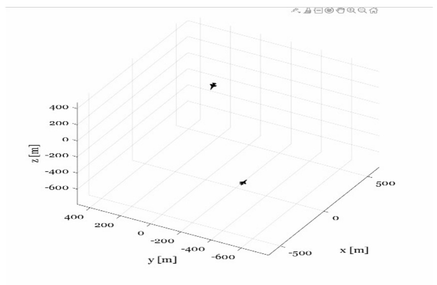

**Against Who**

- AI vs Random agent
- AI vs AI
- AI vs Pool of AI
- AI vs Human



**Tactics**

- Competency
- Sensitivity
- Capability



# DARPA Alpha Dog Fight vs Our Approach

| | DARPA Alpha Dog Fight | Our Approach |
|---|---|---|
| **Initial Condition** | Fixed | Random |
| **Win Condition** | 2 degree cone | 2 degree cone |
| **Observation Space** | Position, Velocity<br>Attitude, Angular Rate<br>Distance | Aspect Angle(AA)<br>Antenna Train Angle(ATA)<br>Line of Sight Vector(LOS) |
| **Action Space** | Roll Rate<br>Pitch Rate<br>Yaw Rate<br>Throttle | Delta Speed<br>Delta Path Angle<br>Delta Heading Angle |
| **Aircraft Type** | 6 DoF Aircraft Dynamics | 3 DoF Point Mass |
| **Simulation Environment** | JSBSim | Custom |
| **Hardware Used** | 5 workstations. Each has 128 core CPU<br>and 6 RTX 6000 GPU | 1 workstation with 128 core CPU<br>and 2 RTX A6000 GPU |
| **Number of Episode** | 4 Billion | ~80 Million |
| **Training Time** | 5 Weeks | 24-36 Hours |

# Reward Functions: Tactic Set 1

## Training Details

**Rewards:**

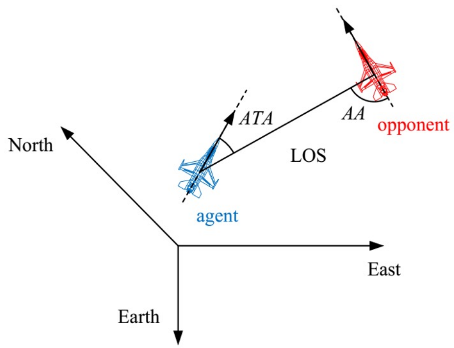
- Stay behind
- Keep nose on target
- Preserve your distance

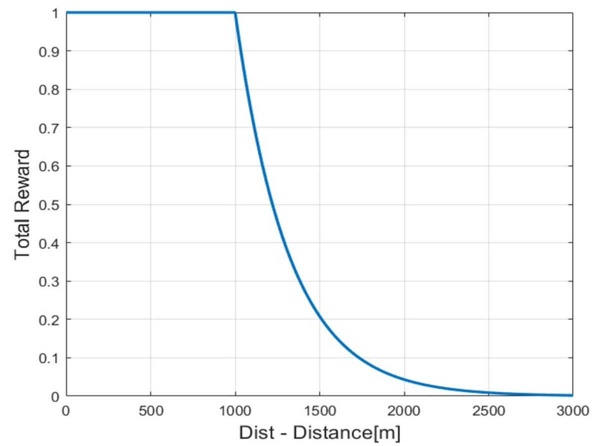
**Initial States:**

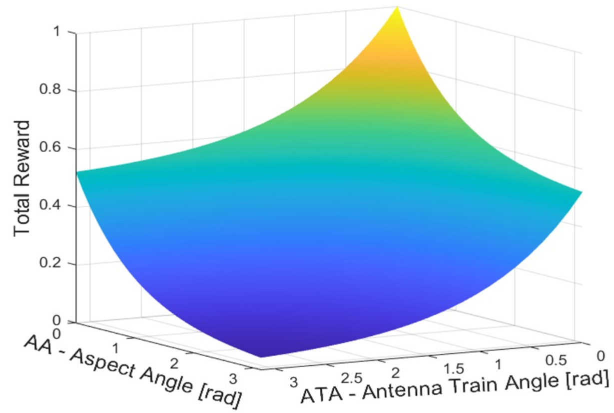
- Fixed orientation
- Fixed relative speed

**Enemy Type:**

- Random Motion



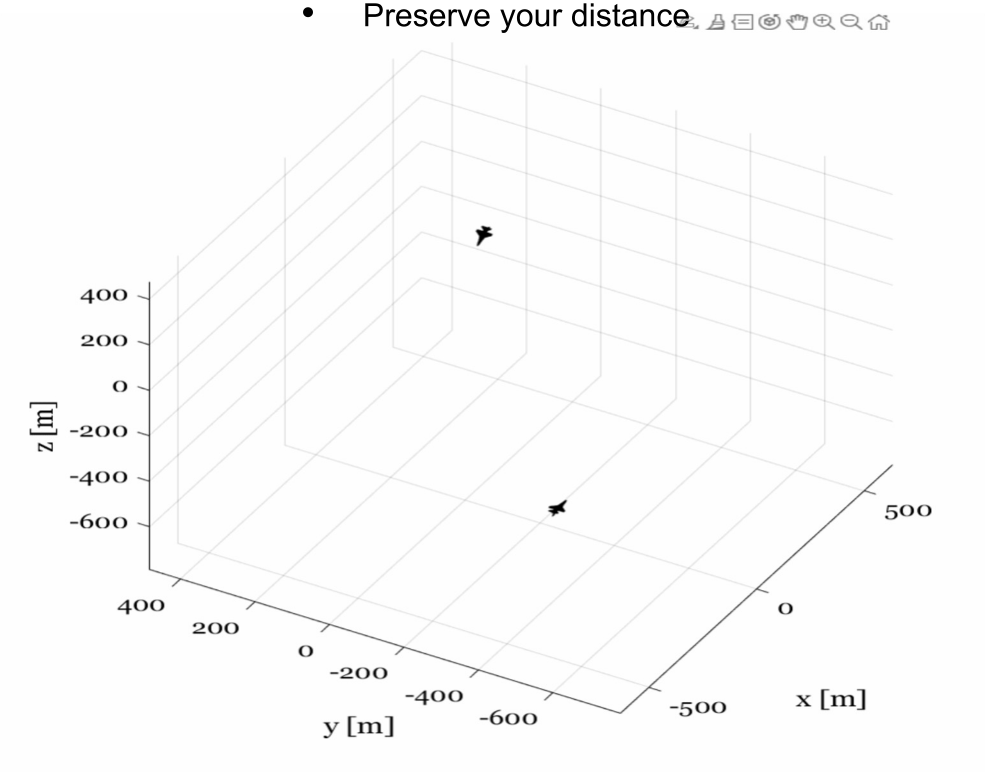



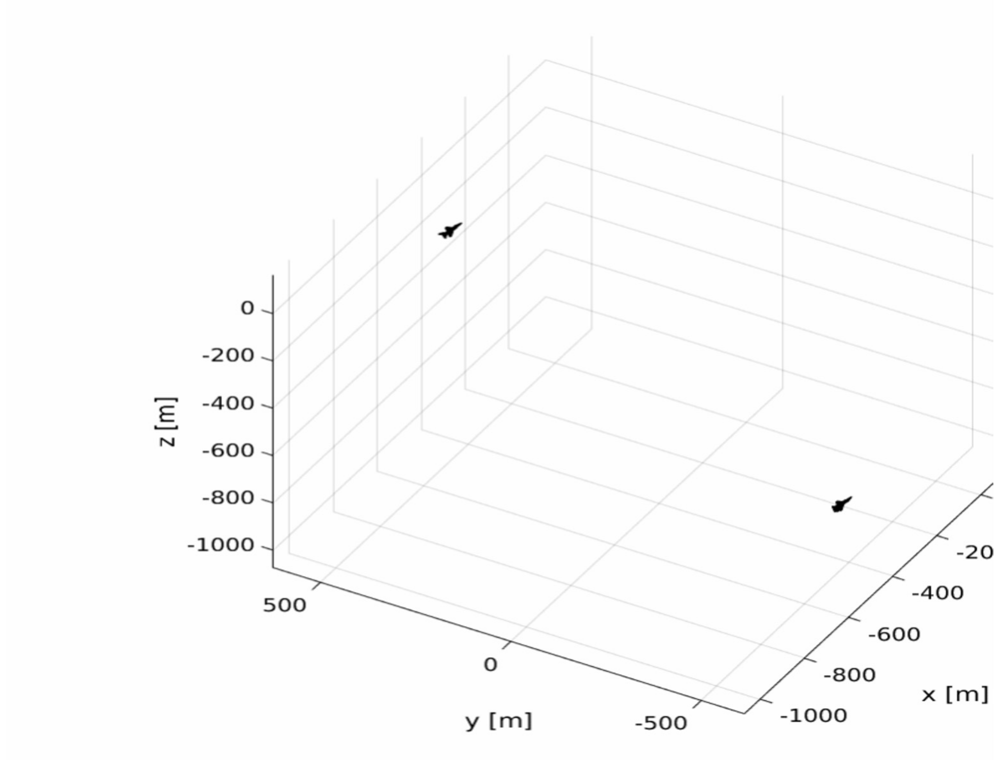



# Tactic Set 1 Results

**Rewards:**

- Stay behind
- Keep nose on target
- Preserve your distance



# Reward Functions: Tactic Set 2

**Training Details**

**Rewards:**

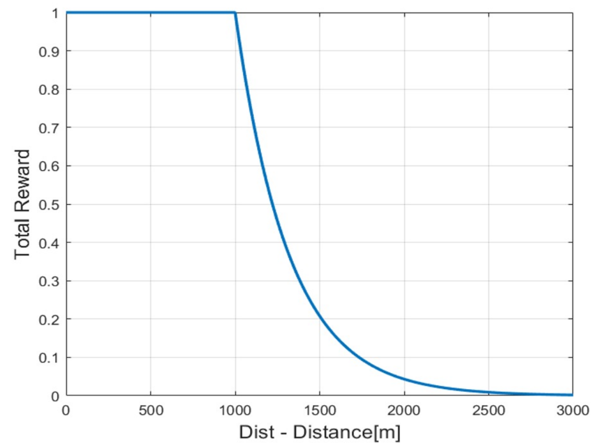
- Keep nose on target
- Preserve your distance
- Don’t let target to get behind

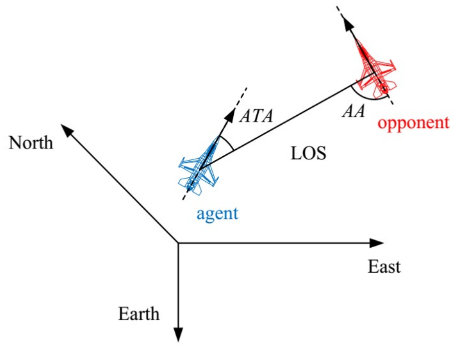
**Initial States:**

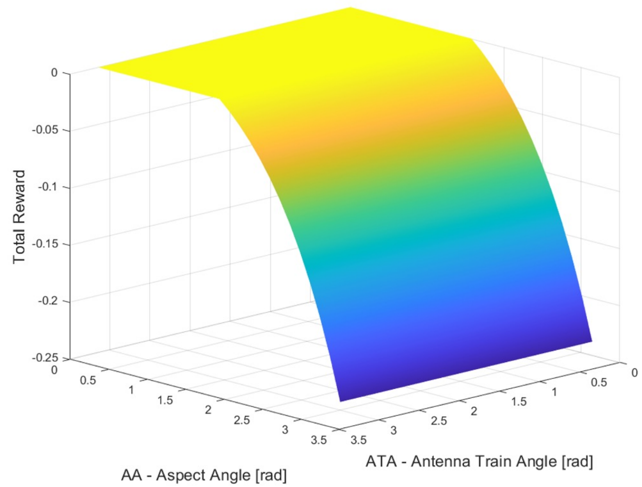
- Random orientation
- Random speed
- Random position

**Enemy Type:**

- Tactic Set 1



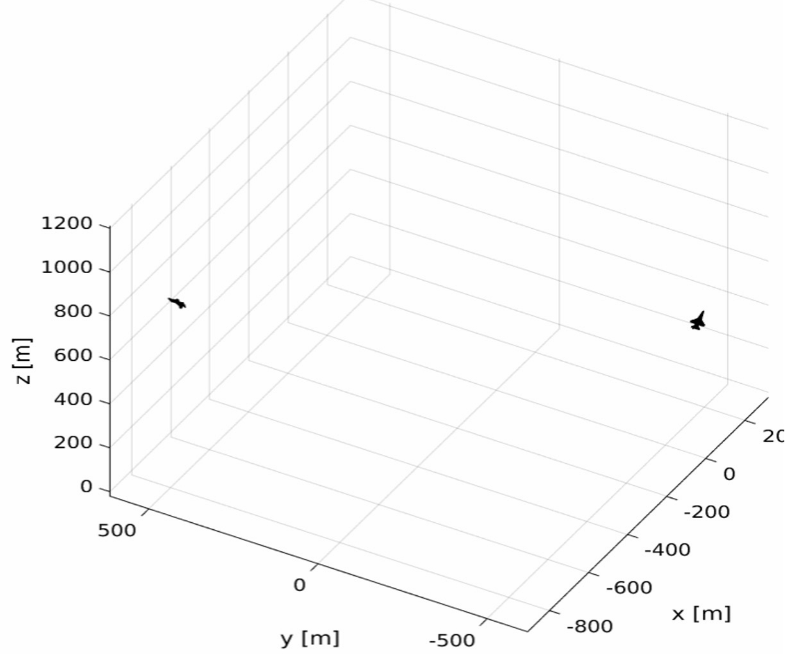



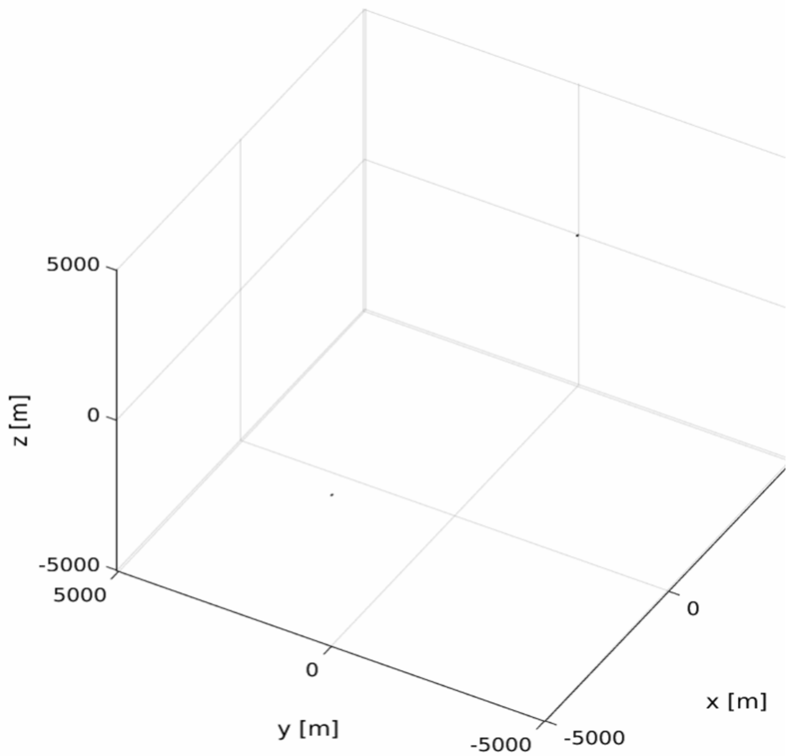



# Tactic Set 2 Results

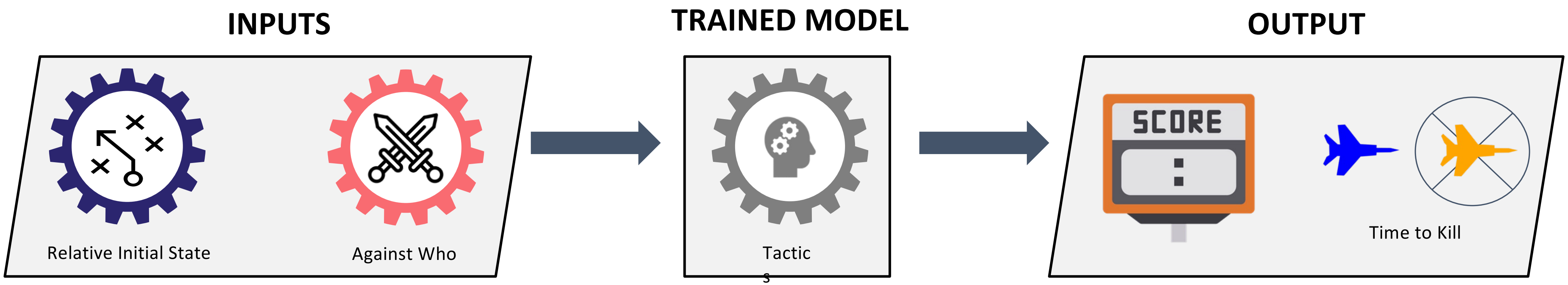
**Rewards**

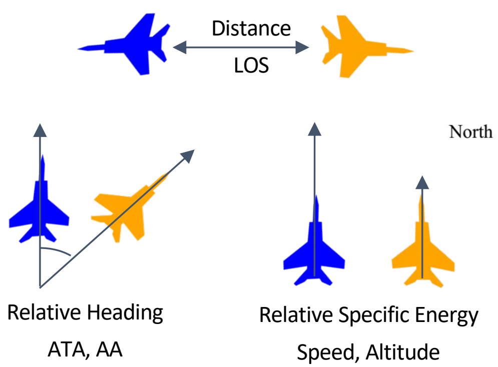
- Keep nose on target
- Preserve your distance
- Don’t let target to get behind

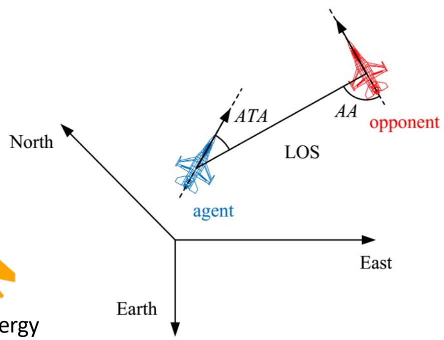


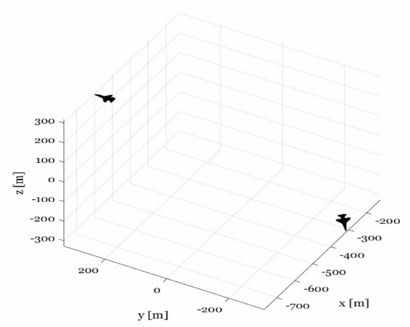



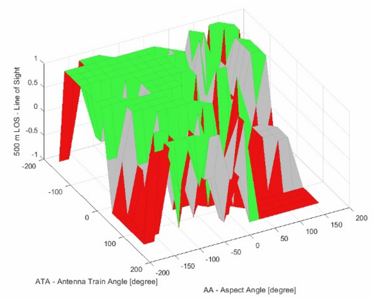
# Winning Assessment with Changing Combat Conditions



# Winning Assessment with Changing Test Condition

**Training Details**

**Rewards:**

- Stay behind
- Keep nose on target
- Preserve your distance

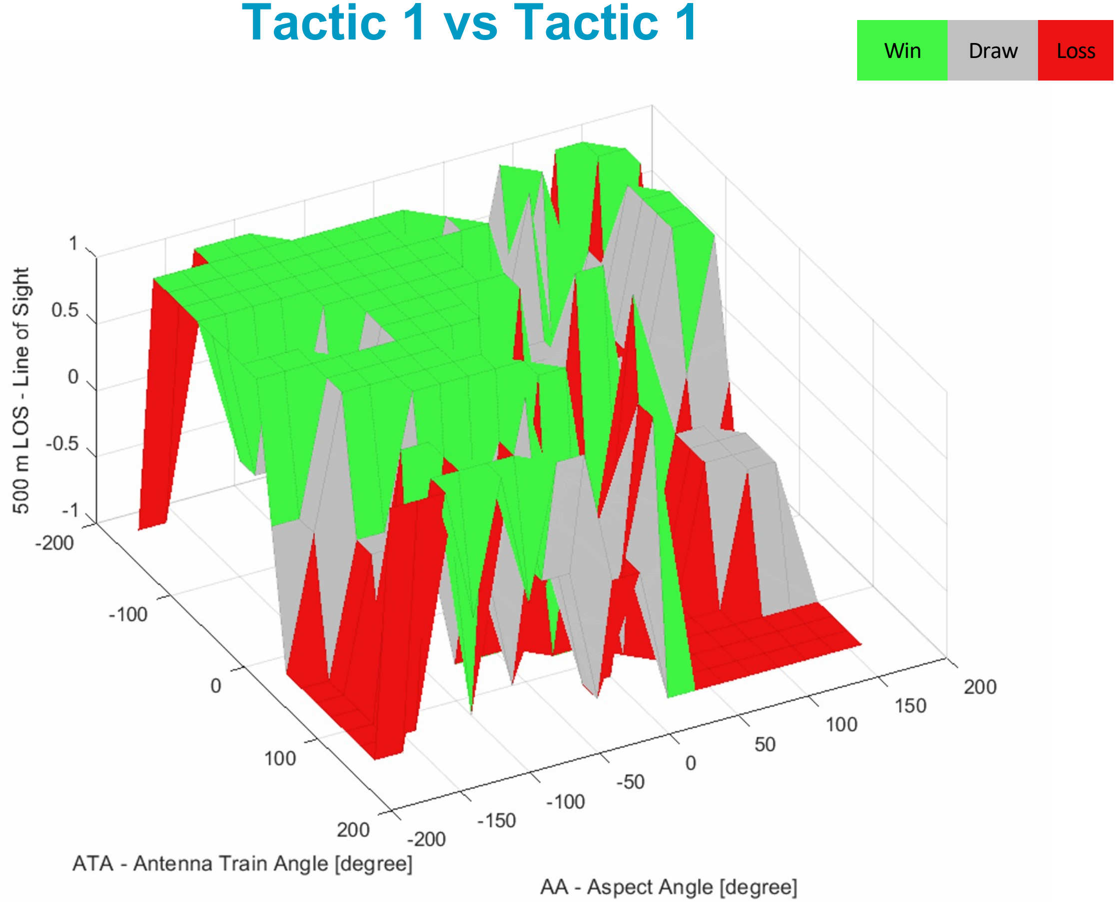
**Initial States:**

- Same heading
- Fixed always behind

**Enemy Type:**

- Random Motion

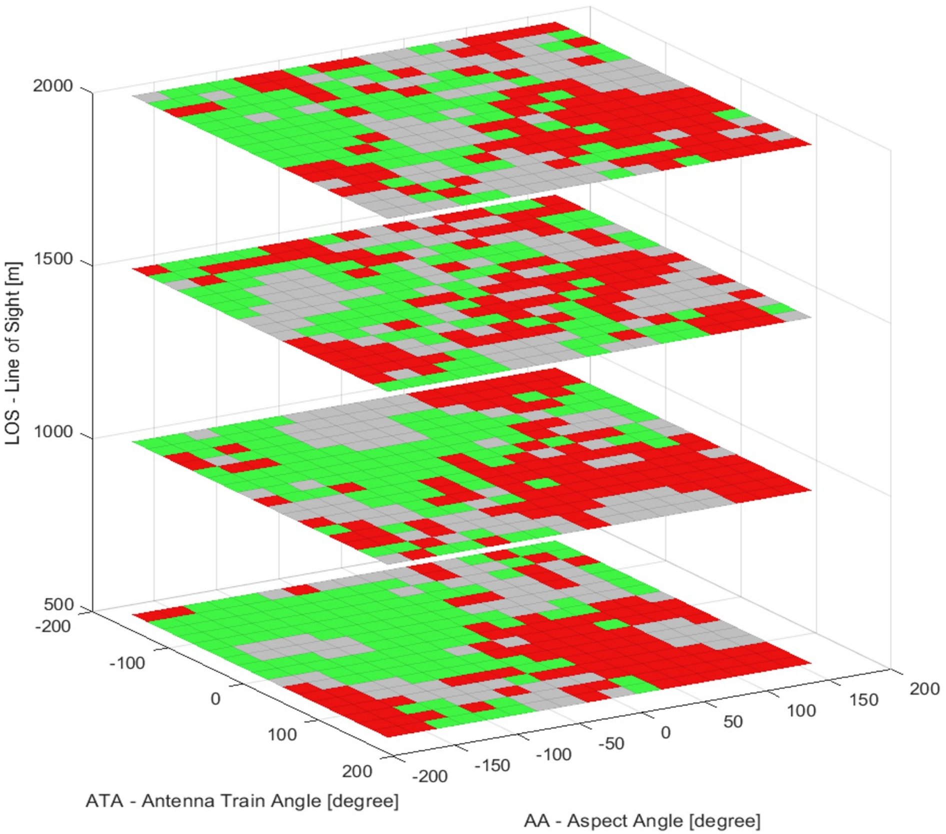
## Tactic 1 vs Tactic 1



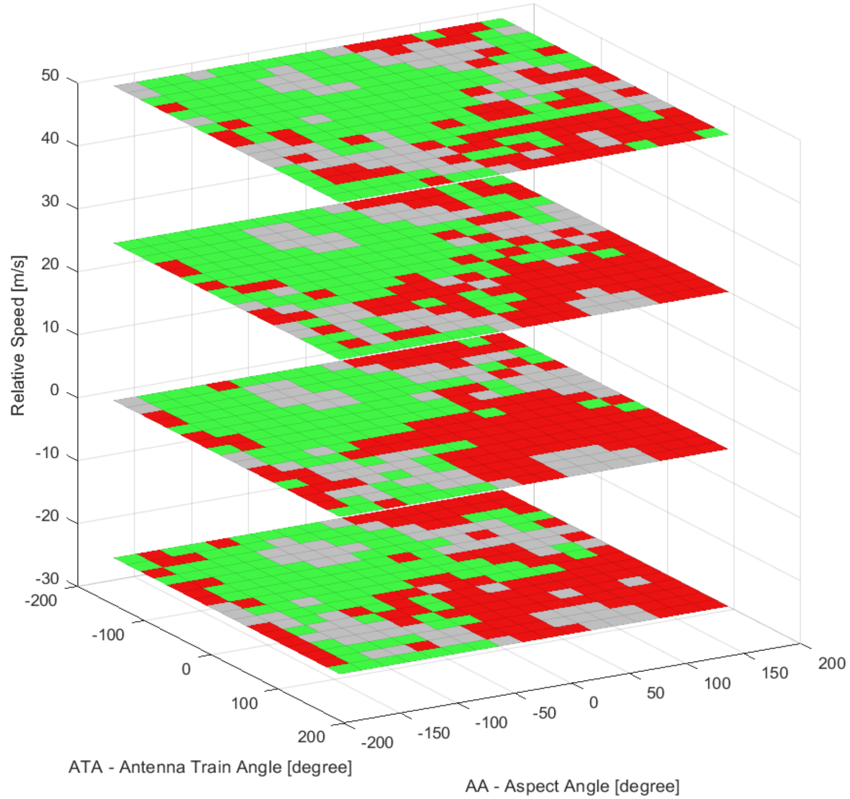
# Winning Assessment with Changing Test Condition

## Tactic 1 vs Tactic 1

**Aspect Angle (AA) - Antenna Train Angle (ATA) - Line of Sight (LOS)**




**Aspect Angle (AA) - Antenna Train Angle (ATA) - Relative Speed**

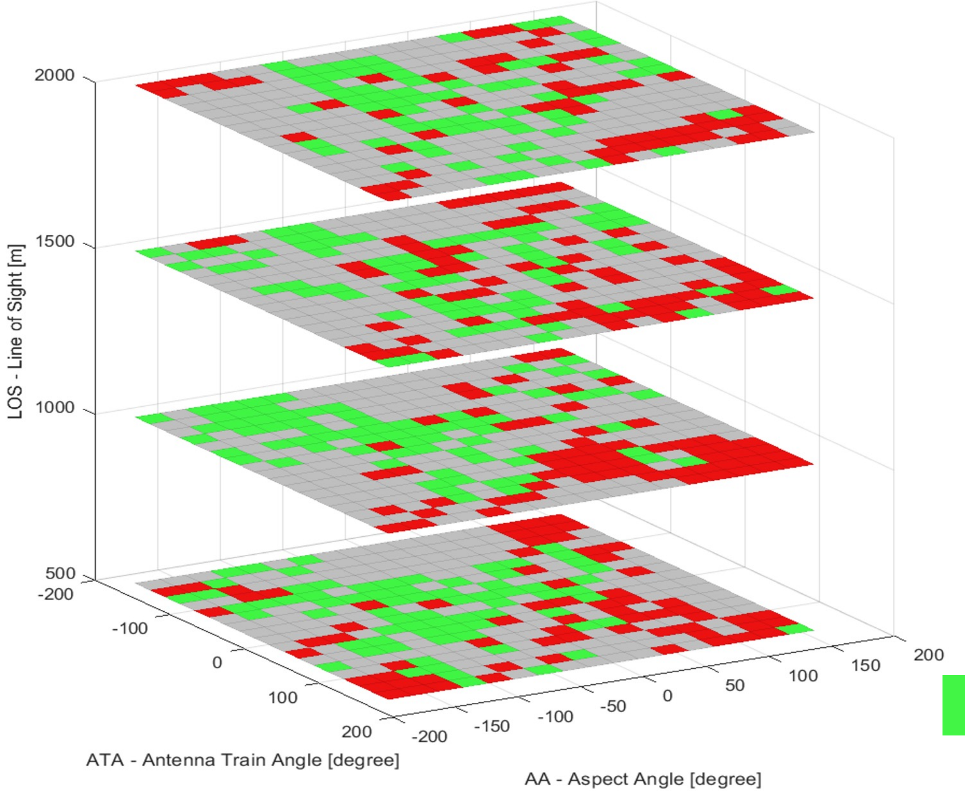



| Win | Draw | Loss |
|---|---|---|

# Winning Assessment with Changing Test Condition

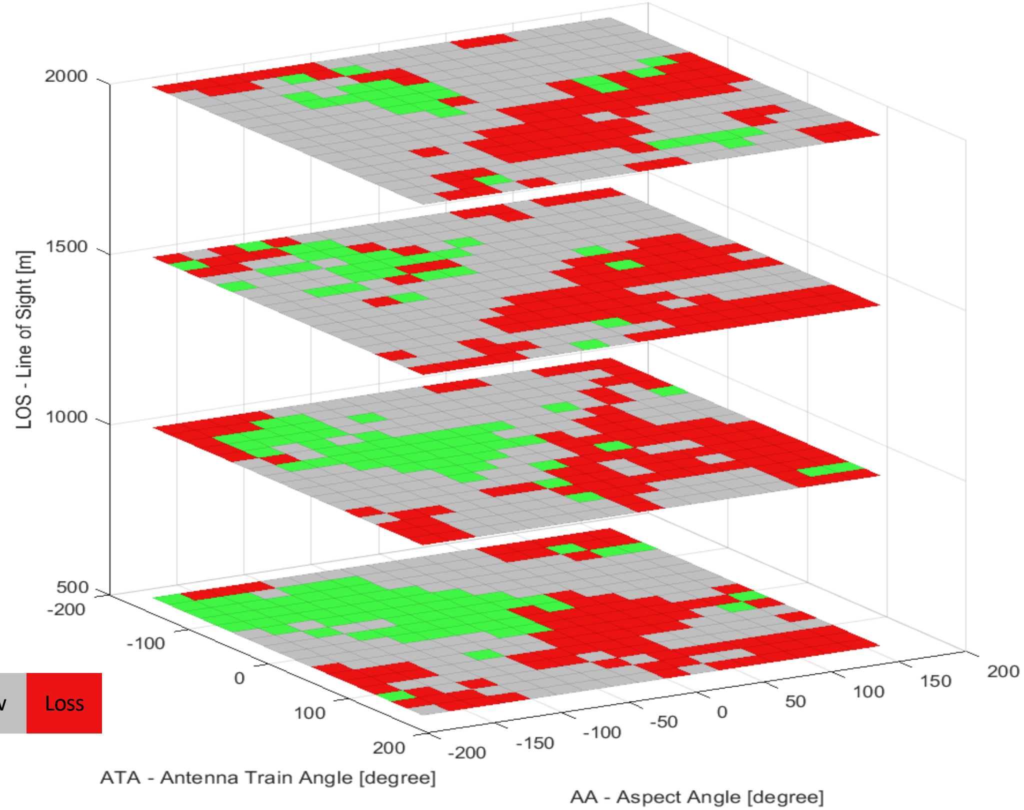
## Tactic 2 vs Tactic 2

**Aspect Angle (AA) - Antenna Train Angle (ATA) - Line of Sight (LOS)**




## Tactic 2 vs Tactic 1

**Aspect Angle (AA) - Antenna Train Angle (ATA) - Line of Sight (LOS)**

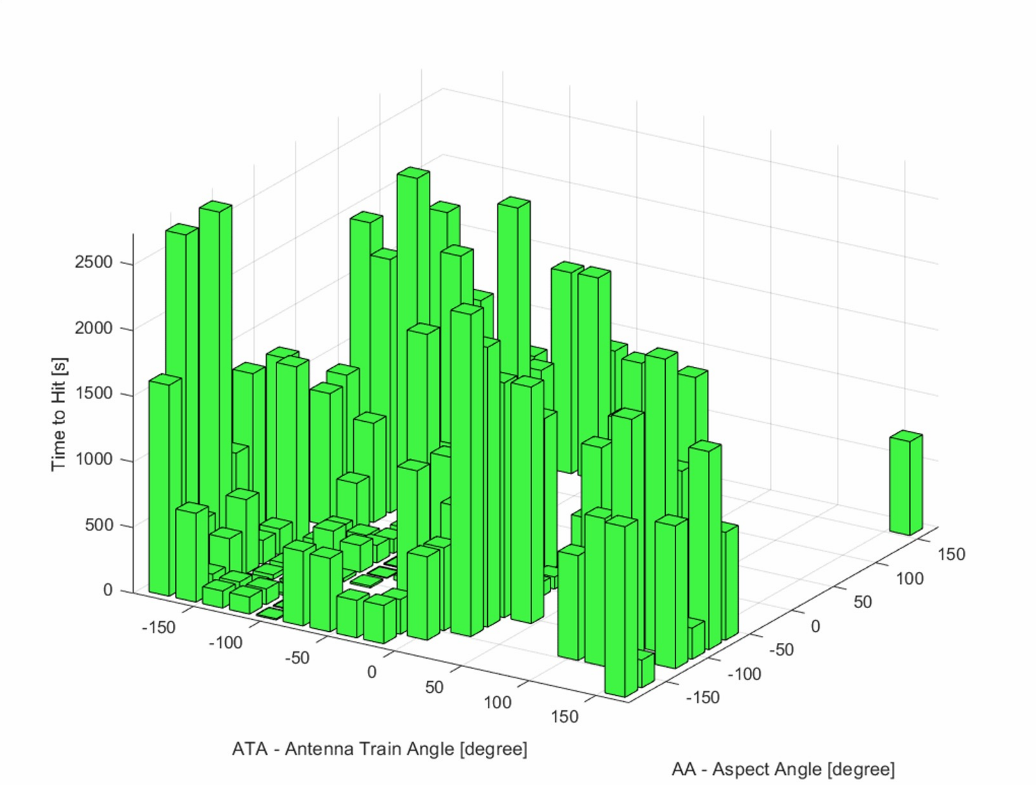



Win Draw Loss

# Winning Assessment with Changing Test Condition

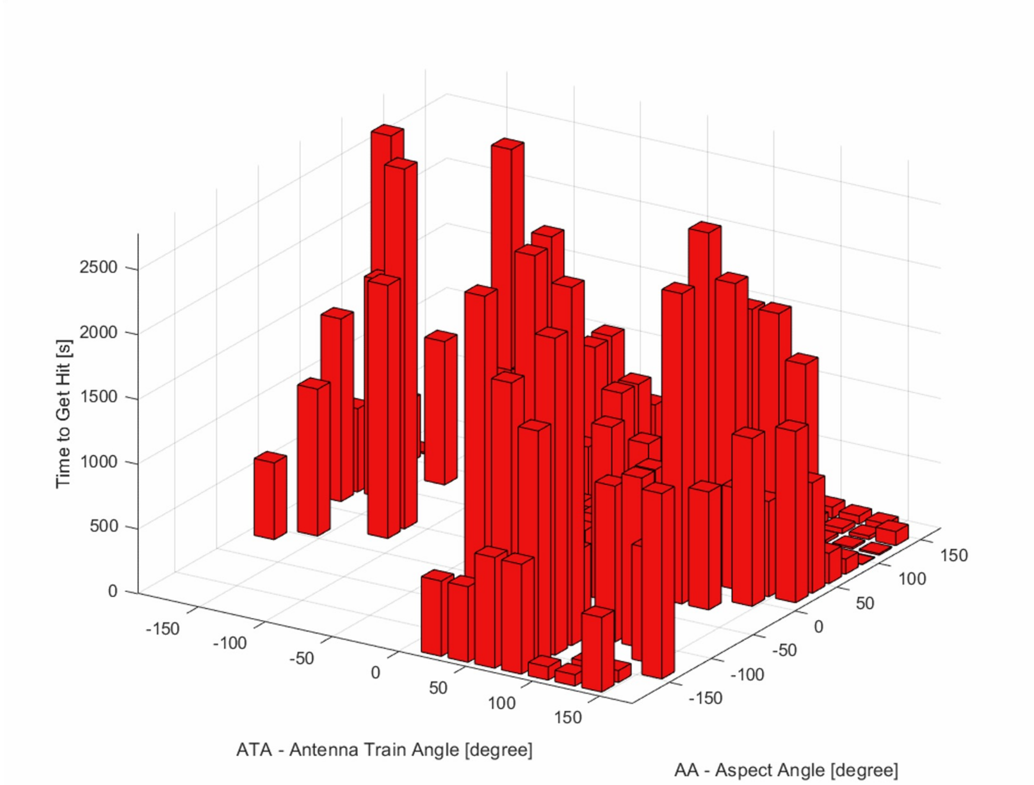
## Time To Hit

**Aspect Angle (AA) - Antenna Train Angle (ATA)**

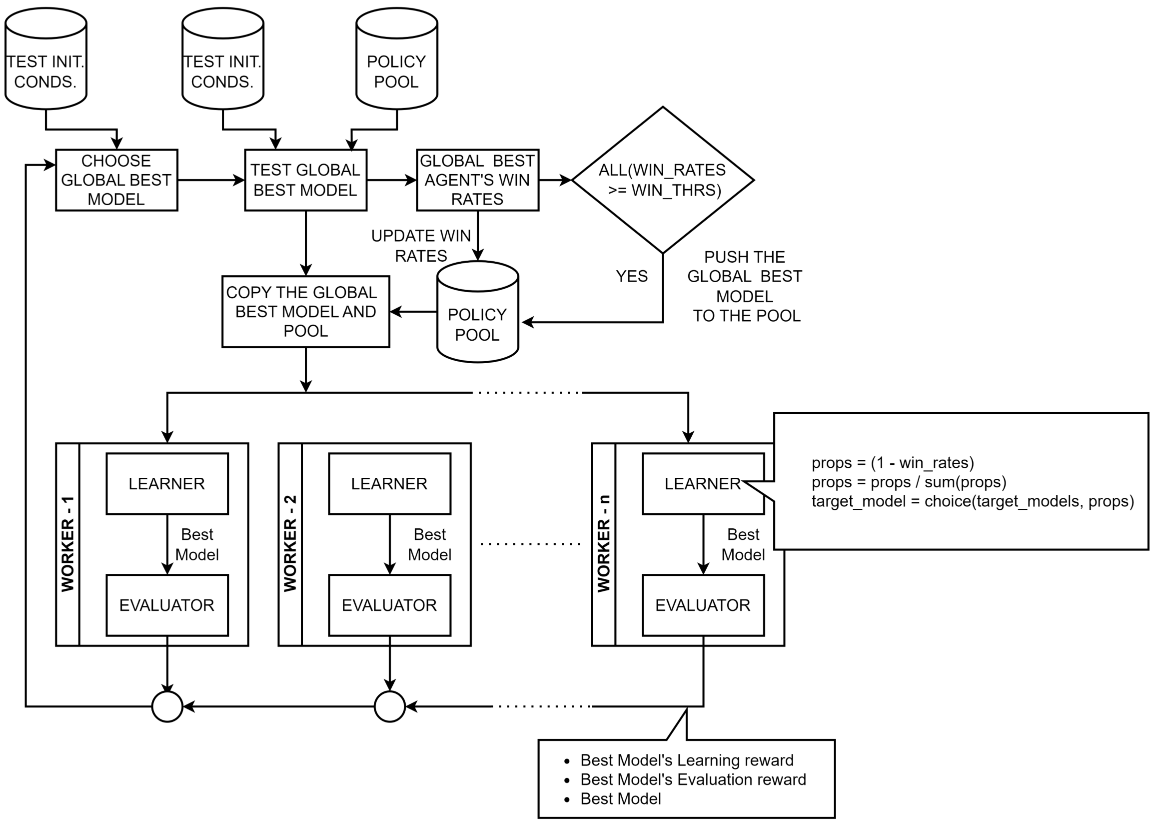


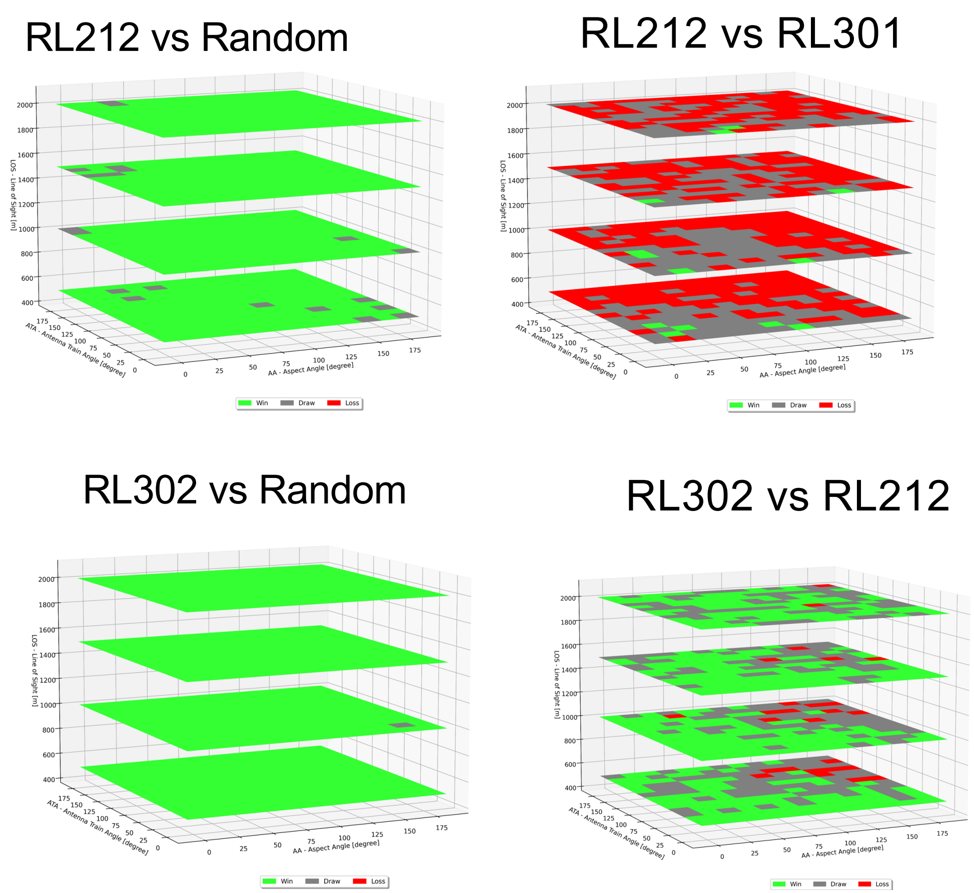

## Time to Get Hit

**Aspect Angle (AA) - Antenna Train Angle (ATA)**



# Designing a Super Agent



# Reward Functions: Tactic Set 3

## Training Details

**Rewards:**

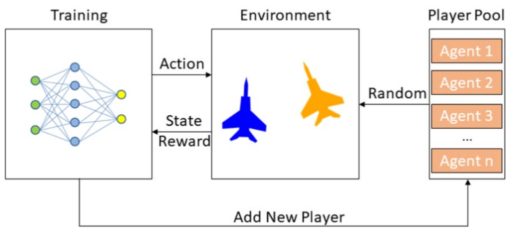
- Keep nose on target
- Preserve your distance
- Don’t let target to get behind

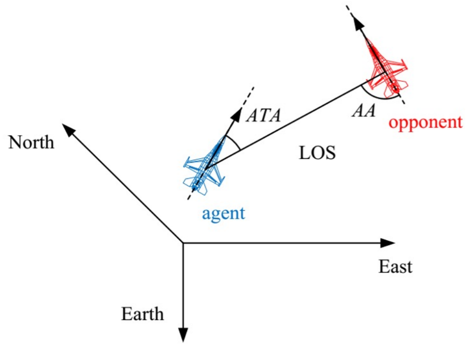
**Initial States:**

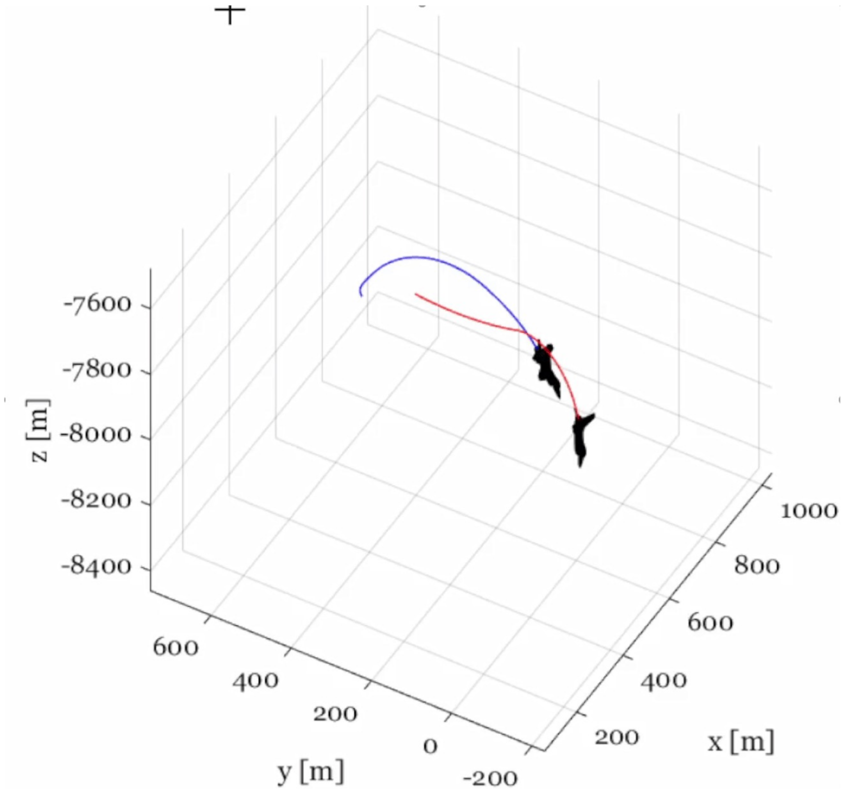
- Random orientation
- Random speed
- Random position

**Enemy Type:**

- Pool of Agents

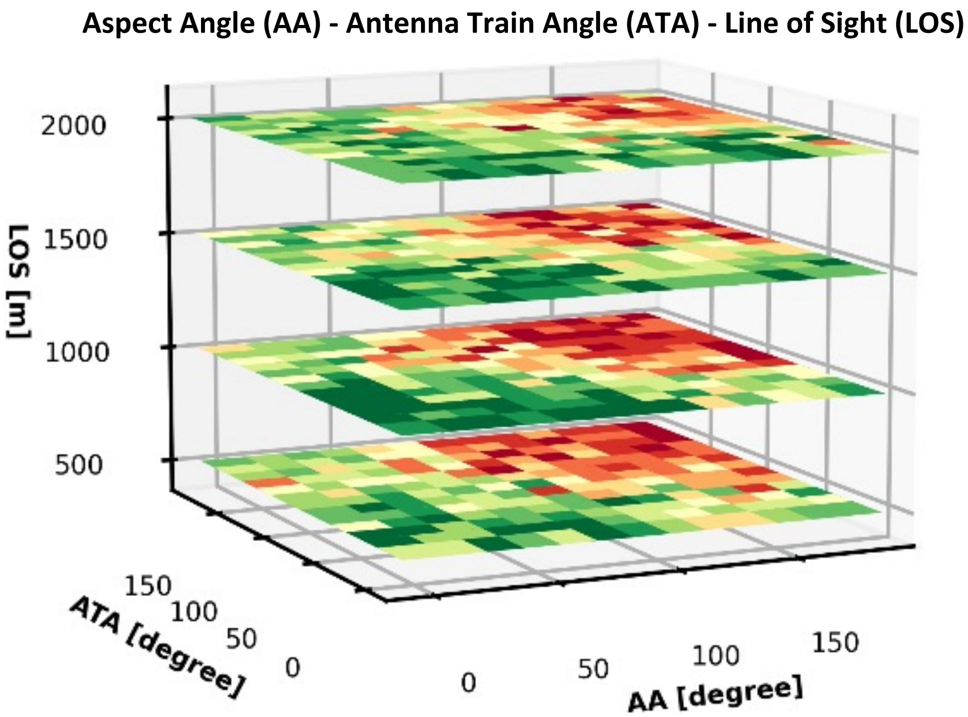


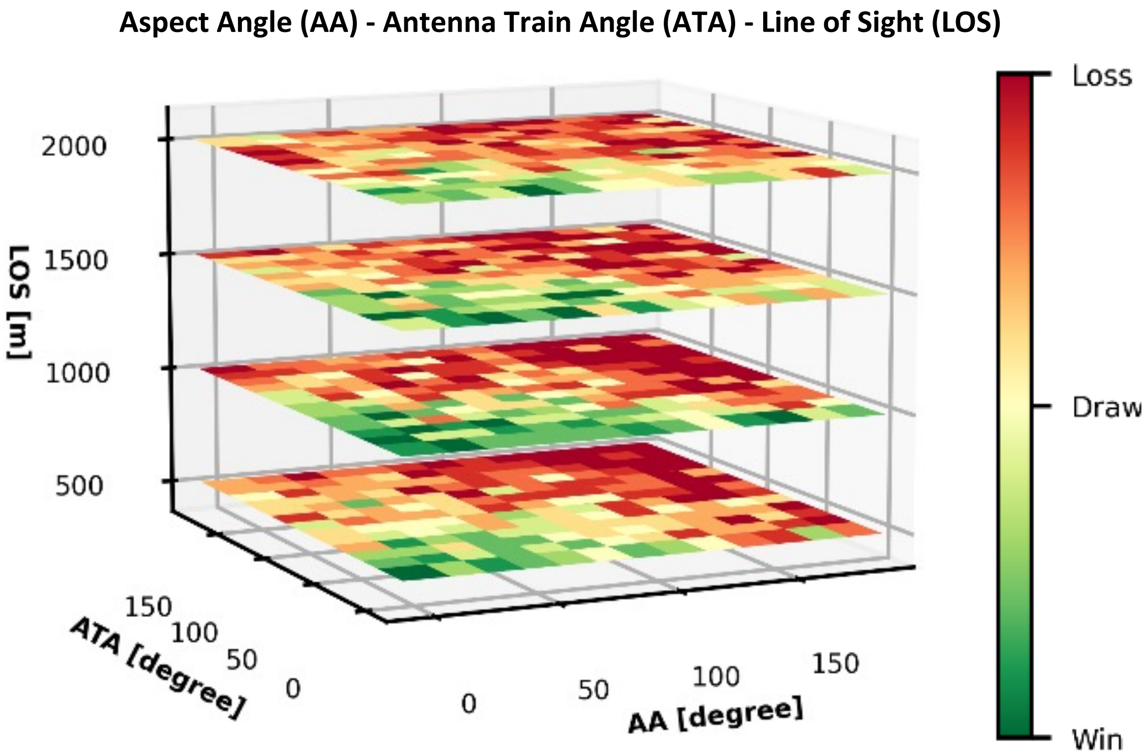



# How does uncertainty come into play?

## Tactic 1 vs Tactic 3

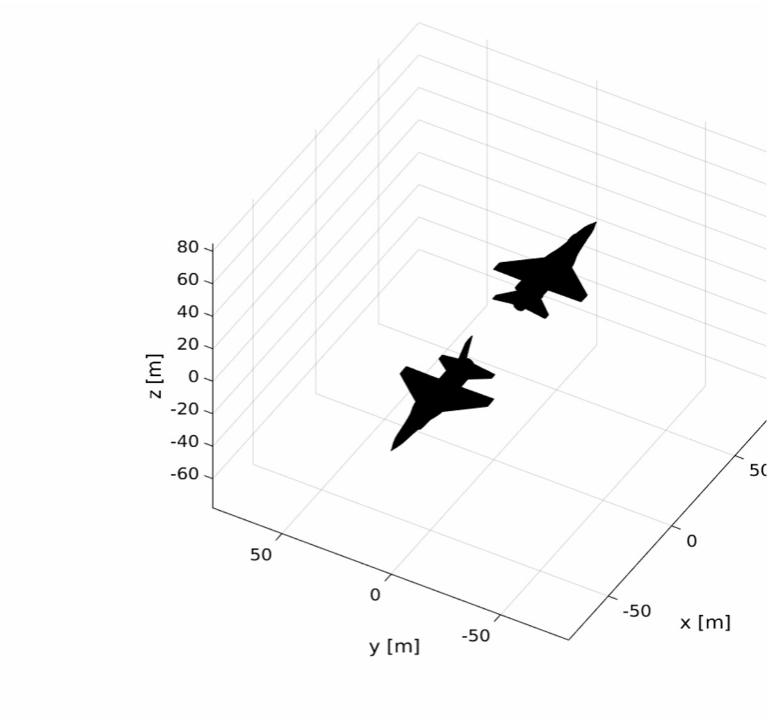



## Tactic 2 vs Tactic 3

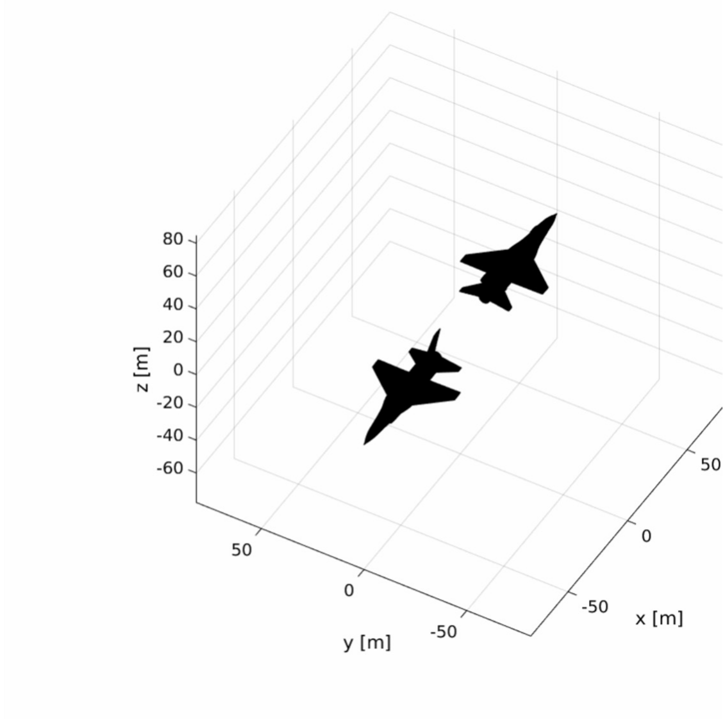


# Generalization of Trained Agent

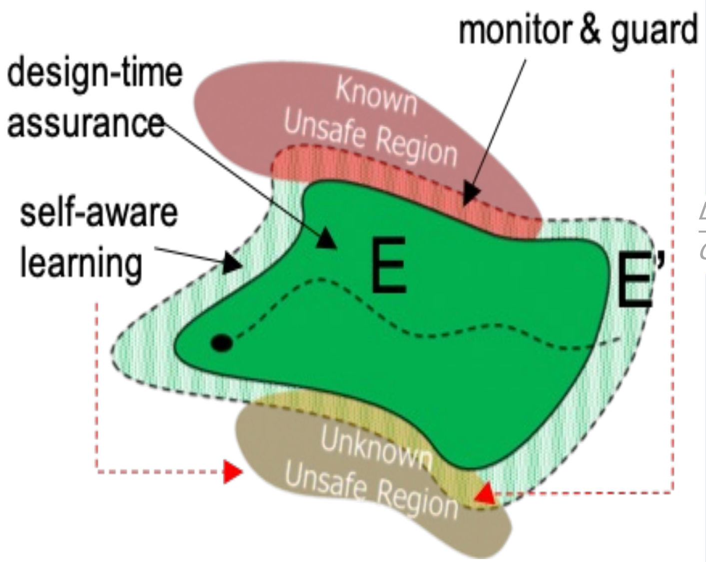
**Default Turn Rate - 20 degree/sec**

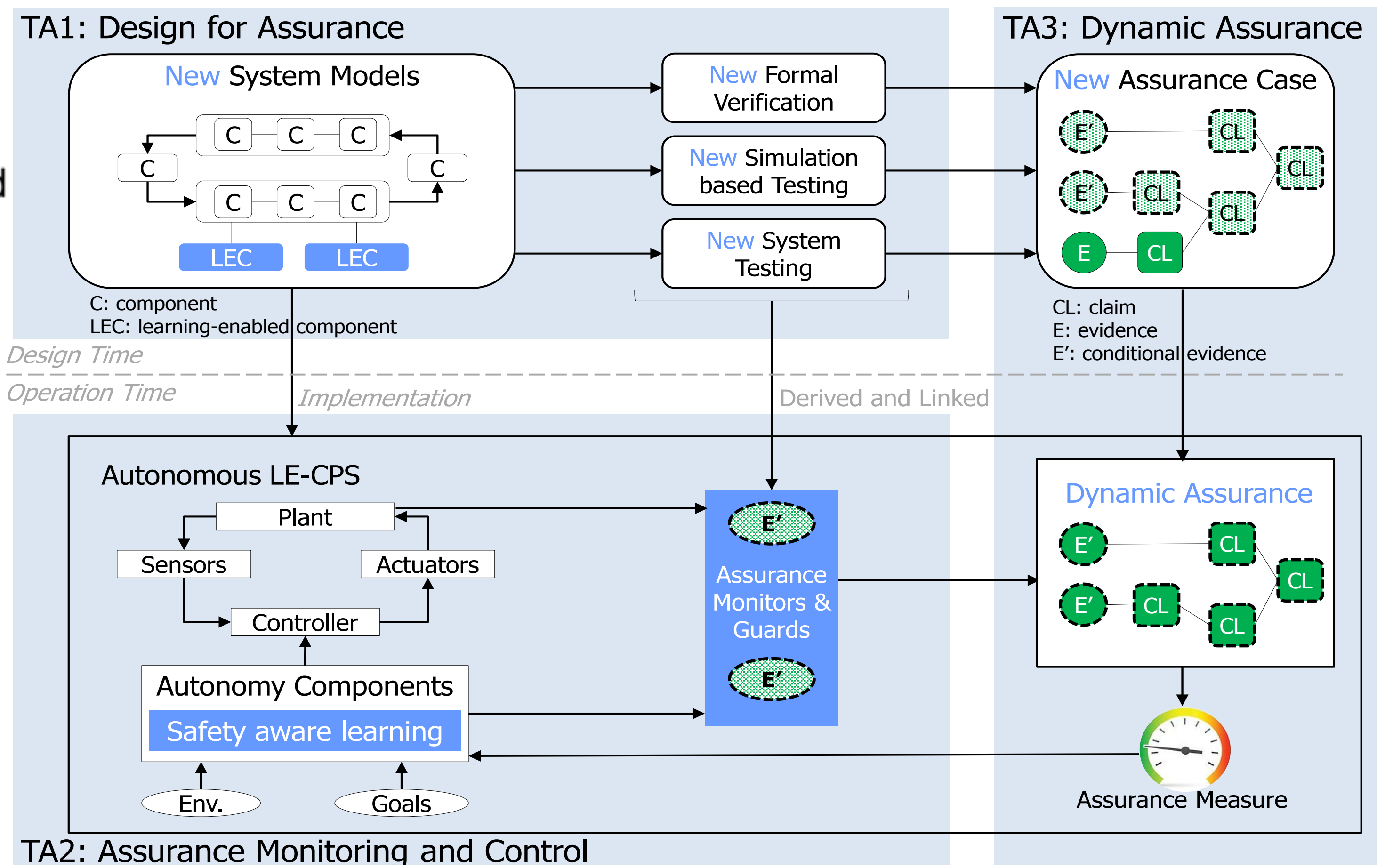



**Turn Rate Doubled - 40 degree/sec**



# Learning Enabled Explainable AI



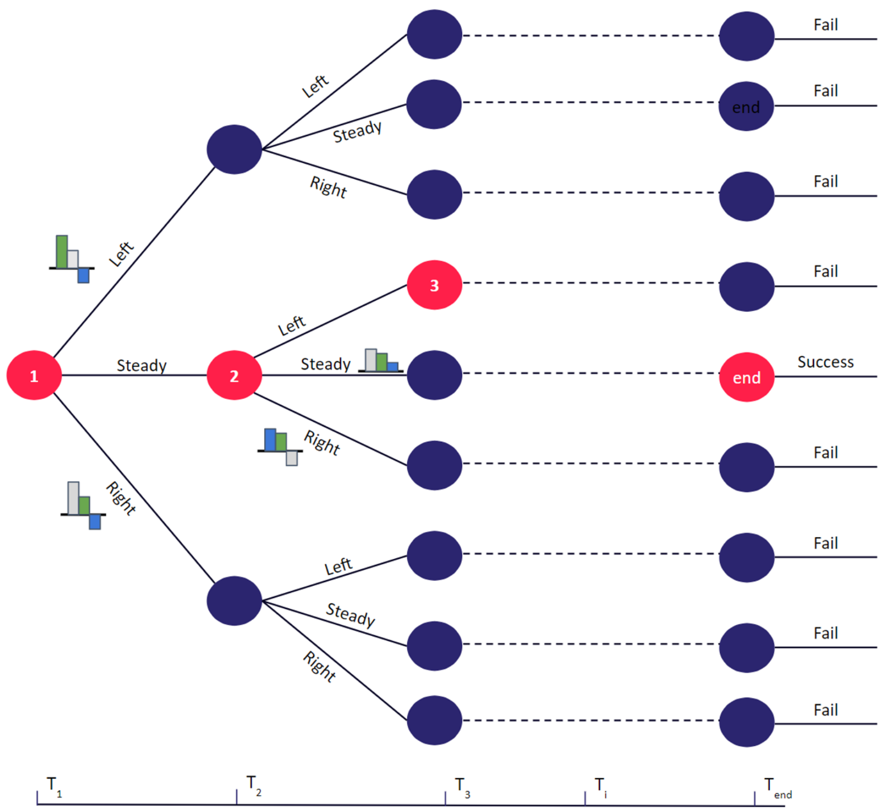





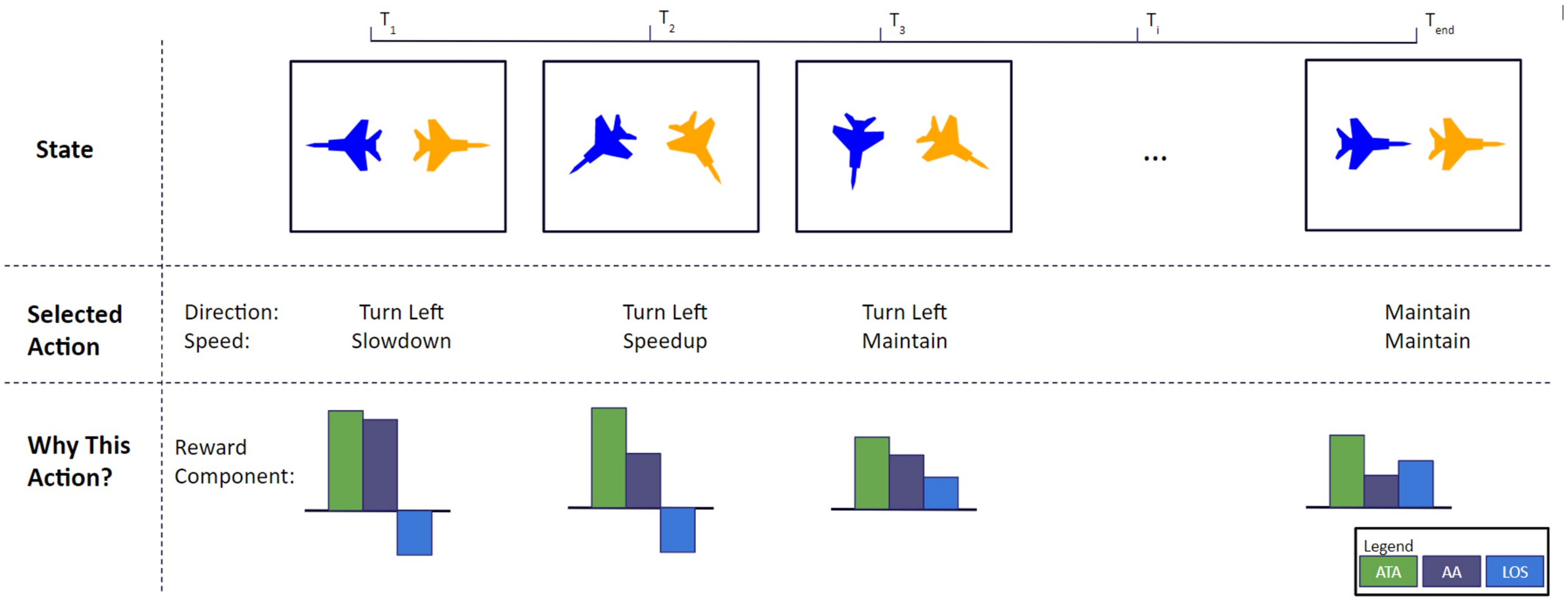

Ref : Darpa Assured Autonomy

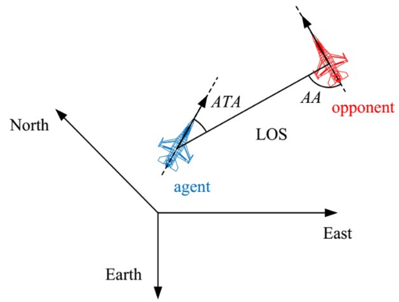
# Explainability Perspective

- Discrete action space can be displayed in a tree graph and can be link with agent’s expected reward at each state.
- Simplified version of action space tree with 3 actions
- Bar graph at each action from red to blue node represents advantages of successful action compare to the non-selected action.



# Explainability Perspective

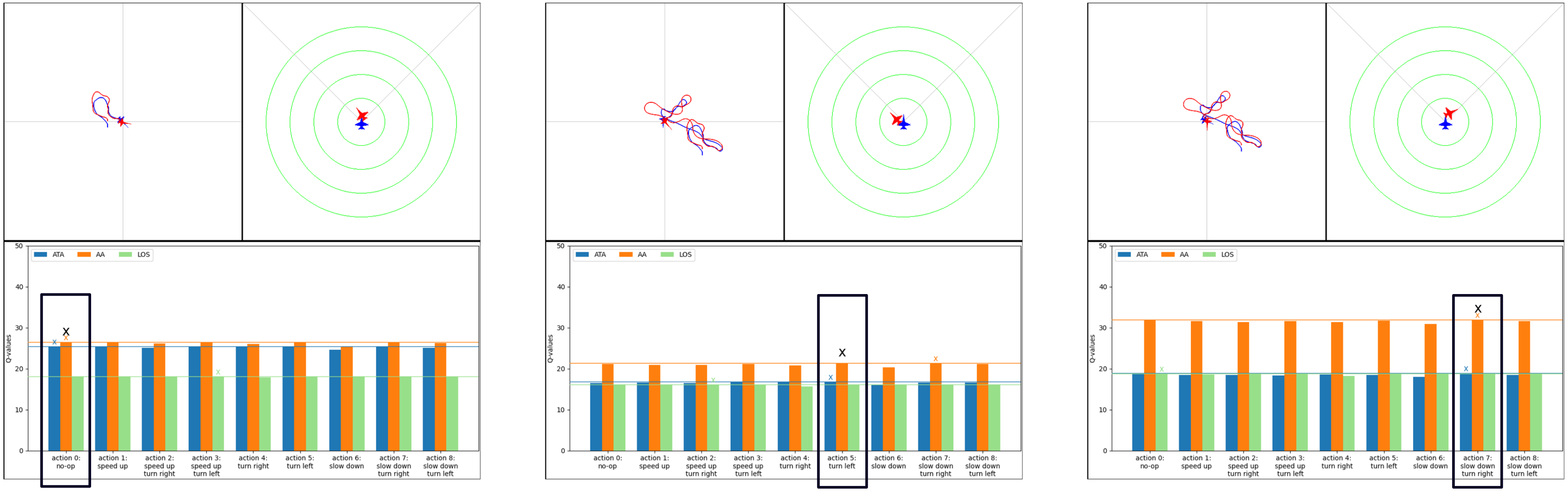



- Why did you choose the current action?
  - Shows which reward component the agent expected to have bigger return, thus deemed most important?
  - Ex: I choose action speed up, because I expected higher return for AA component.
- Why not an another action?
  - Give insight into why current action is more advantages or disadvantages than another action.

# Explainability Perspective

- Step by step explanation.
  - Which reward type(ATA, AA, LOS) contributed to current action.
  - Allow us to better evaluate agent's tactic and debug training process.



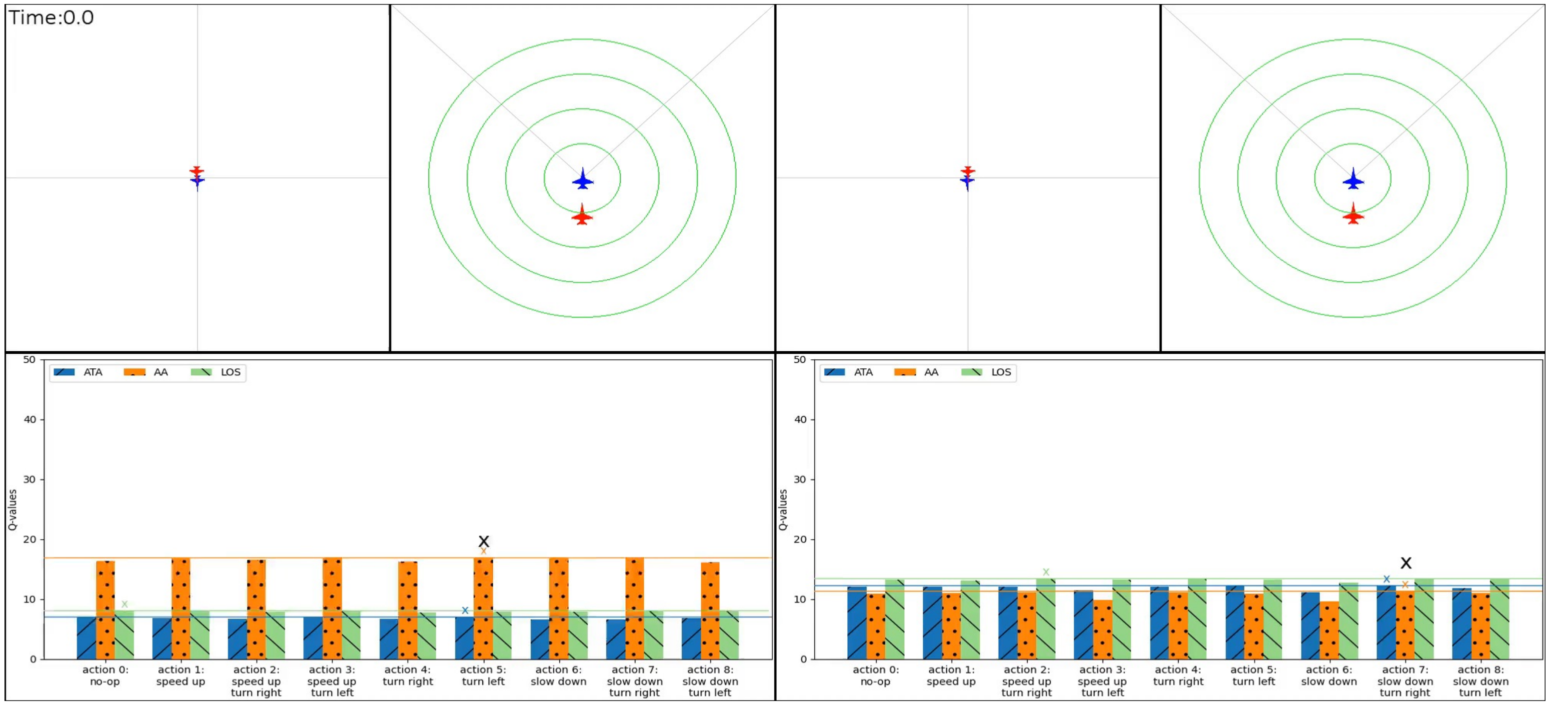
# Explainability Perspective: Debugging

## Left

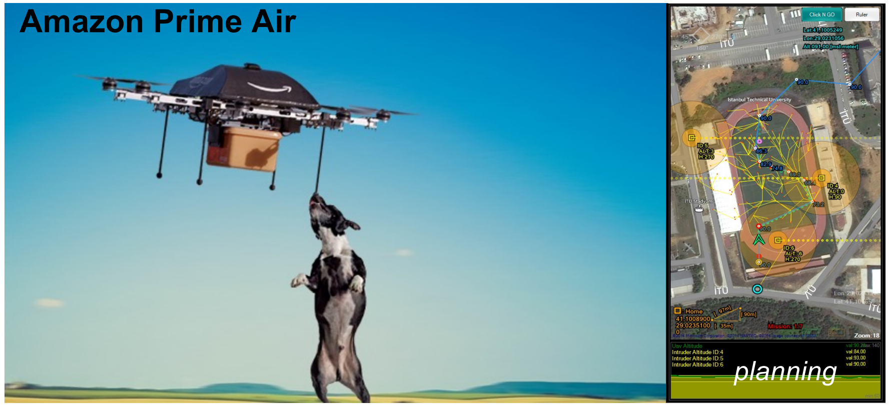
- Default environment settings.

## Right

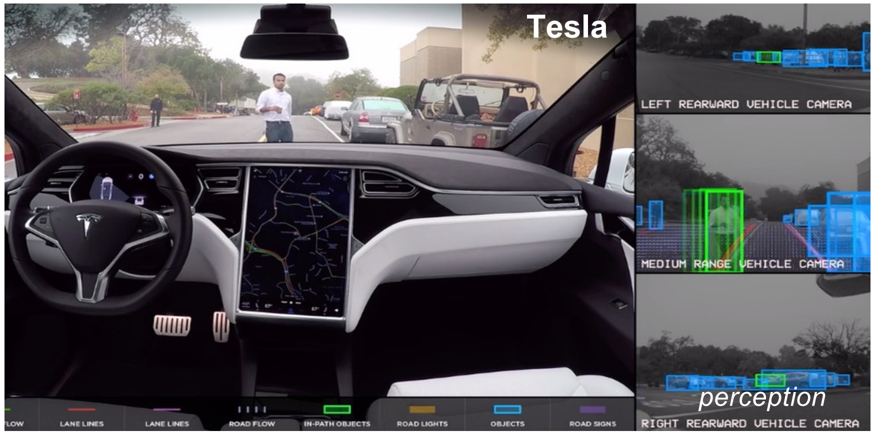
- Target action update period increased.
- Starting position range doubled.



# Autonomous Systems and Artificial Intelligence in a Customer-Led World






**New technology solutions lead to new Business Models, and Autonomy and AI are key enablers**

# Thank you..

Cranfield University

www.cranfield.ac.uk

T: +44 (0)1234 750111

@cranfielduni
@cranfielduni
/cranfielduni