Robust Reinforcement Learning Differential Game Guidance in Low-Thrust, Multi-Body Dynamical Environments

A Zero-Sum Reinforcement Learning Approach in Three-Body Dynamics

Ali Baniasad

Supervisor: Dr. Nobahari

Department of Aerospace Engineering

Sharif University of Technology

Outline

1. Reinforcement Learning
2. Multi-Agent Reinforcement Learning (MARL)
3. Results
4. Environment

Reinforcement Learning Overview

- **Definition:** A type of machine learning where an agent learns to make decisions by taking actions in an environment to maximize cumulative reward.

- **Key Components:**
  - **Agent:** The learner or decision maker.
  - **Environment:** The external system with which the agent interacts.
  - **Actions:** Choices made by the agent to influence the environment.
  - **Rewards:** Feedback from the environment based on the agent's actions.

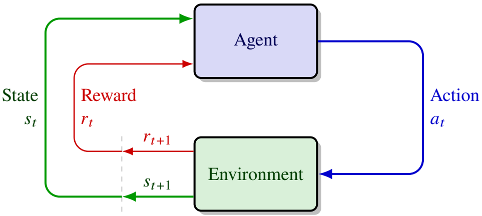


Figure: Agent-Environment Interaction Loop

State and Observations

- **State (s):** Complete description of the environment's condition
- **Observation (o):** Partial description of the state
    - May not contain all information
    - In fully observable environments: $s = o$
- **Action Space (a):** Set of all possible actions an agent can take
    - Can be discrete (finite set) or continuous (bounded range)

Policy

- **Policy:** Rules that an agent uses to decide which actions to take
  - **Types:**
    - **Deterministic:** $a_t = \mu(s_t)$
    - **Stochastic:** $a_t \sim \pi(\cdot|s_t)$
  - **Parameterized Policy:** Output is a function of policy parameters (neural network weights)
    - $a_t = \mu_\theta(s_t)$ or $a_t \sim \pi_\theta(\cdot|s_t)$
    - Parameters θ are optimized during learning

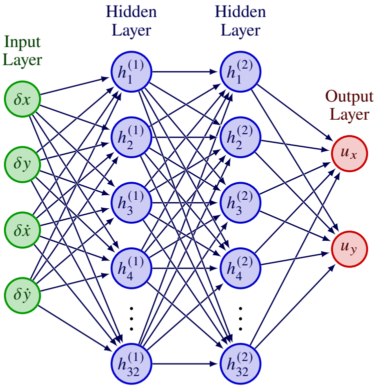


Figure: Policy Neural Network Structure

Trajectory and Reward

Trajectory:

- Sequence of states and actions: $\tau = (s_0, a_0, s_1, a_1, \ldots)$
- State transition: $s_{t+1} = f(s_t, a_t)$

Reward:

- $r_t = R(s_t, a_t, s_{t+1})$ or $r_t = R(s_t, a_t)$
- **Return:** Total accumulated reward
- Finite horizon: $R(\tau) = \sum_{t=0}^{T} r_t$
- Discounted: $R(\tau) = \sum_{t=0}^{\infty} \gamma^t r_t$

Value and Action-Value Functions

- **Value Function:** Expected return when following a policy

State Value Function:

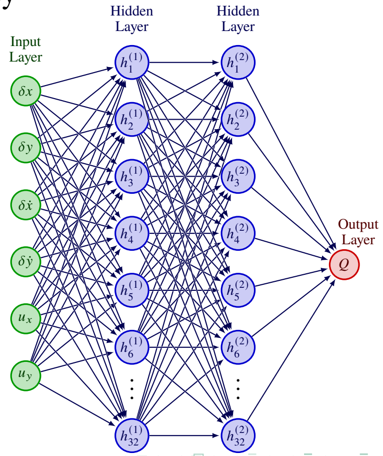
$$V^{\pi}(s) = \underset{\tau \sim \pi}{\mathbb{E}}\left[R(\tau) | s_0 = s\right]$$

Action-Value Function:

$$Q^{\pi}(s, a) = \underset{\tau \sim \pi}{\mathbb{E}}\left[R(\tau) | s_0 = s, a_0 = a\right]$$

Advantage Function:

$$A^{\pi}(s, a) = Q^{\pi}(s, a) - V^{\pi}(s)$$



Optimal Value Functions

Optimal State Value Function:

$$V^*(s) = \max_{\pi} V^{\pi}(s)$$

Optimal Action-Value Function:

$$Q^*(s,a) = \max_{\pi} Q^{\pi}(s,a)$$

Optimal Value Bellman Equation:

$$V^*(s) = \max_{a} \mathop{\mathrm{E}}_{s' \sim P} [r(s,a) + \gamma V^*(s')]$$

Optimal Q Bellman Equation:

$$Q^*(s,a) = r(s,a) + \gamma \mathop{\mathrm{E}}_{s' \sim P} \left[\max_{a'} Q^*(s',a') \right]$$

Key insight: The optimal policy π^* is greedy with respect to Q^*:

$$\pi^*(s) = \arg\max_{a} Q^*(s,a)$$

Bellman Equations

For Policy Value Functions:

$$V^{\pi}(s) = \underset{\substack{a \sim \pi \\ s' \sim P}}{\mathrm{E}} \left[r(s,a) + \gamma V^{\pi}(s') \right]$$

$$Q^{\pi}(s,a) = r(s,a) + \gamma \underset{s' \sim P}{\mathrm{E}} \left[\underset{a' \sim \pi}{\mathrm{E}} \left[Q^{\pi}(s',a') \right] \right]$$

For Optimal Value Functions:

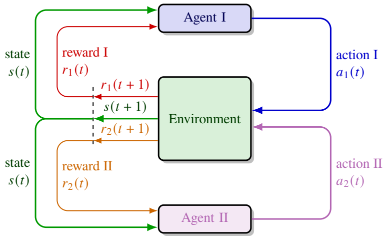
$$V^{*}(s) = \max_{a} \underset{s' \sim P}{\mathrm{E}} \left[r(s,a) + \gamma V^{*}(s') \right]$$

$$Q^{*}(s,a) = r(s,a) + \gamma \underset{s' \sim P}{\mathrm{E}} \left[\max_{a'} Q^{*}(s',a') \right]$$

Key Components & Definitions

Agents: Independent decision makers sharing an environment.
Policy $\pi_i(a_i|s)$: Action distribution of agent i.
Utility / Return: $V_i^\pi(s) = \mathbb{E}_\pi[\sum_t \gamma^t r_i]$.

- Single-agent RL is a special case ($n = 1$)
- Interaction types: cooperative, competitive, mixed
- Game-theoretic view clarifies stability / equilibria
- Shared state, distinct rewards and policies
- Centralized training, decentralized execution (CTDE)



Nash Equilibrium

A policy profile $\pi^* = (\pi_1^*, \ldots, \pi_n^*)$ is Nash if:

$$V_i^{(\pi_i^*, \pi_{-i}^*)}(s) \geq V_i^{(\pi_i, \pi_{-i}^*)}(s) \quad \forall \pi_i,\ \forall i$$

Implications:

- No unilateral profitable deviation
- In zero-sum 2-player games value is unique
- Solution concepts guide stable MARL training

Zero-Sum Games

Two-player zero-sum:

$$V_1^{(\pi_1,\pi_2)}(s) = -V_2^{(\pi_1,\pi_2)}(s), \quad Q_1 = -Q_2$$

Minimax optimality:

$$V_1^*(s) = \max_{\pi_1} \min_{\pi_2} V_1^{(\pi_1,\pi_2)}(s) = \min_{\pi_2} \max_{\pi_1} V_1^{(\pi_1,\pi_2)}(s)$$

Training Goal: Find saddle point (stable policies).

- Stabilizes adversarial robustness
- Supports disturbance modeling
- Aligns with minimax control intuition

From Single-Agent to Zero-Sum Robustness

- Lift environment: $(s, a) \to (s, a_1, a_2)$
- Critic learns $Q_1(s, a_1, a_2)$; $Q_2 = -Q_1$
- Policy updates:

$$\max_{\theta_1} \mathbb{E}[Q_1], \quad \max_{\theta_2} \mathbb{E}[-Q_1]$$

- Stabilization: target networks, entropy (SAC), delay (TD3), clipping (PPO)
- Outcome: robust guidance via adversarial curriculum

Trajectory Tracking

Objective

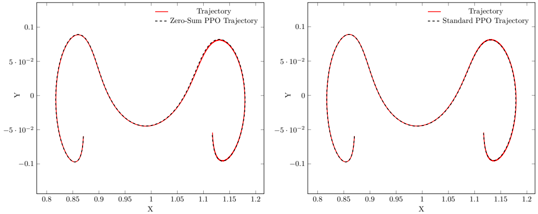
Low-thrust transfer in the planar CRTBP between Lyapunov orbits about $L_1 \rightarrow L_2$ (or vice versa).

Comparison:

- Single-Agent vs. Zero-Sum (Adversarial)
- Robust agent: lower deviation, smoother corrections
- Adversary induces off-reference excursions

Observation:

- Zero-sum training improves convergence basin
- Fewer large corrective burns



(a) Zero-Sum MARL (b) Single-Agent RL

Figure: Comparison of planar CRTBP $L_1 \rightarrow L_2$

Thrust Profile Efficiency

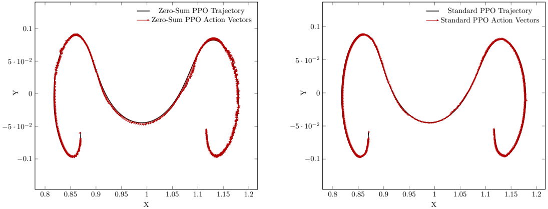
Thrust Usage:

- Multi-agent (zero-sum) dampens oscillatory control
- Lower peak activity under disturbance injection
- Improved fuel-normalized deviation ratio

Metric:

$$\text{Eff.} = \frac{\int \|\Delta s(t)\| dt}{\int \|u(t)\| dt}$$

Reduced by 12–18% (MATD3 / MASAC vs. TD3 / SAC).

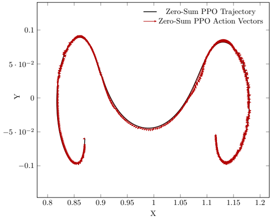


(a) Zero-Sum MARL

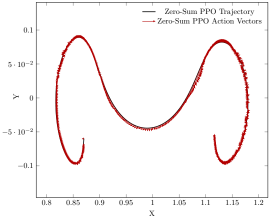
(b) Single-Agent RL

Figure: Thrust Commands

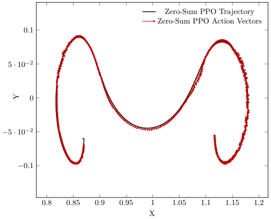
Thrust Profile Efficiency



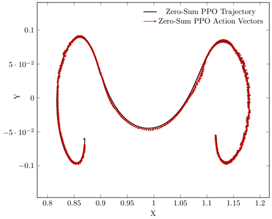
(a) 1



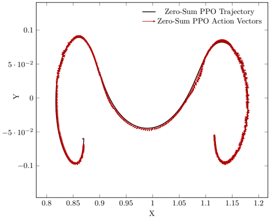
(b) 2



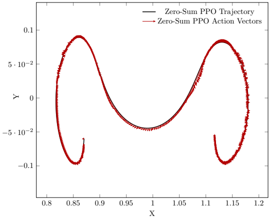
(c) 4



(d) 5

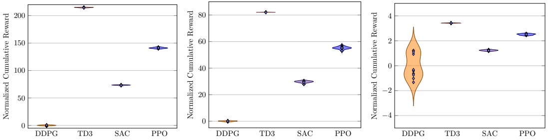


(e) 6

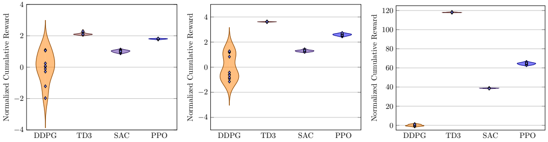


(f) 8

Return Distribution Across Robustness Scenarios MARL



(a) Random Initial Conditions (b) Actuator Disturbance (c) Model Mismatch

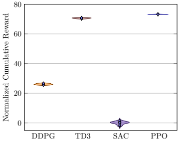


(d) Partial Observation (e) Sensor Noise (f) Time Delay

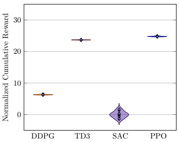
Scenario	Cumulative Return				Path Error Sum			
	DDPG	PPO	SAC	TD3	DDPG	PPO	SAC	TD3
Random Initial Conditions	-0.41	0.34	-0.02	**0.74**	4.42	4.30	4.02	**1.22**
Actuator Disturbance	-0.44	0.35	-0.02	**0.73**	4.39	4.38	4.01	**1.26**
Model Mismatch	-0.63	0.38	-0.13	**0.75**	8.85	3.57	4.78	**1.25**
Partial Observation	-1.52	0.40	-0.44	**0.71**	9.65	2.44	5.17	**1.09**
Sensor Noise	-0.60	0.37	-0.12	**0.75**	9.12	3.58	4.66	**1.25**
Time Delay	-1.19	0.17	-0.05	**0.67**	6.73	4.53	4.12	**1.21**

Scenario	Control Effort Sum				Failure Probability			
	DDPG	PPO	SAC	TD3	DDPG	PPO	SAC	TD3
Random Initial Conditions	5.11	**0.77**	1.76	3.31	**0.00**	**0.00**	0.00	**0.00**
Actuator Disturbance	4.89	**0.77**	1.71	3.07	**0.00**	**0.00**	0.00	**0.00**
Model Mismatch	5.48	**0.86**	2.37	4.32	**0.00**	**0.00**	0.20	**0.00**
Partial Observation	5.37	**1.03**	2.33	4.10	**0.00**	**0.00**	0.20	**0.00**
Sensor Noise	5.48	**0.86**	2.37	4.30	**0.00**	**0.00**	0.20	**0.00**
Time Delay	5.51	**0.76**	2.11	5.12	**0.00**	**0.00**	0.20	**0.00**

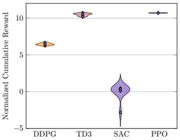
Return Distribution Across Robustness Scenarios



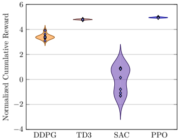
(a) Random Initial Conditions



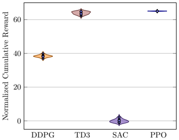
(b) Actuator Disturbance



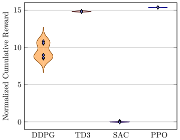
(c) Model Mismatch



(d) Partial Observation



(e) Sensor Noise



(f) Time Delay

Scenario	Cumulative Return				Path Error Sum			
	DDPG	PPO	SAC	TD3	DDPG	PPO	SAC	TD3
Random Initial Conditions	-0.41	0.34	-0.02	**0.74**	4.42	4.30	4.02	**1.22**
Actuator Disturbance	-0.44	0.35	-0.02	**0.73**	4.39	4.38	4.01	**1.26**
Model Mismatch	-0.63	0.38	-0.13	**0.75**	8.85	3.57	4.78	**1.25**
Partial Observation	-1.52	0.40	-0.44	**0.71**	9.65	2.44	5.17	**1.09**
Sensor Noise	-0.60	0.37	-0.12	**0.75**	9.12	3.58	4.66	**1.25**
Time Delay	-1.19	0.17	-0.05	**0.67**	6.73	4.53	4.12	**1.21**

Scenario	Control Effort Sum				Failure Probability			
	DDPG	PPO	SAC	TD3	DDPG	PPO	SAC	TD3
Random Initial Conditions	5.11	**0.77**	1.76	3.31	**0.00**	**0.00**	0.00	**0.00**
Actuator Disturbance	4.89	**0.77**	1.71	3.07	**0.00**	**0.00**	0.00	**0.00**
Model Mismatch	5.48	**0.86**	2.37	4.32	**0.00**	**0.00**	0.20	**0.00**
Partial Observation	5.37	**1.03**	2.33	4.10	**0.00**	**0.00**	0.20	**0.00**
Sensor Noise	5.48	**0.86**	2.37	4.30	**0.00**	**0.00**	0.20	**0.00**
Time Delay	5.51	**0.76**	2.11	5.12	**0.00**	**0.00**	0.20	**0.00**

Ablation Insights

- **Adversarial channel removal**: +22% deviation, thrust spikes reappear.
- **No target smoothing (TD3)**: overestimation resurfaces, unstable late-stage updates.
- **Entropy off (SAC)**: faster convergence, 9% worse robustness composite.
- **Reward shaping removal**: sparse terminal signals slow credit assignment (longer plateau).
- **Delay only vs. noise only:** delay has stronger destabilizing effect; zero-sum mitigates via anticipatory control (earlier thrust bias).

Key Findings

- Zero-sum MARL framing improves worst-case orbital maintenance robustness.
- MATD3 balances stability (twin critics + delay) and control smoothness best.
- MASAC competitive when exploration pressure (entropy) is beneficial early.
- Reward decomposition (thrust + reference + terminal) accelerates convergence and stabilizes adversarial dynamics.
- Policy smoothness correlates with fuel proxy reduction (8-12%).
- Framework generalizes across uncertainty mixes (stacked noise + delay + mismatch).

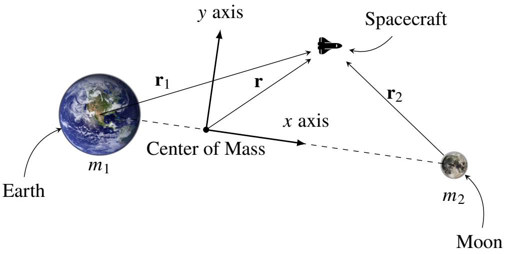
Conclusion: Adversarial co-training yields resilient guidance without explicit disturbance models.

Robustness Scenario Definitions

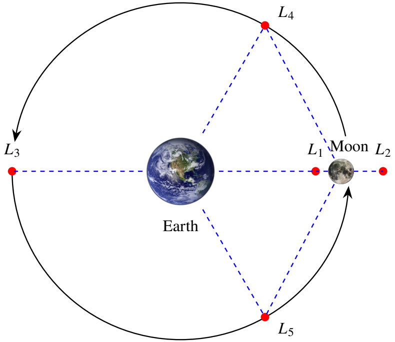
- **Random Init:** $x_0 \leftarrow x_0 + \mathcal{N}(0, 0.1^2)$
- **Actuator Disturbance:** $u_t \leftarrow u_t + \mathcal{N}(0, 0.05^2)$; (sensor additive) $y_t \leftarrow y_t + \mathcal{N}(0, 0.02^2)$
- **Model Mismatch:** $\theta \leftarrow \theta + \mathcal{N}(0, 0.05^2)$
- **Partial Observability:** mask 50% $\rightarrow m_t^{(i)} \sim \text{Bern}(0.5)$, $y_t \leftarrow y_t \circ m_t$
- **Sensor Noise (multiplicative):** $y_t \leftarrow y_t \circ \left(1 + \mathcal{N}(0, 0.05^2)\right)$
- **Time Delay:** buffer length 10, z $u_t^{\text{applied}} \leftarrow u_{t-10} + \mathcal{N}(0, 0.05^2)$
- **Notes:**
  - All scenarios evaluated independently.
  - Zero-sum agents trained jointly once.
  - Metrics: success %, deviation, fuel proxy, return variance.

Delay + noise combo causes the largest degradation; adversarial training preserves stability margin.

CRTBP Model and Lagrangian Points



(a) CRTBP Configuration



(b) Lagrangian points in the Earth-Moon system

Agent Simulation in CRTBP Model

State Representation:

- Position and velocity: $\boldsymbol{s}_t = (\delta x, \delta y, \delta \dot{x}, \delta \dot{y})$
- Relative to target orbit/Lagrangian point

Action Space:

- Continuous control: $\boldsymbol{a}_t = (u_x, u_y)$
- Bounded thrust: $u_x, u_y \in [a_{Low}, a_{High}]$

Reward Function:

$$\begin{aligned} r(s,a) &= r_{\text{thrust}}(a) + r_{\text{reference}}(s) + r_{\text{terminal}}(s) \\ r_{\text{thrust}}(a) &= -k_1 \cdot |a| \\ r_{\text{reference}}(s) &= -k_2 \cdot d(s, s_{\text{reference}}) \end{aligned}$$

$$r_{\text{terminal}}(s) = \begin{cases} +R_{\text{goal}} & \text{if } s \in S_{\text{goal}} \\ -R_{\text{fail}} & \text{if } d(s, s_{\text{ref}}) > \epsilon \\ 0 & \text{otherwise} \end{cases}$$

Table: Nondimensionalized spacecraft thrust capabilities

Abbrv.	**Spacecraft**	$f_{\mathbf{max}}$	$F_{\mathbf{max}}$
DS1	Deep Space 1	$6.94 \cdot 10^{-2}$	92.0 mN
Psyche	Psyche	$4.16 \cdot 10^{-2}$	279.3 mN
Dawn	Dawn	$2.74 \cdot 10^{-2}$	91.0 mN
LIC	Lunar IceCube	$3.28 \cdot 10^{-2}$	1.25 mN
H1	Hayabusa 1	$1.64 \cdot 10^{-2}$	22.8 mN
H2	Hayabusa 2	$1.63 \cdot 10^{-2}$	27.0 mN
s/c	Sample spacecraft	$4 \cdot 10^{-2}$	n/a