Better Reflex-Based Agent for LunarLander

# 1. Mục tiêu

Bài toán yêu cầu **xây dựng một Better Reflex-Based Agent cho môi trường LunarLander-v3** trong thư viện gymnasium. Agent phải dựa trên quan sát (observation) của môi trường để chọn hành động thích hợp, nhằm hạ cánh tàu an toàn nhất có thể.

**Viết được một reflex-based agent hoạt động hợp lý hơn random**, có thể điều khiển thrusters để hạ cánh an toàn hơn. Chạy test 100 lần + in kết quả + 1 lần render để quan sát trực quan.

# 2. Môi trường và quan sát

* **Môi trường**: LunarLander-v3 (Gymnasium)
* **Observation vector (8 chiều):**

**1. x\_pos**: vị trí ngang (so với tâm).

**2. y\_pos**: độ cao.

**3.** **x\_vel**: vận tốc ngang.

**4. y\_vel**: vận tốc dọc.

**5.** **angle**: góc nghiêng tàu (0 = thẳng đứng).

**6.** **angular\_vel**: vận tốc quay.

**7. left\_leg:** 1 nếu chân trái chạm đất.

**8.** **right\_leg**: 1 nếu chân phải chạm đất.

* **Action space (4 hành động):**

**0:** do nothing (không làm gì)

**1:** fire left engine (đẩy sang phải)

**2:** fire main engine (đẩy lên, giảm tốc rơi)

**3:** fire right engine (đẩy sang trái)

# 3. Chiến lược agent

Agent được thiết kế dựa trên **các luật phản xạ đơn giản**, không dùng học tăng cường mà chỉ dựa vào quan sát trực tiếp từ môi trường. Các luật này mô phỏng cách một phi công thủ công điều khiển thruster để hạ cánh an toàn. Có 5 trường hợp chính cần thiết ở đây:

**3.1** **Điều khiển góc (angle)**

**A computer screen shot of white text

AI-generated content may be incorrect.**

* **Hệ số:** Kp\_pos = 0.5, Kd\_vel = 0.3, angle\_threshold = 0.05
* **Mục đích:** giữ tàu thẳng, giảm rung lắc, hướng về trung tâm.

**3.2** **Giảm tốc độ ngang gần mặt đất**

A screenshot of a computer code

AI-generated content may be incorrect.

Khi gần đất (y < 0.5), giảm tốc vx để **tàu đứng yên trên pad**.

**3.3** **Kiểm soát tốc độ rơi (vy)**

A black background with white text

AI-generated content may be incorrect.

Nếu rơi nhanh (vy < -0.3), bật **main engine** để hạ cánh nhẹ nhàng.

**3.4** **Hành động mặc định**

A black background with white text

AI-generated content may be incorrect.

Khi không có điều kiện nào thỏa, agent giữ nguyên trạng thái.

# 4. Luồng hoạt động agent

A screenshot of a computer program

AI-generated content may be incorrect.

**Luồng**:

1. Nhận observation từ môi trường
2. Tính angle\_target dựa trên x và vx
3. So sánh angle với angle\_target → bật thruster trái/phải
4. Nếu y < 0.5 → điều chỉnh vx
5. Nếu vy < -0.3 → bật main engine
6. Nếu không → action 0
7. Lặp lại đến khi tàu hạ cánh hoặc crash

# 5. Thử nghiệm và đánh giá

**5.1. Demo 3 episode**

**A screen shot of a computer program

AI-generated content may be incorrect.**

Quan sát trực tiếp tàu hạ cánh, xem mức ổn định.

**5.2. Đánh giá 100 episode**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**5.3. Kết quả mẫu**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Phân tích:**

* Reward trung bình **cải thiện đáng kể** so với các phiên bản trước (~ -69 → -54).
* Một số episode đã **hạ cánh thành công** (reward ≥100 = 3/100), chứng tỏ agent **giữ thăng bằng tốt**.
* Observation cuối cùng cho thấy **gần mặt đất** với cả hai chân tiếp xúc (left\_leg=1, right\_leg=1).

# 6. Kết luận

Phiên bản **Better Reflex-Based Agent** – ổn định này đã cải thiện đáng kể so với các agent trước đó: reward trung bình từ **~-69 → ~-54.**

Một số **episode** đã hạ cánh thành công **(3/100),** cho thấy agent **đã giữ thăng bằng tốt**, giảm rung lắc, và điều chỉnh vị trí x về gần pad.

Agent **reflex-based, rule đơn giản**, không dùng học máy, nhưng đủ để mô phỏng hạ cánh ổn định.

**Hạn chế**: phần lớn episode vẫn chưa hạ cánh hoàn hảo, reward vẫn âm, chứng tỏ **khó đạt 100% success chỉ với rule-based.**

Phiên bản này phù hợp để trình diễn, làm cơ sở cho các cải tiến tiếp theo, ví dụ: thêm PID tinh chỉnh hoặc **RL-based agent** để tăng success rate.