Reinforcement Learning for Orbital Transfers

James Verbus
linkedin.com/in/jamesverbus

AI WINTER SCHOOL

JAN. 6 - 9, 2026

BROWN
Department of Physics

Center for the Fundamental Physics
of the Universe

Thank you

- **Ariel Green** for coordinating the workshop
- **Chongwen Lu** for helpful feedback
- **Richard Gaitskell** and **Ian Dell'antonio** for inviting me to contribute

2009-2016

2017-now

What will you learn in this workshop?

Build intuition: Hohmann transfer as the analytic baseline (when it works, when it breaks)

Formulate RL: choose **state / action / reward / termination** for an orbital-transfer environment

Train + debug: train RL with discrete & continuous control, then diagnose failures (chatter, micro-thrusting, crashes) and iterate

Figure from 2504.04160v1

Orbital Geometry

Two-body model (baseline):

- Central body is a point mass with gravitational param μ = *GM*
- Spacecraft is a test mass (primary fixed; $M \gg m$)
- No perturbations (e.g., no J2, drag, 3rd body)

Geometric description

- **Size/shape:** *a* (semi-major), *e* (eccentricity)
- **Orientation:** (*i*, *Ω*, *ω*)

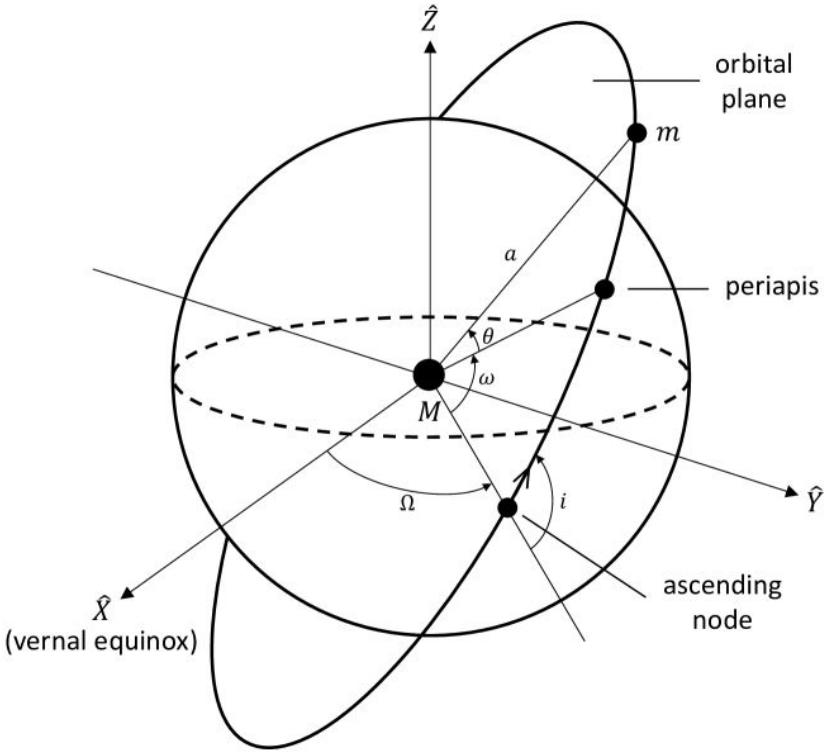


Figure from 2504.04160v1

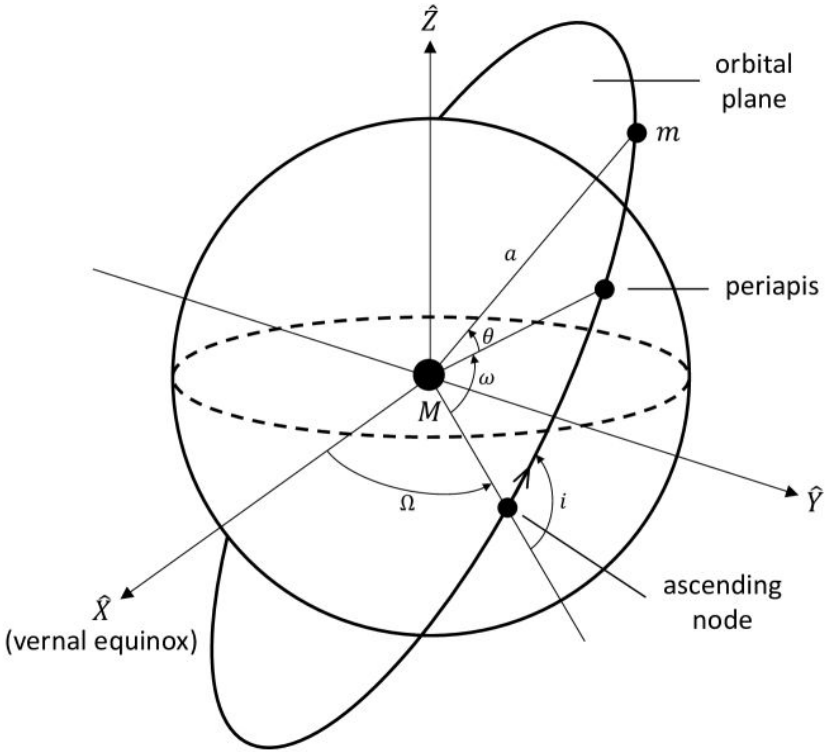
Orbital Geometry

Two-body model (baseline):

- Central body is a point mass with gravitational param $\mu = GM$
- Spacecraft is a test mass (primary fixed; $M \gg m$)
- No perturbations (e.g., no J2, drag, 3rd body)

Geometric description

- **Size/shape:** *a* (semi-major), *e* (eccentricity)
- **Orientation:** (*i*, *Ω*, *ω*)



Bridge: In two-body motion, **energy sets** a and **(energy, angular momentum) set** e

$$\varepsilon = -\frac{\mu}{2a}, \qquad e = \sqrt{1 + \frac{2\varepsilon L^2}{\mu^2}}$$

$$E = m\,\varepsilon = \frac{1}{2}mv^2 - \frac{GMm}{r} = -\frac{GMm}{2a}$$

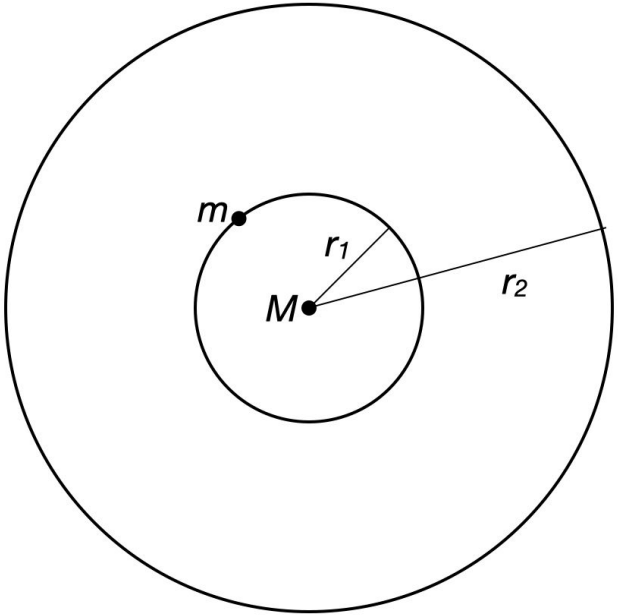
Dynamic invariants (simple 2D; circular)

- **During coast (no thrust):** ε and L are conserved (ideal two-body, 2D)

$$\varepsilon = \frac{\|v\|^2}{2} - \frac{\mu}{r}, \qquad L = r v_t$$

- **These determine shape/size:** $a = -\mu/(2\varepsilon)$, $e = \sqrt{1 + 2\varepsilon L^2/\mu^2}$
- **Modeling choice:** we ignore absolute ω, ν and use rotation-invariant features $(r, v_r, v_t, \varepsilon, L)$

Simple circular orbit transfer

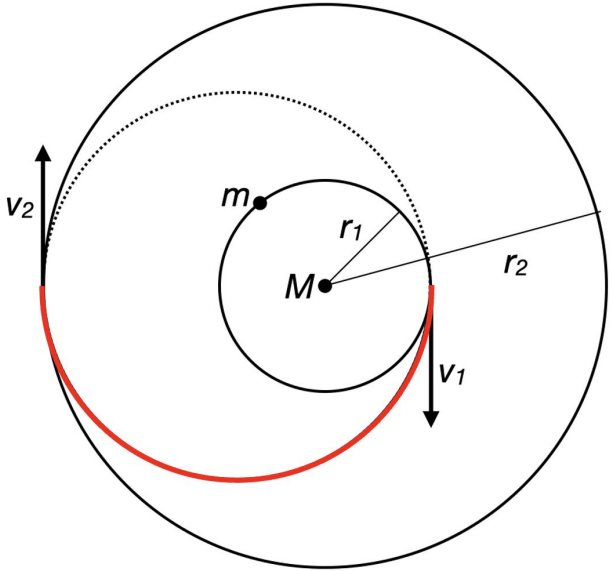


- **Goal:** Get from r_1 to r_2
- **Assumptions:** M >> m
- **Target:** reach r_2 and be near-circular (small v_r)
- **Cost:** minimize total Δ*v (fuel), because:*

$$\Delta v = I_{sp} g_0 \ln\left(\frac{m_0}{m_f}\right)$$

For small propellant usage, $\Delta m/m \approx \Delta v/(I_{sp} g_0) \Rightarrow$ **fuel ∝ Δv**.

Hohmann transfer (analytic benchmark)



- **Setup:** transfer from circular orbit r_1 to r_2 (coplanar, two-body, impulse burns)
- **Concept: minimum-Δ*v*** transfer among all **two-burn** coplanar circular-to-circular maneuvers
- **Maneuver:** Burn at r_1 → coast half transfer ellipse → burn at r_2 to circularize
- **Why we use it:** analytic "ground truth" baseline; real missions (eccentric/3D/constraints/low-thrust) need more general methods (e.g., RL)

- **Closed form results:**
  - $a_t = \dfrac{r_1 + r_2}{2}$
  - $\Delta v_1 = \sqrt{\mu/r_1}\left(\sqrt{\dfrac{2r_2}{r_1 + r_2}} - 1\right)$
  - $\Delta v_2 = \sqrt{\mu/r_2}\left(1 - \sqrt{\dfrac{2r_1}{r_1 + r_2}}\right)$
  - $t_H = \pi\sqrt{\dfrac{a_t^3}{\mu}}$

- **Orbital energy (constant during each coast):**

$$\varepsilon \equiv \frac{v^2}{2} - \frac{\mu}{r} = -\frac{\mu}{2a}$$

so

$$\varepsilon_1 = -\frac{\mu}{2r_1}, \qquad \varepsilon_t = -\frac{\mu}{2a_t} = -\frac{\mu}{r_1 + r_2}, \qquad \varepsilon_2 = -\frac{\mu}{2r_2}.$$

Where:

- ε = **specific** orbital energy (per unit spacecraft mass),
- $\mu = GM$ (since $M \gg m$),
- a is the orbit semi-major axis, and $a_t = (r_1 + r_2)/2$ for the transfer ellipse.

Beyond Hohmann…

Hohmann is optimal when:

- coplanar **circular** → **circular**
- two-body gravity
- **impulsive** burns, no extra constraints

Often a useful first baseline when: “mostly circular, mostly coplanar” Earth-orbit transfers (e.g., LEO → higher circular)

You need more general methods:

- **3D/geometry:** plane changes
- **physics:** perturbations / multi-body
- **control/ops:** low-thrust + constraints + uncertainty
- **large orbit raises:** bi-elliptic (3 burns) can beat Hohmann in Δv when $r_2/r_1 \gtrsim 12$, but takes much longer

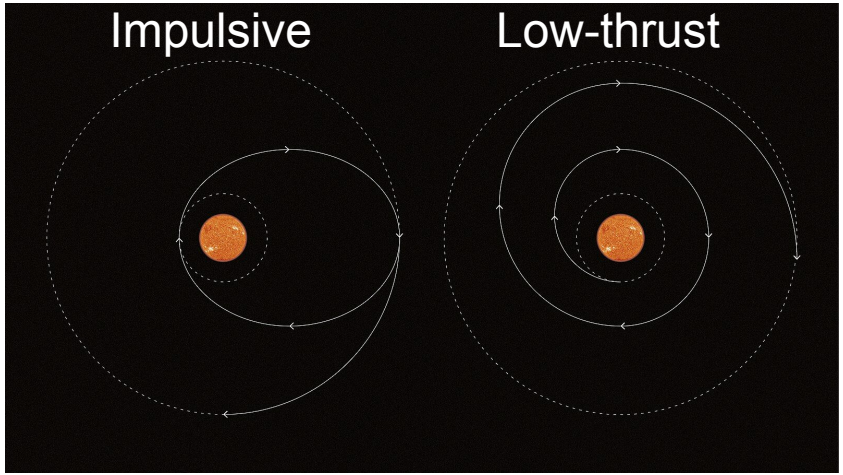


Image credit: Wikimedia CC-BY-4.0 (link)

Reinforcement Learning intuition

Goal: learn a **feedback controller** $\pi(a \mid s)$ from trial-and-error in simulation

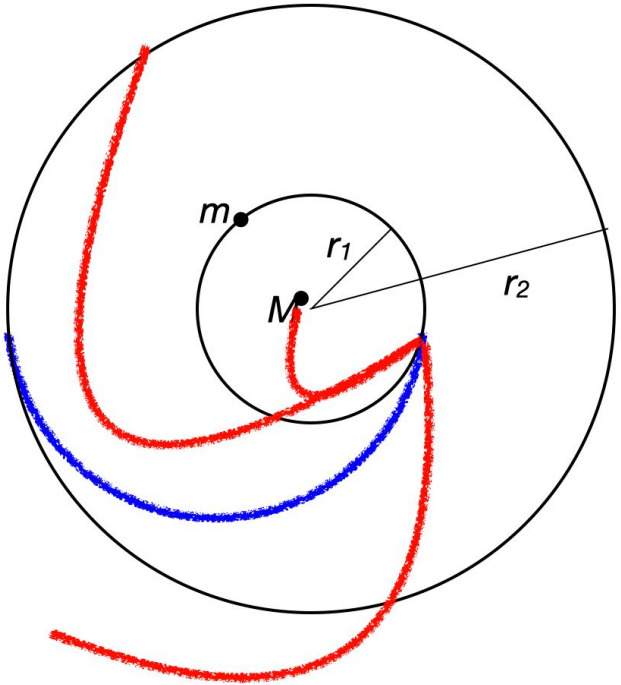
Roles:

- **Agent:** spacecraft controller (policy)
- **Environment:** orbital dynamics simulator

Loop (core idea): observe state s_t → choose action a_t → simulator returns (s_{t+1}, r_{t+1}) → update π to increase long-term reward

Why we care: handles nonlinear dynamics + constraints, and re-plans every step (robust vs one-shot trajectories)

How we use it: train offline with many rollouts → deploy $\pi(a \mid s)$ online for real-time guidance



Now let's kick of a training run…

RL in One Slide

At each step you have:

- a **state** s_t (what you know now; e.g., r, v_r, v_t, E, L, t)
- an **action** a_t (what you do; e.g., thrust / Δv)
- the simulator produces **next state + reward** (s_{t+1}, r_{t+1})

Goal: maximize **long-term** reward (γ~1 = long-term planning)

$$\max \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r_{t+1}\right]$$

What we learn: a policy $\pi(a \mid s)$ = "given the current state, what action should I take?"

Two common ways to learn it:

- **Value-based:** learn $Q(s, a)$ ("how good is action a in state s?") → choose a = argmax_a $Q(s, a)$
- **Actor–critic:** learn the **actor** $\pi(a \mid s)$ plus a **critic,** usually $V(s)$, that estimates "how good the situation is"

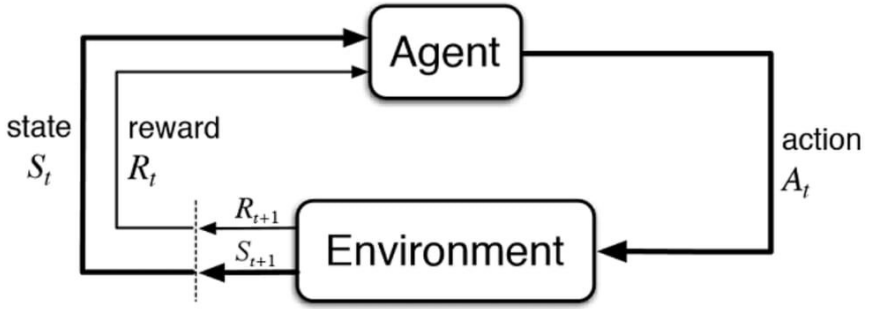


Image source: Sutton, R. S. and Barto, A. G. Introduction to Reinforcement Learning

Turning an Orbital Transfer into an RL Problem

Environment (physics + mission): 2D two-body gravity, μ=1, start r_1=1 → target r_2=1.6, timestep Δt=0.05

Observation (what the agent sees): 6 rotation-invariant features: $\left[\, \tilde{r},\ \tilde{v}_r,\ \tilde{v}_t,\ \widetilde{(L - L^*)},\ \widetilde{(E - E^*)},\ \text{last action} \,\right]$
(no orbital angle → same policy at any phase)

Action (what the agent controls):

- **Discrete:** {coast, +prograde, −retrograde} with per-step impulse $\Delta v = \pm dv\_mag$
- **Continuous:** throttle $u \in [-1,1]$, $\Delta v = u^* dv\_mag$ (applied tangentially each step)

Reward (dense shaping + costs):

- **Get close:** reduce $|E-E^*| + |L-L^*|$
- **Be efficient:** penalize $|\Delta v|$ + ignition
- **Don't die:** crash / timeout

Episode ends on: success (optional terminate), crash (outside radius bounds), or timeout (max steps / orbits)

What we tune in experiments: *dv_mag*, fuel cost, ignition penalty, shaping scale, training steps → trade off Δv efficiency vs robustness vs smoothness

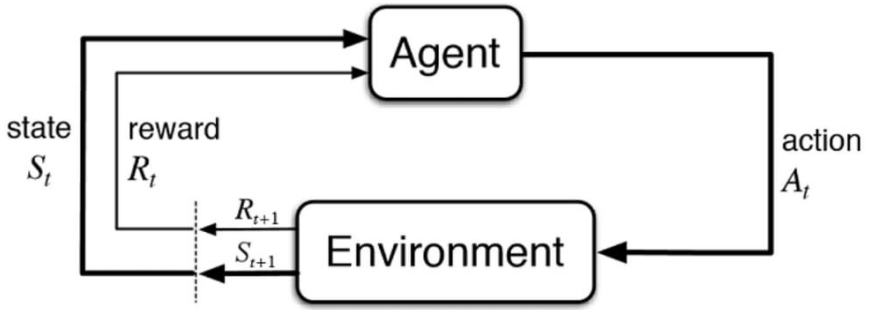


Image source: Sutton, R. S. and Barto, A. G. Introduction to Reinforcement Learning

Key Takeaways & What We'll Do Next

Hohmann is the benchmark: elegant, closed-form, and **Δ*v*-optimal** only for the ideal case (two-body, coplanar, circular, impulsive burns)

Reality breaks the assumptions: perturbations, 3D plane changes, low-thrust, timing/constraints → analytic solutions become incomplete or unavailable

RL viewpoint: learn a **feedback policy** $\pi(a \mid s)$ by trial-and-error in simulation

- good fit for nonlinear dynamics + messy objectives + uncertainty

Workshop plan (hands-on): build a simplified orbital transfer RL environment and iterate experimentally

- choose **state/action** encoding
- design **reward shaping + costs**
- train an agent and inspect trajectories / Δ*v* efficiency / failure modes

Submit your experiment results at the end of the workshop!
https://docs.google.com/forms/d/e/1FAIpQLSc0s4Cle8uoZgxJZadtvsIUW492GozCEghRjJFN3BZAnhXkdg/viewform?usp=header