Control and Synthetic Data Generation for the Inverted Pendulum

Reinforcement Learning with PPO and Generative Modeling

Himani Sharma

Prof. Harikumar Kandath

IIIT Hyderabad

September 17, 2025

Outline

Problem Statement

Project Problem

Determine the minimal proportion of real-world data that, when combined with high-fidelity synthetic data, is sufficient to train a reinforcement learning agent to achieve near-expert performance on the inverted pendulum control task.

Supporting Phases

1. **Phase 1 – Control Problem:**
   Develop and train a PPO controller that balances the inverted pendulum by applying horizontal forces to the cart.
2. **Phase 2 – Synthetic Data Generation:**
   Collect expert PPO trajectories and train a GAN or Diffusion Model to generate realistic state–action–reward sequences.

System Description and Requirements

System Overview

The physical platform is an **inverted pendulum mounted on a motorized cart**.

Features

1. Intrinsically unstable without active control.
2. Nonlinear dynamics.

Control Objective

Balance the pendulum by applying a horizontal force F to the cart.

- Motion assumed to be constrained to the vertical plane.
- Control input: force F.
- Outputs: pendulum angle θ and cart position x.

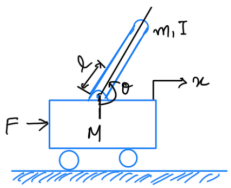


Figure: Inverted Pendulum

Design Requirements

Goal

Ensure the inverted pendulum remains stable and quickly returns to equilibrium after disturbances.

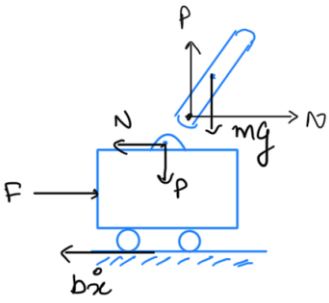
Control Objectives

- Rapid recovery from external impulses or initial offsets.
- Minimal overshoot and sustained deviation of the pole angle.
- Smooth cart motion for commanded position changes.

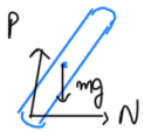
System Criteria

- Stable closed-loop behaviour under continuous control.
- Low steady-state error and robust performance to noise.
- Requirements hold for both classical and RL-based controllers.

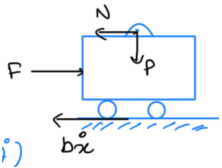
Free-Body and Rotation Diagrams



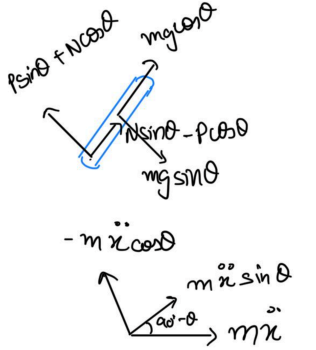
Forces on the System



Pendulum



Cart



Rotation / Torque Diagram

System Dynamics and Rotation

Translational Dynamics

$$(x_c, y_c) = \big(x + l\sin\theta,\ -l\cos\theta\big)$$

Pendulum Force Balance

$$N = m\Big(\ddot{x} - l\sin\theta\dot{\theta}^2 + l\cos\theta\ddot{\theta}\Big)$$

$$P - mg = m\Big(l\cos\theta\dot{\theta}^2 + l\sin\theta\ddot{\theta}\Big)$$

Cart Force Balance

$$F - N - b\dot{x} = M\ddot{x}$$

$$F = (M+m)\ddot{x} + b\dot{x} + m\Big(l\cos\theta\ddot{\theta} - l\sin\theta\dot{\theta}^2\Big)$$

Rotational Dynamics

$$\sum \tau_{\text{pivot}} = I_p\ddot{\theta}, \qquad I_p = I + ml^2$$

Torque balance:

$$P\sin\theta + N\cos\theta - mg\sin\theta = ml\ddot{\theta} + m\ddot{x}\cos\theta$$

Final compact form:

$$\boxed{(I + ml^2)\ddot{\theta} + mgl\sin\theta = -ml\cos\theta\,\ddot{x}}$$

State-Space Model

$$\underbrace{\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-(I+ml^2)b}{I(M+m)+Mml^2} & \frac{m^2gl^2}{I(M+m)+Mml^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-mlb}{I(M+m)+Mml^2} & \frac{mgl(M+m)}{I(M+m)+Mml^2} & 0 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} 0 \\ \frac{I+ml^2}{I(M+m)+Mml^2} \\ 0 \\ \frac{ml}{I(M+m)+Mml^2} \end{bmatrix}}_{B} u$$

Output equation:

$$\mathbf{y} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{C} \mathbf{x} + \underbrace{\begin{bmatrix} 0 \\ 0 \end{bmatrix}}_{D} u.$$

Gymnasium Environment Overview

Environment

- `InvertedPendulum-v5` from `Gymnasium` (MuJoCo backend).
- Task: balance an inverted pendulum by applying horizontal force.

Action Space

- Continuous, one–dimensional.
- Represents horizontal force on the cart.

Observation Space

- 4-dimensional state vector: cart position, pole angle, cart velocity, pole angular velocity.

Rewards

- $+1$ reward for each timestep the pole remains within angle limits.

Starting State

- Sampled from a multivariate uniform distribution with random noise.

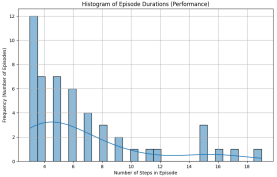
Episode Termination

- Ends if any state value is non-finite.
- Ends if $|\theta| > 0.2$ radians.

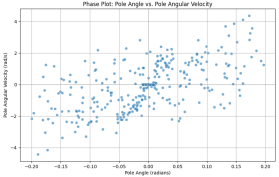
Truncation

- Maximum duration: 1000 timesteps.

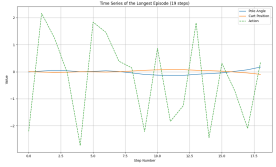
Random Policy Diagnostics ($N = 50$)



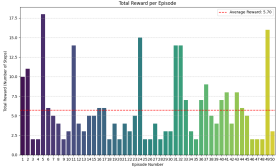
Histogram of Episode Durations



Phase Plot: Angle vs. Angular Velocity



Time Series of Longest Episode



Reward per Episode

Random actions lead to short episodes, large angle deviations, and low rewards—confirming the baseline agent shows no learning.

Conclusion

Random Agent Performance

The four diagnostic plots show that a random action strategy is **completely inadequate** for solving the Inverted Pendulum problem. The agent exhibits no learning or strategy—performance relies purely on chance. These results form a **quantitative baseline for failure** against which more advanced control or reinforcement learning (RL) methods can be compared.

Upgrade from Random to Learning

- Replace random actions with a **Reinforcement Learning policy**.
- Use **Proximal Policy Optimization (PPO)**—stable and well suited for continuous-control tasks.

PPO for Inverted Pendulum: Overview

- **Goal:** Learn a policy $\pi_\theta(a|s)$ that outputs the horizontal force a_t to keep the pole upright.
- **Actor–Critic:**
  - **Actor:** Policy network maps state $s_t = [x, \dot{x}, \theta, \dot{\theta}]$ to action a_t.
  - **Critic:** Value network estimates $V(s_t)$, the expected future reward (time balanced).
- **Trajectory Collection:** Run episodes, record (s_t, a_t, r_t) with reward $r_t = +1$ per time-step the pole stays up.
- **Advantage Estimate:**

$$\hat{A}_t = R_t - V(s_t), \qquad R_t = \sum_{k=0}^{T-t} \gamma^k r_{t+k}.$$

Positive $\hat{A}_t$ means the action balanced the pole better than predicted.

PPO Update Mechanism

- **Policy Ratio:**

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}.$$

- **Clipped Objective:**

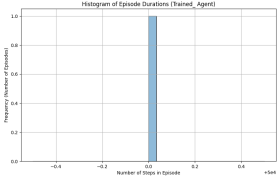
$$L^{\text{CLIP}}(\theta) = \mathbb{E}_t\Big[\min\big(r_t(\theta)\hat{A}_t,\ \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\big)\Big].$$

  Keeps each policy update *proximal*, avoiding large, unstable changes.

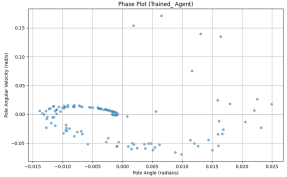
- **Training Loop:**
  1. Collect trajectories with current policy.
  2. Update Critic to minimize $\big(V(s_t) - R_t\big)^2$.
  3. Update Actor by maximizing L^{CLIP}.
  4. Repeat until the policy balances the pole for long horizons.

- **Outcome:** A robust closed-loop controller that makes small, continuous corrections to keep the pendulum upright.

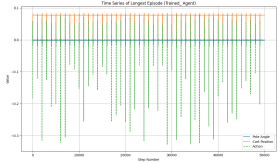
Trained PPO Agent Diagnostics (50,000 Timesteps)



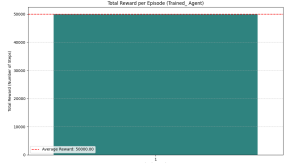
Histogram of Episode Durations



Phase Plot: Angle vs. Angular Velocity



Time Series of Longest Episode



Reward per Episode

The PPO agent trained for 50 000 timesteps achieves **near-perfect control**: one uninterrupted 50k-step episode, pole angle and cart position held at zero, and high-frequency micro-actions that maintain pinpoint equilibrium.

Improvement with PPO Training (50 000 Timesteps)

- After **50,000** training timesteps, the agent exhibits a fundamentally superior control strategy compared with the baseline **random policy**.
- **Random Policy — Open Loop**
  - Actions independent of state lead to rapid divergence.
  - Uniformly poor performance and short episodes.
  - No capability to stabilize the pendulum.
- **Trained PPO Policy — Closed Loop**
  - **Achieves Stability:** actively drives the system to equilibrium.
  - **Sustains Performance:** balances the pole for the full episode limit and beyond (demonstrated over 50 000 steps).
  - **Skillful Control:** applies rapid, precise micro–corrections to counteract instability.
- Extending training further continues to improve stability, robustness, and control accuracy.

Future Work and Expectation

Objective

Minimise the need for real data while retaining expert-level control performance.

Phase 2 – Synthetic Data

A neural generator such as **CTGAN** will be trained on expert PPO trajectories. Using tuples of $(s, a, r, s', \mathrm{done})$, the generator will learn to produce large volumes of high-fidelity synthetic episodes that accurately reflect expert behaviour.

Phase 3 – Hybrid RL

New PPO agents will then be trained on datasets that mix real and synthetic samples in varying proportions, ranging from entirely synthetic to entirely real. Each agent will be evaluated in the `InvertedPendulum-v5` environment to measure how well it learns under different ratios of real data.

Evaluation and Analysis

Phase 4 – Assessment

For every hybrid dataset, the mean episode reward of the trained agent will be measured and compared with the expert baseline. Results will be summarised in a plot of average reward versus percentage of real data, revealing the minimal real-data ratio required to achieve near-expert performance. This phase will demonstrate that predominantly synthetic data can support high-performing controllers while greatly reducing real-world data requirements.

Thank You