HomeostaticEnv: A Multi-Resource Environment for Homeostatic Reinforcement Learning Experiments

Gilzamir Gomes
gilzamir_gomes@uvanet.br
Department of Computer Science
State University of the Acaraú Valey

January 26, 2026

Abstract

Homeostatic regulation—the ability to maintain internal physiological variables within a viable range—is a fundamental property of living organisms. In the context of Artificial Intelligence, Homeostatic Reinforcement Learning (HRL) applies these biological principles to create agents capable of autonomous self-regulation. However, a significant challenge in HRL research is the lack of standardized environments that specifically test an agent's ability to manage high-dimensional physiological state spaces under constant decay. Critically, high-dimensional resource management implies an equally **high-dimensional continuous action space**, creating a complex control problem. This paper introduces `HomeostaticEnv`, a continuous control environment designed to simulate multi-resource homeostatic regulation. We detail the environment's dynamics and formalize three distinct reward mechanisms: Default (quadratic penalty), Euclidean (potential-based), and Operational Regimes (discrete-logic based). We demonstrate the environment's utility through Proximal Policy Optimization (PPO) experiments on high-dimensional tasks (up to 100 simultaneous resources), highlighting the impact of reward signal design on survival and stability in large action spaces.

1 Introduction

Reinforcement Learning (RL) has achieved remarkable success in domains ranging from game playing to robotics. Traditionally, RL agents maximize a cumulative extrinsic reward signal. However, in biological systems, behavior is often driven by *homeostasis*: the imperative to maintain critical physiological variables (such as temperature, glucose, or hydration) within a tight survival range against natural entropy [1].

Homeostatic Reinforcement Learning (HRL) integrates this concept, formulating the objective not as infinite maximization, but as the minimization of deviation from a setpoint. A core difficulty in HRL is the "curse of dimensionality." This challenge is twofold:

1. **State Space**: The agent must monitor N distinct physiological variables simultaneously.
2. **Action Space**: Regulation requires a corresponding high-dimensional continuous action space ($\mathcal{A} \in \mathbb{R}^N$).

An agent must often balance multiple conflicting resources simultaneously. For example, an action that restores one resource might deplete another, or simply time spent foraging for one resource allows others to decay. To facilitate research in this domain, we present `HomeostaticEnv`, a gymnasium-compatible [2] environment. Unlike standard grid-worlds or MuJoCo tasks, `HomeostaticEnv` focuses specifically on the dynamics of resource management, scaling to massive action spaces ($N = 100$) where standard exploration strategies often fail.

2 Foundations

The theoretical underpinning of `HomeostaticEnv` relies on Drive-Reduction Theory. We define an agent's internal state as a vector $S_t = [s_1, s_2, ..., s_N]$, representing N distinct physiological variables.

2.1 Zones of Stability

For each variable s_i, we define three critical regions relative to a target setpoint (typically 0.0):

1. **Target (T_i)**: The ideal homeostatic equilibrium (set to 0.0).
2. **Safety Zone (Z_{safe})**: A range $[-z_{safe}, z_{safe}]$ where the organism operates optimally.
3. **Survival Zone (Z_{surv})**: A critical limit $[-z_{surv}, z_{surv}]$. If any $|s_i| > z_{surv}$, the agent "dies," and the episode terminates immediately.

3 The Environment Description

`HomeostaticEnv` models a continuous control problem where the agent must continuously counteract the natural decay of resources.

3.1 State and Action Space

The observation space is a continuous vector $\mathcal{O} \in \mathbb{R}^{3N}$, consisting of:

- Current physiological state values (s_t).
- Survival zone thresholds.
- Safety zone thresholds.

The action space is a continuous vector $\mathcal{A} \in [-1, 1]^N$. **High Dimensionality Note:** It is important to emphasize that the action space scales linearly with the number of resources. For a simulation with $N = 100$ variables, the agent must output a 100-dimensional vector at every step, deciding the exact intensity of correction for every single resource simultaneously.

3.2 Dynamics

The environment follows a linear difference equation. At each time step t, the state evolves according to:

$$s_{t+1,i} = s_{t,i} - \delta_i + \alpha \cdot a_{t,i} \tag{1}$$

Where:

- δ_i is the decay ratio (fixed at 0.01).
- α is the action scaler (fixed at 0.2).
- $a_{t,i}$ is the clipped action $[-1, 1]$.

3.3 Reward Functions

A critical contribution of this environment is the implementation of three distinct homeostatic reward functions to study multi-objective regulation.

3.3.1 Default (Quadratic Penalty)

This function penalizes the agent based on the sum of squared distances from the target. It mimics a pure minimization of error.

$$R_{default} = -\sum_{i=1}^{N}(s_{t,i} - T_i)^2 \cdot k \tag{2}$$

where k is the reward scale.

3.3.2 Euclidean (Differential)

This is a potential-based reward function. It rewards the *improvement* in the total system state rather than the absolute state.

$$R_{euclidian} = (D_{t-1} - D_t) \cdot k \tag{3}$$

where $D_t = \sum(s_{t,i} - T_i)^2$. This formulation provides dense gradients for moving towards the center, regardless of the current distance.

3.3.3 Operational Regimes

This function implements a logic-based reward that prioritizes safety and penalizes worsening conditions when in danger. For each variable i, let $d_t = |s_{t,i} - T_i|$. We define binary indicators:

- $u_i = 1$ if $d_t \leq Z_{safe}$, else 0 (Inside Safety Zone).
- $v_i = 1$ if $d_t \geq d_{t-1}$, else 0 (Condition Worsening).

The reward is calculated as:

$$R_{regimes} = \sum_{i=1}^{N}[u_i + (1 - u_i)(1 - 3v_i)] \cdot k \tag{4}$$

This logic dictates:

- **Inside Safety:** +1 reward.
- **Outside Safety but Improving:** +1 reward.
- **Outside Safety and Worsening:** -2 penalty $(1 - 3(1))$.

4 Experiments

To validate the environment and compare the efficacy of the reward functions, we utilized the Proximal Policy Optimization (PPO) algorithm from the Stable-Baselines3 library [3].

4.1 Experimental Setup

We tested the environment under two distinct complexity levels to evaluate scalability:

1. **Standard High-Dim:** $N = 20$ variables.
2. **Massive High-Dim:** $N = 100$ variables.

The $N = 100$ case represents a significant challenge, requiring the policy network to map observations to a 100-dimensional continuous action vector, effectively managing 100 concurrent homeostatic loops.

A função de recompensa baseada em regimes operacionais foi testada em duas versões. Na versão A, foi usado uma safety zone fixa em 0.5. Na versão B, foi usada uma safety zone amostrada uniformemente do intervalo real $[0.5,\ 0.9]$.

- **Algorithm:** PPO (MlpPolicy)
- **Total Timesteps:** 250,000
- **Environments:** 8 parallel environments (SubprocVecEnv)
- **Max Steps per Episode:** 1000

Three separate training regimes were executed corresponding to the three reward functions: `default`, `euclidian`, and `operational_regimes`.

5 Results

The main metric in these experiments is the time the episode can be kept active, known as episode length. We also show that the best-performing methods are those with the lowest standard deviation.

In the simulation with 20 variables, the result in terms of episode length is shown in Figure 1, while the standard deviation is shown in Figure 2.

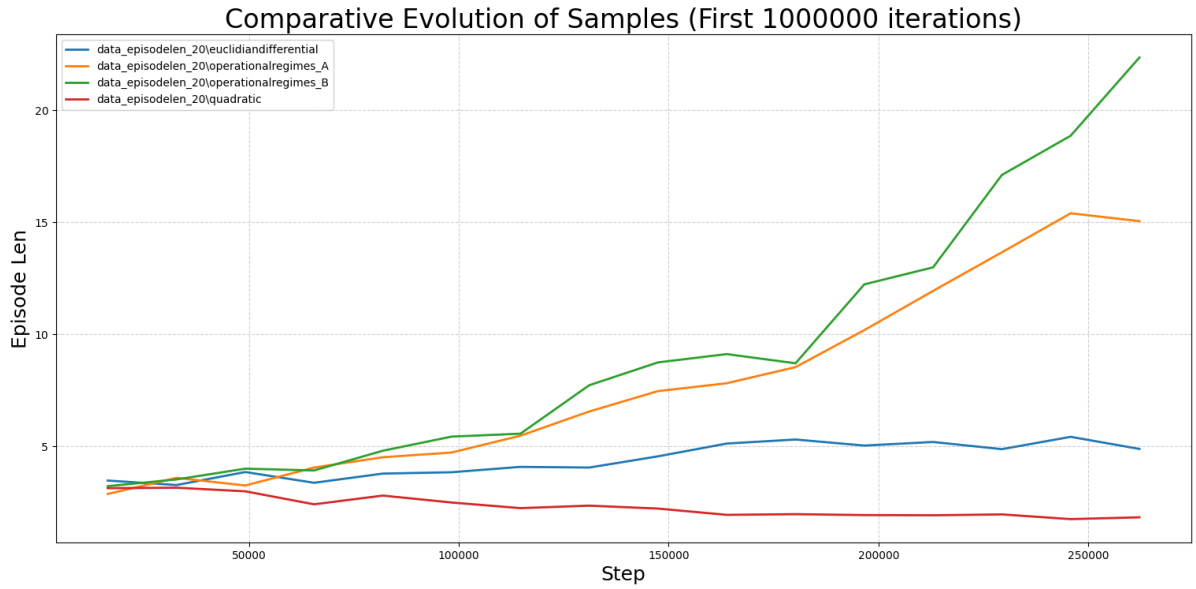

Figure 1: Comparison of Episode Length over 250k timesteps. Episode length of the different reward functions used in the experiment with 20 variables.

In the simulation with 100 variables, the result in terms of episode length is shown in Figure 3, while the standard deviation is shown in Figure 4.

5.1 Survival Analysis in High Dimensionality

In the $N = 20$ settings, differences between reward functions are noticeable but manageable. However, in the $N = 100$ experiments, the dimensionality of the action space exposes critical weaknesses in standard reward formulations.

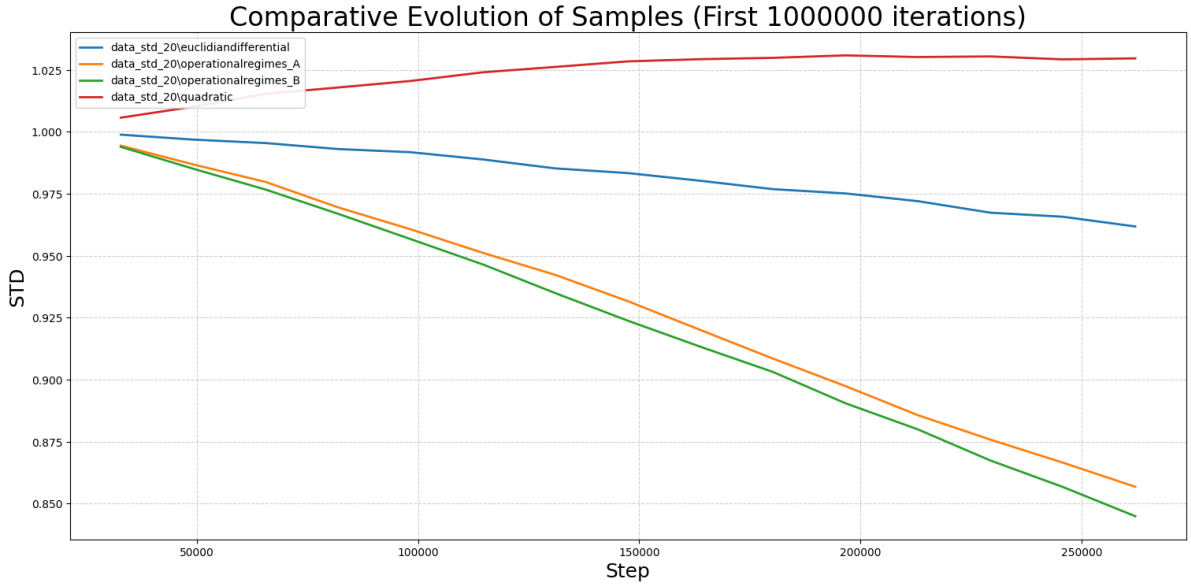

Figure 2: Comparison of Reward Stadard Deviation over 250k timesteps. Standard deviation of the different reward functions used in the experiment with 20 variables.

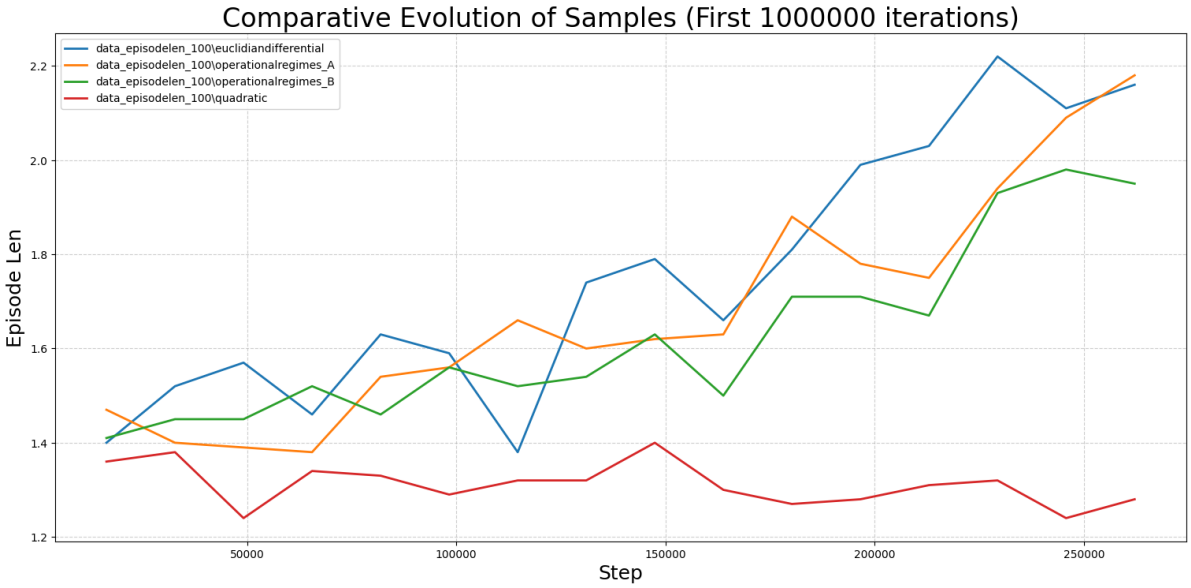

Figure 3: Comparison of Episode Length evolution over 250k timesteps. Episode length of the different reward functions used in the experiment with 100 variables.

The `default` reward function (quadratic penalty) struggled significantly in the 100-variable case. The aggregate penalty from 100 drifting variables creates a noisy signal that often overwhelms the gradient, leading to early convergence to sub-optimal policies or "learned helplessness," where the agent fails to effectively counter decay.

Conversely, the `operational_regimes` reward proved robust even with $N = 100$. By decomposing the complex global task into local binary objectives (safe/unsafe, improving/worsening) for each of the 100 variables, it effectively creates a gradient that prioritizes urgent recovery for critical variables while maintaining stable ones.

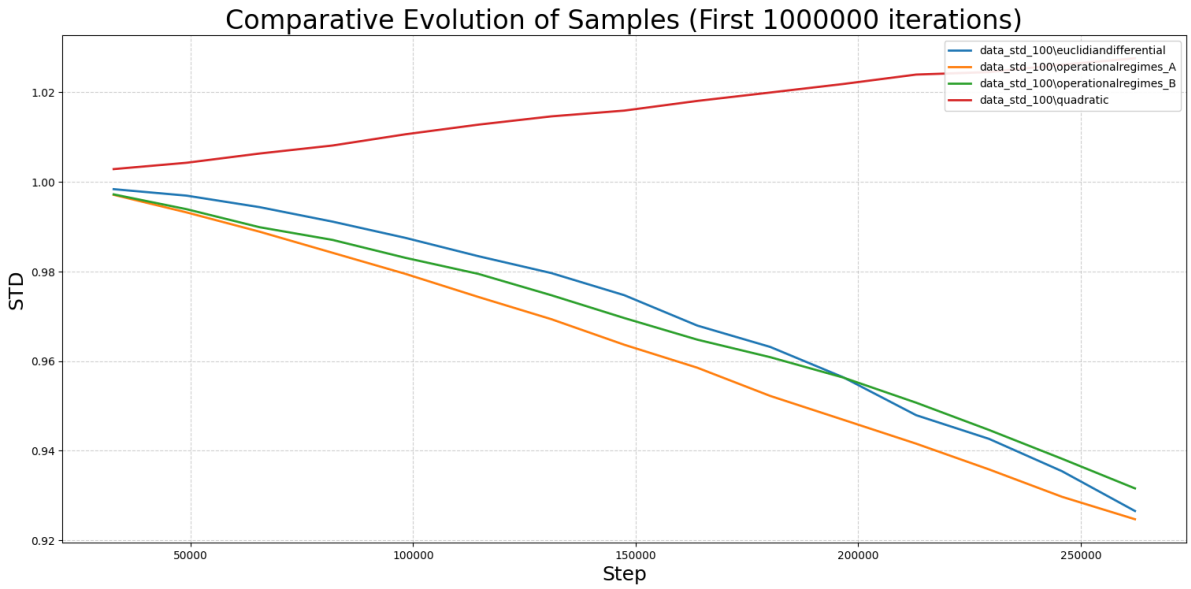

Figure 4: Comparison of Reward Stadard Deviation over 250k timesteps. Standard Deviation of the different reward functions used in the experiment with 100 variables.

5.2 Multi-Resource Management

The `euclidian` reward demonstrated a strong ability to guide the agent back to equilibrium. However, because it relies on the *difference* between steps ($D_{t-1} - D_t$), the agent occasionally learns oscillating behaviors around the safety boundaries to maximize the differential gain. This effect was amplified in the 100-variable case, leading to higher overall system variance compared to the regime-based approach.

6 Conclusions

We introduced `HomeostaticEnv`, a specialized environment for Homeostatic Reinforcement Learning. By allowing configurable dimensionality and distinct reward architectures, it serves as a robust testbed for algorithms dealing with multi-resource management.

Our experiments with PPO on 20 and 100 simultaneous variables indicate that simple quadratic penalties are insufficient for massive high-dimensional survival tasks. The high dimensionality of the continuous action space requires reward signals that are dense and informative. Logic-based "regime" rewards, which distinguish between safe states and critical recovery states, provided the most stable homeostatic policies in these extreme conditions.

References

[1] Cannon, W. B. (1932). *The Wisdom of the Body*. W. W. Norton & Company.

[2] Towers, M., et al. (2023). *Gymnasium: A Standard Interface for Reinforcement Learning Environments*. arXiv preprint arXiv:2106.00000.

[3] Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., & Dormann, N. (2021). *Stable-Baselines3: Reliable Reinforcement Learning Implementations*. Journal of Machine Learning Research, 22(268), 1-8.

[4] Gomes, G. (2025). *A Simple Multi-Resources Environment*. [Software]. Available online.