Cybernetic Reinforcement Learning Agent for the Wumpus World Environment

1st Cristian Romero
Ingeniería de Sistemas
Universidad Distrital Francisco José de Caldas
Bogotá, Colombia
cdromerog@udistrital.edu.co

2nd Cesar Pulido
Ingeniería de Sistemas
Universidad Distrital Francisco José de Caldas
Bogotá, Colombia
capulidoc@udistrital.edu.co

***Abstract*—The Wumpus World is a classic testbed in artificial intelligence, used to explore decision-making under uncertainty and partial observability. In this work, we design and analyze a reinforcement learning agent capable of operating within a Wumpus World environment. Drawing from principles of cybernetics, system dynamics, and deep reinforcement learning, the agent learns to navigate hazardous conditions using feedback-driven adaptation. The system integrates environmental perception, structured reward mechanisms, and real-time visualization to support intelligent behavior. This approach demonstrates the potential of combining IA with cybernetic feedback loops to develop agents that learn and act autonomously in uncertain environments.**

***Index Terms*—Reinforcement learning, cybernetics, Wumpus World, intelligent agents, DQN, system modeling.**

I. Introduction

The Wumpus World is a classic artificial intelligence reference environment for decision making under uncertainty in partially observable domains originally proposed by Genesereth and Nilsson [1]. It consists of a grid-based world populated with hazards such as pits and a creature called the Wumpus, along with a goal element — gold — that the agent must locate and retrieve. The agent receives limited, local perceptual information in the form of binary signals (e.g., breeze near pits, stench near the Wumpus), which it must interpret to survive and complete its objective.

This project focuses on designing and refining an adaptive agent capable of learning to navigate the Wumpus World environment effectively. Drawing from cybernetic principles and reinforcement learning techniques, the agent initially employs a basic Q-learning approach and evolves into a Deep Q-Network framework [2]. This allows the agent to approximate value functions using neural networks, enabling scalable learning in more complex or stochastic versions of the environment.

We established the core reinforcement learning high-level architecture, as shown in Figure 1, defining sensors, actuators, and a structured reward system within environment using Gymnasium and PyTorch. Then, we applied a system dynamics perspective to analyze the feedback loops governing agent behavior. Based on this analysis, we introduced the continuous input variable danger level and redesigned the reward framework to provide more granular, informative feedback.

The ultimate goal of the project is to promote adaptive, efficient, and intelligent behavior in the agent through iterative improvements in perception, control, and learning. By combining system modeling techniques with reinforcement learning, we aim to create a more robust agent capable of operating in dynamic, unpredictable settings while avoiding undesirable behaviors such as infinite loops or unsafe exploration.

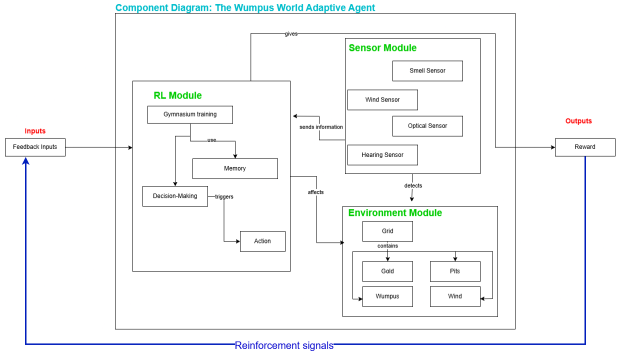

Fig. 1. Component diagram.

II. Method and Materials

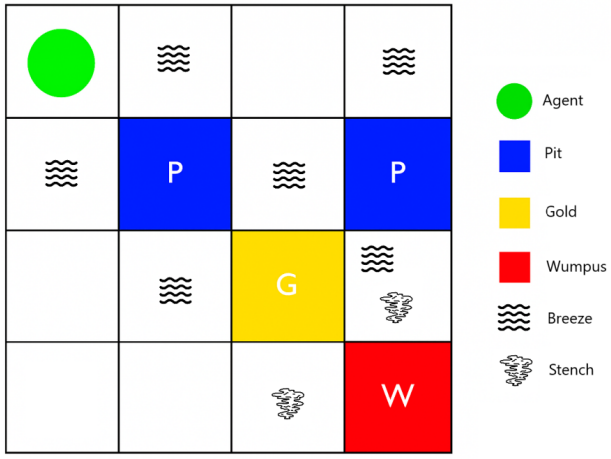

Fig. 2. Diagram of the Environment Layout

The proposed system consists of a reinforcement learning agent operating in a custom implementation of the Wumpus World environment. The environment is modeled as a 4x4 grid, where each cell may contain a hazard (such as pits or the Wumpus) or a reward (gold). The agent begins at the bottom-left cell and must navigate the environment based solely on local perceptual inputs—breeze, stench, and glitter—that indicate nearby hazards or objectives. This setup creates a partially observable and high-risk environment, requiring the agent to learn and adapt through trial-and-error interactions.

The agent functions as a cybernetic system: it perceives input from the environment, takes an action, receives a reward, and adjusts its internal decision policy accordingly. This closed feedback loop underpins the learning process and reflects theoretical principles of adaptive systems and control theory using reinforcement learning [5], specifically a Deep Q-Network (DQN), to estimate action values from observed states. The input to the DQN consists of the agent's current position, perceptual signals, and orientation. The output represents Q-values for six discrete actions.

The environment is implemented using the Gymnasium library, which allows for modular simulation of reinforcement learning tasks with configurable observation and action spaces. The agent's neural architecture is constructed using PyTorch, which enables efficient training, backpropagation, and experimentation with different learning configurations. The reward structure is carefully shaped to penalize fatal or inefficient actions, while rewarding successful exploration, avoidance of hazards, and completion of the goal.

Real-time visualization is achieved through Pygame, which renders the grid, elements, and agent status at each time step. This allows both intuitive debugging and behavioral interpretation of the agent's strategy. The combination of Gymnasium, PyTorch, and Pygame ensures reproducibility, interpretability, and extensibility.

A. Technical decisions

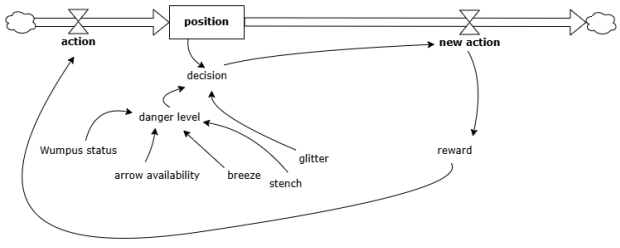

Fig. 3. Agent–Environment Feedback Loop

The conceptual development of the proposed system is guided by several key technical decisions that align with principles from both reinforcement learning and cybernetic systems theory. First, perceptual modeling plays a central role: the agent receives binary sensory inputs such as breeze, stench, and glitter, which provide limited yet essential information about nearby environmental features. These percepts form the foundation for the agent's decision-making under uncertainty.

Second, the reward structure is designed to offer granular feedback. Rather than relying solely on sparse terminal rewards, the system introduces intermediate signals that reinforce safe exploration and penalize ineffective or repetitive actions. This shaping is expected to facilitate more stable learning trajectories.

Third, the entire agent–environment interaction is conceptualized as a feedback loop. The agent interprets perceptual input, selects an action, receives a reward, and updates its internal policy based on this experience. This cyclic structure directly reflects cybernetic control models, in which regulation and adaptation are driven by continuous information exchange.

Fourth, a deep Q-network architecture is selected for learning. The network is planned to consist of multiple fully connected layers with non-linear activations, enabling it to approximate complex value functions that map states to expected future rewards. PyTorch is selected for its flexibility in constructing and training these networks. [6]

Finally, special attention is paid to the issue of stability. The design prioritizes mechanisms such as experience replay and target network separation to reduce training variance and support policy convergence. This focus on system equilibrium, feedback regulation, and behavioral consistency positions the project within the broader scope of dynamic system modeling and control.

The overall architecture is chosen to enable both pedagogical clarity and experimental flexibility. The agent's learning process can be visualized and interpreted in real-time, supporting a deeper understanding of reinforcement learning dynamics, cybernetic feedback, and system stability.

III. Results and Discussion

The proposed agent design is intended to be evaluated through structured testing at multiple levels. Once implemented, the system will undergo unit tests targeting individual components such as environment initialization, state encoding, percept interpretation, and reward computation. The philosophy behind these unit tests is to ensure that each module behaves correctly under both normal and edge-case conditions. It is estimated that 10 to 15 unit tests will be conducted, focused on core environment logic and agent decision-making.

Integration tests will evaluate the interaction between modules, particularly the continuous perception-action-feedback loop that underlies the system. These tests will help ensure that transitions between states, feedback signals, and learning updates work cohesively. Acceptance tests will validate whether the agent is capable of solving the environment's main objective—retrieving the gold and returning to the initial cell—under a variety of conditions.

Key evaluation metrics include the cumulative reward obtained per episode, the number of steps required to reach the goal, the entropy of the agent's action distribution (as an indicator of decision confidence), and the success rate across different environment configurations.

In early episodes, the agent is expected to exhibit erratic or dangerous behaviors, such as entering pit-adjacent cells

or revisiting explored zones. As learning progresses, the frequency of fatal outcomes should decrease, and the agent should develop more direct, cautious trajectories toward the goal. Visualization of trajectories across episodes will help validate behavioral improvement and verify whether the agent avoids cyclic or redundant behavior.

To assess generalization, the system will also be tested under varying random seeds and alternate grid configurations. These tests are intended to examine whether the agent's policy overfits specific layouts or develops transferable strategies that succeed under novel conditions.

Although implementation is not yet completed, planned baseline tabular Q-learning agent [3] will help determine whether the use of deep neural networks contributes to faster convergence or improved generalization.

IV. Conclusion and Future Work

The design presented in this paper bridges reinforcement learning theory and cybernetic modeling, offering a hybrid perspective for intelligent agent design. By embedding system feedback into both the reward structure and perceptual processing, the agent gains the capacity to adjust its behavior iteratively and autonomously.

This paper presented the conceptual design of a cybernetic reinforcement learning agent for the Wumpus World environment. By integrating reinforcement learning with cybernetic principles such as feedback, control, and adaptation, the proposed system aims to model intelligent behavior in a constrained and uncertain environment. The architecture includes modular components for perception, learning, and decision-making, with emphasis on feedback loops and performance stability.

Although implementation is still in progress, the design is supported by a solid evaluation plan based on unit, integration, and acceptance tests. The use of metrics such as entropy and reward accumulation will provide insight into the learning dynamics and reliability of the system.

Future work will focus on completing the agent's implementation and executing the proposed testing strategy. Additional enhancements may include memory of past percepts, exploration of symbolic reasoning layers, and testing in more complex or randomized environments. This work contributes both as a reinforcement learning prototype and as an educational model for dynamic system design.

References

[1] M. Genesereth and N. Nilsson, *Logical Foundations of Artificial Intelligence*. Morgan Kaufmann, 1987.

[2] V. Mnih, K. Kavukcuoglu, D. Silver, A. Rusu, et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[3] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, pp. 279–292, 1992.

[4] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Pearson, 2020.

[5] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. MIT Press, 2018.

[6] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. of NeurIPS*, 2019.