Dynamical System Design for an Autonomous Agent in the Game 2048

Miguel Alejandro Chavez Porras
20212020033
Universidad Distrital Francisco José de Caldas
Email: machavezp@correo.udistrital.edu.co

Edward Julian Garcia Gaitan
20212020136
Universidad Distrital Francisco José de Caldas
Email: ejgarciag@correo.udistrital.edu.co

***Abstract*—This paper presents a dynamical systems analysis of an autonomous agent designed to play the game 2048. Building upon prior work in reinforcement learning and system modeling, we extend our framework to incorporate non-linear dynamics, feedback loop refinements, and criteria for evaluating system stability and convergence. The agent operates within a stochastic, discrete-time environment where state evolution is influenced by both decision-making and random factors. The proposed model highlights how sensitivity to initial conditions and chaotic regimes impact long-term performance and outlines methods for improving adaptability and robustness through enhanced feedback mechanisms.**

I. Dynamical Systems Framework

The 2048 game environment can be abstracted as a discrete-time dynamical system. At each time step t, the agent transitions from one board configuration to another based on its selected action and the inherent randomness of the game. The system state S_t represents the full 4x4 grid of tile values and positions. The system evolves as:

$$S_{t+1} = f(S_t, A_t, R_t) \tag{1}$$

where A_t is the agent's action at time t, and R_t encapsulates the random tile generation component. The function f describes the transition dynamics governed by game rules.

The system is non-linear due to:

- The multiplicative nature of tile merges (powers of two).
- The probabilistic behavior of R_t, which alters the board in ways that compound over time.

Equilibrium in this system is defined not as a fixed state, but as a recurring strategy that consistently achieves high scores or regularly produces high-value tiles. Conversely, chaotic behavior emerges when strategies are too reactive or fail to control the system's randomness, leading to divergent outcomes.

II. Chaos and Sensitivity to Initial Conditions

The system exhibits a strong sensitivity to initial conditions. Small variations in the early placement of tiles, which result from R_t, can cause major differences in future states. As such, two nearly identical initial states can lead to dramatically different results.

This phenomenon requires performance evaluation on multiple simulation runs with varied seeds, highlighting the need for statistical validation. The chaotic regime makes deterministic predictions impractical, emphasizing the agent's need to generalize its policy across a wide range of scenarios.

III. Feedback Loop Refinement

A core component of this system is the feedback loop between the action of the agent and the environmental response. We propose the following refinements:

- **Granular Reward Perception:** Beyond immediate tile merges, the agent should consider board mobility, future merger opportunities, and tile clustering.
- **Adaptive Evaluation Criteria:** As the board becomes more complex, internal evaluations should weigh stability and long-term options more heavily.

These changes support a dynamic response mechanism that improves the agent's regulation in uncertain or rapidly changing states.

IV. Stability and Convergence

Stability refers to the agent's ability to avoid degenerate strategies and consistently maintain non-trivial performance levels. Convergence indicates a tendency to improve the score or the value of the tile over successive games. We define:

A. Stability Criteria

- Low variance in total score across runs.
- Rare occurrence of early game failures.
- Consistent behavioral patterns across different starting states.

B. Convergence Indicators

- Gradual increase in maximum tile values over time.
- Reduction in performance oscillations.

These properties suggest that the agent is learning to operate reliably under the dynamic conditions of 2048.

V. Iterative Design and Testing Outline

To ensure the continued refinement of the agent, we propose the following iterative development cycle:

- Formalize internal models to store and relate state transitions.
- Include evaluation mechanisms that score hypothetical future board states.

- Perform simulations with varying initial states to assess robustness.
- Analyze long-term trends in behavior using summary statistics and qualitative observations.

This approach fosters not only reactivity but also anticipatory behavior, laying the groundwork for a more resilient agent.

VI. Non-linear Dynamics Analysis

The 2048 environment exhibits complex non-linear behavior that can be characterized through several key aspects:

- **State-space complexity:** The state space of the game grows exponentially with each new tile, creating a challenging environment for decision-making.
- **Attractor-like patterns:** Certain board configurations tend to emerge repeatedly during successful gameplay, representing metastable states that facilitate higher scores.
- **Transition thresholds:** Critical points exist where small differences in strategy lead to qualitatively different outcomes (success vs failure).

Understanding these non-linear characteristics is essential for developing effective agent strategies that can navigate the game's complexity while maintaining stability.

VII. Phase Diagram

The graphs are merely illustrative, so the equations its not part of the system. The bifurcation at the critical point ($x = 3$) represents a threshold in the agent's performance: beyond this score, small differences in decision–making are amplified exponentially. If the agent's policy is effective, it follows the "good path" and its effectiveness grows; if it is suboptimal, it follows the "bad path" and effectiveness decays. This symmetric divergence illustrates a sensitivity typical of non-linear dynamical systems.

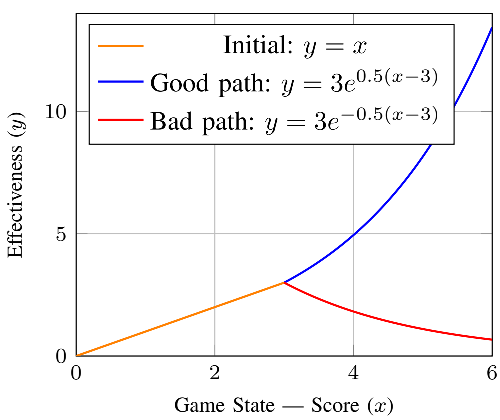

Fig. 1. Phase portrait of agent effectiveness: linear growth ($y = x$) until bifurcation, then symmetric exponential divergence with rate κ.

VIII. Project Updates for Dynamic Modeling

In our updates for the modeling, we found the following:

Applying classic agile frameworks like SCRUM to AI and ML projects can be challenging due to the difficulty of forecasting the time required to produce a functional model. Nevertheless, we will adopt an **iterative development** mindset by exploring two complementary reinforcement learning methods: Q-Learning and Deep Q-Networks (DQN).

- **Q-Learning Phase:** We will begin with tabular Q-Learning to develop a simple but explainable agent. This phase allows rapid prototyping and clear insight into convergence properties.
- **DQN Phase:** Once a baseline is established, we will transition to Deep Q-Networks to handle the large state space of 2048. Neural networks will approximate the Q-function, enabling the agent to generalize better across unseen board configurations.

By alternating between these two methods and evaluating performance at each iteration, we ensure both explainability and scalability in our solution.

IX. Conclusions

The 2048 environment presents a rich testbed for exploring dynamic system behavior in autonomous agents. Through a combination of non-linear modeling, refined feedback loops, and stability criteria, we enhance the foundation laid in Workshop 1. Our analysis demonstrates how dynamical systems theory provides valuable insights for agent design in stochastic environments. Future work will incorporate empirical data to validate convergence and further explore the system's non-linear properties through computational experiments.

Repository Structure

All files, simulation notes, diagrams, and references related to this workshop can be found in the GitHub repository under the folder `Workshop-2`.

References

[1] S. Ravichandiran, *Hands-on Reinforcement Learning with Python*, Packt Publishing, 2020.

[2] N. Kondo and K. Matsuzaki, "Playing game 2048 with deep convolutional neural networks trained by supervised learning," *Journal of Information Processing*, vol. 27, pp. 340–347, 2019.

[3] Supahub, "AI Feedback Loop: What It Is, How It Works, and Its Importance." [Online]. Available: https://supahub.com/glossary/ai-feedback-loop

[4] 2048game.com, "2048 and Math." [Online]. Available: https://2048game.com/blog/2048/2048-and-math