Bamboo GPU Mode: A Game-Theoretic Evolutionary Framework for Resilient Self-Healing Navigation in Extreme Mars DEM Environments

Hiroaki Ueshima
Independent Researcher, Nagaoka, Kyoto, Japan

January 2026

Abstract

Preprint version: v2
DOI: https://doi.org/10.5281/zenodo.18279778

Traditional autonomous navigation systems for planetary rovers, such as A* or Dynamic Window Approach (DWA), frequently experience irreversible failure in extraterrestrial environments due to extreme sensor noise (e.g., dust storms) and unpredictable terrain features (e.g., sand traps).

Bamboo GPU Mode is a resilience-oriented framework that prioritizes system recovery over conventional path optimization. It integrates a game-theoretic payoff matrix with a phototropic mutation mechanism, modeling navigation as a population-based evolutionary process among virtual agents.

System health is quantified using the **Resilience Index (R)**:

$$R = \int_{t_{\text{crash}}}^{t_{\text{recovery}}} \frac{\text{Share}_H(t)}{D_{\text{sys}}(t)}\, dt \tag{1}$$

where $\text{Share}_H(t)$ is the harmony/coordination score (0 to 1.0), and $D_{\text{sys}}(t)$ is aggregate disturbance (noise + terrain complexity).

GPU-accelerated parallel iteration enables real-time evolutionary execution on edge hardware such as NVIDIA Jetson Orin Nano. Simulations on synthetic Mars DEM show **100% revival success** in 20 forced-crash tests, even at $\sigma = 0.6$ noise, recovering to $\text{Share}_H = 1.0$.

This framework offers a self-healing paradigm for Artemis, ExoMars, and future deep-space missions.

1 Introduction

1.1 The Problem

In planetary exploration, rovers often encounter catastrophic failures—such as entrapment in sand pits or complete sensor blackout during dust storms—that lead to mission termination. Conventional reactive planners (e.g., A*, D*) rely on deterministic assumptions and lack inherent mechanisms for post-failure recovery, resulting in high vulnerability in unstructured extraterrestrial terrains.

1.2 The Inspiration

Drawing from the mechanical resilience of bamboo, which bends under extreme pressure yet spontaneously rebounds due to its internal dynamics, this work seeks to imbue robotic navigation systems with analogous self-restorative properties.

1.3 The Innovation

We introduce the first framework that synergistically combines game-theoretic Nash equilibrium stability (via a parameterized payoff matrix) with evolutionary survival selection (via phototropic mutation), explicitly optimized for GPU parallel computing to achieve real-time self-healing on space-grade hardware.

2 Theoretical Framework

2.1 The Payoff Function

Each virtual agent i evaluates its survival prospects in the current terrain state via a payoff function:

$$f_i = C \cdot \mathrm{eff}(d_i, D_{\mathrm{sys}}) - \beta(d_i - D_{\mathrm{sys}})^2 - \alpha d_i \tag{2}$$

where C is a constant scaling the baseline survival benefit, $\mathrm{eff}(d_i, D_{\mathrm{sys}})$ is an efficiency term reflecting alignment with system-wide disturbance, β penalizes deviation from the global disturbance level D_{sys} (promoting coordination), and α applies a linear cost to individual displacement d_i.

This quadratic penalty structure encourages convergence toward a Nash equilibrium, where no agent benefits from unilateral deviation, thereby stabilizing collective navigation decisions under noise.

2.2 Phototropic Mutation

Upon catastrophic failure (system crash detection), the framework triggers a directed mutational bias inspired by plant phototropism—growth toward light as a metaphor for seeking optimal states amid adversity:

$$d_{\mathrm{new}} = d_{\mathrm{old}} + \Delta m(\epsilon), \quad \epsilon = +0.05 + \mathcal{N}(0, \sigma_{\mathrm{mut}}) \tag{3}$$

The small positive bias $+0.05$ serves as the core "restorative impulse," systematically nudging agents toward higher-payoff configurations rather than pure randomness. This directed perturbation, combined with GPU-parallel evaluation of mutation offspring, enables rapid exploration and convergence to a recovered state ($\mathrm{Share}_H \to 1.0$) in few iterations.