Tight Regret Bounds for Mean-Reverting Linear Bandits via Recursive State Estimation

[1]Cloud Computing Research Institute, China Telecom

[2]Department of Computer and Information Sciences, Temple University

Linghang Meng [1]
menglh1@chinatelecom.cn

Jie Wu [1,2]
jiewu@temple.edu

ICASSP2026 Barcelona

Take-home Message

Non-stationary bandits are not always arbitrary: when latent rewards follow mean-reverting dynamics, recursive state estimation enables tight regret characterization and improved online decisions.

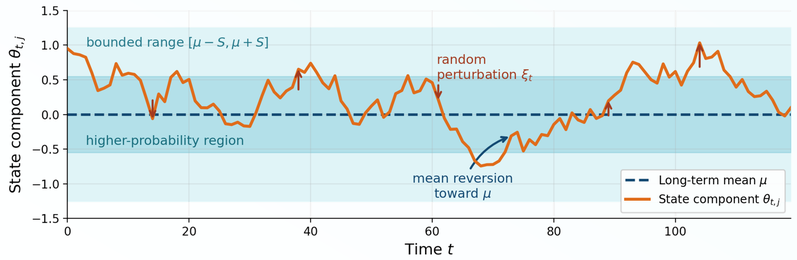
Main Contributions

- Formulate linear bandits with latent mean-reverting dynamics and exploit their temporal structure.
- Establish lower bounds for linear and bounded mean-reverting dynamics.
- Recursive state estimation method via Kalman filtering and particle filtering
- Provide tight steady-state regret characterization and empirical validation

Motivation: Structured Non-Stationary

Many non-stationary bandit algorithms treat environmental changes as arbitrary shifts or bounded total variation. However, in many signal processing, communication, control, and cloud-network systems, the latent state fluctuates around a stable operating point rather than drifting freely.

Time-structure-agnostic view	Mean-reverting view
Environment may change arbitrarily	State fluctuates around long-term mean
Use discounting, sliding windows, or resets	Use recursive state estimation
Regret depends on variation/shifts	Regret characterized by dynamics/noise



Problem Formulation

Reward Model

$$y_t = x_t^\top \theta_t + \eta_t, \qquad x_t \in \mathcal{D} \subset \mathbb{R}^d$$

Non-Stationary Environment

- Linear mean-reverting dynamics

$$\theta_{t+1} = (1-\rho)\theta_t + \rho\mu + \xi_t$$

- Bounded nonlinear dynamics

$$\theta_{t+1} = \Pi_{\mu,S}((1-\rho)\theta_t + \rho\mu + \xi_t)$$

Objective

$$\bar{r} = \lim_{T\to\infty} \frac{R_T}{T}$$

Recursive State Estimation Method

The learner maintains an online estimate of the latent state and uses it to select the greedy action under the current estimate.

Predict $\hat{\theta}_t$ → $x_t = \underset{x\in\mathcal{D}}{\operatorname{argmax}}\, x^\top \hat{\theta}_t$ → Observe y_t → Update posterior

Two Filters for Two Dynamics

For linear dynamics: Kalman Filter

$$\hat{\theta}_t = (1-\rho)\bar{\theta}_{t-1} + \rho\mu, \bar{\theta}_t = \hat{\theta}_t + K_t(y_t - x_t^\top \hat{\theta}_t)$$

For bounded dynamics: Particle Filter

$$\theta_t^{(i)} = \Pi_{\mu,S}\left((1-\rho)\theta_{t-1}^{(i-1)} + \rho\mu + n_t^{(i)}\right), \omega_t^{(i)} \propto \omega_{t-1}^{(i)} \exp\left(-\frac{\left(y_t - x_t^\top \theta_t^{(i)}\right)^2}{2\sigma_e^2}\right)$$

Theoretical Characterization

Key insight: unavoidable regret comes from unobserved state innovations, while the proposed algorithm controls regret through recursive filtering error.

Fundamental Limits for the Decision Process

- Random state evolution makes the next optimal action partially unpredictable:

Thm.1: Lower bound for linear Mean-Reverting Process

$$\mathbb{E}[\bar{r}] \geq \sqrt{2d/\pi}\sigma L\sqrt{\rho/(2-\rho)}$$

- The lower bound depends on the steady-state boundary mass p_s and the conditional regret R_s.

Thm.2: Lower bound for Bounded Process

$$\mathbb{E}[\bar{r}] \geq p_s R_s$$

p_s: steady-state mass on sphere;
R_s: regret conditioned on boundary states

Regret Upper Bound for Proposed Algorithm

- The bound is optimal with respect to σ, L and $\sqrt{d}$, and becomes close to the lower bound when the mean reversion rate is large.

Thm.3: Regret Upper Bound for Linear Case

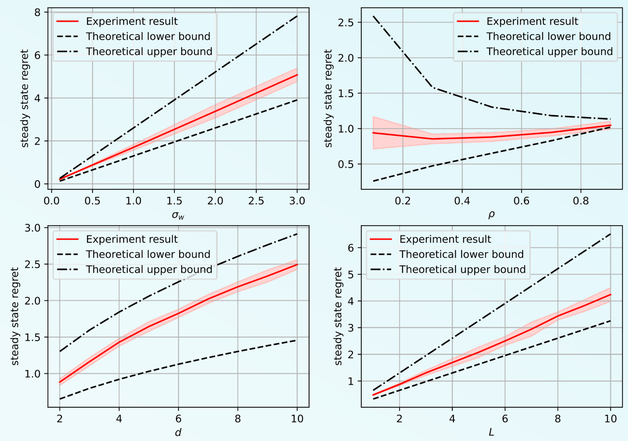
$$\mathbb{E}[\bar{r}] \leq C \cdot \sigma L\sqrt{d/(\rho(2-\rho))}$$

Experiments

Theory Matches Simulation

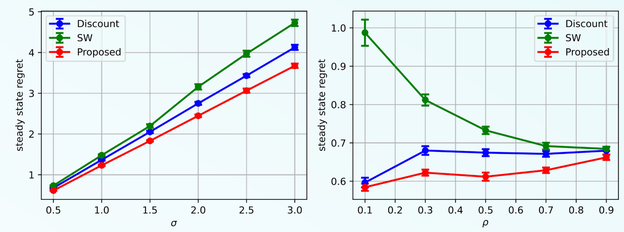
Empirical steady-state regret lies between the theoretical lower and upper bounds and follows the predicted scaling laws.

- Regret grows linearly with σ, L, and $\sqrt{d}$;
- For large ρ, the upper bound is close to the lower bound.



Comparison with Time-Structure-Agnostic Baselines

Linear dynamics: Proposed Kalman-based method vs. baselines



Bounded nonlinear dynamics: Particle-filter-based method vs. baselines