Globally Optimal Policy Gradient Algorithms for Reinforcement Learning with PID Control Policies

Vipul K. Sharma, Wesley A. Suttle,
Sivaranjani Seetharaman

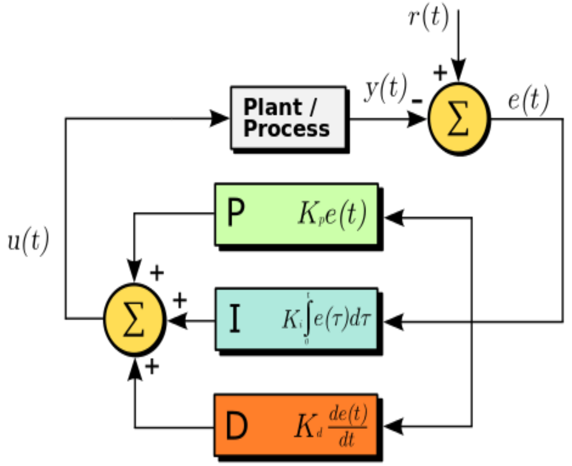
sharma1256/RL-optimal-pid

Formulation, SOTA, and Challenges

Optimal PID Control

$$\min \mathbb{E}\left[\sum_{t=0}^{\infty} y_t^T Y y_t + u_t^T R u_t\right]$$
$$s.t. \quad x_t = Ax_t + Bu_t,$$
$$y_t = Cx_t,\ x_0 \sim \mathcal{D}$$

Proportional-Integral-Derivative (PID) Control



This Photo by Unknown Author is licensedunder CC BY-SA

$$u_t = -K_P y_t - K_D \frac{y_{t+1} - y_t}{\tau}$$
$$-K_I \sum_{j=0}^{t-1} y_j$$

Why PID?

- $> 90\%$ of the industries uses PID Control
- PID - low-dimensional parameterization
- Achieves perfect tracking
- Robust to model uncertainty

Challenges

- SOTA PID tuning methods are heuristic, no optimality
- RL does not employ PID policies
- PID policy gradient expressions unavailable
- Policy gradient (PG) theory difficult to adapt to deterministic policies and dynamics
- Non-convex optimization landscape

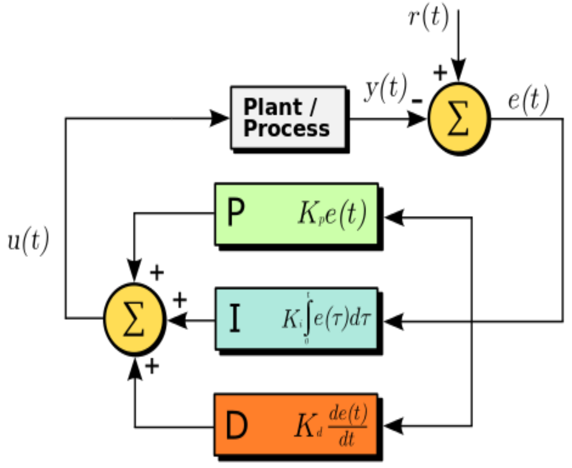
Formulation, SOTA, and Challenges

Optimal PID Control

$$\min \mathbb{E}\left[\sum_{t=0}^{\infty} y_t^T Y y_t + u_t^T R u_t\right]$$

$$s.t. \quad x_t = Ax_t + Bu_t,$$
$$y_t = Cx_t,\ x_0 \sim \mathcal{D}$$

Proportional-Integral-Derivative (PID) Control



$$u_t = -K_P y_t - K_D \frac{y_{t+1} - y_t}{\tau} - K_I \sum_{j=0}^{t-1} y_j$$

Contributions

- Provably optimal and convergent model-based and model-free RL with PID policies
- Outperforms RL with PPO and LQR for benchmark control environments

Technical Innovations

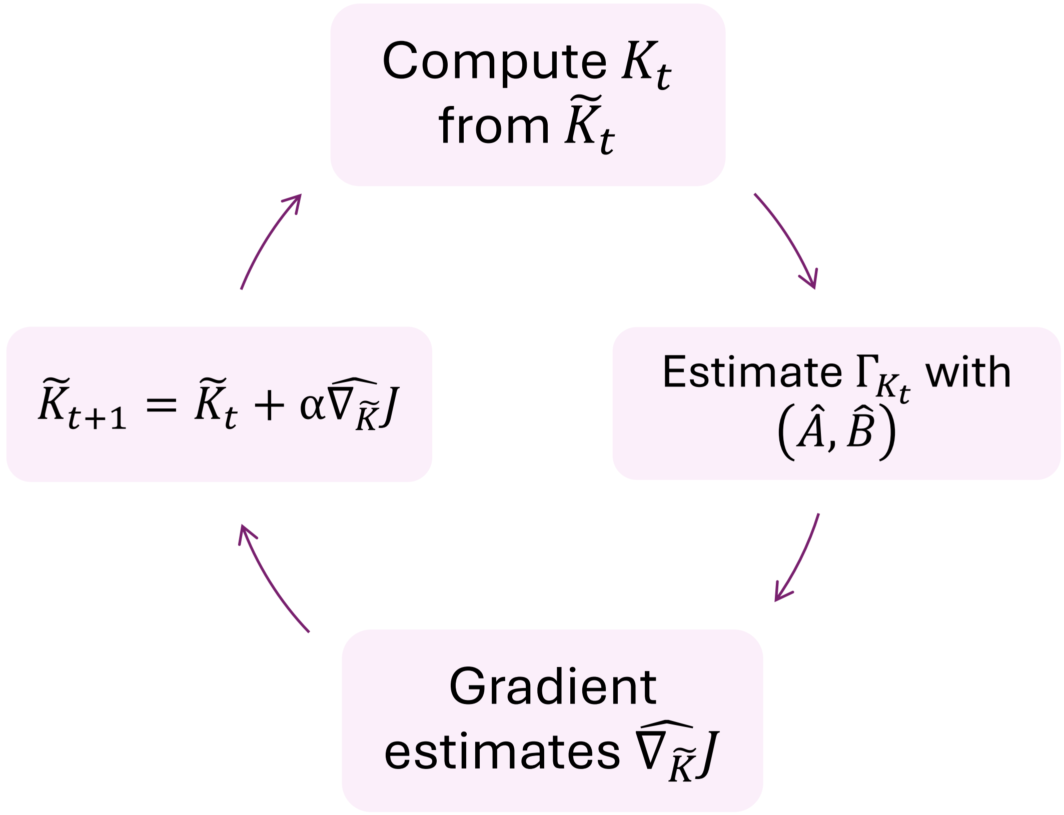
- Closed-form policy gradient expressions in the PID parameters
- Deterministic dynamics and control policies - novel concept of "stochastic wrappers", extending classical PG theorem to deterministic policies with stochastic updates
- Non-convex optimization landscape: weak gradient dominance, convergence, and sample complexity guarantees

Model based RL - PG4PID

Novel PID Policy Gradient Expressions

$$\nabla_{K_P} J = 2F_D E_K \Sigma_K T_x^T C^T$$
$$\nabla_{K_I} J = 2F_D E_K \Sigma_K T_z^T$$
$$\nabla_{K_D} J = 2F_D E_K \Sigma_K (\alpha_1 T_x^T (A - I) C^T - \alpha_2 \alpha_3)$$

- PID parameters $\widetilde{K} = [K_P \quad K_I \quad K_D]^T$
- Gradients depend on system (A, B)
- System Identification to obtain $(\hat{A}, \hat{B})$
- Gradient estimates based on $(\hat{A}, \hat{B})$
- Gradient Descent algorithm with appropriate step size α



Model free RL - PG4PI

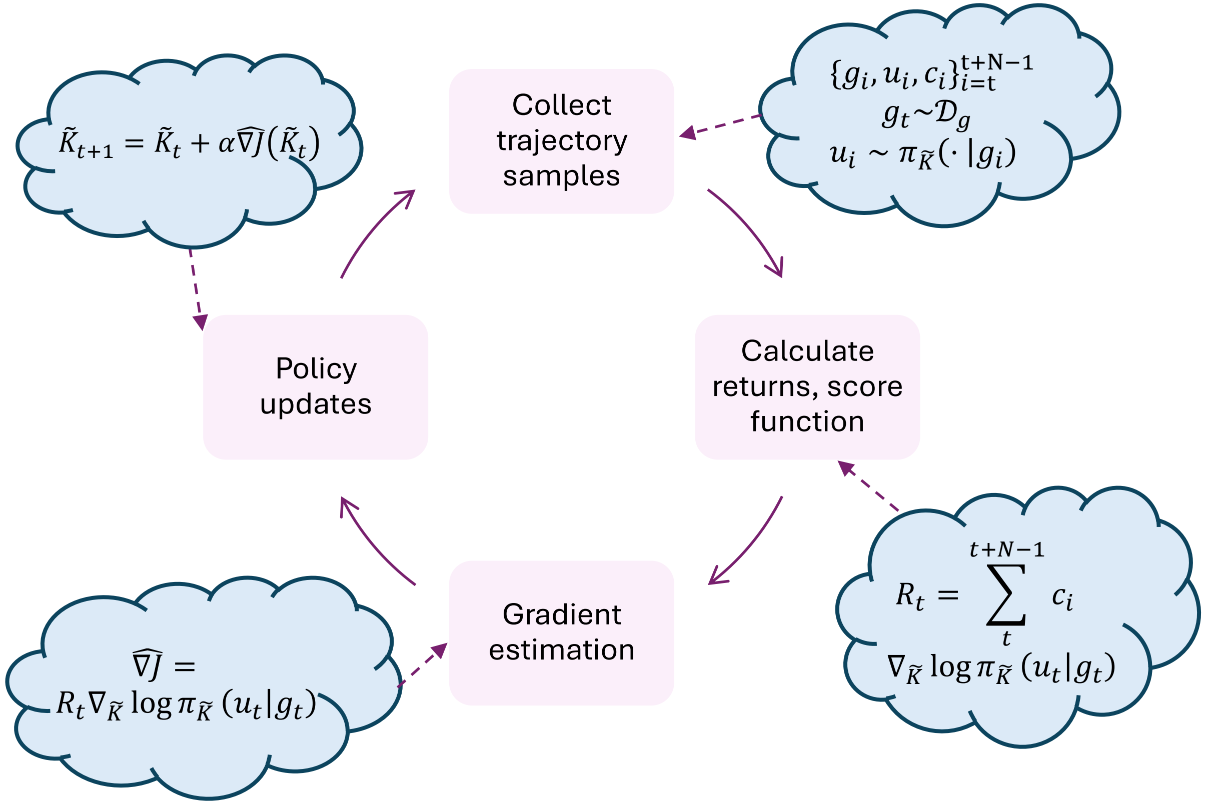
Stochastic Wrappers for Exploration

- Gradient expressions without (A, B) using policy gradient theorem
- PI parameters $\widetilde{K} = [K_P \quad K_I], K_D = 0$
- Deterministic PID policy $\mu_{\widetilde{K}}(g) = -Kg,$

$$K = [K_P C \quad K_I]$$

- Stochastic wrappers - allows us to update deterministic PI parameters using stochastic gradient evaluations

$$\pi_{\widetilde{K}}(u|g) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{\left(u-\mu_{\widetilde{K}}(g)\right)^T\left(u-\mu_{\widetilde{K}}(g)\right)}{2\sigma^2}}$$



Convergence

- First convergence guarantees for PID policies
- PG4PID – Cost J_μ approaches optimal J_μ^* as t $\rightarrow \infty$

Theorem (Informal): With appropriate steps size η and constant $\kappa \in (0,1)$, we have the following contraction

$$J_\mu(\widetilde{K}_{\mathrm{t}}) - J_\mu^* \leq \kappa^t\left(J_\mu(\widetilde{K}_0) - J_\mu^*\right)$$

Convergence

- First convergence guarantees for PID policies
- PG4PID – Cost J_μ approaches optimal J_μ^* as t $\rightarrow \infty$

Theorem (Informal): With appropriate steps size η and constant $\kappa \in (0,1)$, we have the following contraction

$$J_\mu(\widetilde{K}_{\mathrm{t}}) - J_\mu^* \leq \kappa^t \left(J_\mu(\widetilde{K}_0) - J_\mu^*\right)$$

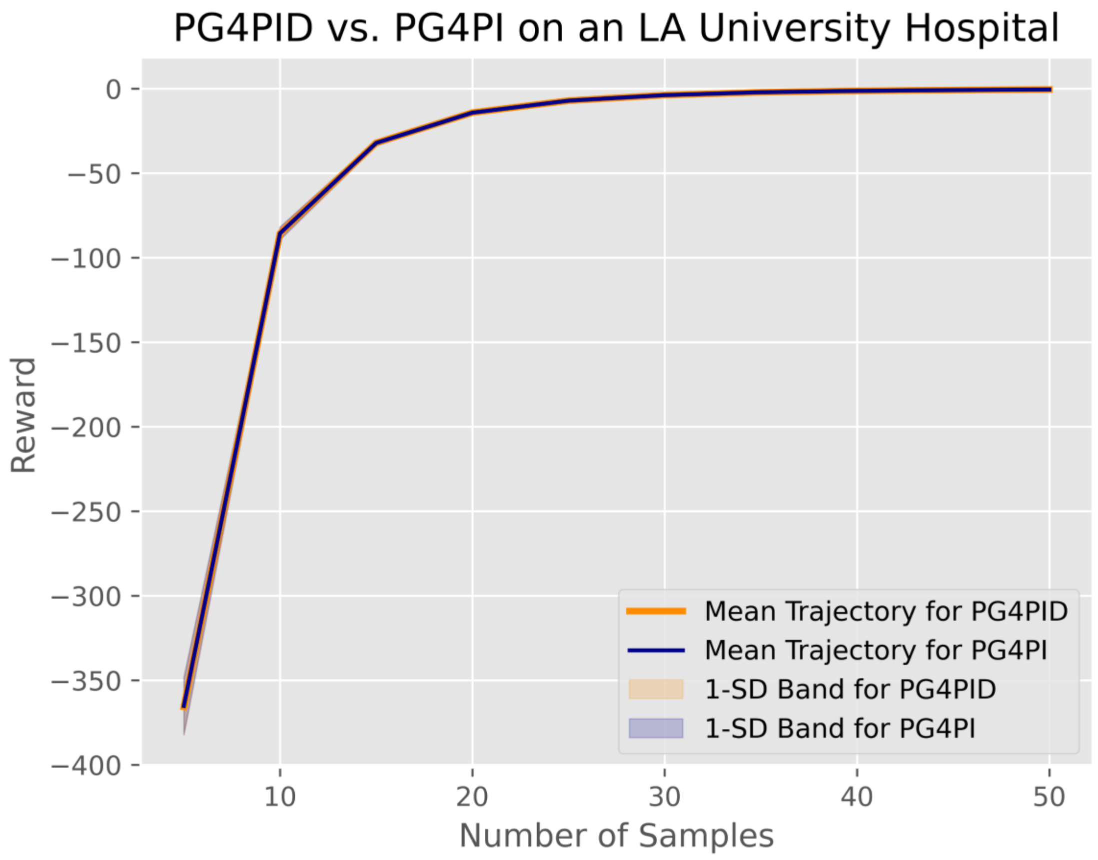
- PG4PI – Cost J_π with optimal value J_π^* with weak gradient dominance

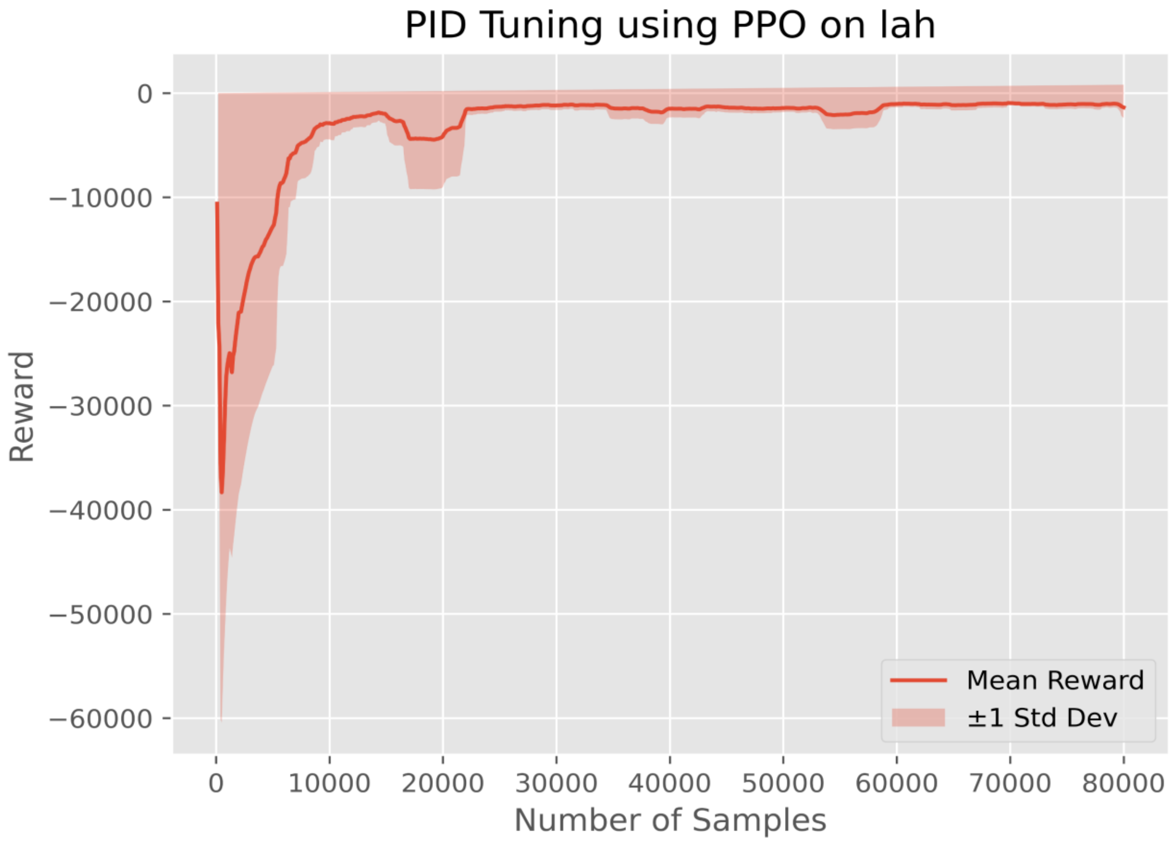
Theorem (Informal): Under mild conditions, with appropriate step size η, rollout length N, to guarantee $\epsilon -$ neighbourhood of sub-optimality, the sample complexity, in T time-steps, is

$$NT = \frac{16\tilde{\alpha}^4 l_{mx}^T \bar{V}}{\epsilon^3} log\left(\frac{c_T}{\epsilon}\right)$$

Benchmark vs. PPO

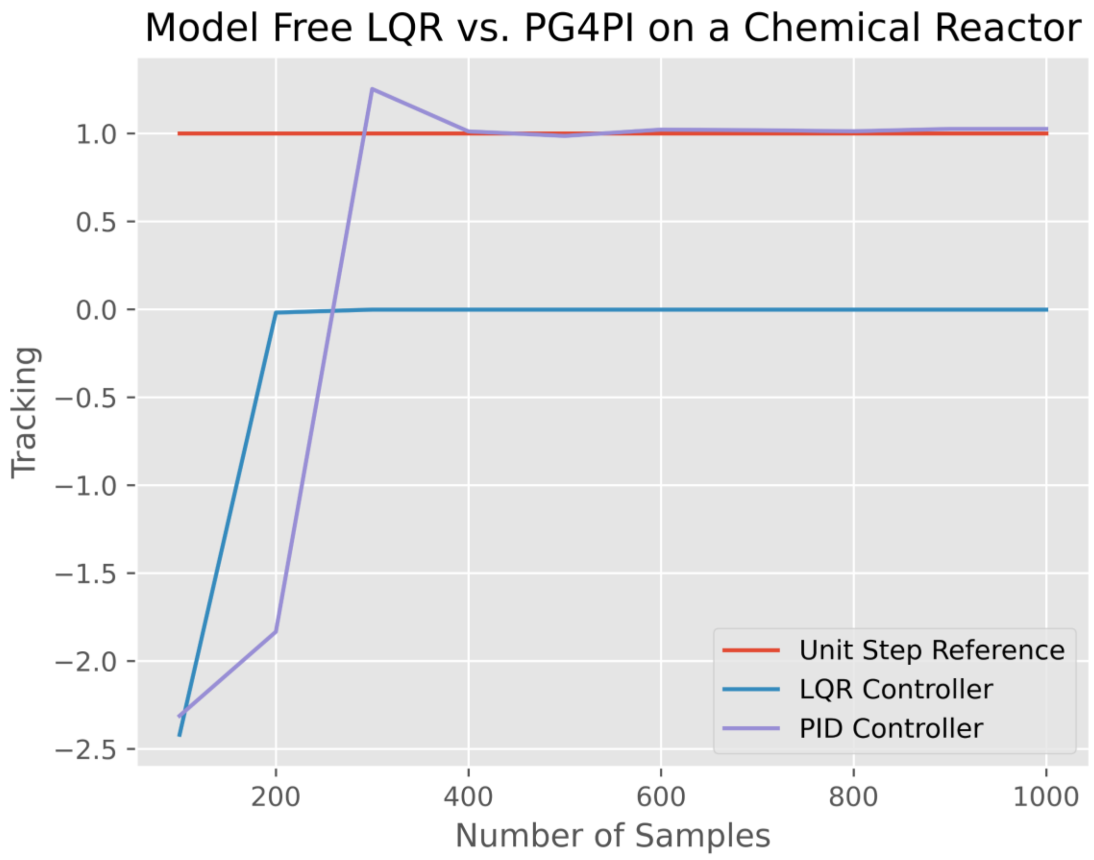
An intelligent RL policy structure yields faster learning on a 48 −dimensional environment

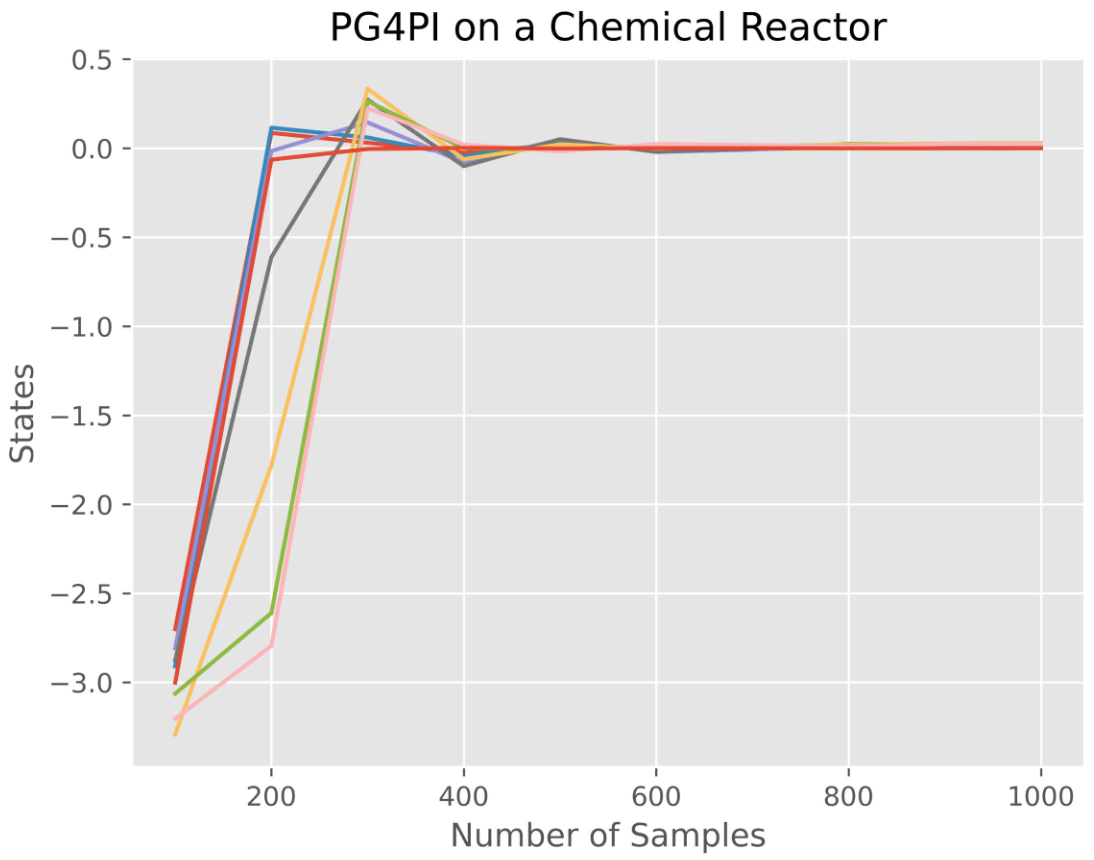




Benchmark vs. LQR : Tracking

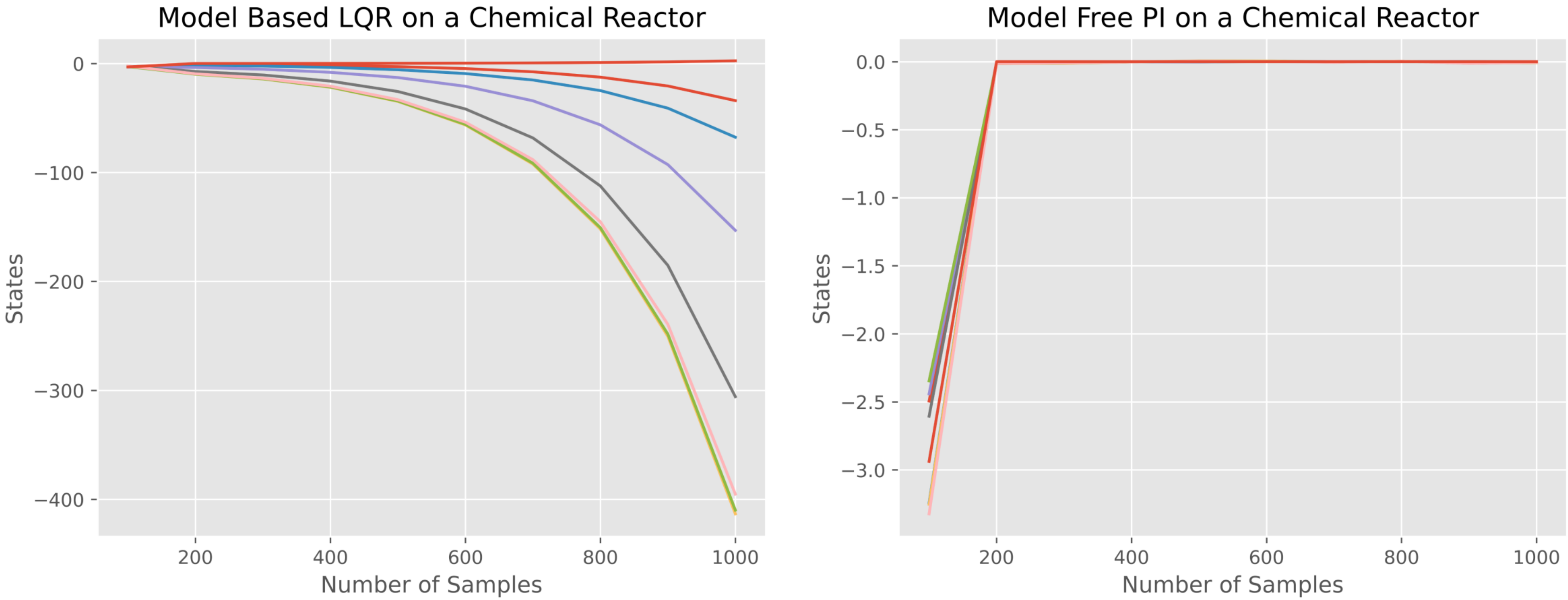
PID achieves perfect tracking even in the model-free setting, while LQR has non-zero steady state error





Benchmark vs. LQR : Robustness

PID is robust to model error while LQR is fragile even for an error in the matrix A as $||\Delta A|| < 0.05$



Conclusions

- What if we pick an intelligent policy parameterization in RL?
- PID policies widely used in industrial applications
- An RL formulation with PID policies
- Novel PID policy gradient expressions
- Model-based PID tuning algorithm (PG4PID)
- Stochastic wrappers for model-free PI tuning algorithm (PG4PI)
- Gradient dominance conditions in PID parameter space
- Optimality and Convergence guarantees for both PG4PID and PG4PI
- PID policy in RL yields faster learning than PPO for tuning PID
- PID more robust than LQR, achieves better tracking performance

Paper

Github Repo