RL agent for PID tuning

May 14, 2025

Anh Tran

Actor and Critics

Q-value *Q(s, a)*: expected return (total reward) if we take action *a* in state *s*, and follow the policy thereafter.

The model has 2 main blocks:

- **Critic**: evaluates how good the chosen action is by estimating the expected future reward (Q-value)
- **Actor**: chooses the best action for a given state by learning a policy (control input in our case)

PID tuning problem

Control input (action):

$$u = \left[\int e\,dt \quad e \quad \frac{de}{dt}\right] \cdot \begin{bmatrix} K_i \\ K_p \\ K_d \end{bmatrix}$$

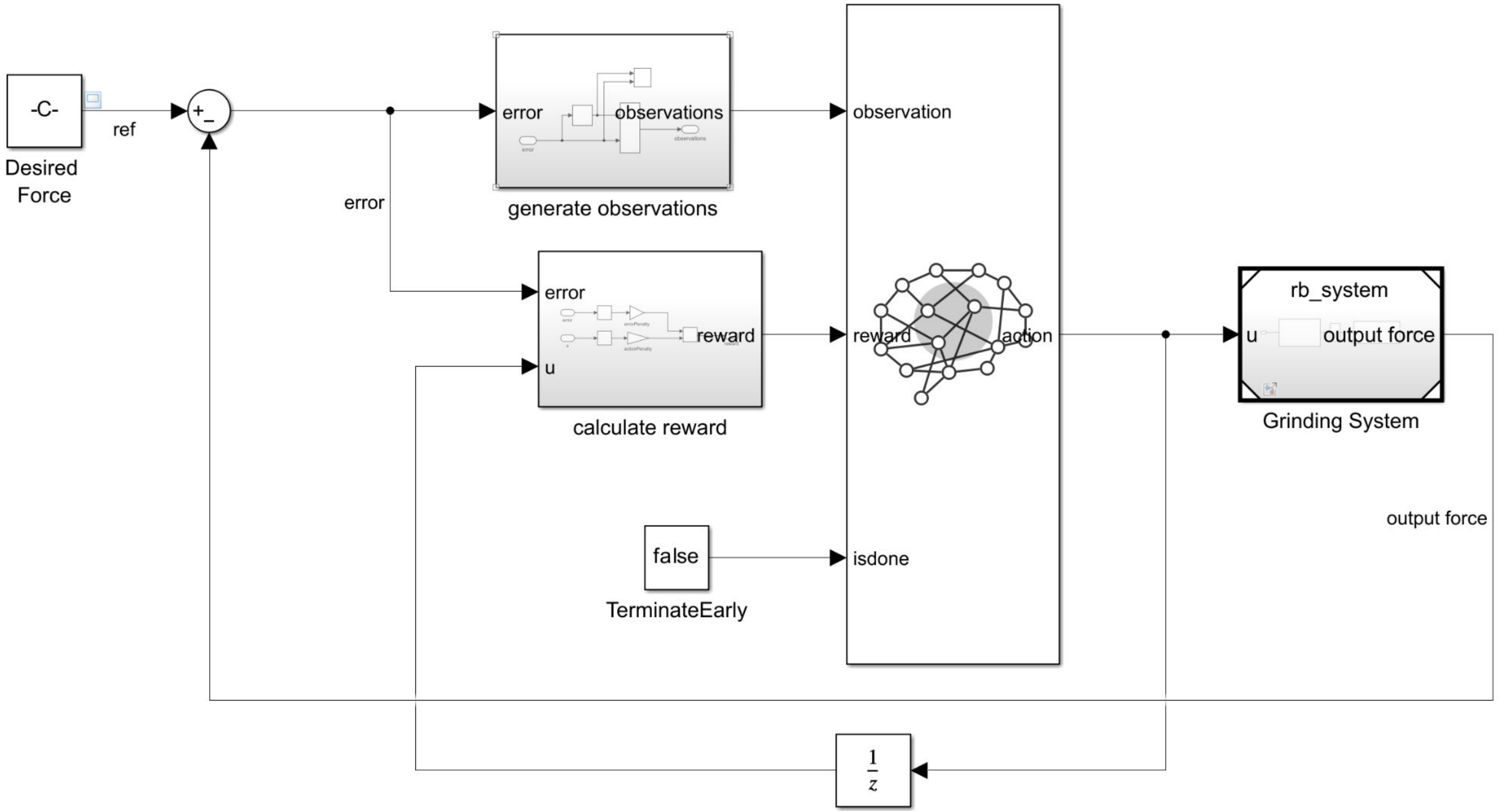
Error:

$$e(t) = F_{\text{desired}} - F(t)$$

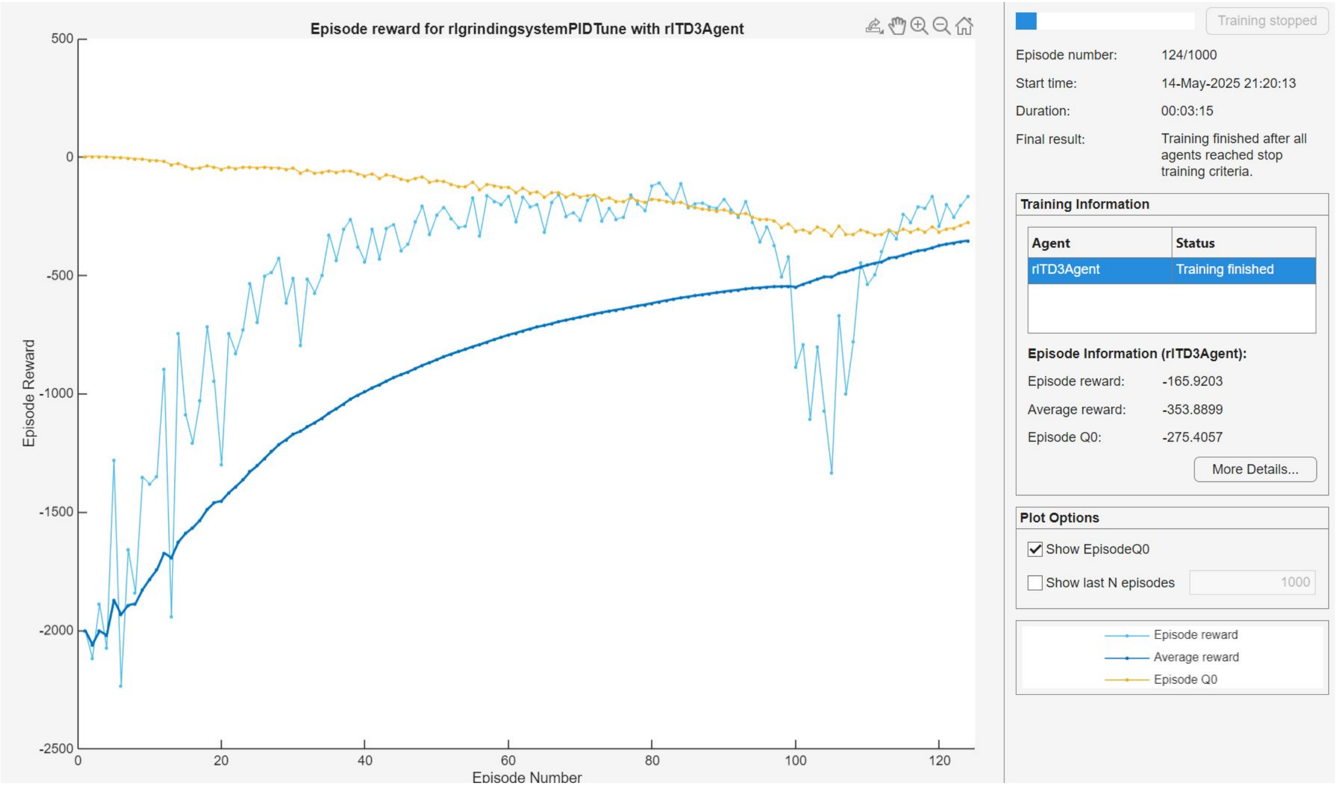
Reward function:

$$\text{Reward} = -\left((F_{\text{desired}} - F(t))^2 + 0.01 \cdot u(t)^2\right)$$

PID tuning problem

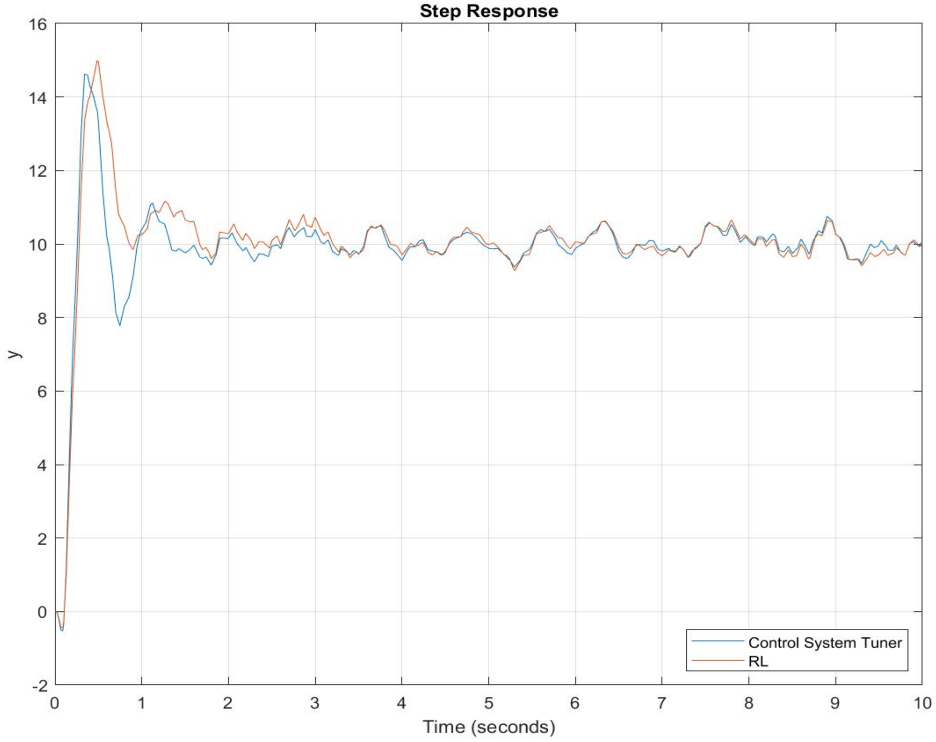


Training process



** In our problem, the Q-value is the accumulation of future rewards. A low Q-value means we are getting closer to the optimal solution.*

Comparison

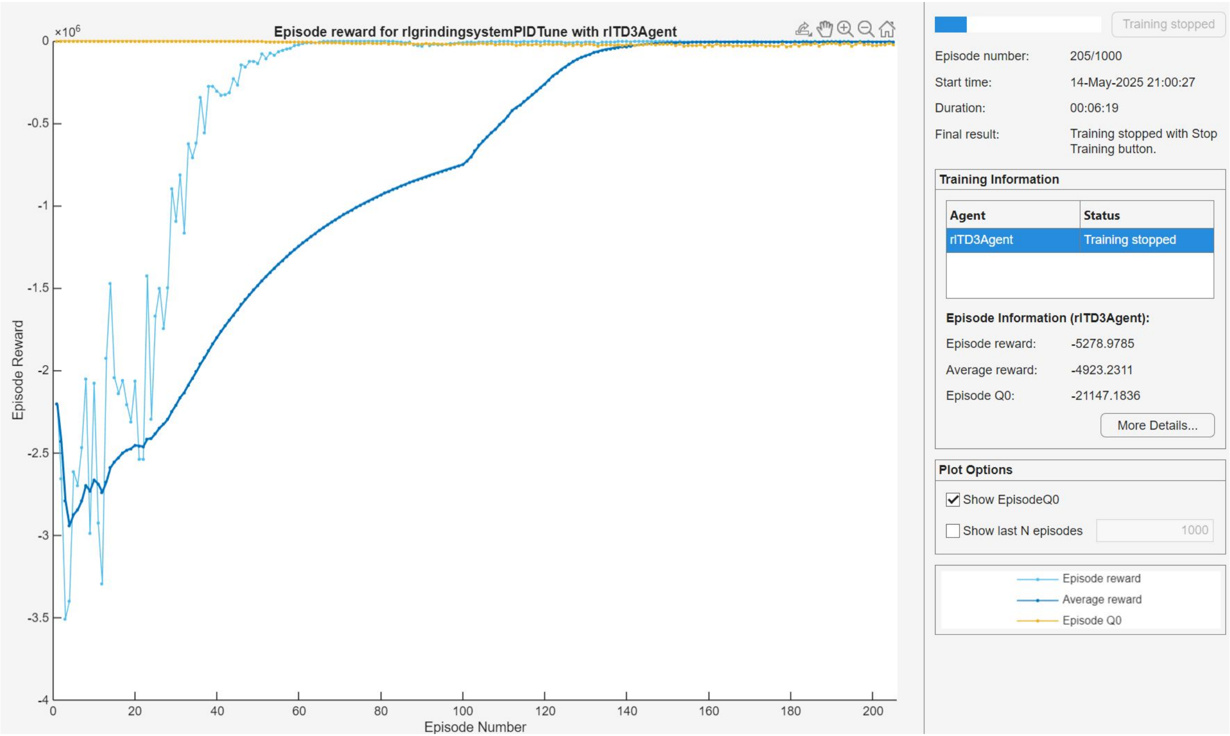


		RiseTime	SettlingTime	Overshoot	Peak
1	**CST**	0.1153	9.8330	44.9474	14.6297
2	**RL**	0.1339	9.8272	49.0196	14.9929

- Both CST and RL show comparable performance in terms of transient and steady-state behavior.
- CST has a faster rise time => quicker initial response.
- The RL agent achieves a slightly shorter settling time => marginally faster stabilization.
- The RL agent has a noticeably higher overshoot.

** desired force is step function with amplitude of 10*

Unstable training



Actors and critics depend on each other; if one is inaccurate, the other will also be affected.

Errors can accumulate over time => training crashes sometimes.

Next steps

- Increase network size for better learning capability
- Finetune training hyperparameters
    - Dynamic training steps, regularization
    - Explore-exploit balance