Temperature Control of a Simulated Nonlinear Greenhouse System with Reinforcement Learning

Aaron Rosenberg

ECE 607

12/12/2023

Definition of Reinforcement Learning (RL)

- Branch of machine learning
- "Reward-driven trial-and-error process" [1]
  - Agent interacts with environment
  - Agent's goal is to maximize expected reward
- Stochastic or Deterministic policies
- Model-based or Model-free

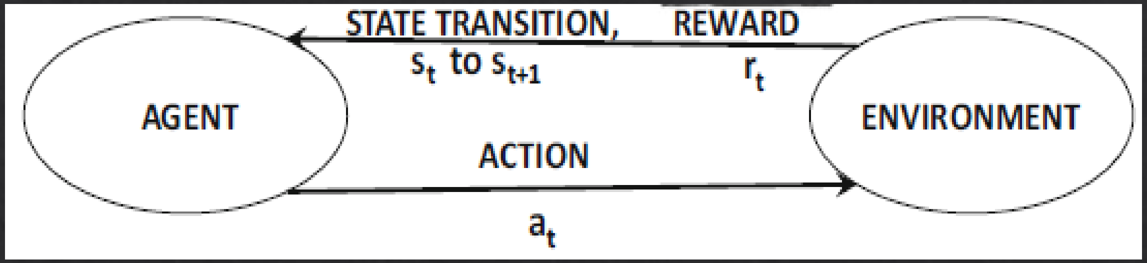


Fig. 1 RL Framework [1]

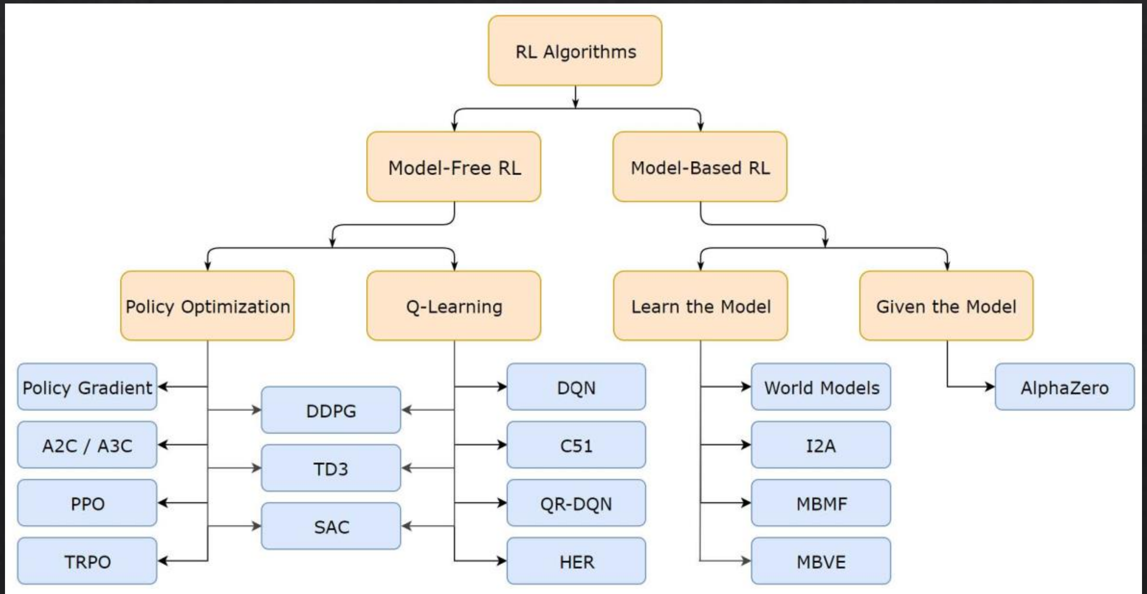


Fig. 2 Common RL Algorithms [2]

Application of RL in Control Problems

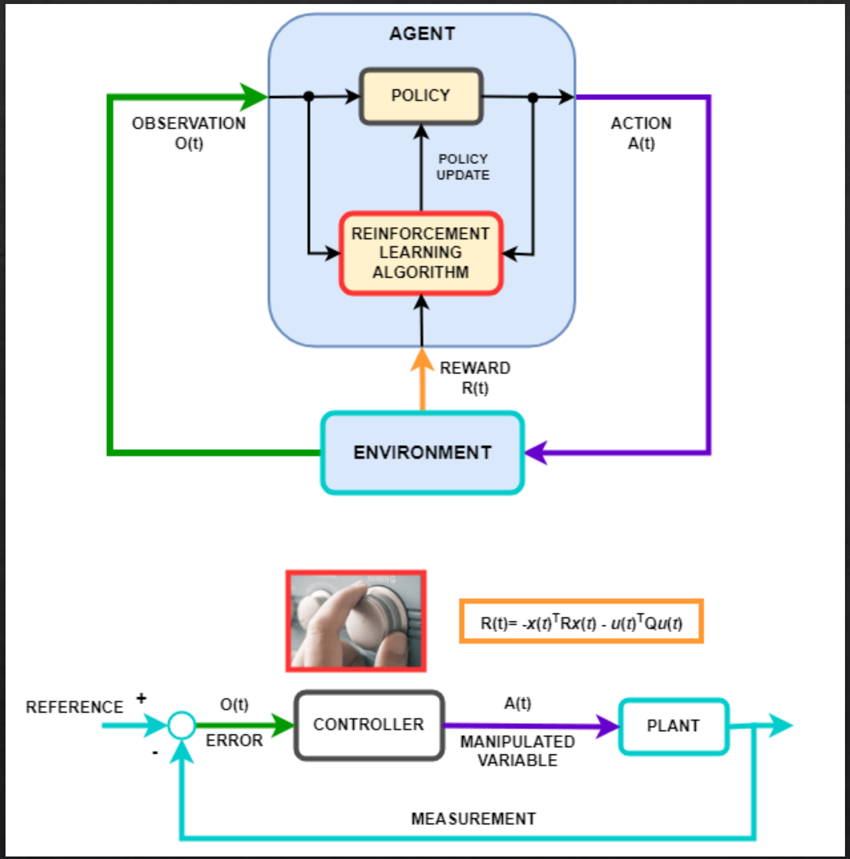


Fig. 3 Similarities between RL and Control Theory [3]

Pros and Cons of RL in Control Problems:

Pros:

- Can solve complex problems
- Can correct errors during training
- Model of system not required; algorithm can learn through interaction with environment

Cons:

- Difficult to implement on real systems
- No methods to check robustness
- Computational complexity

Greenhouse System

Goal: Maintain the inside temperature and inside humidity at desired set point

- Non-linear, coupled system
- MIMO system; 3 input, 2 output, 3 disturbance inputs

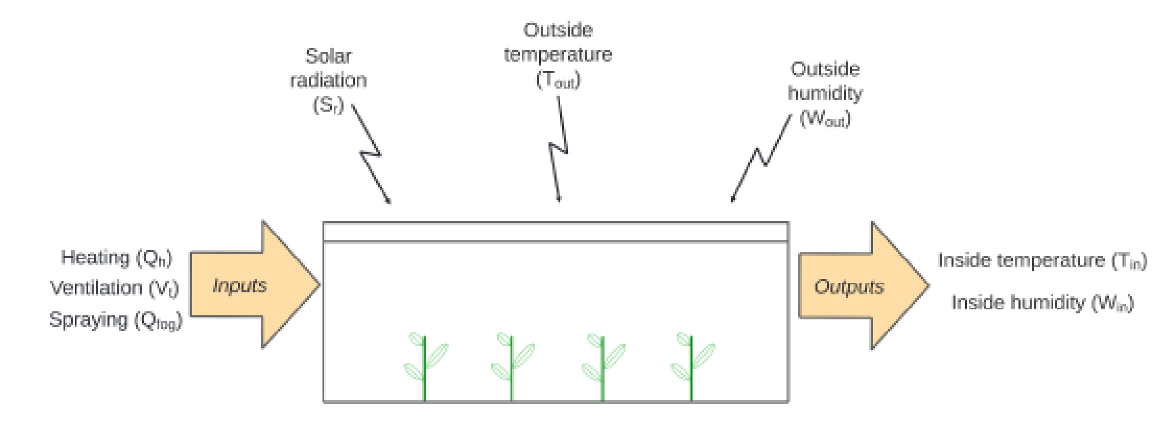


Fig. 4 Greenhouse System [4]

$$\frac{dT_{in}}{dt} = \frac{Q_h}{\rho V C_p} + \frac{S_i}{\rho V C_p} - \frac{\gamma Q_{fog}}{\rho V C_p} - (T_{in} - T_{out})[\frac{V_t}{V} + \frac{UA}{\rho V C_p}]$$

$$\frac{dW_{in}}{dt} = \frac{Q_{fog}}{\rho V} + \frac{E}{\rho V} - \frac{V_t(W_{in} - W_{out})}{\rho V}$$

Fig. 5 Greenhouse System Governing Equations [4]

Deep Deterministic Policy Gradient (DDPG)

- Appropriate for continuous action spaces
- Replay buffer
- Consists of 4 networks:
  - Actor - chooses action
  - Critic - evaluates actor's action
  - Target critic/target actor – "lag" behind main actor/critic for stability

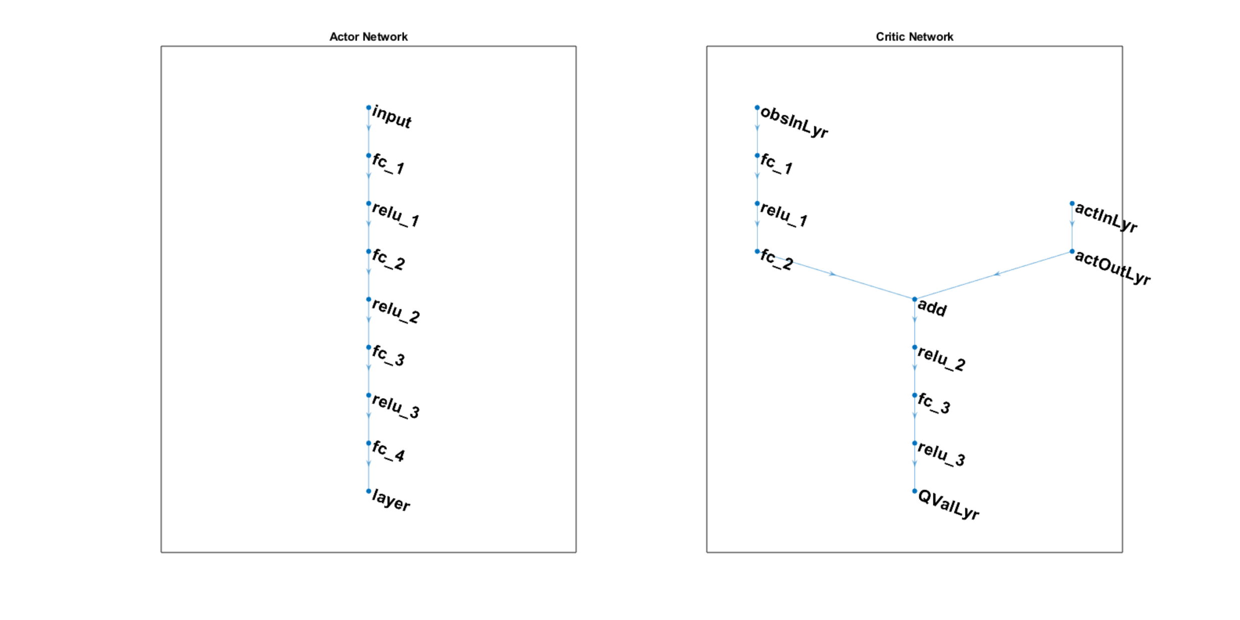


Fig. 6 Actor/Critic Network Structure

Application to Greenhouse System

Environment

- Properties
  - System State, $s = [T_{in}; T_{out}; S_r; W_{in}; W_{out}]$
  - Reference Signal (setpoints): $T_{in} = 25°C$, $W_{in} = 16g/m^3$
- Step Function = step(action)

$$\frac{dT_{in}}{dt} = \frac{Q_h}{\rho V C_p} + \frac{S_i}{\rho V C_p} - \frac{\gamma Q_{fog}}{\rho V C_p} - (T_{in} - T_{out})[\frac{V_t}{V} + \frac{UA}{\rho V C_p}]$$

$$\frac{dW_{in}}{dt} = \frac{Q_{fog}}{\rho V} + \frac{E}{\rho V} - \frac{V_t(W_{in} - W_{out})}{\rho V}$$

- Reset Function
  - WMO Data for Norfolk,VA
  - Solar Data for Norfolk,VA
- Reward Function, R

$Action = [Q_h; Q_{fog}; V_t]$

s

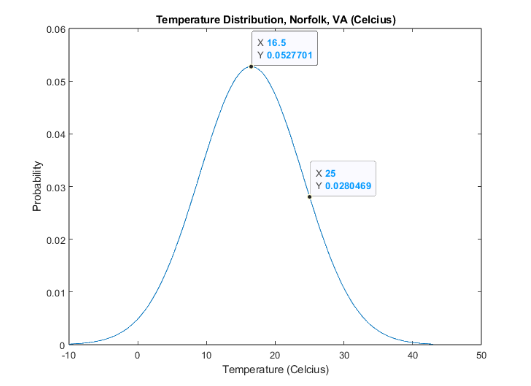
s, R

Agent

- Actor Network (Deterministic)
  - $A = \mu(s|\theta_\mu)$

Action

- Critic Network
  - $Q(s,A|\phi)$



Training and Rewards

- Several Attempts to Train
- Computationally demanding
- Reward shaping difficult:
  1. Standard control error, $e = Ref - y$
  2. Time based-penalties
  3. Issue of ignoring one input
  4. Time-based rewards

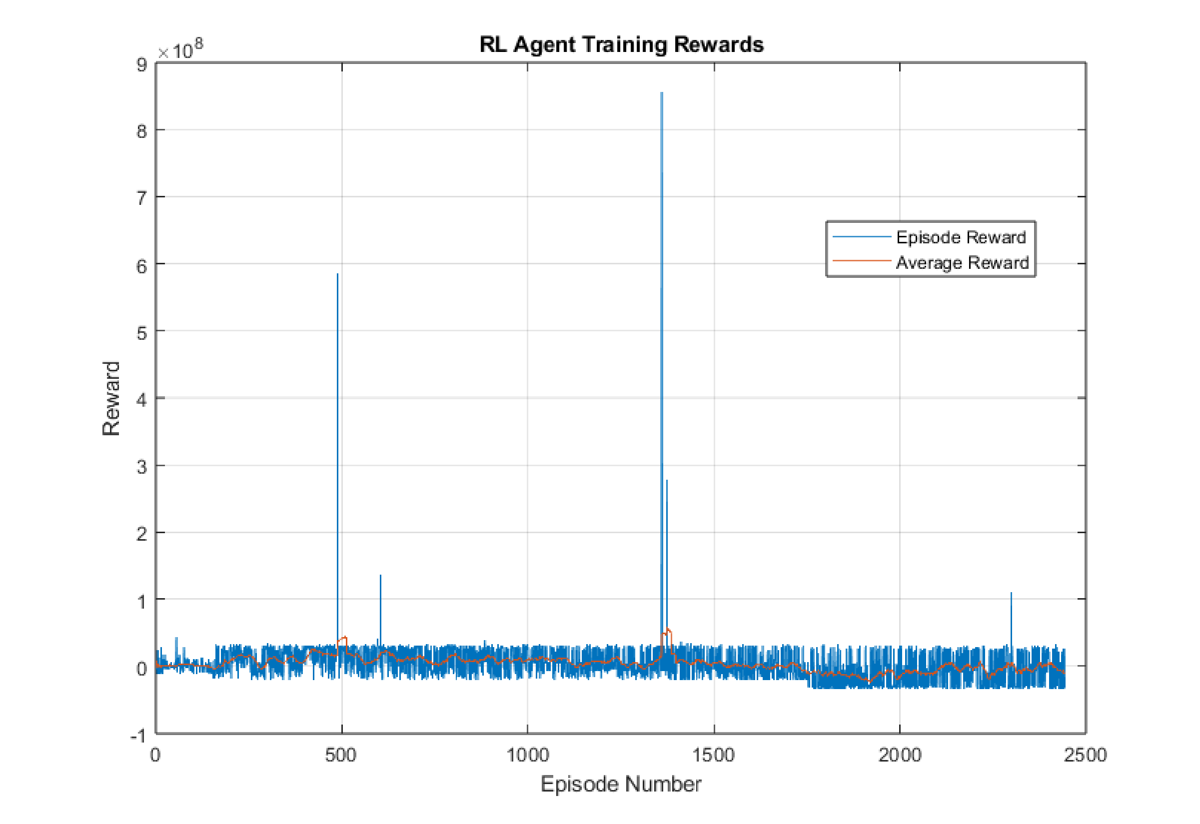


Fig. 7 Training Rewards

RL Results

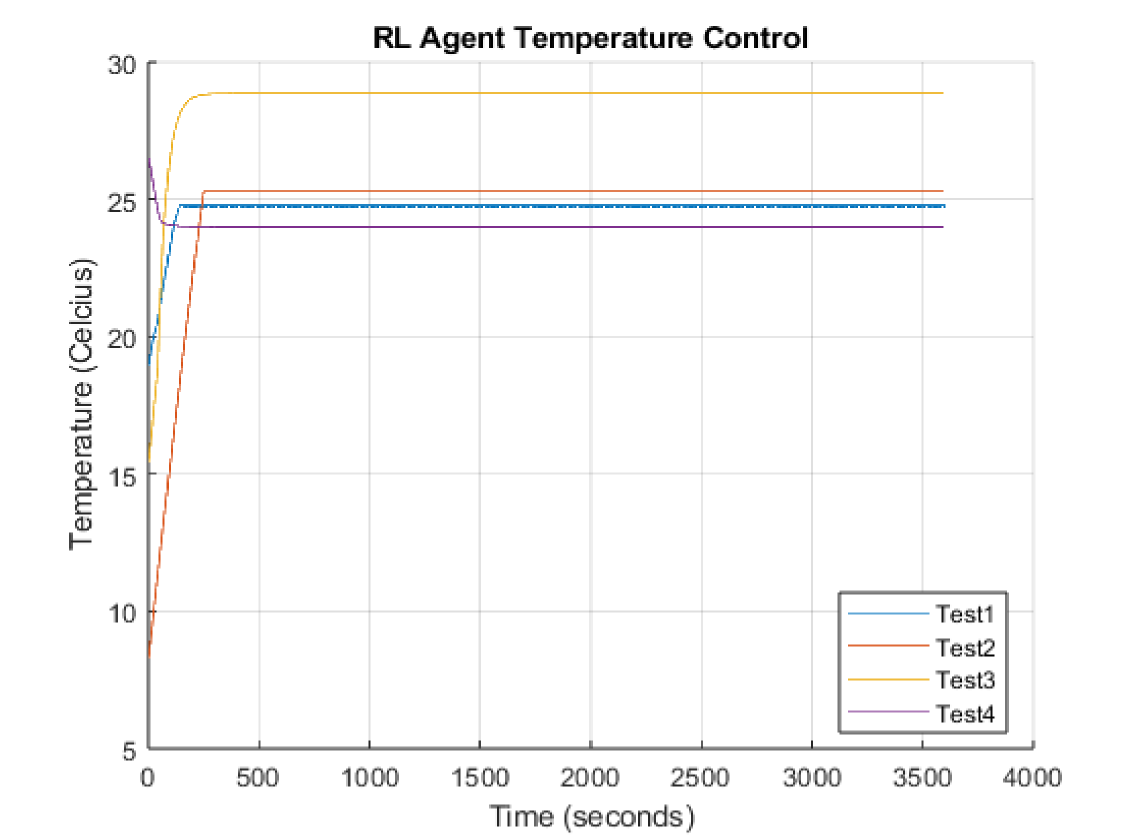


Fig. 8 Testing Results with Four Different Initial Conditions

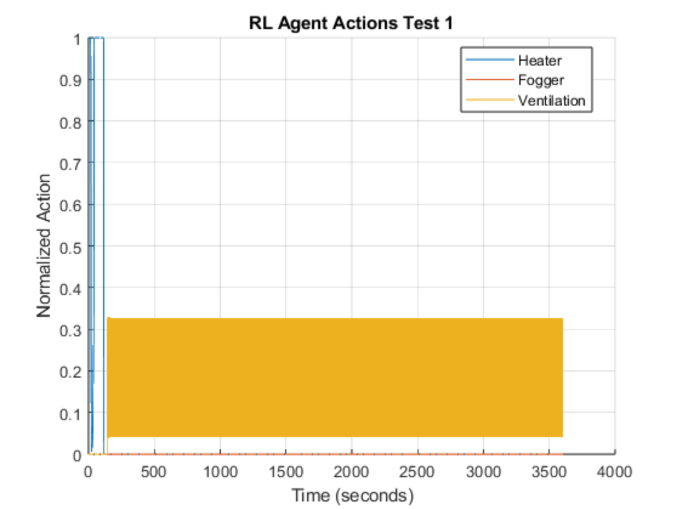


Fig. 9 Agent Actions in Test 1

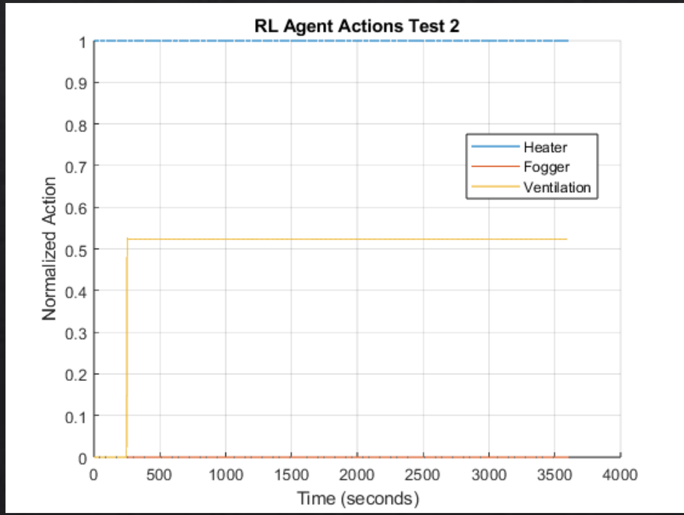


Fig. 10 Agent Actions in Test 2

Comparison of RL Solution to PID

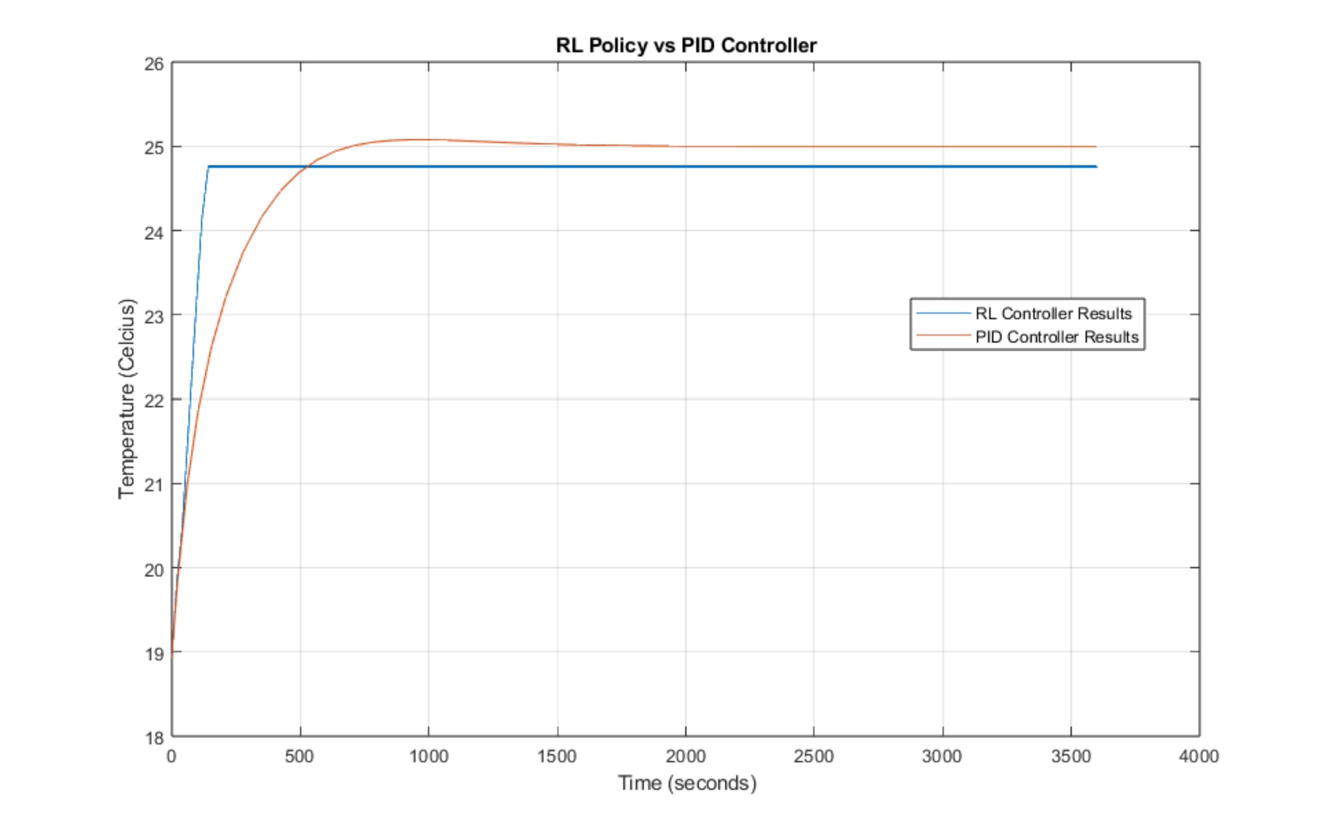


Fig. 11 Comparison of RL controller to PID

- RL controller is faster but carries steady state error
- Compared controllers using Integral Squared Error (ISE) = $\int e^2(t)dt$

Test	Initial State $[T_{in} ; T_{out} ; S_r]$	RL ISE (10^3)	PID ISE (10^3)
1	[18.93 ; 19.27 ; 143.62]	1.27	8.12
2	[8.28 ; 20.57 ; 215.91]	1.05	21.25
3	[15.43 ; 26.02 ; 253.55]	-13.00	12.42
4	[26.52 ; 3.76 ; 93.24]	3.54	0.820

Conclusion and Future Work

Conclusion:

- RL controller results in comparable results to the PID when initial $T_{in} < T_{set}$
- Improvements required, including:
  - Learn better actions when $T_{in} > T_{set}$
  - Improve steady-state error

Future Work:

- Ideas to implement improvements:
  - More training episodes
  - Change state to include derivatives
  - Continue adjusting reward function
  - Separate agents trained for specific ranges of initial conditions
  - Multi-agent RL (MARL); separate agents for each action

References

[1] C. C. Aggarwal, Neural Networks and Deep Learning: A Textbook. New York: Springer, 2018.

[2] D. Bates, "Virtual Reinforcement Learning for Balancing an Inverted Pendulum in Real Time.," Dissertation, North Carolina State University, 2021.

[3] "Reinforcement Learning for Control Systems Applications," Reinforcement learning for control systems applicationsMATLAB&Simulink,https://www.mathworks.com/help/reinforcement-learning/ug/reinforcement-learning-for-control-systems-applications.html (accessed Sep. 15, 2023).

[4] G. Cevallos, J. Pinzon, and O. Camacho, "A microclimate greenhouse multivariable control: A guide to use hardware in the Loop Simulation," 2022 IEEE International Conference on Automation/XXV Congress of the Chilean Association of Automatic Control (ICA-ACCA), 2022.

[5] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra, Continuous control with deep reinforcement learning, CoRR abs/1509.02971 (2015).