# Our recent paper on RL

LINKÖPINGS UNIVERSITET

Farnaz Adib Yaghmaie

Linkoping University, *Sweden*
*farnaz.adib.yaghmaie@liu.se*

April 6, 2021

- Farnaz Adib Yaghmaie, Fredrik Gustafsson, and Lennart Ljung, **“Linear Quadratic Control using Model-free Reinforcement Learning”**, *IEEE Transactions on Automatic Control*, 2021, conditionally accepted.

# Paper Objective

- RL algorithms assume that the state variable is exactly measurable.
- We assume that noisy measurements of state are available.
- Objective: to analyze DP-based RL routines when observation noise is present.

- **Dynamics:**

$$x_{k+1} = Ax_k + Bu_k + w_k$$
$$y_k = x_k + v_k,$$

- **State and action:**

$$s_t \in \mathbb{R}^n,$$
$$u_t \in \mathbb{R}^m$$

- **Cost function ($\equiv$ negative of reward):**

$$r(y_k, u_k) = r_k = y_k^T R_y y_k + u_k^T R_u u_k$$

where $R_y \geq 0$ and $R_u > 0$.

**Solvability Criterion:** Minimize $V$ using $\pi = Ky_k$

$$V(y_k, K) = \mathbf{E}[\sum_{t=k}^{+\infty}(r(y_t, Ky_t) - \lambda(K))|y_k] \tag{1}$$

where $\lambda$ is the average cost

$$\lambda(K) = \lim_{\tau \to \infty} \frac{1}{\tau}\mathbf{E}[\sum_{t=1}^{\tau} r(y_t, Ky_t)] \tag{2}$$

- $\lambda \neq 0$ when process and measurement noises appear
- $V(y_k, K)$ in (1) measures the quality of transient response
- The mentioned problem is equivalent to an LQR problem

The agents learn a quadratic $Q$ function

$$Q(y_k, a_k) = \begin{bmatrix} y_k^\dagger & a_k^\dagger \end{bmatrix} \begin{bmatrix} G_{11} & G_{12} \\ G_{12}^\dagger & G_{22} \end{bmatrix} \begin{bmatrix} y_k \\ a_k \end{bmatrix} = z^\dagger G z \qquad (3)$$

**Classical $Q$-learning:**

Greedy w.r.t. the last $Q$-function

$$\pi = K^{i+1} y_k = -(G_{22}^{i})^{-1} G_{12}^{i\dagger} y_k$$

**Average $Q$-learning:**

Greedy w.r.t. the average of all previous $Q$-function

$$\pi = K^{i+1} y_k = \sum_{j=1}^{i} -(\hat{G}_{22}^{j})^{-1} \hat{G}_{12}^{j\dagger} y_k$$

*and a few more technical differences.*

1. Compute the empirical average cost $\lambda = \frac{1}{T}\sum_{t=1}^{T} r_t$
2. Collect data
   - Observe $y_t$ and select $a_t$
   - Apply $a_t$ and observe $r_t,\ y_{t+1}$.
   - Add $y_t,\ a_t,\ r_t,\ y_{t+1}$ to the history.
3. Estimated the kernel of $Q$ by Least Squares Temporal Difference (LSTD)

$$vecs(G) = \left(\frac{1}{T}\sum_{t=1}^{T}\Psi_t(\Psi_t - \Psi_{t+1})^{\dagger}\right)^{-1}\left(\frac{1}{T}\sum_{t=1}^{T}\Psi_t(c_t - \lambda)\right)$$

$$z = \begin{bmatrix} y \\ a \end{bmatrix},\ \Psi = [z_1^2, 2z_1z_2, ..., 2z_1z_n, z_2^2, ...,\ 2z_2z_n,\ ...,\ z_n^2]^{\dagger}.$$

4. Update the controller gain

$$K^{i+1} = \sum_{j=1}^{i} -(\hat{G}_{22}^{j})^{-1}\hat{G}_{12}^{j\dagger}$$

- **Dynamics:**

$$x_{k+1} = \begin{bmatrix} 1.01 & 0.01 & 0 \\ 0.01 & 1.01 & 0.01 \\ 0 & 0.01 & 1.01 \end{bmatrix} x_k + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} u_k + w_k,$$

$$y_k = x_k + v_k,$$

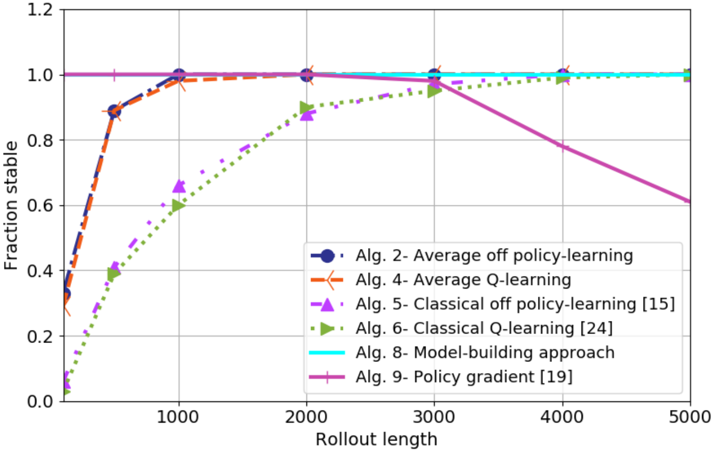
$$W_w = I, \; W_v = I.$$

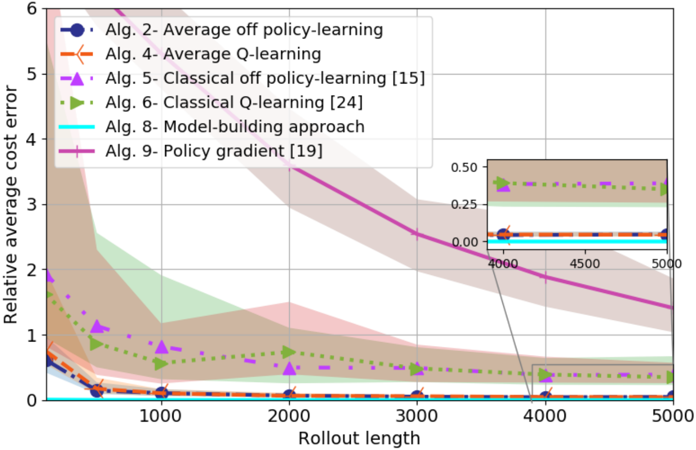
- **Cost function ($\equiv$ negative of reward):**

$$r(y_k, u_k) = 0.001 y_k^T y_k + u_k^T u_k.$$

# Algorithms to be compared

- Average off-policy learning
- Average $Q$-learning
- Classical off-policy learning
- Classical $Q$-learning
- Model-building approach
- Policy gradient
- Analytical solution


1.2
1.0
0.8
0.6
0.4
0.2
0.0
Fraction stable
1000
2000
3000
4000
5000
Rollout length
Alg. 2- Average off policy-learning
Alg. 4- Average Q-learning
Alg. 5- Classical off policy-learning [15]
Alg. 6- Classical Q-learning [24]
Alg. 8- Model-building approach
Alg. 9- Policy gradient [19]


Alg. 2- Average off policy-learning
Alg. 4- Average Q-learning
Alg. 5- Classical off policy-learning [15]
Alg. 6- Classical Q-learning [24]
Alg. 8- Model-building approach
Alg. 9- Policy gradient [19]
Relative average cost error
Rollout length
0
1
2
3
4
5
6
1000
2000
3000
4000
5000
0.50
0.25
0.00
4000
4500
5000

# Important observations

- Observation noise can deteriorate performance
- PG does not achieve good results
- Model-building approach is superb!
- Our proposed algorithms produce more stable controller gains
- Performance of $Q$ learning-types algorithms improve as the trajectory length increases.

# Email your questions to

*farnaz.adib.yaghmaie@liu.se*