Harvard John A. Paulson School of Engineering and Applied Sciences

Distributed Reinforcement Learning for Decentralized Linear Quadratic Control

A Derivative-Free Policy Optimization Approach

Yingying Li, Yujie Tang, Runyu Zhang, Na Li
Harvard John A. Paulson School of Engineering and Applied Sciences

Full paper available at
https://arxiv.org/abs/1912.09135

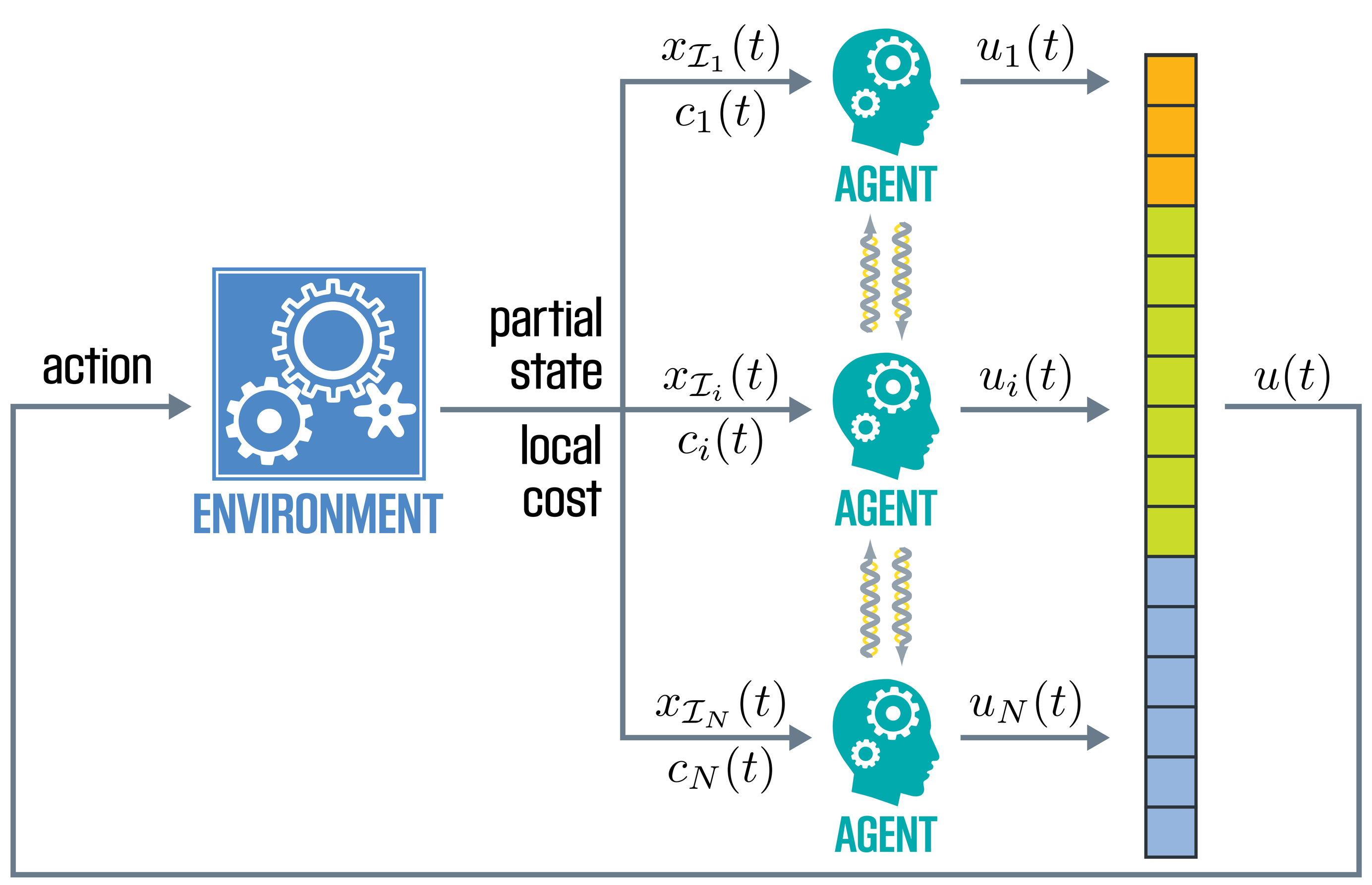
Summary

- **Problem setup**: distributed learning for decentralized LQ control
  - unknown dynamics
  - limited communication
  - local costs
  - partial state observation
- **Algorithm**: Zero-Order Distributed Policy Optimization

ZODPO
- policy gradient
- zero-order optimization
- consensus algorithms

Performance Guarantees:
- All generated policies are stabilizing w.h.p.
- Sample complexity $O(1/\epsilon^4)$

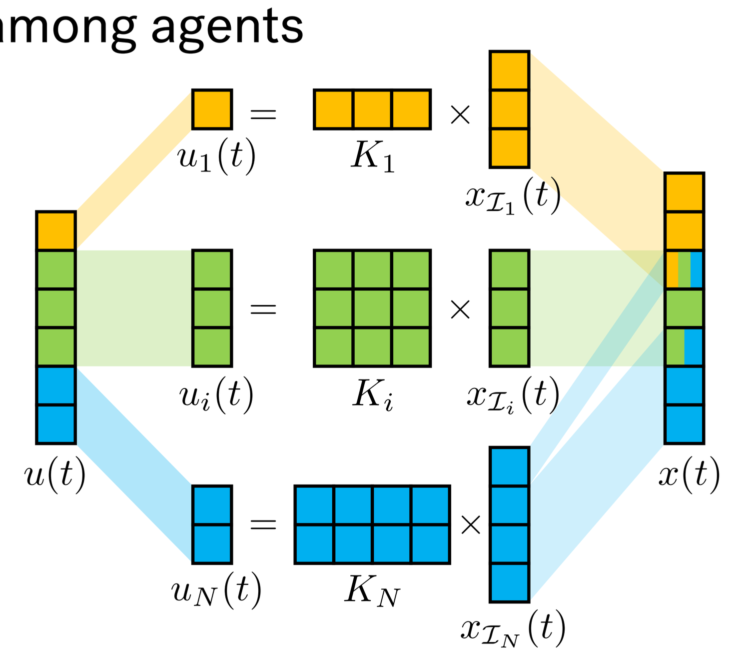
Problem Setup



- **Unknown LTI dynamics:** $x(t+1) = A\,x(t) + B\,u(t) +$ Gaussian noise
- **Partial observation:** Agent i observes a subvector $x_{\mathcal{I}_i}(t)$ of $x(t)$
- **Quadratic costs:** $c_i(t) = x(t)^\top Q_i x(t) + u(t)^\top R_i u(t)$
- **Limited communication** via a network among agents
- **Linear controllers:** $u_i(t) = K_i x_{\mathcal{I}_i}(t)$
- **Goal:** Find controllers that minimizes *infinite-horizon* average cost

$$\underset{K_1,\dots,K_N}{\text{minimize}} \quad J(K_1,\dots,K_N)$$

$$J(K_1,\dots,K_N) = \lim_{T\to\infty} \mathbb{E}\left[\frac{1}{TN}\sum_{t=1}^{T}\sum_{i=1}^{N} c_i(t)\right]$$



Algorithm

Algorithm: Zero-Order Distributed Policy Optimization (ZODPO)

Input: smoothing radius r, step size η, $\bar{J} > 0$, termination steps T_G and T_J, initial controllers $K_{1,0}, \dots, K_{N,0}$.

Initialize $K_i(1) \leftarrow K_{i,0}$.
for $s = 1, 2, \dots, T_G$ **do**
  // *Step 1: Sampling from the unit sphere*
  Each agent i generates $D_i(s) \in \mathbb{R}^{m_i \times n_i}$ by the subroutine `SampleUSphere`.
  // *Step 2: Local estimation of the global objective*
  Run `GlobalCostEst`$((K_i(s) + rD_i(s))_{i=1}^N, T_J)$, and let agent i's returned value be denoted by $\tilde{J}_i(s)$.
  // *Step 3: Local estimation of partial gradients*
  Each agent i estimates the partial gradient $\frac{\partial J}{\partial K_i}$ by

$$\hat{G}_i^r(s) = \frac{n_K}{r}\hat{J}_i(s) D_i(s), \quad \hat{J}_i(s) = \min\left\{\tilde{J}_i(s), \bar{J}\right\}.$$

  // *Step 4: Distributed policy gradient on local controllers*
  Each agent i updates $K_i(s+1)$ by

$$K_i(s+1) = K_i(s) - \eta \hat{G}_i^r(s).$$

end

Subroutine `SampleUSphere`:
Each agent i samples $V_i \in \mathbb{R}^{n_i \times m_i}$ with i.i.d. entries from $\mathcal{N}(0,1)$, and lets $q_i(0) = \|V_i\|_F^2$.
for $t = 1, 2, \dots, T_S$ **do**
  Agent i sends $q_i(t-1)$ to its neighbors and updates

$$q_i(t) = \sum_{j=1}^{N} W_{ij} q_j(t-1).$$

end
return $D_i := V_i / \sqrt{N q_i(T_S)}$ *to agent i for each* $i = 1, \dots, N$.

Subroutine `GlobalCostEst`$((K_i)_{i=1}^N, T_J)$:
Reset the state of the linear dynamical system to $x(0) = 0$.
Each agent i implements K_i, and set $\mu_i(0) \leftarrow 0$.
for $t = 1, 2, \dots, T_J$ **do**
  Each agent i sends $\mu_i(t-1)$ to its neighbors, observes $c_i(t)$ and updates $\mu_i(t)$ by

$$\mu_i(t) = \frac{t-1}{t}\sum_{j=1}^{N} W_{ij}\mu_j(t-1) + \frac{1}{t}c_i(t).$$

end
return $\mu_i(T_J)$ to agent i for each $i = 1, \dots, N$.

Core ingredients:

- **Policy gradient** (Step 4)

$$K_i(s+1) = K_i(s) - \eta\,\hat{G}_i(s) \qquad \hat{G}_i(s) = \text{estimate of } \frac{\partial J}{\partial K_i} \text{ at } (K_1(s),\dots,K_N(s))$$

- **Zero-order gradient estimator** (Step 3)

$$\mathsf{G}^r(\mathsf{K},\mathsf{D}) = \frac{n_K}{r} J(\mathsf{K} + r\mathsf{D})\mathsf{D}$$

smoothing radius | uniformly sampled from the unit sphere | problem dimension

$$\|\mathbb{E}_{\mathsf{D}}[\mathsf{G}^r(\mathsf{K},\mathsf{D})] - \nabla J(\mathsf{K})\| = O(r)$$

- **Consensus algorithms** (Steps 1 & 2, `SampleUSphere` & `GlobalCostEst`)

$$y_i(t+1) = F_t\left(\textstyle\sum_{j=1}^{N} W_{ij}\, y_j(t), \text{new info}_i(t)\right)$$

W: communication matrix doubly stochastic

Performance

THEOREM

Given sufficiently small $\epsilon > 0$, suppose the parameters of ZODPO satisfy

$$r = O(\sqrt{\epsilon}) \qquad \eta = O\left(\frac{\epsilon^2}{n_K^2}\right) \qquad \bar{J} \geq 50 J_0$$

$$T_J = \Omega\left(\frac{n_K}{\epsilon}\max\left\{\theta_0, \frac{N}{1-\rho_W}\right\}\right) \qquad T_S = \Omega\left(\ln\frac{1}{\epsilon}\right) \qquad T_G = \Theta\left(\frac{n_K^2}{\epsilon^3}\right)$$

Then with high probability, all the controllers generated by ZODPO are stabilizing, and

$$\mathbb{E}\left[\frac{1}{T_G}\sum_{s=1}^{T_G}\|\nabla J(K_1(s),\dots,K_N(s))\|^2\right] \leq \epsilon$$

J_0: initial objective value; θ_0: constant determined by the system and initial controller; $\rho_W = \|W - N^{-1}\mathbb{1}\mathbb{1}^\top\|$

- **Sample complexity:** $T_G T_J = O\left(\frac{n_K^3}{\epsilon^4}\max\left\{\theta_0, \frac{N}{1-\rho_W}\right\}\right)$
  - Scalability: sample complexity is polynomial in
    ϵ^{-1}: inverse of error tolerance
    n_K: number of controller parameters
    N: number of agents
  - Impact of network structure: $O\left(\frac{N}{1-\rho_W}\right)$
- **Numerical example: multi-zone HVAC control**