Shared Linear Quadratic Regulation Control: A Reinforcement Learning Approach*

Murad Abu-Khalaf[1], Sertac Karaman[2], Daniela Rus[1]
[1]CSAIL, MIT; [2]LIDS, MIT

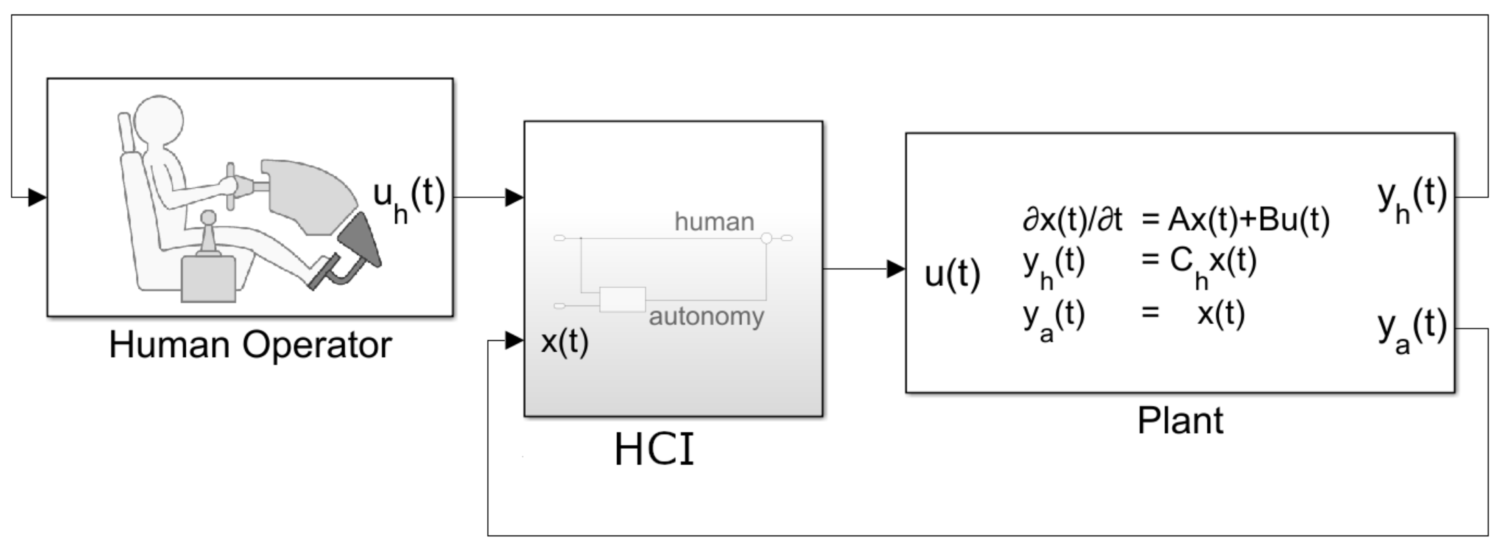
*We acknowledge the funding provided by the Toyota Research Institute (TRI)

MIT CSAIL

Objective

The parallel autonomy system **learns optimal policies** to assist a human operator in regulating a process – from continuous improvements with minimal interventions, to taking over full-control when necessary.

Parallel Autonomy



System Dynamics

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t),\\ y_a &= x,\\ y_h &= C_h x,\\ u &= u_h + u_a,\end{aligned}$$

where

- $x \in \mathbb{R}^{n\times 1}$: state
- $y_a \in \mathbb{R}^{n\times 1}$: output accessed by *autonomy*
- $y_h \in \mathbb{R}^{p\times 1}$: output accessed by *human*
- $u \in \mathbb{R}^{m\times 1}$: input of the plant
- u_h : human generated control
- u_a : autonomy computed control

and

- $A \in \mathbb{R}^{n\times n}$: internal dynamics matrix
- $B \in \mathbb{R}^{n\times m}$: input matrix
- $C_h \in \mathbb{R}^{p\times n}$: human observation matrix

Reinforcement Learning

❑ On-Policy

$$V_i(x(t_k)) = \int_{t_k}^{t_k+\tau} (x^\top(t)Qx(t) + u_i^\top(t)Ru_i(t))\,dt + V_i(x(t_k+\tau)).$$

$$u_{i+1} = -\tfrac{1}{2}R^{-1}B^\top \tfrac{dV_i}{dx}$$

→

$$x(t_k+\tau)^\top P_i x(t_k+\tau) - x(t_k)^\top P_i x(t_k) = -\int_{t_k}^{t_k+\tau} (x^\top(t)Qx(t) + u_i^\top(t)Ru_i(t))\,dt$$

→

$$\begin{aligned}P_i(A+BK_i) + (A+BK_i)^\top P_i + K_i^\top RK_i + Q &= 0,\\ K_{i+1} &= -R^{-1}B^\top P_i,\\ PA + A^\top P - P^\top B^\top R^{-1}B^\top P + Q &= 0,\end{aligned}$$

❑ Off-Policy

$$\begin{aligned}\dot{V}_i &= \tfrac{dV_i}{dx}^\top (Ax+Bu)\\ &= \tfrac{dV_i}{dx}^\top (Ax+Bu_i) + \tfrac{dV_i}{dx}^\top B\Delta(u,u_i)\\ &= -x^\top Qx - u_i^\top Ru_i + \tfrac{dV_i}{dx}^\top B\Delta(u,u_i)\end{aligned}$$

is integrated over $\varphi(t, x_0, u(t))$ →

$$V_i(x(t_k+\tau)) - V_i(x(t_k)) - \int_{t_k}^{t_k+\tau} \tfrac{dV_i}{dx}^\top B\Delta(u,u_i)dt = -\int_{t_k}^{t_k+\tau} (x^\top Qx + u_i^\top Ru_i)\,dt.$$

$$u_{i+1} = -\tfrac{1}{2}R^{-1}B^\top \tfrac{dV_i}{dx}$$

$$\Delta(u,u_i) = u - u_i$$

Assumptions

- ❑ Human policy is linear and is given by
  $$u_h(x) = K_h y_h(x) = K_h C_h x$$
- ❑ The matrices *A*, K_h, and C_h are unknown to the autonomy system.
- ❑ Input matrix *B* is known to the autonomy system.
- ❑ The autonomy system can measure $u_h(x)$.

Optimal Control Formulation

- ❑ **Problem 1** (*Minimum Intervention*): Solve the infinite-horizon optimal control problem
  $$J(x_0, t_0, u_h) = \inf_{u_a} \int_{t_0}^{\infty} (x^\top Qx + u_h^\top(x)Mu_h(x) + u_a^\top Ru_a)\,dt$$
- ❑ **Problem 2** (*Take Over*): Solve Problem 1 with $u_h = 0$.

Human-in-the-Loop Reinforcement Learning

In this case, we have $\dot{x} = A_h + Bu_a(x)$ where $A_h = A + BK_hC_h$.

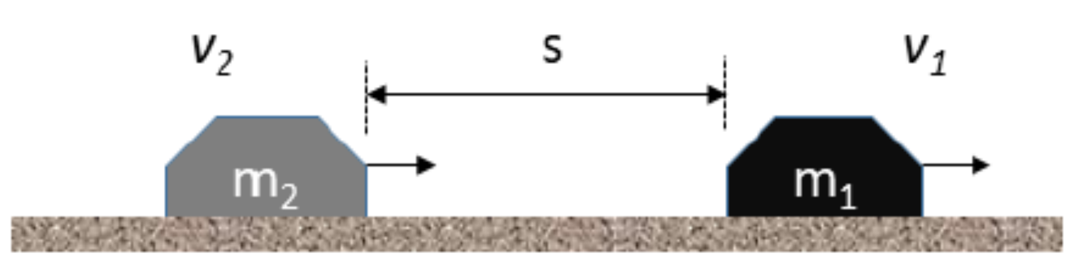
❑ Minimum Intervention:

- On-Policy: Let $u_a(x) = u_i(x)$ then Iterate on $u_i(x)$ by letting $u_0(x) = 0$.
- Off-Policy: Let $u_a(x) = 0$. The off-policy is u_a and thus $\Delta(u_a, u_i)$.
  We iterate on $u_i(x)$ with $u_0(x) = 0$.

❑ Take Over:

- Off-Policy: Let $u_a(x) = 0$. The off-policy is $u_h + u_a$ and thus $\Delta(u_h + u_a, u_i)$.
  We iterate on $u_i(x)$ with $u_0 = u_h$.

Application to Car Following



Car Following.

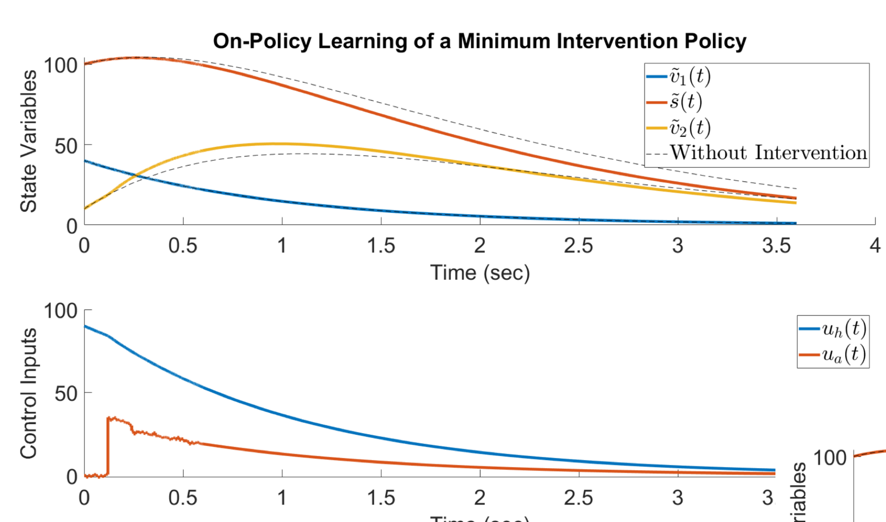
The error dynamics are

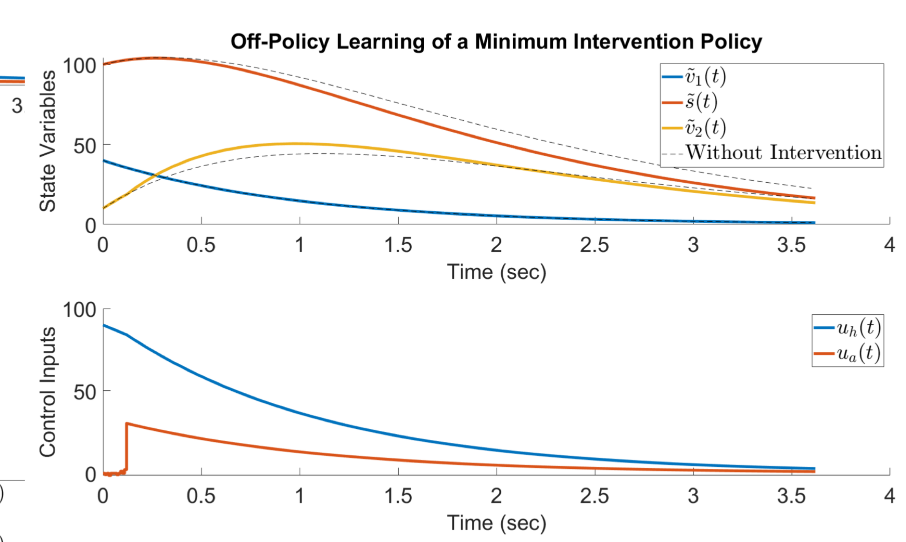
$$\begin{aligned}\dot{x}_1(t) &= -\frac{\alpha_1}{m_1}x_1(t),\\ \dot{x}_2(t) &= x_1(t) - x_3(t),\\ \dot{x}_3(t) &= -\frac{\alpha_2}{m_2}x_3(t) + \frac{1}{m_2}u,\end{aligned}$$

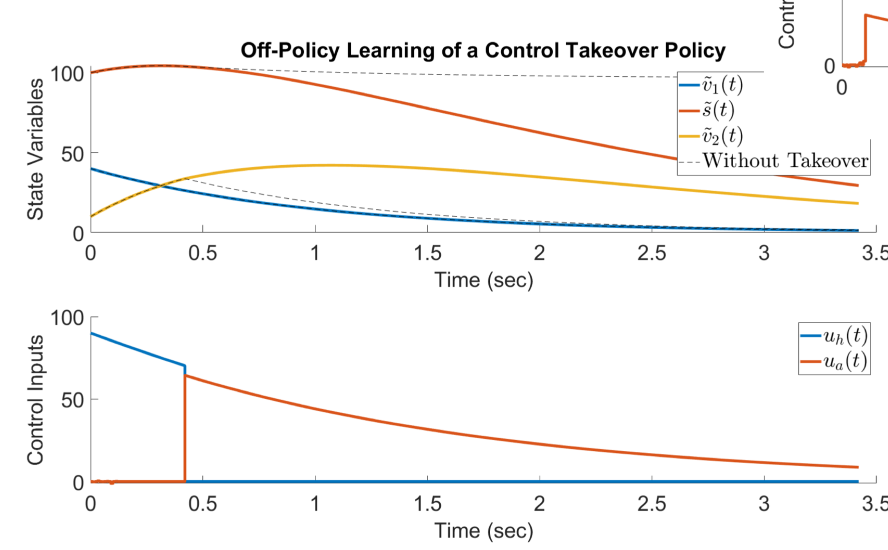
where $x_1(t) = \tilde{v}_1(t)$, $x_2(t) = \tilde{s}(t)$, $x_3(t) = \tilde{v}_3(t)$ and $u(t) = \tilde{f}_2(t)$. Moreover, $\tilde{v}_1$, $\tilde{v}_2$ are the speed error variables and $\tilde{s}$ is the spacing error variable and $\tilde{f}_2(t)$ is the force error applied to the following car. Let $m_1 = m_2 = 1$ and $\alpha_1 = \alpha_2 = 1$.

Human operator partially observes the state:

$$C_h = \begin{bmatrix} 0 & 0 \\ 0 & I_2 \end{bmatrix} \quad K_h = [0\ 1\ -1]$$







Main Analysis Results

- ❑ We avoid learning along a **single state-space trajectory** which we show leads to **data collinearity** under certain conditions such as algebraic multiplicity of eigenvalues.
- ❑ We show that exploring a minimum number of **pairwise distinct state-space trajectories** is necessary to avoid collinearity in the learning data.
- ❑ We make a clear separation between **exploitation** of learned policies and **exploration** of the state-space, and propose an exploration scheme that requires **switching to new state-space trajectories** rather than injecting noise continuously while evaluating the cost-to-go. This **avoidance of continuous noise injection minimizes interference with human action**, and avoids bias in the convergence to the stabilizing solution of the underlying algebraic Riccati equation.
- ❑ We show conditions under which existence and uniqueness of solutions can be established for off-policy reinforcement learning in continuous-time linear systems; namely a **required knowledge of the input matrix *B***.

More details at: https://arxiv.org/abs/1905.11524

Presented at Learning for Dynamics and Control (L4DC) Conference at MIT; May 30-31, 2019