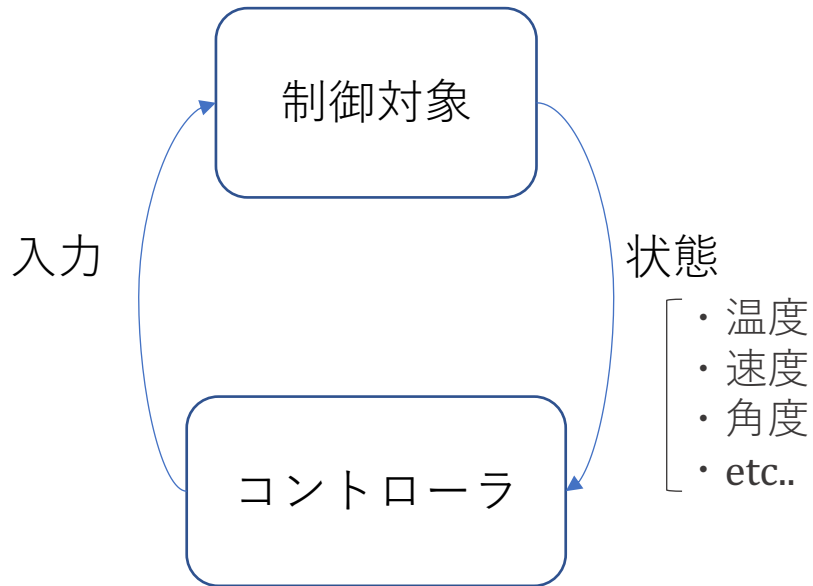


セルフトリガー制御に対する 深層強化学習

数理工学専攻 制御システム論分野
竹内 維吹

フィードバック制御と連続時間システム

- フィードバック制御
 - 制御対象の情報を用いて制御入力を設計

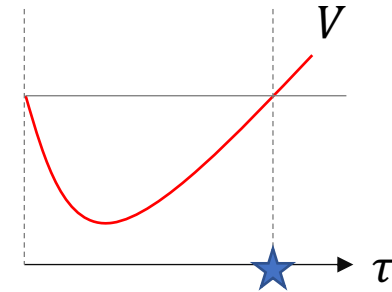


- 連続時間システム
 - 常微分方程式で記述
 - 実用上(デジタル制御)は離散化が必要
- 離散化の手法
 - サンプル値制御(従来法)
 - 一定時間間隔
 - 非効率な通信になる可能性も
 - セルフトリガー制御
 - 臨機応変な通信間隔
 - 制御器自身で意思決定

セルフトリガー制御における先行研究

- 1ステップの最適化による手法
 - [1]は連続時間システムに対して

$$\begin{aligned} \min_u & u^T u \\ \text{s.t. } & \mathcal{L}_f V(s) + \mathcal{L}_g V(s)u + \varepsilon V(s) \leq 0 \end{aligned}$$



の解 u を加え続けた際, 次ステップでの不安定度 V (リアプノフ関数)が大きくなならない, 最大の通信間隔を τ とする手法を提案

- 長時間制御全体の通信コストの最適性は考慮していない
 - 陽に考慮した最適化問題を考える(本研究)

[1]: G. Yang et al., "Self-triggered Control for Safety Critical Systems Using Control Barrier Functions.", In *Proc. of ACC*, 2019.

最適セルフトリガー制御問題の定式化

- 最適化問題

$$\min_{\pi} \mathbb{E}_{s \sim d_0}[V^{\pi}(s)]$$

$$s.t. V^{\pi}(s) = \int_0^{\infty} e^{-\alpha t} \mathbb{E}_w[s^T(t)Es(t) + u^T(t)Fu(t) + \beta\delta_c(t) | s(0) = s, \pi] dt$$

$$\dot{s} = f(s) + g(s)u + \dot{w}$$

- $\alpha > 0, \beta > 0, E \geq 0, F > 0$: ハイパーパラメータ
- $\pi(s)$: セルフトリガー制御則, 入力 u と通信間隔 τ を出力
- $\delta_c(t)$: $\int_0^{\infty} e^{-\alpha t} \beta \delta_c(t) dt = \sum_{t \in \mathcal{C}} e^{-\alpha t} \beta$ (\mathcal{C} は通信した時刻の集合)
(↑ 通信した時刻の $e^{-\alpha t} \beta$ が積分に足されていくイメージ)
- 本研究では, この問題を強化学習で解けるのかを検討

準備: 強化学習とは

- 目的: 全ステップの累積コストを最小化する方策 π^* を求める

$$\min_{\pi} \mathbb{E}_{s \sim d_0}[V^{\pi}(s)]$$

$$s.t. V^{\pi}(s) = \mathbb{E}_{\mathbf{w}} \left[\sum_{i=0}^{\infty} \gamma^i r(s_i, \pi(s_i)) \mid s_0 = s \right] \quad i: \text{ステップ数}$$

環境雑音

- $V^{\pi}(s)$: 全ステップ, 方策 π で制御した際の累積コスト
 - $r(s, a)$: 状態 s で行動 a をとったときのコスト
 - $\gamma \in [0, 1)$: 割引率, 小さいほど先のステップのコストを軽視
- 行動価値関数 $Q^{\pi}(s, a)$
 - 状態 s でまず自由に行動 a を行い, 次ステップから方策 π で制御したときの割引付き累積コスト

$$Q^{\pi}(s, a) = r(s, a) + \gamma \mathbb{E}_{s'}[V^{\pi}(s'(s, a))] \quad (\text{ベルマン方程式})$$

準備: 方策勾配型(深層)強化学習

- 方策 π をパラメータ θ をもつ関数(DNN等)で表現
- 評価関数 $J(\pi_\theta) = \mathbb{E}_{s \sim d_0}[V^\pi(s)]$ の θ 勾配を用いて方策を更新
- 決定的方策勾配定理[2]
 - 方策が状態 s から行動 a への関数の場合の勾配

$$\nabla_\theta J(\pi_\theta) = \int_S \rho^{\pi_\theta}(s) \nabla_\theta \pi_\theta(s) \nabla_a Q^{\pi_\theta}(s, a)|_{a=\pi_\theta(s)} ds$$
$$\rho^{\pi_\theta}(s) = \int_S \sum_{t=0}^{\infty} \gamma^t d_0(s_0) Pr(s_0 \rightarrow s, t, \pi) ds_0$$

[2]: D. Silver et al., “Deterministic Policy Gradient Algorithms”, In *Proc. of ICML*, 2014.

強化学習問題としての定式化

- 強化学習はステップ毎のコストの和に対する最適化問題
- 通信することを1つのステップとみなして, $V^\pi(s)$ を分解

$$\begin{aligned} V^\pi(s) &= \int_0^\infty e^{-\alpha t} \mathbb{E}_w[s^T(t)Es(t) + u^T(t)Fu(t) + \beta\delta_c(t) | s(0) = s, \pi] dt \\ &= \sum_{i=0}^\infty e^{-\alpha t_i} \int_0^{\tau_i} e^{-\alpha t} \mathbb{E}_w[s^T(t)Qs(t) + u_i^T Ru_i + \beta | s(0) = s_i, \pi] dt \\ &\hspace{15em} \text{コスト関数 } r(s_i, u_i, \tau_i) \end{aligned}$$

- 本問題におけるベルマン方程式

$$Q^\pi(s, u, \tau) = r(s, u, \tau) + e^{-\alpha\tau} \mathbb{E}_{s'}[Q^\pi(s'(s, u, \tau), \pi(s'(s, u, \tau)))]$$

- 次ステップ以降の価値にかかる割引率が τ によって変動する
- 決定的方策勾配定理が使えない

本研究での主結果

- 最適セルフトリガー制御問題に対する決定的方策勾配

$$\begin{aligned} & \nabla_{\theta} J(\pi_{\theta}) \\ &= \sum_{i=0}^{\infty} \int_S \cdots \int_S d_0(s_0) \Pr(s_1 | s_0, \pi_{\theta}) \cdots \Pr(s_{i+1} | s_i, \pi_{\theta}) \\ & \quad e^{-\alpha t_i} \{ \nabla_{\theta} \pi_{\theta}(s_i) \nabla_a Q^{\pi_{\theta}}(s_i, a) |_{a=\pi_{\theta}(s_i)} + \nabla_{\theta} e^{-\alpha \tau(s_i)} V^{\pi_{\theta}}(s_{i+1}) \} ds_{i+1} ds_i \cdots ds_0 \end{aligned}$$

- $\Pr(\cdot | s, \pi)$: s から π で制御した時の次ステップ状態の確率密度

本研究での主結果

- 最適セルフトリガー制御問題に対する決定的方策勾配

$$\nabla_{\theta} J(\pi_{\theta})$$

$$= \sum_{i=0}^{\infty} \int_S \cdots \int_S d_0(s_0) \Pr(s_1 | s_0, \pi_{\theta}) \cdots \Pr(s_{i+1} | s_i, \pi_{\theta})$$

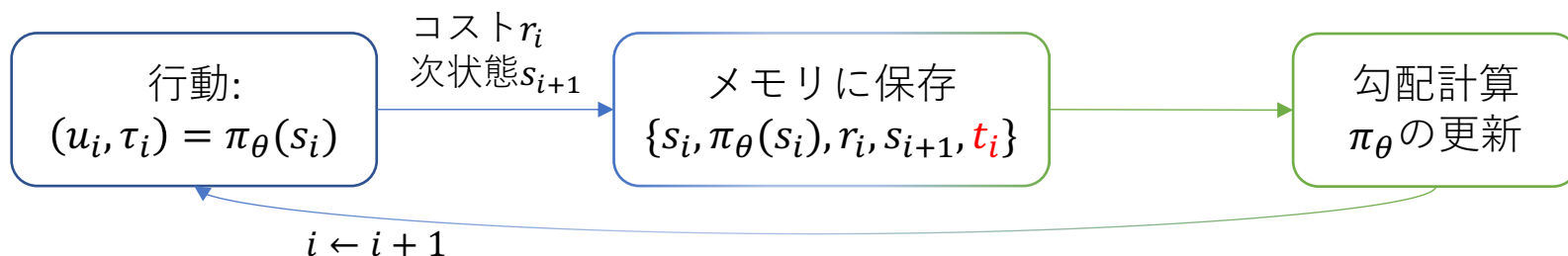
$$e^{-\alpha t_i} \{ \nabla_{\theta} \pi_{\theta}(s_i) \nabla_a Q^{\pi_{\theta}}(s_i, a) |_{a=\pi_{\theta}(s_i)} + \nabla_{\theta} e^{-\alpha \tau(s_i)} V^{\pi_{\theta}}(s_{i+1}) \} ds_{i+1} ds_i \cdots ds_0$$

- $\Pr(\cdot | s, \pi)$: s から π で制御した時の次ステップ状態の確率密度
- 右辺は, 制御パス毎に計算される $G = \sum_{i=0}^{\infty} e^{-\alpha t_i} \{ \sim \}$ の期待値
 - 無限個の初期状態 $s_0 \sim d_0$ に対し, 各 s_0 から無限本の制御パスをとり, それぞれのパスで計算した G を平均すればいい
 - 方策更新ごとにデータを取るのは非現実的

方策勾配の近似計算に対する提案手法

- データ収集と方策更新を交互に繰り返す

t : 制御時間



- メモリは π_θ で制御した時のデータのみがあることが望ましい
 - 方策の更新幅を小さくする & 古いデータを削除する
→メモリには、似た方策で制御したパスしか残らない

- メモリに保存したデータから勾配を近似計算

$G = \sum_{i=0}^{\infty} e^{-\alpha t_i} \{ \sim \}$ を表現

- 各データ組を確率 $e^{-\alpha t}$ の重み付きで N 個選んでデータセット E を作成
- データセット E の各データに対して、以下を平均

$$\nabla_{\theta} \pi_{\theta}(s) \nabla_{\alpha} Q^{\pi_{\theta}}(s, u, \tau) |_{(u, \tau) = \pi_{\theta}(s)} + \nabla_{\theta} e^{-\alpha \tau_{\theta}(s)} Q^{\pi_{\theta}}(s', \pi_{\theta}(s'))$$

計算に用いる Q 関数

- 近似方策勾配の計算には $Q^{\pi_\theta}(s, u, \tau)$ を用いる

$$\nabla_\theta \pi_\theta(s) \nabla_a Q^{\pi_\theta}(s, u, \tau)|_{(u, \tau) = \pi_\theta(s)} + \nabla_\theta e^{-\alpha \tau_\theta(s)} Q^{\pi_\theta}(s', \pi_\theta(s))$$

- 真の $Q^{\pi_\theta}(s, u, \tau)$ は未知なので, 関数 $Q(s, u, \tau | \omega)$ を用いて近似
 - データセット E に対する TD 誤差の MSE を ω に関して最小化する

$$\frac{\partial}{\partial \omega} \frac{1}{N} \sum_{(s, u, \tau) \in E} (Q(s, u, \tau | \omega) - \{r(s, u, \tau) + e^{-\alpha \tau} Q(s', \pi(s') | \omega)\})^2$$

TD 誤差: 真の Q^{π_θ} はこれを 0 にする

数値実験

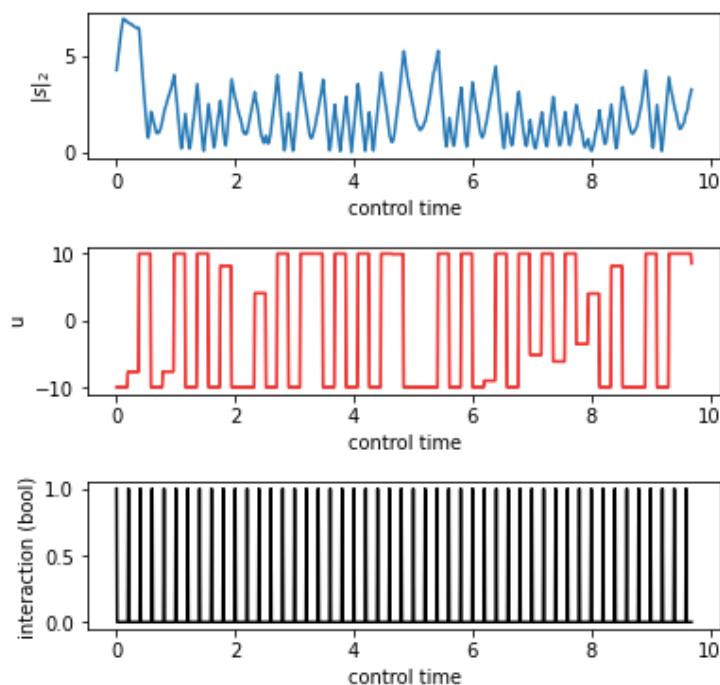
- 制御対象: 非線形システム (入力アフィンシステム)

$$\frac{d}{dt} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} s_2 \\ \frac{3g}{2l} \sin s_1 + \frac{3}{ml^2} u \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} \dot{w} \quad \dot{w}: \text{ウィーナー過程による雑音}$$

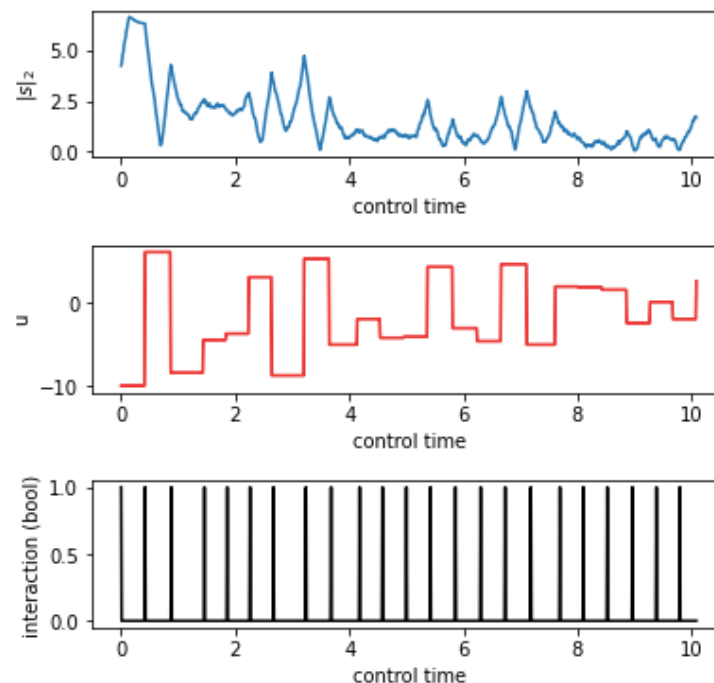
- 初期方策
 - サンプル値制御($\tau = 0.2$)
 - 各時刻の制御入力は原点付近で線形化したシステムの連続時間LQRによって設計

数値実験の結果

- 初期方策(左)と, 学習で得た方策(右)の制御性能比較
 - 初期値 $s_0 = [3., 3.]$ からの制御
 - 上から, 状態の2ノルム $|s|_2$, 各時刻の入力 u , 通信の有無を表す真偽値



$$J = 62.5$$



$$J = 30.6$$

状態変化を抑えながら, 通信回数が減少

結論

- 長時間制御全体での通信コストを考慮した最適セルフトリガー制御問題の定式化
- 定式化した問題に対する方策勾配の式の導出
 - 及び, それを用いた強化学習法の提案
- 非線形システムに対する有効性の確認
 - 線形システムに対しても同様の結果

付録A: Q 関数の近似

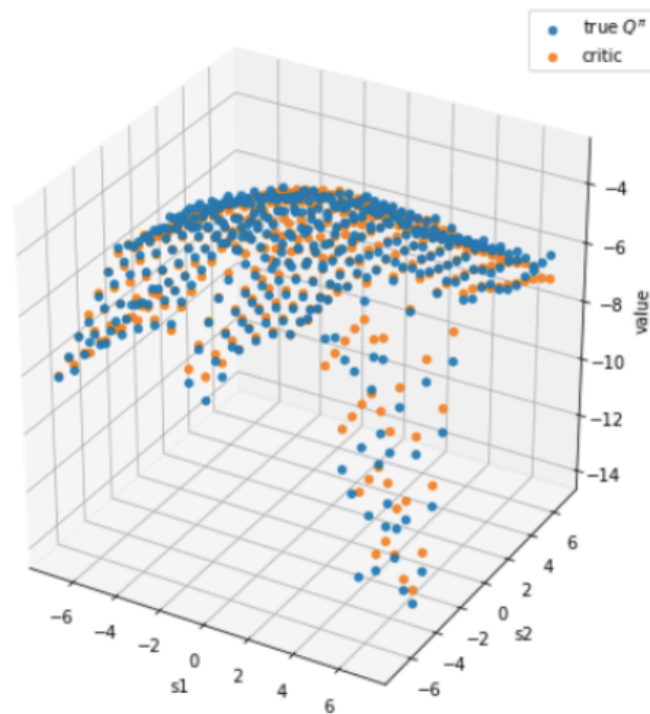
- 格子状に状態変数をとってTD学習

- 相対誤差の平均

- $V^{\pi_\theta}(s) = Q^{\pi_\theta}(s, \pi_\theta(s))$
- $V^{\pi_\theta}(s)$ と $Q(s, \pi_\theta(s)|\omega)$ を比較

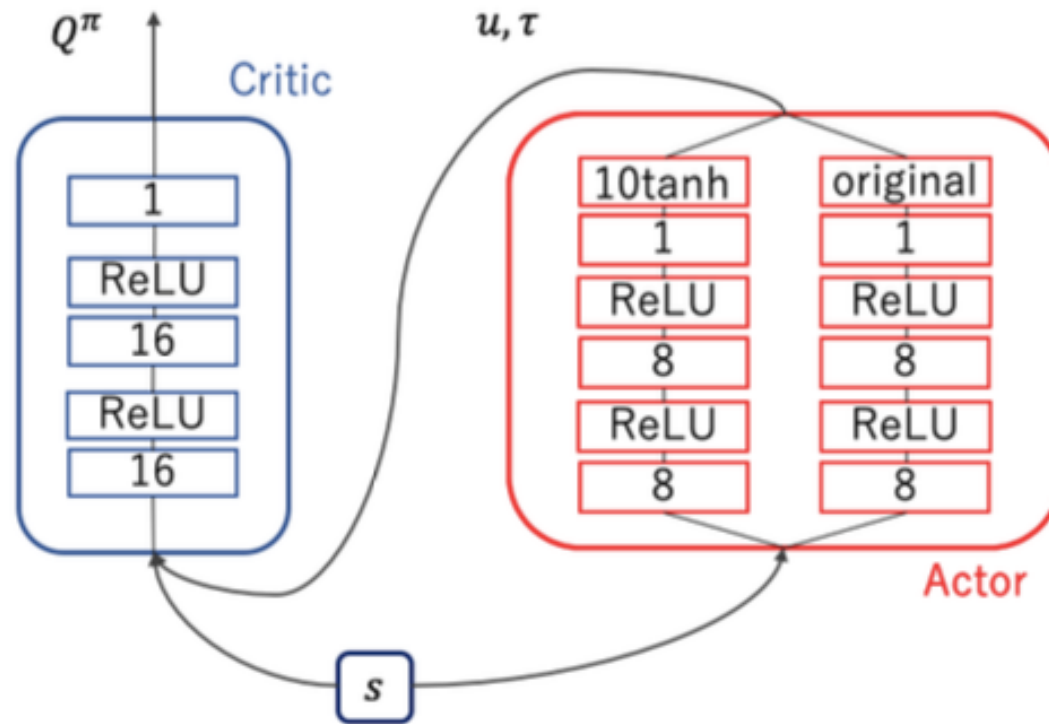
$$\frac{|V^{\pi_\theta}(s) - Q(s, \pi_\theta(s)|\omega)|}{|V^{\pi_\theta}(s)|}$$

- 平均相対誤差 = 0.03



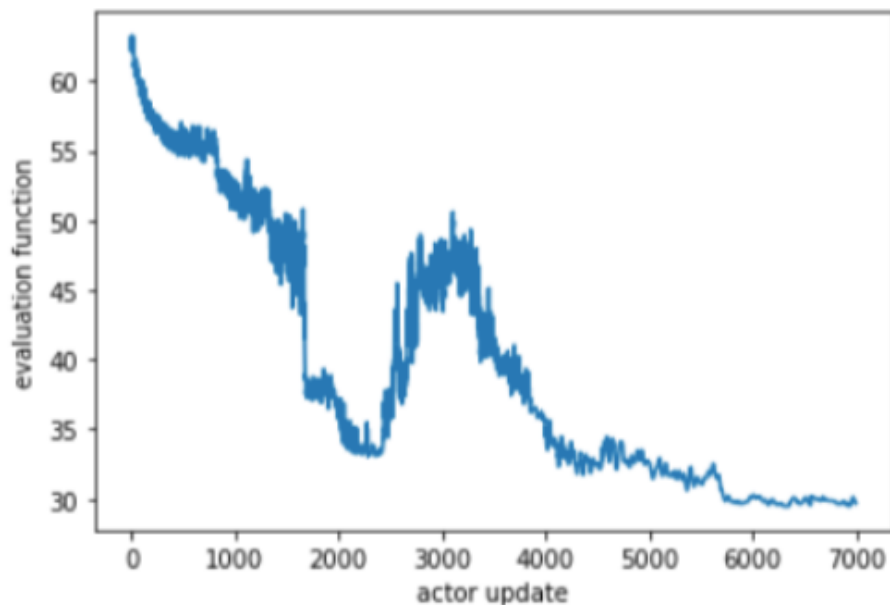
付録B: モデルの設定

- ニューラルネットワーク



付録C: 方策更新に伴う評価関数の履歴

- うまく学習ができた例



- TD学習に求められる精度は非常に高く、その近似精度が低いと、方策が悪化することよくあった

付録D: 数値実験 (線形システム)

- 制御対象

$$\dot{s} = \begin{bmatrix} -1 & 4 \\ 2 & -3 \end{bmatrix} s + \begin{bmatrix} 2 \\ 4 \end{bmatrix} u + \begin{bmatrix} 0.6 \\ 0.3 \end{bmatrix} \dot{w} \quad \dot{w}: \text{ウィーナー過程による雑音}$$

- 初期方策

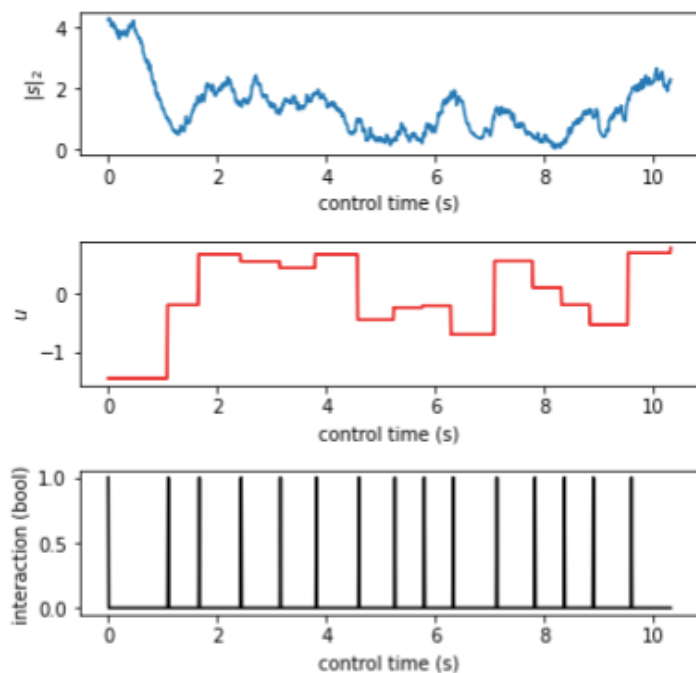
- 素朴に設計した方策

$$\pi(s) = \underset{u, \tau}{\operatorname{argmin}} \{u^2 - \lambda\tau + V_{cont}^*(s'_e, \Sigma)\} \quad \begin{aligned} s'_e &= \mathbb{E}_w[s'(s, u, \tau)] \\ \Sigma &= \operatorname{Var}[s'(s, u, \tau)] \end{aligned}$$

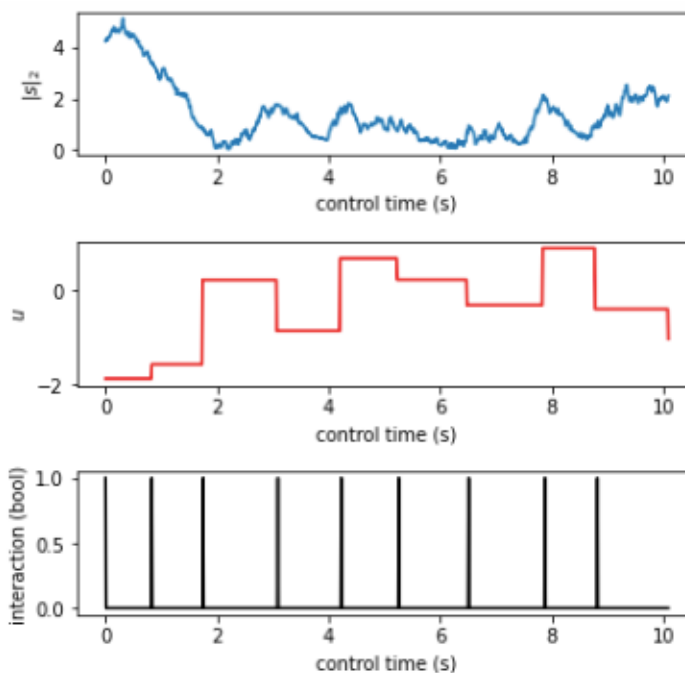
- $V_{cont}^*(s'_e, \Sigma)$ は, 連続的に最適制御した際の制御コスト (× 通信コスト)
- 次ステップで高い制御コストを必要とする状態に行かないようにしたい

付録D: 数値実験の結果 (線形システム)

- 初期方策(左)と, 学習で得た方策(右)の制御性能比較
 - 初期値 $s_0 = [3., 3.]$ からの制御
 - 上から, 状態変数の2ノルム, 各時刻の入力 u , 通信の有無を表す真偽値



$$J = 11.3$$



$$J = 6.9$$

状態変化を抑えながら, 通信回数の減少