制御システム論分野研究会資料

2020 年 8 月 20 日

M2 竹内 維吹

進捗報告 8.20

1　問題設定 (作成中)

以下のような入力アフィン系システムを考える.

$$\dot{s} = f(s) + g(s)a \tag{1}$$

ここで, s_t, a_t はそれぞれ状態ベクトル, 入力ベクトルである.

さらに, 図 1 のようなシステム (1) に対するイベント駆動型制御を考える.

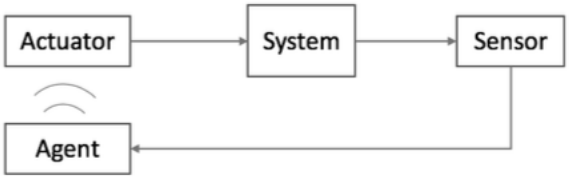

図 1　イベント駆動型制御

ここではエージェントの制御則 $\pi(s)$ を設計するとし, $\pi(s)$ はシステムに加えるべき入力信号 a と, それを送信するかどうかを決定する 2 値変数 c ($c = 1$ ならば送信) で構成されるとする. つまり

$$\pi(\cdot) = [\ a, c\ ]^\top \tag{2}$$

であるとする. このようなイベント駆動型制御に対して, 以下の条件を満たす最適イベント駆動型制御則 $\pi^*(\cdot)$ を導出する問題を考える.

$$\pi^*(\cdot) = \underset{\pi}{\mathrm{argmax}}\, \mathbb{E}_{s_0 \sim d_0}\left[V^\pi(s_0)\right] \tag{3}$$

$$V^\pi(s_0) = \sum_{t=0}^{\infty} (-s_t^\top Q s_t - \pi^\top(s_t) R \pi(s_t) - \lambda \gamma_t) \tag{4}$$

ここで, γ_t は時刻 t においてエージェントがアクチュエータと通信を行ったかを表す 2 値変数である. また, Q, R, γ はそれぞれ正定値, 半正定値, 正のハイパーパラメータであり, d_0 は初期状態を与える確率分布である. 式 (4) より, 「最小限の入力エネルギーで」かつ「最小限の通信回数で」「状態 s を 0 に素早く漸近させる」と V が最大化される.

さて, システム (1) が未知であるという設定のもとで, 実環境とのインタラクションによってデータ組 (s_t, a_t, r_t, s_{t+1}) を収集し, それらを活用することで上記の問題を解いていたのは [1] である.

本研究ではさらに一歩踏み込んで, データ収集の実環境とのインタラクションの際に状態制約

$$s \in C, C \subset S \tag{5}$$

を全時刻において満たしながら $\pi^*(\cdot)$ を求める問題を考える. ただし, 式 (3) における d_0 は $\mathrm{support}(d_0) \subset C$ を満たすとする. また, S は実環境において考えられうる全状態の集合であるとする.

(現状, 上記の問題を解くことが研究の目標であると考えています. ただ, この先どのような仮定を置くかはまだ考えられていません.)

2 DDPG と CBF を用いた解法

システム (1) に対して, 関数 $h(s)$ が以下の式を満たすならば, $h(s)$ は制御バリア関数 (CBF) と呼ばれる.

$$\sup_{a \in A} \left\{ \frac{\partial h}{\partial s}(f(s) + g(s)a) + K(h(s)) \right\} \geq 0 \tag{6}$$

ただし, $K(s)$ はクラス K 関数である.

さて, 式 (5) における状態制約 C が

$$C = \{s \in S \mid h(s) \geq 0\} \tag{7}$$

として与えられているとする. このとき $h(s)$ が制御バリア関数であるならば, 状態 $s \in C$ に対して次の時刻における状態 s' が $s' \in C$ を満たすようにする入力が存在することを保証する. そのような入力集合は現時刻での状態 s に依存し,

$$U(s) = \left\{ a \in A \mid \frac{\partial h}{\partial s}(f(s) + g(s)a) + K(h(s)) \geq 0 \right\}, \forall s \in C \tag{8}$$

としてその集合を与える.

さて, DDPG などの方策 on 型の強化学習では, 制御則 $\pi(\cdot)$ を用いて実環境とインタラクションを行い, データを収集・利活用することで, より V^π を大きくできるような制御則 $\pi'(\cdot)$ を模索する. その過程で制御入力 $\pi(s)$ が $\pi(s) \notin U(s)$ となるならば, 次時刻において状態制約 C を満たさない状態に遷移してしまう. それを避けるため, $\pi(s) \notin U(s)$ となった各時刻では, $\pi(s)$ ではなく, $\pi(s)$ に最も近い $U(s)$ の元を用いてインタラクションを行う手法を考えてみる.

3 倒立振子による実験

倒立時の振子の角度を $\theta = 0$ とし, 加えられる入力が $A = [-10\mathrm{N} \cdot \mathrm{m}, 10\mathrm{N} \cdot \mathrm{m}]$ と制限されるような倒立振子を考える. この倒立振子のダイナミクスは, 以下のように与えられる.

$$\theta_{t+1} = \theta_t + \dot{\theta}_t \delta_t + \frac{3g}{2l} \sin\theta_t \delta_t^2 + \frac{3}{ml^2} a \delta_t^2 \tag{9}$$

$$\dot{\theta}_{t+1} = \dot{\theta}_t + \frac{3g}{2l} \sin\theta_t \delta_t + \frac{3}{ml^2} a \delta_t \tag{10}$$

これは式 (1) に対応する. 本実験ではこのダイナミクスが既知であるとして $U(s)$ を構築し, 状態制約 (5) を満たしながら π^* を求めることができるのかを検証する. ただし, δ_t は離散化定数であり $\delta_t = 0.005$ とする.

ただし, これは離散化された状態方程式であるため, CBF による状態制約の前進不変性をより厳密に議論するために対応する連続時間システムを書き下すと

$$\frac{\mathrm{d}}{\mathrm{dt}} \begin{pmatrix} \theta \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \dot{\theta} \\ \frac{3g}{2l} \sin\theta + \frac{3}{ml^2} a \end{pmatrix} \tag{11}$$

となる. 与えられた関数 $h(s)$ が制御バリア関数か否かを調べるには, 式 (11) と式 (6) を用いる.

さて, $s = [\ \theta, \dot{\theta}\ ]^\top$ とすると $S = \{s \mid \theta \in [-\pi, \pi], \dot{\theta} \in \mathbb{R}\}$ である. ここで, 状態制約集合 $C \in S$ を

$$C = \{s \in S \mid h(s) \geq 0\} \tag{12}$$

とし, $h(s) = (1 - \theta^2 - \alpha\dot{\theta}^2), \alpha > 0$ とおく. すると, 状態制約集合 C は以下のように分布する.

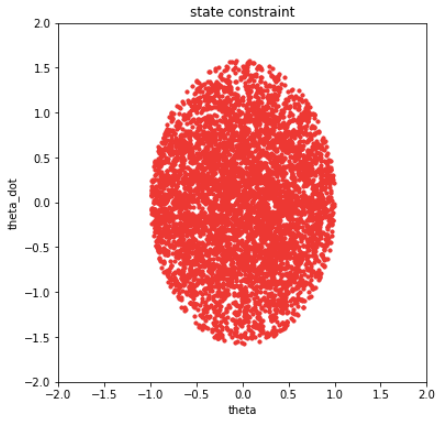

図 2　状態集合 C

式 (6) における K を $K(x) = \gamma x$ とおくと式 (6) は次のようになる.

$$\sup_{a \in A} \left\{ -2\theta\dot{\theta} - \frac{3g\alpha}{l}\dot{\theta}\sin\theta - \frac{6\alpha}{ml^2}\dot{\theta}a + \gamma(1 - \theta^2 - \alpha\dot{\theta}^2) \right\} \geq 0 \tag{13}$$

括弧の中は a に関する 1 次式となっているため $\dot{\theta}$ の正負に合わせて $U(s)$ を定義できる ($\frac{6\alpha}{ml^2}$ は必ず正のため).

さて, 括弧の中身を

$$p(a) = -2\theta\dot{\theta} - \frac{3g\alpha}{l}\dot{\theta}\sin\theta - \frac{6\alpha}{ml^2}\dot{\theta}a + \gamma(1 - \theta^2 - \alpha\dot{\theta}^2) \tag{14}$$

としたとき, $p(a) = 0$ の解を a^* とすれば, a は 1 次元なので $\theta < 0$ のとき $U(s)$ は以下のようになる.

$$U(s) = \begin{cases} [\ a^*, 10\ ] & \text{if} \quad a^* > -10 \\ [\ -10, 10\ ] & \text{if} \quad a^* \leq -10 \end{cases} \tag{15}$$

また, $\theta > 0$ のときも同様にすると

$$U(s) = \begin{cases} [\ -10, a^*\ ] & \text{if} \quad a^* < 10 \\ [\ -10, 10\ ] & \text{if} \quad a^* \geq 10 \end{cases} \tag{16}$$

として与えることができる.

4　実験結果

Open-AI Gym Pendulum-v0 と Keras-RL を用いてシミュレーションを行ったについて考察する. Keras-RL では学習を行う合計ステップ数をいくつかの episode に分割する. ここでは episode のステップを 2000 とし, 合計ステップを 3000000 に設定して実験を行った. 計算時間は 1 回あたり 5,6 時間である. 以下では (1):episode を通して状態制約を満たしているのか, またその時 CBF の働きはどうか. (2):イベント駆動制御がなされているか. (3):episode を通しての蓄積 reward は増えているか. という観点で結果を確認していく.

4.1　CBF による入力射影ルールの変更

ここまで状態 s における制御入力 $\pi(s)$ が $U(s)$ の元でない場合, $\pi(s)$ ではなく, $U(s)$ の元のうち最も $\pi(s)$ 近いものを, 実際にその時刻でシステムに加える入力だとしていた. しかしこの設定で学習を行っても, 方策の改善は見られなかった. 図 3 は横軸を episode, 縦軸を各 episode を通しての reward として, その変化を示したグラフである. オレンジ線は 30episode 移動平均線である. この図から, episode を進めて行っても方策が改善せず, また振子を倒立させた (この時は 0 付近の episode reward になる)episode がほとんどないことが見て取れる.

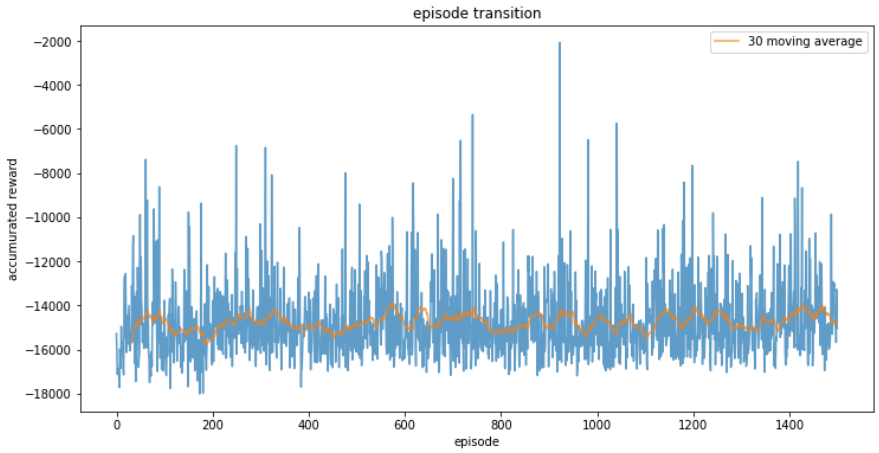

図 3　episode 毎の蓄積 reward

また, 3000000 ステップの学習によって得られた $\pi(\cdot)$ によって, 1episode 内でどのような制御が行われるかを見てみる.

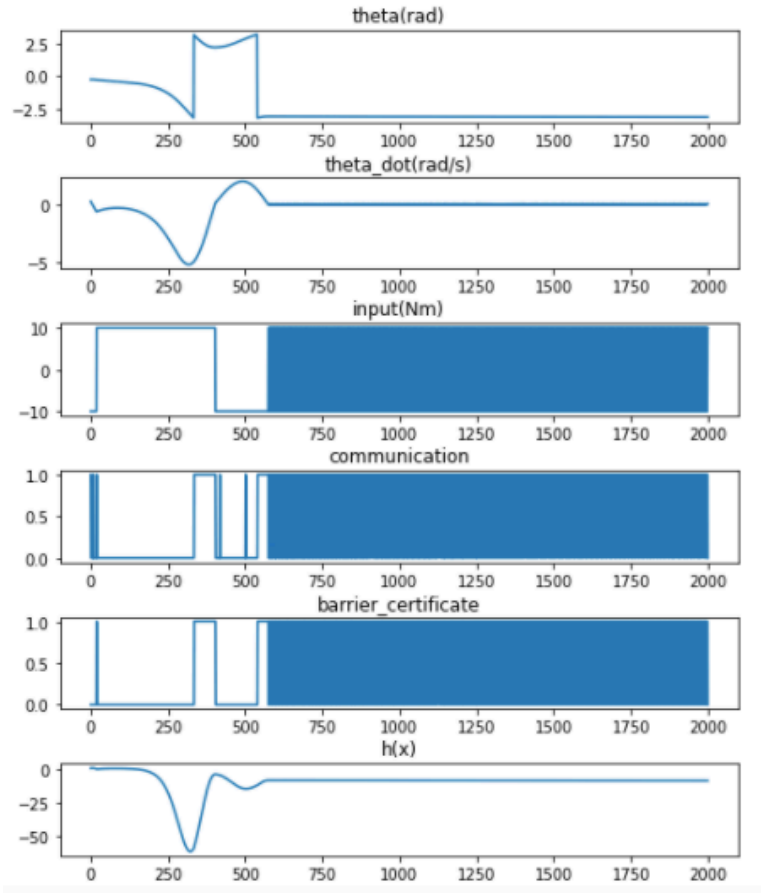

図 4　制御内容

図 4 は上段からそれぞれ $\theta, \dot{\theta}$, a(入力), 通信の有無, CBF による入力射影の有無, 各時刻の $h(s)$ が, 時間ステップ毎にどう変化するかを表したグラフである. 図から離散化の影響によって 200 ステップあたりから状態制約を満たさなくなってしまい, $\dot{\theta}$ を 0 にすることによって reward を大きくするという局所最適方策に陥っているように見える.

これを受けて, CBF による入力射影ルールを少し改変した. これは $h(s) < 0.1$ となった時のみ, $U(s)$ の元のうち最も $\pi(s)$ 近いものではなく, 最も効率の良い入力 (例えば, $\theta < 0$ であれば $a = 10$) を選ぶようにする

というものである. 次節からはこの入力ルールによる学習の内容を示す.

4.2 状態制約と CBF

図 5 にランダムに入力を返すような初期方策 $\pi_0(\cdot)$ による episode の実績を示す. 両図ともに横軸はステップを示す. 上図はステップ毎に $h(s)$ をプロットした図 (参考に y 座標が 0 となるラインをオレンジで引いている) である. 下図は CBF によって $\pi_0(s_t)$ が変更された場合に 1, それ以外で 0 となるようにプロットした図である.

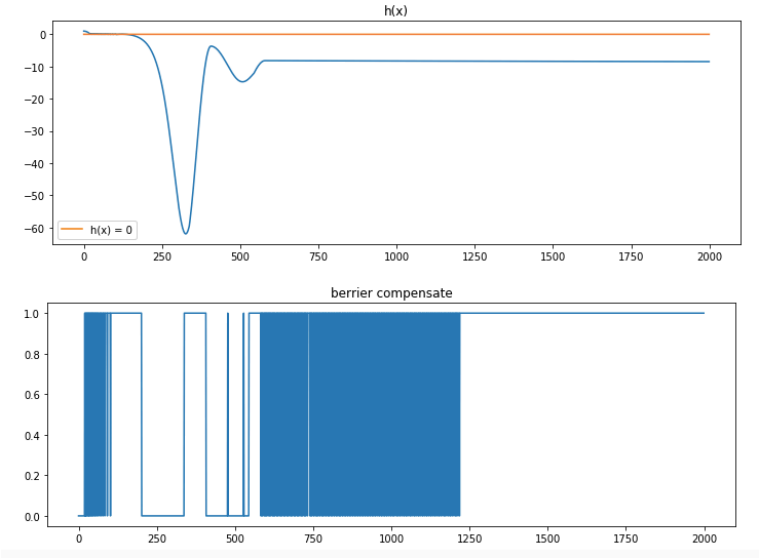

図 5　学習初期の $\pi(\cdot)$ の実績

この図を見ると 200 ステップ目までは状態制約 $h(s) \geq 0$ を保てているが, それ以降では制約を満たせていない ($h(s) < 0$). 前節で述べたとおり, CBF は連続時間システムに対して成り立つ論理なので, 離散化したシミュレーションではその論理が破綻することがある. これはその影響である.

次に学習後期の方策方策 $\pi_{t_{late}}(\cdot)$ の実績を見てみる.

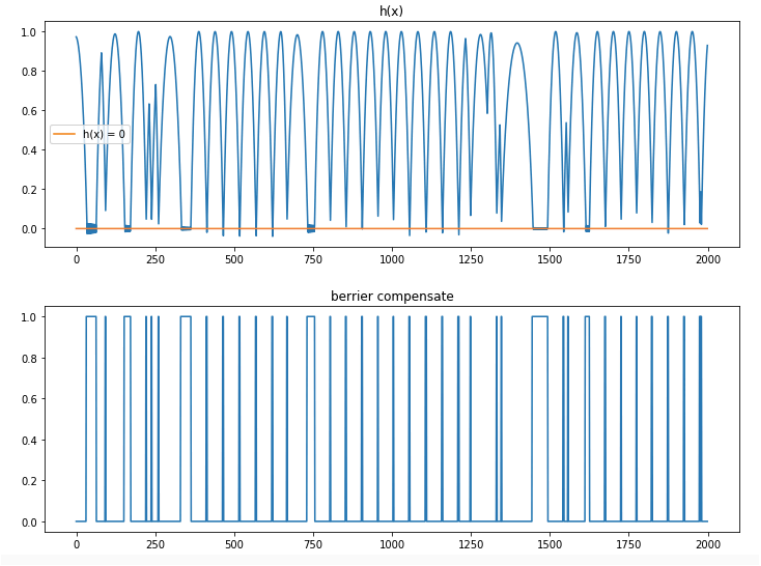

図 6　学習後期の $\pi(\cdot)$ の実績

こちらでは episode を通して状態制約 $h(s) \geq 0$ を保てている. 初期方策との違いは, ランダムな制御則では

なく, reward を大きくしようとする制御則になっていることである. 状態制約が保たれる理由は, これにより制約集合内に留まろうとする回数が多くなり, 離散化による CBF の破綻が起こる回数が少なくなっているためと考えられる.

最後に, 全 episode を通して同じグラフを描いた結果を示す.

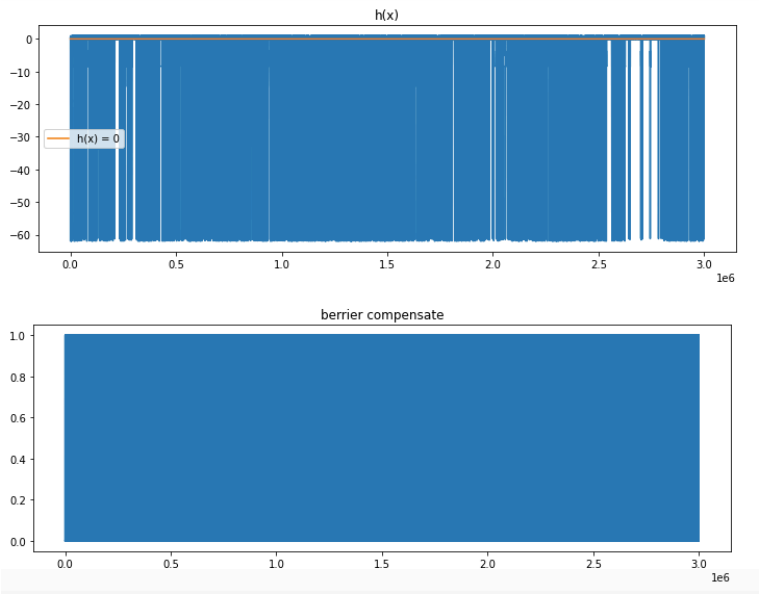

図 7　全エピソードの実績

エピソードの始まりに, 状態が $h(s_{init}) > 0$ となるように選んでいるため, 各エピソードの開始時に強制的に h のグラフが 0 より大きくなる. 図 5 を見ると, 0 以上から最小値付近に遷移する回数が, 後半になるにつれて少なくなっている. これは, 状態制約を保てる回数が増えていることを示し, エージェントが状態制約集合の中にある最適状態 (倒立) を留めておける方策の学習を進めることができていることの表れだと考える.

4.3　最適イベント駆動制御になっているか

次に学習した方策はイベント駆動制御になっているのかを見てみる.

図 8 は図 4 と同じクラスのグラフである. 図 8 の 4 段目を見ると通信を行っている回数が 10% ほどに抑えられており, イベント駆動制御になっていると言える. また, 1,2 段目を見ると $\theta, \dot{\theta}$ は 0 付近にとどまっており, 振り子が倒立していることを示している.

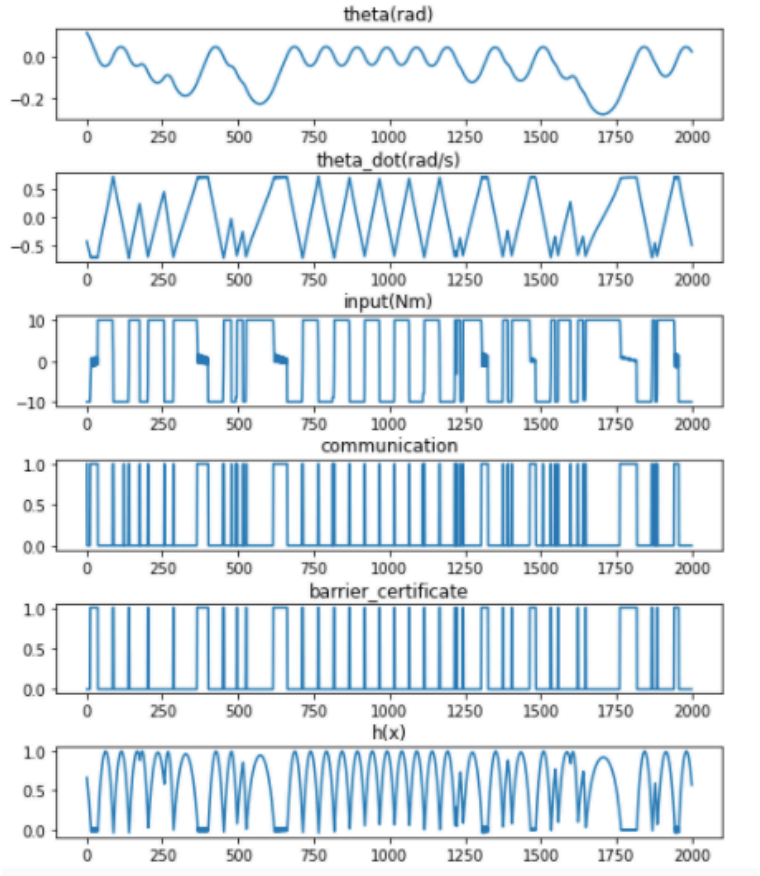

図 8　制御内容

4.4　episode reward は増えているのか

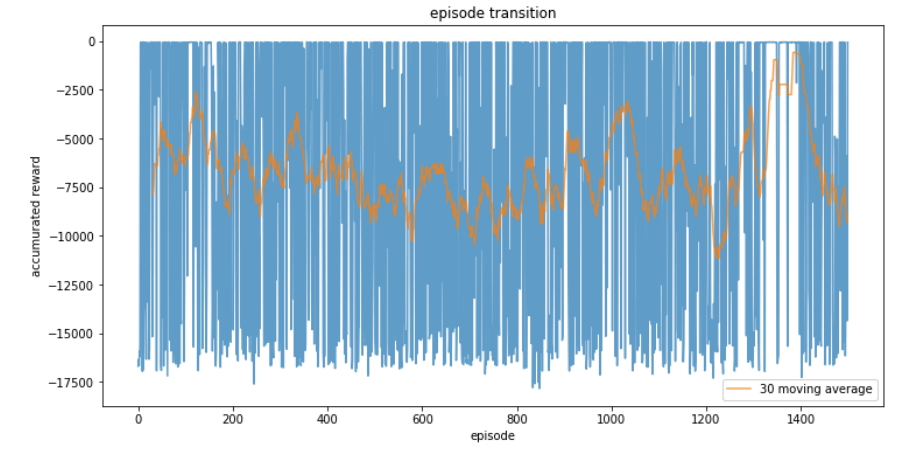

図 9　episode 毎の蓄積 reward

図 9 でも図 3 と同じように全 episode を通しての reward の変化をグラフにした．30episode 移動平均を見ても上昇していることは確認できないが, 図 3 とは異なり, 0 に近い値をとっている．これは振子を倒立させることに成功している episode で確認される reward である．図 9 と図 3 のちがいは $U(s)$ を如何に用いるかの違いなので, CBF による強化学習は, 離散化による影響をどれだけ抑えられるのかに焦点を当てるべきかもしれない．

4.5 課題

離散化によって状態制約を満たさなくなってしまうトリガーがなにかわかっていないので, それについては今後の課題である.(θ ではなく $\dot{\theta}$ による影響が強いのではないかと予想しています.)

参考文献

[1] D. Baumann, J. J. Zhu, G. Martius, and S. Trimpe. “Deep Reinforcement Learning for Event-Triggered Control.” *In Proc. of the 57th IEEE International Conference on Decision and Control*, 2018.

[2] Li Wang, Evangelos A Theodorou, and Magnus Egerstedt. “Safe learning of quadrotor dynamics using barrier certificates,” *In 2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2460-2465, 2018

[3] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick. “End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks,” *Thirty-Third AAAI Conference on Artificial Intelligence*, 2019.