強化学習による振子の振り上げ制御

2010SE051 市川順也
指導教員：大石泰章

1　はじめに

近年, ロボットの存在が私たちの生活に身近なものになってきた．ロボットの可能性を広げるためには，目的を与えれば目標軌道を自動生成するような，自律化された制御法が望ましい．強化学習はそのような制御法として期待を集めている．

従来の強化学習ではシステムの状態や入力は離散的であると仮定していたが，Doya[1] は連続な状態や入力を持つシステムの強化学習を提案し，ロボットなどの制御に応用する道をひらいた．

本研究では，文献 [1] に沿って強化学習による振子の振り上げを試み，実装上の問題点の抽出と実用性の検証を行う．

2　制御対象

本研究では，単純でありながら非線形性の強い振子の振り上げ問題に対して強化学習の適用を試みる．以下に振子の概略を示す．

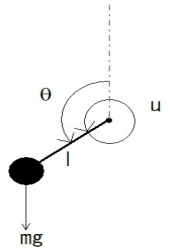

図 1　振子の概略

ただし，鉛直上向きの位置を基準とした振子の角度を θ[rad]，振子の先端の質量を m[kg], 振子の長さを l[m] とする．

このとき振子の運動は

$$ml^2\ddot{\theta} = -b\dot{\theta} + mgl\sin\theta + u \tag{1}$$

に従う．本研究では $m = 1$, $l = 1$，重力加速度を $g = 9.8$, 粘性摩擦係数を $b = 0.01$ とする．状態変数を $\mathbf{x} = (\theta \ \ \dot{\theta})^{\mathrm{T}}$ と定めたとき，状態空間表現は

$$\dot{\mathbf{x}} = f(\mathbf{x}(t),\, u(t)) = \begin{pmatrix} \dot{\theta} \\ -\frac{b}{ml^2}\dot{\theta} + \frac{g}{l}\sin\theta + \frac{1}{ml^2}u \end{pmatrix} \tag{2}$$

と書ける．

3　強化学習の適用

3.1　報酬の設定

強化学習では，状態変数 $\mathbf{x}$ と入力 u のある関数が大きくなるように学習を行う．これを報酬と呼ぶ．ここでは次の形の報酬を考える：

$$r(\mathbf{x},\, u) = R(\mathbf{x}) - S(u). \tag{3}$$

ただし，$R(\mathbf{x})$ は状態変数 $\mathbf{x}$ に依存する報酬関数であり，$S(u)$ は入力が大きすぎないようにするためのコスト関数である．

本研究では $R(\mathbf{x}) = \cos\theta$ と設定した．また，制御対象の入力には $|u| \le u^{\max}$ という制限がある場合を考え，

$$S(u) = c\int_0^u s^{-1}(\frac{u}{u^{\max}})\mathrm{d}u \tag{4}$$

と定める．ここでは特に，$c = 0.1$，$u^{\max} = 5$，$s(x) = \frac{2}{\pi}\arctan(\frac{\pi}{2}x)$ と定めた．$u = 0$ では，$S(u) = 0$ であるが，u が $\pm u^{\max}$ に近づくと $S(u)$ は無限大に発散する．

3.2　価値関数の学習

強化学習では，今後得られると予想される報酬に基づいて与える入力を決定する．これを価値関数と呼ぶ．ここでは入力 u を $\mathbf{x}$ の関数として $\mu(\mathbf{x})$ のように定めるが，この関数 μ を固定したとき，価値関数は以下のように定義される：

$$V^{\mu}(\mathbf{x}(t)) = \int_t^{\infty} \mathrm{e}^{-\frac{s-t}{\tau}} r\Big(\mathbf{x}(s), \mu(\mathbf{x}(s))\Big)\mathrm{d}s. \tag{5}$$

ただし，$\tau > 0$ は未来の報酬を割り引く時定数を表す．

学習の際に，価値関数を式 (5) に従って計算するのは現実的でない．そこで式 (5) を満たす $V^{\mu}(\mathbf{x}(t))$ が

$$\delta(t) := r\Big(\mathbf{x}(t), \mu(\mathbf{x}(t))\Big) - \frac{1}{\tau}V^{\mu}(\mathbf{x}(t)) + \dot{V}^{\mu}(\mathbf{x}(t)) = 0 \tag{6}$$

を満たすことに着目し，この式を満たすように $V^{\mu}(\mathbf{x}(t))$ を学習することを考える．

価値関数 $V^{\mu}(\mathbf{x}(t))$ の候補として，正規化されたガウス分布の重み付き和を使い，$\delta(t) = 0$ となるように重みを調整することを考える．これをノーマライズド・ガウシアン・ネットワークと呼ぶ．以下にこの場合の具体的な学習方法を示す．

3.2.1 ノーマライズド・ガウシアン・ネットワークを使った関数近似

状態空間中の考えたい範囲に K 個の点 $\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_K$ を配置する．今回考える範囲は，$-\pi \leq \theta \leq \pi$，$-\frac{5}{4}\pi \leq \dot{\theta} \leq \frac{5}{4}\pi$ とし，$\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_K$ は 16×16 の等間隔の格子状に配置する．格子点それぞれに対して，以下のようにガウス分布を構成する：

$$a_k(\mathbf{x}) = \mathrm{e}^{-||\mathbf{x}-\mathbf{c}_k||^2} \quad (k = 1, 2, \cdots, K). \tag{7}$$

さらに $a_k(\mathbf{x})$ を関数 $b_k(\mathbf{x})$ として正規化する．

このとき，価値関数の候補 V をパラメータ $\mathbf{w} = (w_1, w_2, \cdots, w_K)$ を用いて

$$V(\mathbf{x};\, \mathbf{w}) = \sum_{k=1}^{K} w_k b_k(\mathbf{x}) \tag{8}$$

のように選ぶ．パラメータ $\mathbf{w}$ は，式 (6) が成り立つように逐次調整する．

4 シミュレーション結果

4.1 振子の振り上げ

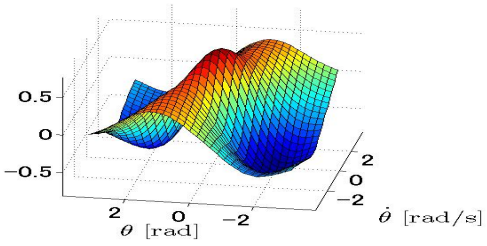

図 2 価値関数

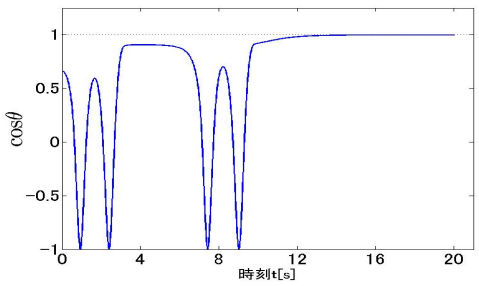

図 3 $\cos\theta$ の時間による変化

図 2 は試行回数を $n = 65$ として学習を行った後の価値関数のグラフである．なお本研究における試行とは，初期値から 20[s]，学習を行いながら振子を動かすことをいう．初期値は，角度 θ をランダムで与え，角速度を $\dot{\theta} = 0$ とする．図 2 からも分かるように，学習を行うと角度 $\theta = 0$，角速度 $\dot{\theta} = 0$ 付近での価値が最大となる．この価値関数を使って，最適な入力 u は決定される．

図 3 は試行回数を $n = 65$ として学習を行った後，価値関数を使って入力 $u(t)$ を決めたときの，$\cos\theta$ の変化のグラフである．$\cos\theta$ の値が 1 に近いほど，振子が高い位置にあることを示す．図 3 からも分かるように，トルクの制限を破らないように何度か振り上げ動作を行って勢いをつけたあと，速やかに倒立姿勢に入っていることが分かる．

4.2 基底数と学習速度の関係

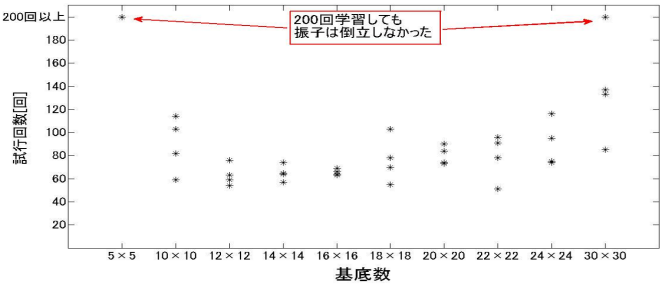

図 4 基底数と目標達成までの試行回数比較

図 4 は基底数を 5×5 から 30×30 まで変化させたときの，目標達成までの試行回数の比較である．試行時間 20[s] のときに $\cos\theta > 0.9$ を，5 回連続で満たした場合を目標達成とする．本研究では試行回数が 200 回を超えても目標達成されない場合は，学習失敗とし，学習を打ち切った．なお，その場合の試行回数は，図 4 では「200 回以上」にプロットした．また，本研究ではそれぞれの基底数に対して 4 度の学習を行った．

図 4 からも分かるように，基底数が 5×5 では，4 度とも学習に失敗した．これは，基底数が少なすぎて精度よく価値関数 V の近似が表現できなかったためと思われる．また逆に，基底数が 30×30 では，4 度中 1 度学習に失敗し，他の 3 度とも比較的多くの試行回数が必要であった．これは，価値関数 V の近似の表現の幅が増える一方，学習すべきパラメータの数が多くなるため，目標達成にはより多くの試行回数が要求されたのだと思われる．

以上の結果より，本研究では 14×14 から 16×16 の基底を配置することで，少ない試行回数で目標を達成できることが分かった．

5 おわりに

本研究では，強化学習の実用性の検証を行った．また，基底数と目標達成までの試行回数の比較を行い，適切な基底数を考察した．今後の課題として，さらに試行回数の短縮を試み，実機に実装したいと考える．また，他のシステムに対しても強化学習を実装したいと考える．

参考文献

[1] Kenji Doya: Reinforcement learning in continuous time and space. *Neural Computation*, vol. 12, no. 1, pp. 219–245, 2000.