1R-2

確率的２分木の動的生成に基づく Actor-Critic による ヒューマノイドロボットの運動学習

中田 尚吾　加藤 昇平　伊藤 英則

名古屋工業大学 大学院 工学研究科 情報工学専攻

1　はじめに

近年，ヒューマノイドロボット（HR）の制御手法として強化学習が注目されている．従来，強化学習で扱ってきた問題は離散状態空間であるが，HR を制御するためには連続状態空間を効率的に扱う必要がある．木村氏の研究 [1] では，行動選択において行動空間の位相構造に着目し多数の類似した行動を効率良く扱うために階層的行動選択を行う強化学習方式を提案している．しかし，この手法では学習の初期段階から 2 分木の全ノードにおいてパラメータ学習を行う必要がるため，1）与えられた環境において不必要な行動に対しても学習を行う必要がある，2）木構造が設計者に依存するため目標運動に特化した木を作れない，等の問題がある．そこで本研究では，確率的 2 分木を用いた Actor-Critic アルゴリズムにおいて 2 分木の動的生成を提案することにより，ロボットが状況に応じた必要な行動を動的に獲得することができる強化学習手法を提案する．

2　動的行動生成に基づく強化学習

2.1　2 分木を用いた Actor-Critic 法

本研究では，HR 制御のための強化学習法として，Actor-Critic 法を使用する．Actor-Critic 法は，Actor と呼ばれる制御器と Critic と呼ばれる評価器で構成される．

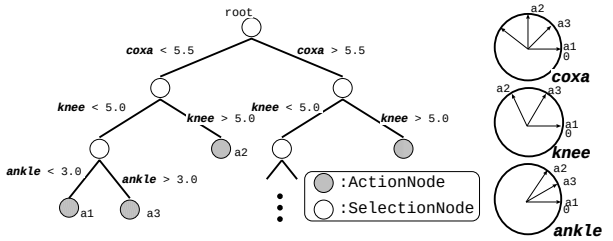

図 1: 行動選択 2 分木

本研究では，Actor における行動決定に確率的２分木 [1] を用いる．確率的２分木はの図 1 のように選択ノード（中間ノード）と行動ノード（葉ノード）から構成され，行動ノードには HR の制御情報として微小時間における各関節の角度変位が格納される．HR の状態を示す特徴ベクトル $\boldsymbol{s}_t$=$(x_1(t), x_2(t), \cdots, x_i(t), \cdots, x_n(t))$ が与えられたとき，2 分木を用いて採るべき行動を探索する．選択ノード j における分岐確率 f_j は以下の式で定義される．

$$f_j = \frac{1}{1+\exp(-\sum_{i=1}^{n}\theta_{ji}x_i)} \tag{1}$$

ここで θ_{ji} は政策パラメータであり，n は状態 $\boldsymbol{s}_t$ の特徴ベクトルの要素数である．得られた分岐確率 f_j で右子ノード（もしくは 1-f_j で左子ノード）を選択し，遷移を行なう．遷移先においても同様の処理を繰り返し，たどり着いた行動ノード（葉ノード）が示す行動を HR に実行させる．しかし，この手法は学習初期に 2 分木の木構造を与える必要があり，その木構造が学習の可否と効率に大きくに影響する．

2.2　行動ノードの評価

そこで本稿では行動ノードの動的生成手法を提案する．本手法では，全ての行動ノードに対し，行動の**行き詰まり度合**を示す変数 e_{act} を設定する．この変数は，学習中に得られる TD 誤差 $\delta(t)$ を政策の安定度合と見なし，$\delta(t)$ に応じた大きさの値をとるように更新される．

$$e_{act} \leftarrow e_{act} + h(\delta(t))r(t) \tag{2}$$

ここで，$h(\delta(t))$ は TD 誤差に応じた重みを返す正規分布の関数，$r(t)$ は時刻 t における報酬である．e_{act} は初期値として 0 をとる．確率的 2 分木を用いた Actor-Critic 法は，図 2 の処理を繰り返すことで，Critic は価値関数 $V(\boldsymbol{s})$ を正しく推定し，Actor は $V(\boldsymbol{s})$ を最大化する行

1) 初期化:Critic の n 個の変数 $\omega_i(i = 1 \cdots n)$ およびその適正度の履歴を保持する n 個の変数 $\overline{e_{vi}}$,Actor の $n(m-1)$ 個 (m-1 はノードの個数) の政策パラメータ変数 $\theta_{ji}(j = 1 \cdots m-1)$ およびその適正度の履歴を保持する $n(m-1)$ 個の変数 $\overline{e_{\pi ji}}$ を保持する．
2) 状態 $\boldsymbol{s}_t$ の特徴ベクトル $(x_1(t), x_2(t), \cdots, x_i(t), \cdots, x_n(t))$ を環境から観測する．
3) Actor の 2 分木より行動を選択する．
4) 報酬 r_t を受け取り，次の状態 $\boldsymbol{s}_{t+1}$ を観測する．Critic は以下の式により TD 誤差 $\delta(t)$ を算出する．ただし，γ は割引率．
$$\delta(t) = r_t + \gamma\, V(\boldsymbol{s}_{t+1}) - V(\boldsymbol{s}_t).$$
$$= r_t + \gamma(\textstyle\sum_{i=1}^{n}\omega_i x_i(t+1)) - \sum_{i=1}^{n}\omega_i x_i(t)$$
5) Critic は TD 誤差に基づき $V(\boldsymbol{s})$ を以下のように更新する．ただし，α_v は学習率．
$$e_{vi}(t) = x_i(t)$$
$$\overline{e_{vi}} \leftarrow e_{vi}(t) + \overline{e_{vi}}$$
$$\Delta\omega_i(t) = \delta(t)\overline{e_{vi}}(t)$$
$$\omega_i \leftarrow \omega_i + \alpha_v\Delta\omega_i(t)$$
6) Actor は適正度 $e_{\pi ji}(t)$ に基づき行動選択確率を更新する．ただし，α_π は学習率．
$$\delta(t) = r(t) + \gamma V(\boldsymbol{s}(t+\Delta t)) - V(\boldsymbol{s}(t)).$$
$$\overline{e_{\pi ji}}(t) \leftarrow e_{\pi ji}(t) + \overline{e_{\pi ji}}(t)$$
$$\Delta\theta_{ji}(t) = \delta(t)\overline{e_{\pi ji}}(t)$$
$$\theta_{ji} \leftarrow \theta_{ji} + \alpha_\pi\Delta\theta_{ji}(t)$$
7) 適正度の履歴を以下のように更新する．ただし，λ_π, λ_π は減衰パラメータである．
$$\overline{e_{vi}}(t+1) \leftarrow \gamma\lambda_v\overline{e_{vi}}(t).$$
$$\overline{e_{\pi ji}}(t+1) \leftarrow \gamma\lambda_\pi\overline{e_{\pi ji}}(t).$$
$$V(\boldsymbol{s}(t)) = \textstyle\sum_i^N v_i b_i(\boldsymbol{s}(t)).$$
8) 試行の終了条件を満たしてなければ，時刻を更新してステップ 2 に戻る．
$$t \leftarrow t + \Delta\, t.$$
9) 試行における TD 誤差の絶対値の平均と 2 分木における行き詰まり度の最大値を観測し，新たな行動の生成条件を満たすか判断する．

図 2: 動的行動生成に基づく Actor-Critic

*Rainforcement Learning of Humanoid Robots using Actor-Critic based on Dynamic Construction of Binary Tree Action Selector, Shogo NAKATA, Shohei KATO, and Hidenori ITOH, Department of Computer Science and Engineering, Graduate School of Engineering Nagoya Institute of Technology, Gokiso-cho, Showa-ku, Nagoya 466-8555, Japan.

動を選択するように学習を行う．学習の更新には適正度の履歴 [2] を用いる．

2.3 新規行動ノードの生成条件

提案手法の Actor-Critic は以下の条件を満足するとき，新たに選択できる行動を生成する．

$$|TD|_{ave} < T_{add} \quad \text{and} \quad \max(e_{act\ i}) > e_{max} \qquad (3)$$

ここで，T_{add}, e_{max} は閾値である．また，$|TD|_{ave}$ は一試行を通してロボットが得た TD 誤差の絶対値の平均であり，その試行における学習の収束度合を表す．$max(e_{act\ i})$ は一試行の終了時における全行動ノード中の行き詰まり度 e_{act} の最大値を表す．つまり，$|TD|_{ave}$ が閾値より低いことで学習の収束を判断し，そのとき，行動ノードの e_{act} の最大値が閾値より超えたか否かで，学習が行き詰まっているか安定して収束しているかを判別する．

2.4 新たな行動の生成

ある行動ノード a_1 が式 (3) の条件を満足した場合，a_1 が保持する各関節の制御角が最も近い行動ノード a_2(図 1(右) 円盤参照) との間に新たな行動を生成する．新たな行動 a_3 の制御値は，2 つの行動 a_1, a_2 の行き詰まり度 e_{act1}, e_{act2} の比に基づき，以下の式を満足する値をとる．

$$e_{act1} : e_{act2} = ang_{2-3} : ang_{1-3} \qquad (4)$$

ここで ang_{1-3} は a_1, a_3 の持つ制御角の差の絶対値（図 1(右) 円盤上における a_1, a_3 の角度）を表す．つまり a_1, a_2 の制御角の内分比が行き詰まり度の比と等しくなるように a_3 が作成される．a_1 は新たに選択ノード b に置換され，選択ノード b の子ノードは，a_1 と新規行動ノード a_3 となる．b に対する選択確率は $f_b = 0.5$ に初期化される．a_1 と a_3 の左右関係は，2 分木において行動の類似性と整合するように決定される．また a_1 の行き詰まり度 e_{act1} は初期値にリセットされる．

3 動作実験

本手法の有効性を確認するために，HR の椅子からの立ち上がり動作の学習を行う（図 3）．本実験では，富士通オートメーション製 HOAP-1（図 4）の機構データを用いて動力学シミュレータを作成した．学習シミュレーションには，Open Dynamics Engine[3] を用いた．

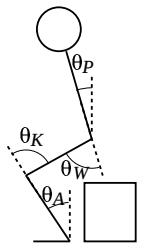

図 3: 学習動作

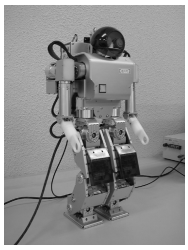

図 4: HOAP-1

Actor の行動出力は，腰，膝および足首の角度変化量 $(\Delta\theta_W, \Delta\theta_K, \Delta\theta_A)$ とした．確率的 2 分木への入力は，$(\theta_W, \theta_K, \theta_A, \theta_P)$ とした．学習パラメータは，$\gamma = 1.0$ $\alpha_v = 0.02$, $\alpha_\pi = 0.02$, $\lambda_v = 0.9$, $\lambda_\pi = 0.9$, $TD_{\text{add}} = 70.0$, $e_{\max} = 50.0$, $\Delta t = 0.01[s]$, とした．報酬 $r(t)$ は以下の式で与えた．

$$r(t) = -|\frac{y - l_s}{l_s - l_d}| \qquad (5)$$

ただし，y は胸の重心位置，l_s は起立時の胸の重心位置 ($l_s = 0.351$)，l_d は転倒時の胸の重心位置 ($l_d = 0.20$) である．また，立ち上がりの途中段階においてサブゴールとして正の報酬を与えた．1 試行は，300 回の行動出力を行う，あるいは転倒したら終了とした．

4 評価

上記の条件下で実験を行なったところ，142 回目の試行で図 5 に示すような運動を獲得することができた．このとき作成された Actor の行動数は 33 個であった．作成された 2 分木では腰に関する選択ノードが 19 個，膝に関する選択ノードが 9 個，足首に関する選択ノードが 4 個得られ，腰を比較的細かく制御してバランスを保つ運動が獲得された．本手法の効果を検証するため，行動数を 20 個で固定した [1] との比較実験を行なった．図 5 に学習過程における獲得報酬の推移を示す．同図より，本手法の方が良い政策を得ていることがわかる．

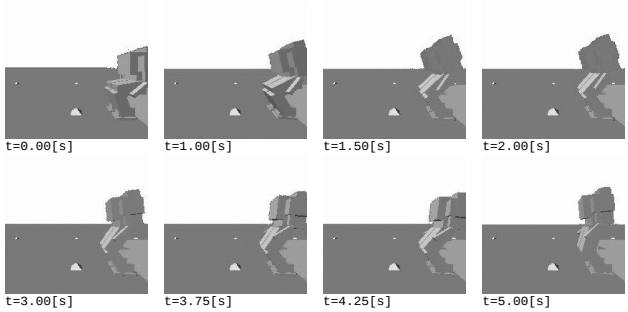

図 5: 獲得された運動

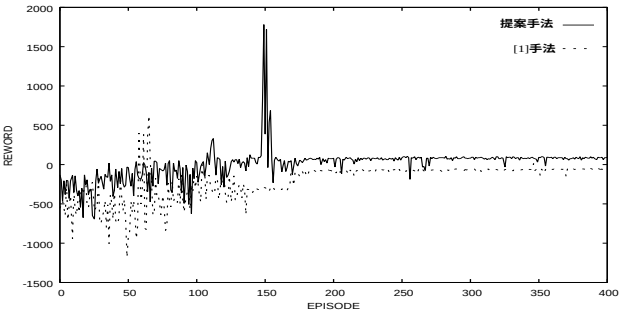

図 6: 獲得報酬の推移（5 回平均）

5 おわりに

本研究ではロボットの運動制御において，必要な行動を与えられた環境に応じて生成し，その制御を 2 分木で扱う手法を提案した．提案手法の適応的な動作獲得に基づく学習により，椅子からの立ち上がりにおいて，より良い政策を獲得することに成功した．本稿では制御を細分化させる手法を提案したが，今後，細分化した行動の汎化や，学習中に木構造を再構築する手法を提案する．また，他の運動制御について提案手法を適用する必要がある．

参考文献

[1] 木村 元，小林 重信：確率的 2 分木の行動選択を用いた Actor-Critic アルゴリズム:多数の行動を扱う強化学習計測自動制御学会論文集, Vol.37, no.12, pp.1147–1155 (2001)

[2] 木村 元，小林 重信：Actor に適正度の履歴を用いた Actor-Critic アルゴリズム-不完全な Value-Function のもとでの強化学習- 人工知能学会誌, Vol.15, No.2, pp.267–275 (2000).

[3] Russell Smith：Open Dynamics Engine，http://opende.sourceforge. net/ode. html.