ISSN 2186-5647

2-2

動的計画法と強化学習を統合した複数ロボットアームのピッキング制御

日大生産工(院)　○日暮　一登　　日大生産工　風間 恵介
日大生産工　丸茂 喜高

1. 緒言

現在の産業用ロボットは，ティーチングコスト削減のため，強化学習の導入が注目されている．しかし，既存のロボットアームと強化学習に関する研究は，ロボットアームと運搬対象が1対1という単純な環境が多くを占める．また，複数のロボットアームの協調動作に関する研究も存在するが，向かい合わせのロボットアーム2台が，机に設置されたボタンを押すという簡易的なもの[1]のみであり，コンベアやロボットアーム先端に付けるハンド部分を扱う実用性を伴った制御を行っているものはない．これは，時間毎の関節角度を出力させるような動的制御を，強化学習で実現しようとすると，数千万以上のStepが必要となり，それを複数で行うとなると，更に膨大な学習コストが必要になるためである．一方で，ロボットアームの可動エリアが干渉する状況という，複雑な動作の最適化は，アルゴリズムを組んだプログラムによる直接制御で実現することは難しい．

以上を踏まえ，本研究では強化学習と動的計画法を組み合わせたロボットアームの制御手法を提案する．通常，複数のロボットアームでコンベア上を流れる搬送物を移動させるタスクにおいては，①決められた手順で把持動作をするロボットアームを選定，②決められた座標でその動作を行うシーケンス制御，が一般的である．このロボットアームの選定と，把持動作の座標を，強化学習を用いて改善する．Unity[2]を用いたシミュレーションにより2つのロボットアームの制御の評価を行い，直接制御と強化学習との収集率を比較し，提案手法の有効性を確認した．

2. 強化学習とシミュレーションの概要

強化学習には様々なアルゴリズムがあるが，共通して状態 s，行動 a，報酬 r，各種ハイパーパラメータの適切な設定が必要である．式（1）に本研究で使用する Proximal Policy Optimization (PPO) アルゴリズム [3]の報酬関数を示す．PPO における行動 a は，離散値（0, 1, 2…）と連続値（–1.0～1.0）に対応している．PPO 最大の特徴として，更新前の方策 $\pi_{\theta_{\mathrm{old}}}(s \mid a)$ と更新後の方策 $\pi_{\theta}(s \mid a)$ の比率 $r_t(\theta)$ を，ハイパーパラメータ $1-\varepsilon$ から $1+\varepsilon$ に収まるよう clip し，極端な方策変更を防ぐというものがあり，制御方面で多くの実績がある．

$$L^{\mathrm{CLIP}}(\theta) = E_t[\min(r_t(\theta)A_t,\ \mathrm{clip}(r_t(\theta)),\ 1-\varepsilon,\ (1+\varepsilon)A_t)] \quad (1)$$

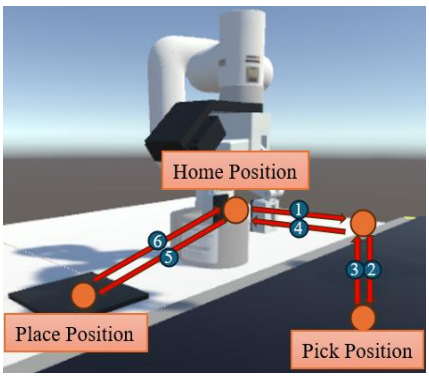

図1　動的計画法の軌跡

ロボットアームの動的計画法は，先端座標から角度を逆算し，この座標を時間の関数で表すことで動的制御を可能にするものである．図1に動的計画法の軌跡を示す．ホームポジションから始動して，把持座標の上部を経由し，把持座標で搬送物を把持して同じ経路で戻り，搬送先へ移動したらホームポジションに戻る．この図の4点の座標を一巡して移動する動作を1タスクという単位でまとめる．ホームポジションと搬送先は変化しないため，把持動作を行うロボットアームと把持座標の指定だけをすれば，搬送物がその位置へ到達するタイミングでホームポジションからスタートさせ，自動で搬送物が回収される．タスクの実行データは，搬送物が特定の座標を通過した際に生成される．タスク実行中に搬送物を検知した場合でも，搬送物に間に合う場合に対応するため，各ロボットアームに3つまで実行データを格納できる．実行中のタスク終了後，速やかに次のタスクを実行する．図2に強化学習における行動と実行動作の対応を示す．離散値（0, 1, 2）は，搬送物を無視（0）するか，2台のロボットアームのいずれかに把持動作を実行させる（1, 2）選択に対

Integrated Dynamic Programming and Reinforcement Learning for Multi-Robot Arm Picking
Kazuto HIGURASHI, Keisuke KAZAMA and Yoshitaka MARUMO

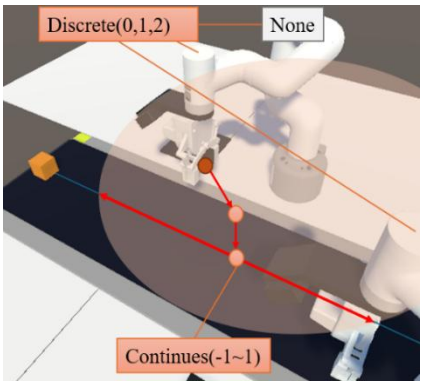

図2　強化学習の行動と実行動作の対応

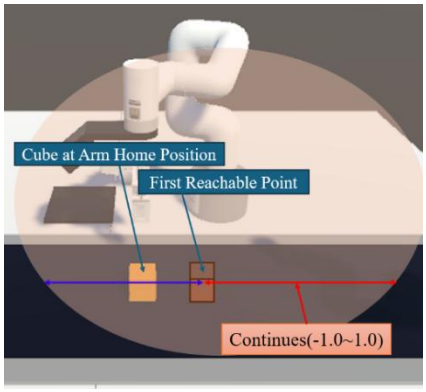

図3　到達可能位置に制約をつけた連続値のスケーリング

応している．次に選択したロボットアームがどの位置で把持動作を行うかを，ロボットアーム可動範囲に対して，連続値（–1.0～1.0）にスケーリングする．しかし，このままでは，前方では搬送物を取得できないが後方までアームを伸ばせば搬送物を取得可能というケースに対応できない．これを解決するために，到達可能位置に制約をつけた連続値のスケーリングを図 3 に示す．直線移動の偏差射撃（移動目標に対して先回りして取得するための式）の計算を応用して取得可能な範囲を割り出し，奥側のロボットアーム可動範囲とでスケーリングする．また，この偏差射撃の計算結果が奥側のロボットアーム可動範囲を超えていた場合は実行不能とみなして，そのタスクは破棄される．仮に別のロボットアームならそのタスクを請け負えた場合でもリカバリーは行わない．これは，強化学習が起こす行動が，実際に起こる現象と非対応になることで法則性を見つけにくくなるのを防ぐためである．

図 4 にオブジェクトの移動条件を示す．搬送物は速度 50mm/s，間隔 160mm で送る．ロボットアームは，先端速度 100mm/s で空間上を移動し，それに追従した角度変更という制約で動作する．

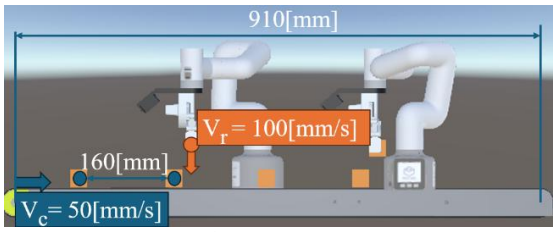

図4　オブジェクトの移動条件

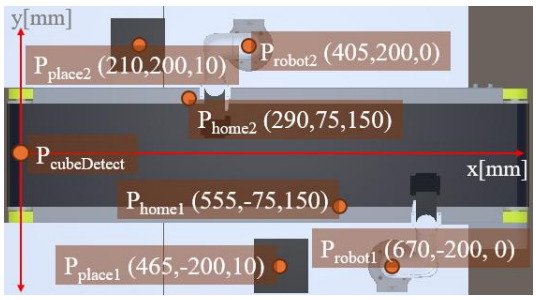

図5　オブジェクトの座標

図 5 にオブジェクトの座標を右手座標系で示す．これは図 4 を上から見た図で，単位は mm で統一されている．また，使用するロボットアームは Elephant Robotics 社製の myCobot 280-M5 である．

表1は強化学習の入力情報として加える状態である．画像ではなく数値の配列を採用しており，合わせて(3×3＋2)×2＝22個の数字で環境を認識する．図6に学習全体のフローチャートを示す．このフローは左上の搬送物の検知に戻るまでのループを1秒に50回のペースで繰り返す．搬送物を検知していない間も必ずReward –0.0002を通るため，–0.0002×50 =–0.01より，ロボットアームの動作に関わらず，1秒間に–0.01の報酬が発生する．この形式の報酬を生存コストと呼ぶ．左列中心に書かれているアクションが適切であるかの判断は，更新された離散値の行動が，タスクを請け負えるロボットアームの選択，またはどのロボットアームもタスクを請け負えない条件を選択できていた場合は+0.1，そうでなければ–0.05という基準で行われる．接触せずに1000Stepに到達すれば報酬をそのまま獲得できるが，接触した場合は–Step×0.2の報酬が発生し，大きく減点される．

表 1　入力情報として加える状態

Additional states	type	the number of robots and buffers
Can tasks be added?	bool	×2
Fastest arrival Y-coordinate	float	
RobotTask.PickPosition	float	×2×3
RobotTask.RemainingTime	int	
Robot.Task.flag	bool	

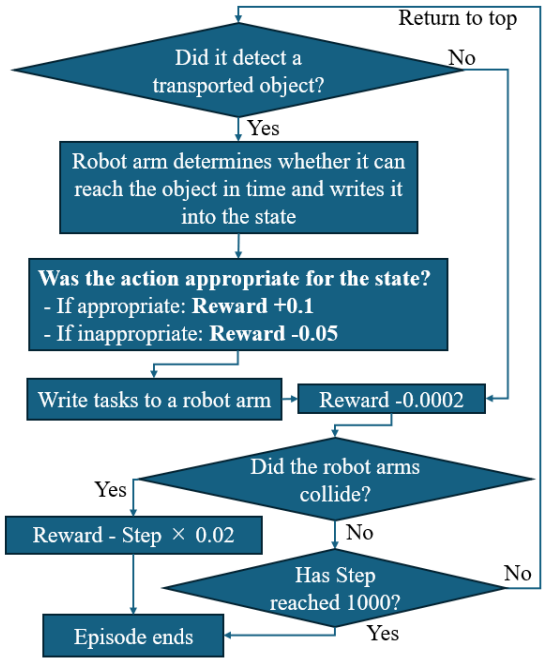

図6　学習全体のフローチャート

3.　シミュレーション結果

PPO におけるハイパーパラメータの典型な値は，割引率γが 0.99 程度，Generalized Advantage Estimation(GAE)パラメータλが 0.95 程度，クリッピング係数εが 0.2 前後である．しかし，今回のシミュレーションは搬送物の検知時にのみ状態遷移し，時間的な変化を Agent は認識せず，似た動作を永続的に反復する学習であり，これらが大きく異なる可能性が高い．また，学習コストと精度のトレードオフとなる多くのパラメータと異なり，これらの最適値は学習の安定性と性能に直結する．そのため，γ，λ，εの 3 つのハイパーパラメータを，典型的な値の周辺ではなく，0〜1.0 全てのレンジでグリッドサーチを行った．

図 7 にγによる報酬の比較を示す．γが低いほど最終的な報酬が高くなる傾向が確認された．これは，報酬を十分に積むと接触を起こし，エピソードを終了させる動作を，γを少なくすることによって，過去の報酬の影響を減らし，防ぐ役割があったと考えられる．図 8 にλによる報酬の比較を示す．その他のパラメータは図 7 にて最も最終報酬が高かったものから引き継いだ設定を用いた．λが 0.1 の場合と 0.95 の場合に高いパフォーマンスが見られるが，その中間の値に大きな変化はない．このことから，短期的に搬送物を取りに行く動作と，長期的に見てロボットアームの接触を避ける動作との，どちらの方針を極端に重視するような Agent がこの学習環境では適していると考えられる．図 9 にεによる報酬の比較を示す．その他のパラメータは図 8 の時と同様である．εは方策をどれだけ許容するかのパラメータである．今回の学習環境は，0.5 付近の比較的大きな方策変更が最も報酬が高い．これは，一連の長期的な動作を一つにまとめた行動を出力するため，大きな方策変化が必要になり，逆の U 字の相関が表れたのだと考えられる．

表 2 に搬送物 1000 個を流したときの取得性能を示す．参考のため，把持座標を固定したプログラムのみで動作する直接制御も，比較として示す．搬送率は強化学習（図 9 のε=0.1 のモデルを使用）が 71.0%，直接制御は 78.6%であった．表 2 の一番下の離散値の正解率は，搬送率に加え，どちらのロボットアームも搬送物を取得できない状況で搬送物を無視する選択も成功とし，あえて見逃すという選択のみを失敗とした比率である．直接制御の搬送率と同じ 78.6%は，ロボットアームが多くのタスクを抱え，搬送物を取得できないという状況にならなかったことを示している．

表 3 に離散値の正解率とエピソード終了時のステップから期待される報酬を示す．この報酬は，1 タスクの平均所要時間の 9 秒分を生存コストを引いて概算している．　表 2 より，強化学習の離散値の正解率は 99.3%≒100%とする．表 3 の離散値の正解率 100 の行の報酬を用いる．1000Step 中の接触回数が 28 回より，1 エピソードの平均 Step は 35 回なので，エピソード終了時の Step は 30〜40 であり，Reward は 1.05〜1.4 の間ということになる．直接制御をモデルとして評価すると 1000Step まで離散値の正解率 80%程度で生存したため，Reward は 25 になる．

この差は，状態へ加えた「ロボットアームが搬送物を取得可能か」の（0, 1）情報が指標として大きく影響した一方で，「接触するかどうかを判断するための情報」の影響が小さかったことに起因すると考えられる．さらに，搬送物を取得可能な状況であえて無視する選択にも–0.05 の報酬が発生し，逆に両方のロボットアームが搬送物を取得できない状況でも搬送成功時と同等の報酬を得られる設定であったため，短期的な離散値の正解率を重視する傾向が生じた．これらの状態情報の重み付けと報酬設計のバイアスが相互作用し，結果としてタスクにかかる時間を軽視したものと考えられる．対策として，搬送物をあえて無視する選択への減点と，搬送物を取得できない状況で加点が発生する報酬設計を変更することがあげられる．

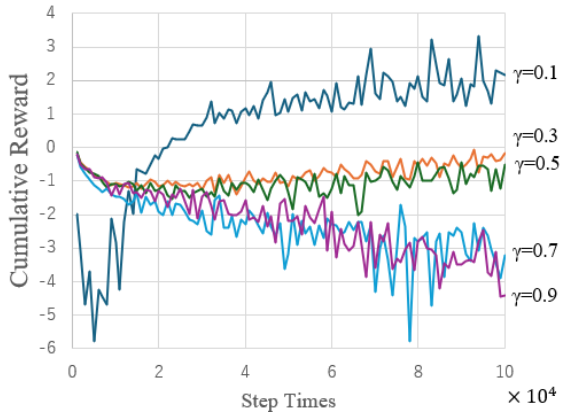

図7　γによる報酬の比較

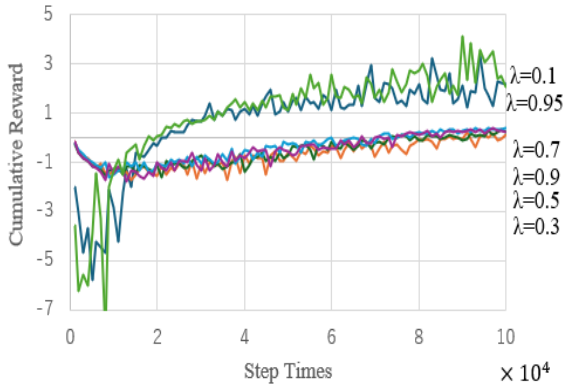

図8　λによる報酬の比較

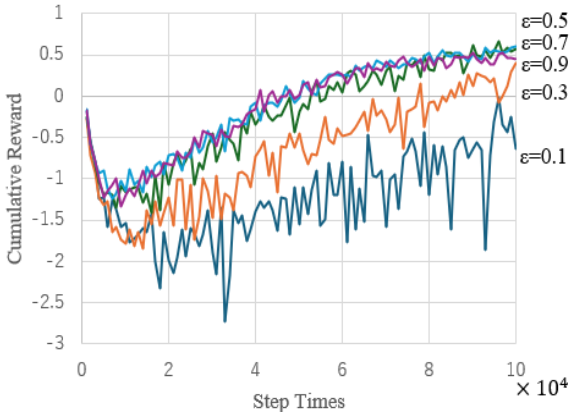

図9　εによる報酬の比較

表2　搬送物1000個を流したときの取得性能

	Reinforcement Learning	**Direct Control**
Number of pickups	710	786
Number of misses	290	214
Number of collisions	28	0
Percentage of correct discrete	99.3%	78.6%

表3　離散値の正解率とエピソード終了時のステップから期待される報酬

Percentage of correct discrete	Episode end step					
	10	20	30	40	50	1000
0	-1.15	-2.30	-3.45	-4.60	-5.75	-95.0
10	-1.00	-2.00	-3.00	-4.00	-5.00	-80.0
20	-0.85	-1.70	-2.55	-3.40	-4.25	-65.0
30	-0.70	-1.40	-2.10	-2.80	-3.50	-50.0
40	-0.55	-1.10	-1.65	-2.20	-2.75	-35.0
50	-0.40	-0.80	-1.20	-1.60	-2.00	-20.0
60	-0.25	-0.50	-0.75	-1.00	-1.25	-5.00
70	-0.10	-0.20	-0.30	-0.40	-0.50	10.0
80	0.05	0.10	0.15	0.20	0.25	25.0
90	0.20	0.40	0.60	0.80	1.00	40.0
100	0.35	0.70	1.05	1.4	1.75	55.0

4.　結語

本研究では，ベルトコンベア上の搬送物を運搬するロボットアームの制御を行うにあたって，直接制御よりも高精度な動作が可能な強化学習モデルの実現を目的とした．ハイパーパラメータγ, λ, εの調整をしてシミュレーションを行い，動的計画法による制御と適合する数値を検討した．

現段階では直接制御を上回る性能を得られなかったが，高精度のモデルを作る手法として以下のものが考えられる．

1. ハイパーパラメータや報酬
2. 状態の最適化，画像学習を用いる
3. 直接制御のデータで模倣学習をする
4. 別のアルゴリズムを用いる

また，搬送物のランダム性を上げる，コンベアやロボットアーム先端の速度関係を変えるなど，条件を変更して検討を進める．

参考文献

1）　北野郁弥, 内方英利, 松下光次郎, 志賀元紀, 佐々木実，“複数ロボットアームの協調動作における機構と制御の最適化”，日本 AEM 学会誌，Vol.29, No.1 (2021), pp. 161-162.

2）　布留川英一，“Unity ではじめる機械学習・強化学習 Unity ML-Agents 実践ゲームプログラミング v2.2 対応版”，株式会社ボーンデジタル，(2022).

3）　J. Schulman, F. Wolski, P. Dhariwal, A. Radford and O. Klimov, “Proximal policy optimization algorithms”, arXiv preprint arXiv:1707.06347, (2017).