深層強化学習を用いたロボットの自然言語による制御への取組み

橋本さゆり † 金子 晃 † 小林一郎 †
† お茶の水女子大学 人間文化創成科学研究科 理学専攻 情報科学コース
†{g1320530, kanenko, koba}@is.ocha.ac.jp

1 はじめに

近年，社会の高齢化に伴い，ロボットの普及が広がりつつある．今後，家庭内にロボットが入り込んだ際に人間がロボットに自然言語による指示を行い，ロボットが人間の言葉から新しい行動を自発的に学習してくれることが期待される．そこで本研究では，深層強化学習を援用したロボットの自然言語による動作制御の試みを目指す．そのため，ロボットの動作を自然言語の言葉で表現する仕組みの構築を試みる．すなわち，

1. ロボットにさせたい仕事（ジョブ）をそれに必要な動作を表す通常の言葉の列として表現する．
2. ロボットが深層強化学習で得た行動をそれらの言葉で人間に説明できるようにする．

の二つを目標とする．本発表では物理シミュレータ MuJoCo の上で動くロボットアーム（以下簡単のためロボットと略称する）を用い，具体例として，以下の図 1 のように右側の白い円柱とタブがある状態から，左側の状態にするというジョブを取り上げる．このジョブを我々は「(円柱をタブに）はめる」という言葉で表現するが，これを実現するのに，ロボットはどのような動作を行なわなければならないかを考える．

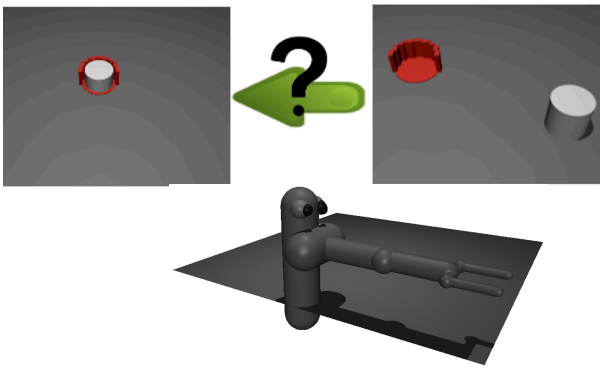

図 1: 「はめる」に対応する動作

このジョブを実現するためのロボットの動きは，関節の物理的な動きの系列となるが，それは人間には理解不能であり，かつ他のロボットへの転移学習も困難なデータとなる．これを人間に分かりやすい自然言語で表現可能ないくつかの中間動作（タスク）の流れとして表現し，更に個々のタスクを単位動作の組み合わせとして表現できれば，ロボットへの指示も出しやすく，他のロボットへの転移学習もやさしくなると考える．最初に挙げた例では,「はめる」は「掴んで」「運び」「押す」という一連のタスクの系列となり，個々のタスクは更に，例えば「掴む」は，腕を回し，手を下げ，指を閉じること，のように分解されて図 2 のような基本動作の系列になる．

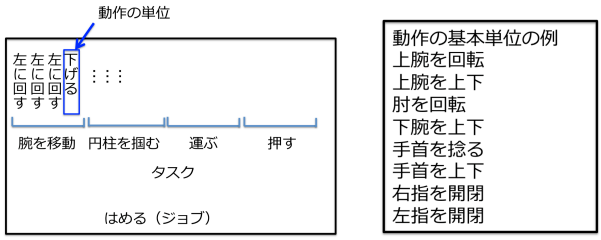

図 2: 動作の言語化

ここでのジョブとは，ロボットに与えられる課題であり，ロボットの動作を表現するのに目的語（ロボットの動作対象物）や補語（目標地点など）を伴う文章で表される．また，タスクとはジョブ達成のためのステップであり，これも目的語や補語を伴う文章で表される．ロボットは各タスクを一連の動作の系列で遂行する．この動作の基本単位に我々は自然言語の単語を割り当てる．本研究では更に，より難しい「詰める」というタスクを，ロボットに円柱を持ち上げて空中を移動させタブに上からはめる一連の動作として実現することも目指している．

2　記号列によるロボット制御

そこで以下のような動作の基本単位とそれを呼称する言葉の辞書を用意する．この表の順序は使用しているロボットのシミュレーションモデルにおける関節の番号に従っているが，我々は記号列ではなく，自然言語表現の共通化を目指しているので，記号の割り当ても共通化さえできれば良いとする．我々が提案したいのは，プログラミング言語で例えればC言語等の機種非依存の高級言語に相当する部分であり，アセンブラに対応する実際のロボットの制御は個々の機種に任せるという立場である．

名称	方向	ID	方向	ID	動きの単位
上腕を回転	左	A	右	B	0.05×60
上腕を上下	下	C	上	D	0.05×60
肘を回転	右	E	左	F	0.05×60
下腕を上下	下	G	上	H	0.005×60
手首を捻る	右	I	左	J	0.05×60
手首を上下	下	K	上	L	0.05×60
右指を開閉	左	M	右	N	0.05×60
左指を開閉	左	O	右	P	0.05×60

上の表の最後のカラムの数値データはそのような“アセンブラ的”な部分を少しだけ示したもので，このモデルにおけるロボットの移動単位による数値であり，例えば1行目の基本動作はこれに対応する関節に対して0.05単位の移動を60回繰り返したものを割り当てているという意味であり，無次元である．そのため実機の場合にはここにmm等の単位付きの値を与えることになる[1]．

この辞書を用いて，ロボットに白い円柱を掴ませて赤いタブにはめることを手動で学習させた様子を以下に示す．使用した記号列は次のようであった（この表記はランレングス圧縮法により短縮できる.）：

AAAAACCCCCCMPMPMPCCHGGMPMPM
PMPMPAAAAAAAAAGGGCCGGGMGGGA

この系列を上記の辞書に従い，自然言語に直すと「上腕を左に回転し,...，上腕を下げ,...，指を閉じて,...，上腕を下げ,.....，上腕を左に回転する」という動きとなる．

[1]ロボットによっては力の単位ともなりうる．実際的な重力場の下では更に重力に逆らって腕を水平に保つためにも逆向きのトルクが必要になるがこれも機械依存の“アセンブラ”に委ねる．

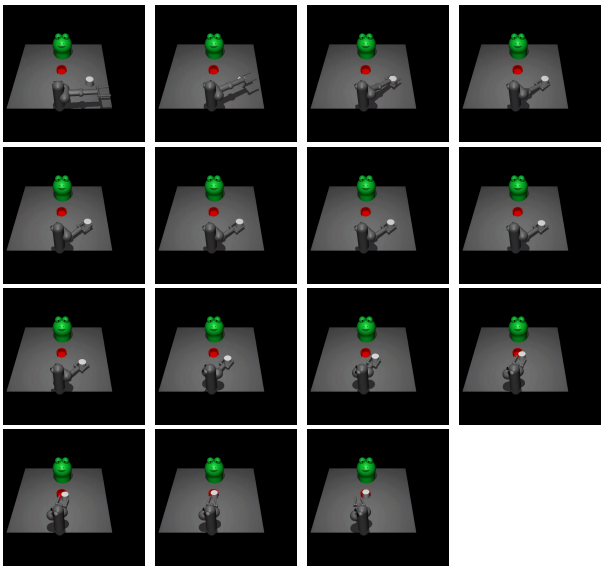

図3: 手動学習時

補足1. 今回の実験では用いていないが，この他にサブ辞書として記号を小文字に変えたものに×6の動きを割り当て,「少し」という形容詞付きの同じ動作を表現させて，より細かい動きが必要な場合に備えた．ただし，本研究の立場は工業用ロボットのようにミクロン単位の制御をしなくてもよい家庭内での使用に耐える程度のアバウトな作業の実現を目指しているので，最初に挙げた辞書で間に合わせることが理想である．

補足2. 動きの能率化のため，2個（以上）の動作を同時に行なわせることも許容できる．例えば上の手動制御の記号列の例に出てきた，2本の指を連動させて指を閉じるという動作を表しているMPは二つの関節を同時に作動させたほうが自然である．ロボットによっては2本の指が有るがこのようなシンクロナイズされた動きしかできないものもある．このような複合動作は関節のアクションベクトルの異なる成分に同時に値をセットすることで実現可能であり，上記テーブルの拡張として，例えば次のようなものを追加できる：

指を閉じる	Q	M and P
指を開く	R	N and O
腕を縮める	S	C and H
腕を伸ばす	T	D and G
肘を曲げる	U	B and F
肘を伸ばす	V	A and E

なお，実機や完全な物理シミュレータなどでは，慣性の影響で，上腕を動かすとその先の部品が予期しない方向に連られて動くことが有り，そのような場合には逆方向にトルクをかけて調整する必

要が生じるが，それは機械固有の処理に任せることとする．従って機械によっては実は複合動作の方が自然な基本動作になっているということが有り得る．

3 深層強化学習

本研究では，動作の獲得に強化学習を用いる．強化学習では，動作を行なう主体であるエージェント（ここではロボット）が毎回の試行（エピソード）ごとに，環境から状態を受け取り，その状態の下で行動を選択し，その結果として報酬を受け取るという動作を繰り返す．エージェントは各エピソードで貰える報酬の重み付き総和（累積報酬）を最大にすることを目標として行動し，その結果として目的の状態が達成されることを期待する．本研究で使用する深層強化学習とは，強化学習と深層学習を組み合わせたもので，強化学習の行動価値関数や政策を深層ニューラルネットで関数近似するものである．本研究では，Asynchronous Advantage Actor-Critic (A3C)[1] というアルゴリズムを用いて学習させることを試みた．A3C では強化学習の手法の一つである Actor-Critic というアルゴリズムを用いている．Actor-Critic では，エージェントが行動を選択する Actor と，その選択した行動を評価する Critic という構造を持っている．Actor と Critic をそれぞれニューラルネットで表現したものが A3C である．以下では ChainerRL 中の example として実装されていたものを本実験用に修正して使用した[2]．深層強化学習としてはこのアルゴリズムより以前にも，Deep Q-Network（DQN）[2] や Deep Deterministic Policy Gradient（DDPG）[3] といったアルゴリズムが知られており，一定の成果を出していたので本研究ではこれらを用いた学習も試みた．その結果，DQN はパラメータの選択が非常に難しく学習がうまく行かず，また DDPG は行動が連続値の場合のみしか扱えず，学習に多くの時間がかかってしまうため，最終的に本研究では離散化した action に対して A3C を用いて学習を行なうこととした．ロボットを家庭内で実用に供するためには，絶えず新しい仕事を学習させる必要があるが，それらを丸ごと連続的な深層強化学習で学ばせるのは家庭の環境資源的に無理があり，自然言語による離散化とそれによる仕事の表現はこのような要求に役立つとも考えられる．

[2] https://github.com/chainer/chainerrl

4 実験

記号化した基本動作の離散化 action による深層強化学習で，課されたタスクに対するロボットの動作概念の獲得を試みた例として，ロボットに「はめる」という動作を学習させてみた．図 3 のようにロボットが白い円柱を掴み，赤いタブまで持っていくという課題を深層強化学習で実現させる試みである．

4.1 使用するシミュレータとロボット

本研究では，MuJoCo[3][4] という物理シミュレータを使用している．またロボットアームは GitHub 上で公開されていた Pusher[4] を改変し 2 本の指を付け加えた ”Picker” を使用している．このロボットモデルは全部で 8 自由度を持ち，構成は図 4 のようになっている．

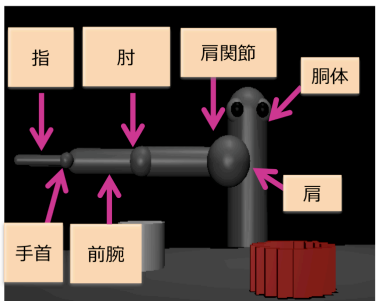

図 4: ロボットの構成図

4.2 実験設定

上記のロボットを OpenAIGym[5] というフリーの強化学習環境に組み込んで，chainerrl の上で深層強化学習を行なった．1 エピソードを 100 回の試行とし，全 80,000,000 エピソード学習させる．報酬については，ロボットの指 2 本，腕の軸と白い円柱の距離と，白い円柱と赤いタブの距離をそれぞれ計算し，係数を掛けて足し合わせたものとしている．

4.3 実験結果

以下に学習時の経過を示す．報酬関数の係数を様々な値に変化させて実験を行なったが，図のように白い円柱を挟まずに腕が当たって赤い枠の近くまで持っていくような動作を行なう様子が観察された．これ以外

[3] http://www.mujoco.org/
[4] https://github.com/openai/gym/pull/557
[5] https://github.com/openai/gym

にも円柱を掴まずに円柱の上に腕を置いた状態で止まってしまうものなども観察された．

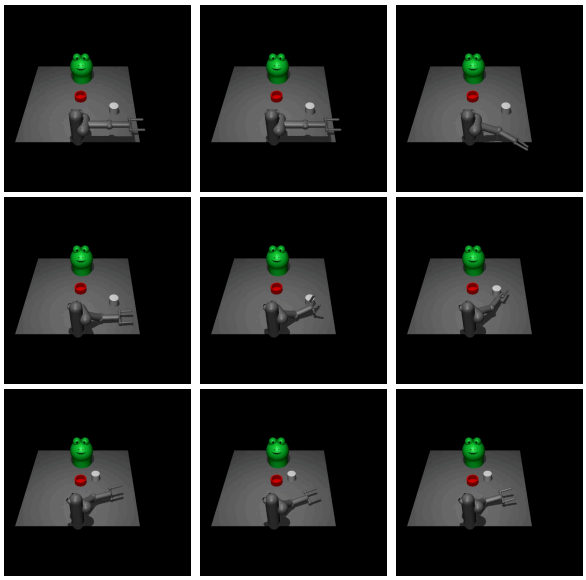

図 5: 学習経過図

4.4 考察と課題

報酬関数をいくつか変更して実験を行なったが，ロボットがうまくつかんではめるという目標の仕事を学習させることはできなかった．原因として，はめるを手動で学習させた際に複雑な記号列になったことから，深層強化学習で学習するにあたっても学習に時間がかかることが考えられる．また,「掴む」という動作は左の指を右側に，右側の指を左側に動かすという動作を行なわないとできない動作であるため，報酬関数をよりデリケートに設定する必要があると考えられる．なお，今後の課題としては，実用的には現在用いている幾何学的に計算された報酬関数に代わって画像から円柱やタブの位置関係を判断し，それに基づいた報酬計算を行うことで，これらの新しい位置関係への対応を自動化できるようにし，転移と汎化の能力を高めることを考えている．

5 まとめと展望

本研究では，動作は元来のロボットの物理的可動性よりは大まかな最小単位の動作を設定し，それらの組み合わせで実用的に必要な様々な仕事を表現できると考えた．実験ではこれら最小単位の動作と対応する言葉の辞書を用意し，動作の記号化を行なってロボットに動作概念獲得を試みた．これにより人間によるロボットの制御と，リアルタイムで必要となる強化学習の簡易化・高速化を目指した．今後は，我々の方式を用いてロボットが獲得した動作を他のロボットに伝える転移学習の検証や，タスクやジョブレベルでのロボット自身及びロボット間の動作の説明を目指したいと考えている．更に，ロボット動作の連続な空間での表現から離散的な言葉による表現にし，それを word embedding 操作で再び連続空間に戻すことで，複雑かつ機械依存の連続表現を互換性の有るものにし，同時に次元圧縮もできるであろうと考えている．

参考文献

[1] Volodymyr Mnih,Adrià Puigdoménech Badia,Mehdi Mirza, Alex Graves,Tim Harley,Timothy P. Lillicrap,David Silver,Koray Kavukcuoglu,“Asynchronous methods for deep reinforcement learning”, PMLR 48:1928-1937, 2016.

[2] Volodymyr Mnih,Koray Kavukcuoglu,David Silver,Daan Wierstra, Alex Graves,Ioannis Antonoglou and Martin Riedmiller,“Playing Atari with Deep Reinforcement Learning”,In NIPS Deep Learning Workshop. 2013.

[3] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, NicolasHeess, Tom Erez, Yuval Tassa, David Silver and Daan Wierstra,“Continuous control with deep reinforcement learning”, ICLR2016.

[4] Emanuel Todorov, Tom Erez and Yuval Tassa, “MuJoCo: A physics engine for model-based control” IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012.