

目標物到達機構を持つ自律的な小型ロボットの制御を行う BDI エージェントの構築

樽井 志織 兼松 明未 新出 尚之

我々は、実世界で自律的に目的を達成するロボットの実現を目指して研究を行っている。このようなロボットの実現には、人間と同様に目的達成の意図を保持してそれを実行しようとする、BDI エージェントを用いるのが適していると考えられる。そこで我々は、安価な小型ロボットを使用して、目的達成のプランを選んで実行する BDI エージェントを実現することを目指し、そのために必要な、プラン内で基本行為を行うためのライブラリの構築を行っている。本論文ではその 1 つとして、ロボットが目標物に到達するという行為を行う機能を学習によって実現することについて述べ、また、BDI エージェントがこの機能をプラン内で用いることによって、いくつかの経由地を経て最終的にいきたい場所に到達するという目的を達成する例を示す。

1 はじめに

近年、機械学習技術の発達につれて、ロボットに人間の持つ様々なスキルを習得させ、実世界で活用するという可能性が広がってきている。しかし、ロボットが何らかの目的を持ち、それを達成するためには、個々のスキルを基本行為として獲得するとともに、それらの基本行為を、目的を達成するための手段として周囲の環境に応じて適切に選んで実行する能力が必要となる。そこで我々は、機械学習によって獲得した能力を用いて実世界で行動するエージェントの構築を試みている。

先行研究 [4] [6] では、自律的に行動するエージェントを構築する足掛かりとして、安価なロボットを対象に、ロボットが活動する環境に動的に適応した移動能力を獲得できるよう、ロボットの車輪の部分をコントロールするモーターの動作を機械学習を利用して調整し、一定の距離あるいは一定の方向に移動するという基本行為を実現できるライブラリの構築を行った。しかし、何らかの目標物に到達する機能はまだ実現で

きていなかった。この機能は、実世界で活動するには必要とされる機会が多いと考えられる。

そこで本研究では、ロボットが画像認識を行い、学習の結果を利用して目標物を識別し、その目標物を探して接近するといった行動を取るための機能を実現し、これを基本行為として目的達成のためにプラン内で用いることを容易にするようなライブラリとして構築した。

2 学習機能

1 節で述べたように、目標物を認識するためのライブラリを構築するにあたって、学習機能を利用した。本研究で使用するロボットとして、実験の再現を行いやすい安価な小型ロボットを想定しているため、使用するライブラリに関しても利用のしやすさを重視し、機能を作成する際に使用するライブラリには OpenCV を使用した。また、言語については先行研究に合わせて Python を使用している。これの詳細については三節にて述べる。

本研究ではこの OpenCV がサポートしている学習機能の 1 つである `traincascade` を使用し学習を行った。目標物の写った画像である正解画像と目標物の写っていない非正解画像を収集し、正解画像内の目標物が存在する座標と個数を `traincascade` へ与える。

Developing BDI agent for controlling autonomous small robot with ability to reach target object.

Shiori Tarui, Akemi Kanematsu, Naoyuki Nide, 奈良女子大学人間文化研究科, Nara Women's University.

学習の方法は Haar-Like 特徴・LBP 特徴・HOG 特徴の 3 つから選択することが可能で、今回は Haar-Like 特徴を選択した。そして学習した結果である cascade ファイルを小型ロボットの制御プログラムで読み込み、目標物の認識を行うことが可能になる。

3 小型ロボットの制御

本節では研究にて使用した小型ロボットについて述べる。

3.1 小型ロボットについて

実験に使用するロボットには教育用 LEGO Mindstorms NXT (以下 NXT と記載) [2] を採用した。安価なロボットを使用することで、使える機会を増やすとともに、実験の再現を行いやすくするためである。また、従来研究で得られていた既存ライブラリの開発に NXT を用いていたことからの継続性も考慮している。

NXT は LEGO 社から販売されていたロボットで、頭脳の役割を果たしているインテリジェントブロックにさまざまな形や機能が備わったブロックを組み合わせ、自由にロボットの製作を行う事ができる。インテリジェントブロックには出力ポートが 3 つ、入力ポートが 4 つ存在していて、センサーやモーターの機能が備わっているブロックを接続することで、コンピュータから指示を送りロボットの操作を行ったり、ロボットが感知した環境の情報をコンピュータで受け取ったりする。また NXT の制御プログラムは、USB の接続または Bluetooth の通信でコンピュータから転送することが可能であり、本研究では USB による通信によって NXT の操作を行った。

なお本研究のロボットでは、NXT のインテリジェントブロックにロボットの移動のためのモーターを左右 1 つずつで計 2 つ接続している。また、NXT のセンサーとは別にカメラをロボットに装着し、それをコンピュータへ接続してロボットの目標物の認識を行っている。

3.2 NXT Python の拡張

NXT を制御するためのライブラリはいくつか存在している。我々は、メモリ管理が不要であること、GUI の設計が容易であることなどをはじめ、開発が容易であることを考慮し、開発言語として Python を選択したため、Python を使用したライブラリである NXT Python を選択した。しかしこのライブラリは、モーターの回転数を直接制御するといった下位レベルの制御命令が基になっており、複雑な行動をロボットに行わせようとすると、下位レベルの制御命令をいくつも組み合わせてプログラムを記述する必要があった。このようにプログラムを記述すると、上位レベルの制御命令でプログラムを記述するよりもプログラムが長くなり、記述ミスが増加してしまうといったことや、改良を行い辛いといった欠点となる。この欠点を改善するため、先行研究[5]ではライブラリの拡張が行われた。この拡張されたライブラリは NXT Python+ と名前が付けられ、下位レベルの制御命令を組み合わせで作成された上位レベルの制御命令を使用することや、構築された学習機能によって環境やロボット毎にモーターのパワーの大きさを学習することなどが可能である。また、プログラミングに慣れない者であってもロボットの操作が行えるように、スライダやボタンで視覚的にロボットのポートやモーター出力の大きさなどの設定を行うことができる GUI も備えている。

3.3 拡張ライブラリと ROS によるロボットの制御

自律的に目標を達成するエージェントの構築にあたり、ロボットの制御部の開発を容易にするため、本研究ではロボット開発用のフレームワークである ROS (Robot Operating System) [3] を使用して、NXT Python+ を利用するエージェントの構築に取り組んだ。ROS はオープンソースで提供されているロボット製作のためのオペレーティングシステムで、構築したプログラムの再利用を支援することを目的としている。ROS では、ソフトウェアの構造の一単位であるパッケージが存在し、その中のライブラリの実行ファイルなどをノードと呼んでいて、このノード同士が

ROS クライアントライブラリによって通信を行うことによって、異なるノードがお互いにやりとりしながら協調して動作を行う。これにより、ロボットの各機能を独立したプログラムとして作成して、それらを結合することによる開発を可能としている。

この ROS を使ったエージェント部と NXT Python+ の仲介および実装については 5 節にて述べる。

4 BDI エージェントの構築

自律的に目的を達成する小型ロボットの自律エージェント構築にあたって、ロボットを制御する側のライブラリとは異なり、よりエージェントの構築に適したプラットフォームである Jason [1] を使用した。これは BDI エージェント構築のためのプログラミング言語である AgentSpeak のインタプリタである。BDI エージェントとは Belief (信念), Desire (願望), Intention (意図) の 3 つの心理状態を使って行動の決定を行うエージェントである。エージェントはまず、信念と願望から目標を導きだし、この目標を達成させる手段であるプランを選択してこれを意図とする。次にこの意図つまり手段を実行して、実行後のエージェント外部の変化を知覚し、その知覚をもとにして再び信念と願望から目標を導き出す。このサイクルによって目的を達成するのが BDI エージェントである。Jason はこの AgentSpeak を拡張した言語のインタプリタであり、Prolog に似た形でエージェントが目的を達成するためのプランを記述する。さらに、エージェントが置かれている外部環境を表すプログラムを Java で実装すれば、環境とエージェントが相互作用することで知覚を得たり行為を実行したりすることが可能となる。われわれはこれを用いて、エージェントが我々のライブラリで実現される基本行為をプラン内で用いることを可能とし、プランをエージェントが選択することにより、ロボットが環境に応じて柔軟な行動をとることを可能とする。

5 実装

3 節にて述べた小型ロボットを制御する機能を、4 節にて述べた BDI エージェントにて制御する。エー

ジェント側の制御プログラムを jason フォルダ、小型ロボット側の制御プログラムを scripts フォルダとし、jason_ros という名前の ROS パッケージを作成した。なおこの scripts フォルダは ROS の実行可能スクリプトを格納する scripts フォルダと同一のものである。jason フォルダ内には、BDI エージェントの環境プログラムである mas2j ファイルとゴール・プランのプログラムである asl ファイルがあり、これらは Jason によって用いられる。scripts フォルダには主なプログラムとして、Jason と通信を行うプログラムである tcp と、目標物の検出を行うプログラムである haar_like、そして NXT の制御・通信を行うプログラムである nxt の 3 つが格納されている。これらは ROS でのノードにあたり、パッケージである jason_ros を動かせば 3 つのノードが動きだし、3 節後半で述べたよう動作を行う。これら 3 つのノードは ROS クライアントライブラリによって通信を行うが、エージェント側のプログラムとはクライアントライブラリで通信を行わないので、ロボット側の通信はノードのうちの一つが TCP を介して通信を行っている (図 1)

Jason と ROS が起動されると、まず nxt ノードが NXT と ROS の通信を行う。NXT と通信が可能であれば接続した後、この情報を tcp ノードへと送信し、これを受け取った tcp ノードはエージェント側へと状況を伝える。接続が完了すると、次にエージェント側から到達すべき目標物の情報が tcp ノードに渡される。これが haar_like に伝わると、haar_like はまず目標物の cascade ファイルが存在しているかどうかを確認し、これが確認できれば tcp ノードを通してエージェントにファイルの有無が伝えられる。ファイルが存在し、目標物の認識が可能な場合、haar_like と nxt は目標物の認識、出来ない場合は検索などを行い、目標物に対して何らかの行動を取ることが出来る。なお本研究では、目標物を認識した後一定距離まで接近するという行動を目標としてプログラムを構築した。NXT が設定された目標を達成し、または達成が不可能であることが認識された場合、この情報がまた tcp ノードを介してエージェントに伝えられ、伝えられた情報を基にエージェントの環境が更新され、更新され

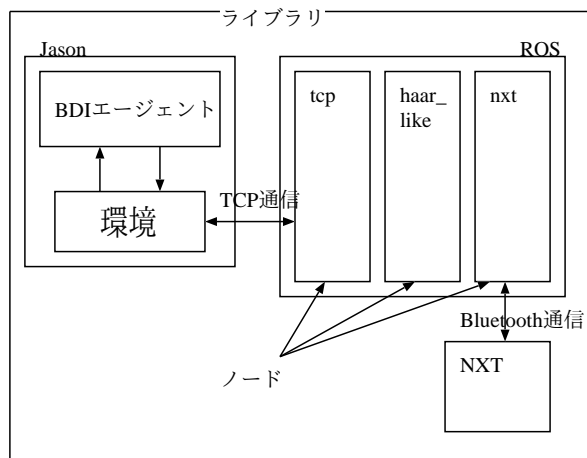


図1 Jason および ROS による実装

た環境から次の目標が設定される。

6 検証

6.1 検証内容

構築したエージェントが目標物に対し、自律的な行動を行うことが可能であるか、実際に目標物と、達成すべき目的を設定して検証を行った。目標物は市販の熊のぬいぐるみ、カエルのぬいぐるみなどを設定し、目標物が写った正解画像と目標物が写っていない不正解画像を各 2000 枚程度収集した。前述の通り学習には Haar-Like 特徴量を使用し、あらかじめ cascade ファイルの作成を行った。また、エージェントの達成すべき目的には、認識させたぬいぐるみを検索、認識し、距離が離れている場合には一定距離まで接近するという行動を設定した。なお、認識させる目標物は複数設定することも可能で、この場合 1 番目に指定された目標物に接近した後、そこから 2 番目に指定された目標物を検索して接近することで目的達成となる。

まず、障害物のない空間に NXT と目標物であるぬいぐるみを設置する。条件を 1. 目標物は 1 つで視界に入っている場合、2. 空間に存在しているが視界に入っていない場合、3. 目標物が 2 つでどちらも空間に存在している場合、の 3 つの条件に分けて検証を行った。

また他に目標物の認識率に関する実験も行った。この実験の結果については発表時に述べる。

6.1.1 条件 1

NXT の視界内に目標物を設置した場合、初めから目標物を認識しているので、NXT は目標物を検索することはなく、認識した目標物へと接近した。

6.1.2 条件 2

NXT の視界外に目標物を設置した場合、初めは目標物を認識できないため、まず回転の動作で周囲を確認して、目標物を検索する行動から開始する。その後、視界の外に置かれていた目標物を発見することに成功し、目標物へと接近した。

6.1.3 条件 3

上記の 1 と 2 の目標物に加えて、新たに 1 つ目標物を追加した。まずエージェントは 1 つ目の目標物への接近を達成すべき目標とし、NXT に信号を送る。NXT は 1 つ目の目標物が周囲に存在しているかどうかを確認し、目標物を認識した後に 1 つ目の目標物に接近した。1 つ目の目標物を達成すれば、次はエージェントの指示を受けて 2 つ目の目標物の検索を開始する。1 つ目と同じく周囲を確認し、2 つ目の目標物を発見してから問題なく接近することが出来た。

7 まとめ

本研究において構築された BDI エージェントは、達成すべき目的と外部の環境を考慮し、あらかじめ記述されているプランライブラリから適切な手段を選んで、自律的に目的を達成することが出来た。今回達

成すべき目的には、単純に目標物に接近する事のみを設定したが、これをプログラマが自らの望ましい目的へと組み直すことにより、より有意義にエージェントを活用することが可能である。このような今後の発展として、本研究で検証した行動の他にも、1つ目の目標物が見つからなかった場合におけるプランや、目標物搜索の際に障害物が存在していると想定したプランなどの構築が挙げられる。また、本研究では機械学習に Haar-Like 特徴量を収集する方法を使用した。他の特徴量を使用してエージェント構築を行うことや、他の学習方法を導入することで、より効率よく自律的に目的を達成するエージェントを構築出来る可能性も挙げられる。

また本研究の課題として、エージェントを稼働させるコンピュータによってはエージェントのロボット制御の指示と目標物の認識のタイミングが合わず、NXT の動作が想定とずれてしまうといった問題点などが存在する。今後これらの課題を解決し、新たなプ

ランの拡張、学習方法の検討などを通じて、更なるライブラリの発展を行っていく。

謝辞 本研究では、奈良女子大学理学部学生会等参加支援経費の支援を受けた。

参考文献

- [1] H.Bordini, R., Hübner, J. F., and Wooldridge, M.: *programming multi-agent systems in AgentSpeak using Jason (Wiley Series in Agent Technology)*, Wiley Interscience, 2007.
- [2] Lego Group: LEGO.com エデュケーションオフィシャルサイト, <http://education.lego.com/>.
- [3] Open Source Robotics Foundation: ROS.org, <http://www.ros.org/>.
- [4] 江川鈴菜: 強化学習による環境別のロボットの基本行為の獲得について, 2014 年度卒業論文, 奈良女子大学理学部情報科学科, 2015.
- [5] 小島侑子: 小型ロボット制御のための汎用ライブラリの構築, 2011 年度修士論文, 奈良女子大学大学院人間文化研究科, 2012.
- [6] 樽井志織: 小型ロボット制御のための汎用ライブラリの拡張, 2014 年度卒業論文, 奈良女子大学理学部情報科学科, 2015.