

動的環境下におけるロボットの行動プラン最小修正について

The minimum plan modification of robot's actions under the dynamical environment

藤田 恵*¹ 後藤 勇樹*² 新出 尚之*³ 佐藤 健*⁴ 細部 博史*⁵
Megumi Fujita Yuki Goto Naoyuki Nide Ken Satoh Hiroshi Hosobe

*¹奈良女子大学大学院人間文化研究科

Graduate School of Humanities and Sciences, Nara Women's University

*²京都大学数理解析研究所

Research Institute for Mathematical Sciences, Kyoto University

*³奈良女子大学大学院自然科学系情報科学領域

Faculty, Division of Natural Sciences, Nara Women's University

*⁴国立情報学研究所情報学プリンシプル研究系

Principles of Informatics Research Division, National Institute of Informatics

*⁵法政大学情報科学部デジタルメディア学科

Faculty of Computer and Information Sciences, Hosei University

In this paper, we propose a method to determine appropriate robot's behavior in a high-level manner using Prolog, while each command in the Prolog program works with conjunction with the robot's basic actions under the real world. Since robot's basic actions are implemented considering reflective action skills, robot's behavior is sophisticated and robot can move naturally. Using this method, we can control robot's action in real-time. Our method is expected for robot to take appropriate actions in the real world and to be suitable for autonomous intelligent robot control.

1. はじめに

本研究では、外部からの人による遠隔操作 [ISHIGURO 10] やパラメータ調整とハードコーディング [中岡 11] によるロボットの制御ではなく、ロボット単体が推論を行い、ロボット自身が行動決定を行うといった、完全な自律的制御を目指している。

我々のこれまでの研究 [藤田 12] では、高位レベルにおける行動決定を行うエージェントを用いたが、このエージェントはすでに外界に対してグリッドで区分けされたマップを保持しており、この区分けされた升目を中心に行動を行うという動作制御を行ってきた。しかし、これではロボットの行動による位置的誤差が生じた場合、ロボットの行動による位置情報と実世界での位置が一致なくなる問題が生じてしまう。従って今回は、グリッドマップを用意することなく、ロボットが向いている方角を元に行動を決め、初期位置とモータの出力の情報を元に自己位置の推定を行えるようにすることで、行動している空間の位置情報を推測するようにした。

これを利用して、今回の研究では、ロボットが柔軟な行動を行うことができるよう、基本行為の中で外界における障害物への衝突などを対処することによって、高位レベルの制御における自律的制御が円滑に行えることを目指す。

2. 実世界における行動計画

ロボットを実世界で自律的に支障なく動かすには、例えば高位レベルな概念を扱う動的環境に対応したプランナ

連絡先: 藤田恵, 奈良女子大学大学院人間文化研究科,
奈良市北魚屋西町, 0742-20-3555, 0742-20-3555,
saboten@ics.nara-wu.ac.jp

[Ayan 07][Sánchez-Garzón 11] を扱いロボットを制御するだけでは、実際にロボット自体の行動による位置的誤差が生じた場合、適切な制御はできない。

従って、ロボットの実世界での自律的制御を行うには、これに加えてロボット自体の基本行為と高位レベルの制御との連動をとる必要がある。今回我々は、ロボットの基本行為を反射的行動を取り入れた形で実装し、この基本行為の単位とロボットを高位レベルで制御する行動命令の単位とをうまく連動させることにより、実世界でロボットが支障なく自律的に動作できることを目的とする手法を提案する。

今回のロボットの行動決定において、我々は Prolog を使用し高位レベル命令を下すよう実装した。具体的には、Prolog をプランライブラリとして用いて、再帰的な行動計画を立てる事ができることにより、繰り返し参照するものを事実としてプログラム中に追加する形でより柔軟な制御を可能にしている。また、各プログラム節は宣言的に記述しており、プログラムとしての読みやすさも持っている。Prolog を利用する場合、バックトラックによる別解探索が問題となる。しかし、今回のように現実世界で動作するものを記述する場合、そもそもバックトラックをさせるべきではない。なぜなら、時間は過去から未来へと非対称的に流れて行くため、一度行った基本行為を無かった事には出来ないためである。その為、今回は別解探索を行わない制御をしている。単一化をする変数は、全てセンサ情報の取得の為に用いており、基本行為が失敗するという概念をそもそも持たない。別解探索を行わない為に、プログラムの実行は一本道になっていて、現実世界でのロボットの行動と連動してくれるのも強みである。



図 1: Q.bo Pro Evo

3. ロボットの説明

今回 TheCorpora 社において販売されている、Q.bo Pro Evo というロボットを使用した。Q.bo は開発環境を Ubuntu Linux 上で構築しており、図 1 のようなロボットである。

移動方法は前方のキャスターと後方の 2 つのモータつき車輪で動くようになっている。また、音声認識や発話を行うことも可能であり、画像認識も可能である。これらのコンポーネントは ROS[ROS 13] と呼ばれるロボット用のオープンソースによるオペレーティングシステムを使用することによって制御している。

4. ロボットの制御

ロボットの制御として、ロボットの基本行為を行動決定プログラムで使用し、随時ロボットの行動を管理するシステムを構築した。

ロボットの基本行為は Python で記述し、行動決定プログラムは Prolog で実装し、これらを TCP 通信をすることにより、ロボットの側からはロボットが現在向いている方向と初期位置を元にした x 座標と y 座標を返し、行動決定プログラムの方からはロボットから得た情報を元にして、必要な高位レベルな行動命令を返すことにする。

4.1 ロボットの基本行為

Q.bo を動かすには、ROS のサービスを使用して制御プログラムを作成することが必要となる。今回は移動計画を中心に基本行為を作成していくため、前方に障害物があるかどうかを確認する超音波センサーによる距離判定をする ROS サービスと、車輪を回転させて移動する ROS サービスと、ジャイロセンサーを利用したロボットの車体の方角を表示させる ROS サービス、車体の速度を測定する ROS サービスを使用し、Python で基本行為を制御するプログラムを作成した。

基本行為は以下のように作成した。基本行為を作成する上で、それぞれの基本行為はある程度反射的な行動を含む形で実装した。例えば、`forward_Qbo(Direction)` は `Direction` の方向に向けて決められた距離を移動し、障害物が存在するとその決められた距離の移動を中止して停止する。この手法は低レイヤで反射的行動を行わせるという点でサブサンクションアーキテクチャ[Brooks 86] と類似した設計になっている。

- 障害物を避けて前進する (`search_Qbo(Direction)`)
 - Q.bo は自身を回転させて、`Direction` の方角を元に障害物のない方向を探しだし、前進する
- 前進する (`forward_Qbo(Direction)`)
 - Q.bo は障害物を検知しない限り、決められた距離を `Direction` の方向に前進する
 - Q.bo は障害物を検知すると、決められた距離に関係なく停止し、次の指示を待つ
- Q.bo の車体の角度補正 (`correction_QBO(Initial_Direction)`)
 - 進むべき方角 `initial_Direction` の方向にジャイロセンサーを元に Q.bo を回転して修正する

4.2 ロボットの行動決定

4.1 節で述べた基本行為を行動決定プログラムが Q.bo に命令として伝達し、Q.bo は動作する。

行動決定プログラムは 4. 節の最初で述べたように Prolog で実装しており、Prolog における解導出をおこなって推論した結果、Q.bo がどのような行動をとることが必要かを返す仕組みになっている。

今回は障害物のある平らな道を Q.bo が一定の方角を向いて前進し、うまく障害物を避けて前進できるかというシナリオで Prolog プログラムを作成した。

Prolog プログラムの概要は次の通りである。

1. ロボットの方から行動開始時の初期位置に関する情報を得る
2. ロボットから得た情報を元にロボットが前進するための解導出をする
3. ロボットに解導出をした結果の解を伝達する
 - 以下のような解を行動プランとして伝達
 - 障害物がある場合は障害物を避けて前進
 - 障害物がない場合は前進
 - Q.bo の車体が移動による誤差で傾いた場合、補正
 - Q.bo が反対方向に向いてしまった場合、停止
4. ロボットが解 (行動プラン) を実行し、外部に対して知覚をした結果を返す

2~4 を繰り返すことにより、移動する。

5. 実験

4. 節で述べた、基本行為と行動決定プログラムを利用して次のような実験を行った。

Q.bo を障害物を設置した場所で初期位置で向いていた方角に前進させ、障害物を実際に避けて行動可能かどうかを検証した (図 2)。実際に動作させた結果、図 3 のような結果となり、Q.bo は障害物の前で停止し、障害物を避ける方角を探しだし、その方角へ前進した後に元の進むべき方角へ車体を修正し、前進した。



図 2: 実験の様子

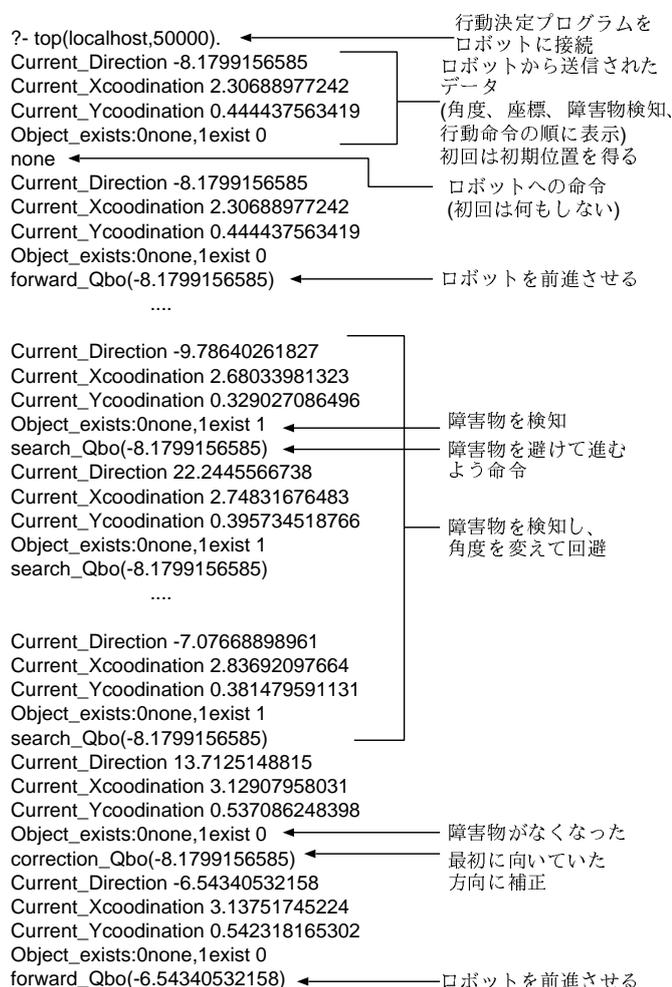


図 3: 行動決定プログラム実行結果

6. 考察

今回実装したロボットの基本行為は Prolog による高位命令に対して、実世界上で柔軟にロボットを行動させることができたので、それぞれのレベルの命令の単位の同期がとれているこ

とがリアルタイムにロボットを制御する上で重要であることが判明した。

また、Prolog で行動決定のプラン制御を扱っているので、適切なロボットの基本行為を増やす (画像認識など) ことによって、それを Prolog の規則の中に設けることで、より複雑な行動をロボットにさせることが期待される。

このように、行動の性質に対して考察を進め、プランでのゴールの数を最小限になるよう記述した点と、基本行為の中に反射的行動に近いものを記述し、現実世界で起きうる様々な問題に対し、その場その場を対処することができるようにした点はロボットの自律的な行動制御に必要であると思われる。

7. 関連研究との比較

センサと振る舞いモジュールの階層構造からなるロボットアーキテクチャとしては、先に述べた Brooks のサブサンクションアーキテクチャ [Brooks 86] がある。これは、複数の目標を達成するために、多数のセンサに反応し、頑健かつ漸進的に拡張可能なロボットを容易に設計できる方法論であり、以下の特徴を持つ。

1. タスクを階層的なレイヤに分け、下位のレイヤが構築されれば、それを変更することなく、上位のレイヤを構築できる。
2. 内部処理をほとんど必要としないような単純なセンサと行動の結合によって、環境と実時間での相互作用ができる。
3. 知能は中央集権化されたものではなく、複数の緩やかに結合されたプロセスから知的振る舞いが創発されると考える。

今回の我々の手法もこのような下位のレベルでの制御が基本行為が実行されている間働いており、類似した点もある。しかし、[Brooks 86] は上位のレイヤにおけるタスクは決定的であり、複数のタスクを取り扱う機構はない。我々の手法では、Prolog プログラムの規則として、さらにプログラムを拡張することにより、複数のタスクを取り扱う機構を設けることが可能であると考えられる。

8. まとめ

本論文では、ロボットの制御に対して、適切な基本行為と行動決定による高位命令のそれぞれの単位行動の連動がとれているという点が、ロボットの自律的制御に関して非常に重要であることを述べた。

今後はさらに実世界に適応した複雑な基本行為を取り入れて、このシステムがどのように構築することによってより有効になるのかという点に着目することが必要である。

本研究は奈良女子大学若手女性研究者支援経費の助成を受けたものである。

参考文献

- [Ayan 07] Ayan, N. F., Kuter, U., Yaman, F., and Goldman, R.: HOTRiDE: HierarchicalOrdered Task Replanning in Dynamic Environments, in *Principles and Practices for Planning in Execution*, pp. 31–36 (2007)
- [Brooks 86] Brooks, R. A.: A Robust Layered Control System for a Mobile Robot, *IEEE Journal of Robotics and Automation*, Vol. RA-2, No. 1, pp. 14–23 (1986)

[ISHIGURO 10] ISHIGURO, H.: Studies on Humanoids, *Journal of the Japan Society for Precision Engineering*, Vol. 76, No. 1, pp. 20–23 (2010)

[ROS 13] ROS.org, <http://www.ros.org/wiki/ja> (2013)

[Sánchez-Garzón 11] Sánchez-Garzón, I., Fdez-Olivares, J., and Castillo, L.: Monitoring, Repair and Replanning Techniques to support Exception Handling in HTN-based Therapy Planning Systems, in *International Workshop on Artificial Intelligence in Healthcare and Biomedical Applications (CAEPIA 2011)* (2011)

[中岡 11] 中岡 慎一郎, 三浦 郁奈子, 森澤 光晴, 金広 文男, 金子 健二, 梶田 秀司, 横井 一仁: ヒューマノイドロボットのコンテンツ技術化に向けて — クリエイターによる多様な表現の創出が可能な二足歩行ヒューマノイドロボットの実現 —, *Synthesiology*, Vol. 4, No. 2, pp. 80–91 (2011)

[藤田 12] 藤田 恵, 片山 寛子, 新出 尚之, 高田 司郎: 実世界の多様性に適応した BDI ロボットについて, *情報処理学会論文誌数理モデル化と応用*, Vol. 5, No. 1, pp. 50–64 (2012)