

自然言語によるロボットプログラミング

板谷 純希† 中村 友昭† 長井 隆行†

†電気通信大学大学院電気通信学研究科

1 はじめに

一般に、ロボットの行動はあらかじめプログラムされている必要があり、プログラムされていない行動を行うことはできない。全ての行動を事前にプログラムしておくことは困難であるため、実際の動作環境に応じてプログラミングを行う必要がある。しかし、ロボットのユーザーが必ずしもプログラミングの知識を持っているとは限らず、特に家庭用ロボットの普及を想定した場合、ユーザーが一般のコンピュータ言語を用いてプログラミングすることは困難である。従って、ロボットが人間からの自然言語による命令や説明を柔軟に理解し、自らが実行できるプログラムへと変換する能力が必要であると思われる。

そこで本稿では、ユーザーの言語による命令を解析し、それをロボットが実行できる形へ自動的に変換する枠組みを提案する。ここでは特に、変換の基本となる部分を実際のロボットに実装し、ロボットに関する知識の全くないユーザーによるタスクの説明を、どの程度自動的にプログラムに変換し実行できるかを評価する。

関連研究としては、文献 [1] の走行プログラムの自動生成が挙げられる。しかしこれは、GUI による操作のみを対象としており、入力された走行経路図からの走行プログラム生成のみを対象としている。文献 [2]–[3] では、ヒューマノイドロボットの自動プログラミングを扱っているが、基本的にはユーザーの実演をいかにロボットで実行するかが主眼である。また、ロボットによる言語理解の研究も多く存在するが、対話システムにおける応答生成など、コミュニケーションが主眼であり、複雑なタスクを行うようなものは少ない。一方、ロボットによる言語獲得や模倣学習の研究も進んでおり、本質的な問題はこうしたアプローチにより解決される可能性がある。しかし、現段階ではまだ実用的であるとはいえず、また実用上はある程度の作りこみと学習のハイブリッドな手法が現実的であると思われる。さらに、CG を自然言語から自動生成する研究も行われている [4]。アプローチとしては非常に近いものの、実世界における物理的な動作と CG とでは本質的な違いがある。

2 ロボットシステム

2.1 モジュール構成

本稿で想定するロボットのソフトウェアは、モジュール群として実装される。これは、RTM などのミドルウェアを用いたネットワーク構造で実現可能である。本稿で用いるロボットでは、DiGOROnet と呼ぶ独自のミドルウェアを用いる。これは黑板モデルをベースとしており、モジュールの接続・切断が比較的柔軟に行えるのが特徴である。ロボットをプログラミングするためのソフトウェアも、一つのモジュールとして実装される。図 1 に、ロボット内のモジュール構成の概要を示す。Natural Language Robot Programming (NLRP) モジュールにおいてユーザの指示に基づいたスクリプトが自動生成され、そのスクリプトをスクリプト実行モジュールで実行する。

2.2 スクリプト実行モジュール

ロボットの行動は、基本的な動作をあらかじめ DLL として用意しておき、それを組み合わせて順に実行す

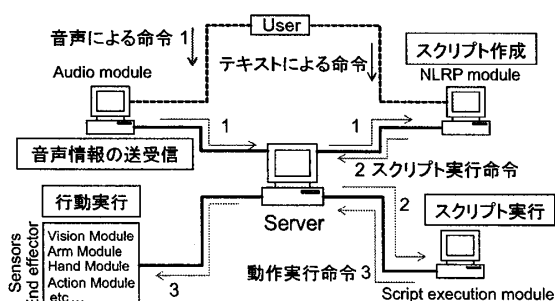


図1 ロボットシステムの概要

```

[ dll ]
1 HandLookOnDesk.dll, //机の上を見る
2 VisionAddrRedObjectTarget.dll, //赤い物の座標を記憶
3 赤い物を見つけました, //発話内容
4 HandOpen.dll, //手を開く
5 ArmMoveObjectTarget.dll, //腕を記憶している座標へ動かす
6 HandClose.dll, //手を閉じる
  
```

図2 スクリプトの例

ることで実現する。行動プログラムは、図2のようなDLLの名前が順に書いてあるスクリプトとなり、書かれているDLLを順番に呼び出すことで実行していく。基本的な動作のDLLは、設計者が事前に設計することになるが、動作学習によって新たに学習させることも可能である。本稿で実験に用いたロボットは基本動作として、「アームの移動」や「手の開閉」などが用意されている。また、基本動作を複数組み合わせた「物体を掴む」などの上位行動も複数用意されている。

3 言語からのスクリプト自動生成

3.1 提案システムの概要

前節のシステムを用いることにより、自然言語によるロボットプログラミングは、いかに自然言語からスクリプトを自動的に作成するかという問題になる。図1におけるNLRPモジュールがこの仕事を行うことになるが、基本的な考え方は、用意した基本動作や上位行動に単語（動詞）を割り当てておき、ユーザーからの命令文を構文解析することで、動詞に対応したDLLを並べていくというものである。この際、主語や述語となる名詞は動作DLLの引数となるため、センサ情報処理結果を参照することで、その存在の有無や位置などの情報に変換する。形容詞は、係り受け解析の結果をもとに、名詞とセンサ情報のマッチングなどに利用される。自然言語を用いて命令する場合の大きな問題は、命令文の曖昧性である。最終的には、背景知識の獲得や経験からの学習、対話による曖昧性の解消といった機能を実現する必要がある。図3に構想の全体像を示すが、本稿ではその最初のステップとして、プログラミングの中核となる構文解析に基づくスクリプトの自動生成のみを考える。

3.2 基本処理

3.2.1 名詞の処理

名詞処理の基本は、その対象が空間的にどの位置に存在するのかを特定することである。これは、物体認識などセンサ情報処理結果との比較によって実現される。従って、命令文の形態素解析によって得られた名

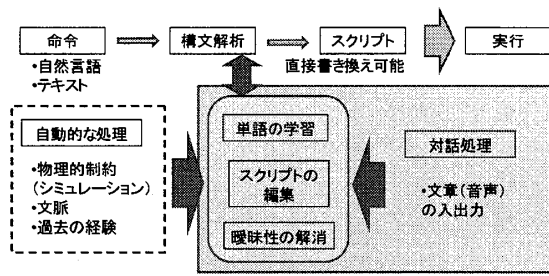


図3 自然言語によるロボットプログラミングの全体像

詞は、物体特定 DLL に形容詞、センサ情報とともに渡され、最終的には認識結果が出力される。

3.2.2 動詞の処理

動詞は、対応する基本動作や上位行動を選択し、実行時の引数として物体特定 DLL によって処理された名詞の情報を受け取る。ここで問題となるのは、例えば「置く」のように手に持っていないければ実行できない暗黙の前提行動が存在する場合である。この前提行動は、直接的に指示される場合と省略される場合があるため、動作 DLL 自体がこれを判断し、必要に応じて前提行動に対応する基本動作を追加する。

3.2.3 制御構造

条件分岐

複雑な行動をさせる場合、行動の結果によって次の動作を決定する条件分岐が必要となる。ユーザの明示的な条件分岐命令は、「～ならば」や「～たら」といった仮定形の助動詞、もしくは「～時は」や「～場合は」のような副詞可能名詞と係助詞に注目することで判断できる。

ループ

ロボットに同じ動作を繰り返し実行させたい場合、回数を指定する場合が多い。ここでは、動詞に数を表す名詞と「個」などの助数詞が係っていた場合に、その数だけ動作を繰り返すことでループを実現する。

並列処理

並列処理は、順次命令されたものを自動的に並列化することで実行時間の短縮を図るものであり、ここでは確認発話など事前に設定した特定のものを並列化する。また、ユーザの指示自体が並列性を持っている場合は、それらの物理的な行動可能性をチェックした上で並列化する。ただし、こうした指示の並列性は、接続助詞の「ながら」が文章内の二つの動詞をつなげているといった構文解析によって判断される。

3.3 未知語の学習

動詞と基本動作の組を全て事前に用意することは難しいため、基本動作を組み合わせた新たな行動に動詞を結びつけることで未知語を学習させる。例えば、「持ち上げる」を学習する場合、「掴む」動作を行った後に「腕を上げる」動作を実行させ、「持ち上げる」という言葉と「学習」という言葉を入力することで、直前に実行された動作のつながりと「持ち上げる」という言葉が結び付けられる。

3.4 スクリプト編集

生成したスクリプトは、直接書き換えることが可能であるが、文章の命令による編集も可能である。現段階では、ある命令の前に何か他の命令を挿入するといった簡単な編集のみを実装している。

4 実験

実験に用いたロボット（図4）は、6自由度のアーム、4自由度のハンド、赤外線カメラと2台の CCD カメラ、触覚アレーセンサ、マイクロフォンから構成されてい

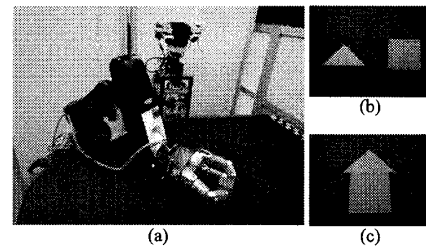


図4 実験で用いたロボットとタスクの例

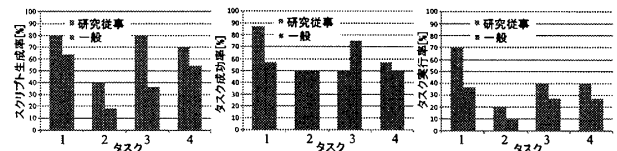


図5 実験結果 (a) スクリプトの生成率 (b) 生成されたスクリプトのタスク成功率 (c) 全体でのタスクの成功率

る。カメラはパンチルト台に載っており、周囲を見まわし物体の色情報と位置情報、人の顔検出などを行うことができる。また、カメラで検出した物体を把持することが可能である。実際に文章からプログラムが生成できるかを評価するために、4種類のタスクを数人のユーザに命令してもらい、スクリプトの生成率とタスクの成功率を評価した。被験者は、ロボットについて知識のある研究従事者 10 名と全く知識のない一般人 11 名（男性 5 名、女性 6 名）である。

一つ目のタスクは、図4(b)のようにロボットの目の前に置かれている物体を図4(c)のような状態にする、というものである。これは物体の位置指定を理解できるかどうかを調査しており、実際に「青い物体を赤い物体の手前に置いて」のような命令が行われた。二つ目は、状況によって行動が変わるタスクであり、例えば「赤い物体があれば青い物体の左に置き、黄色い物体があれば青い物体の右に置き」のような指示を出されたときにその通りに行動できるかを評価する。三つ目は「赤い物体を2つ私に渡して」のような同じ動作を繰り返す事を指示する命令が出されるタスクを行い、四つ目は「目を閉じながら手を振って」のような二つの動作を同時に実行しなければならないタスクを行った。実験結果を図5に示す。タスク2のスクリプト生成率が低いのは、これは複数の動作が必要な複雑な状況では多くの人が動詞を省略してしまい、ロボットが命令を解析できなかったことが原因であった。また、ロボットに関する知識の有無で結果を比較すると、知識の有る人が命令した場合の方がスクリプトの生成率やタスクの成功率が高いことが分かる。

5 まとめ

本稿では、ロボットのプログラミングを容易にする枠組みを提案した。実際に、基本となるモジュールをロボット上に実装し、簡単な評価を行った。今後は、対話機能や曖昧性の解消などより高度なプログラム生成機構を検討する予定である。

参考文献

- [1] 青山ほか, “次世代ロボット知能化技術開発プロジェクト RTM 化におけるプログラム開発効率と品質の向上”, 第 27 回日本ロボット学会学術講演会, 2D2-06, 2009
- [2] S.Knoop *et al.*, “Automatic robot programming from learned abstract task knowledge”, in Proc. of IROS07, 2007
- [3] M.Pardowitz *et al.*, “Learning Sequential Constraints of Tasks from User Demonstrations”, in Proc. of HUMANOID2005, 2005
- [4] M.Oshita, “Generating Animation from Natural Language Texts and Framework of Motion Database”, in Proc. of CW2009, 2009