

# インタラクティブな学習によるロボットの行動命令に含まれる不明確性の解消 Solving Vagueness in Robot-controlling Commands by Interactive Learning

徳永 陽<sup>†</sup>  
Nodoka Tokunaga

徳永 健伸<sup>†</sup>  
Takenobu Tokunaga

田中 穂積<sup>†</sup>  
Hozumi Tanaka

Generally, a word has various interpretations. This phenomenon is called “vagueness”. This paper shows that vagueness in robot-controlling commands can be solved by interactive learning. A robot runs along a crank-shaped course according to user commands. When a command is input, the robot acts based on a interpretation of the command. In cases when the interpretation is not appropriate, adjustment commands are necessary. Through repetition, the appropriate action to each controlling-command is learned, i.e., vagueness is solved. User’s commands, robot’s position and appropriate actions are memorized and this information reused in experiments on different courses from the one with a good performance level.

## 1 はじめに

我々が日常使う会話の言葉には、多くの不明確性 (vagueness)<sup>1</sup>[1] を内包している。インタラクティブに行う会話では、その不明確性を解消する過程が含まれていることがある。多様な解釈が可能な不明確な言葉も、その言葉が使われる状況や他の言葉の組合せの中で使われ方を学習し、適切な1つの解釈を獲得する。本稿では、その学習の過程をアルゴリズムとして取り出し、ロボットに組み込むことを試みる。実験では、決められた走行路を操作者(人間)の発する言葉(命令)を解釈してゴールに至るロボットを用いる。

ロボットとの対話システムに関する研究には、これまでも様々あるが、不明確性の問題解決は状況に依存して決まり、一般にルール化が困難であるなどの理由から、ほとんど取り上げられていない。不明確性を扱った研究である Acorn-II[6] は「大きく回れ」という命令の動作量に含まれる形容詞「大きい」の持つ不明確性を学習により解消するものである。しかし、状況に応じてどれだけの大きさを回れば良いかを、ワークステーション上のマウスを用いて入力し、マウスの軌跡通りに行動するのに必要なステップモータの制御値をワークステーション上で計算し、ロボットに与えている。これは状況に応じて具体的に何度曲がるかその絶対値を教示するのと等価である。

それに対し、本稿では、命令は全て不明確性を含む言葉で与える。ロボットは、言葉の発せられた時刻や間隔、走行方向、走行速度、8つの赤外線センサの出力値による位置を状況情報として検出し、それらを根拠に操作者の言葉(命令)に含まれる不明確性の(状況に依存した)適切な解釈を徐々に学習しながら、脱輪することなく操作者の意図した経路を走行するタスクを実行する。命令に含まれる不明確性のために、ロボットが、操作者の思い

通りに行動をしない場合には、(状況に応じて)修正用の命令を繰り返し与えて学習させる。この修正用の命令は、数値による絶対値を与えるのではなく、不明確性を含む言葉で与える。我々は、この(状況に依存した)ルール化しにくい不明確性の問題を、ロボットとの対話を通じた学習により解消することを試みる。

具体的には、行動命令が出された状況が、それまでの履歴として蓄えている状況情報データベース(行動命令、状況情報、動作量、時間からなる)の中にあるどの状況情報と類似しているか、また同じ種類の命令であるか参照し利用する。類似しているものがない場合はあらかじめ割り当てられているデフォルト値を動作量として用いる。ロボットへの行動命令には、「右に」、「左に」などの基本命令のほかに、「ちょっと動いて」、「ちょっと戻して」などの調整命令を加えた11の言葉を用いる。先に述べたように、基本命令に含まれる不明確性に関する判断の誤り修正を行う調整命令(言葉)そのものにも不明確性が含まれていることに注意したい。また、修正用命令に対して、更に修正用命令が行われることもある。このような言葉による教示は、親が子供に動作を学習させる場面で、よく用いられる自然な方法である。

ロボットの行動命令における不明確性解消のコアとなるのは、学習方法及び学習結果を用いた類似状況の判定方法である。本稿では、センサと壁との距離に関する状況情報をベクトルとして持ち、ベクトル距離で状況の類似度を判定することとした。

## 2 学習による不明確性の解消

### 2.1 実験のシナリオ

・使用する11の命令

「直進」、「止まれ」、「右に」、「左に」の4つを不明確性を、学習させる対象とする基本命令として用い、「ちょっと動いて」、「ちょっと戻して」、「ちょっと行き過ぎ」、「ちょっと右に」、「ちょっと左に」、「ほんのちょっと右に」、「ほんのちょっと左に」の7つは、基本命令の行動量を修正する調整命令として用いる。それぞれの命令には、あらかじめデフォルト値が与えられている。

<sup>†</sup>東京工業大学大学院情報理工学専攻  
Department of Computer Science,  
Graduate School of Information Science and Engineering,  
Tokyo Institute of Technology  
E-mail: {nodoka, take, tanaka}@cl.cs.titech.ac.jp

<sup>1</sup>類似語である曖昧性 (ambiguity) は、語や語の結合が2つ以上の異なる意味を有するものであり、1つの言語表現に無限の解釈の可能性がある不明確性とは異なる。

### ・学習方法

基本命令の動作が操作者の思い通りの行動でなかった場合は修正用の調整命令を出す。この調整命令の出される状況と頻度を用い、基本命令に対する状況に応じた適切な動作量を学習させる。たとえば、基本命令「右に」に対し、「ちょっと戻して」などの調整命令が出された場合は、その状況における「右に」に対する動作量を小さい値に変更(学習)する。具体的には、調整命令の出された回数  $n$  を用いて式(1)より、各状況における基本命令の動作量を算出する。ここで算出される動作量は、次に類似した状況において、同じ基本命令が出された時に、その命令に対する動作量として用いられる。

$$\text{動作量} = [\text{基本命令の動作量}] \pm [\text{調整命令の動作量}] \times N \quad (1)$$

$$\text{但し, } N = \begin{cases} n-1 & \text{命令間隔が3秒以上} \\ n & \text{命令間隔が3秒未満} \end{cases}$$

なお、ロボットがクランクの道幅(21cm)の約半分を進む時間は3秒であることからこの時間を基準とした。式(1)における命令間隔は、基本命令が出されてから最初の調整命令が出されるまでの時間である。また、「ちょっと動いて」命令が出された場合は、 $+$ [調整命令の動作量] $\times N$ とし、「ちょっと戻して」、「ちょっと行き過ぎ」命令が出された場合は、 $-$ [調整命令の動作量] $\times N$ とする。基本命令の動作量の算出法は、後述する学習結果の利用の項で述べる。算出された動作量は状況情報とともにデータベースに書き込まれる。

### ・状況情報データベースの作成

操作者により命令が与えられると、状況情報を取得する。この状況が類似している事例がある場合は、データベース上にあるすでに学習された動作量で動作する。無い場合は、各命令に割り当てられているデフォルト値で動作する。これらの動作が、操作者の思い通りでない場合は、繰り返し調整命令を出すことにより、動作を修正する。調整命令が出されなくなると、式(1)により動作量を算出し状況情報と学習結果の利用に必要なカウンタ<sup>2</sup>とともにデータベース(状況情報データベース)に蓄積する。図1に状況情報に対応した動作量のデータベース作成フローを示す。

### ・学習結果の利用

基本命令が出された時、今、直面している状況における動作量を状況情報データベースを用いて決定する。すなわち、学習した結果を蓄えている状況情報データベースにアクセスし、今直面している状況とすでにデータベースに蓄えている状況が類似状況であるかどうか判断する。これは、データベースは小規模であるが事例ベース推論[2]である。類似判定に関しては2.2節で説明する。データベースに蓄えられている事例の中で、類似状況におい

<sup>2</sup>命令が出されるごとに1ずつインクリメントされるカウンタである。これは、動作量を算出する際、類似事例が複数個ある時に最近のものに重みを置くために使用する。

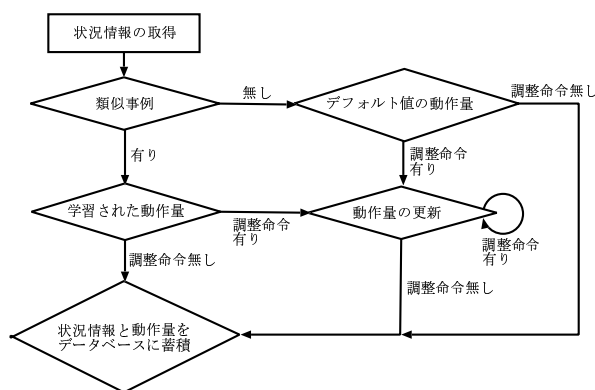


図1: 状況情報データベースの作成フロー

て同じ命令が出されたものがある場合は、基本命令が出された時にデータベースの中にある類似状況の時の動作量分ロボットを動作させる。類似状況が複数個存在する場合は、式(2)により基本命令の動作量を算出する。

$$\text{動作量} = \frac{\sum_{i \in \text{類似事例}} w_i v_i}{\sum_{i \in \text{類似事例}} w_i} \quad (2)$$

但し、 $v_i$  は命令の動作量、 $w_i$  は新しく出された命令ほど寄与率を高く評価するための重み(状況情報データベースの作成で述べたカウンタ)とする。

データベースに類似状況がない場合は、あらかじめ基本命令に対して与えられているデフォルト値で動作させる。

### 2.2 類似判定

行動命令が出された時の状況が、状況情報データベースにある過去の事例と類似しているか判定を行うには、状況情報データベースに蓄えられている事例を逐次参照する。

最初に命令の系が同じであるかどうか照合する。すなわち「右に」、「ちょっと右に」、「ほんのちょっと右に」は「右系」、「左に」、「ちょっと左に」、「ほんのちょっと左に」は「左系」とする。「直進」、「止まれ」は1つしかないの系とはしない。今出された行動命令と参照している事例が異なる系の場合は、類似でない判定し、次の事例と照合を行う。

系が同じ場合は、ロボットのいる場所が類似しているか判定を行う。センサの特性(図2)<sup>3</sup>より具体的に各センサと壁との距離を算出する。KheperaIIに搭載されている8個のセンサと壁との距離を8次元ベクトルとして扱う。今、直面している状況のベクトルを  $s = (s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8)$ 、参照している事例のベクトルを  $t = (t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8)$  とすると、両者の余弦値は以下の式で算出される。

$$\cos \theta = \frac{s \cdot t}{\|s\| \cdot \|t\|} \quad (3)$$

<sup>3</sup>KheperaIIに搭載されている8個の赤外線センサの出力値を1cm刻みで測定したものの平均値を各区間で直線近似する。

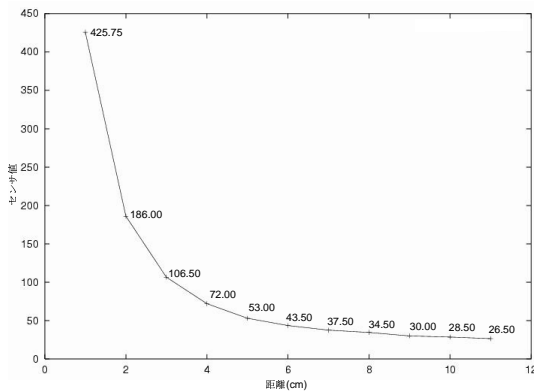


図 2: 壁との距離 (横軸:cm) とセンサ値 (縦軸)

この余弦値が閾値より大きい場合は類似状況と判定する。この手法による判定は、センサと壁との距離は大きく変わっても距離の比が類似している走行路の変更に対しては、同様に対応可能であると考えられる。これらの詳細は 3 章で述べる。

### 2.3 実装

本稿においては、判断・命令を行うブレイン (知能部) と動作するボディ (ロボット本体) を分離して行う方式を採用する [7]。すなわち、コンピュータ側に知能をおき、そこで行動命令の処理、推論を行い、ロボットは処理された通りの動作の実行とセンサ情報とコンピュータ側への送出手を担務する。

ブレインとボディを分離して実験を行うため、kRobot クラス [4] を用いた。このクラスは「右に 60 度曲がれ」という命令を `turnRight(60)` と記述することにより、KheperaII が理解できる形式の命令に変換し、シリアルインタフェースを経由して KheperaII に送信する。つまり、KheperaII とコンピュータとのインタラクティブな通信を行う役割を果たす。この方法を採用すると、ロボット側にプログラムがある場合では困難なコンピュータ側のファイルの読み書きが容易に行うことができる。また、プログラムを変更する都度プログラムをダウンロードする方法とは異なり、プログラムをコンパイルするだけで実行することができる。そのため、開発に要する時間を短縮することができる。

### 2.4 「ちょっと戻して」、「ちょっと行き過ぎ」

「ちょっと戻して」は、単純な命令文である。それに対して「ちょっと行き過ぎ」は、それ自身は平叙文であるが、実際に発話を行う操作者の意図しているのは「行き過ぎているから戻って欲しい」という間接発話行為である。本稿では、間接発話行為処理のために特別な処理を行わず、機能的には「ちょっと戻して」命令と同じ扱いをしている。

「ちょっと戻して」と発話した場合、基本命令により動かし角度が大きすぎたので戻す場合と、基本命令が出された後に進んだ距離が大きすぎて戻す場合の 2 通りが考えられる。そこで状況を考慮し、基本命令が出され

て、「ちょっと戻して」命令が出されるまでの時間間隔が短い場合は角度を戻し、長い場合は基本命令が出された後に進んだ距離が長すぎると判断し後退させる。

「ちょっと行き過ぎ」の意味していることは「ちょっと戻して」と同じであるが、操作者 (発話者) が使用する「言い返し」の自由度を増すため命令文の「ちょっと戻して」と「ちょっと行き過ぎ」の 2 つを用いることができるようにした。

「直進」命令の後に「ちょっと戻して」や「ちょっと行き過ぎ」が出された場合は、進んだ距離が長すぎるので後退させることを意味していると解釈する。目標とする位置にロボットを誘導する際「直進」「ちょっと行き過ぎ」「ちょっと戻して」「ちょっと行き過ぎ」...と命令を繰り返すと、デフォルト値を割り当てておくだけでは望んでいる点で止まらず振動する可能性がある。動かす距離を命令が出されるごとに  $2^{-n}$  することにより<sup>4</sup>、振動することなく、操作者の望んでいる点に収束させることができると考えられる。具体的な解決策については考察で述べる。

## 3 実験及び考察

実験用のロボットには、KheperaII [3, 5] を用いた。本ロボットは、制御装置 (コンピュータ) の指示に従い自由な角度を取ることができ、4cm/秒 (本実験では固定) で走行する。また、ロボットの前後左右にある 8 つの赤外線センサにより外壁からの位置を検出する。実験は以下の 3 種類を行い、ロボットの走行軌跡を分析するために、実験状況をビデオカメラで撮影した。また、命令の入力は、各命令に割り当てられたキー操作により行う。入手可能な現状の音声入力システムでは、本実験で求める性能が得られなかったからである。音声入力方式の導入は、今後の課題である。

### 3.1 「ちょっと行き過ぎ」実験

直進している時と回転している時に与えられる「ちょっと行き過ぎ」に含まれる不明確性の (学習による) 解消過程を、実験により考察する。

常態で前に進んでいる時に「ちょっと行き過ぎ」と命令を受けた場合、どれだけ後退すれば良いかという不明確性を状況に応じて解釈する。ここでの「ちょっと行き過ぎ」という命令は、後ろに下がって欲しいということの意味しているのは明らかだが、どれだけの後退と解釈すれば良いのかは不明であり、これは状況に依存し、一義的に定めることができない。

本稿では、命令が発せられると、最初は、あらかじめ設定されたデフォルト値でロボットを動かし、その動きが適切でないと判断されると、調整命令によって修正が加えられる。この修正値は学習を通して、はじめの命令に対する適切な解釈が行われる。そのときの命令や状況情報を、状況情報データベースに蓄え、次に同じ状況において「ちょっと行き過ぎ」が出された時、修正された値

<sup>4</sup> 予備検討の結果、距離を半分ずつ修正すると、振動せずに操作者の意図している地点まで誘導できることが示された。

を用いる。学習により状況に応じた基本命令と「ちょっと行き過ぎ」の適切な解釈が行なえるようになった。

「ちょっと行き過ぎ」「ちょっと戻して」「ちょっと行き過ぎ」...と繰り返し命令が行われた場合、動作量を更新するが、操作者の望んでいる点に誘導することができず振動することがあった。そこで動かす距離を  $2^{-n}$  ( $n$ は何回目に出されたかを表す) することにより、所望の点に誘導することができた。しかし、この方法で操作者の望んでいる点に誘導することができたのは、上記の命令を繰り返していただいただけであるが、操作者や状況によっては、「大きく動かして」といった命令が必要なことも考えられる。命令語数の拡張やそれに伴う新たな問題をどのように解決するかは今後の課題である。

回転を行った後に、「ちょっと行き過ぎ」という命令を受けた場合は、角度を戻すか、距離を戻すかをまず解釈しなければならない。その上で、距離または角度の適切な解釈が求められる。これも前と同様に状況情報との対応から学習と解釈を繰り返すことになる。その結果、次に同じ状況に直面したときには、直ちに「ちょっと行き過ぎ」の意味を適切に解釈できるようになった。例えば、センサ値が 4 0 16 12 0 0 4 12 の状況の時は 2cm 後退すると適切に解釈された。

本実験より、「ちょっと行き過ぎ」の不明確性が、状況と過去の事例を参照する方法により解消されることが示された。

### 3.2 クランク型走行路実験 (I)

ロボットに命令を与えてクランク型走行路を最短経路で走破する(図3)ことを目標にしたタスクを実行する。最初は、調整命令を含む多くの命令を出すことが必要となるが、学習によって、基本命令だけで図3の目標経路をたどり走破できるようになることが期待される。デフォルト値は「右に」は80度、「ちょっと右に」は60度、「ほんのちょっと右に」は40度(左系も同様)、また調整命令である「ちょっと動いて」は10度、「ちょっと戻して」と「ちょっと行き過ぎ」は10度戻す又は2cm後退に設定した。

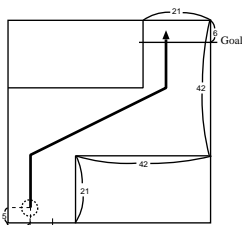


図3: クランク型走行路と目標とするロボットの走行経路(単位: cm)

「調整命令」を出さずに「基本命令」だけでロボットを最短経路(図3)でゴールまで導く過程を分析し、命令(言葉)の適切な解釈(不明確性の解消)に至る経緯を明らかにする。このクランク型走行路実験では、「右に」と「左に」の命令を必ず使わなければならない。以下に、3回

の走行実験で適切な解釈がなされるようになり、4回目では「調整命令」を出さずに、クランク型走行路を最短経路で走破することができた例を示す(図4, 5, 6)。

### 3.3 クランク型走行路実験 (II)

実験2と同じタスクを実行するが、デフォルト値を大きく設定する。すなわち「右に」で120度、「ちょっと右に」で110度、「ほんのちょっと右に」で100度(左系も同様)に設定した。実験3.2のデフォルト値を大きくすることにより、操作を難しくしたものである。

5回の学習後6回目では「調整命令」なしに最短経路を走破することができた例を示す(図7, 8, 9)。また、この実験における「右に」命令に対する動作角度として63度を獲得する過程を以下に示す。

- 1回目  
センサ値が 0 12 44 12 0 36 20 0 で120度
- 2回目  
センサ値が 28 0 20 32 0 0 0 24 で25度
- 3回目  
センサ値が 36 0 0 16 16 0 0 32 で56度
- 4回目  
センサ値が 32 0 48 0 0 0 8 28 で56度
- 5回目  
センサ値が 0 40 32 0 0 52 0 0 で60度
- 6回目  
センサ値が 36 0 12 0 12 36 0 0 で63度

### 3.4 走行路を変更した実験

既学習結果が適切な動作量の獲得(不明確性の解消)に有効に働くかについて道幅、曲がる角度/形状、開始点を変更した実験により、議論する。

はじめに道幅21cmのクランクで学習させた状況情報データベースを用いて道幅14cmの相似形に縮小したクランクを走行させる実験を行った。図10で示すように、幅21cmのクランクでは、最短経路走行ができる。この状況情報データベースを用いて、走行実験を行った結果が図11である。

調整命令を一度も用いることなく、最短経路走行を実現することができた。走行路の幅がロボットの大きさより充分広ければ、一般に最短経路で走行する経路も相似形となる。すなわち言葉(命令)に対する適切な動作量も等しくなり不明確性の解消は、既になされていると考えられる。

次に曲がる角度/形状変更に対し、既学習結果の有効性を考察する。この実験では、各走行路を走行させるごとに、学習結果を蓄積しそれに基づき動作量を決定する。まず初めに、ゴール地点の道幅を狭めた走行路1(図12)を用いた実験を行った。実験3.2の学習結果を用いているので「右に」の動作量は適切である。「左に」は動作量の調整が必要であるが、変更量は小さい。したがって、1

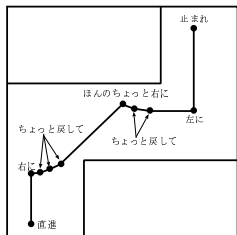


図 4: 実験 3.2(1 回目)

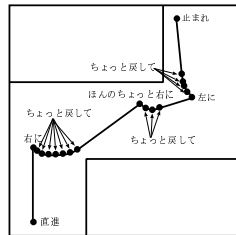


図 7: 実験 3.3(1 回目)

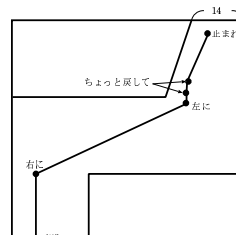


図 12: 走行路 1(1 回目)

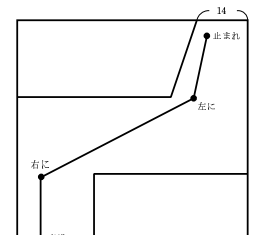


図 13: 走行路 1(2 回目)

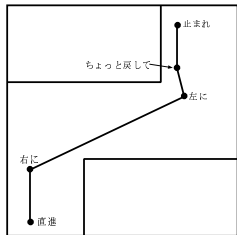


図 5: 実験 3.2(3 回目)

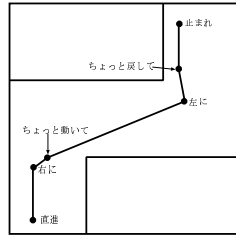


図 8: 実験 3.3(4 回目)

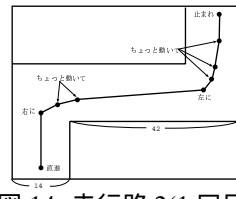


図 14: 走行路 2(1 回目)

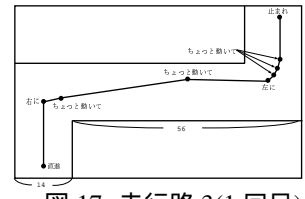


図 17: 走行路 3(1 回目)

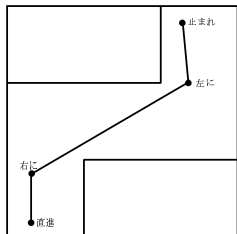


図 6: 実験 3.2(4 回目)

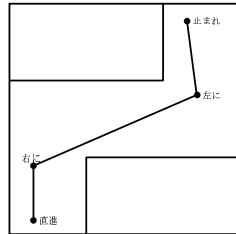


図 9: 実験 3.3(6 回目)

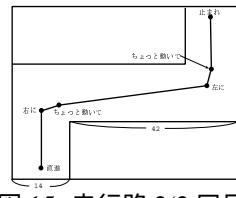


図 15: 走行路 2(2 回目)

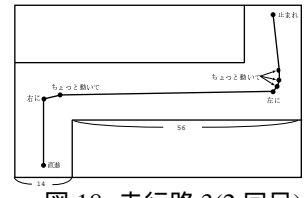


図 18: 走行路 3(2 回目)

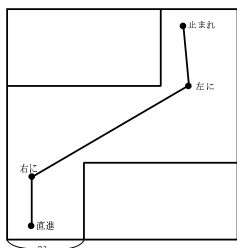


図 10: 幅 21cm での学習結果

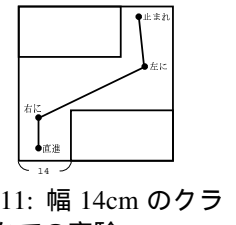


図 11: 幅 14cm のクランクでの実験

回目の調整命令により獲得した動作量が適切となり、2 回目には (1 回の学習だけで) 基本命令のみで最短経路に近い走行が可能になった (図 13)。

次に、道幅を 14cm に狭め、真ん中の部分だけを長くした走行路 2 (図 14) の実験を行った。走行路 1 と実験 3.2 の学習結果を蓄積しており、1 回目の「右に」は、2 回の調整が行われた。「左に」は、走行路 1 で小さい角度に調整した影響があり、角度を大きく変える必要があるため、調整命令が 3 回出されている。2 回目は、1 回目 (図 14) の適切な学習結果が反映され、「右に」、「左に」とも 1 回のみ調整が行われた (図 15)。3 回目は、2 回の学習により適切な動作量を獲得 (不明確性の解消) したため、基本命令のみで最短経路に近い走行が可能になった (図 16)。

走行路 3 (図 17) は、真ん中の部分を更に長くしたものである。走行路 2、走行路 1、実験 3.2 の学習結果を蓄積しており、1 回目の「右に」は、走行路 2 以上に動かさなければならぬので、2 回の調整、「左に」は「右に」を

大きく動かしたため、3 回の調整命令が出された (図 17)。2 回目は「右に」は 1 回目での学習結果が反映されたため、1 回のみ調整命令が出されたが、「左に」命令は出される位置が 1 回目と異なるため同じ状況であると判定されず、1 回目と同様 3 回の調整命令が出された (図 18)。3 回目は、正しい動作量が獲得されたため基本命令のみで最短経路に近い走行が可能になった (図 19)。

同じ走行路 3 での実験を、今まで蓄積した状況情報データベースを用いずに実験 3.2 で用いたデフォルト値を用いて、行った実験結果を図 20～24 に示す。走行路 3 で最短経路の走行を行うためには、4 回の試行が必要であった。この結果から、状況に応じて必要となる適切な動作量と初めに設定された動作量の差が大きいほど多くの試行が必要であることが分かる。

走行路 4 は、幅 14cm のクランクを 2 つつなげたものである。この走行路の実験においては、今までに学習した結果を用いることにより、一度も学習させることなく (調整命令を出すことなく)、最短経路に近い走行ができた (図 25)。

最後に、開始点を右端 1/3、真ん中に変更して実験を行った (図 26, 27)。調整命令を一度も使うことなく、既

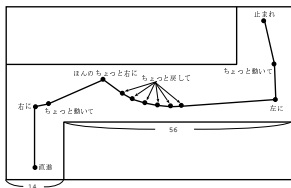


図 20: 学習無し (1 回目)

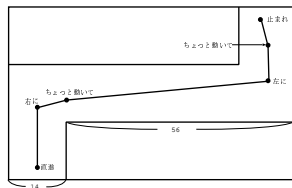


図 21: 学習無し (2 回目)

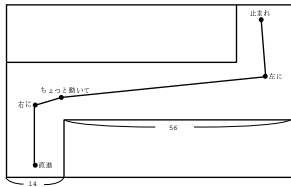


図 22: 学習無し (3 回目)

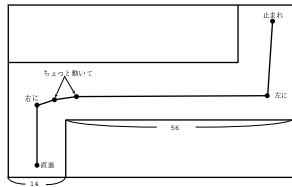


図 23: 学習無し (4 回目)

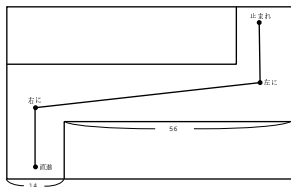


図 24: 学習無し (5 回目)

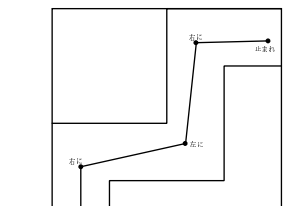


図 25: 走行路 4(1 回目)

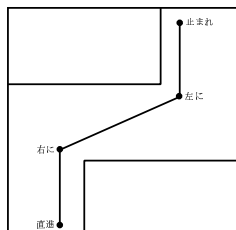


図 26: 開始点を右端 1/3

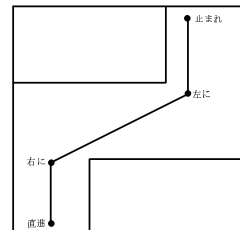


図 27: 開始点を真ん中

学習の結果を用いることにより、最短経路走行を行うことができた。

以上の実験から、相似縮小した走行路では既学習の結果がそのまま利用できる。すなわち基本命令(言葉)に対する不明確性は解消されている。異なった状況では、適切な動作量に近ければ少ない学習回数で適切な動作量を獲得する。すなわち、効率よく不明確性が解消されることが示された。

#### 4 おわりに

本稿では、言葉(命令)に含まれる不明確性の解消をテーマに取り上げ、学習により、言葉の適切な意味を獲得する過程に着目し研究を進めた。具体的には、クランク型の走行路を走行するロボットを用い、操作者の発する行動命令(言葉)を、状況から適切に解釈(適切な動作量の獲得)するための実験をおこなった。ロボットに対する行動命令には、ロボットの行動に関する調整命令も含まれるが、調整命令にも不明確性の問題が含まれてい

る点が特徴である。

実験には位置情報を検出するセンサを具備した KheperaII を用い、行動命令(言葉)が出される時間間隔に着目して命令に対する動作量をロボットが更新(学習)する。位置や動作量をデータベースに蓄積し、次の命令が出された時、このデータベースから類似の状況を検索して動作量を決定する。学習を重ねると行動命令(言葉)を正しく解釈し適切な動作を行うようになり、命令に対する不明確性は解消される。なお、類似判定には、ベクトル空間距離を用いた。

走行路を種々変更した実験を行い、異なった状況では、既学習結果が適切な動作量に近ければ少ない学習回数で適切な動作量を獲得する。すなわち、効率よく不明確性が解消されることが示された。

今後の主な課題は、状況情報や人に依存する言葉の持つ不明確性の解消過程の分析、「適切な解釈(不明確性の解消)」の幅の分析などである。今回の実験では、行動命令として各命令に割り当てられたキーボードのボタン(信号)を用いていたが、操作者の発する様々な行動命令に対応する方法についても検討を行う。本稿では、単純な走行路で実験を行ったが、より複雑な走行路で不明確性の解消がどのように行われるのかを考察することも残された課題である。その時、既学習結果がどのように利用されるのかについても検討する必要がある。

#### 参考文献

- [1] Rosanna Keefe and Peter Smith, editors. *VAGUENESS: A READER*. The MIT Press, 1997.
- [2] Janet Kolodner. *Case-Based Reasoning*. Morgan Kaufmann, 1993.
- [3] Francesco Mondada, Edoardo Franzini, and Paolo Ienne. Mobile robot minituarisation: A tool for investigation in control algorithms. In *Proceedings of Third International Symposium on Experimental Robotics*, 1993.
- [4] Robert M. Harlan, David B. Levine, and Shelley McClirigan. The Khepera Robot and the kRobot Class: A Platform for Introducing Robotics in the Undergraduate Curriculum. In *the 32nd SIGCSE Technical Symposium on Computer Science Education*, pp. 105–109, 2001.
- [5] KTeam. *KheperaII User Manual*, 2002.
- [6] 岡田豊史, 開一夫, 安西祐一郎. ロボットコマンド学習システム Acorn-II とその評価. 人工知能学会誌, Vol. 9, No. 6, pp. 882–889, 1994.
- [7] 稲葉雅幸. ロボット研究の方法論. 人工知能学会誌, Vol. 10, No. 5, pp. 697–703, 1995.