

目次

1	序論	1
1.1	背景	1
1.2	従来研究	3
1.3	先行研究 Motion Space	3
1.3.1	Motion Space の概要	3
1.3.2	Motion Space の問題点	4
1.4	研究目的	5
1.5	本論文の構成	6
2	先行研究-Motion Space	7
2.1	先行研究の概念	7
2.2	Motion Space における知識空間の定義	8
2.3	Motion Space におけるロボット動作手法	8
2.3.1	動作知識化部	9
2.3.2	動作生成部	10
2.4	先行研究の問題点	12
3	提案手法-時間情報を抽象化した Motion Space	14
3.1	時間情報の抽象化方法の検討	14
3.2	Motion Space TS における知識空間の定義	15
3.3	Motion Space TS におけるロボット動作手法	16
3.3.1	動作知識化部	16
3.3.2	動作生成部	18
4	実験	22
4.1	実験環境	22
4.2	実験 1. 時間の長さの違う動作データを用いた実験	23
4.2.1	実験概要	23
4.2.2	実験設定	24
4.2.3	実験結果	26
4.3	実験 2. 位相 (開始点) が違う動作データを用いた実験	31
4.3.1	実験概要	31
4.3.2	実験設定	31
4.3.3	実験結果	34

5	まとめ	40
5.1	本論文のまとめ	40
5.2	今後の展望	40

1 序論

1.1 背景

ロボットのハードウェア性能は年々進歩している．ハードウェアの進歩とは，センサの高精度化，CPUの高速化，アクチュエータの軽量・強力化，ボディ素材の軽量・頑強化などである．ハードウェア面での進歩に伴いロボットに遂行可能な作業が増え，様々な目的に応じたロボットが作られている．それに伴って，工場などの整備された環境だけではなく，ロボットのために整備されていない環境でのロボットの活躍にも期待が高まっている．ロボットのために整備されていない環境とは，深海や月面などの自然環境や，家庭やオフィス・公共施設などの人間のために整備された環境である．本論文では特に後者を扱う．

近年，家庭やオフィス・公共施設などの環境におけるロボットに関する研究への関心が高まっている．家庭やオフィス・公共施設などの環境で活動するロボットの代表的なところとしては，家庭用清掃ロボットなどが挙げられる [1]．また，人の往来する屋外での道案内ロボットの性能を競う研究コンテストとして，つくばチャレンジが 2007 年から 2011 年までの 5 年間開催されていた [2, 3]．家庭やオフィス・公共施設などの環境では人間が生活しているため，環境が大きく様変わりすることは少ないが，刻一刻と変化し続ける動的な環境である．例えば，室内においては家具の配置が変更されることはあまりないが，物が置かれている場所などは日々変化する．屋外ならば，天気が刻々と変化する．そこで，本研究では主に，環境の変化としては小さく緩やかだが断続的な変化を扱う．

環境への適応の方法は 2 種類に大別される．1 つは，ロボットの身体ないし動作を，環境の変化に耐えられるよう設計することである．もう 1 つは，環境の変化に応じてロボットの動作を対応させることである．本研究で取り扱うのは後者である．

また，今回扱う家庭やオフィス・公共施設などの環境は，ロボットのために整備されていない環境であると同時に，ロボットを扱うための専門知識を持たない人々が生活する場でもある．なので，本論文で対象とする利用者は，ロボットを扱うための専門知識を持たない人々であるとする．利用者は，ロボットの仕組みにそれほど詳しくなく，また複雑な数式を扱うことができないものとする．そのような人々でもロボットに任意の動作を指示することができる方法を考える．行動生成に関する問題は，行動計画問題，軌道生成問題，経路計画問題の 3 つに大別される．本研究では，軌道生成問題の領域に着目する．ロボットに動作を指示する方法で，現在用いられている主な方法としては，以下のようなものが挙げられる．

- 人間がリアルタイムに操作を行う方法

コントローラなどを用いて，人間がリアルタイムにロボットに動作の指示を出す方法．操作する人間がロボットが置かれた状況を把握することにより，他の手法と比較して容易に未知環境への適応が可能である．しかし，多自由度を有するロボットの操作は複雑なものとなり，操作者の操作技

術の熟練が必要となる。

- 教示した動作を再生する方法

コントローラなどを用いてロボットに動作を教示し、後からその情報を再生することによって作業を行わせる方法。実際の動作を確認しながら動作を指示することができ、またやり直しもきくため、他の方法に比べ動作の指示が容易である。また、リアルタイム操作と違い、作業時に利用者が操作する必要がない。しかし、ロボットは一定の動作しか行うことができないため、環境の変化には対応できない。環境変化のほぼない工場などでは古くから用いられている手法である。

- ロボットの動きを設計する方法

動作パターンのプログラミング、もしくは動作生成規則によってロボットの動作を設計する方法。センサ情報を用いて、環境の変化に対応した動作を指示することが可能である。しかし、複雑な指示を出すためには、相応の手間と知識が必要となる。また、設計者が想定していない環境でのタスク達成は困難である。

- 学習によって動作を獲得させる方法

ロボットに目的を与え、その過程を試行錯誤によって習得させる。未知環境に対してもタスク達成のための動作を獲得することができる。しかし、軌道生成の領域においてはロボットの自由度増加に伴い動作の獲得が困難になる。

本研究ではこの中でも、“教示した動作を再生する方法”と“ロボットの動きを設計する方法”に注目する。まず、教示した動作を再生する方法に注目した理由は、この中で最も動作の指示が容易であるためである。次に、ロボットの動きを設計する方法に注目した理由は、環境に適応しやすい方法であるためである。学習によって動作させる方法も環境に適応可能ではあるが、リアルタイム性を損なう恐れがあり、また利用者の意図に反した動作をする可能性がある。つまり、本論文で対象とする利用者にとって、教示によって入力された情報によってロボットの動きを自動的に設計する手法が利用できることが理想である。

ロボットと人間が混在する環境で行動するロボットには、タスクの達成に最適だけでなく、人間の要望に沿って動作を行うことが重要となる。ここでは、人間の思ったとおりの動作をいかにしてロボットに伝えることができるかが鍵となる。しかし、ロボットの動作を設定した時点で予想していた人の行動と、実際に稼働させたときの人の行動が違った場合、そのつどロボットの動きを修正しなければならない。しかし、多くのロボット制御方法の場合ロボットに動作を指示する方法は複雑であるため、人々はエンジニアの助けを借りなければならない。利用者自身がロボットに任意の動作を指示できないのでは、ロボットの普及も難しいといえる。

1.2 従来研究

ロボットに教示した動作を再生する手法としては、ティーチングプレイバック [4] などがある。ティーチングプレイバックとは、ロボットを操作し、その動作を記録再生することによって作業を遂行させる手法である。この手法では一般に、ティーチングペンダントと呼ばれるコントローラを用いてロボットを動作させ、その動作を記録する。ロボットは記録された動作をそっくりそのまま繰り返すことによって作業を遂行する。この手法は実際に工場などで用いられている。ティーチングプレイバックは、遂行させたい動作はすべて教示しなければならないという問題もある。そのため環境の変化に適応できず、一般家庭などの動的な環境下では用いることが難しい。

本論文では、環境の変化に適応した動作生成を考える上で、人間の知能に注目する。これまで、人間の知的活動を分析し、環境の変化に適応したロボットの動作を自動的に生成する手法がいくつか考案されてきた。例えば、人間のニューロンにおける信号伝達をモデル化した神経振動子というものがある。この神経振動子によって、ロボットにおいて周期性のある動作を生成する手法がある [5, 6]。この手法は持続的なロボットの運動を生成可能な反面、任意の動作を行わせるためには神経振動子の仕組みや特性を理解する必要がある。他には、アトラクタを設計し、ロボットの動きを誘引することで繰り返し動作を実現する方法がある [7, 8]。アトラクタによってロボットの身体状態を目標とする状態に引き込み、ロボットの動作を生成する。ただしこの方法では、動作を生成するためのアトラクタを設計するのに力学系の知識を必要とする。また、試行錯誤によって目的を達成するのに最適な動作を習得する Active Learning Method という手法がある [9]。この手法では前二者のような動きそのものの設計が必要ない反面、試行錯誤によって生成された動作に対する評価を報酬という形で設定しなければならない。報酬関数を設定するに当たり、ロボットの身体構造や力学系に関する知識が必要となる。以上のように、これらの手法では環境に対してロボットの動作を生成することができる一方で、ロボットの動きを自由に設定するためには専門的な知識が必要であるという問題があった。

1.3 先行研究 Motion Space

1.3.1 Motion Space の概要

我々は過去に Motion Space という手法を提案した [10, 11, 12]。これは、人間が動きを知識化する過程を模倣するというアプローチに基づいた研究である。ここで、動作は何かをするときの身体における姿勢の移り変わり、動きはある結果に達する動作の集まりのことを示す。人間は歩く、座る、コップを持つ、ボールを投げるなどの動きを獲得するのに、動作を繰り返して覚える。ある動きに属する動作をたくさん行うほどその動きに習熟し、最適な動作を行えるようになってゆく。このことから人間は、過去に行ってきた一つ一つの動作を一つの動きとして知識化していると考えられる。Motion Space の研究では、人間が一つ一つの動作を無意識のうちに統合している、この“動きの知識”にあたるものをロボット上で再現することを目指した。

Motion Space では、ティーチングプレイバックで行われるような形で、ロボットの動作に対する各センサの値を入力として得る。その情報によって、知識空間と呼ばれる空間に動作を決定するのに必要な動きの知識を形成する。このとき、教示者が何度も同じ動きを教えることによって動きの知識は深化し、教示者が望む動きの知識に近づく。これらいくつもの動作の情報を統合し、ひとつの動きの知識とする。そして、動きの知識に基づいて、そのつど状況に応じた動作を生成する。動きの知識を用いてロボットを動作させるときに、そのときのロボットの状態を考慮して動作を生成する。実際のロボットは、実環境で動作する関係上、指示したとおりに動作ができない場合がある。そこで、Motion Space では動きの知識に基づきそのつど動作を修正することによって、指示された動作を実行する。

Motion Space の特徴としては、まず教示者が知識空間形成のプロセスに触れずとも、任意の動作を知識化することができる点があげられる。Motion Space の手法では知識空間の形成及び修正を、動作の教示を繰り返すだけで行うことができる。また、最初に知識空間を形成した後でも、生成される動作を確認したうえで知識空間に追加の修正を加えることが可能である。2 つ目の特徴としては、実際にロボットを動作させたときのセンサ情報を用いて知識空間を形成する点があげられる。事前に動きを設計する手法では、動きの設計者が想定していない入出力の関係性を設計することはできないため、想定した動きと実際の動きが違ってしまう可能性がある。Motion Space の知識空間は、実際にロボットを動作させたときのセンサ情報を用いて知識空間を形成するため、教示者が想定していなかった入出力の関係性も知識化可能である。

1.3.2 Motion Space の問題点

従来の Motion Space で動きの知識化をおこなうにあたっては、いくつか問題点があった。1 つは、適切な動作時間の長さを決められない点である（図 1）。Motion Space では、空間をセルに分割する際に、動作知識空間の範囲を有限範囲に限定する。ロボットの状態の範囲、すなわち各センサ軸の限界は、各センサの能力によって決定されていた。しかし、時間軸の範囲、すなわち動作時間の長さは、教示する動作によってまちまちで、一意に決定できない。その結果、あらかじめ決めた以上の動作を知識化できない場合や、Motion Space の一部しか使われない場合があった。

もう 1 つは、位相のずれた同じ動作が、別々の動きとして知識化されてしまう点である（図 2）。動作の知識を表すのに、絶対的な時間経過を基準にしていた。これにより、動作の始点がずれると、同じであるはずの動きが別々の動きとして知識化されてしまうということが起こっていた。これらの問題点の原因は、動きの知識が時間軸に依存していることであると考えられる。Motion Space では動作開始からの経過時間をカウントし、一定時間おきに各時点での状態を取得し、それを動きの情報としている。従来の Motion Space においては、動きの情報とその経過時間の関連性をそのまま利用して知識化を行っていた。そのため、動きの情報を得るために動きの開始時間、終了時間をあらかじめ決めておかねばならなかった。

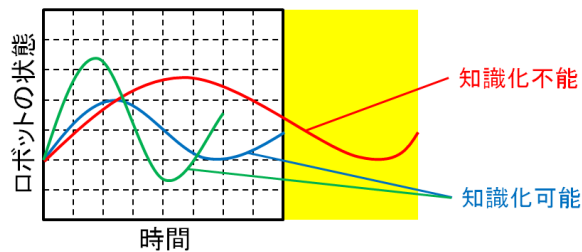


図 1 時間軸の問題 1

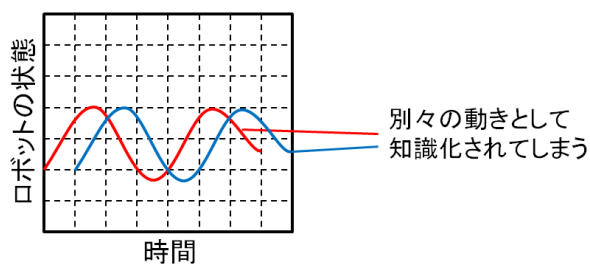


図 2 時間軸の問題 2

1.4 研究目的

本論文の目的は、Motion Space の知識空間の概念を動きの知識が時間軸に依存しないように拡張し、知識空間をより多種の動きに対応できるようにすることである。そのためには、従来の Motion Space において教示・生成される動作が知識空間上にある時間軸によって制約を受けてしまう問題を解決する必要がある。この問題を解決するために、我々はまず Motion Space の知識空間から時間軸をなくすことを考える。時間軸をなくす代わりに、時間軸がなくとも動作を生成することができる方法を考案する。その上で、知識空間に何らかの形で時間情報を抽象化した情報を取り入れ知識化を行う。時間以外に動作を再現可能な情報を用い、また動作を補完することにより、時間軸がなくとも入力された動作を再現する手法を考える。

1.5 本論文の構成

第1章では本研究の背景,目的,アプローチを示した。第2章では先行研究である Motion Space の詳細について説明する。第3章では先行研究を元に,その問題点を解決した新たな手法 Motion Space TS を提案する。第4章では,提案手法 Motion Space TS が従来手法 Motion Space における問題を解決していることを,実ロボットを用いた実験により確認する。第5章ではこの論文を総括し,今後の展望についても言及する。

2 先行研究-Motion Space

2.1 先行研究の概念

先行研究である Motion Space[10, 11, 12] の概念について説明する．先行研究の目的は，ロボットに関する専門知識を持たない利用者でもロボットに任意の動作を指示することができ，かつ環境の変化に応じて自動的にロボットの動作を生成する手法を提案することであった．専門知識を持たない利用者でもロボットに任意の動作を指示することができる方法として，Motion Space ではティーチングプレイバックを参考としていた．ティーチングプレイバックによる動作教示は，ロボットを実際に動作させながら動作を指示することができ，専門知識を持たない利用者にも比較的扱いが容易な手法であるといえる．また，先行研究ではいくつかの教示情報を複合的に利用することによって，利用者（教示者）が真に要求している動作を見つけ出すことができる可能性にも注目していた．ティーチングに熟練していない利用者は，一度で要求する動作を指示することは難しい．しかし，教示を複数回行うことで，その傾向から要求する動作を類推することができると考えられる．

環境の変化に応じて自動的にロボットの動作を生成する手法としては，ロボットに人間の知的活動の一部を再現することを試みていた．ロボットに動作を教示することは，ロボットに動きを覚えさせることと言い換えることができる．人間は，歩く，コップを持つ，ボールを投げるなどの一つの動きを覚えるにあたって，その動きを何度も繰り返すことでその動きがよりスムーズに行えるようになる．これは，動きを繰り返したときの一つ一つの動作の情報を一つの知識に統合して利用しているからであると考えられる．先行研究では，この動きの知識化の活動をロボット上で再現することを考えた．

先行研究では，ロボットにとっての動作の情報とは何かを考えることで，動きの知識化を実現した．ロボットの各センサ情報からなる状態空間内において，ロボットの動作中のセンサ入力軌跡として表示される．このとき，同じ動作を何度も教示していると，その空間内でセンサ入力の軌跡が特定の部分を何度も通過することとなる．そこで，状態空間内に動作の軌跡が特定の部分を通過する頻度の情報を記録することによって，複数の動作入力を統合した情報が得られる．空間内の頻度が高ければ教示者が要求している動作である可能性が高く，低ければ教示者が要求している動作である可能性が低い．この頻度の高さを利用してロボットの動作を生成すれば，教示者が要求する動作を再現できると考えた．この空間を知識空間，頻度を選択頻度と呼ぶ．

知識空間において，選択頻度の高いところが教示者が要求している動作である．しかし，ロボットの状態が急激に変化することによって，ロボットの動作の継続が難しくなる場合がある．また，ロボットの状態や外部からの影響によって，選択頻度が高い行動が取れないこともある．そのため，知識空間とロボットの状態の両方を考慮した動作生成方法を考える必要がある．知識空間内の選択頻度が高いところでは教示者の望む動きを優先し，選択頻度が全体的にあまり高くないところでは教示された知識に追従しながらも，ロボットの状態を優先したロボットの動作を生成すること

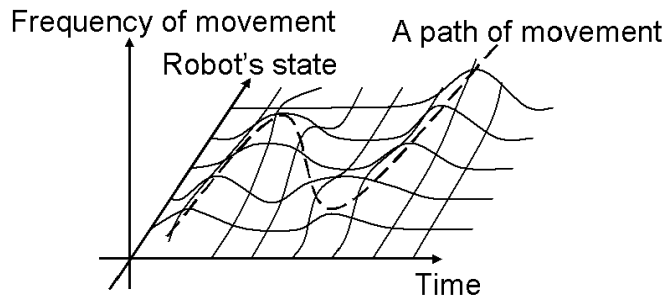


図3 Motion Space の概念

が望ましい．以上の概念を元に，Motion Space が提案された．

2.2 Motion Space における知識空間の定義

Motion Space とは，ロボットの動きを表現するための手法である．Motion Space の知識空間はロボットの状態と時間，そして選択頻度によって構成される（図3）．ロボットの状態とは，ロボットに搭載されるセンサが持つ値の組である．時間とは，Motion Space に入力・出力される動作の時間である．入力された動作が通った場所は，動作が出力されるときも通過する可能性が高いといえる．その入力された動作を，動作を実行する頻度として記録する．この頻度は，その座標の示す状態を選択する頻度，選択頻度という．複数の動作から得られた選択頻度を足し合わせることによって，Motion Space の知識空間はそれら複数の動作の情報を統合した動きの知識として機能する．知識空間の構造を図4に示す．実際の知識空間は，セルによって分割され，離散化される．

従来の l 次元の知識空間内に示される，教示・動作中のある時間におけるロボットの状態を

$$M_t = \begin{pmatrix} t \\ s_{1t} \\ s_{2t} \\ \vdots \\ s_{lt} \end{pmatrix} \text{ とする． } s_{nt} \text{ は } S_n \text{ 軸における時刻 } t \text{ のときの入出力値である．この要素はセンサ}$$

値である．Motion Space 中の各セルの選択頻度を p_{i_1, i_2, \dots, i_l} で表す． i_n は S_n 軸のセル座標を表す．本章では，従来手法を説明するとき，2次元 Motion Space $M_t = \begin{pmatrix} t \\ s_t \end{pmatrix}$ である場合の例を示す．

2.3 Motion Space におけるロボット動作手法

Motion Space 知識空間を利用してロボットを動作させる手法について説明する．Motion Space におけるロボット動作手法の概要図を図5に示す．この手法は，動作知識化部と動作生成部という2つの部分に分けられる．動作知識化部とは，ロボットに搭載されたセンサから得られる動作デー

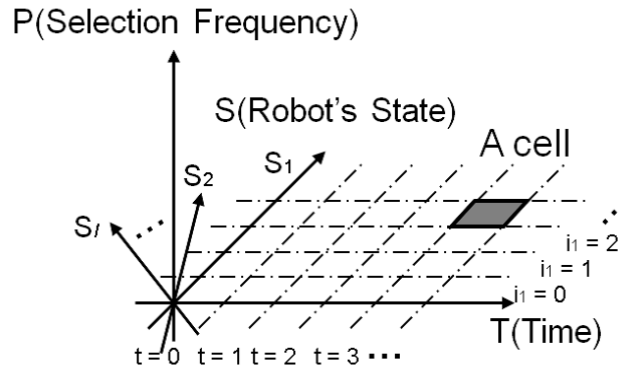


図 4 Motion Space の構造

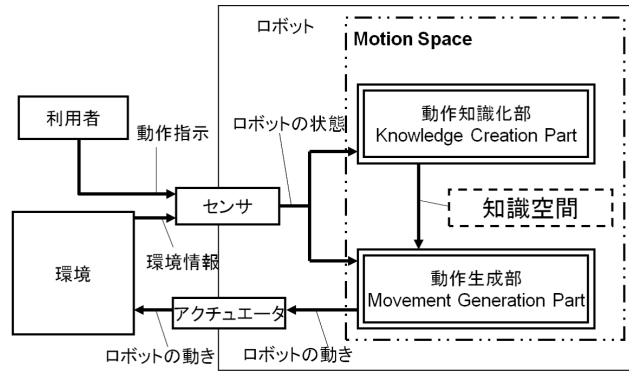


図 5 システム全体図

タを用いて、知識空間に知識を蓄積する部分である。動作生成部とは、動作知識化部によって知識化された知識空間とそのときのロボットの状態を元に、ロボットの動作を生成する部分である。

2.3.1 動作知識化部

動作知識化部 (knowledge creation part) とは、入力された動作データを動きの知識に変換する部分である。本論文では、動作データを動きの知識に変換することを動作の知識化という。知識化する動作データとして、ロボットのセンサ値やアクチュエータ値の時間推移を用いる。利用者があらかじめ決めた一定時間内に入力されたこれらの値を、ひとつの動きの情報として扱う。システムは、動作知識化部によって動きの情報を選択頻度と呼ばれる Motion Space で扱う情報に変換する。いくつもの動作の情報を変換し蓄積することによって、知識空間に動きの知識を形成する。

動作知識化のプロセスについて説明する。利用者がロボットに動きの情報を入力するとき、システムは一定時間 ΔT おきにロボットの状態を得る。ロボットの状態は、ロボットに装備された各センサから取得する。ロボットの状態を取得した後、その状態から知識空間に追加する選択頻度を求める。知識空間内の全てのセルに対して、加算する選択頻度を計算する。各セルに加算する選択頻

度の値は式 (1) によって求められる。

$$u_{t,i} = \frac{C}{r+1} \quad (1)$$

$u_{t,i}$ は、ある時間 t におけるロボットの状態 i を表すセル (t, i) に追加する選択頻度の値である。 C は一度の加算でそれぞれのセルに加算される最大の選択頻度の値であり、事前に利用者が設定する。 r はシステムに入力されたロボットの状態を中心とした、計算対象となるセルの範囲である。最後に、各セルに対して計算された選択頻度を、式 (2) によって更新する。

$$p_{t,i} \leftarrow p_{t,i} + u_{t,i} \quad (2)$$

$p_{t,i}$ は、知識空間においてある時間 t におけるロボットの状態 i を表すセル (t, i) が持つ選択頻度の値である。システム利用開始時の知識空間の各セル $p_{t,i}$ の初期値は 0 である。

利用者は何度も動作データを入力し、知識空間に動きの知識を蓄積していく。選択頻度は、動作データを入力する度に蓄積されていく。ここで形成された動きの知識が、動作生成部でロボットの動作を生成するために利用される。

2.3.2 動作生成部

動作生成部 (movement generation part) とは、動作生成部で形成された動きの知識を利用してロボットの動作を生成する部分である。ロボットに搭載されたセンサから得られる状態データと知識空間内の選択頻度分布を元に、ロボットの動作を生成する。センサから得られるロボットの状態は、必ず知識空間内のどこかに対応する。以降、このロボットの状態を元に知識空間内に表せる一点を、状態点 (state point) と呼ぶ。そして、状態点に対応する知識空間内の位置における選択頻度が高いほど、その状態は望ましいといえる。そこで、状態点が知識空間内の選択頻度の高いところに徐々に接近していくモデルを考える。知識空間内の各セルが持つ選択頻度を、状態点を引き寄せる引力に見立てる。このとき状態点は質量を持ち、近くのセルから受ける引力によって運動するものとする。そして、選択頻度の高いところに引き寄せられた後の状態点に対応する状態になるようにロボットを動作させれば、知識空間に知識化された動作が再現されると考えられる。ただし、実際のロボットが必ずしも知識空間から得られた状態の通りに動作可能であるとは限らない。ロボットが外部からの影響を受け、出力した動作データと実際の動作結果がずれてしまうことが考えられる。そこで、 ΔT ごとにロボットの状態を取得し、そのつど動作を生成するための計算を行う。

ロボットの動作を生成する手順を説明する。まずは、現在のロボットの状態を取得し、知識空間内における状態点の位置を求める。知識空間内において状態点を中心とした範囲 r に含まれるセルを状態点に影響を与える範囲、すなわち影響範囲とする。影響範囲 r 内のセルのもつ選択頻度を参照し、その中で最も高い選択頻度を持つセルを選択セル (chosen cell) とする。選択セルを探索する過程の概要を図 6 に示す。

この選択セルの情報を元に、状態点にかかる引力を算出する。現在の状態点の位置から選択セルの中心までの方向を、引力の指向とする。引力の大きさは、選択セルが持つ選択頻度と選択セルの周辺のセルが持つ選択頻度の大きさの比と、状態点と選択セルの間の距離によって決まるよう定義

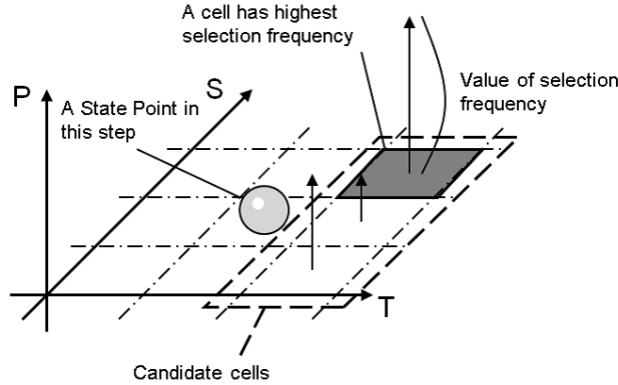


図6 対象セル選択の概要図

する．式 (3) は引力の大きさを求める式である．

$$F_t = k_{t+1} \times (N_{t+1} - M_t) \quad (3)$$

F_t はステップ t における引力の大きさである．選択セルが持つ選択頻度と選択セルの周辺のセルが持つ選択頻度の大きさの差が大きいくほど，引力は大きくなる．また，状態点と選択セルの間の距離が大きいくほど引力は大きくなる． k_t は選択セルが持つ選択頻度と選択セルの周辺のセルが持つ選択頻度の大きさの比である．この比は式 (4) によって求められる． N_{t+1} は選択セルが表すロボットの状態である． M_t は，センサによって求められる現在のロボットの状態である．

$$k_{t+1} = \frac{p_{t+1,max}}{\sum_{i=1}^n p_{t+1,i}} \times \alpha \quad (4)$$

$p_{t,i}$ はステップ t ，ロボットの状態 i に対応するセルが持つ選択頻度の値である． $p_{t,max}$ は選択セルが持つ選択頻度の値である． α は，選択セルが持つ選択頻度と選択セルの周辺のセルが持つ選択頻度の大きさの比による値が引力に与える影響度の大きさを決定するパラメータである．式 (3)，(4) を元に，知識空間から引力を元に状態点にかかる加速度を計算する式を求める．状態点に加速度をかけ，状態点の速度ベクトルの大きさや方向を変化させることによって，ロボットの状態を変化させる．状態点は仮想の速度を持ち，知識空間内を移動する．知識空間内の各状態はロボットの状態に対応しているため，状態点の動きをロボットに反映させることによってロボットを動作させることができる．システムは状態点に加速度をかけることによって，仮想の動きを変化させる．図7は加速度をかけて状態点の動きを変化させる方法の概要図である．仮想の加速度は式 (5) によって求める．

$$a_t = \frac{k_{t+1} \times (N_{t+1} - M_t)}{m} \quad (5)$$

a_t はステップ t における，状態点にかかる加速度を表す． m は状態点の仮想的な質量である．

最後に，式 (5) によって求められた状態点の加速度 a_t を元に，知識空間内におけるロボットの次状態を計算する． $t + 1$ 時刻における状態点の速度 v_{t+1} を，式 (6) によって求める．

$$v_{t+1} = v_t + \Delta T a_t \quad (6)$$

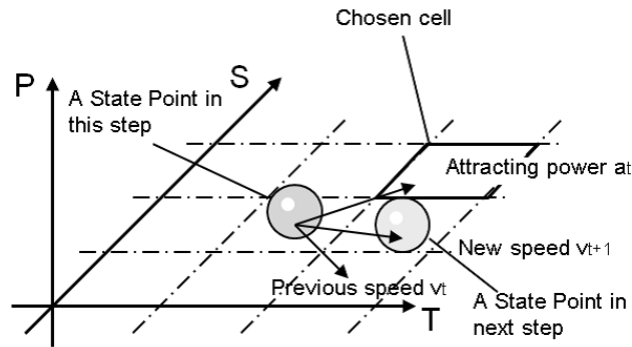


図 7 状態点の仮想的な速度計算の概要図

v_{t+1} は時刻 t から $t+1$ までにおける状態点の速度である． ΔT は現ステップから次ステップまでの時間である． $t+1$ ステップ時点での知識空間内におけるロボットの状態を，式 (7) によって求める．

$$M'_{t+1} = M_t + \Delta T v_{t+1} \quad (7)$$

M'_{t+1} は，知識空間から求められた，ロボットの状態を表す行列である．得られた M'_{t+1} に含まれる状態値を用いてロボットを動作させ， ΔT 経過後に実際に得られたロボットの状態を M_{t+1} として次ステップの計算に用いる．

2.4 先行研究の問題点

ここでは，第 1 章で用いた図を再掲しつつ，先行研究の問題点を深く掘り下げて解説する．従来の Motion Space で動きの知識化をおこなうにあたっては，いくつか問題点があった．1 つは，適切な動作時間の長さを決められない点である (図 8)．Motion Space では，空間をセルに分割する際に，動作知識空間の範囲を有限範囲に限定する．ロボットの状態の範囲，すなわち各センサ軸の限界は，各センサの能力によって決定されていた．しかし，時間軸の範囲，すなわち動作時間の長さは，教示する動作によってまちまちで，一意に決定できない．その結果，あらかじめ決めた以上の動作を知識化できない場合や，Motion Space の一部しか使われない場合があった．

もう 1 つは，位相のずれた同じ動作が，別々の動きとして知識化されてしまう点である (図 9)．動作の知識を表すのに，絶対的な時間経過を基準にしていた．これにより，動作の始点がずれると，同じであるはずの動きが別々の動きとして知識化されてしまうということが起こっていた．これらの問題点の原因は，動きの知識が時間軸に依存していることであると考えられる．Motion Space では動作開始からの経過時間をカウントし，一定時間おきに各時点での状態を取得し，それを動きの情報としている．従来の Motion Space においては，動きの情報とその経過時間の関連性をそのまま利用して知識化を行っていた．そのため，動きの情報を得るために動きの開始時間，終了時間をあらかじめ決めておかねばならなかった．それは，入力に用いる動作データを投影するのに時間軸があったほうが都合良かったためである．また，各時刻において選択頻度が最も高い状態が一

意に決まるため、動作再現の確実性のためともいえる。

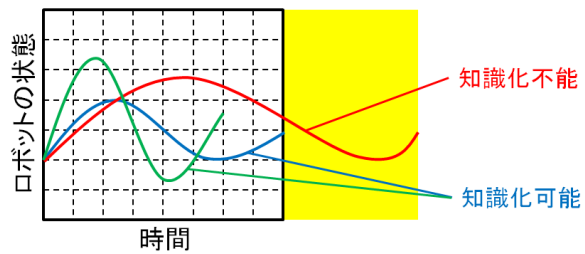


図 8 知識空間の大きさに対する時間軸の問題

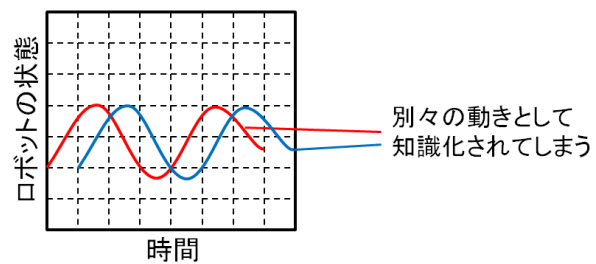


図 9 入力動作のずれに対する時間軸の問題

3 提案手法-時間情報を抽象化した Motion Space

3.1 時間情報の抽象化方法の検討

まず，時間軸がない場合の知識空間からの動作生成の方法について考える．時間軸がある場合とない場合の仮想球の移動方向についてを図 10 に示す．仮想球とは，知識空間上において，ロボットの状態を表すある一点であり，この球が空間上を移動することでロボットの動作を生成する．時間軸がある場合（図左），仮想球は時間軸方向に進むと決まっていた．しかし時間軸がない場合（図右），次時刻における仮想球の進行方向が定まらない．これにより，時刻ごとに仮想球が進む方向が一貫せず，停止状態に陥りやすくなる．時間軸は，空間の中で次に進むべき方向を示す指向性情報を担っていたと考えられる．よって，時間軸がない場合にも仮想球の進行方向に一貫性を持たせる何らかの手段が必要である．

時間軸を持たない知識空間上の仮想球に指向性を持たせる方法を考える．従来手法では，次時刻にあたるセルの中でも，現在の仮想球の位置に近いいくつかのセルを移動先候補と定めている．従来手法では，時間軸の制約によって一貫した方向に進んでいる．そこで，提案手法では今まで移動してきた方向を用いることで，仮想球の指向に一貫性を持たせる．まず，一時刻前のロボットの状態と現時刻の仮想球の位置から，仮想球の移動速度ベクトルを得る．仮想球が何者からも力を受けなければ，次時刻も同じベクトルで進み続けると考えられる．提案手法では，ベクトルの進行方向を中心とした扇形の範囲を移動先候補とする．扇形の大きさは，仮想球の速度ベクトルによって変化するものとする．この扇形の範囲に含まれるセルから仮想球が力を受けることにする．これにより，仮想球が突然進行方向の逆側のセルに引っ張られて停止状態に陥ることを防止する．従来手法と提案手法の仮想球の移動先選択方法の概念を図 11 に示す．

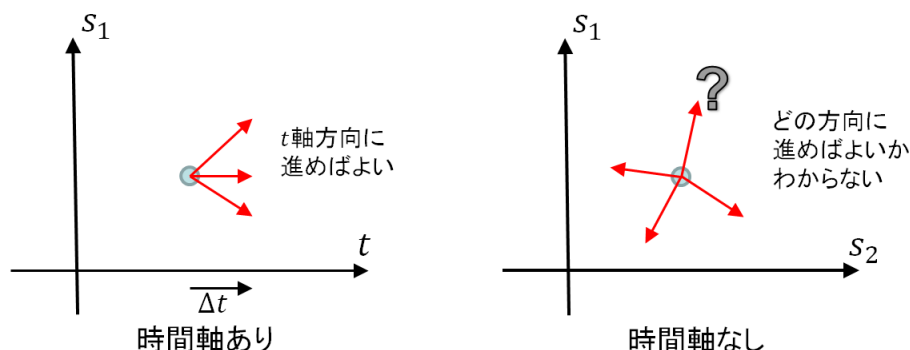


図 10 動作生成の指向性の問題

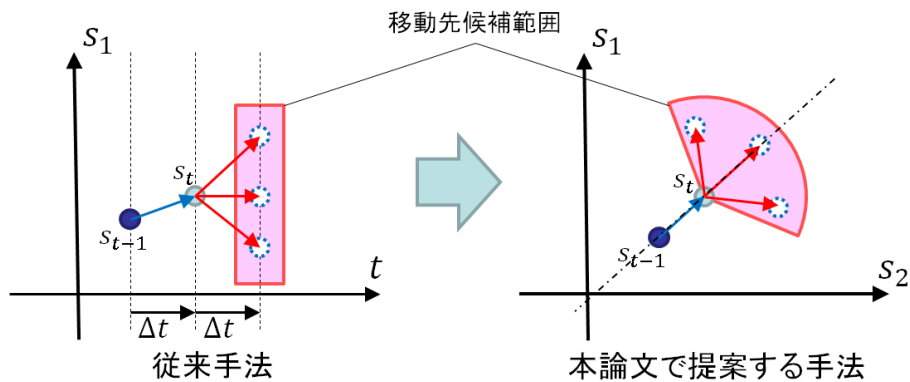


図 11 仮想球の移動先決定方法に関する手掛かり

さらに、知識空間に何らかの形で時間情報を抽象化した情報を取り入れ知識化を行う。本論文では“時間情報を抽象的に持つ動きの情報”として入力情報の時間微分を用いる。例えば、アクチュエータ出力の変位を時間微分すると、アクチュエータ出力の変化速度となる。これらは入出力値に関わる情報であり、かつ時間とも間接的に関係している。本論文ではこれらの情報としてサーボモータの角速度、角加速度を用いる。以上により実現される手法を Motion Space TS (Time Substitution) と命名する。

3.2 Motion Space TS における知識空間の定義

Motion Space TS における知識空間は、全ての状態軸がセンサ情報からなる。時間を抽象化

した l 次元の Motion Space 中におけるロボットの状態を $M_t = \begin{pmatrix} s_{1t} \\ s_{2t} \\ \vdots \\ s_{lt} \end{pmatrix}$ とする。知識空間内

に時間の要素は無いが、動作の入出力には時間の流れがあるため、その場合は時間の流れを t をつけて表す。 s_{nt} は S_n 軸における時刻 t のときの入出力値である。この要素はセンサ値もしくはセンサ値を時間微分した情報である。Motion Space 中の各セルの選択頻度を p_{i_1, i_2, \dots, i_l} で表す。 i_n は S_n 軸のセル座標を表す。本章では、手法を説明する図において、2次元 Motion Space $M_t = \begin{pmatrix} s_{1t} \\ s_{2t} \end{pmatrix} = \begin{pmatrix} \theta_t \\ \dot{\theta}_t \end{pmatrix}$ である場合の例を示す。

3.3 Motion Space TS におけるロボット動作手法

Motion Space TS 知識空間を利用してロボットを動作させる手法について説明する．Motion Space TS におけるロボット動作手法の概要図を図 12 に示す．この手法は，動作知識化部と動作生成部という 2 つの部分に分けられる．動作知識化部とは，ロボットに搭載されたセンサから得られる動作データを用いて，知識空間に知識を蓄積する部分である．動作生成部とは，動作知識化部によって知識化された知識空間とそのときのロボットの状態を元に，ロボットの動作を生成する部分である．

Motion Space TS は Motion Space に基づいているため，手法全体の構成は同じである．しかし，知識空間の定義の変更に伴って，各部の内容を変更している．動作知識化部では選択頻度更新式を変更している．また，動作生成部では動作の生成手順を全体的に変更している．

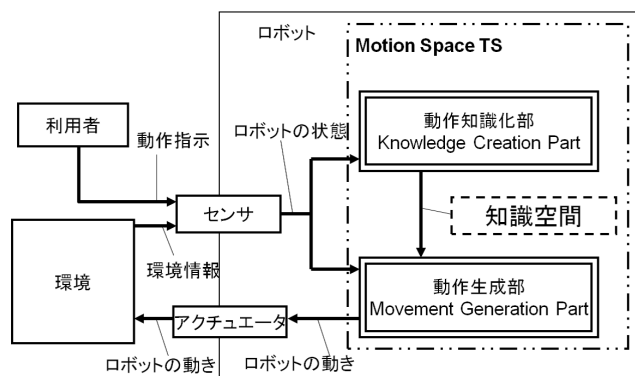


図 12 システム全体図

3.3.1 動作知識化部

動作知識化部 (knowledge creation part) とは，入力された動作データを動きの知識に変換する部分である．本論文では，動作データを動きの知識に変換することを動作の知識化という．知識化する動作データとして，ロボットのセンサ値やアクチュエータ値の時間推移を用いる．利用者があらかじめ決めた一定時間内に入力されたこれらの値を，ひとつの動きの情報として扱う．システムは，動作知識化部によって動きの情報を選択頻度と呼ばれる Motion Space で扱う情報に変換する．いくつもの動作の情報を変換し蓄積することによって，知識空間に動きの知識を形成する．

動作知識化のプロセスについて説明する．知識化の手順を図 13 に示す．利用者がロボットに動きの情報を入力するとき，システムは一定時間 ΔT おきにロボットの状態を得る (図 13(a))．ロボットの状態は，ロボットに装備された各センサから取得する．ロボットの状態を元に知識空間内に表せる一点を，状態点という．

ロボットの状態を取得した後，その状態から知識空間を更新するための選択頻度を求める．前の手順で求めた状態点から距離 Q の間に含まれる全てのセルに対して，更新する選択頻度を計算す

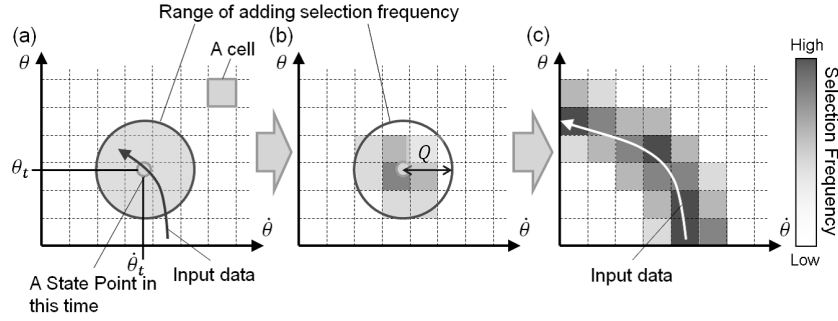


図 13 知識化の手順

る(図 13(b)). 知識空間内の選択頻度の大きさは, 後述のロボット動作の生成に影響を与える. そのため, 選択頻度を更新するにあたって教示回数によって選択頻度の値が大きく変化しない更新方法である必要がある, よって本研究では知識空間内のセルに選択頻度を加算した分だけ周辺セルの選択頻度を減算する更新式を設定する. 図 14 に選択頻度加減算方式の概要を示す. 加算される選択頻度の合計と減算される選択頻度の合計は, 同じ計算式を用いた場合でも知識空間の次元数によって変化する. そこで, 次元数の増加とともに生じる選択頻度合計の差を補正する必要がある. パプス・ギュルダンの定理より, “回転体の体積 = 回転する図形の面積 × 重心の移動距離” である. このことから, 1 次元のとき加算される選択頻度と減算される選択頻度の合計がほぼ同じであれば, 知識空間の時限が増加する毎に “選択頻度加算の中心から重心までの距離の比の分だけ差が生じる” ということがいえる. そこで, 知識空間の次元数に応じて減算する選択頻度にバイアスを掛け, 加算する選択頻度の合計と減算する選択頻度の合計に生じる差が可能な限り小さくなるようにする. 以上より, 各セルを更新する選択頻度の値を式 (8) によって決定する.

$$u_{i_1, i_2} = \begin{cases} \left(\frac{2}{33(q_{i_1, i_2} Q)^2 + 1} - \frac{1}{16(q_{i_1, i_2} Q)^4 + 1} \right) \times C & (q_{i_1, i_2} \leq 0.18Q) \\ \left(\frac{2}{33(q_{i_1, i_2} Q)^2 + 1} - \frac{1}{16(q_{i_1, i_2} Q)^4 + 1} \right) \times C(0.2^{D-1}) & (0.18Q < q_{i_1, i_2} \leq Q) \\ 0 & (\text{otherwise}) \end{cases} \quad (8)$$

u_{i_1, i_2} は知識空間においてあるロボットの状態 i_1, i_2 を表すセル (i_1, i_2) に加算する選択頻度の値である. C は一度の加算でそれぞれのセルに加算される最大の選択頻度の値であり, 事前に利用者が設定する. q_{i_1, i_2} は状態点とセル (i_1, i_2) との距離である. Q は一度の選択頻度更新で選択頻度を更新する範囲である. D は知識空間の次元数である.

最後に, 各セルに対して計算された選択頻度を, 式 (9) によって更新する(図 13(c)). 式 (9) では, 知識空間全体の選択頻度が 1 を超えないように, 選択頻度の高い場所では加算する選択頻度を減らすように計算する.

$$p_{i_1, i_2} \leftarrow \begin{cases} p_{i_1, i_2} + u_{i_1, i_2} & (1 - p_{max} \geq C) \\ (p_{i_1, i_2} + u_{i_1, i_2}) \times \frac{1 - p_{max}}{C} & (\text{otherwise}) \end{cases} \quad (9)$$

p_{i_1, i_2} は, 知識空間においてあるロボットの状態 i_1, i_2 を表すセル (i_1, i_2) が持つ選択頻度の値であ

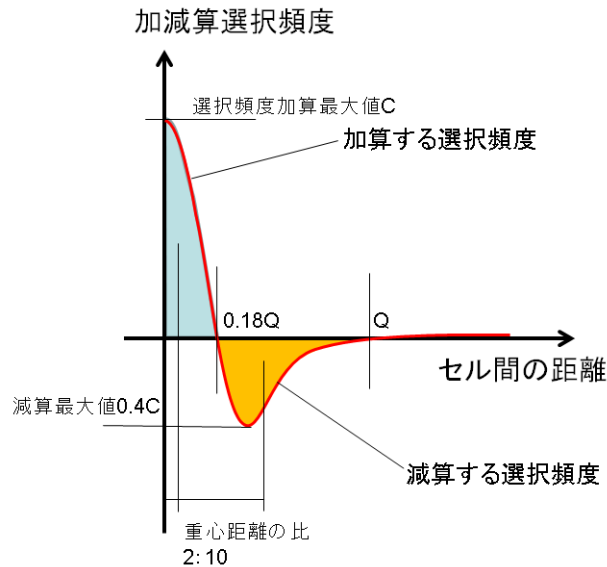


図 14 選択頻度加減算の概要

る． p_{max} は，選択頻度を更新する範囲内に含まれているセルの中で最も高い選択頻度である．システム利用開始時の知識空間の各セル p_{i_1, i_2} の初期値は 0.5 である．

利用者は何度も動作データを入力し，知識空間に動きの知識を蓄積していく．選択頻度は，動作データを入力する度に蓄積されていく．ここで形成された動きの知識が，動作生成部でロボットの動作を生成するために利用される．

3.3.2 動作生成部

動作生成部 (movement generation part) とは，動作生成部で形成された動きの知識を利用してロボットの動作を生成する部分である．ロボットに搭載されたセンサから得られる状態データと知識空間内の選択頻度分布を元に，ロボットの動作を生成する．センサから得られるロボットの状態は，必ず知識空間内のどこかに対応する．このロボットの状態を元に知識空間内に表せる一点を，状態点と呼ぶ．そして，状態点に対応する知識空間内の位置における選択頻度が高いほど，その状態は望ましいといえる．そこで，状態点が知識空間内の選択頻度の高いところに徐々に接近していくモデルを考える．知識空間内の各セルが持つ選択頻度を，状態点を引き寄せる引力に見立てる．このとき状態点は質量を持ち，近くのセルから受ける引力によって運動するものとする．そして，選択頻度の高いところに引き寄せられた後の状態点に対応する状態になるようにロボットを動作させれば，知識空間に知識化された動作が再現されると考えられる．ただし，実際のロボットが必ず

しも知識空間から得られた状態の通りに動作可能であるとは限らない．ロボットが外部からの影響を受け，出力した動作データと実際の動作結果がずれてしまうことが考えられる．そこで， ΔT ごとにロボットの状態を取得し，そのつど動作を生成するための計算を行う．

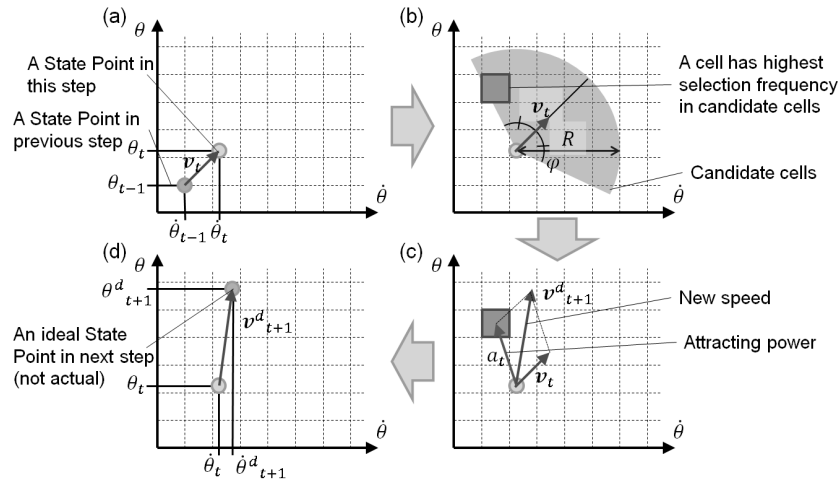


図 15 動作生成の手順

動作生成の手順を図 15 に示す．まず，現在のロボットの状態を取得し，知識空間内における状態点の位置を求める．現在の状態点と 1 ステップ前の状態点の位置関係から，速度ベクトル v_t を求める（図 15(a)）．ここで， t は動作生成開始からの時刻である．このベクトルの方向が，状態点が次に向かう方向となる．

次に，セルの引力が状態点に影響を与える範囲，すなわち影響範囲を決定する．状態点を原点とし，速度ベクトル v_t の方向に広がる扇形の範囲を影響範囲とする．探索範囲を決定する扇形の半径 R_t と角度 Φ_t は，各ステップの速度の大きさ $|v_t|$ に応じてそのつど決定する．角度 Φ_t は，速度ベクトル v_t と半径のなす角であり，扇形の中心角は $2\Phi_t$ となる．この扇形は，速度 $|v_t|$ が大きいほど半径 R_t が長く，角度 Φ_t が狭くなるものとする．式 (10)，(11) に具体的な計算式を示す．

$$R_t = \begin{cases} \beta_1 + |v_t| \times \beta_2 & (\beta_1 + |v_t| \times \beta_2 < \beta_3) \\ \beta_3 & (\text{otherwise}) \end{cases} \quad (10)$$

$$\Phi_t = \pi \frac{\beta_1^2}{R_t^2} \quad (11)$$

$|v_t|$ は，時刻 t における速度ベクトルの大きさである． β_1 は，速度 $|v_t|$ が 0 でも探索を行う最低影響範囲である． β_2 は，速度と候補先範囲の広さの関係を決定する係数である． β_3 は，速度 $|v_t|$ が一定の値以上であるとき探索を行う最大影響範囲である．移動先候補範囲と式パラメータの関係を図 16 に，速度の変化と移動先候補範囲の変化の対応を図 17 に示す．

影響範囲を決定した後，影響範囲内に含まれるセルの持つ選択頻度を参照し，その中で最も選択頻度の高いセルを選択セル (chosen cell) とする（図 15(b)）．選択されたセルの持つ状態を N_{t+1}

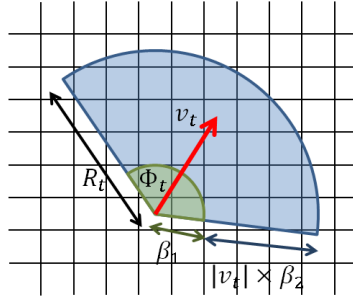


図 16 移動先候補範囲と式パラメータの関係

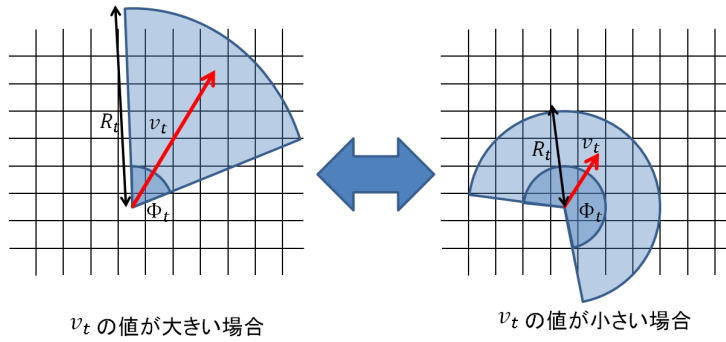


図 17 速度の変化と移動先候補範囲の変化の対応

とし、そのセルの持つ選択頻度を $p_{t,max}$ とする。この選択セルの情報を元に、状態点に対して N_{t+1} に近づくような仮定の引力 F_t を加える（式 (12)）。

$$F_t = k_{t+1} \times (N_{t+1} - M_t) - v_t \times \gamma \quad (12)$$

式 (12) をニュートンの運動方程式 $F = ma$ に基づき加速度を求める式に変形すると、式 (13) となる。

$$a_t = \frac{k_{t+1} \times (N_{t+1} - M_t) - v_t \times \gamma}{m} \quad (13)$$

この加速度 a_t を状態点にかけ、状態点の速度ベクトルの大きさや方向を変化させることによって、ロボットの状態を変化させる。状態点は仮想の速度を持ち、知識空間内を移動する。知識空間内の各状態はロボットの状態に対応しているため、状態点の動きをロボットに反映させることによってロボットを動作させることができる a_t は、影響範囲に含まれる選択頻度に対する $p_{t,max}$ の割合 k_t

と, M_t と N_{t+1} の距離によって求める (式 (13)). k_t は周辺セルの選択頻度に対して $p_{t,max}$ が高いほど高い値となる (式 (14)). a_t は, k_t の値が高く, かつ M_t と N_{t+1} の距離が離れているほど大きな力となる. γ は, 抵抗係数である. 現在の速度 v_t の値が大きいほど, 加速度 a_t が大きくなりすぎないように制限する.

$$k_{t+1} = \frac{p_{t,max}}{\sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_2} p_{i_1,i_2} w_{i_1,i_2}} \times \alpha \quad (14)$$

w_{i_1,i_2} は, セル (i_1, i_2) が扇状の範囲に含まれているかどうかの判定値である. (式 (15)).

$$w_{i_1,i_2} = \begin{cases} 1 & (\varphi \leq \Phi, r \leq R) \\ 0 & (otherwise) \end{cases} \quad (15)$$

w_{i_1,i_2} は, 図 15(b) の影響範囲にセル (i_1, i_2) の中心が含まれていれば 1, 範囲外では 0 となる. φ は速度ベクトル v_t と, 状態点とセル (i_1, i_2) を結ぶ直線のなす角である. r は状態点とセル (i_1, i_2) との距離である..

最後に, 式 (13) によって求められた状態点の加速度 a_t を元に, 知識空間内におけるロボットの次状態を計算する (式 (16), (17)).

$$v_{t+1}^d = v_t + \Delta T a_t \quad (16)$$

$$M_{t+1}^d = M_t + \Delta T v_{t+1}^d \quad (17)$$

v_{t+1}^d は時刻 t から $t+1$ までにおける状態点の仮想速度である. ΔT は現ステップから次ステップまでの時間である. M_{t+1}^d は, 知識空間から求められた, ロボットの状態を表す行列である. 得られた M_{t+1}^d に含まれる状態値を用いてロボットを動作させ, ΔT 経過後に実際に得られたロボットの状態を M_{t+1} として次ステップの計算に用いる. v_{t+1} も同様に, 実際に得られたロボットの状態を元に算出する.

4 実験

本章では、提案手法 Motion Space TS の有用性について、2 つの検証実験を行う。実験 1, 2 では、従来手法 Motion Space の問題点が提案手法 Motion Space TS によって解決されていることを確認する。

実験 1. 時間の長さの違う動作データを用いた実験

実験 2. 位相（開始点）が違う動作データを用いた実験

実験 1. 時間の長さの違う動作データを用いた実験

知識化する動作データの時間の長さに関わらず、知識空間のサイズを変更せずに知識化することが可能であることを確認する。長さの違う動作データの組をいくつか用意し、同じ設定の知識空間にそれぞれの長さごとに入力する。それぞれの知識空間ならびに生成された動作を検証し、それぞれの動作が知識化されていることを確認する。

実験 2. 位相（開始点）が違う動作データを用いた実験

経過は同じであるが開始点のずれた動作データを入力したとき、それらの動作を一つの動きとして知識化することができることを確認する。一つの知識空間に位相をずらした複数の波状の入力データを与えて知識化し、そこから生成される動作を確認する。従来手法による結果と提案手法による結果を比較し、従来手法における問題が解決されていることを確認する。

4.1 実験環境

今回、実験では実機を用いる。実験に用いるロボットの概観と身体構造を図 18 に、実験時のロボットの姿勢と腕の角度の対応を図 19 にそれぞれ示す。実験 1, 2 では図 18 中に示したサーボモータを用いる。動作中には使用しないサーボモータは角度を固定し、姿勢変化が起らないようにする。ロボットの動作を遮るものは、ロボット自身の身体及び椅子以外には存在しないものとする。

実験 1, 2 では入力データはあらかじめ用意したデータセットを用い、動作生成時のみこのロボットを用いる。動作生成時には実際にロボットのサーボモータを動作させ、そのつどセンサよりロボットの状態を取得する。各サーボモータの角度値は、それらに内蔵されたモータエンコーダより取得が可能である。

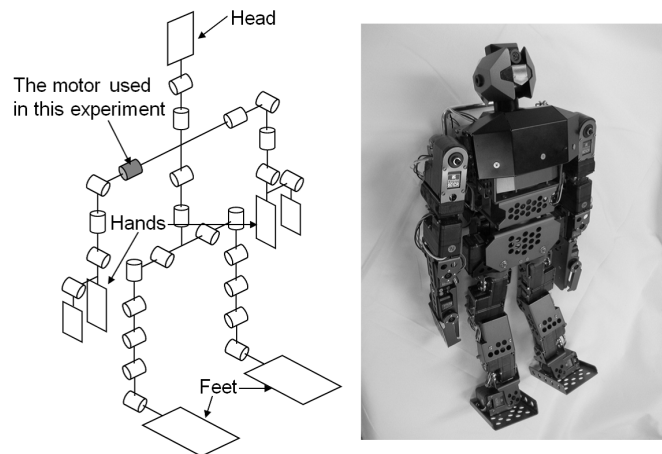


図 18 本実験に用いるロボットの身体構造

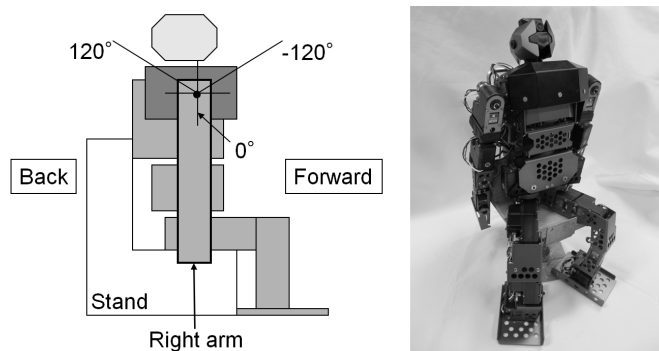


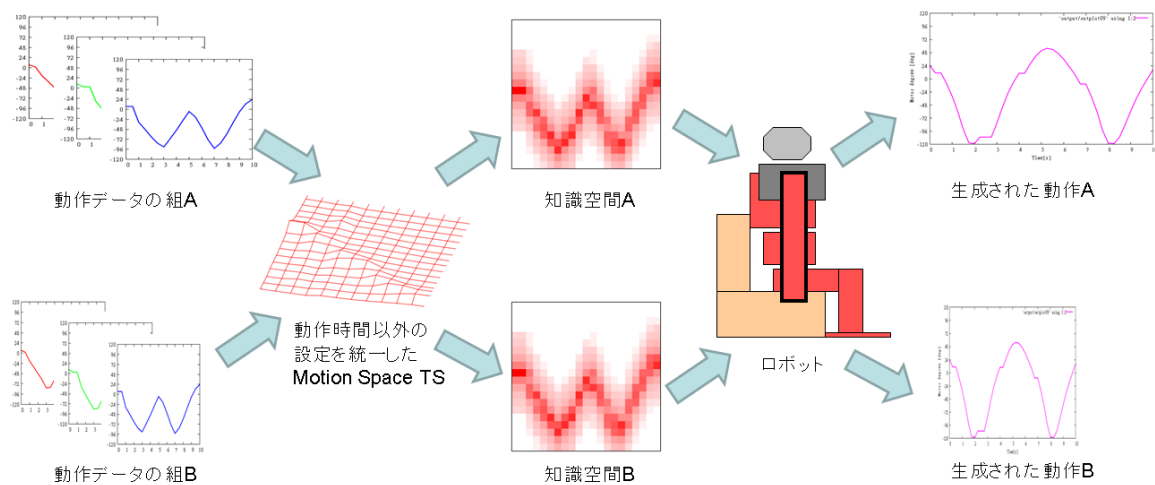
図 19 実験時におけるロボットの姿勢とサーボ角度

4.2 実験 1. 時間の長さの違う動作データを用いた実験

4.2.1 実験概要

この実験の目的は、知識化する動作データの時間の長さに関わらず、知識空間のサイズを変更せずに知識化することが可能であることを確認することである。長さの違う動作データの組をいくつか用意し、同じ設定の知識空間にそれぞれの長さごとに入力する。それぞれの知識空間ならびに生成された動作を検証し、それぞれの動作が知識化されていることを確認する。

実験 1 の手順を図 20 に示す。まず、動作時間の長さが違う 2 種類の動作データの組を用意する。今回は、5 秒の動作と 10 秒の動作の理想的なデータをそれぞれ用意し、各々 3 回ずつ入力することで実験を行う。用意した動作データを、動作時間以外全く同じ条件の Motion Space TS によって知識化する。5 秒と 10 秒でそれぞれ別の知識空間を形成する。知識空間を形成したら知識空間の内容を比較する。そして、それら 2 つの知識空間を用いてロボットを動作させる。そして、それぞれの知識空間から別々の動作が生成されることを確認する。



1. 動作時間の長さが違う2種類の動作データの組を用意する

2. それぞれの動作データを用いて同じ設定の知識空間に知識化を行い, 2種類の知識空間を形成する

3. それぞれの知識空間を元にロボットの動作を生成し, それぞれ違った動作が生成されることを確認する

図 20 実験 1 の実験手順

4.2.2 実験設定

実験に用いた入力データは次の通りである．今回は 5 秒，10 秒の動作を扱う．各時間それぞれ動作データを 3 回ずつ入力し，知識化を行う．また，パラメータ設定は表 1 の通りである．

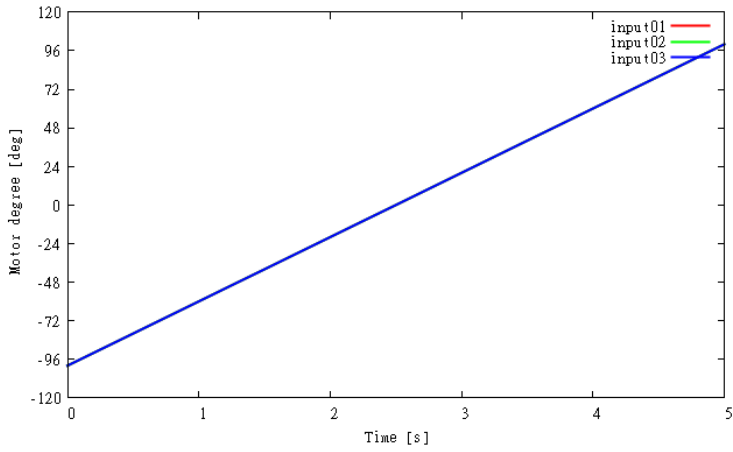


図 21 5 秒の場合に用いた入力動作

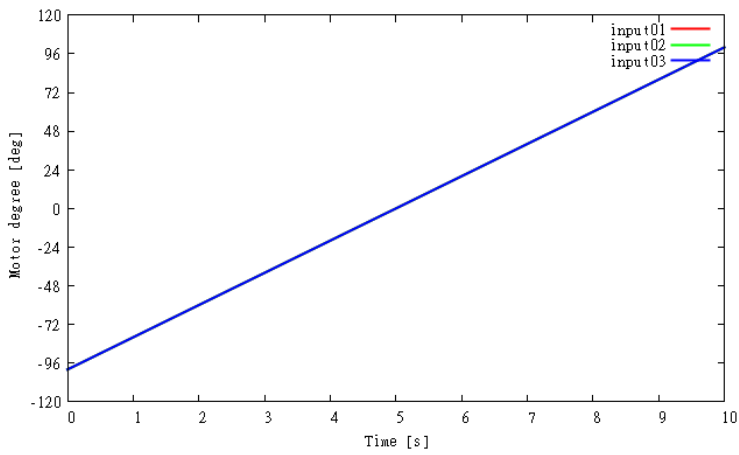


図 22 10 秒の場合に用いた入力動作

表 1 実験 1 におけるパラメータ設定

パラメータ	記号	数値
教示時間 [s]	-	5.0,10.0
動作時間 [s]	-	5.0,10.0
時刻幅 [s]	ΔT	0.25
初期角度 [deg.]	-	-100
角度範囲 [deg.]	-	-120.0 ~ 120.0
角度刻み幅 [deg.]	$\Delta \theta$	6.0
角速度範囲 [deg./s]	-	-180.0 ~ 180.0
角速度刻み幅 [deg./s]	$\Delta \dot{\theta}$	9.0
角加速度範囲 [deg./s]	-	-480.0 ~ 480.0
角加速度刻み幅 [deg./s]	$\Delta \ddot{\theta}$	24.0
選択頻度加算範囲半径	Q	1.0
最低影響範囲	β_1	0.2
速度半径係数	β_2	2.0
最大影響範囲	β_3	0.5
仮想球の質量	m	0.5
傾斜係数	α	800.0
減速係数	γ	0.2
選択頻度加算最大値	C	0.05

4.2.3 実験結果

まず、それぞれの動作を知識化した後の知識空間の選択頻度分布を確認する。今回の Motion Space の表示方法について説明する。今回全ての実験において、知識空間の次元数が 3 次元を超える場合が発生する。このとき、知識空間全体を表示することは困難である。その場合は必要に応じて、知識空間の一部を切り取って表示する。具体的には、知識空間の断面を 2 次元グラフとして表示する。

実験 1 では Motion Space TS の次元数は 3 次元なので、複数の断面を表示することで知識空間の様子を示す。図 23 は、実験 1 の知識空間全体の概要図である。5 秒、10 秒の知識空間ともに、図で示された方向から見た断面図を表示する。

図 24 と図 25 の知識空間をそれぞれ確認すると、今回はいずれも始点と終点と同じ等速度運動なので、角度方向には同じだけ伸びている。角速度方向に図 24 は図 25 の倍だけプラス方向にずれていることがわかる。角速度方向もほぼ 0 近辺に集中している。

それぞれの知識空間を用いて実際に動作を生成してみると、図 26 ならびに図 27 のようになる。これらから、それぞれ違った動きが生成されているということがわかる。以上のことから、Motion Space TS においては動作にかかる時間に関わらず同じ大きさの知識空間に知識化が可能であるといえる。これは、Motion Space の具体的な問題点のひとつを解決しているといえる。

しかし、動きが知識化されたときにあまり差が無かったためか、結果的に生成された動作には

あまり大きな差は見られない(図 28)。これは、動作の差に対して知識空間の(角速度成分の)スケールが大きすぎたためと思われる。

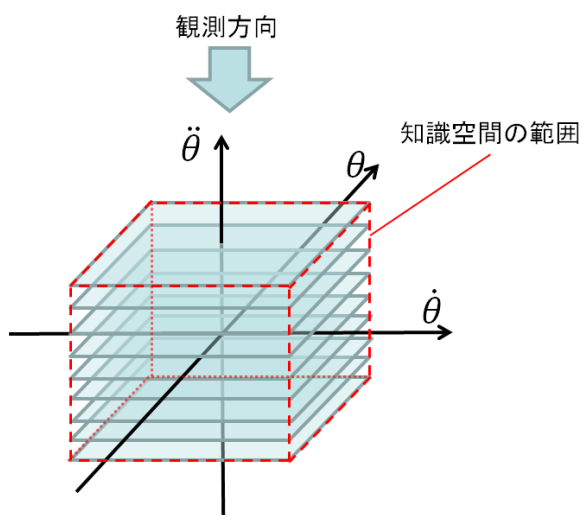
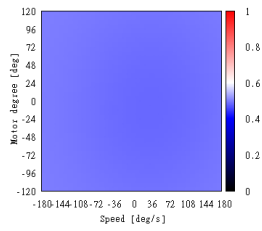
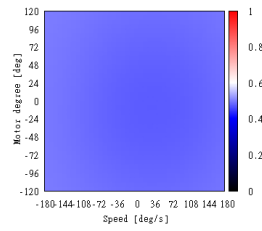


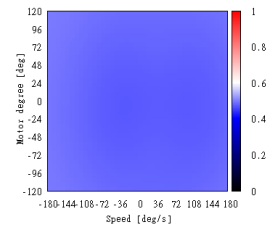
図 23 実験 1 における知識空間の表示方法



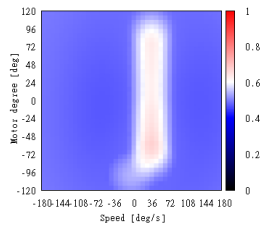
(a) $\ddot{\theta} = -468$



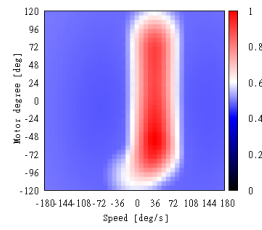
(b) $\ddot{\theta} = -348$



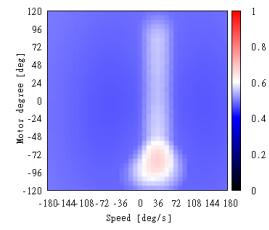
(c) $\ddot{\theta} = -228$



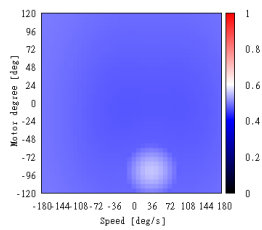
(d) $\ddot{\theta} = -108$



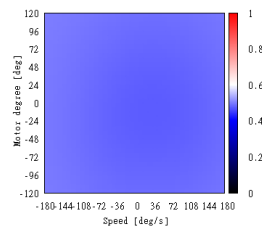
(e) $\ddot{\theta} = 12$



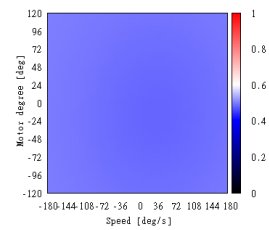
(f) $\ddot{\theta} = 132$



(g) $\ddot{\theta} = 252$

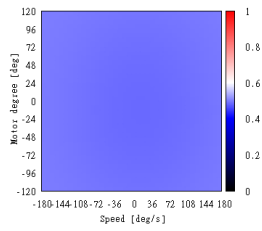


(h) $\ddot{\theta} = 372$

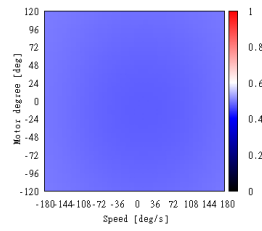


(i) $\ddot{\theta} = 468$

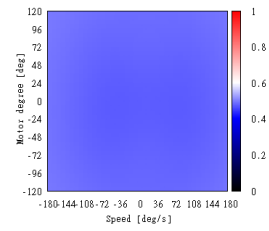
図 24 5 秒の場合の知識空間の選択頻度分布



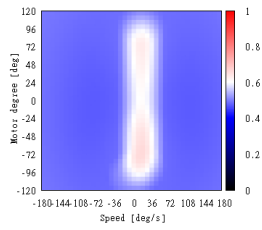
(a) $\ddot{\theta} = -468$



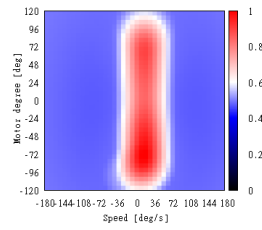
(b) $\ddot{\theta} = -348$



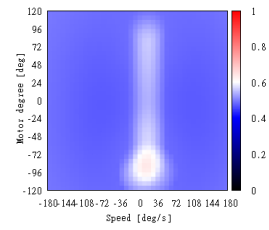
(c) $\ddot{\theta} = -228$



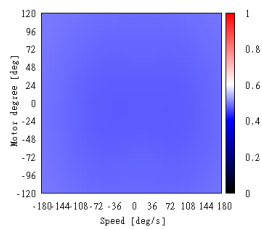
(d) $\ddot{\theta} = -108$



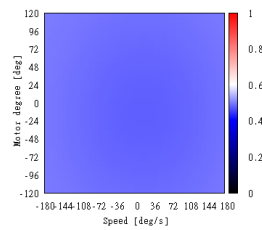
(e) $\ddot{\theta} = 12$



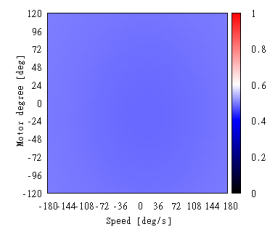
(f) $\ddot{\theta} = 132$



(g) $\ddot{\theta} = 252$



(h) $\ddot{\theta} = 372$



(i) $\ddot{\theta} = 468$

図 25 10 秒の場合の知識空間の選択頻度分布

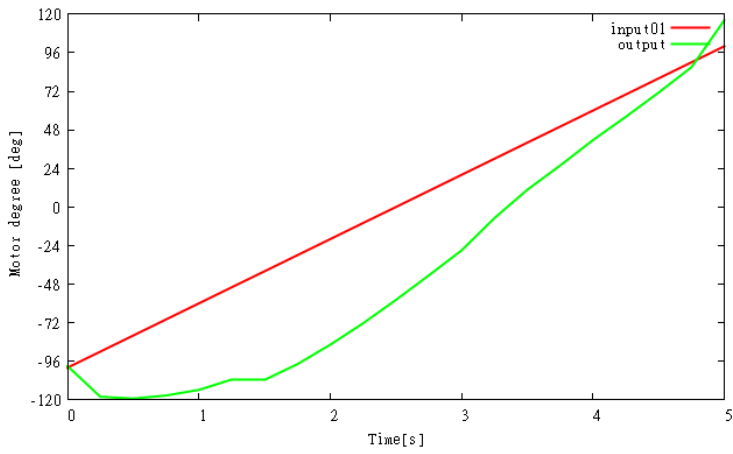


図 26 5 秒の場合の出力結果

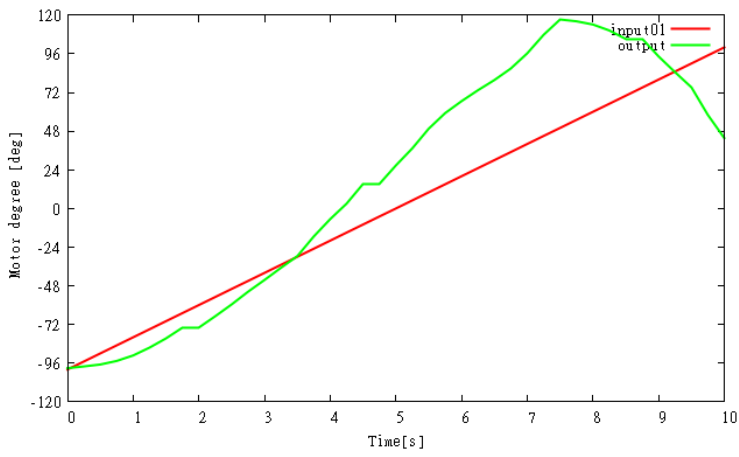


図 27 10 秒の場合の出力結果

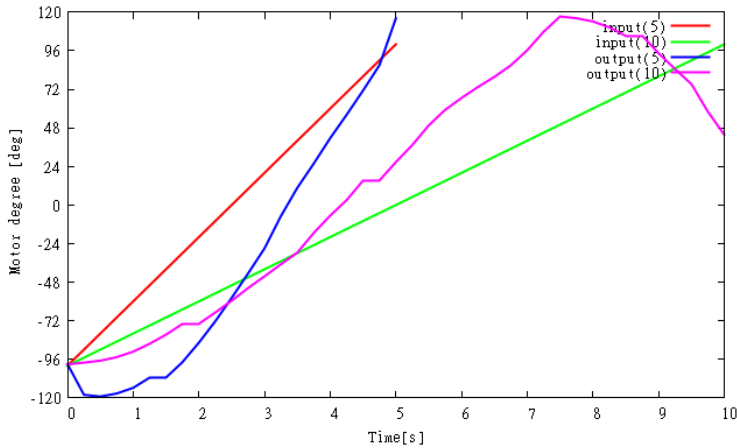


図 28 5 秒の場合の入出力，10 秒の場合の入出力の比較

4.3 実験 2. 位相（開始点）が違う動作データを用いた実験

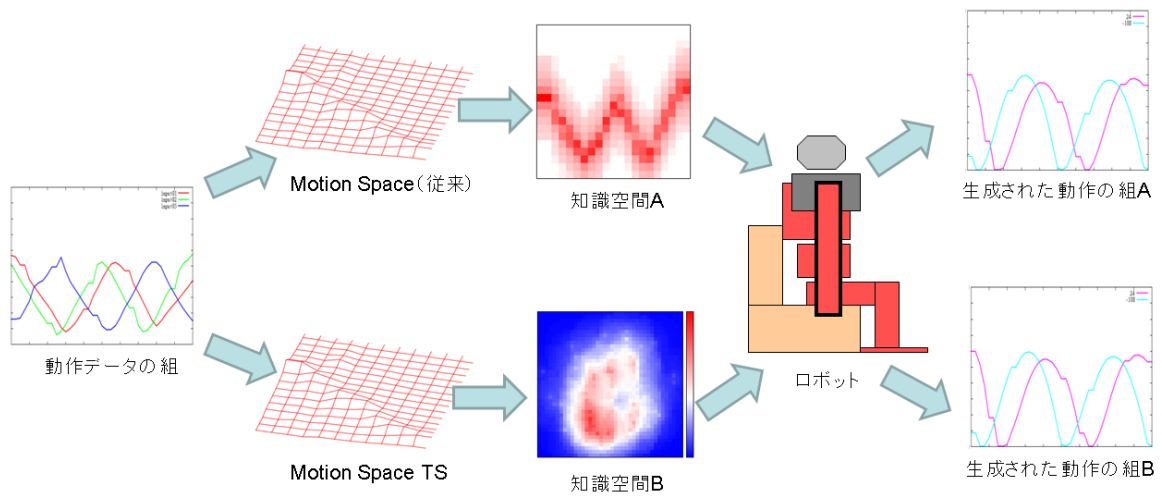
4.3.1 実験概要

この実験の目的は，経過は同じであるが開始点のずれた動作データを入力したとき，それらの動作を一つの動きとして知識化することができることを確認することである．一つの知識空間に位相をずらした複数の波状の入力データを与えて知識化し，そこから生成される動作を確認する．従来手法による結果と提案手法による結果を比較し，従来手法における問題が解決されていることを確認する．

実験 2 の手順を図 29 に示す．まず，同じ波形となる動作データを時間に対してずらしたものを 2 種類用意する．この動作データを用いて Motion Space TS と従来の Motion Space によって知識化及び動作の生成を行う．それぞれの手法に対して用意した 2 種類の動作データを交互に 3 回ずつ，計 6 回の知識化を行う．それぞれの手法によって知識化された知識空間を確認した後，それぞれ動作の生成を行う．複数の初期状態からの動作の生成を行い，その動作結果を手法間で比較する．

4.3.2 実験設定

今回用いた入力を，図 30 に示す．この動作データは，理想の入力を想定したサイン波を与えている．角度の変化は，奇数回の入力するとき式 (18) を，偶数回の入力するとき式 (19) を用いた．また，角速度・角加速度は共通して式 (18) ならびに式 (19) を用いた．また，従来手法ならびに提案手法のパラメータ設定は表 2，3 の通りである．



1. 位相をずらした複数の
同じ動作データからなる
動作データの組を用意する

2. 動作データを従来手法,
提案手法に入力し, それぞれ
知識化を行う

3. それぞれの知識空間を元に
ロボットの動作を生成し, その
結果を比較する

図 29 実験 2 の実験手順

$$\theta_t = 80 \sin(T_t - 2) - 30 \quad (18)$$

$$\theta_t = 80 \sin(T_t + 2) - 30 \quad (19)$$

$$\dot{\theta}_t = \begin{cases} 0 & (t = 0) \\ \frac{\theta_t - \theta_{t-1}}{\Delta T} & (otherwise) \end{cases} \quad (20)$$

$$\ddot{\theta}_t = \begin{cases} 0 & (t = 0) \\ \frac{\dot{\theta}_t - \dot{\theta}_{t-1}}{\Delta T} & (otherwise) \end{cases} \quad (21)$$

表 2 実験 2，従来手法におけるパラメータ設定

パラメータ	記号	数値
教示時間 [s]	-	20.0
動作時間 [s]	-	20.0
時間刻み幅 [s]	ΔT	0.25
初期角度 [deg.]	-	-100,40
角度範囲 [deg.]	-	-120.0 ~ 120.0
角度刻み幅 [deg.]	$\Delta\theta$	6.0
選択頻度加算範囲半径	Q	1.0
候補セル選択範囲半径	R	0.175
仮想球の質量	m	1.0
傾斜係数	α	30.0
減速係数	γ	2.0
選択頻度加算最大値	C	0.1

表 3 実験 2，提案手法におけるパラメータ設定

パラメータ	記号	数値
教示時間 [s]	-	20.0
動作時間 [s]	-	20.0
時間刻み幅 [s]	ΔT	0.25
初期角度 [deg.]	-	-100,40
角度範囲 [deg.]	-	-120.0 ~ 120.0
角度刻み幅 [deg.]	$\Delta\theta$	6.0
角速度範囲 [deg./s]	-	-180.0 ~ 180.0
角速度刻み幅 [deg./s]	$\Delta\dot{\theta}$	9.0
角加速度範囲 [deg./s]	-	-480.0 ~ 480.0
角加速度刻み幅 [deg./s]	$\Delta\ddot{\theta}$	24.0
選択頻度加算範囲半径	Q	1.0
最低影響範囲	β_1	0.2
速度半径係数	β_2	2.0
最大影響範囲	β_3	0.5
仮想球の質量	m	0.5
傾斜係数	α	800.0
減速係数	γ	0.2
選択頻度加算最大値	C	0.05

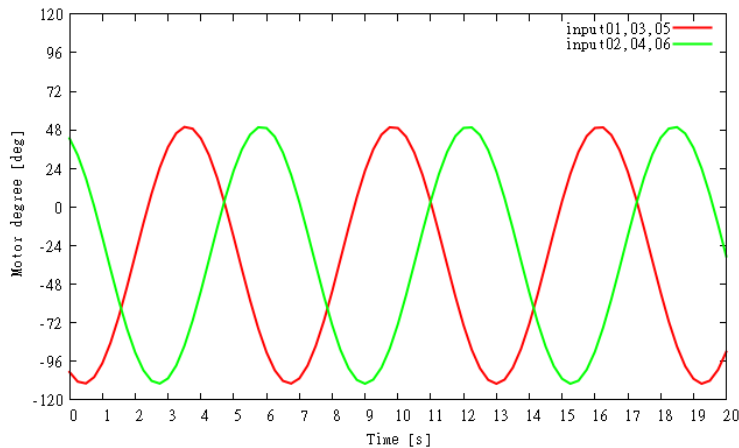


図 30 用いた入力動作

4.3.3 実験結果

まず，従来手法の結果を示す．実験 2 における Motion Space では知識空間の次元数は 2 次元なので，1 つの図によって知識空間の選択頻度分布を表す．従来手法における知識空間を図 31 に示す．従来手法では動きの知識は絶対的な時間と対応する形で知識化される．よって，2 種類の動作入力は別々の動きとして知識化される．図 31 を見ると，2 種類の入力が交差して互いに干渉しあったまま知識化されていることが確認できる．

従来手法での動作出力結果を図 32 に示す．従来手法では，入力に用いた単純な周期動作とは違った動作が出力された．この動作は，時間のずれた 2 種類の動作入力が混ざってしまったためであると考えられる．それぞれの動作出力結果と入力に用いた動作データの比較を図 33, 33 に示す．これらの結果を見ると，入力に用いた 2 種類の動作データの特徴が交互に現れていることがわかる．従来手法では，それぞれの動作が別々のものとして知識化されている上にそれぞれの知識が重なり合っているため，元の動作を再現できなくなっていると考えられる．よって，図 33, 33 のような，結果的にどちらの入力ともかけ離れた動作が出力されたものと考えられる．

次に，提案手法での結果を示す．実験 2 における Motion Space TS では知識空間の次元数は 3 次元なので，複数の断面を表示することで知識空間の様子を示す．図 35 は，知識空間全体の概要図である．図で示された方向から見た断面図を表示する．

提案手法では相対的な状態の変化によって動きを知識化する．よって，過程が同じ動作であればひとつの動作として知識化される．図 36(e) の知識空間断面やその前後 (図 36(d), 図 36(f)) を見ると，動きの知識が角加速度方向にやや傾いたドーナツ状に形成されていることが確認できる．2

種類の動作を知識化した結果、知識の経路としてはひとつになっているので、2つの動きをまとめて効率的に知識化したと言える。

また、図 37 を見ると、生成された 2 つの動作はほぼ同じ動作を時間方向にずらしたものとして生成されていることが確認できる。提案手法では、絶対的な時間に関わらず動きの相対的な変化量によって知識化される。よって、提案手法によって出力される動作は、ロボットの初期状態によってどの地点から開始する動作になるかの違いはあるものの、以上のことから、Motion Space TS においては、位相のずれた同じ動きを一つの動きとして知識化することができているといえる。これは、Motion Space の具体的な問題点のひとつを解決しているといえる。

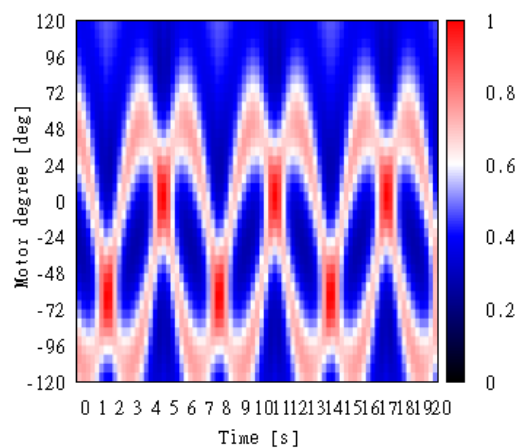


図 31 知識化後の従来 Motion Space の選択頻度分布

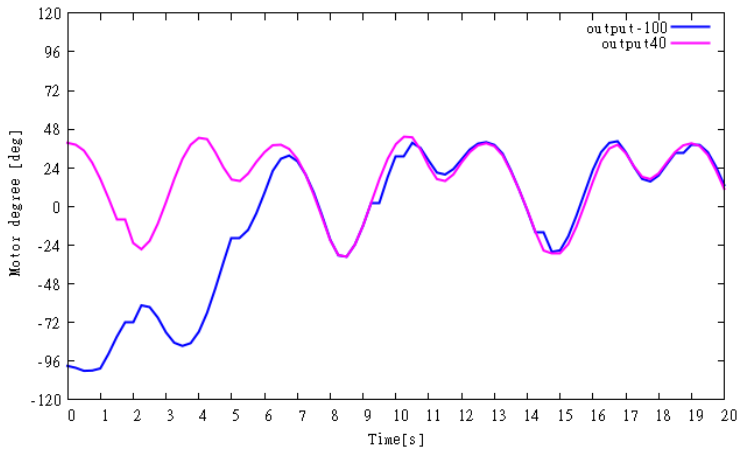


図 32 従来手法による初期値-100°，40°における動作出力結果

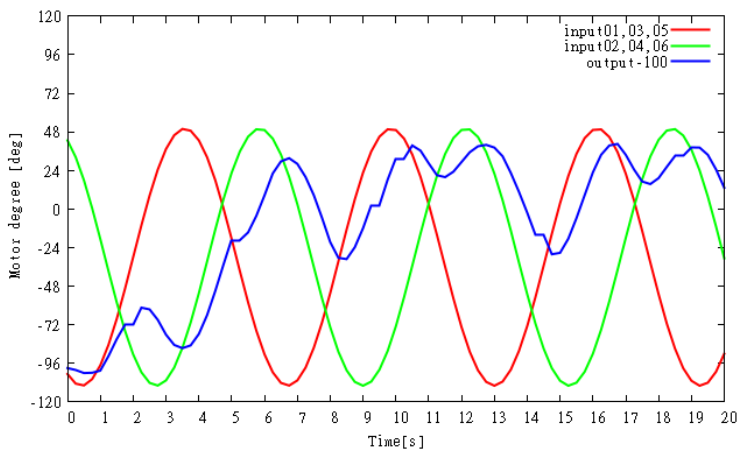


図 33 従来手法による初期値-100°における動作出力結果と入力に用いた動作データの比較

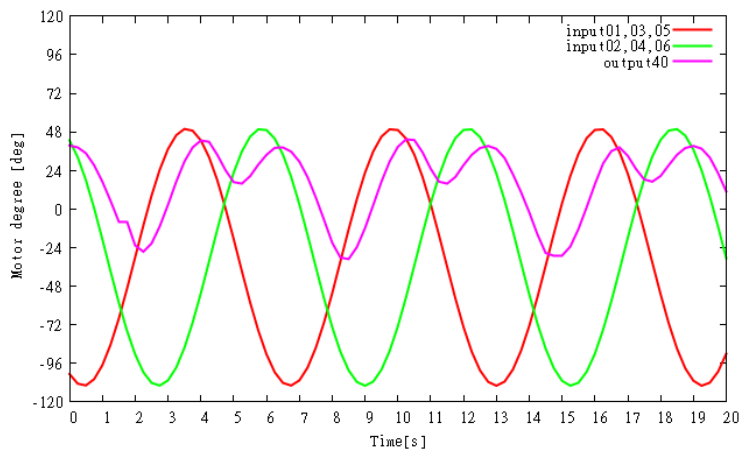


図 34 従来手法による初期値 40°における動作出力結果と入力に用いた動作データの比較

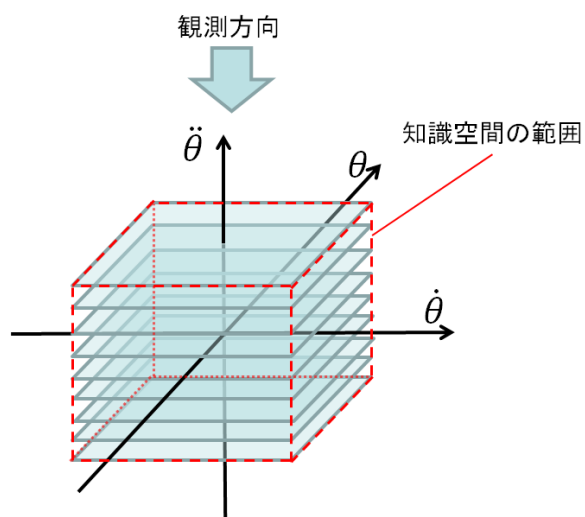
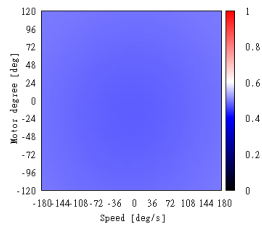
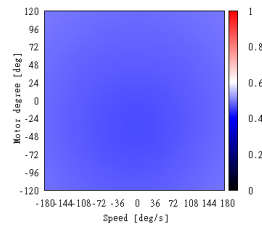


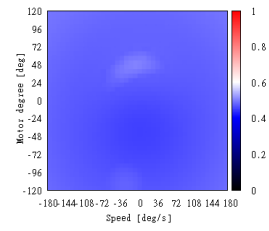
図 35 実験 2 における知識空間の表示方法



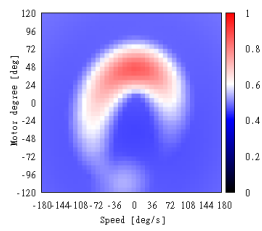
(a) $\ddot{\theta} = -468$



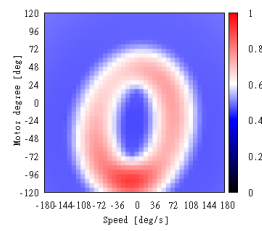
(b) $\ddot{\theta} = -348$



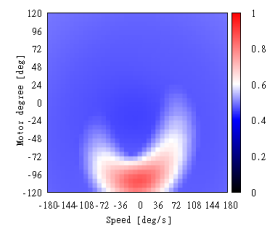
(c) $\ddot{\theta} = -228$



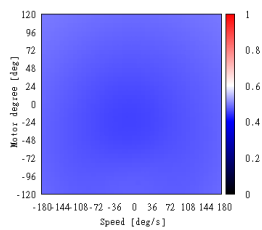
(d) $\ddot{\theta} = -108$



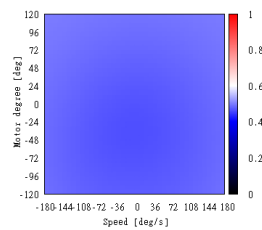
(e) $\ddot{\theta} = 12$



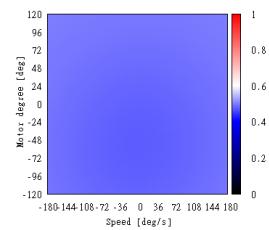
(f) $\ddot{\theta} = 132$



(g) $\ddot{\theta} = 252$



(h) $\ddot{\theta} = 372$



(i) $\ddot{\theta} = 468$

図 36 知識化後の Motion Space TS の選択頻度分布

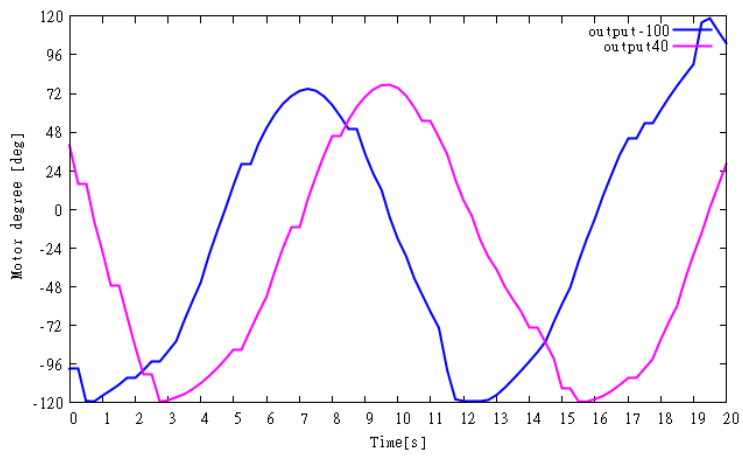


図 37 提案手法による初期値-100°，40°における動作出力結果

5 まとめ

5.1 本論文のまとめ

本論文の目的は、Motion Space において教示・生成される動作が、知識空間上にある時間軸によって制約を受けてしまうという問題を解決することであった。まず、この論文の置かれた背景を説明した。次に、この論文の基礎となる研究として、Motion Space という手法の概念を示した。また、Motion Space におけるいくつかの問題点を示し、その原因を分析した。そして、Motion Space の問題点を解決するために、Motion Space を元にした新たな手法、Motion Space TS (Time Substitution) を提案した。Motion Space TS では、知識空間によって動作の情報を利用するという従来の Motion Space の概念を引き継ぎ、その上で、システムから生成される動作が知識空間上の時間軸によって制約されてしまうという問題の解決を図った。最後に、Motion Space TS によって従来の Motion Space にあった問題が解決できたことを確認するために、検証実験を行った。実験の結果、従来の Motion Space で問題となっていた状況でも、Motion Space TS によって解決できるという結果を得た。以上のことから、Motion Space TS によって、Motion Space の問題点は解決できたといえる。

5.2 今後の展望

Motion Space TS を用いることで、従来の Motion Space で起こっていた問題を解決することができるということがわかった。しかし、Motion Space TS は従来手法に対し、教示に用いた動作と出力された動作との差が大きという結果が得られた。先行研究の Motion Space においては、教示者が望んだ動作の再現も重要な要素である。よって、今後の大きな課題として動作の再現性の向上があげられる。この問題に関して、時間情報を抽象化したことに対し、従来の Motion Space で定義されていた力の生成式は十分に適合していないことが考えられる。そこで、動作の再現性の向上に当たって、動作の生成式を見直す必要がある。知識空間のどの範囲から情報を得るのか、またどのように力の大きさや方向を決定付けるのかなどを検討する必要があると思われる。

実ロボットを用いると、環境やハードウェアエラーなどの色々な要素が影響する。そのため、手法内の式の検討を行ううえでは、実ロボットを用いた実験は不向きであると考えられる。よって、まずは実ロボットの身体性の問題や外部からの影響を排除し、シミュレーション等を用いて動作の生成式を検討する必要があると考えられる。今後はシミュレーションによって手法を検討し、決定した後に実ロボットによる実験を行いその効果を実証することが必要となるであろう。

Motion Space TS においては時間軸を用いずにロボットの動きを知識化することができた。しかし、従来手法と比較して知識空間の次元数が高くなり、それに伴って空間を表現するデータ量が膨大になってしまうという問題の発生が予想される。その解決策として、分割頻度空間 [14] (以後、本論文では Motion Space AD と呼称する。AD は Auto Division の略) の考えを取り入れた新たな手法を考案することが考えられる。Motion Space AD とは、知識空間上において選択頻度

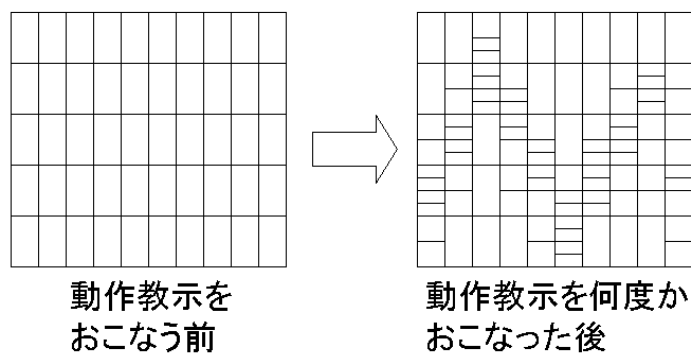


図 38 Motion Space AD による知識空間自動分割の概略

の高い部分のセルだけを細分化し，その部分の動きの知識を精密化する手法である（図 38）．このとき，選択頻度の低い部分のセルは細分化されず，それによって動きの知識のデータ量を少なくすることが可能である．これら二つの手法の概念を組み合わせた新たな手法によって，汎用性の向上が期待できる．

参考文献

- [1] 松日楽 信人, 小川 秀樹, “先端技術をリードするホームロボットの開発動向”, 東芝レビュー, Vol.59, No.9, pp.2-8, 2004
- [2] 油田 信一, 水川 真, 橋本 秀紀: “つくばチャレンジのねらいと成果”, Journal of the Society of Instrument and Control Engineers, Vol.46, No.9, pp.572-578, 2010.
- [3] 井上 一道, Sam Ann Rahok, 尾崎 功一, “つくばチャレンジにおける移動ロボットの設計方針の提案と考察”, 日本ロボット学会誌, Vol.30, No.3, pp.234-244, 2012.
- [4] Haruhiko Asada and Yukio Asari: “The Direct teaching of tool manipulation skills via the impedance identification of human motion”, Proc. of IEEE Int. Conf. on Robotics and Automation, pp.1269-1274, 1988.
- [5] 琴坂 信哉, Stefan Schaal: “神経振動子を用いたロボットのリズム的な運動生成”, 日本ロボット学会誌, Vol.19, No.5, pp.580 ~ 583, 2001.
- [6] Tomofumi Kasuga and Minoru Hashimoto: “Human-Robot Handshaking using Neural Oscillators”, Proceedings of the 2005 IEEE International Conference on Robotics and Automation, pp. 3802-3807, 18-22 April 2005.
- [7] Masafumi Okada, Koji Tatani and Yoshihiko Nakamura: “Polynomial Design of the Non-linear Dynamics for the Brain-Like Information Processing of Whole Body Motion”, Proceedings of the 2002 IEEE International Conference on Robotics and Automation, vol.2 pp.1410-1415, 2002.
- [8] Masafumi Okada and Masaaki Watanabe: “Controller Decomposition and Combination Design of Body / Motion Elements based on Orbit Attractor”, Proceedings of the 2009 IEEE International Conference on Robotics and Automation, pp.1364-1369, 12-17 May 2009.
- [9] Yoshitaka Sakurai, Nakaji Honda, and Junji Nishino: “Acquisition of Knowledge for Gymnastic Bar Action by Active Learning Method”, Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol.7, No.1, pp. 10-18, 2003
- [10] 幾世橋 将文: “ロボットの動きに関する新しい表現方法”, 室蘭工業大学卒業研究, 2008
- [11] Naoki Kitayama, Kentarou Kurashige: “Proposal of method “Motion Space” to express movement of the robot”, Proc. of IWACIII2011 CD-ROM, GS1-3, November 19-23, 2011, Suzhou, China
- [12] Kentarou Kurashige, Naoki Kitayama and Masafumi Kiyohashi: “Proposal of Method “Motion Space” to Express Movement of Robot”, Journal of Advanced Computational Intelligence and Intelligent Informatics(JACIII), Vol.16, No.6, pp. 704-712, September 2012
- [13] 北山 直樹, 倉重 健太郎: “センサ情報に基づく動きの知識化 -時間情報の抽象化による周期運

動の実現-”, 第 30 回日本ロボット学会学術講演会, September 17-20, 2012

- [14] 北山 直樹: “ロボットにおける動きの知識化 -データ空間の自動分割の実現-” 室蘭工業大学卒業研究, 2011

謝辞

本論文を結ぶにあたり，日頃より熱心なご指導を賜りました倉重健太郎先生，ならびに多忙ながら中間発表会などご助言，ご指導を頂きました畑中雅彦先生，佐賀聡人先生，本田泰先生，この場を借りて厚く御礼申し上げます．また，査読などで論文執筆にご協力頂きました認知ロボティクス研究室メンバーの木島康隆先輩，梅津祐介さん，澁谷和さん，杉本大志さん，高泉昇太郎さん，三浦文典さん，木村敏久さん，挟間重直さん，平間経太さんに感謝します．

研究業績

Journal

[1] Kentarou Kurashige, Naoki Kitayama, Masafumi Kiyohashi, "Proposal of Method " Motion Space " to Express Movement of Robot", Journal of Advanced Computational Intelligence and Intelligent Informatics(JACIII), Vol.16, No.6, pp. 704-712, September 2012

Proceedings

[1] Naoki Kitayama, Kentarou Kurashige, "Proposal of method "Motion Space " to express movement of the robot", Proc. of IWACIII2011 CD-ROM, GS1-3, November 19-23, 2011, Suzhou, China

[2] 北山直樹, 倉重健太郎, "センサ情報に基づく動きの知識化 -時間情報の抽象化による周期運動の実現-", 日本ロボット学会第 30 回記念学術講演会, RSJ2012AC4F1-5, 札幌, 北海道, 2012.9.17-20