

目次

第1章 序論.....	1
1.1 背景.....	1
1.2 従来研究.....	2
1.3 研究目的.....	3
1.4 論文構成.....	3
第2章 先行研究：Motion Space.....	4
2.1 Motion Space の概要.....	4
2.2 知識空間.....	5
2.3 動作知識化部.....	6
2.4 動作生成部.....	7
2.5 Motion Space の問題点.....	8
第3章 Motion Space TS 及び提案手法.....	10
3.1 Motion Space TS の概要.....	10
3.2 Motion Space TS の動作知識化部.....	12
3.3 提案手法のアプローチ.....	13
第4章 実験.....	17
4.1 実験目的.....	17
4.2 実験環境.....	17
4.2.1 シミュレーションで想定するロボット.....	17
4.2.2 シミュレーション概要.....	18
4.2.3 仮想ロボットの動作指令計算.....	18
4.3 実験概要.....	19
4.3.1 提案手法による動作生成の確認とその再現性.....	19
4.3.2 提案手法による動作生成の有効性.....	21
4.4 実験結果.....	21
4.4.1 提案手法による動作生成の確認とその再現性.....	21
4.4.2 提案手法による動作生成の有効性.....	25
第5章 結論.....	27
5.1 本論文のまとめ.....	27
5.2 今後の展望.....	28
謝辞.....	29
参考文献.....	30

第1章 序論

1.1 背景

ロボットは技術の進歩に伴い、様々な分野に進出している。実用初期のロボットは、産業ロボットが主であり、運用環境は工場などの運用に適した場所に限定していた。運用に適した場所とは、環境に変化がなく、常に同一の環境を保持している場所である。環境に変化がある場合、それがロボットの動作にとって障害になる可能性もある。そのため、ロボットは環境の変化を予測・管理することが可能な工場などで運用されていた。

近年、家庭やオフィスなどの場所で運用することを目的としたロボットの開発が増えている。これらの環境は、人間が常に生活しているため、環境の変化が発生する。例えば、人間が住む家の室内では家具から小物まで多くのものが置かれている。それらの位置は、人間が生活する中で気まぐれに変わる可能性がある。このように、ロボットが運用される環境は工場などの静的な環境から、動的な環境へと変化している。そこで、本研究では環境の変化に応じてロボットに適した動作をさせる方法を扱う。

動的な環境に対応したロボットの動作を設計するためには、ロボットの構造や内部の処理などの専門知識を有する必要がある。しかし、家庭やオフィスなどの環境では、ロボットの利用者はロボットについての知識を持たない人である可能性が高い。例えば、家庭において日頃使用する電化製品を利用することが出来ても、仕組みや内部の処理、電気回路などを熟知しているわけではない。そこで、専門知識を持たない利用者でも任意の動作を設計・指示できる方法を考える。

利用者がロボットへの動作指示が容易である点と、ロボットが環境の変化に応じて動作できる点に着目し、いくつかの方法を挙げる。まず、利用者がリアルタイムに指示する方法がある。環境の変化を人間が観測し、その都度適した動作を指示し、ロボットを操作する。操作により、ロボットが環境の変化に応じて動作することは、容易である。しかし、ロボットの可動部が増え、構造が複雑になれば、可動部全てに指示を出すことは困難である。つまり、操作が困難になり、操作技術を必要とするようになる。構造の複雑さに応じた操作性の維持が必要である。

次に、指示した動作を記録し、後でその動作を再生する方法がある。事前に指示するため、指示した動作の確認や指示の修正が可能であり、前述の方法とは指示の容易性が異なる。事前に記録するため、実際の環境変化に対応してロボットが動作できない。また、環境変化を全て予測することも困難である。

また、ロボットの動きを設計する方法がある。この動きとは、動作のパターンや一定の規則であり、それを設計する。環境の変化を認識しても、動きの法則を基に動作を生成することで目的に沿った動作の生成を可能とする。しかし、動きの設計には専門的な知識が必要であり、パターンや規則で補えない環境の変化に対して動作を生成することは困難である。

本研究では、動作を記録・再生する方法と動きを設計する方法の2つを対象として扱う。ロボットに動作を指示することで教示でき、教示動作を基に、環境の変化にも対応可能な動きの知識として記録・設計することができる手法を対象として研究を行う。特に、動作教示後にロボットが環境に対応した動作を行う部分について研究を行う。

1.2 従来研究

指示した動作を記録し、後でその動作を再生する方法としてティーチングプレイバックという手法が存在する。この手法は、人がロボットに動作を設定し、その動作をロボットが再生するという手法である。ロボットに動作を設定する際、ロボットのアクチュエータの回転角度などの値を設定する。動作を再生するときは、その値を目標として動作する。この手法では、1度動作を教示すれば、教示動作を何回でも再生可能である。反面、設定した動作しか行うことができず、環境の変化に対応不可能である。

ロボットの動きを設計する際、環境の変化に応じて動作できる動きが必要になる。この分野の従来研究として、「力学的情報処理による運動の表現と設計」[1]が存在する。この研究では、人間の身体と運動の概念は数学的に結合でき、身体の動きがある軌道を描くアトラクタへと引き込まれていることに着目している。目標運動を定めず、アトラクタへ引き込むことで環境に応じた動作の安定化が行えるという。動作を行う部分は、時間的拘束がなく身体と環境とアトラクタへ引き込むコントローラから創発されるため、方向性が近い。しかし、この手法では、ロボットの動きをアトラクタとして事前に設計する必要があり、それには専門的な知識が必要になる。

そこで、先行研究として **Motion Space**[2,3,4]という手法を対象とする。これは、人間が動作を覚えるときに繰り返し行うことで習熟する様子を基に考えられた。ここでは、ロボットの状態の変化を動作、その動作の集合的な概念を動きと呼ぶ。人間が動作を繰り返し行うことで上手になるのに対し、ロボットは最適な動作を行えるようになる。人間が繰り返し行った動作が1つの動きの知識になっていると考える。この動きの知識という考えを表現する方法とそれを利用する手段を提案している。

Motion Space では、ティーチングプレイバックと同じように動作を指示する手法で動きの知識を生成する。このとき、センサから取得した値を入力動作として単純に記録するわけでは無い。取得した値を動作の目標として記録する方法とは異なり、その目標へと動作をするための動きの知識を形成する。形成した知識は動きの知識となり、その知識を基に動作を生成することで自然に目標や目的を達成する動作となる。

従来 of **Motion Space** には動きの知識に対して動作時間に関する制約の問題が存在する。それにより、いくつかの不具合がある。1つは、動作時間の決定が必要である。**Motion Space** は、動きの知識をある空間に形成する。空間であるため、事前にその範囲を決定して形成する必要があるが、動作時間の軸は教示動作によって各々異なる。そもそも事前に正確な値を決定できる値ではない。想定した時間より長い動作を教示する場合は、時間軸範囲以降の動作が教示されない。短い場合は、逆に意図していない動作を教示してしまう可能性がある。

また、教示動作が同じでも開始時刻が異なってしまう場合について考える。動作の記録時開始時刻が異なると同一の動作を異なる動作として知識空間に記録する。記録した動作を動きの知識から生成するときに上手く生成できない不具合を起こす。

さらに、動作生成時にも不具合がある。生成開始時に取得した状態が動きの知識の動作時間途中にある場合でも、知識の開始時刻付近の状態を目標に動作する不具合である。

そこで、**Motion Space** で生成される動きの知識が、時間軸の制約を受けないようにする目的で **Motion Space TS** が提案されている。この手法では、主に **Motion Space** で使用した動きの知識から動作時間を取り除き、センサ情報の時間微分を用いている。動きの知識の構成と教示動作を知識化する部分が提案されており、今回その動きの知識から動作を生成する手法を提案する。

1.3 研究目的

本研究の目的について説明する。動きの知識に動作時間が存在する問題を解決するために提案された **Motion Space TS** に対して、動作生成手法の提案を行い、有効性を検証することである。シミュレーションによって、提案手法と従来手法である **Motion Space** の生成動作とを比較し、有効性を検証することを目指す。本研究では、1.2節で動きの知識から動作時間の情報を取り除くことで時間的制約を解消する手法を紹介した。動作時間の情報を取り除くため、動作を生成する情報が不足する。そのため、動きの知識に新たな時間情報の代用を追加しつつ、動作を生成する手法が必要になった。本研究では、提案手法によって前述の動きの知識から動作を生成し、従来手法の問題点による不具合を解決するとともに、その有効性を検証することを目的とする。

1.4 論文構成

本論文の構成は次のとおりである。

第1章では、本研究の背景を通して、研究領域を定めた。次にその領域での従来研究とその問題点を紹介し、研究対象を定めた。最後に研究目的で研究対象に対する提案手法の目的について説明した。

第2章では、先行研究である **Motion Space** 及び **Motion Space TS** という手法について説明する。手法の構造を特徴とともに説明する。最後に先行研究の問題点についても説明する。

第3章では、本研究の提案手法について説明する。提案手法のアプローチを説明し、次に提案手法の内容を先行研究の比較とともに説明する。

第4章では、提案手法を用いたシミュレーション実験について説明する。シミュレーション実験の目的を説明し、実験設定を説明する。実験概要で実験の内容を説明した後、実験結果とその考察を示す。

第5章では、前章の結果と考察を整理する。整理した内容を基に提案手法の特性をまとめ直す。

第2章 先行研究：Motion Space

本章では、先行研究である Motion Space について説明する。最初に Motion Space の概要を説明する。次に、Motion Space の各構成を説明する。その後、Motion Space の問題点について説明する。

2.1 Motion Space の概要

Motion Space とは、利用者がロボットの運用前に教示動作を指示することで動きの知識を保存し、運用時にロボットが動きの知識を基に環境に対応した動作を行う手法である。

Motion Space はロボットに適用することで使用する。Motion Space には2つの性質がある。1つは、利用者がロボットの専門知識を持たない人でも動作指示が可能である点である。Motion Space は、ロボットに動作を教示する方法としてティーチングプレイバックと同様の方法を用いている。ロボットを実際に動作させることで動作を指示することができる。このとき、ロボットは自身に搭載されているセンサによって自身と環境の情報を取得して動作を認識する。また、1度の教示による動きの知識で動作が生成できなくても、複数回行うことで知識からより利用者が望む動作を生成できる(図2.1.1)。

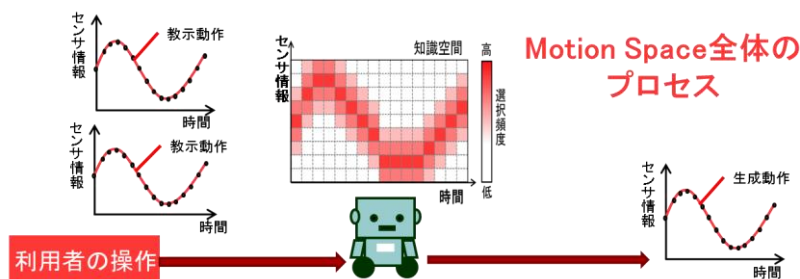


図 2.1.1 Motion Space 利用の流れ

Motion Space の動きの知識化は、複数動作の教示の集合である。1動作に対してセンサの値として取得した時系列データは、センサから取得する値と時間で構成される状態空間に軌跡を描く。複数回動作を入力すると軌跡が重なり交わる場所が発生する。その場所は教えた動作の動きの中で特に教えた動作であることがわかる。そのため、この軌跡が通った場所に対して選択頻度という値を与える。軌跡が交わり、重なるところは選択頻度が高くなる(図2.1.2)。

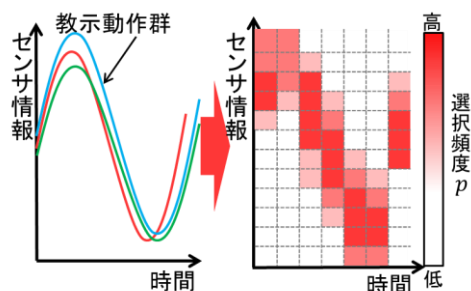


図 2.1.2 教示動作と動きの知識の形成

Motion Space は、2つのシステムで構成される。システムの概要図を図2.1.3に示す。1つは、「動作知識化部」と呼ばれるシステムである。このシステムは、ロボットに入力された動作「教示動作」を知識として記録する動作知識化を行うシステムである。もう1つは「動作生成部」と呼ばれるシステムである。このシステムは取得したセンサの値を基に、動きの知識から動作生成を行うシステムである。また、動きの知識は前述した状態空間と選択頻度で構成された「知識空間」に表現する。

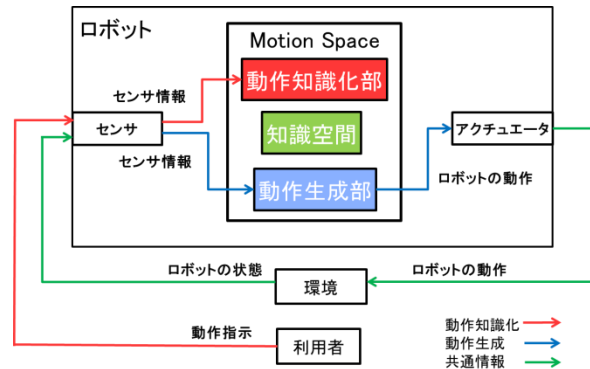


図 2.1.3 Motion Space のシステム概要図

2.2 知識空間

知識空間とは、Motion Space で用いる知識形態である。知識空間は、大きく3つの要素で構成する。1つは、ロボットの状態である。ロボットを構成するセンサが扱う情報の値の組である。知識空間はセンサ情報を離散的に扱う。ロボットのセンサの数だけ軸をもち、軸ごとに利用者の任意の間隔で離散化する。この間隔を1つの状態「セル」として扱う。

2つ目の要素は、時間であり、教示動作の経過時間を扱う。

3つ目の要素は2.1節で前述した選択頻度であり、前述のセル1つ1つに個別に与えるパラメータである。選択頻度はロボットが動作知識化を行う度、取得するセンサ値が示すセルに加算する。動作を生成するときは、この選択頻度を基に次時刻の動作生成を行う。

図2.2.1左に n 次元知識空間の構成を示す。図中の S_1, \dots, S_n は、ロボットが取得する各々のセンサ情報の軸である。 t 軸が動作時間を示す。図2.2.1右に動作と選択頻度の関係を示す。

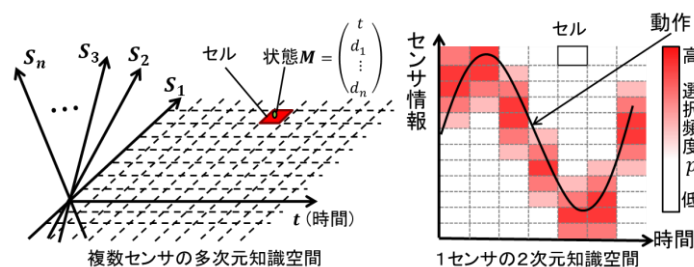


図 2.2.1 知識空間の構成と動作と選択頻度の関係

2.3 動作知識化部

動作知識化部は、教示動作から取得したセンサ情報を基に、知識空間にロボットの動作を知識化するシステムである。入力されたデータの組が示す知識空間内のセルとその周囲のセルに選択頻度を加算することで知識空間の形成を行う。このときの「知識空間の形成」とは知識空間のセルがもつ選択頻度の値を操作することを示す。選択頻度を持たない(選択頻度が0)初期状態の知識空間の形成は知識化部の働きに含まれず、事前に用意する。

知識化の際、ロボットは教示中に自身の状態と環境を任意のサンプリングタイムで取得する。サンプリングタイムごとの動作知識化の手順を説明する。システムはサンプリングタイムごとにロボットの状態をセンサ情報として取得する。取得したセンサ情報を基に、知識空間上の点を決定する。その点を含むセルをあらかじめ設定した選択頻度を与える範囲 R の中心として考える。そして、その点を含むセルと付近のセルに選択頻度を加算する。(2.3.1)式によって範囲内の各セルに与える選択頻度 $u_{l,m}$ を計算する。範囲内の各セルは、(2.3.1)式の通り、中心セルからの距離によって、与える選択頻度は異なる。中心セルに近いほど与える選択頻度は高く、離れるほど低い。計算した選択頻度 $u_{l,m}$ は、(2.3.2)式を用いて更新する(図2.3.1)。

これらの計算式を使用した知識空間の形成処理を、サンプリングタイムごとに行い、教示動作を知識化する。教示動作の軌跡に対応したセルとその周辺に選択頻度を加算することで動作を動きの知識として知識化する。

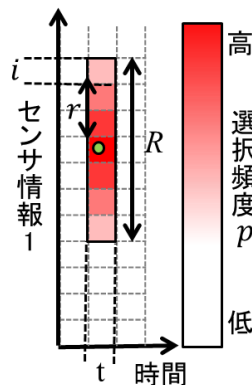


図 2.3.1 動作知識化の様子

$$u_{t,i} = \frac{C}{r+1} \quad (2.3.1)$$

$$p_{t,m} = p_{t,m} + u_{l,m} \quad (2.3.2)$$

$u_{t,i}$: $p_{t,i}$ に加算する選択頻度

C : 一度の加算で加える選択頻度の最大値

r : 中心のセルを0とした, 中心からの加算する対象セル数

$p_{t,i}$: セルの選択頻度

t, i : t は時間軸の番号, i はセンサ情報の軸の任意のセル番号

2.4 動作生成部

動作生成部とは、センサ情報を基に動きの知識である知識空間からロボットの動作を生成する部分である。動作生成とは、動作再現と異なる。前者は、教示した動作を基に動きの知識を形成し、その知識とセンサ情報を基に最適な動作を生成する。後者は、教示した動作をそのまま再現する。

知識空間を基にロボットの動作を生成する方法について説明する。動作の生成時には形成した知識空間が必要になる。動作の生成手法として、知識空間上で仮想球の運動生成を行う。仮想球に運動法則や知識空間からの力を与えて運動させる。取得したセンサ情報と対応する知識空間上の点に仮想球を置く。仮想球は知識空間の選択頻度をもつセルから仮想的な引力を受ける。選択頻度の高低に応じて力は比例する。選択頻度に応じて力が増減するため、教示動作と同じ時間に同じ目標を目指すとは限らない。しかし、複数回の動作教示から総合的に動きの知識が形成されているため、センサで取得した自身の状態と動きの知識から適した動作を生成できる。

仮想球運動の計算方法について説明する。まず、現在の状態をセンサで観測して、そのセンサ情報に対応する知識空間上の点を判定する。次時刻の目標状態を決定するためには、1ステップ前から現在までの実際のロボットの動作によるベクトルと、知識空間からの引力によるベクトルが必要になる。

前者は、1ステップ前のセンサ情報による状態 D_{t-1} と現在の状態 D_t から考えられる。2点間の位置関係から、 D_{t-1} から D_t までの仮想球の運動を計算し、その運動の速度ベクトル v_t を求める。

次に探索範囲を決定する。探索範囲は、仮想球が知識空間から受ける引力をもつセルを探索する領域である。探索範囲内で一番選択頻度が高いセルから力を受ける。探索範囲は、1ステップ先の時間軸で構成されているセル以外に展開しない。探索範囲は現在、仮想球が存在するセルの1ステップ先のセルを中心に考える。そのセルを中心に展開する任意の探索範囲から最大の選択頻度を持つセルを判定し、そのセルの中心に注目する。そのセルの中心との関係から(2.4.1), (2.4.2)式を使用して仮想的な力による加速度 a_t を計算する。(2.4.3)式より、加速度 a_t にサンプリングタイム $T_{t+1} - T_t$ を乗じて、速度ベクトルに変換し、前の運動から求めた速度ベクトル v_t と合成する。この2つのベクトル合成により、仮想球が一番高い選択頻度をもつセルを急に目指すことなく、生成動作が1時刻前の動作を考慮して適した動作になる。(2.4.4)式で動作が次の動作の目標となる座標 D_{t+1} を求める。これらの計算式を使用した動作生成処理を、サンプリングタイムごとに行い、ロボットは動作する。

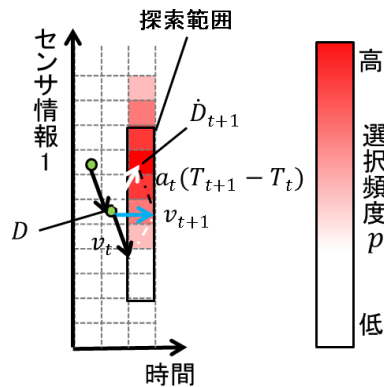


図 2.4.1 動作生成の様子

$$k_t = \frac{p_{t+1,max}}{\sum_{i=1}^n (p_{t+1,i} \times w_{t+1,i})} \times \alpha \quad (2.4.1)$$

$$a_t = \frac{k_t \times (D_{t+1} - D_t)}{m} \quad (2.4.2)$$

$$v_{t+1} = v_t + a_t(T_{t+1} - T_t) \quad (2.4.3)$$

$$D_{t+1} = D_t + v_{t+1}(T_{t+1} - T_t) \quad (2.4.4)$$

$p_{t+1,max}$: 選択頻度が最も高いセル(移動先探索範囲の中で)

$w_{t+1,i}$: セル($t + 1, i$)が探索範囲に含まれる場合 1, 含まれない場合は 0

D : ロボットの状態

D_{t+1} : 選択頻度が一番高いセルの中心

m : 仮想球の質量

α : 知識空間から受ける力の割合

2.5 Motion Space の問題点.

Motion Space には、知識空間の構成に動作時間の軸が含まれているという問題点が存在する。動作時間とは、ロボットの動作が開始されてから経過した時間である。動作時間を用いた知識空間には、動作を時系列順に管理できる利点がある。反面、いくつかの不具合が存在する。

1つは、動作時間を知識空間の要素として扱うには、動作時間を事前に設定する必要があることである。知識空間はセンサ情報の軸と時間軸で構成された空間である。センサ情報の軸の範囲は、センサ能力によって様々だが、センサ能力の限界を範囲とすればよい。センサ情報の軸は教示動作によって変化することはなく、固定である。しかし、動作時間の軸は教示動作によって各々異なる。教示動作が設定した時間より長い場合や短い場合に不具合が発生する。教示動作の知識化が未完になる可能性や、余分な動作まで知識化する可能性がある。しかし、教示動作の動作時間を精確に把握することは困難と考えられる(図2.5.1)。

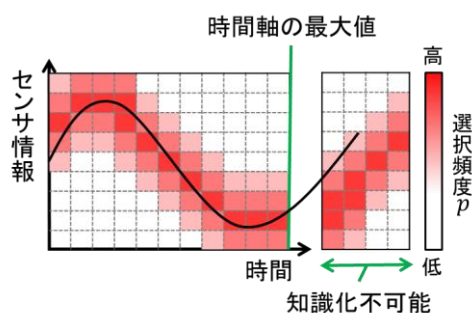


図 2.5.1 動作時間設定による不具合

また、教示動作が同じでも開始時刻が異なってしまう場合がある。教示動作が同一の周期をもつ一意な動作であったとしても、動作の教示時開始時刻が異なると同一の動作を異なる動作として知識化する。同一動作を知識空間に知識化したとしても、教示動作を上手く生成できない不具合を起こす(図2.5.2)。

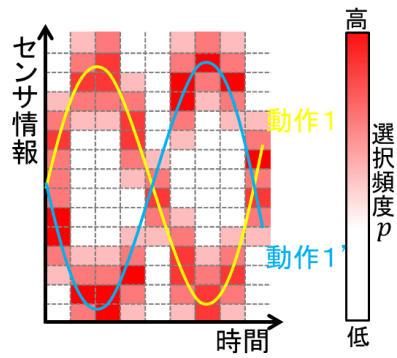


図 2.5.2 同一動作の教示開始が異なる知識化

さらに、動作生成時にも不具合がある。生成開始時に取得した状態が知識空間の動作時間途中にある場合でも、知識の開始時刻付近の状態を目標として動作を生成する。そのため、動作開始時のロボット自身の状態や環境に適した動作を生成することができない。

第3章 Motion Space TS 及び提案手法

本章では、先行研究の問題点を解決するために提案された Motion Space TS という手法と、今回提案する動作生成手法について説明する。最初の節で Motion Space TS の概要について説明する。次に Motion Space TS の動作知識化部について説明する。そして、Motion Space TS における動作生成のアプローチを説明し、提案手法である Motion Space TS の動作生成を説明する。

3.1 Motion Space TS の概要

Motion Space TS は、Motion Space の問題点を解決するために提案された。大きな変更点は、知識空間の構成である。Motion Space の知識空間を利用する際、2.5節で前述した不具合が存在する。その問題点は、知識空間の構成に動作時間の軸を用いていることにある。Motion Space TS では、動作時間の軸がない知識空間で動作知識化・動作生成を行う。

動作時間の軸が取り除く場合、代用となる情報が必要になる。Motion Space の知識空間において、動作時間は現時刻のセンサ情報による状態と次時刻の状態の前後関係を担っている。動作時間の軸が存在することで、動作知識化において動作の前後関係を保持した動きの知識を形成している。動作生成において次時刻の状態を正しく参照することで、次時刻の動作を生成することができる。状態の前後関係について図3.1.1に示す。

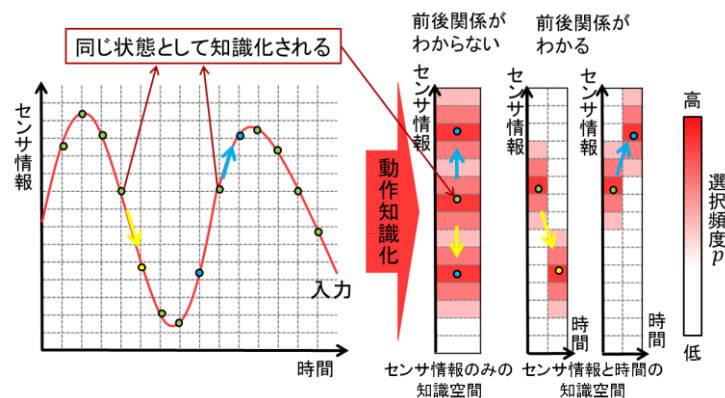


図 3.1.1 状態の前後関係

状態の前後関係を補うため、Motion Space TS の知識空間では、センサ情報とそのセンサ情報の時間微分を構成要素として用いる。センサ情報の時間微分を用いることで状態の前後関係を表現できる。動作知識化においては、同じ状態で知識化される状態をセンサ情報の時間微分によって別の状態に分けることができる。動作生成においては、次時刻と現時刻のセンサ情報の変化分が、次時刻のセンサ情報の時間微分と正負が逆である場合、その次時刻の状態が誤りであることがわかる(図3.1.2)。

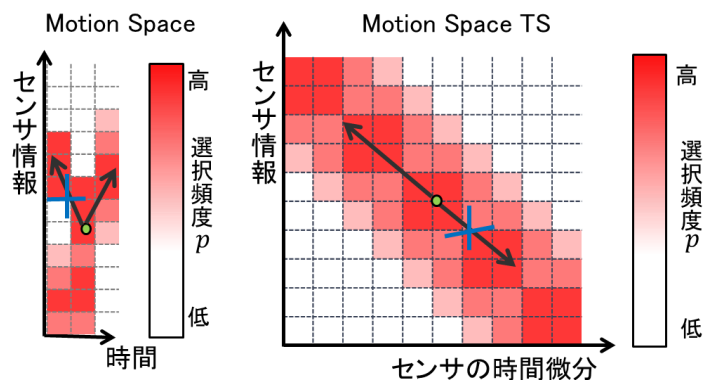


図 3.1.2 Motion Space と Motion Space TS の次時刻の動作の表現の違い

Motion Space TS は、Motion Space と同様の利用システムが存在する。中身の処理は変更を加えているが、システムの構成は同様である。また、手法の特性である専門知識を持たない利用者でも教示が容易である点と動きの知識を形成している点も Motion Space と同様である。

Motion Space TS の知識空間は、時間軸を除去したことで知識空間の大きさが一定になる。教示動作の時間に左右されず、知識空間に動作を知識化できる。また、任意の動作時間の教示動作を同じ知識空間に知識化することができる。これにより、Motion Space の不具合が解消できると考えられる。

知識空間をセンサ情報とその時間微分を用いるため、動作知識化部・動作生成部の入力と動作生成部の出力はセンサ情報とその時間微分になる(図3.1.3)。

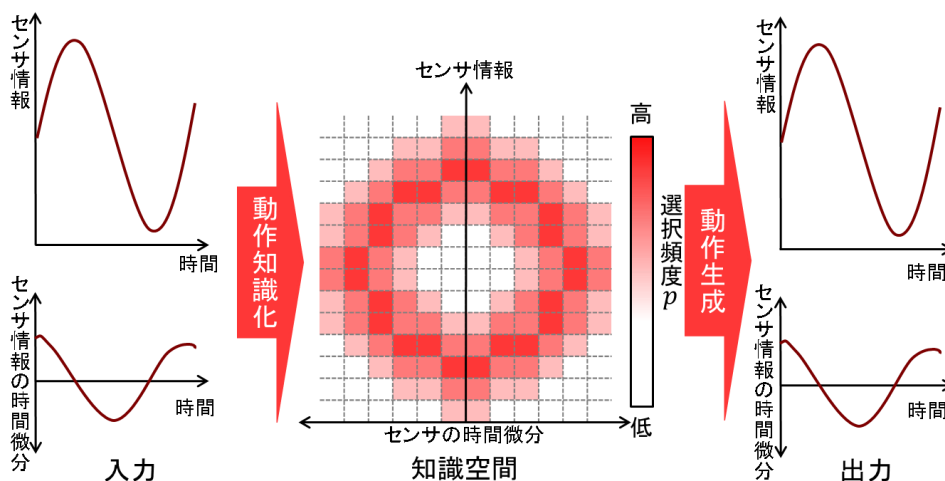


図 3.1.3 Motion Space TS の知識空間とその利用の様子

センサ情報の時間微分はセンサ情報に対して動作知識化・動作生成共通のサンプリングタイムを用いる。

3.2 Motion Space TS の動作知識化部

サンプリングタイムごとの動作知識化の手順を説明する。システムはサンプリングタイムごとにロボットの状態をセンサ値で取得する。取得したセンサ値とその時間微分を基に、知識空間上の点を決定する。そして、その点を含むセルと付近のセルに選択頻度を円状に与える。与える選択頻度は取得した状態の点が含まれるセルに高い値を加算し、周囲のセルは距離に応じて反比例した値を加算する。この手順を、サンプリングタイムごとに行い、動作を知識化する。

入力されたセンサ値とその時間微分の組である状態に対して行われる、知識化の選択頻度加算処理を説明する。最初に、センサによって取得した状態が知識空間上を示す点を判別する。その点をあらかじめ設定した選択頻度を加算する範囲 R の中心として考える。半径 R の円状範囲内の各セルへ選択頻度を加算する。(3.2.1)式によって範囲内の各セルに与える選択頻度 $u_{l,m}$ を計算する。範囲内の各セルは、(3.2.1)式の通り、中心からの距離によって、与える選択頻度は異なる。中心に近いほど与える選択頻度は高く、離れるほど低い。図3.2.1の例で見ると、セル番号 l,m が交差するセルに一番高い選択頻度が加算される。計算した選択頻度 $u_{l,m}$ は、(3.2.2)式を用いて更新する。

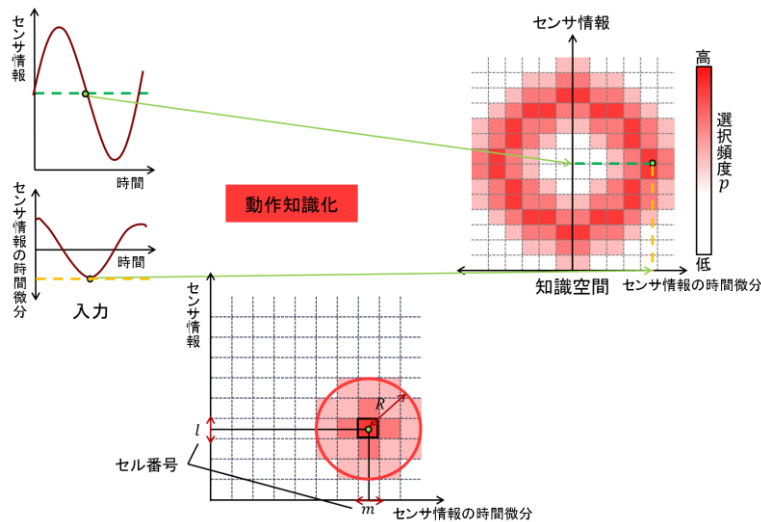


図 3.2.1 動作知識化の手順

$$u_{l,m} = \frac{C}{r+1} \dots (3.2.1)$$

$$p_{l,m} = p_{l,m} + u_{l,m} \dots (3.2.2)$$

$u_{l,m}$: $p_{l,m}$ に加算する選択頻度

C : 一度の加算で加える選択頻度の最大値

r : 中心のセルを0とした、中心からの加算する対象セルとの距離

$p_{l,m}$: セルの選択頻度

l,m : 各々の状態軸の任意のセル番号

3.3 提案手法のアプローチ

提案手法である Motion Space TS の動作生成について考える。Motion Space 及び Motion Space TS が生成する動作には考慮すべき点がある。1つは、知識空間の選択頻度が高い場所を目指す動作である。知識空間の選択頻度の高さは、教示動作の軌跡が多く通過した場所であり、利用者の希望する教示動作に近いと考えられる。動きの知識による動作を生成するためには、次時刻の動作生成において選択頻度の高い場所の影響を受けるようにする。

もう1つは、1時刻前からのロボットの実動作を考慮することである。動作生成部が次時刻の目標状態を出力して、実際のロボットがその目標状態へ動作できたとは限らない。1時刻前から現状態の動作が動作生成部の出力した状態に到達していない場合、次時刻の動作にはそれを考慮する必要がある。考慮しなければ、図3.2.1で示している選択頻度の高い場所のみを目指し、ロボットのハードの制約を超えた無理な動作を生成する可能性がある。

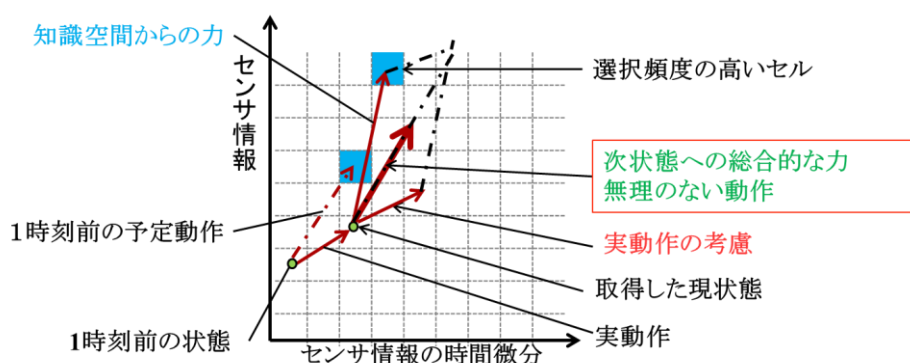


図 3.3.1 1時刻前の実動作の考慮

そこで、Motion Space の動作生成と同様に知識空間上で仮想球運動を行うことで動作を生成する。仮想球を運動させ、仮想球の次時刻の状態が、動作生成部が出力するロボットの次時刻の状態となる。図3.2.1のように1時刻前の状態から現状態までの変化を仮想球の速度ベクトルとして扱う。この速度ベクトルを仮想球の運動として次時刻の仮想球運動に用いることでロボットの実動作を考慮する。

また、知識空間の選択頻度の高い場所を目指す動作は知識空間から仮想球へ仮想的な引力を与えることで再現する。この力は知識空間の各セルがもつ選択頻度の大きさに比例する。仮想球の周囲から知識空間の力を受けることで選択頻度の高い場所を目指す力のベクトルを仮想球に与える。仮想球に2つのベクトルを与えることで、動きの知識とロボットの実動作を考慮した次時刻の動作を生成できる。

提案手法では知識空間からの力の受け方において Motion Space から変更している。その違いを図3.2.2に示す。この図で、 S_1 はセンサ情報、 \dot{S}_1 はセンサ情報の時間微分、 t は時間を示している。Motion Space において知識空間から受ける力の計算には、探索範囲の中で一番選択頻度が高いセルの選択頻度を用いる。しかし、この方法では探索範囲内の他のセルの選択頻度、または仮想球周囲のセルの選択頻度は考慮に入れておらず、動きの知識の活用機会を減らしている。また、Motion Space TS の知識空間では円状に選択頻度を加算しており、周囲の状態は次時刻の状態として目標とする余地がある。そのため、仮想球が、あるセルに存在するとき、知識空間から受ける仮想的な引力は、探索範囲全体から受けるように考える。これにより、動きの知識をより考慮した次時刻の動作を生成できると考えられる。

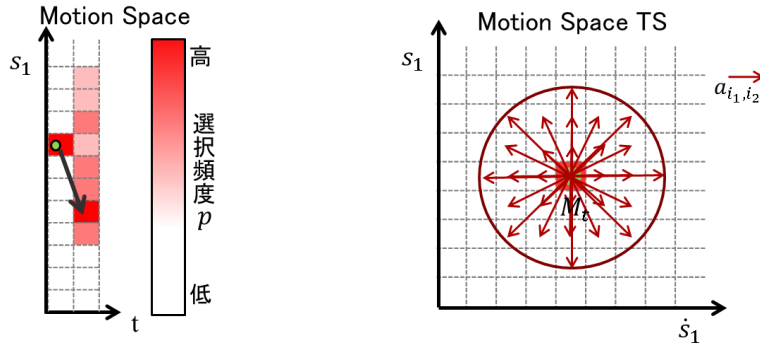


図 3.3.2 知識空間から受ける引力において従来手法と提案手法の違い

3.4 提案手法：Motion Space TS の動作生成部

Motion Space TS における提案する動作生成部について説明する．動作の生成手法として，知識空間上で仮想球運動を行う．取得したセンサ値とその時間微分から現在の状態として対応する知識空間上の点に仮想球を置く．仮想球運動を行い，次時刻の仮想球の状態を求めることで，ロボットの次時刻の状態は出力する．

新しい動作生成法の手順を説明する．まず，ロボットから取得した状態を知識空間上にプロットする．その点 M_t を中心に設定したセル分 R の半径をもつ円状の範囲を決定する．円状の範囲内の各々のセルから力を計算する(図3.4.1)．各々のセルから受ける力の係数 k_{i_1,i_2} を(3.4.2)式で計算する．これは現在存在するセルと選択したセルとの差を計算している．現在より高い選択頻度をもっていれば力は正になり，低ければ負になる．状態係数 k_{i_1,i_2} を基に，各セルから受ける力による加速度 a_{i_1,i_2} を(3.4.1)式で計算する．このとき選択したセルとの距離を絶対値でとり，探索範囲半径から引く．Motion Space の式では，ばね運動の式より距離が大きいくほど力が強くしていた．それを近いほど強いのが自然と考えた．また， k_{i_1,i_2} によって正負を決定するため，距離の要素を絶対値とした．

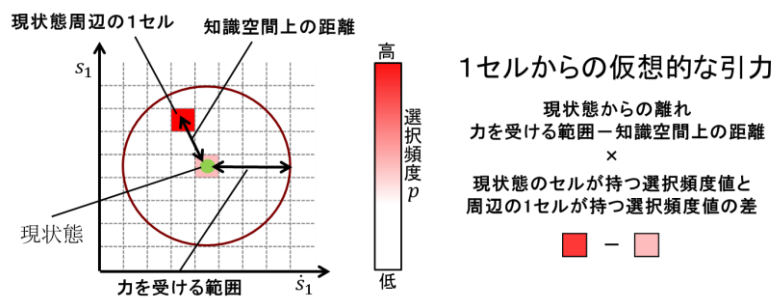


図 3.4.1 各セルからの仮想的な引力

$$a_{i_1,i_2} = \frac{k_{i_1,i_2} \times (R - |N_{i_1,i_2} - M_t|)}{m} \dots (3.4.1)$$

$$k_{i_1,i_2} = (p_{i_1,i_2} - p_t) \times \alpha \dots (3.4.2)$$

a_{i_1,i_2} : 円状範囲内の各セルから受ける仮想的な力による加速度

- N_{i_1, i_2} : 円状範囲内の各セルの中心
- k_{i_1, i_2} : 円状範囲内の各セルから受ける仮想的な力の係数
- M_t : ロボットが取得した現状態のプロット点
- p_{i_1, i_2} : 円状範囲内の各セルの選択頻度の値
- p_t : 現在状態のセルの選択頻度の値
- α : 円状範囲内の選択頻度全体に対する割合
- R : 知識空間からの力を参照する範囲の半径

計算した各セルから受ける力を1つの力として扱う。各セルから受ける力による加速度 a_{i_1, i_2} を(3.4.3)式で1つの加速度 a_t とする。

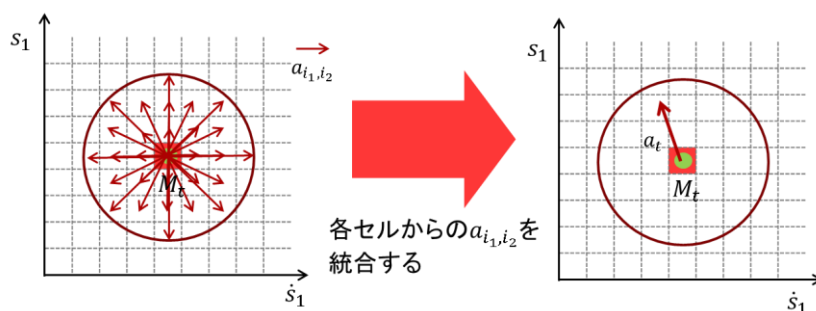


図 3.4.2 各セルからの力を統合する概要図

$$a_t = \sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_2} a_{i_1, i_2} \dots (3.4.3)$$

a_t : 知識空間から受ける仮想的な力の加速度の総和

次に、1時刻前のロボットの実際の動作を考慮した仮想球の速度ベクトルを計算する。この速度ベクトルは、 ΔT 前の状態 M_{t-1} と現在の状態 M_t から考えられる。2点間の位置関係から、 M_{t-1} から M_t までの仮想球の運動を計算し、その運動の速度ベクトル v_t' を求める(図3.4.3)。

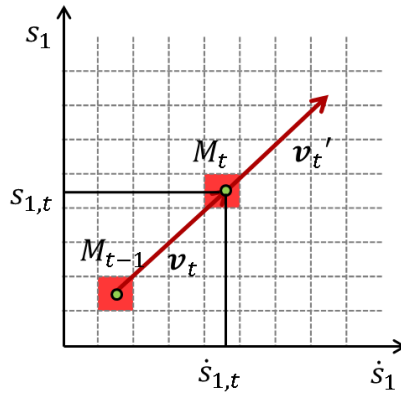


図 3.4.3 1 ステップ前から受ける速度

前動作からの速度ベクトル v_t' と、周囲セルからの力による加速度 a_t を(3.4.4)式で変換した速度ベクトルを合成して、仮想球の目標状態への方向と速度ベクトル v_{t+1} を決定する。その後、現状態 M_t と、目標状態への速度 v_{t+1} を用いて(3.4.5)式より次状態 M_{t+1} を求める(図3.4.4)。この仮想球の次状態がロボットの次時刻の状態を示し、動作を決定する。

$$v_{t+1} = v_t' + a_t \times \Delta T \dots (3.4.4)$$

$$M_{t+1} = M_t + v_{t+1} \times \Delta T \dots (3.4.5)$$

v_t' : 前動作による力の速度

v_{t+1} : 知識空間上の目標へと向かう運動の方向と速度

M_{t+1} : 仮想球の次状態

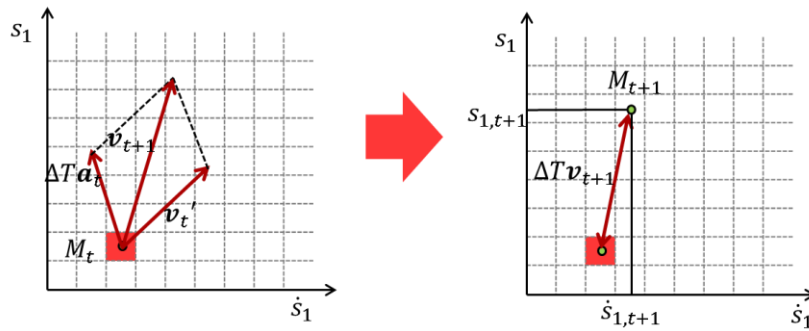


図 3.4.4 2つの力による速度の合成

第4章 実験

前章で提案した動作生成部を適用した仮想のロボットを用いた、動作生成のシミュレーション実験を行う。

4.1 実験目的

本実験の目的は、提案した動作生成部を適用した仮想のロボットの動作生成をシミュレーション上でを行い、生成動作の検証を行うことである。まず、提案手法を用いることで知識空間から動作を生成できることを確認する。そして、同一周期で開始点の異なる動作を同一の知識空間から生成できることを確認する。次に、生成動作に対して、知識空間作成に使用した入力動作と比較を行い、再現性を検証する。最後に、先行研究の Motion Space による生成動作と再現性を比較し、提案手法の有効性を検証する。

4.2 実験環境

本節では、ロボットの動作生成を行う実験のシミュレーションについて説明する。シミュレーション上で仮想的に用意したロボットの動作を再現する。シミュレーションの詳細について説明する。

4.2.1 シミュレーションで想定するロボット

本実験で想定するロボットは、身体の構成にアクチュエータとセンサを1個ずつ搭載している。アクチュエータの自由度は、円 $360[\text{deg}]$ に対して、その稼働範囲を $-120[\text{deg}] \sim 120[\text{deg}]$ で全体 $240[\text{deg}]$ とする。センサは、アクチュエータの絶対角度を値として取得することが可能である。

アクチュエータの動作は目標角度と動作時間の2つで指定する。指定された目標角度へ指定された動作時間までに到達する。目標角度に対して動作時間を平均速度で動作する。つまり、目標角度と動作時間によってその速度が決定する。目標角度は絶対角度を指定し、動作時間は指令後から目標角度到達までの時間で指定する。

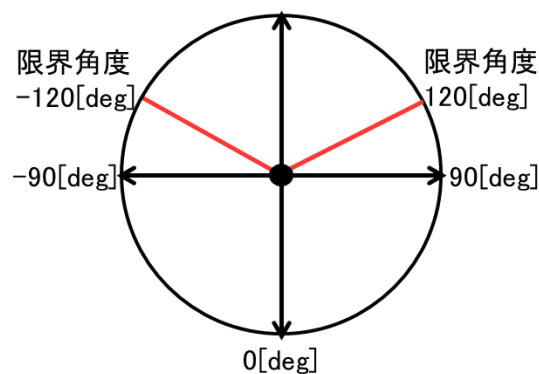


図 4.2.1.1 アクチュエータの自由度

4.2.2 シミュレーション概要

シミュレーションの概要を図4.2.1.1に示す。シミュレータの役割は提案手法を適用した仮想のロボットの動作を再現することである。動作生成を行う際、提案手法へロボットの状態を入力し、提案手法が出力した次時刻の状態を基にロボットへ動作指令を行う。その後、動作指令を目標にロボットを動作させる。この工程を指定した出力時間のサンプリング数だけ繰り返す。提案手法の入力と出力はサーボモータの角度と角速度からなるロボットの状態を使用する。生成した動作の結果は、センサが取得したロボットの状態である角度を扱う。

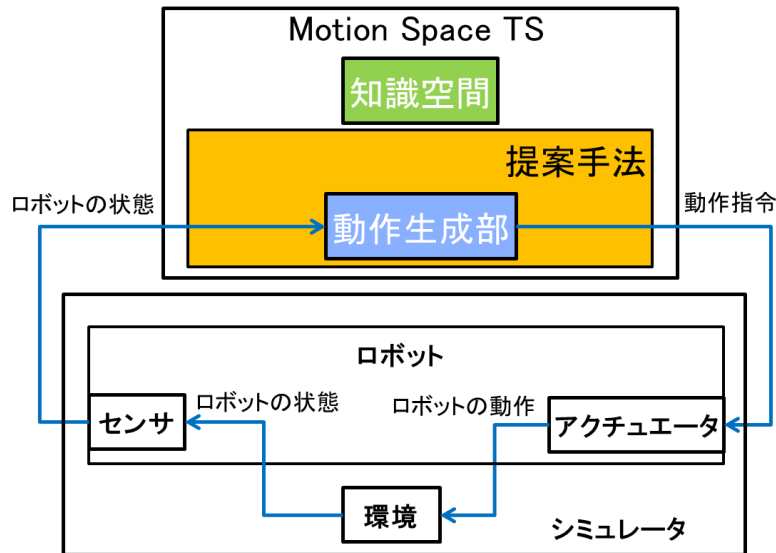


図 4.2.2.1 シミュレーション実験概要図

4.2.3 仮想ロボットの動作指令計算

提案手法からの出力であるロボットの次時刻の状態は、センサ値とセンサ値の時間微分の組である。そのため、現時刻のセンサ値と次時刻のセンサ値の差分を時間微分した値が、次時刻のセンサ値の時間微分になる必要がある。今回のシミュレーションで使用する仮想のロボットの場合、アクチュエータの角度と角速度がそれに該当する。しかし、次時刻のロボットの状態を示す仮想球にその必要はない。そのため、出力される両者の値に時間微分の関係がない可能性が大きくなる。そのため、下記の計算式で次時刻の角度と角速度の関係性を補完する。補完することで、次時刻の角度と角速度を次時刻に入力として使用できる。

動作指令の計算について説明する。現時刻 T の角度 θ_T と動作生成部が決定した次時刻 $T + \Delta T$ の角度 $\hat{\theta}_{T+\Delta T}$ の変化分に注目する。(12)式より、動作生成部が決定した次時刻 $T + \Delta T$ の角速度 $\hat{\theta}_{T+\Delta T}$ で角度 $\hat{\theta}_{T+\Delta T}$ へ到達するのにかかる実時間 Δt を求める。 Δt とサンプリングタイム ΔT が異なる場合、角度 $\hat{\theta}_{T+\Delta T}$ と角速度 $\hat{\theta}_{T+\Delta T}$ は時間微分の関係にない。今回の実験では、角速度 $\hat{\theta}_{T+\Delta T}$ に合わせて、次時刻の角度 $\theta_{T+\Delta T}$ を補完する。(13)式により、角速度 $\hat{\theta}_{T+\Delta T}$ でサンプリングタイム ΔT の間動作することで到達する角度 $\hat{\theta}_{T+\Delta T}$ に到達する角度 $\theta_{T+\Delta T}$ を求める。角度 $\theta_{T+\Delta T}$ と ΔT を動作指令とし、角度 $\theta_{T+\Delta T}$ と(14)式より求めた角速度 $\hat{\theta}_{T+\Delta T}$ を次時刻の入力として用いることでセンサ値の時間微分の関係性を補完する。

$$\Delta t = \frac{\hat{\theta}_{T+\Delta T} - \theta_T}{\hat{\theta}_{T+\Delta T}} \quad (12)$$

$$\theta_{T+\Delta T} = \theta_T + \hat{\theta}_{T+\Delta T} \cdot \frac{\Delta T}{\Delta t} \quad (13)$$

$$\dot{\theta}_{T+\Delta T} = \hat{\theta}_{T+\Delta T} \quad (14)$$

4.3 実験概要

4.3.1 提案手法による動作生成の確認とその再現性

1つ目の概要について説明する．初期角度が異なる同一の周期動作2つを1回ずつ知識化した知識空間を入力とする．その後，初期角度が90[deg]と−90[deg]の動作生成を行う．生成動作を入力動作と比較することで，同一知識空間から初期角度が異なる2つの周期動作が生成可能であることを確認する．

図4.3.1.1に動作生成に使用した知識空間に知識化した2つの入力動作をグラフに示す．左の赤色のプロットは，周期10[s]で初期角度90[deg]の cos 波の入力動作を示している．右の黄緑色のプロットは，周期10[s]で初期角度−90[deg]の cos 波の入力動作を示している．動作生成に使用した知識空間を図4.3.1.2に示す．赤色の濃さが選択頻度値の高低を示している．周期動作の知識化したことにより選択頻度の存在する部分が円状の軌跡を描いている．

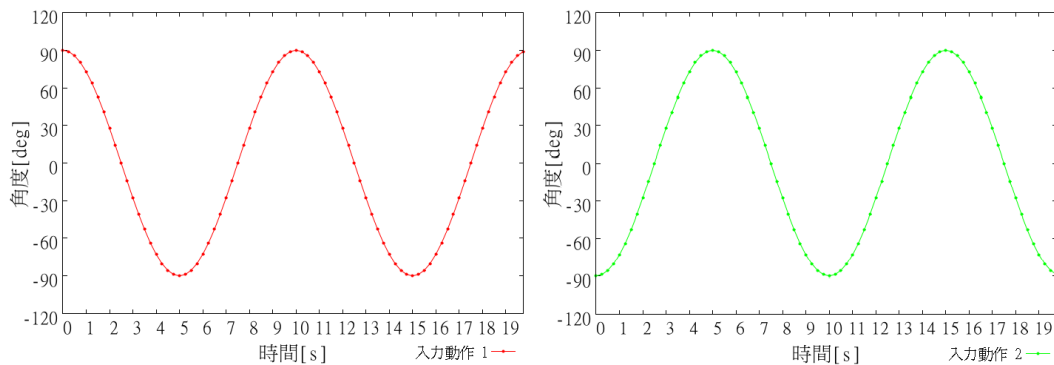


図 4.3.1.1 動作生成に使用した知識空間に知識化した 2 つの入力動作

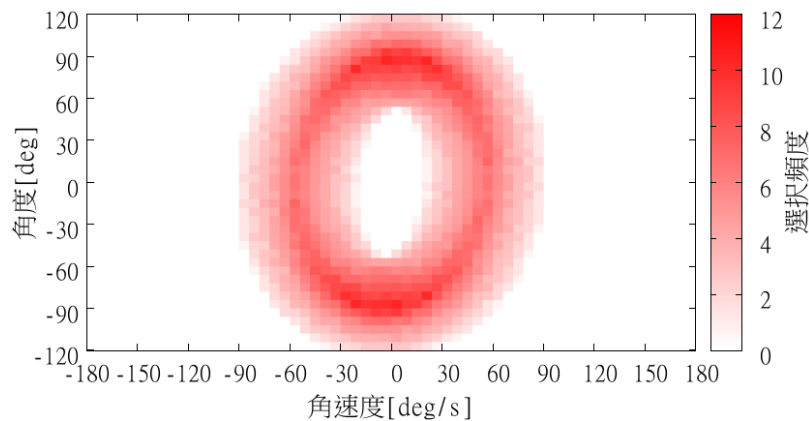


図 4.3.1.2 動作生成に使用した知識空間

提案手法で用いた実験のパラメータを表4.3.1.1に示す。動作時間は、入力に使用した知識化動作の時間であり、出力動作の時間でもある。サンプリングタイムは、動作生成を行う際に次時刻の動作を決定する時間間隔である。選択頻度加算範囲の半径は、入力に使用した知識空間を形成した際、1状態入力に対して選択頻度を加算する状態点を中心とした円状範囲の半径である。選択頻度探索範囲の半径は、動作生成時仮想球が受ける仮想引力が作用する仮想球を中心とした円状範囲の半径である。選択頻度加算最大値は、入力に使用した知識空間を形成した際、1状態点入力に対して1セルに加算可能な選択頻度の最大値である。知識空間の角度範囲・角速度範囲は、知識空間の各軸の最大値を示している。1セルの角度範囲・角速度範囲は、セル1つ分が扱うセンサ値の範囲を示している。傾斜係数及び仮想球の質量は、動作生成時に使用するパラメータである。

また、動作生成を行う際、初期値を与えている。初期角度90[deg]の出力動作では0[s]～0.25[s]にかけて角度が90[deg]から89[deg]に変化するように与えている。初期角度-90[deg]の出力動作では0[s]～0.25[s]にかけて角度が-90[deg]から-88.8[deg]に変化するように与えている。そのため、差分を比較する際には、0[s]～0.25[s]の角度を含めていない。

表 4.3.1.1 実験パラメータ

パラメータ	記号	値
動作時間[s]	-	0.0～19.75
サンプリングタイム[s]	ΔT	0.25
選択頻度加算範囲の半径[セル]	Q	6
選択頻度探索範囲の半径[セル]	R	5
選択頻度加算最大値	C	1.0
知識空間の角度範囲[deg]	-	$-120 \leq \theta \leq 120$
知識空間の角速度範囲[deg/s]	-	$-180 \leq \dot{\theta} \leq 180$
1セルの角度範囲[deg]	-	6
1セルの角速度範囲[deg/s]	-	6
傾斜係数	α	0.13
仮想球の質量	m	1

4.3.2 提案手法による動作生成の有効性

2つ目は、従来手法の Motion Space による出力動作の再現性と、提案手法の出力動作の再現性を比較することで有効性を検証する。Motion Space において図4.3.1.1に示した初期角度90[deg]の周期動作が知識化された図4.3.2.1の知識空間を入力に使用する。その知識空間から Motion Space によって動作生成した図4.3.2.2の出力動作を、図4.3.1.1に示した初期角度90[deg]の周期動作と比較し、再現性を求める。その再現性を4.4.1節で求める提案手法の再現性と比較することで提案手法の有効性を判断する。

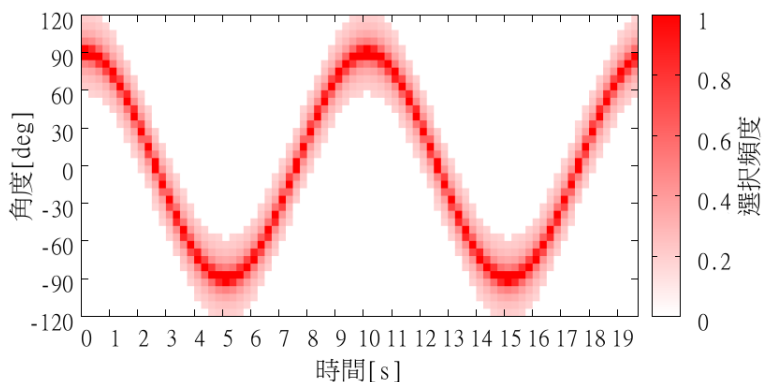


図 4.3.2.1 初期角度 90[deg]の周期動作を知識化した Motion Space の知識空間

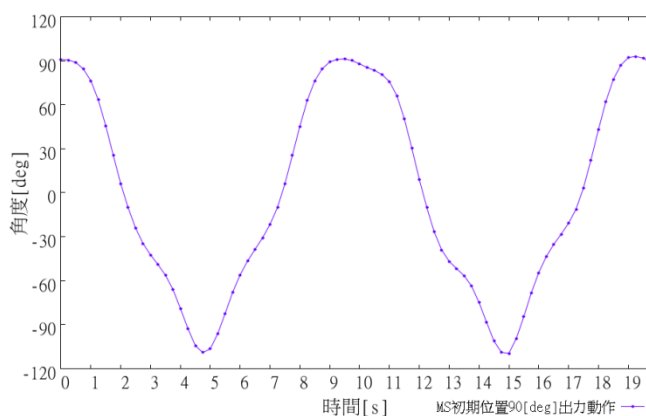


図 4.3.2.2 上記の知識空間を入力とした Motion Space による出力動作

4.4 実験結果

4.4.1 提案手法による動作生成の確認とその再現性

図4.3.1.2の知識空間から初期角度90[deg]によって生成した動作を表示したグラフを図4.4.1.1に示す。水色のプロットが出力動作を示している。0[s]のとき角度90[deg]から動作が開始され、角度が減少している。5[s]のとき角度の減少が終わり、その後10[s]まで増加する。角度が約5[s]間隔で増加と減少を繰り返していることから周期動作を生成していることがわかる。しかし、2回目の負の値の最大振幅は14[s]～15[s]の間で確認できる。また、2回目の正の値の最大振幅も約19[s]付近で確認できる。このことから出力動作の周期は一定ではなく、動作時間の長さに反比例して周期が短くなっていることがわかる。

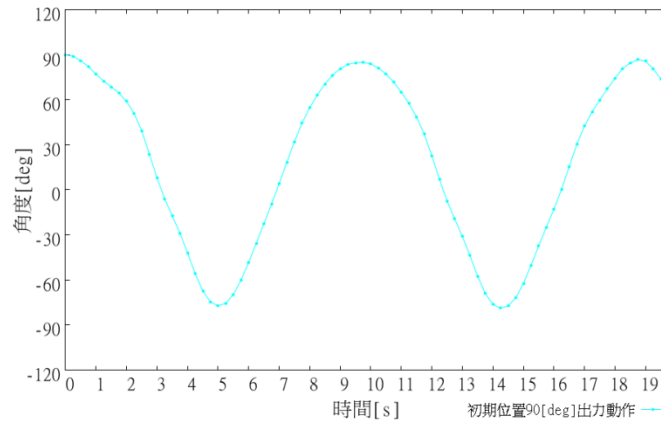


図 4.4.1.1 初期角度90[deg]による出力動作

図 4.4.1.1 で示される出力動作を図 4.3.1.2 で示した知識空間上に表示したグラフを図 4.4.1.2 に示す。黒色のプロットが出力動作を示している。選択頻度の高い部分を出力動作の軌跡が通過していることがわかる。また、知識空間上に円状に形成された知識空間と同じく、出力動作も円状にその軌跡を描いている。

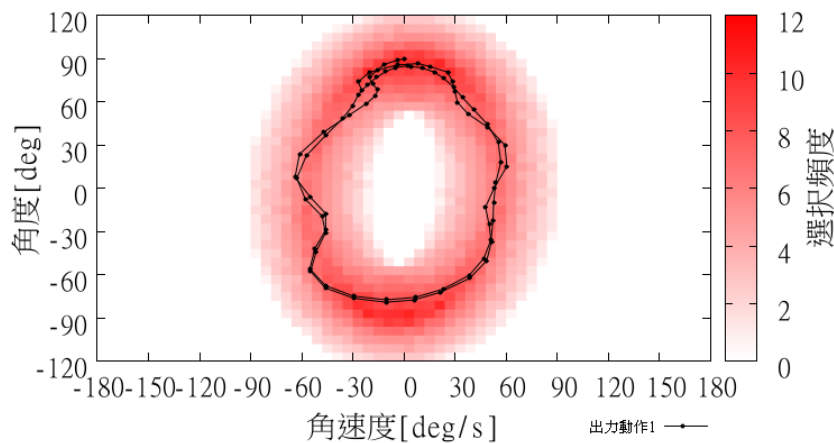


図 4.4.1.2 初期角度90[deg]による出力動作(知識空間上)

次に、初期角度 90[deg]による出力動作と入力動作を同時に示したグラフを図 4.4.1.3 に示す。赤色のプロットが入力動作を示し、水色のプロットが出力動作を示す。2つの動作を比較する上で周期と振幅に注目する。出力動作は入力動作と比較して、最大振幅の絶対値が小さいことがわかる。また、特に、負の角度においてその傾向が確認できる。その影響で出力動作の周期が入力動作より短くなっている可能性がある。振幅が小さいために、周期が短くなっていると考えられる。

2つの動作の差分の絶対値をとり、その最大値、最小値、平均を表 4.4.1.1 に示す。動作時間の長さに反比例して出力動作の周期が短くなっているため動作時間の後半に差分の最大値が示されている。同様に、動作開始の付近で差分の最小値が示されている。

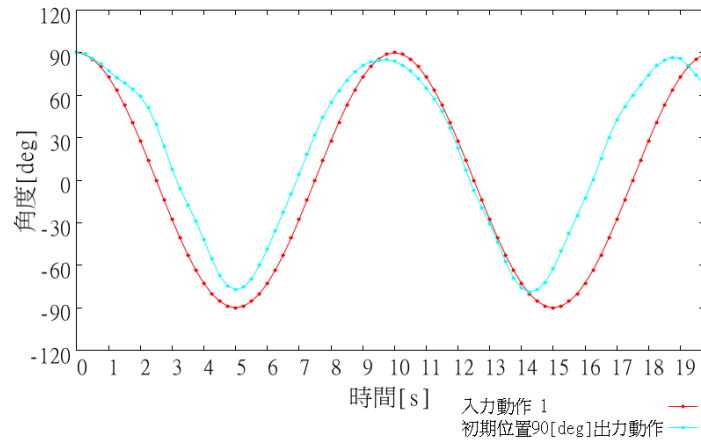


図 4.4.1.3 初期角度 90[deg]の出力動作と入力動作

表 4.4.1.1 初期角度 90[deg]の入力動作と出力動作の差分

	時間[s]	角度[deg]
最大差分	16.75	71.0599
最小差分	0.50	0.480034
平均差分	-	23.32176

次に、図4.3.1.2の知識空間から初期角度 -90 [deg]によって出力した動作を表示したグラフを図4.4.1.4に示す。紫色のプロットが出力動作を示す。0[s]のとき角度 -90 [deg]から動作が開始され、角度が増加している。5[s]のとき角度の増加が終わり、その後減少する。その後も増減を繰り返していることから周期動作を生成していることがわかる。しかし、振幅の最大値を確認できる間隔が短くなっている。1回目は約5[s]付近、2回目は9[s]付近、3回目は14[s]付近、4回目は18[s]付近で確認できる。このことから周期は一定ではなく、動作時間の長さに反比例して周期が短くなっていることがわかる。

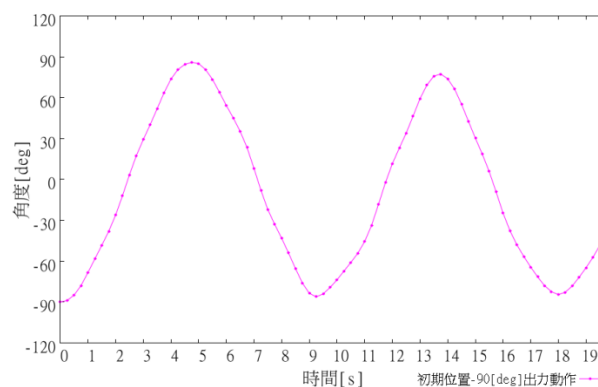


図 4.4.1.4 初期角度 -90 [deg]による出力動作

図 4.4.1.4 で示される出力動作を図 4.3.1.2 で示した知識空間上に表示したグラフを図 4.4.1.5 に示す。黒色のプロットが出力動作を示している。選択頻度の高い部分を出力動作の軌跡が通過していることがわかる。また、知識空間上に円状に形成された知識空間と同じく、出力動作も円状に軌跡を描いている。

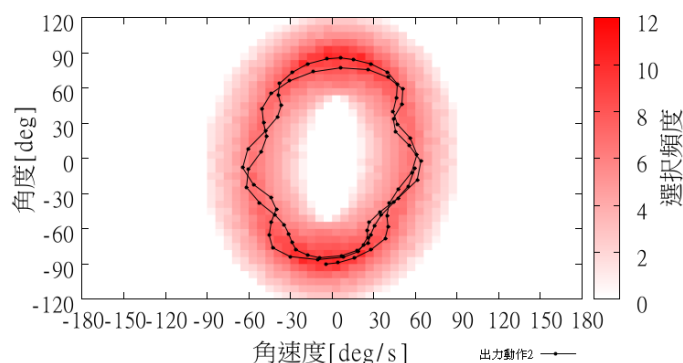


図 4.4.1.5 初期角度 $-90[\text{deg}]$ による出力動作(知識空間上)

次に、初期角度 $-90[\text{deg}]$ による出力動作と入力動作を同時に示したグラフを図 4.4.1.6 に示す。黄緑色のプロットが入力動作を示し、紫色のプロットが出力動作を示す。2つの動作を比較する上で周期と振幅に注目する。出力動作は入力動作と比較して、最大振幅の絶対値が小さいことがわかる。その影響で出力動作の周期が入力動作より短くなっている可能性がある。振幅が小さいために、周期が短くなっていると考えられる。特に、14[s]付近以降の動作から周期の大きなズレを確認できる。

2つの動作の差分の絶対値をとり、その最大値、最小値、平均を表 4.4.1.1 に示す。動作時間の長さに反比例して出力動作の周期が短くなっているため動作時間の後半に差分の最大値が示されている。同様に、動作開始の付近で差分の最小値が示されている。

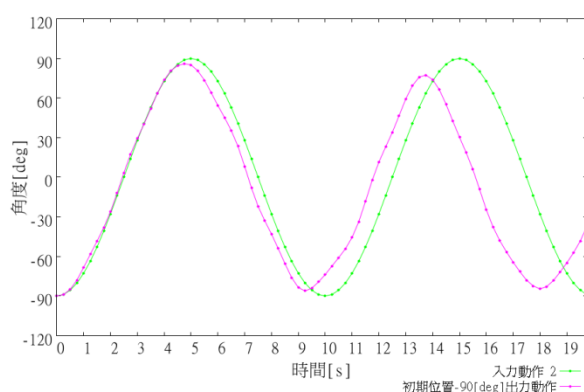


図 4.4.1.6 初期角度 $-90[\text{deg}]$ による入力動作と出力動作

表 4.4.1.2 初期角度 $-90[\text{deg}]$ の入力動作と出力動作の差分

	時間[s]	角度[deg]
最大差分	16.25	101.248
最小差分	3.75	0.020298
平均差分	-	27.38901

2つの周期動作を知識化した同一の知識空間を入力として、初期角度の異なる周期動作を生成したことから提案手法によって Motion Space の問題点を解決した動作が生成できることが確認できた。しかし、動作時間の長さに反比例して周期が短くなる問題も確認した。

4.4.2 提案手法による動作生成の有効性

従来手法である Motion Space において初期角度90[deg]の周期動作を知識化し、生成した出力動作の再現度と、提案手法の出力動作の再現度を比較する。入力動作と従来手法の出力動作と、提案手法の出力動作の3種類を同時に表示したグラフを図4.4.2.1に示す。赤色のプロットが初期角度90[deg]の入力動作を示す。水色のプロットが提案手法による初期角度90[deg]の出力動作を示す。紫色のプロットが従来手法による初期角度90[deg]の出力動作を示す。

3種類の動作を比較するとその振幅の大きさの違いを確認できる。従来手法の出力動作は振幅の最大値・最小値が入力動作を超過しているのに対して、提案手法の出力動作は振幅の最大値・最小値が入力動作を超過していない。

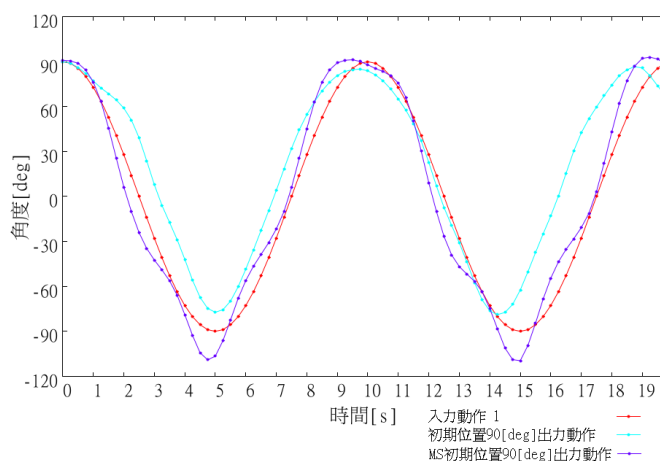


図 4.4.2.1 初期角度90[deg]の従来手法と提案手法の出力動作と入力動作

入力動作と各手法の動作の差分の絶対値をとり、その最大値、最小値、平均を表 4.4.2.1 に示す。提案手法の差分の最大値は、動作時間の長さに反比例して出力動作の周期が短くなっているため動作時間の後半に差分の最大値が示されている。対して、従来手法の出力動作は動作時間の後半というほどではない。同様に、提案手法は動作開始の付近で差分の最小値が示されているのに対し、従来手法では動作開始付近で示されているわけではない。さらに、差分の平均値は従来手法の方が提案手法より小さい。このことから、提案手法は従来手法より再現性に劣る結果が示されている。

表 4.4.2.1 初期角度 90[deg]の入力動作と各手法の出力動作の差分

	提案手法出力動作		従来手法出力動作	
	時間[s]	角度[deg]	時間[s]	角度[deg]
最大差分	16.75	71.0599	12.50	26.74668
最小差分	0.50	0.480034	13.75	0.010402
平均差分	-	23.32176	-	11.65314

振幅のみに着目し，入力動作と各手法の出力動作との差分を比較し，表 4.2.2.2 に示す．振幅を比較しているため，同一時間の角度を比較していない．振幅のみで比較した場合，提案手法の方がより差分が小さいことがわかる．また，前述どおり提案手法の振幅は入力動作の振幅を超過していない．そして，振幅の平均差分においては提案手法の方が小さい．このことから，出力動作の振幅のみに注目すれば，提案手法の再現性の方が高い．

表 4.4.2.2 初期角度 90[deg]の入力動作と各手法の出力動作の振幅のみの差分

振幅	入力動作		提案手法出力動作			従来手法出力動作		
	時間[s]	角度[deg]	時間[s]	角度[deg]	差分[deg]	時間[s]	角度[deg]	差分[deg]
1 回目	5.00	-90	5.00	-77.2699	12.73013	4.75	-109.017	19.01743
2 回目	10.00	90	9.75	85.11165	4.888351	9.50	91.28275	1.282753
3 回目	15.00	-90	14.25	-78.885	11.11504	15.00	-109.569	19.56864
平均					9.57784			13.28961

第5章 結論

5.1 本論文のまとめ

本研究では、Motion Space を適用したロボットへの動作教示と動作生成に着目した。Motion Space とは、ロボットに動作を動きの知識として記憶し、その知識を基に動作を生成する手法である。動きの知識とは人が動作を覚えるときに反復して動作を行うとき知識形成に着目している。複数の動作を1つの知識として知識化する。動きの知識として記憶されるため、環境の変化によって動作が変化しても動きとしての知識を基に動作を生成できる。この手法をロボットに適用することで環境の変化に適応した動作を生成することができる。また、ティーチングプレイバックに類する教示方法を適用している。この手法は利用者が専門知識を持たなくてもロボットに動作を教示できる手法である。ロボットは指示された動作を行うときの自身の状態をセンサで取得する。その取得した値を動作として記憶し、再生する手法である。Motion Space は教示が容易であることと、自動的に環境の変化に適応した動作を生成できることの2つを性質として併せ持つ。

Motion Space には動作時間による問題が存在する。それにより、動作時間を決定する必要がある不具合がある。教示動作が決定した時間より長くなれば、その分は切り捨てられ、短ければ無駄な動作を知識化してしまう可能性がある。また同じ周期をもつ動作が開始点の違いから別々の動作として知識化される不具合もある。その場合、動作を生成する際に教示した動作を生成できない可能性がある。これを解決するための手法として時間軸を用いない知識空間による手法 Motion Space TS が提案されている。状態の前後関係を表現していた時間情報の代用として取得したセンサ情報の時間微分を知識空間の構成に用いている。センサ情報の時間微分は状態の前後関係を補完することができる。現在、Motion Space TS において動作知識化部までは提案されている。本研究では、動作生成部を研究対象とし、時間軸のない知識空間から動作を生成する手法を提案した。特に、動作生成手法では知識空間上で現在の状態に対して周囲から総合的に選択頻度の高い場所への引力を受ける式を提案した。

本研究の提案手法の評価にはシミュレータによる実験で評価した。簡易的なシミュレータを作成した理由として Motion Space TS の提案が挙げられる。手法の検証に実機を使用することが最終目標だが、パラメータ等の調整には使用するには時間がかかる。また、新しい手法を提案した場合でもシミュレーションによって実機の前段階を置くことは手法の開発に有効であると考えられる。

シミュレーション実験を行い、仮想のロボットに手法を適用し、動作生成を行った。そして、提案手法で時間軸のない知識空間を基に動作が生成できることを確認した。また、周期動作において開始時刻が異なる動作を生成できることを確認した。従来手法の Motion Space の生成動作と比較して、総合的な再現度は劣るが、振幅における再現性の高さを確認した。

5.2 今後の展望

本節では、今回の提案手法の問題点とともに、その解決方法の予測を行う。また、今後の展望についても述べる。

提案手法における動作生成は入力動作と比較して周期が短くなる傾向が確認できた。周期の再現性の低さが目立つ結果となった。原因として提案した動作生成式では知識空間の選択頻度の高低差の影響が強過ぎる可能性がある。選択頻度の高い場所では入力動作に近い動作を生成していたが、低い場所では出力動作の軌跡の変化が入力動作より大きくなっていった。これは、動作生成に用いた仮想球運動において、高い場所と低い場所のバランスがとれておらず、高い場所に最適な式になっていた可能性がある。解決策としては、動作生成式のさらなる提案か動作知識化手法の改善が考えられる。動作生成手法は知識空間に大きく左右される。今後も、仮想球運動による動作生成を行うならば、適した動作知識化も検討の余地がある。

本研究の実験では、シミュレーションによって行われている。シミュレータ部で動作生成部の出力を仮想ロボットの指令に変換している。そのため、実験結果の動作生成にシミュレータの影響が含まれている可能性があり、提案手法自体の評価が満足していない可能性がある。しかし、**Motion Space TS** を実ロボットに搭載する際も全てのセンサ値とその時間微分が直接動作指令に使われる可能性は低い。

今回のシミュレーション実験を通して、**Motion Space TS** の使用法についての展望を述べる。**Motion Space TS** は動作生成の出力は次時刻の状態であり次時刻の動作の指針ではある。しかし、それ単体での使用を前提とせず、各ロボットにおいて実験のようなシミュレータ部を個別に加えることでロボットの動作生成を行えると考える。今回の実験で作成したシミュレータ部を様々なロボットごとに個別に作成し、シミュレーションを行う。それにより、シミュレータ部を洗練し、ロボットに搭載する。この一連の手順による **Motion Space TS** の実ロボットへの適用を今後の展望とする。

謝辞

本論文を結ぶにあたり、日頃より懇切丁寧なご指導・ご助言を賜りました倉重健太郎先生に深く感謝の意を表します。また、ご助言・ご指導いただいた畑中雅彦先生、本田泰先生、佐賀聡人先生に感謝の意を表します。そして論文の査読や助言をしていただいた認知ロボティクス研究室の木島康隆さん、北山直樹さん、杉本大志さん、梅津祐介さん、渋谷和さん、高泉昇太郎さん、三浦丈典さん、木村敏久君、平間経太君に感謝いたします。

参考文献

- [1]岡田昌史：“力学的情報処理による運動の表現と設計”，日本ロボット学会誌 Vol.25No.5 pp.678～683(2007)
- [2]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(2007)
- [3]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(2010)
- [4]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