

# 日常生活支援のための机上作業のモデル化 およびその認識と支援軌道の生成

佐藤 知正\* 久保寺 秀幸\* 原田 達也\* 森 武俊\*

## Modeling, Recognition and Supporting Trajectory Generation of Daily Object-handling based on Acquired Motion Models

Tomomasa Sato\*, Hideyuki Kubotera\*, Tatsuya Harada\* and Taketoshi Mori\*

This paper proposes a robotic assistance system for object handling based on imitative learning. At first, the system learns temporally short segments of motion called “motion primitives” from observation of human object handling tasks. Secondly daily human object-handling is recognized as a sequence of motion primitives. Then the occurrence of an appropriate assisting task defined as a sequence of motion primitives is predicted. Finally the corresponding assisting trajectory is generated from the sequence of motion primitives. The system is composed of such algorithms as object handling motion clustering, human motion recognition, assisting task prediction and trajectory generation, which are learned from human motion. On the other hand, the user specifies the tasks beforehand which the system should support. The validity of the proposed algorithms is confirmed through the experiment of object-handling assistance utilizing a cup.

**Key Words:** Motion Primitives, Sequential Data Clustering, Object-handling Recognition, Supporting Task Prediction, Trajectory Generation

### 1. 結 論

物品片付けなどの日常生活を、さりげなく支援するロボットが必要とされている。しかし人間の行動や使用環境は多様であり、あらかじめロボットの支援動作や環境データをプログラミングしておくことは難しい。そこで“模倣”による行動獲得の研究が近年行われている [1]~[5]。これらの研究の特徴は視覚等のセンサを用いて人間の行う作業を観察し、それを単位的な動作モデルで記述することである。こうすることで一連の軌道をそのまま再現する方法とは異なり、学習時と実行時の環境とが異なっても目的の作業を達成することが可能になる。しかしこれらの研究では、あらかじめ動作モデルの種類を与えているため新奇な作業が観察された場合、対応する動作モデルの諸機能を定義し追加する必要がある。

この問題を解決するために、ロボットが観察された作業を自動的に分類し動作モデルを獲得することが考えられる。筆者らは“pick-and-place”などの物体操作の多数回の観察から“lift-up”のような短い時間単位の動作要素を複数獲得し、それらの列として人間の物体操作を認識し対応するロボットの手先軌道を生

成する研究を行ってきた [6]。この短い時間単位の動作要素のことを「動作プリミティブ」と呼ぶ。同様に Ogawara らは複数物体が関連する物体操作の複数回の観察から重要な部分を発見し、重要な部分に注目した軌道生成により物体操作の再現を実現している [7]。

しかし、人間が指示することなしに必要なときにさりげなく支援するためには、作業を模倣するだけでなく次に行われる作業が支援すべきか否か前もって予測することが必要となる。支援すべき作業が何であるか観察から発見することは難しいが、それが発生するタイミングは典型性が存在する人間の行動パターンから予測することが可能である。

また物体操作を実行する際、特に環境との接触時には力フィードバックによる物体の制御が重要である。本研究ではこのような制御則をあらかじめ定義することなく、物体操作の視覚情報と同時に環境に加わる力情報を計測しておき観察から環境との接触時における力制御則を抽出することを試みる。

これより本研究の目的を、あらかじめルールを与えずに観察から得た知識をもとに支援を実行する汎用的なロボットシステムの構築を目指し、基本的な要素となる複数回の観察からの動作プリミティブに基づく物体操作の認識・予測・軌道生成手法の提案とする。さらに「コップを引き寄せて中身を飲み片付ける」という例題を挙げ、提案手法をロボットシステムに実装し、リアルワールドでも実演可能な手法であることを実証する。なお、

原稿受付 2005年3月28日

\*東京大学大学院情報理工学系研究科

\*The University of Tokyo, Graduate School of Information Science and Technology

この手法を提案するに当たり、いくつかの仮定を設ける。一つ目の仮定は、日常的な比較的ゆっくりとした支援を念頭に置く点である。二つ目の仮定は、任意の物体の把持・操作可能なロボットシステムが実現されていると考える点にある。三つ目は支援される人のメンタルモデルを考慮しないことである。一つ目の仮定は、既存の計算機、ロボット、センサでも実時間で実行するシステムを可能とするためである。二つ目の仮定は、任意の物体の把持・操作可能なロボットシステムが存在しないために、実機で支援軌道を生成する際に物体操作・力制御手法にある程度のアドホックな手法を許容することになる。また三つ目の仮定は、支援が支援された人にとって真の意味でうれしかどうかを判定しないということに当たり、最低限必要な支援のタイミングを発見し実行することを目標とすることになる。

次に他の模倣による行動獲得の研究と比較した本研究のポイントについて述べる。初めのポイントは、定性値表現による動作プリミティブの自動獲得にある。定性値表現を用いることにより、一定の単位時間により切り出した動作セグメント群から比較的シンプルなモデルによりノイズに強い動作プリミティブの自動抽出が可能である。次のポイントは、各動作プリミティブに力学系を仮定した動作生成器を構築し、多様な軌道が各動作生成器を組み合わせることにより生成可能とした点にある。また実機動作のための最低限のルールは組み込むものの、リアルワールドで重要となる力制御則を自動獲得した点も本研究の特色である。さらにポイントとして支援すべきかどうか判断し支援操作の発生予測を行うことにある。ここで提案する手法は支援手法としては荒い手法ではあるが、支援を見据えたロボットへの第一歩としては重要であると考えられる。

以下第2章では動作モデルとそれに基づく支援システムの概要を述べる。第3章では動作モデルの獲得方法、第4章では物体操作の認識方法、第5章では支援物体操作の予測方法、第6章では支援物体操作のロボットの軌道生成方法について述べる。そして第7章では支援システムの各機能の検証実験結果を示し、第8章で結論を述べる。

## 2. 動作プリミティブに基づく物体操作の認識・予測・軌道生成

本研究では短い時間単位の動作プリミティブを動作モデルとする。これは“lift-up”などの動作を担当し、人間の物体操作を認識する機能や、作業を実行する際の状況にあわせたロボットの軌道を生成する機能を持つ。また一連の物体操作は動作プリミティブの列により記述され、この列は支援すべき物体操作を定義する際やその発生を予測する際に用いられる。

また手・操作物体と、操作物体が接触する面（環境）を注目点とする。操作物体が接触する面としては本論文ではテーブル面のみを考えるが日常生活を考えると他に収納棚やキッチン台などが考えられる。

本支援システムでは、まず人間が行う机上作業における手と物体の位置・姿勢をステレオカメラで、物体と環境との干渉を力センサが取り付けられたテーブルによって計測する。物体と環境との干渉は手に力センサの取り付けられたグローブを装着し計測することも考えられるが、日常生活での使用には向かな

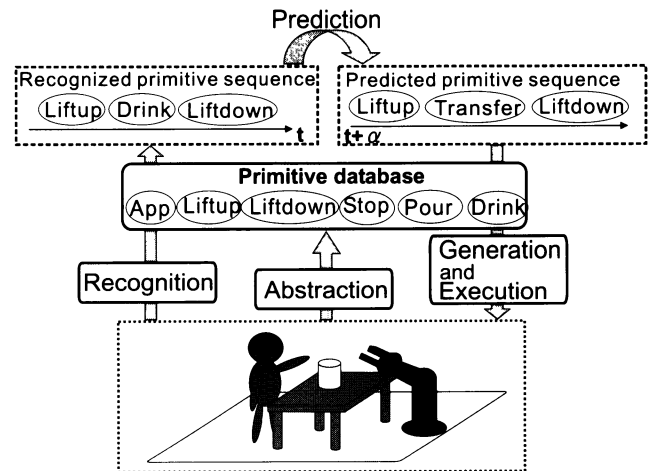


Fig. 1 System overview

い。そして計測されたデータをもとに支援に関する知識を獲得し支援を実行するシステムは、次に示す四つの機能により実現することとした。その概要は Fig. 1 のとおりである。

**動作プリミティブの獲得（机上作業のモデル化）：**計測データを短い時間単位に分節化し、各セグメントを統計的手法により分類する。分節化は人間が意味付け可能な単位として行うことが望まれるが、それに対応する汎用的なルールを決めるのは難しいため、一定時間幅で行うこととする。

**物体操作の認識：**獲得された動作プリミティブの列として人間の物体操作を記述する。

**支援物体操作の発生予測：**認識された動作プリミティブ列を入力として、次に行われる物体操作が支援すべきか否かを予測する。

**支援軌道の生成：**次に行われる物体操作が支援すべきだと予測された場合、そのときの物体の位置や環境の状況に対応した支援物体操作の手先軌道を生成する。このとき物体と環境との干渉は人間の物体操作時のものを再現するように実行されるが、ロボットの関節角の障害物回避や把持計画などは本論文では扱わない。これは人間とロボットの機構の違いによるもので、物体操作において模倣する対象ではないと考えているからである。

本システムの第一の特徴は動作プリミティブを統計的手法により自動的に獲得することであるが、それが人間の感覚と合致している保証はない。そもそも人間は類似する動作に異なる名前を付けたりするなど、その分類は曖昧で一意ではない。そこで、基本的には動作プリミティブはシステムなりの物体操作の解釈であり、人間の感覚と合致していなくてもよいものとする。ただし支援物体操作の発生を予測するためには、動作プリミティブがある程度人間が解釈できる分類となっている必要があると考えられる。また動作プリミティブの時間的な長さとなる分節化の幅は、人間が物体操作を分節化する際の最小の長さであるとする。例えば“pick-and-place”では“grip”や“place”の時間が意味付け可能な最小の長さである。このように短い時間単位の動作プリミティブとすることで、操作時間やその順序が多様な物体操作も比較的少数の動作プリミティブの列として表現

することができる。また動作の認識を行うに当たり、複雑で多次元の現象を抽象的な動作プリミティブに一度落とし込むことにより、その後の処理で高速かつロバストな高次の認識が実現されると期待される。さらに軌道生成において、複雑な軌道であっても限定された動作プリミティブの組み合わせにより表現可能であることが挙げられる。

本システムのもう一つの特徴として観察データとともにそれを定性的な表現に変換した定性値を利用する点があげられる。本研究で対象としている日常的な比較的ゆっくりとした物体を運ぶような支援を考えた場合、速度の大きさよりもその方向が重要になると考えられる。また力のかかり方も力の値よりも物体の接触の判定が重要となる。つまり、速度なら連続値から+、-、0のように定性値表現に変換することが可能となる。また、例えば“lift-up”の場合、クラスタリングにその速度など連続値を用いると意味的には同じ物体操作であってもその速度等によって異なる物体操作と分類されてしまう。一方、物体の運動の方向などといった定性的な表現を用いると同じ物体操作として分類される。また人間が物体操作の制御方法を表現する際にも、“障害物に近づいたら避ける”などのような定性的な表現を用いる。このように定性値はロボットによる物体操作の認識や“コツ”の発見に利用可能である。なお本研究で“コツ”とは、物体と環境（テーブル）との接触時における物体の制御法であるとする。さらに手や物体の速度、傾きなどを連続値として取り扱おうと非常にノイズに弱いシステムとなる。この状況で多数の動作セグメントをモデル化すると、意味のある動作プリミティブを得ることは難しい。しかしながら定性値表現にすることで、各動作プリミティブ内の動作セグメントはノイズに強い比較的安定した傾向を持つものと期待できる。

以下に本支援システムの使用例を示す。“コップの中身を飲みコップを片付ける”というパターンが多数回観察され“片付け”が支援対象と指定されていたとすると、日常生活中で“コップの中身を飲む”に対応する動作プリミティブの列が認識されると次に“コップを片付ける”ことが予測され、それに対応した動作プリミティブ列からそのときの物体の位置に対応した支援軌道が生成される。

### 3. 支援システムの機能1：動作プリミティブの獲得

動作プリミティブは、一連の計測データを時間軸方向に分節化し得られたセグメントを時系列クラスタリング手法により分類することで得られる。クラスタリング手法を適用するためには時系列データであるセグメント間の類似度を計算する必要がある。そこでまず類似尺度について考察し、それから具体的なクラスタリング手法を示す。

#### 3.1 類似尺度について

時系列データの類似尺度にはDPマッチングのようなデータそのものを利用して類似度を計算する方法があるが、それではノイズ等の影響を受けやすくなってしまふ。そこで時系列データの背後に存在する文脈情報を表現可能な隠れマルコフモデル[8]（以後HMM）に基づく類似尺度を利用することにする。

Inamuraらは二つの異なる動作を別々のHMMでモデル化し、その間の類似度をKullback-Leibler距離により計算してい

る[9]。しかし本研究のように多数のセグメント間の類似度を計算する場合、セグメント数分のHMMをモデル化する必要がある計算時間が問題となる。そこで本研究では多数のセグメントを一つのHMMでモデル化し、そのHMMに基づいて類似度計算を行う周辺化カウントカーネル[10]を使用する。周辺化カウントカーネルは観測データとHMMの文脈（隠れ状態）情報を両方考慮にいたれた類似度計算法である。例えば観測データ $\mathbf{x}$ と対応するHMMの隠れ状態 $\mathbf{h}$ が

$$\mathbf{x} = [+ , + , 0 , - , -]^t$$

$$\mathbf{h} = [s_1 , s_1 , s_2 , s_2 , s_2]^t$$

の場合、観測値と隠れ状態を結合した出現回数は、

$$c_{+,s_1} = 2, c_{0,s_1} = 0, c_{-,s_1} = 0,$$

$$c_{+,s_2} = 0, c_{0,s_2} = 1, c_{-,s_2} = 2.$$

となる。ここで $\mathbf{z} = [\mathbf{x}, \mathbf{h}]$ ,  $k \in \{+, 0, -\}$ ,  $l \in \{s_1, s_2\}$ とし、比較対象の時系列データとその隠れ状態 $\mathbf{x}'$ ,  $\mathbf{h}'$ が得られたとすると、出現回数の内積をとった結合カーネルは、

$$K_z(\mathbf{z}, \mathbf{z}') = \sum_k \sum_l c_{k,l} c'_{k,l} \quad (1)$$

となり、これを隠れ変数で周辺化した、

$$K(\mathbf{x}, \mathbf{x}') = \sum_{\mathbf{h}} \sum_{\mathbf{h}'} p(\mathbf{h}|\mathbf{x})p(\mathbf{h}'|\mathbf{x}')K_z(\mathbf{z}, \mathbf{z}') \quad (2)$$

が周辺化カウントカーネルとなる。

#### 3.2 動作セグメントのクラスタリング法

人間動作の中でも物体操作を中心に考えるため人間の手先と物体の軌道に注目することとし、ステレオカメラシステムにより手先の三次元位置と物体の三次元位置・鉛直軸からの傾きを計測する。なお計測データは鉛直方向とテーブルの2辺を軸とする直交座標系により記述するものとする。これは物体操作には重力方向が重要であり、また机上の物品整理は直交座標上で行われることが多いと考えるからである。また物体とテーブルとの接触を計測するために非拘束に計測可能なテーブルセンサによりテーブルにかかる負荷を計測する。

次に計測データを次に示す八次元の定性値時系列データに変換する。

- 手物体の各軸方向の速度を  $\{+, 0, -\}$  に離散化（三次元 × 2）
- 物体の鉛直軸からの傾きを傾いて  $\{\text{いる}, \text{いない}\}$  に離散化（一次元）
- テーブル負荷の変化を  $\{+, 0, -\}$  に離散化（一次元）

得られた定性値時系列データを利用し、以下の手順により物体操作の動作プリミティブを抽出する。

**分節化：**時系列データを一定時間幅  $T_{seg}$  ごとに分節化し、分節化された動作セグメントの集合を  $\mathbf{X} = \{X_i\}_{i=1}^N$  とする。 $N$  はセグメント数を表し、動作セグメント  $X_i$  はそれぞれ八次元で長さ  $T_{seg}$  の時系列データである。本研究では、スポーツのようなすばやい動作ではなく、日常的な比較的ゆっ

くりした動作の非常に短い時間のモデル化を想定している。このような状況では、同一動作プリミティブに属すると考えられる動作セグメントが多く得られるために、単に時間で区切った動作セグメントであってもランダムな動作セグメントがマイノリティとなり、動作プリミティブには同一の傾向を見出すことが可能となる。

**HMMによるモデル化：**HMMにより動作セグメントの集合  $\mathbf{X}$  をモデル化する。ただし  $\mathbf{X}$  は八次元の離散値のデータなので多次元離散 HMM [11] を使用する。また HMM は一方向性であるとし、状態数は BIC により統計的に決定する。ゆっくりした動作の非常に短い時間のモデル化において、あるステータス ( $s_i$ ) から別のステータス ( $s_j$ ) に遷移し、さらに元のステータス ( $s_i$ ) に遷移する状況が非常に少ないと考えられる。そのため全結合型ではなく一方向性の HMM を採用した。また全結合型よりも一方向性の方がパラメータ数を少なくできるので学習時間も短くできる長所がある。ただし、今回の手法では一方向性 HMM が妥当であると考えられるが、全結合 HMM を利用したとしても本論文の手法の流れに影響を与えるものではない。

**類似度の算出：**推定した HMM に基づき、周辺化カウントカーネルにより動作セグメント間の類似度  $K(X_i, X_j)$  を算出する。ただし各次元ごとに式 (1) で各次元の結合カーネルを計算しその和をとったものを結合カーネルとし周辺化する。

**動作プリミティブ空間への写像：**類似度情報を元にカーネル主成分分析 [12] により各動作セグメントを多次元空間 (動作プリミティブ空間と呼ぶ) 上の 1 点に写像し、それを  $\phi(X_i)$  とする。こうすることでクラス数評価など様々な定量的評価が可能となる。カーネル主成分分析においてはカーネルの設計が重要となる。本研究では 3.1 節に示した観測値の時刻のずれや、挿入、欠落に頑健なカウントカーネルを利用し、カウントカーネルでは表現しきれない時系列情報を組み込むために HMM を利用している。このようなカーネルの選択はどの手法よりも優れているとは言い切れないが、観測値の時刻のずれや、挿入、欠落に頑健かつ意味レベルのような時系列情報を考慮しているために、少なくともベターなカーネルであると考えられる。

**混合ガウス分布によるクラスタリング：**動作プリミティブ空間で表された動作セグメント  $\phi(X_i)$  を混合ガウス分布によりクラスタリングする。EM アルゴリズムによって推定された混合ガウス分布の各ガウス分布に対する  $\phi(X_i)$  の尤度を  $\{\mathcal{L}_k(\phi(X_i))\}_{k=1}^K$  とすると、各ガウス分布へのメンバシップ値は、

$$M_k(\phi(X_i)) = \frac{\mathcal{L}_k(\phi(X_i))}{\sum_{k=1}^K \mathcal{L}_k(\phi(X_i))} \quad (3)$$

となる。ただし  $K$  は混合数 (クラスタ数) である。このメンバシップ値が最大のクラスタを  $\phi(X_i)$  の属するクラスタとする。このときクラスタ数は BIC により統計的に決定する。

**動作プリミティブの定義：**得られた各クラスタを動作プリミティブとして定義し、動作プリミティブの代表動作セグメン

トをクラスタのうちメンバシップ値が最大の動作セグメントとし  $X_c^{(k)}$  とする。また同様にメンバシップ値が最小の動作セグメントを境界動作セグメントとし  $X_b^{(k)}$  とする。また動作プリミティブ間の類似度を代表動作セグメント間の類似度  $K(X_c^{(k)}, X_c^{(k')})$  とする。

#### 4. 支援システムの機能 2：物体操作の認識

Fig. 2 に認識方法の概要を示す。

前章と同様に一定時間幅で分節化され定性値に変換された入力データを  $\tilde{X}$  とする。認識は各動作プリミティブごとに代表動作セグメントと入力データの周辺化カウントカーネル  $K(X_c^{(k)}, \tilde{X})$  を計算する。そして代表動作セグメントと境界動作セグメントとの周辺化カウントカーネル  $K(X_c^{(k)}, X_b^{(k)})$  を閾値として、

$$K(X_c^{(k)}, \tilde{X}) = \begin{cases} K(X_c^{(k)}, \tilde{X}), & K(X_c^{(k)}, \tilde{X}) \geq K(X_c^{(k)}, X_b^{(k)}) \\ 0, & otherwise \end{cases} \quad (4)$$

とする。そしてすべての動作プリミティブとの周辺化カウントカーネルの値  $K(X_c^{(k)}, \tilde{X})$  が 0 である動作セグメント  $\tilde{X}$  を、“新奇的動作”として認識する。また新奇的動作として認識を行わなかった入力動作セグメントが所属する動作プリミティブ ( $\hat{k}$ ) を

$$\hat{k} = \arg \max_k K(X_c^{(k)}, \tilde{X}) \quad (5)$$

と決定する。

#### 5. 支援システムの機能 3：支援物体操作の発生予測

まず支援対象の教示方法について述べる。システムはあらかじめ動作プリミティブの獲得に用いたデータを利用し、物体が運動している間の動作プリミティブの列を物体操作として抽出しておく。そしてそれが連続している部分を物体操作として定義する。次に同じデータを人間が見て、支援してほしい物体操作のうち任意の時刻を選択する。システムはその時刻が含まれる

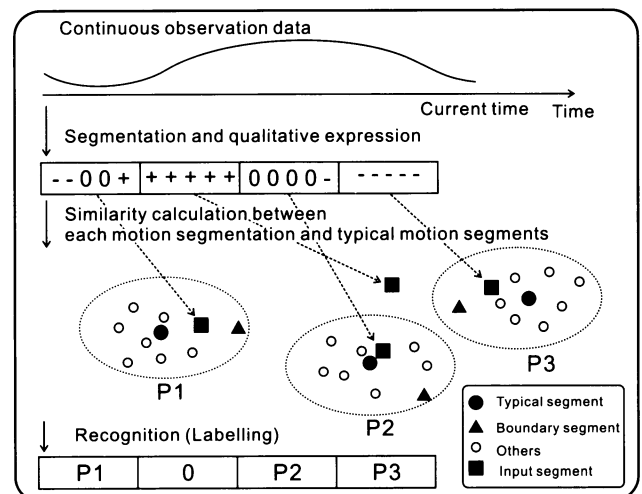


Fig. 2 Object-handling recognition utilizing acquired motion primitives

物体操作の動作プリミティブ列を支援物体操作として定義する。

教示された支援物体操作が次に発生するかどうかを、現在までの人間動作の動作プリミティブ列から予測するモデルとして2クラス識別器である Support Vector Machine (SVM) [14] を使用する。SVM で識別する内容は、現在から数秒先に支援物体操作を行うべきかどうかである。そこで支援物体操作の始めの時刻  $t_0$  からさかのぼって、 $T_{step}$  個より前の  $T_{span}$  個の動作プリミティブ列

$$\mathbf{P}_{in} = (P_{t_0-T_{step}-T_{span}}, P_{t_0-T_{step}-T_{span}+1}, \dots, P_{t_0-T_{step}})$$

を正例のときの入力とする。負例の入力は支援対象以外の物体操作を考え正例のときと同様に定め、SVM を学習する。ただし SVM のカーネルには、各時刻の動作プリミティブ間の類似度の和、

$$\sum_{t=0}^{T_{span}-1} K(P_{t_0-T_{step}-t}, P'_{t_0-T_{step}-t})$$

を使用する。ここで各時刻の動作セグメント間の類似度の積を用いると、本当に動作プリミティブ列が一致していないと類似度が0となり都合が悪くなる。各時刻の動作セグメント間の類似度の和を利用することでこの問題を回避し、時間伸縮性がある程度考慮することが可能になる。なお SVM のソフトマージンの罰則項や  $T_{step}$ 、 $T_{span}$  はあらかじめ定義するものとする。 $T_{step}$  は発生が予測されてから物体操作を実行するまでに必要なシステムの準備時間と考え、これは実行システムの特性にあわせて設定されるべきパラメータである。 $T_{span}$  は支援物体操作の発生予測の手掛かりとなる物体操作などの情報を含む必要があり、支援物体操作の特性に依存するので個々に最適なものを設定する必要があると考える。これについて将来的には SVM の汎化性能を指標にして最適化する方法 [15] などが考えられる。 $T_{step}$ 、 $T_{span}$  についてはその大小によって予測精度にどのような影響があるか、実験により検証する。

学習した予測器をもとに支援物体操作の発生を予測する際には、上記と同様に各時刻で SVM の入力  $\mathbf{P}_{in}$  を定めて SVM の出力を計算し、出力が正ならばその時点から  $T_{step}$  後に支援物体操作が発生するものと予測する。なお連続した時刻で出力が正の場合は始めの時刻のみを考えることにする。

## 6. 支援システムの機能4：支援軌道の生成

5章のように動作プリミティブ列として定義されている支援物体操作の手先軌道は、動作プリミティブごとに観察から学習された動作生成器を用いて生成される。一連動作の種類が増えた場合それぞれの軌道を学習するのは困難である。しかしながら限られた数の動作プリミティブごとに軌道を学習し動作生成器を設計するのは比較的容易であり、動作生成時にはそれらを組み合わせることで多様な一連動作を生成可能であると考えられる。また環境からの力フィードバックに基いた物体の制御則を獲得し、それを用いて環境との干渉を考慮に入れた物体操作を実現する。以下、個々の動作プリミティブごとに学習される動作生成器と物体の力制御則について詳細を述べ、支援物体操

作の動作プリミティブ列から実行軌道を生成する方法について述べる。

### 6.1 動作生成器：Dynamic Neural Network

予測された支援対象の物体操作の実行軌道を生成するために、初期位置のずれに対応した軌道が生成可能な Dynamic Neural Network (DNN) [16] を利用する。DNN は時間進展する力学系を担当するオシレータと形状変換を担当する多層ニューラルネットワーク (FFNN) からなる。類似の手法である recurrent neural network に比べ、学習時間や安定性の点で利点がある。また、同じ動作プリミティブでも場所により軌道が異なる場合には、場所に応じた適切な軌道を生成する必要があるが、オシレータと FFNN の組み合わせは後述するようにこの能力を備える。さらに本研究では各動作プリミティブに動作生成器を割り当ての方針を採っているために、各プリミティブからの生成軌道を滑らかに接続する必要があるが、この手法は初期状態が異なる場合でも望む軌道に収束しようとする性質があるため、各軌道を滑らかに接続することが可能となる。

#### 6.1.1 オシレータ

オシレータは次式のようにあらかじめ設計する。

$$\dot{\mathbf{o}}(t) = \frac{1}{\tau} \begin{pmatrix} -\mu & -1 \\ -\mu & 1 \end{pmatrix} \mathbf{o}(t) + \frac{\mu r_0}{\tau} \mathbf{a}(t)$$

ただし  $\mathbf{o}(t)$  はオシレータの軌道であり、

$$\mathbf{o}(t) = \begin{pmatrix} o_1(t) \\ o_2(t) \end{pmatrix}$$

$$\mathbf{a}(t) = \begin{pmatrix} \frac{o_1(t)}{\sqrt{o_1^2(t) + o_2^2(t)}} \\ \frac{o_2(t)}{\sqrt{o_1^2(t) + o_2^2(t)}} \end{pmatrix}$$

である。このオシレータは半径  $r_0$  の円をリミットサイクルに持つ力学系となり、周期は  $\tau$  によって、リミットサイクルへの収束速度は  $\mu$  によって調整することが可能である。また本研究のように学習する軌道が開軌道である場合は、

$$o_1(t) = r(t) \cos \phi(t), \quad o_2(t) = r(t) \sin \phi(t) \quad (6)$$

のように極座標表現に変換したオシレータ、

$$\tau \dot{\phi}(t) = 1 \quad (7)$$

$$\tau \dot{r}(t) = -\mu(r(t) - r_0) \quad (8)$$

を使用する。以下オシレータの出力を  $\mathbf{o}(t) = [r(t), \phi(t)]^t$  とする。

#### 6.1.2 形状変換3層ニューラルネットワーク

オシレータの初期値を  $\mathbf{o}(0) = [1.0, 0.0]^t$  とし、式 (7) と式 (8) により生成された軌道  $\mathbf{o}(t)$  を FFNN である  $\mathbf{h}(\cdot)$  に入力し所望の教師軌道  $\mathbf{y}(t)$  に

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{o}(t)) \quad (9)$$

のようにして変換する。本研究での教師軌道は各動作プリミティブにおける手先軌道である。

軌道学習時には、オシレータのパラメータのうちリミットサイクルの半径  $r_0$ 、収束速度  $\mu$  を与え周期に対応する  $\tau$  は教師軌道の長さから定めオシレータ出力  $\mathbf{o}(t)$  を計算し、それと教師軌道  $\mathbf{y}(t)$  のマッピングを学習すればよい。

軌道生成時には、所望の周期と再現したい教師軌道の物体の初期位値を入力とし、式 (9) により軌道を生成する。このとき実行したい軌道の初期値を  $\mathbf{d}$  とした場合、式 (9) の逆問題、

$$\mathbf{o}(0) = \mathbf{h}^{-1}(\mathbf{d}) \quad (10)$$

を解くことでオシレータの初期値を求め、軌道生成を行う。しかし教師軌道のはる空間に  $\mathbf{d}$  が含まれない場合式 (10) の解は存在しない。例えば教師軌道の次元がオシレータの次元 (2) よりも大きい場合にこの問題が起こる。

そこで初期値  $\mathbf{d}$  からの誤差二乗を最小化する  $\mathbf{o}(0)$  をオシレータの初期値とすることにする。

$$\mathbf{o}(0) = \underset{\mathbf{o}}{\operatorname{argmin}} \|\mathbf{h}(\mathbf{o}) - \mathbf{d}\|^2 \quad (11)$$

ただし  $\mathbf{h}(\cdot)$  は FFNN なので解析的に解くことができない。そこで最急降下法などで数値的に求めることにする。今、

$$\dot{\mathbf{z}}(t) = -\tau_2 \mathbf{z}(t) \quad (12)$$

で定義される力学系を用意し、 $\mathbf{z}(0) = \mathbf{h}(\mathbf{o}(0)) - \mathbf{d}$  とする。そして DNN の出力を、

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{o}(t)) - \mathbf{z}(t) \quad (13)$$

と書き直す。こうすることで  $\lim_{t \rightarrow \infty} \mathbf{z}(t) = 0$  より、

$$\mathbf{y}(0) = \mathbf{d}, \quad \lim_{t \rightarrow \infty} \mathbf{y}(t) = \mathbf{h}(\mathbf{o}_\infty(t)) = \mathbf{y}_\infty(t) \quad (14)$$

が成り立つ。

また FFNN の入力にオシレータ出力以外のものをもとめることもできる。本研究では動作生成器の学習時に、教師軌道観察時の物体の初期位置を FFNN の入力に追加し結合荷重を学習する。こうすることにより軌道生成時には再現したい教師軌道の物体の初期位置を与えることで物体の場所に応じた異なる軌道を学習可能である。

## 6.2 HMM に基づく力制御則の獲得と利用

DNN により生成した軌道を、観察から得た定性的な力制御則と環境からのカフィードバックを利用してテーブルと物体との干渉を考慮に入れた軌道に修正する方法を述べる。定性的な制御則とは、例えば物体を “place” する際における、“物体を下方に動かし”、“テーブル負荷が増加した”ら “物体を静止させる” といったものである。このような定性的な制御則は “place” など環境側の状態が大きく変化する部分に有効であると考えられる。このような制御則を利用し、もし観察時と異なるような定性値の遷移が検出された場合には、その矛盾を解消するような軌道に修正する。以下、力制御則とそれに基づく軌道の制御方法について述べる。

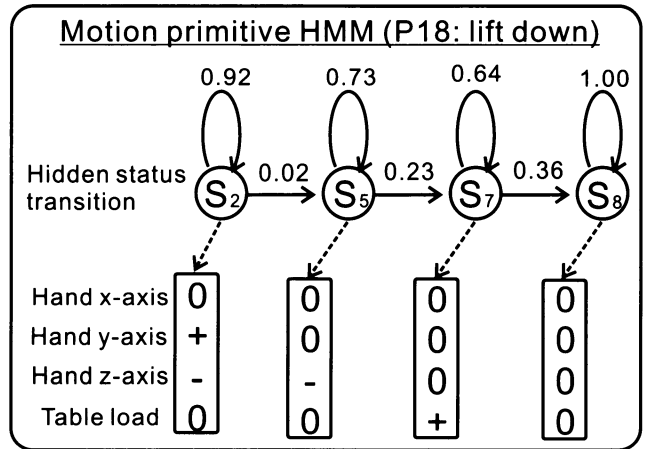


Fig. 3 HMM for force control

### 6.2.1 HMM による力制御則の表現

観測された定性値の時系列データから制御則 (=定性値の遷移パターン) を獲得することが考えられるが、ノイズ等の影響でその遷移図は大きくなってしまふ。そこで定性値の時系列データをモデル化した動作プリミティブ HMM の隠れ状態遷移を利用することが考えられる。Fig. 3 は 3 章の方法で得られた動作プリミティブ HMM の一例である (実際隠れ状態は  $S_1, \dots, S_8$  の 8 つあるがそのうち代表的な遷移のみを、出力定性値は出力確率が最大のもののみを示している)。初期状態  $S_2$  と次の状態  $S_5$  においては手の鉛直方向が “-” でテーブル負荷は “0” となっている。さらに次の状態  $S_7$  においては手の鉛直方向が “0” でテーブル負荷が “+” となっている。したがって  $S_2 \rightarrow S_5 \rightarrow S_7$  の遷移で “place” におけるテーブル負荷に基づく手の定性的な制御則を表しているといえる。

### 6.2.2 力制御則による軌道修正

手先実行軌道の 3 軸方向の定性値と同時に計測されたテーブル負荷の定性値に対する動作プリミティブ HMM の最適隠れ状態遷移を viterbi アルゴリズムにより推定する。その隠れ状態における最大出力確率のテーブル負荷定性値と実測のテーブル負荷定性値が矛盾している場合、次のようなルールにより軌道修正をする。

ルール 1 実測のテーブル負荷変化がない場合：前の隠れ状態時の時刻までオシレータの時刻を戻す

ルール 2 実測のテーブル負荷変化がある場合：テーブル負荷変化が一致する隠れ状態まで遷移し、その隠れ状態における手の速度定性値が一致するまでオシレータの時刻をすすめる

このようにして DNN の振る舞いを制御することで環境との干渉を考慮に入れた軌道が生成できる。例えば Fig. 3 において、手先の実行軌道の定性値が “[0, 0, -]” でテーブル負荷定性値が “+” でその時の最適隠れ状態が  $S_5$  の場合テーブル負荷の最大出力確率は “0” となっており計測値と矛盾するので、テーブル負荷の最大出力確率が一致する隠れ状態  $S_7$  に遷移し、そのときの手の定性値 “[0, 0, 0]” になるようにオシレータの時刻を進める。こうすることで観察時にはあまり見られなかったような

手（物体）の運動と対応するテーブル負荷が実行時に検出された場合には、文脈的にそれに近く観察時に典型的に見られたものに修正することができ、人間の定性的な物体の力制御方法を再現することが可能になる。

### 6.3 動作プリミティブ列からの物体操作の軌道生成

以上で個々の動作プリミティブにおける軌道生成・修正法を述べた。ここでは動作プリミティブの列として定義されている一連の支援物体操作の軌道生成法をまとめる。

物体操作の動作プリミティブ列を  $\mathbf{P} = [P_1, \dots, P_n]$  とし、計測から得られる物体の初期位置を  $\mathbf{y}_0^{(obj)}$  とする。また手の初期位置は  $\mathbf{y}_0^{(obj)}$  に一致するものとする。そして時刻を  $t = 0$ 、動作プリミティブ列の位置を  $i = 1$  として次の手順により軌道を生成していく。あらかじめ  $\mathbf{y}_0^{(obj)}$  の位置にハンドを移動させ物体を把持しているものとする。ただ実際把持時において、人間の手と物体の位置関係と、マニピュレータと物体の位置関係は一致しないのでその差は軌道の平行移動により補完するものとする。

- (1) 動作プリミティブを  $P_i$  にセットする。
- (2) 手物体の初期位置より式 (11) に基づき DNN のオシレータの初期値を決定する。
- (3)  $t \leftarrow t + dt$  として式 (13) により手先軌道  $\mathbf{y}_t$  を生成する。
- (4) テーブル負荷を取得し手・物体・テーブル負荷の定性値を計算する（把持時は物体と手の定性値は一致するものとする）。
- (5) もし  $i = n$ （最終動作プリミティブ）で手・物体の定性値がすべて“0”ならばハンドを“open”にする。
- (6) 力フィードバックに基づく軌道の制御を行う。もし修正条件 1 の場合で HMM の隠れ状態が最終状態になっても矛盾が解消されない場合は、 $P_{i+1}$  の HMM を利用してさらに探索し、 $i \leftarrow i + 1$  として 1. に戻る。
- (7) もしオシレータのうち  $\phi(t)$  が  $2\pi$  を超えたら  $i \leftarrow i + 1$  として 1. にもどる。そうでなければ 2. に戻る。
- (8) 終了条件は  $i > n$  である。

## 7. 評価実験

### 7.1 実験条件

“コップを引き寄せ中身を飲みもとの位置に片付ける”といった一連の物体操作 36 試行分を対象に、机上作業支援システムの各機能の検証実験を行った。コップや手にはマーカーを取り付け、ステレオビジョンにより三次元位置と姿勢（コップのみ）を取得した。また物体とテーブルとの接触を計測するためにロードセル 4 点支持式テーブルセンサを利用した。データのサンプリングレートは 30 [Hz] である。これらの 36 試行分の八次元のデータを 3.2 節で述べた定性値化を行い HMM に入力して学習する。離散化の閾値はセンサ値のばらつきを参考に速度の閾値は 1 [mm/frame]、傾きの閾値は 20 [deg]、テーブル負荷は 0.25 [N] とした（実験に用いた物体の重量は約 4 [N]）。動作プリミティブの獲得で利用する分節化の分割幅 ( $T_{seg}$ ) は 0.5 [s] とし、一つの動作プリミティブは 15 点の時系列データとなる。動作プリミティブ空間へ写像する際の、動作プリミティブ空間の次元は、主成分の累積寄与率が 0.99 となる次元とした。

支援物体操作の発生予測において、“物体が運動している間

の動作プリミティブ”の定義が重要となるが、本実験では物体の各軸方向の速度定性値のうち少なくとも一つが“0”ではない部分が動作プリミティブの長さ ( $T_{seg}$ ) の 10% を占めるものとした。

支援軌道の生成において、DNN の収束速度  $\mu$  を 0.5、半径  $r_0$  を 1.0 とした。また FFNN の中間層の数を 50 とした。

### 7.2 動作プリミティブの獲得実験

まず 3.2 節の方法で動作プリミティブのクラスタリングを行った。一連の物体操作から 1,347 の動作セグメントを得た。このクラスタリングの過程で作成された HMM の状態数は 56 となった。また得られた HMM の遷移確率、出現確率の一部について提示する。便宜的に隠れ状態に 1 から 56 の番号を振り、隠れ状態の遷移確率を Fig. 4 に示す。ノードの番号が隠れ状態の番号、アークの数字が遷移確率を示す。例えば、隠れ状態 1 番の場合、単位時刻後に元の状態にいる確率が最も高く 0.847、次に 4 番目の隠れ状態に遷移する確率が 0.153、その他の状態に遷移する確率はほとんどないことを示している。実験では、動作の速度は比較的ゆっくりであり、サンプリングレートが 30 [Hz] であることから、どの隠れ状態でも元の状態に遷移する確率が高くなっていると考えられる。

次に隠れ状態 1 から 3 番目の出現確率を示す。Table 1 が隠れ状態 1 からの出現確率、Table 2 が隠れ状態 2 からの出現確率、Table 3 が隠れ状態 3 からの出現確率を示している。Table 1 の結果より、手と物体の動きがほとんどなく、物体も傾いておらず、テーブル負荷は増加傾向にあることから、隠れ状態 1 は物体をテーブルに置くような動作を表現しているものと考えられる。Table 2 の結果から、手は z 軸方向に増加傾向、物体は y 軸方向に減少傾向、物体は傾き、テーブル負荷はないことから、人がコップ（物体）を飲む（飲むとする）動作を表現しているものと考えられる。Table 3 の結果より、手の速度は x 軸、z 軸とも減少傾向、y 軸は増加傾向、物体は静止していると読めるので、物体を取りに行くかもしくは離れた後の動作と考えられる。

上記のような傾向は、他の隠れ状態にも見られ、人の直感とは外れない程度の HMM の学習が行われていると考えられる。これからも一定の単位時間区で切った動作セグメントを定性値表現に変換し一つの 1 方向 HMM のみで学習したとしても、人と直感に合う結果を得ることから有効性が確認できたと考えられる。

また、クラスタリングした結果をあらかじめ各動作セグメントごとに人間がつけたラベルと結果を比較したものを Table 4 に示す。表中の占有率は、クラスタ内に含まれるラベルの数をクラスタ内のセグメントの総数で割ったものと定義する。多くのクラスタで一つのラベル占有率が高くなっており、ある程度人間のラベリングと類似する結果が得られた。また複数のラベルが存在するクラスタは、例えばクラスタ 5 などは人間がラベル付けする際に“approach”と“grip”の境界が曖昧であったことが原因と考えられる。またクラスタ 14 などは、“pick-and-place”全体が一つのクラスタに分類されたため“lift-up”と“lift-down”が回数ずつ含まれている。しかし本研究では各動作プリミティブが人間の直感にあっていなくても、支

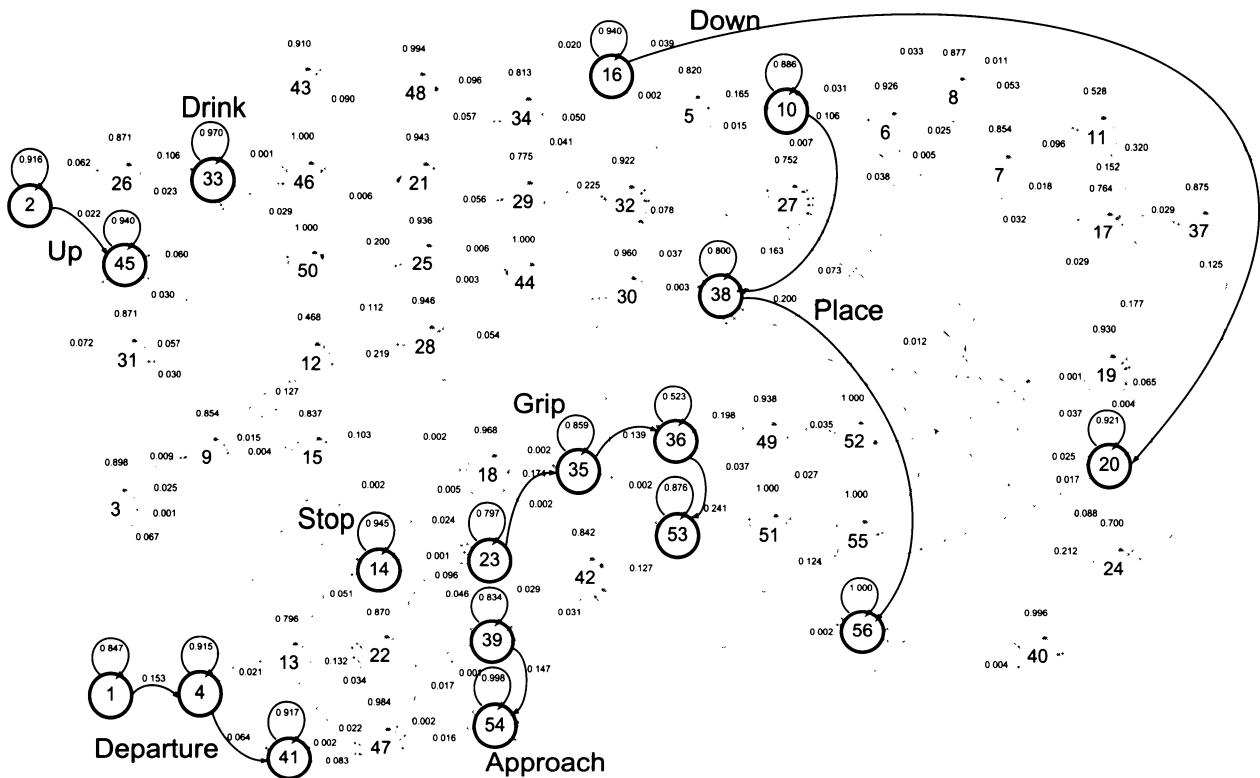


Fig. 4 Transition probability of HMM and typical transition path of motion primitives

Table 1 Observation probabilities of hidden status 1 for motion primitives

Velocity and load change	-	0	+
Object inclination	Not inclined	Inclined	—
Hand velocity x-axis	0.000	0.868	0.132
Hand velocity y-axis	0.040	0.941	0.019
Hand velocity z-axis	0.006	0.919	0.075
Object velocity x-axis	0.000	1.000	0.000
Object velocity y-axis	0.000	1.000	0.000
Object velocity z-axis	0.047	0.953	0.000
Object inclination	1.000	0.000	—
Table load change	0.018	0.357	0.625

Table 3 Observation probabilities of hidden status 3 for motion primitives

Velocity and load change	-	0	+
Object inclination	Not inclined	Inclined	—
Hand velocity x-axis	0.991	0.010	0.000
Hand velocity y-axis	0.001	0.083	0.916
Hand velocity z-axis	0.730	0.105	0.165
Object velocity x-axis	0.000	1.000	0.000
Object velocity y-axis	0.000	1.000	0.000
Object velocity z-axis	0.000	1.000	0.000
Object inclination	1.000	0.000	—
Table load change	0.000	1.000	0.000

Table 2 Observation probabilities of hidden status 2 for motion primitives

Velocity and load change	-	0	+
Object inclination	Not inclined	Inclined	—
Hand velocity x-axis	0.000	1.000	0.000
Hand velocity y-axis	0.067	0.904	0.029
Hand velocity z-axis	0.000	0.069	0.931
Object velocity x-axis	0.000	1.000	0.000
Object velocity y-axis	0.956	0.044	0.000
Object velocity z-axis	0.000	0.000	1.000
Object inclination	0.039	0.961	—
Table load change	0.000	1.000	0.000

分かる。また Table 1~3 の出現確率との関係から、隠れ状態 1 は物体をテーブルに置くような動作であると説明したが、隠れ状態 1 は、物体から手が離れる動作の始めの状態となっており、1 → 4 → 41 は、物体を置き終わり手が離れる動作を表現していると考えられる。隠れ状態 2 は人がコップで飲む（飲むと）動作を表現していると述べたが、この状態は物体を持ち上げる動作の前半を示しており、コップを口に持っていくまでの過程でコップが徐々に傾いていることを示すものであったと解釈できる。隠れ状態 3 は、物体を取りに行くもしくは離れた後の動作と述べたが、状態 3 は状態 14 と 23 の前の状態であり、つかむ、止まるといった動作の直前の状態と見ることができると、人間の感覚と外れない程度の結果を得ていると言える。

7.3 物体操作認識・支援物体操作予測実験

次に別に計測した“コップを引き寄せ中身を飲んで元の位置に片付ける”といった一連の物体操作を 4 章の方法で認識し、同時に支援物体操作の発生を予測する実験を行った。なお、あら

援物体操作の予測や軌道生成が可能であればよいものとする。

Fig. 4 にラベル付けした動作プリミティブの隠れ状態における代表的な遷移パスを太い線で示す。これより“approach”であれば隠れ状態番号が 39 → 54, “grip”であれば 23 → 33 → 36 → 53, “place”であれば 10 → 38 → 56 と遷移することが



Table 4 Clustering of motion segments

Prim. index	Label (occupation ratio: %)							
	app.	grip	up	drink	down	place	dep.	stop
1	<b>54.7</b>						1.6	43.8
2	<b>100</b>							
3	<b>98.3</b>							1.6
4	<b>97.1</b>							2.9
5	19.0	<b>62.0</b>	8.9					10.1
6		16.9	<b>61.5</b>		21.5			
7		11.1	<b>88.9</b>					
8			<b>100</b>					
9			<b>79.3</b>	20.7				
10			<b>65.7</b>		32.8			1.5
11			<b>55.2</b>		44.8			
12			3.5	<b>96.6</b>				
13				<b>100</b>				
14		2.0	44.9		<b>53.1</b>			
15				27.3	<b>72.7</b>			
16			6.4		<b>93.6</b>			
17					<b>80.7</b>	19.4		
18					<b>66.7</b>	33.3		
19					42.5	<b>57.5</b>		
20						15.7	<b>84.3</b>	
21							<b>100</b>	
22	2.3						<b>97.7</b>	
23							<b>98.2</b>	1.8
24							40.0	<b>60.0</b>
25								<b>100</b>
26								<b>100</b>

はじめ3試行分の“片付け”に対応する物体操作を支援物体操作として教示し、予測の際の時間パラメータ  $T_{step}$ ,  $T_{span}$  はそれぞれ 2.5, 5 [s], SVM の罰則項は 1,000 と経験的に定めた。Table 5 は左から、時刻、人間がつけたラベル名、認識された動作プリミティブ (Table 4 中の最大占有率のラベル名)、支援対象発生の予測結果 ( $T_{step}$  後に発生するか否か)、を示す。

認識結果を見ると、例えば“drink”とラベル付けされている 8.0~9.0 [s] で、“drink”が 100%占有する動作プリミティブ 13 に認識されている。“片付け”の一部である 13.5~14.5 [s] では動作プリミティブ 14 に認識されているが、これはさきほども述べたとおり lift-up と lift-down がほぼ同数含まれる、“片付け”全体に対応する動作プリミティブであると考えられるので妥当であると考えられる。予測結果を見ると、支援物体操作の発生が唯一 12.0 [s] で予測されているため実際発生するのはそれから  $T_{step} = 2.5$  [s] 後の 14.5 [s] と予測されていることになる。実際 13.5~15.0 [s] で“片付け”が行われていたことから、その発生が予測できていることが分かる。

また動作プリミティブ抽出時とは異なる 21 試行分の同様の一連物体操作を検証データとし、予測器のパラメータ  $T_{step}$ ,  $T_{span}$  や“片付け”の教示回数を変えた場合の予測性能の比較実験を行った。予測は、上記のように発生が予測された時刻が実際“片付け”の最中の場合は成功とし、そうでない場合は失敗とした。そして次に定義する予測成功率と予測失敗率を評価指標とした。

$$\text{予測成功率} = \frac{\text{予測の成功回数}}{\text{“片付け”の発生回数}}$$

$$\text{予測失敗率} = \frac{\text{予測の失敗回数}}{\text{予測の成功回数} + \text{予測の失敗回数}}$$

Table 5 Recognition and prediction result. “p” is prediction result (1 or 0) of the supporting object-handling occurrence

Time	Label (ref.)	Prim.index (max.label)	p	Time	Label (ref.)	Prim.index (max.label)	p
0.0	app.	4 (app.)	0	8.0	drink	12 (drink)	0
0.5	app.	4 (app.)	0	8.5	drink	13 (drink)	0
1.0	grip	7 (up)	0	9.0	drink	13 (drink)	0
1.5	up	7 (up)	0	9.5	down	16 (down)	0
2.0	up	10 (up)	0	10.0	down	19 (place)	0
2.5	down	10 (up)	0	10.5	down	19 (place)	0
3.0	down	17 (down)	0	11.0	place	19 (place)	0
3.5	place	19 (place)	0	11.5	dep.	21 (dep.)	0
4.0	dep.	21 (dep.)	0	12.0	stop	26 (stop)	1
4.5	dep.	24 (stop)	0	12.5	app.	3 (app.)	0
5.0	app.	3 (app.)	0	13.0	app.	2 (app.)	0
5.5	app.	1 (app.)	0	13.5	up	14 (down)	0
6.0	up	5 (grip)	0	14.0	down	14 (down)	0
6.5	up	8 (up)	0	14.5	down	14 (down)	0
7.0	up	9 (up)	0	15.0	place	19 (place)	0
7.5	up	13 (drink)	0	15.5	dep.	21 (dep.)	0

結果は以下のとおりである。

- $T_{step} = 2.5$  [s],  $T_{span} = 5.0$  [s] で、“片付け”教示回数が 3 回, 9 回の場合  
予測成功率=0.48, 0.62  
予測失敗率=0.00, 0.00
- “片付け”教示回数が 9 回,  $T_{step} = 2.5$  [s] で,  $T_{span} = 2.5, 5.0, 7.5$  の場合  
予測成功率=0.67, 0.62, 0.00  
予測失敗率=0.73, 0.00, 0.00
- “片付け”教示回数が 9 回,  $T_{span} = 5.0$  [s] で,  $T_{step} = 1.5, 2.5, 5.0$  の場合  
予測成功率=0.52, 0.62, 0.29  
予測失敗率=0.00, 0.00, 0.00

まず教示回数が増えた方が多様なバリエーションの“片付け”を正解例として予測器の学習が行えるので、当然予測率が上昇することが分かる。しかし教示回数が 3 回でも約半分の“片付け”の発生を予測できていることから、少ない教示回数でもある程度の支援物体操作の予測が可能なが分かった。また入力データの長さ  $T_{span}$  は短すぎると情報が少なすぎて予測が失敗し、逆に長すぎると、SVM のカーネルの特性上学習データの正例にすべての時刻で類似した場合しか出力が正にならないため、“片付け”の発生が何も予測されなくなっている。これについては  $T_{span}$  の自動最適化や時間伸縮性のあるカーネルを使用することが考えられる。また  $T_{step}$  に関しては、あまりに長いとその分遠い先の事象を予測することになり明確な手掛かりがない場合は一般に予測が難しくなるので、予測成功率が小さくなる。これについてもある程度短い時間に設定する必要がある。支援システムとしては予測が 10 割成功し失敗は 0 であることが望まれるが、予測の成否が人間によって評価されればそれを用いて予測器を再学習しこの目標に近づけることが可能な枠組みであると考えられる。

#### 7.4 支援物体操作の軌道生成・実行実験

まずフィードバックなしで DNN のみを用いて“片付け”の

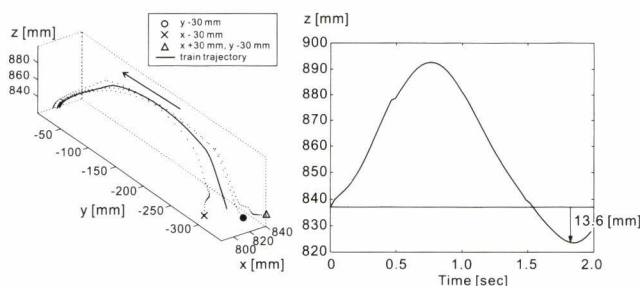


Fig. 5 Generated trajectory utilizing only DNN

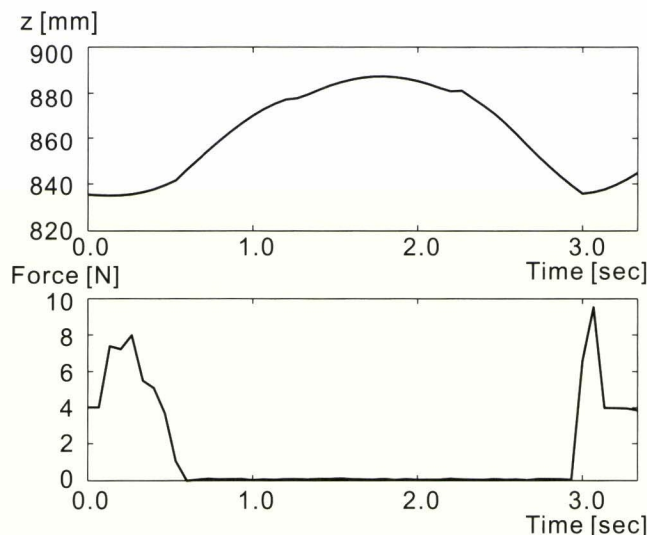


Fig. 6 Executed trajectory and measured force

動作プリミティブ列に対応した軌道を生成する実験を行った。このとき、学習データの物体の初期位置から  $[x, y]$  方向（平面方向）にそれぞれ  $[-30, 0]$ ,  $[0, -30]$ ,  $[30, -30]$  [mm] ずらした初期位置とした。Fig. 5 のうち左側は三次元プロットで実線が学習データの軌道で、点線はそれぞれの初期位置からの軌道生成結果である。右側は初期位置を  $x$  方向にのみ  $-30$  [mm] ずらした場合の  $z$  軸方向の軌道の時間変化を示す。  $y$  の正の方向が人から遠ざかる方向となっているので“片付け”に対応する軌道が生成できていることが分かる。ただし  $z$  軸の時間変化を見ると初期時の高さより  $13.6$  [mm] ほど下方で“place”するような軌道となってしまう、環境との干渉が問題となる軌道となっている。これは、ステレオカメラの計測誤差と DNN の学習誤差に起因するものだと考えられる。次に DNN による動作生成器に HMM による力制御を適用し、実機により軌道を生成した結果のうち  $z$  軸方向の軌道値の時間変化と同時に計測されたテーブル負荷を Fig. 6 に示す。この場合の物体の初期位置はステレオカメラで計測された三次元位置を使用している。HMM による力制御の結果、ほぼ物体の初期時の高さで“place”が行われていることが分かる。またそのとき発生するテーブル負荷も物体の重量  $4$  [N] の約  $2$  倍の負荷におさまっており、人間が“place”する際も大きいときはこのくらいの負荷がかかることから妥当な結果であるといえる。

最後に人間の物体操作を認識し“片付け”の発生を予測し、予

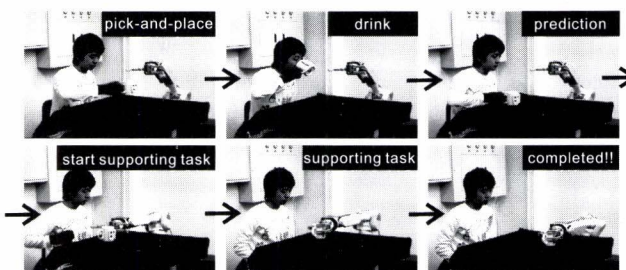


Fig. 7 Recognition, prediction and execution

測された物体操作を実行する一連の流れの風景を Fig. 7 に示す。人間がコップの中身を飲んだ直後に“片付け”の発生を予測し、その支援軌道の生成・実行を正しく行えていることが分かる。

## 8. 結 論

本研究では、あらかじめルールを与えずに観察から得た知識をもとに支援を実行する汎用的なロボットシステムの構築を目指し、その基本的な要素となる複数回の観察からの動作プリミティブに基づく物体操作の認識・予測・軌道生成手法の提案と実現を行った。さらに提案手法をロボットシステムに実装し、リアルワールドでも実演可能な手法であることを実証した。

具体的には、Hidden Markov Model (HMM) に基づく時系列クラスターリング手法により動作プリミティブを獲得し、動作プリミティブの列として人間動作を認識する手法、Support Vector Machine (SVM) により支援物体操作の発生タイミングを予測する手法、Dynamic Neural Network (DNN) により物体位置に対応した支援軌道を生成する手法、それを環境との干渉を考えた軌道に修正する制御則を HMM の隠れ状態遷移から獲得する手法を示した。“コップを引き寄せて中身を飲み片付ける”というシナリオを用いた検証実験により、獲得した動作プリミティブから人間動作を認識し適切なタイミングで支援物体操作の発生が予測可能であることを示した。また学習時と異なる状況における支援物体操作の軌道生成が可能であることを示した。

本研究の特色は以下の4点にまとめられる。(1) 定性値表現による動作プリミティブの自動獲得、(2) 各動作プリミティブに対応した動作生成器を組み合わせることにより多様な軌道の生成、(3) リアルワールドで重要となる力制御則の自動獲得（ただし、実機動作のための最低限のルールは組み込む）、(4) 支援操作の発生予測。またこの手法を提案するに当たり以下の三つの仮定を設けた。(1) 日常的な比較的ゆっくりとした支援を念頭に置く。(2) 任意の物体の把持・操作可能なロボットシステムが実現されている。(3) 支援される人のメンタルモデルを考慮しない。

後付したラベルと得られた動作プリミティブの関係を見ると、おおむね一致する傾向を得ることができた。つまり今回利用した定性値を利用すれば、手が物体に近づく、物体から離れる、持ち上げる、下げる、物体をつかむことが分かるために、これらを組み合わせて実現可能なレベルの支援は行える可能性がある。どのような物体を利用した机上作業を行ったとしても、物体に

手を伸ばす、つかむ、持ち上げるような基本的なタスクは必ず含まれるため、この点について、本提案手法は汎用性があると考えられる。しかしながらこの基本動作の組み合わせ以外の情報が無いとできないような作業の認識、支援は難しく、定性値選択を含めた本論文の手法の限界と言える。

将来課題としては、まず新奇な物体操作が観察された際に対応する動作プリミティブを追加し知識を増やすことが考えられる。新奇な物体操作として認識する枠組みにはなっているので、蓄積された新奇データに対し提案した動作プリミティブのクラスタリング手法を適用することで実現可能であると考えている。また、獲得物体が複数存在する場合などにより、複雑な環境下における入力データの選択方法や定性表現方法などを規格化することが必要になる。さらに、環境側に力センサを仮定しない場合、ロボットが物体操作の試行から自分のハンドなどに取り付けられた力センサを用いて力制御則を発見することに、本論文で提案した力制御則の獲得手法を応用することが考えられる。

**謝辞** この研究の一部は、学術振興会科学研究費、学術創成研究（課題名：知能ロボットによる模倣の構成論的研究、課題番号：13GS0006）により行われた。

## 参考文献

- [1] 平井成興, 佐藤知正: “テレロボットにおけるワールドモデル管理のための操縦行動の理解機能”, 日本ロボット学会誌, vol.7, no.6, pp.714-724, 1989.
- [2] K. Ikeuchi and T. Suehiro: “Toward an assembly plan from observation, part 1: Task recognition with polyhedral objects,” IEEE Trans Robotics and Automation, vol.10, no.3, pp.368-385, 1994.
- [3] 國吉康夫, 井上博允, 稲葉雅幸: “人間が実演して見せる作業の実時間視覚認識とそのロボット教示への応用”, 日本ロボット学会誌, vol.9, no.3, pp.295-303, 1991.
- [4] S. Schaal and C.G. Atkeson: “Robot Juggling: Implementation of Memory-Based Learning,” IEEE Control Systems, vol.14, no.1, pp.57-71, 1994.
- [5] D. Bentevegna, G. Cheng and C.G. Atkeson: “Learning from observation and from practice using behavioral primitives,” 11th International Symposium on Robotics Research, 2003.
- [6] 久保寺秀幸, 原田達也, 森武俊, 佐藤知正: “動作プリミティブに基づく日常物体操作のロボットによる支援”, 第5回計測自動制御学会システムインテグレーション部門講演会予稿集, pp.567-568, 2004.
- [7] K. Ogawara, J. Takamatsu, H. Kimura and K. Ikeuchi: “Modeling manipulation interactions by hidden markov models,” Proceedings of the 2002 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems, pp.1096-1101, 2002.
- [8] X. Huang, Y. Ariki and M. Jack: Hidden Markov Models for Speech Recognition (Edinburgh Information Technology Series, 7). Edinburgh University Press, 1990.
- [9] T. Inamura, H. Tanie and Y. Nakamura: “Keyframe compression and decompression for time series data based on the continuous hidden markov model,” Proceedings of the IEEE/RSJ Intl. Conference on Intelligent Robots and Systems, pp.1487-1492, 2003.
- [10] K. Tsuda, T. Kin and K. Asai: “Marginalized kernels for biological sequences,” Bioinformatics, vol.18, no.Suppl.1, pp.S268-S275, 2002.
- [11] J. Yang, Y. Xu and C.S. Chen: “Human action learning via hidden markov model,” IEEE Transactions on Systems, Man, and Cybernetics, vol.27, no.1, pp.34-44, 1997.
- [12] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda and B. Schölkopf: “An introduction to kernel-based learning algorithms,” IEEE Transactions on Neural Networks, vol.12, no.2, pp.181-202, 2001.
- [13] A.P. Dempster, D.B. Rubin and N.M. Laird: “Maximum likelihood from incomplete data via the EM algorithm,” Journal of Royal Statistical Society, Series B (Methodological), vol.39, no.1, pp.1-38, 1977.
- [14] V.N. Vapnik: The Nature of Statistical Learning Theory (Statistics for Engineering and Information Science). Springer Verlag, 1995.
- [15] T. Joachims: “Estimating the generalization performance of an svm efficiently,” Proceedings of the Seventeenth International Conference on Machine Learning, pp.431-438, 2000.
- [16] P. Zegers and M.K. Sundareshan: “Trajectory generation and modulation using dynamic neural networks,” IEEE Transactions on Neural Networks, vol.14, no.3, pp.520-533, 2003.



佐藤知正 (Tomomasa Sato)

1948年9月22日生。1971年東京大学産業機械工学科卒業。1976年同博士課程修了後、電子技術総合研究所入所。1991年東京大学先端科学技術研究センター教授に移籍。1998年東京大学工学系研究科機械情報工学専攻に移籍。2000年同大学情報理工学系研究科知能機械情報学専攻に改組。知能ロボットの研究に従事。日本機械学会、IEEEなどの会員。

(日本ロボット学会正会員)



原田達也 (Tatsuya Harada)

1972年6月14日生。1996年東京大学工学部機械情報工学科卒業。2001年3月東京大学大学院工学系研究科機械工学博士課程修了。2000年1月から2001年12月まで日本学術振興会特別研究員。2006年4月より東京大学大学院情報理工学系研究科講師となり現在に至る。人の動作行動の観察・モデリング・知覚、生体計測、動作・生体モデリング、ワイヤレスネットワークを活用した支援に興味を持つ。博士(工学)。IEEE、計測自動制御学会等の会員。

(日本ロボット学会正会員)



久保寺秀幸 (Hideyuki Kubotera)

1980年8月30日生。2003年東京大学工学部機械情報工学科卒業。2003年より東京大学大学院情報理工学系研究科知能機械情報学修士課程において、模倣学習に基づく日常物体操作のロボットによる支援に関する研究を行った。2005年に修士課程を修了後、一般企業に就職し現在に至る。

(日本ロボット学会正会員)



森 武俊 (Taketoshi Mori)

1967年9月8日生。1990年東京大学工学部機械工学科卒業。1995年東京大学大学院工学系研究科博士課程修了(情報工学専攻)。同年東京大学先端科学技術研究センター助手。1998年同講師。2001年米国MIT客員研究員。現在東京大学大学院情報理工学系研究科助教授。動作認識、行動理解、パベイシブコンピューティング、ヒューマンコンテンツの研究をしている。博士(工学)。IEEE、ACM、電子情報通信学会、人工知能学会、計測自動制御学会等の会員。

(日本ロボット学会正会員)