解　説

液体マニピュレーションのためのロボット技術

Robotic Manipulation of Liquids

山　口　明　彦[*1]　クリストファー　アトケソン[*2]
Akihiko Yamaguchi[*1] *and* Christopher G. Atkeson[*2]

1. は　じ　め　に

液体の操作は食品マニピュレーションの中でも重要なタスクのひとつであり，ロボティクスにおいてもチャレンジングなタスクのひとつとして考えられている．ここでは，粉体や粒子など，液体に準ずる特性を持っているものも操作対象に含め，液体操作の中でも特に「注ぐ」操作について考える．「単一の液体を単一の容器から注ぐ」という操作なら，状況が限定的であるため作りこみによってロボットに実装可能である．一方で，様々な形状の容器，多様な液体や粉体に対して汎化するような行動方策の設計には困難を伴う．様々な形状の容器から液体を注ぐ操作を計画するために，qualitative physics（記号による物理学の定性的知識表現）の枠組みで推論（qualitative reasoning）によるアプローチが試みられたが [1]，粘性などの液体の特性を無視しており，現実世界の注ぐ操作を計画するには不十分である．人間の注ぐ動作からの模倣学習に関する研究には多くの事例があるが [2]〜[8]，液体や粉体，容器の多様性に対して汎化できていない．教師あり学習によるモデル学習とモデル予測制御の組み合わせによる液体の制御や [9]，深層強化学習による注ぐ動作の獲得 [10] などの研究事例もあるが，特に液体の多様性を扱うものは少ない．このように，汎用的な注ぐ操作はロボティクスや人工知能の分野において挑戦的なタスクであり，料理などの日常生活や食品加工工場などにおける産業でも重要なプロセスのひとつである．

本論文では，注ぐタスクをロボットで実現するために行っている取り組みの中から 3 つの研究を紹介する．(1) 人間の注ぐ動作を考察し，多数のスキルを集約した「ライブラリ」にもとづく動作生成が重要であるという仮説の検討．特に代替スキルの存在が，多様な状況に対して行動を汎化するカギであるとわかった [11] [12]．(2) 双腕ロボット PR2 においてスキルライブラリにもとづく「注ぐ動作」を実装し，この仮説が有力か検証し，どのような数理的手法が必要になるか理解するためのケーススタディを行った [11] [12]．(3) スキルライブラリにもとづく動作生成を一般化するための手法として，グラフ構造によって構造化されたダイナミクスに対するモデルベース型強化学習アルゴリズムを開発した [13]〜[15]．

原稿受付
キーワード：液体マニピュレーション, 強化学習, 模倣学習, スキルライブラリ

(a) Shaking A.　(b) Shaking B.　(c) Tapping.

Fig.1　Human demonstrations of shaking and tapping.

2. 人間の動作の考察から得た知見：スキルライブラリの重要性

人間がどのように多様な状況で注ぐタスクを達成しているか調べるために，人間の注ぐ動作を観察した．容器の位置姿勢，注がれた対象物の量を計測した．注ぐ対象は，注がれた量の画像認識の簡略化のため，乾燥した豆を用いた．この観察から，以下のようなことがわかった [11] [12]．

- 単一の操作の中でも分節化が起きている．例えば容器を傾けながら注ぐ場合，対象物の流れが最初に観測されるまで，対象物が注がれ始めた後，目標量まで注いだ後の 3 つのフェーズが観測された．
- 人間は多様な状況に対応するため，様々な手段を用いる．容器を傾けて対象物が注がれない場合，容器を振るといった代替手段を選択する．「容器を振る」動作にも種類があり，垂直に振る，角度をつけて振る，などの離散的な違いが見られた（Fig. 1）．
- 「多様な状況」にはタスク目標の多様性も含まれる．対象物を少量注ぐような場合（例えばインスタントコーヒーの粉を注ぐ場合），人間は両手を使って容器の縁を叩く操作を行った（Fig. 1）．

これらの考察で重要なポイントは「多様な状況に対応するため，様々な手段を用いている」ことである．この手段を「スキル」と呼ぶことにすると，多様な状況へ汎化するような行動方策のモデル化には，多数のスキルを集めたライブラリと，スキルライブラリにもとづく動作生成手法がカギであるという仮説が導かれる．スキルライブラリは，把持や運搬といった基本的なスキルも含むが，重要なのは容器を「傾けて注ぐ」「振りながら注ぐ」といった代替手段をスキルとして用意しておくことである．

3. 双腕ロボットによるケーススタディ：スキルライブラリの有効性を検証

双腕ロボット PR2 において，スキルライブラリにもとづく注ぐ動作を実装し，容器の形状や内容物の種類に対して汎化するか調べた [11] [12]．この際以下の指針で行った：

- 汎用的な注ぐタスクの達成を第一目標とする．
- 実装手法の理論的整合性については重要視せず，モーションプラニング，機械学習，最適化など，あらゆる技術の中から，最もシンプルに目的を達成できるものを組み合わせる．

3.1 スキルライブラリ

個々のスキルは人手で実装した．状況の多様性に対応できるように，各スキルにパラメータを導入した．定義した主なスキルは以下の通り：

把持 (Grasp): PR2 のパラレルグリッパを用いて把持を行う．左腕を用いることとする．注ぎ元の容器に対する把持位置姿勢が調整用のパラメータだが，容器の把持可能な部分を円柱でモデル化し，高さ，円柱の中心軸周りの姿勢，水平方向の姿勢の 3 つに制限した．

運搬 (Move): 注ぎ元の容器を把持した後，注ぐ位置姿勢まで容器を移動させる動作である．注ぐ位置姿勢は既に決定されていると仮定する．姿勢は現在から目標まで線形補間する．これは，容器から内容がこぼれることを防ぐためである．位置は現在と目標を結ぶ 3 次元軌道を 4 つの制御点を持つスプラインで表現し，5 つの調整可能なパラメータを設定する．

容器を傾けて注ぐ (Tipping): ある回転軸の周りの回転運動として定義する．回転軸は，注ぎ先の容器の口の上，注ぎ元の容器の縁付近に設定する．液体の流れが観測されるまで，目標量が注がれるまで，目標量を注いだ後の 3 つのフェーズで構成し，制御は状態遷移マシンによって記述した．回転軸の中心座標と回転軸の向きは，注ぎ元の容器と注ぎ先の容器それぞれについて設定する必要があり，3 つの調整可能なパラメータを設定した．これらは，以下の振りながら注ぐスキルでも同様に定義される．

容器を振りながら注ぐ-A (Shaking-A): 注ぎ元の容器を鉛直逆向きにして振りながら注ぐ動作．振る方向は鉛直方向であり，振る速さはパラメータである．

容器を振りながら注ぐ-B (Shaking-B): 容器を傾けて注ぐスキルと同様に注ぎ元の容器を傾けてゆき，最初の流れを観測した角度で振りながら注ぐ動作．振る方向は，1 次元のパラメータ ϕ によって決定する．振る速さも調整用パラメータである．

3.2 注ぐ動作全体のモデル化

前述のスキルを組み合わせて注ぐ動作全体をモデル化する．状態遷移マシンによって表現した．運搬スキル後に分岐 (Select) があり，注ぐスキルの代替スキルが接続されている．各注ぐスキルから Select に戻る経路があるのは，選択したスキルでうまく注げなかった場合に学習し，再選択する機能を実装しているためである．

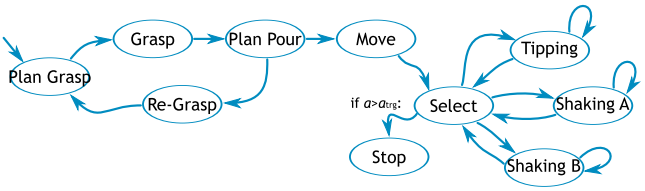


Fig.2 Pouring state machine.

3.3 スキル選択・パラメータの推論と学習

以上より，注ぐ動作の状態遷移マシンを定義し，個々のスキルを定義した．残る問題は，各スキルを実行するために必要なパラメータの決定と，状態遷移マシンの分岐における選択である．

代替スキルの選択（注ぐスキルの選択）は，容器の形状や注ぐ対象の物理特性に依存するため，解析的に方策を立てることは困難である．そこで学習による手法を用いた．

スキルパラメータの決定問題には 2 種類ある．ひとつは解析的な方策が導きやすいものであり，例えば運搬スキルにおける軌道パラメータの決定である．このようなパラメータは，推論（計画）によって決定した．ただしモデル化誤差などにより，得られた方策が失敗する場合もあるため，経験から学習する手段も追加した．他方で，振る速さなど，解析的なアプローチでは決定しにくいパラメータも存在する．これらについては，学習による手法を採用した．

3.4 スキルパラメータの学習

前述のスキルの中で，人手で決定することが困難なパラメータは，振りながら注ぐスキルにおける振る方向と振る速さである．評価関数は平均流量で定義し，これを最大化することが問題設定となる．パラメータベクトル $\mathbf{x}$ を入力とする評価関数 $f(\mathbf{x})$ を最大化する $\mathbf{x}$ を求める問題であり，$f(\mathbf{x})$ の関数形がモデル化されていないので，強化学習や導関数不要の最適化手法が必要となる．我々の実装では Hansen らによって提案された CMA-ES (Covariance Matrix Adaptation Evolution Strategy [16]) を用いた．CMA-ES は共分散行列にもとづいて確率的に生成された複数の探索点のそれぞれについて評価関数を評価し，その値から共分散行列を更新することを繰り返す最適化手法である．CMA-ES が提案する $\mathbf{x}$ について $f(\mathbf{x})$ を評価することを繰り返すだけで $f(\mathbf{x})$ を最大化する $\mathbf{x}$ が求められる．我々の文脈では $\mathbf{x}$ は振りながら注ぐスキルのパラメータであり，$f(\mathbf{x})$ はスキルをパラメータ $\mathbf{x}$ で実行した際に注がれた量を観測することで得られる．

3.5 代替スキルの選択

経験したことのない容器の形状や注ぐ対象に対してスキル選択を学習する際，内容物を周りに散らさないため，まず傾けて注ぐスキルを試し，流量が非常に少ない場合に振りながら注ぐスキルを試すといった戦略が考えられる．その場合でも，振りながら注ぐスキルのうち A, B いずれを選択すればよいかは自明ではない．このような Multi-armed bandit 問題に対して，各スキルを実行した際の評価を平均

と分散の形で保存しておき，Softmax 方策によって評価が高いスキルが優先的に選択されるような学習手法を用いる．ただし，Softmax では各スキルの評価の平均ではなく，平均プラス標準偏差の UCB (upper confidence bound) を用いる．標準偏差がそのスキルの「伸びしろ」をエンコードしていると考えられるためである．傾けて注ぐ，振りながら注ぐ-A, B のそれぞれの評価の平均を 1, 0.5, 0.5 のような割合で初期化しておくと，最初は傾けて注ぐスキルを選択する安全な学習戦略が表現できる．

3.6 スキルパラメータの推論と学習

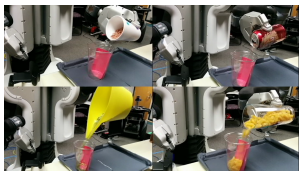
把持，再把持，運搬，および注ぐスキルにおける回転軸と回転中心を決めるパラメータが，推論対象である．ここでは評価関数をアドホックに設計し，最適化によってパラメータを決定するというアプローチを取る．それぞれのスキルについて，異なる評価関数を設計する．いずれの場合も衝突は回避すべきであり，逆運動学の解が存在する必要がある．これらは共通項として評価関数に組み込まれる．この他，把持では現在の姿勢に近い方がよい，把持姿勢が水平に近い方がよい，運搬では移動距離が短いほうがよい，などといった評価を追加し，評価関数を構成する．現在の観測状態とパラメータが入力されれば，これらの評価関数はモデルベースに計算できるため，適当な最適化計算によって推論が実行できる．

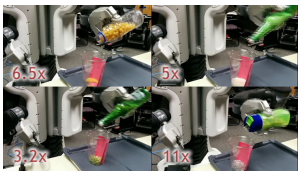
ここで設計した評価関数では，モデル化誤差や想定違いによって推論されたパラメータが適切に機能しないことがあった．そこで，経験から学習する手段として，失敗した場合にそのサンプルを使って評価関数を変形する方法を導入した．これは，失敗した際の状態とパラメータベクトルの周辺だけ評価値を下げる項を導入するという単純なものである．

3.7 実験

PR2 に実装し，実験を行った [11] [12]．注ぎ元の容器は様々な形状のものを用意し，注ぐ対象物も数種類用意した．ただし，失敗に対する機材の被害を最小化するため，乾燥した豆などを持ちいた．注ぎ先の容器は，注がれた量が RGB カメラによって計測できるように片側のみ透明に細工したものを用いた．容器の形状は円柱などのプリミティブの組み合わせでモデル化し，別途注ぎ口の情報を多角形でモデル化した．容器の位置姿勢は RGB-D カメラから推定した．参考動画(Fig. 3): `https://youtu.be/cGci9FOl680` `https://youtu.be/GjwfbOur3CQ` 各種実験により，以下が明らかとなった．

- 状態遷移マシンによる注ぐ動作は，目標量の変化に対して汎化できた．
- 注ぐ対象が同じでも，容器の形状が異なると適切に注げるスキルは異なるが，スキル選択の学習によって適切なスキルが選択できることがわかった．
- CMA-ES によるスキルパラメータの学習によって，振りながら注ぐスキルのパラメータが最適化できた．
- 最適化にもとづくスキルパラメータの推論によって，容器の初期位置や容器の形状に対して，把持，運搬スキルのパラメータや，注ぐスキルにおける回転軸と回転中心が計算できた．
- スキルパラメータの推論結果が失敗する場合にも，失敗からの学習により復帰できることがわかった．


(a) Pouring with tipping



(b) Pouring with shaking.


(c) Tapping.

Fig.3 Pouring experiments.

3.8 考察

このケーススタディから，スキルライブラリにもとづく動作生成のアプローチが有効であることが確認された．特に代替スキルの存在が多様な状況に汎化する際に効果的であるとわかった．設計した行動デザインの汎化性および適応能力について以下にまとめる．

- 把持，運搬，注ぐスキルについて，容器の形状や初期位置，異なる目標量の多様性に対して汎化できる．
- 注ぐスキルについて，未知の対象物と容器の形状の組み合わせに対して適応できる．
- 注ぐスキルについて，未知の対象物と容器の形状の組み合わせに対して汎化できない．

つまり，経験したことがない容器の形状や対象物が与えられたとき，スキル選択や注ぐスキルのパラメータを学習によって最適化できるが（適応できる），学習なしに推論することはできない（汎化できない），ということである．これは，スキル選択・パラメータの推論と学習の手法の欠陥である．また，これらの手法はアドホックに組み合わせているため，他のタスクへのスケールアップが難しい．

4. グラフ微分動的計画法と強化学習：スキルライブラリに基づく動作生成自動化

前節のケーススタディで明らかになった課題を解決するためには，以下のような数理的手法があればよい．

- スキルライブラリを知識ベースとして持つ，または推論・学習により構築できる．
- スキルの組み合わせを大まかに推論できる．
- 状況に応じた代替スキルの選択，スキルのパラメータを決定できる．

いずれもチャレンジングな問題だが，我々は 3 番目の問題に着手した．離散パラメータと連続パラメータが混在する動的計画問題であり，モデルが未知であるため，離散パラメータと連続パラメータが混在する強化学習である．

前節のケーススタディで，学習に基づく方策決定は，ごく一部に留まった．中心的だったのは，モデルに基づく推

論（最適化）である．このことは，モデルベース型強化学習を採用した直感的な動機だが，より掘り下げて考えると，モデルベース型強化学習には以下の利点がある [17].

- 状況の多様性に対して汎化しやすい [18].
- 学習するモデルは順モデルであり，異なるタスク間で共有しやすい.
- 順モデルは報酬に依存しないため，報酬関数を変更しても再学習の必要がない.

モデルベース型の課題として，モデルが不正確な場合の simulation biases がある [19]．モデルから行動を最適化する際（動的計画問題），将来の状態を予測するためにモデルを時間積分する．モデルが不正確だと，誤差が積分によって増幅され，将来予測が不確かとなる.

我々は，この課題がスキルの導入によって大幅に解消しうることに気づいた．つまりダイナミクスを時間微分の形式でモデル化するのではなく，スキルの実行前後の状態変化をモデル化する「スキルダイナミクス」を考えるのである．これにより，時間積分による誤差の蓄積を大幅に減らすことができる.

このアイディアにもとづき，

- スキルダイナミクスを機械学習によりモデル化 [14].
- グラフ構造を持つダイナミクスに対する微分動的計画法（Differential Dynamic Programming, 以下 DDP）を導出し，スキルの選択と連続行動の最適化 [15].

という組み合わせでモデルベース型強化学習を構成することを提案した.

スキルダイナミクスを導入したとしても, simulation bias の問題は完全には無くならない．そこで，スキルダイナミクスを確率的なプロセスとしてモデル化し，DDP も確率的なダイナミクスを考慮した手法をベースに開発した.

4.1 確率的 DDP のためのニューラルネットワーク

スキルダイナミクスを確率的なプロセスとしてモデル化し，確率的 DDP でスキルパラメータを計画する際に利用する．回帰モデルが利用できるが，確率的 DDP で利用するために，以下の要件を満たす必要がある.

(1) 予測誤差とノイズをモデル化できる.
(2) 入力が確率分布として与えられた場合に，出力の確率分布を計算できる.
(3) 出力の期待値の微分が計算できる.

Locally Weighted Regression などの回帰モデルはこれらの要件を満たす [20]．我々は近年の深層学習の成果を利用するため, ニューラルネットワークを拡張し, 確率的 DDP で利用可能にした [14].

(2)(3) については，ランダムサンプリングにより推定する方法もあるが, 確率的 DDP で利用するためには解析的に計算できることが望ましい．しかし一般にニューラルネットワークは活性化関数として非線形関数を含むため，特に (2) の計算が複雑になる．そこで，中間層で使用される活性化関数は ReLU (rectified linear units)，入力は多変量正規分布を想定し，各層の出力も多変量正規分布として近似した．これにより，スキルダイナミクスを多段に接続した場合にも，初期状態から後続の状態（確率分布）を計算できるため，確率的 DDP の順方向の伝搬計算が実行できる.

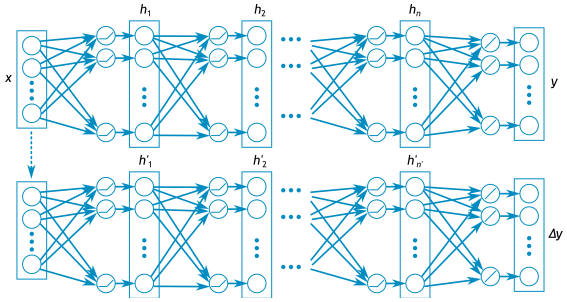


Fig.4 Neural network architecture for probabilistic DDP.

(1) を実現するためのニューラルネットワークの構造を Fig. 4 に示す．このネットワークは入力の正規分布の平均ベクトル $\mathbf{x}$ を入力した場合に，出力の正規分布の平均ベクトル $\mathbf{y}$ を出力するモデル $\mathbf{F}(\mathbf{x})$ と，誤差ベクトル $\mathbf{\Delta y}$ を出力するモデル $\mathbf{\Delta F}(\mathbf{x})$ から構成される．入力 $\mathbf{x}$ は共有される．誤差ベクトルは，平均ベクトルを出力するモデルのモデル化誤差およびノイズを含んだモデルとなっている．両者は切り分けていない.

学習データ $\{\mathbf{x}, \mathbf{y}\}$ が与えられたとき, 平均ベクトルを出力するモデル $\mathbf{F}(\mathbf{x})$ はこの学習データから直接学習できる．一方, 誤差ベクトルを出力するモデル $\mathbf{\Delta F}(\mathbf{x})$ の学習には工夫が必要である．提案手法では, まず $\mathbf{F}(\mathbf{x})$ を学習し, そのモデルを使って新たなデータセット $\{\mathbf{\Delta y}\} = \{\mathbf{f}_{\rm abs}(\mathbf{y} - \mathbf{F}(\mathbf{x}))\}$ を作る．ただし $\mathbf{f}_{\rm abs}$ は要素ごとの絶対値を出力する関数である．$\mathbf{\Delta F}(\mathbf{x})$ を $\{\mathbf{\Delta y}\}$ の包絡線を近似するように学習する.

4.2 Graph-DDP

グラフ構造を持つダイナミクスに対して確率的 DDP を拡張した Graph-DDP [15] では，スキルの組み合わせ方や順序が有向グラフ構造として与えられ，スキルダイナミクスが与えられる，もしくは前述のニューラルネットワークによって学習されていることを出発点とする．グラフ構造の向きはスキルの実行順序を表し，分岐は代替スキルを表す．Graph-DDP は，代替スキルから適切なものを選択し，各スキルを実行するために必要なパラメータを計算する．目的関数は，各スキル実行後の状態に対して設定される報酬関数の総和を最大化することと定義される.

従来，DDP は線形な構造のダイナミカルシステムに対して導出されていた [21]．DDP の計算は，初期状態から後続の状態および報酬を順次計算する順方向の計算と，チェーンルールによって後段から前段へ微分を逆伝搬する逆方向の計算から構成される．順方向と逆方向の計算を行うと，目的関数の行動パラメータに関する微分が得られ，勾配法によって行動パラメータを更新する．これらの計算を，収束条件が満たされるまで実行すると，所望の行動パラメータ系列が得られる.

Graph-DDP では，まずループの分解によってグラフをツリー構造に変換し，ツリー構造に対する順方向と逆方向の計算を導出することで，DDP と同様の計算を実現する.

以下に重要なアイディアを列挙する．

(1) ループの分解は，各ノードの訪問回数に上限を設け，訪問回数ごとに異なるノードであるとしてグラフを展開する．最初のノードをルートとするツリーが得られる．

(2) スキルダイナミクスを拡張し，様々な順方向の計算を表現できるようにする．パッシブなダイナミクス（制御対象ではないダイナミクス），キネマティクス，特徴点の検出などの空間変換，報酬などが含められる．表現の幅が広がる，報酬関数もグラフの分岐として扱えるため式の表現が簡潔になる，といった利点がある．

(3) グラフの分岐を拡張し，代替スキルだけでなく，結果に複数のモードがあるような事象（例えば把持スキルに対して，成功，滑って失敗，空中を掴んだ，などの結果），並列して起こる事象や計算の表現を許容する．それぞれの分岐で「グループ」と「遷移確率」を導入する．遷移確率とはある枝に遷移する確率である．グループとは枝の集合であり，あるグループに属する枝のうちひとつにのみ遷移することを表現する．逆に異なるグループに属する枝には，並列して遷移する．つまり，各グループで遷移確率の総和は 1 となる．報酬の計算と後続のスキルは別のグループとして表現され，これらは並列して起こる．代替スキル群は同じグループに属す．

(4) あるノード，そのノードを起点とする分岐，グループ，遷移確率モデル，各枝における順モデルをまとめたものを「分岐プリミティブ」と呼ぶ．グラフ構造は分岐プリミティブの集合として記述できる．

(5) ツリー構造に対する DDP の導出は，分岐プリミティブに対して順方向および逆方向の計算を導出することが中核となる．

(6) 一般に順モデルは非線形であり，DDP の評価関数（報酬の合計の期待値）は，スキルパラメータに対して非線形で局所解が多い．DDP は勾配探索であるため，局所解に陥りやすい．そこで多点探索と組み合わせる．

(7) 代替スキルの選択は離散パラメータの最適化であるため勾配法で扱えない．これらのパラメータは多点探索によってランダムに探索される．

分岐プリミティブについて，順方向の計算は遷移確率と順モデルの計算である．逆方向の計算は以下のようになる．

$$J_k(\mathbf{x}_k) = \sum_b \mathbf{F}_k^{\mathrm{P}b}(\mathbf{x}_k) J_{k+1}^b(\mathbf{F}_k^b(\mathbf{x}_k)) \qquad (1)$$

$$\frac{\partial J_k}{\partial \mathbf{x}_k} = \sum_b \left[\frac{\partial \mathbf{F}_k^{\mathrm{P}b}}{\partial \mathbf{x}_k} J_{k+1}^b + \mathbf{F}_k^{\mathrm{P}b} \frac{\partial \mathbf{F}_k^b}{\partial \mathbf{x}_k} \frac{\partial J_{k+1}^b}{\partial \mathbf{x}_{k+1}^b} \right] \qquad (2)$$

ここで k は注目しているノード，J_k はノード k から先の評価関数，b は分岐，$\mathbf{x}_k$ は k における状態，$\mathbf{F}_k^{\mathrm{P}b}$ は b に遷移する確率，$\mathbf{F}_k^b$ は b の順モデルを表す．表記を簡単にするため，状態ベクトル $\mathbf{x}_k$ には行動パラメータも含めている．ここではわかりやすさのため，順モデルは決定的なプロセスとしているが（分岐のみ確率的），入出力を確率分布とする拡張も自然に計算できる．通常の DDP と同様，ツリー全体に渡って順方向の計算を行い，その後逆方向の計算を行うと，評価関数のパラメータに関する勾配が求まる．この勾配を用いてパラメータを更新するまでが DDP における 1 反復である．これを収束するまで繰り返せば，スキルパラメータが最適化される．多点探索と組み合わせて，異なる初期値およびスキル選択の離散パラメータについて探索し，もっとも評価が高いものを選ぶことで，スキルパラメータおよびスキル選択が得られる．

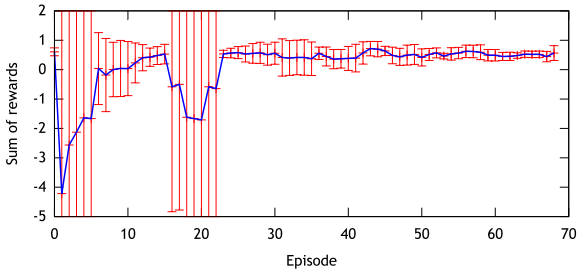


Fig.5 Learning curve; sum of rewards per episode (mean±1-SD).

4.3 実験

2 つの実験を紹介する．ひとつ目は確率的ニューラルネットワークと線形構造に対する確率的 DDP を組み合わせた強化学習で，「振りながら注ぐ」スキルのパラメータなどを最適化する [14]．この実験では PR2 を用いた．Fig. 5 は学習曲線を示し，横軸にエピソード回数（1 エピソードは注ぐタスクの 1 試行），縦軸に評価値をプロットしている．図の曲線は 5 回実験を行った際の平均 ±1 標準偏差である．この実験では on-line, on-policy で学習を行っており，1 エピソードごとにスキルダイナミクスをモデル化するニューラルネットワークを更新する．各エピソードにおいて，初期状態を観測した後，最新のスキルダイナミクスモデルを使って，スキルパラメータを DDP によって最適化する．報酬は注ぐ量の目標との誤差および容器からこぼした量に対するペナルティである．スキルはあらかじめ手作業で実装されており，調整パラメータを強化学習しているため，学習は少ないエピソードで行われた．学習の初期段階では容器からこぼすことによって発生するペナルティが支配的で，学習が進むにつれてこぼさずに注ぐパラメータを計画できるようになった．参考動画: `https://youtu.be/aM3hE1J5W98`

ふたつ目の実験は，確率的ニューラルネットワークと Graph-DDP を組み合わせた強化学習の検証をシミュレーションで行ったものである [15]．この実験では傾けて注ぐスキルと振りながら注ぐスキルを代替スキルとして定義した．シミュレータは Open Dynamics Engine を利用しており，液体は小さな球の集まりで表現している．これらの球の物理特性および球間に加える仮想力によって粘性など様々な特性の液体をシミュレーションできる．容器口のサイズも変化させる．同一の容器であっても，液体のパラメータによっては傾けて注ぐスキルで注げる一方，別の液体のパラメータでは傾けただけでは液体が出てこず，振りながら注ぐスキルが有効である状況をシミュレートできており，現実の液体のダイナミクスに近い特性が再現できていると考える．

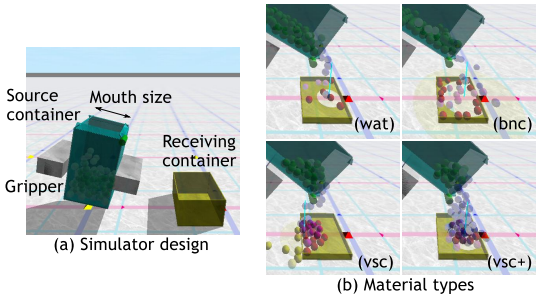


Fig.6 (a) Pouring simulation setup. (b) Types of poured materials.

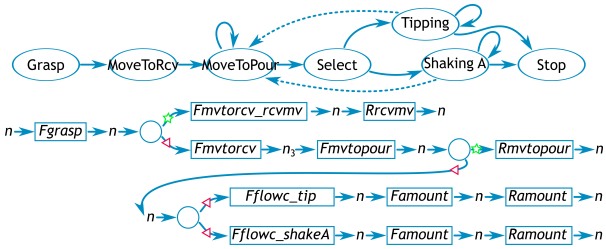


Fig.7 State machine of pouring behavior (top) and its graph-structured dynamical system (bottom). F_* denotes a skill dynamical system, and R_* denotes a reward model.

Fig. 7 にスキルの組み合わせを表現する状態遷移マシンおよび対応するダイナミクスモデルを示す．各スキルは手作業でプログラムした．それぞれ，調整パラメータを持つ．各スキルダイナミクスは確率的ニューラルネットワークで学習する．上の実験と同様 on-line, on-policy で学習を行った．この実験では，効率的に学習ができるように，reward shaping や学習のスケジューリング（簡単な状況から段階的に学習）を検討した．最終的に得られた結果は，容器の形状や液体のパラメータに対して汎化するものだった．つまり，経験したことのない容器の形状や液体パラメータの組み合わせが与えられても，スキルの選択やパラメータを最適化できた．参考動画: `https://youtu.be/_ECmnG2BLE8`

5. ま と め

著者らの液体マニピュレーションの研究を紹介した．容器の形状や液体・粉体の物理特性の多様性を考慮すると問題はとたんに難しくなる．著者らはスキルライブラリ，特に代替スキルの導入が重要であるとの仮説にもとづき，検証実験や，スキルライブラリを扱うための数理的手法の開発を行った．詳細は各文献を参照されたい．

参 考 文 献

[1] E. Davis, “Pouring liquids: A study in commonsense physical reasoning,” *Artificial Intelligence*, vol. 172, no. 12, pp. 1540–1578, 2008.
[2] M. Mühlig, M. Gienger, S. Hellbach, J. J. Steil, and C. Goerick, “Task-level imitation learning using variance-based movement optimization,” in *the IEEE International Conference on Robotics and Automation (ICRA’09)*, 2009, pp. 1177–1184.
[3] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, “Learning and generalization of motor skills by learning from demonstration,” in *the IEEE International Conference on Robotics and Automation (ICRA’09)*, 2009, pp. 763–768.
[4] M. Tamosiunaite, B. Nemec, A. Ude, and F. Wörgötter, “Learning to pour with a robot arm combining goal and shape learning for dynamic movement primitives,” *Robotics and Autonomous Systems*, vol. 59, no. 11, pp. 910–922, 2011.
[5] K. Kronander and A. Billard, “Online learning of varying stiffness through physical human-robot interaction,” in *the IEEE International Conference on Robotics and Automation (ICRA’12)*, 2012, pp. 1842–1849.
[6] O. Kroemer, E. Ugur, E. Oztop, and J. Peters, “A kernel-based approach to direct action perception,” in *the IEEE International Conference on Robotics and Automation (ICRA’12)*, 2012, pp. 2605–2610.
[7] L. Rozo, P. Jiménez, and C. Torras, “Force-based robot learning of pouring skills using parametric hidden Markov models,” in *the IEEE-RAS International Workshop on Robot Motion and Control (RoMoCo)*, 2013.
[8] S. Brandl, O. Kroemer, and J. Peters, “Generalizing pouring actions between objects using warped parameters,” in *the 14th IEEE-RAS International Conference on Humanoid Robots (Humanoids’14)*, Madrid, 2014, pp. 616–621.
[9] Z. Pan and D. Manocha, “Feedback motion planning for liquid pouring using supervised learning,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 1252–1259.
[10] C. Do, C. Gordillo, and W. Burgard, “Learning to pour using deep deterministic policy gradients,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 3074–3079.
[11] A. Yamaguchi, C. G. Atkeson, S. Niekum, and T. Ogasawara, “Learning pouring skills from demonstration and practice,” in *the 14th IEEE-RAS International Conference on Humanoid Robots (Humanoids’14)*, Madrid, 2014, pp. 908–915.
[12] A. Yamaguchi, C. G. Atkeson, and T. Ogasawara, “Pouring skills with planning and learning modeled from human demonstrations,” *International Journal of Humanoid Robotics*, vol. 12, no. 3, p. 1550030, 2015.
[13] A. Yamaguchi and C. G. Atkeson, “Differential dynamic programming with temporally decomposed dynamics,” in *the 15th IEEE-RAS International Conference on Humanoid Robots (Humanoids’15)*, 2015.
[14] ——, “Neural networks and differential dynamic programming for reinforcement learning problems,” in *the IEEE International Conference on Robotics and Automation (ICRA’16)*, 2016.
[15] ——, “Differential dynamic programming for graph-structured dynamical systems: Generalization of pouring behavior with different skills,” in *the 16th IEEE-RAS International Conference on Humanoid Robots (Humanoids’16)*, 2016.
[16] N. Hansen, “The CMA evolution strategy: a comparing review,” in *Towards a new evolutionary computation*. Springer, 2006, vol. 192, pp. 75–102.
[17] A. Yamaguchi and C. G. Atkeson, “Model-based reinforcement learning with neural networks on hierarchical dynamic system,” in *the Workshop on Deep Reinforcement Learning: Frontiers and Challenges in the 25th International Joint Conference on Artificial Intelligence (IJCAI2016)*, 2016.
[18] E. Magtanong, A. Yamaguchi, K. Takemura, J. Takamatsu, and T. Ogasawara, “Inverse kinematics solver for android faces with elastic skin,” in *Latest Advances in Robot Kinematics*, Innsbruck, Austria, 2012, pp. 181–188.
[19] J. Kober, J. A. Bagnell, and J. Peters, “Reinforcement learning in robotics: A survey,” *International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.
[20] S. Schaal and C. Atkeson, “Robot juggling: implementation of memory-based learning,” in *the IEEE International Conference on Robotics and Automation (ICRA’94)*, 1994, pp. 57–71.
[21] D. Mayne, “A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems,” *International Journal of Control*, vol. 3, no. 1, pp. 85–95, 1966.

山口 明彦（Akihiko Yamaguchi）
2006 年京大工学部電気電子工学科卒．2008 年奈良先端大情報科学研究科博士前期課程修了．2011 年同博士後期課程修了．工学博士．行動獲得の研究に従事．2011～15 年奈良先端大情報科学研究科特任助教，2015～17 年カーネギーメロン大学ロボット研究所研究員，2017 年～東北大学大学院情報科学研究科助教．`http://akihikoy.net/`
（日本ロボット学会正会員）

クリストファー アトケソン（Christopher G. Atkeson）
カーネギーメロン大学ロボット研究所およびヒューマン・コンピュータ・インタラクション研究所教授．ハーバード大学で応用数学（コンピュータ・サイエンス）の修士号，M.I.T. で脳科学および認知科学の博士号を取得．1986 年～M.I.T. 助教を経て准教授，1994 年～ジョージア工科大学准教授，2000 年～カーネギーメロン大学．