

間引きを用いたシュートモーション学習

Shot Motion Learning Using Thinning-out

小林 隼人¹ 畑埜 晃平² 石野 明¹ 篠原 歩¹

Hayato Kobayashi¹, Kohei Hatano², Akira Ishino¹, and Ayumi Shinohara¹

¹ 東北大学大学院情報科学研究科 ² 九州大学大学院システム情報科学研究科

¹ Graduate School of Information Sciences, Tohoku University

² Graduate School of Information Science and Electrical Engineering, Kyushu University

{kobayashi@shino., ishino@, ayumi@}ecei.tohoku.ac.jp and hatano@i.kyushu-u.ac.jp

Abstract

Shooting is one of the most important skills in soccer games, as it directly affects scoring points. In the four-legged robot league, we must develop accurate and strong shot motions only by changing the parameters of the motions, since the hardware is limited to AIBO. This task consumes much of developers' time and energy and is needed whenever the environment (e.g., friction of the field) is changed. By using machine learning, we can efficiently reduce these burdens. Existing learning methods, however, take much evaluation time for each trial in the motion learning. In this paper, we propose a new concept, thinning-out, for reducing the number of trials in the motion learning. Thinning-out means to skip over such trials that are unlikely to improve motions, in the same way that gardeners thin-out weak seedlings. We show that our thinning-out technique significantly reduces the number of trials. In addition, we show that our virtual robots can acquire a sophisticated motion that is much different from the initial motion as the result of utilizing a hybrid learning method combining meta-heuristics and the thinning-out technique.

1 はじめに

ロボットが実世界で機能するには、未知の環境に適応する能力、つまり学習能力が必要である。この能力は、RoboCup サッカーにおいても重要な役割をしめる。特に四足ロボットリーグやヒューマノイドリーグにおける身体ロボット

(肢体を持つロボット)は、歩く、シュートする、パスするといった基本技術を獲得するための能力が必要である。この、身体ロボットが基本技術を学習する能力は、身体学習 (physical learning) として知られており、重要視されている。

ここ数年でも、多くの身体学習に関する研究が存在する。Kim と Uther [2]は、歩行の軌跡を四角形でモデル化して、ロボットに高速な歩行を学習させた。Saggar ら [6]は、歩行の軌跡を楕円形でモデル化して、ロボットに視覚情報を使いながら安定した歩行を学習させた。Fidelman と Stone [1]は、ボールを掴む技術の学習方法を提案した。この学習は、1 層目に歩行学習、2 層目にボールを掴む技術の学習を導入した階層学習である。Kobayashi ら [4]は、強化学習で動いているボールのトラップ技術を 1 次元で学習させた。

本論文では、Zagal と Solar [8]と同様に、シミュレータ上でロボットにシュートモーションを学習させる。素早く強力なシュートは直接得点に結びつくため、シュートモーションの作成はRoboCup サッカーにおいて重要な作業のひとつである。しかしシュートモーションの学習は問題のモデル化が難しいため、上で述べた他の作業と比較すると難しい問題といえる。Zagal と Solar は、動作モデルを仮定することなくシュートモーションを直接学習させたが、彼らの方法は足の関節のみを対象にしており、探索する次元数も固定している。我々も同様にモデルを置かないが、我々の方法はすべての関節を対象としており、次元数も固定しないため、どんなモーションにも適用することができる。しかし、それゆえに学習には多くの試行回数を必要とってしまう。一回の試行に非常に時間がかかるロボットの学習においては、この問題は深刻である。

本論文では、この問題を解決するために、学習過程における試行回数を削減する間引き (thinning-out) を提案する。間引きは、決定木学習における枝刈り (pruning) と同様の概念であり、園芸家が悪い苗を間引くように、学習

過程において結果に貢献しない試行を省くことを意味する。我々は、間引きを使うことで効果的に試行回数を削減できることを示す。この間引きは、メモリーベース学習 (memory-based learning) の一種であり、Memory-based Fitness Evaluation Genetic Algorithm (MFEGA) を提案した Sano[7]らと動機は同じである。しかし、我々の提案する間引きは、スコア関数の形を推定しながら試行自体を直接省くという点で完全に彼らの手法とは異なる。

本論文の構成は以下のとおりである。第2節では、我々のフレームワークにおけるモーション再生の仕組みとシュートモーションを作成するためのモーションエディタを説明し、本論文で取り扱う学習問題を定式化する。第3節では、間引きの方法について説明し、それを実現するための2つの手法を提案する。そして、間引きの候補点を選択するためのメタヒューリスティクスについて説明する。第4節では、提案した学習手法を用いて、シミュレータ上のロボットにシュートモーションを学習させ、その実験結果について考察する。第5節では、全体のまとめと今後の課題について述べる。

2 シュートモーションの学習

2.1 シュートモーション

AIBOのモーションは、首や足の15個の関節角度で構成されるフレーム (frames) を8msごとにOVirtualRobotへ送り込むことで実現される。OVirtualRobotはプロキシオブジェクトの一種であり、AIBOのソフトウェア開発キットOPEN-Rで定義されている。我々のフレームワークでは、これらのフレームはキーフレーム (key-frames) によって生成される。キーフレームは、それぞれのモーションの骨組みとなる特徴的なフレームのことである。たとえば、キックモーションはボールを押し出すために足を一度後ろに下げてから前に出さなければならないので、少なくとも2つのキーフレームが必要である。我々はそれぞれのキーフレームに対して補間数を指定し、それを元に線形補間で完全なフレームを生成する。それゆえ、我々のモーションを実行するには、補間数を n とすると $8n$ msが必要である。

2.2 モーションエディタ

我々は、モーションを簡単に作成し調整するために、図1のようなスプレッドシートを持つモーションエディタを作成した。このエディタを使うことで、我々はそれぞれのモーションのためのキーフレームとその補間数を指定することができる。しかし、通常はロボットを見ただけではその関節角度の正確な値を推定することができないため、手作業で関節角度のすべてを入力するのは困難である。したがって、エディタにロボットの関節角度をキャプチャする機能を持たせた。我々のロボットプログラムは、

肉球 (足裏の接触センサ) をクリックすることでその足を脱力させることができ、もう一度クリックすることで固定することができるため、我々はパラパラ漫画やクレイアニメを作成するのと同様に、直感的にさまざまなモーションを作成できる。

2.3 学習問題

本章では、キーフレームを直接探索空間として使うことでシュートモーションの学習を行う。我々は、学習後のモーションの骨組みを決める大雑把なモーションを作成するだけでよい。そうすれば、その骨組みから、その近傍のよいシュートを学習することができる。今回は補間数を学習の対象としないため、キーフレーム数を n とすると、取り扱う探索空間は $15n$ 次元となる。

シュートを評価するためのスコア関数は、ボールの相対距離 r_b 、ボールの相対角度 θ_b として以下のように設計した。

$$score(r_b, \theta_b) = r_b \cdot \left(1 - \frac{|\theta_t - \theta_b|}{\theta_c} \right)$$

この式は、シュートの目的角度 θ_t とボールの相対角度との差に反比例して、スコアを線形的に小さくする。 θ_c はスコアの減少具合と決める定数であり、本論文では $\theta_c = 45$ とした。

3 学習方法

3.1 間引き

本節では、不必要な試行を間引く方法について議論する。まず我々は、スコア関数が探索空間上で連続で、ある程度なめらかであることを仮定する。ロボットの動きは連続であり、パラメータの小さな変更はスコアにそれほど影響を及ぼすことはないため、この仮定は妥当なものといえる。この仮定に基づき、我々はスコア関数の局所的な形を推定する。与えられた試行の候補点に対して、我々は試行の履歴中の最近傍点との距離を使うことで候補点のスコアを推定する。もし推定されたスコアの上限が現在の最高点よりも低い場合は、その候補点を間引く。このとき、現在の最高点との距離を考慮にいれていないことに注意されたい。我々は真の最高点が現在の最高点の近くにあることを仮定していない。それゆえ、高いスコアが得られると予測された点は、たとえ現在の最高点よりも遠かったとしても試行される可能性がある。したがって、我々の方法は局所解に陥りにくいという意味で頑健であるといえる。

ここで我々は、リプシッツ条件 (Lipschitz condition) を持ちいて、スコア関数の局所的ななめらかさを定義する。リプシッツ条件は、微分積分の標準的な教科書にあるように定数 c に対する c -リプシッツ連続を定義するものだが、本論文ではその自然な拡張として関数 g に対する g -リプシッツ連続を定義する。

HT	HP	NT	RF1	RF2	RF3	LF1	LF2	LF3	RR1	RR2	RR3	LR1	LR2	LR3	n
3	93	50	135	93	127	135	93	127	120	93	120	120	93	127	
-83	-93	-20	-120	-15	-30	-120	-15	-30	-135	-15	-30	-135	-15	-30	
-80	-1	-11	12	86	61	0	89	40	0	87	59	16	70	46	30
-75	59	-9	13	69	72	1	43	73	-20	48	89	12	52	65	30
-78	-41	49	3	26	73	-25	-8	58	-29	46	100	-40	16	81	30
-77	32	-10	37	51	68	-61	46	56	-49	1	98	5	29	71	30
-79	-7	50	14	7	68	-44	65	-6	-101	50	120	-5	62	72	30

Figure 1: 作成したモーシオンエディタの一部．スプレッドシートの 1 行がキーフレームを意味する．HT, HP, ..., LR3 の列はそれぞれの関節角度を意味し, n の列はそのキーフレームまでの補間数を意味する．

定義 1 (リプシッツ条件, Lipschitz condition) X を探索空間とする． $f: X \rightarrow R$ を探索空間上のスコア関数とする．ある関数 $g: R \rightarrow R$ について, どのような $x_1, x_2 \in R$ に対しても以下の条件を満たすとき, f は g -リプシッツ連続 (g -Lipschitz continuous) であるという．

$$|f(x_2) - f(x_1)| \leq g(|x_2 - x_1|).$$

f を g -リプシッツ連続とする．そのとき, どのような x_1, x_2 に対しても, 次のように $f(x_1)$ の上限が得られる．

$$f(x_1) \leq f(x_2) + g(|x_2 - x_1|).$$

これより, 我々はスコア関数 f が, ある g に対して g -リプシッツ連続であると仮定する．我々の間引き戦略は, 候補点のスコアの上限が得られるように, スコア関数 f を特徴づける関数 g を推定することである．もしもその上限が現在の最高点よりも小さいとき, 我々はその候補点を試行する必要はない．関数 g を推定する方法の詳細は後に述べる．我々の間引き条件は以下のように定式化される．

定義 2 (間引き条件, Thinning-out condition) x_b を現在の最高点とする． x_c を試行の候補点とする． x_n をその候補点の最近傍点とする．ある推定関数 $\hat{g}: R \rightarrow R$ について, 次の条件を満たすとき, x_c は \hat{g} に関して間引き条件 (thinning-out condition) を満足するという．

$$f(x_n) + \hat{g}(|x_c - x_n|) \leq f(x_b)$$

ここで, 我々は g を推定する 2 つの手法を提案する．

最大傾斜法 (Max Gradient Method)

まず, X 上の任意の 2 点 x_1, x_2 について, スコア関数 f の最大の傾斜

$$c = \max_{x_1 \neq x_2, x_1, x_2 \in X} \frac{|f(x_2) - f(x_1)|}{|x_2 - x_1|}$$

を知っていると仮定する．このとき, $g(d) = c \cdot d$ によって定義される関数によって, f が g -リプシッツであることは容易にわかる．それゆえ, g は安全に候補を間引くこと

ができる．しかし実際には, c 自身を知ることはできないので, 我々は過去の試行におけるすべての 2 点における最大傾斜でそれを置き換える．これはおそらく, 十分な試行のあとには c のよい近似となるはずである．したがって我々はこの方法を最大傾斜法 (MG) と呼ぶ．

この方法は, スコア関数上で起伏の一番大きい場所を基準とする単純なものだが, 一番安全な方法である．しかし, 関数 g が返す値は多くの場合大きすぎるため, 起伏の大きいスコア関数上ではほとんど候補を間引くことはできないと予想される．

差分収集法 (Gathering Differences Method)

差分収集法 (GD) は, 試行の履歴中の任意の 2 点において, それらの差の小さなものから昇順で, 候補点と最近傍点との距離以上になるまで, それらのスコアの差分を足し合わせることで関数 g を推定する．Algorithm 1 にこの方法のアルゴリズムを示す．

この方法は, 現在の最高点付近のスコア関数の局所的な形を推定するので, 多くの候補点を間引くことができる．しかし, この方法は単なるヒューリスティクスであり理論的妥当性はないため, 間違っ間引く可能性は MG よりも高いと予想される．

3.2 メタヒューリスティクス

我々が提案した間引きは単に候補点を省く手法であるため, その候補点を取ってくるための効率のよいサンプリング手法が必要である．シュートモーシンの学習問題ではスコア関数が未知であるため, いくつかのメタヒューリスティクスを利用する．Kohl と Stone [5] は, 遺伝的アルゴリズム (Genetic Algorithm: GA), 山登り法 (Hill Climbing: HC), 政策勾配法 (Policy Gradient: PG), アメーバ法 (Amoeba) の 4 つのメタヒューリスティクスを用いて, 四足歩行の学習をおこなった．本研究でも, これらのメタヒューリスティクスを利用するが, アメーバ法だけは除いた．これは, アメーバ法が決定的アルゴリズムであるため, 間引きを直接適用することができないからである．それゆえ, アメーバ法の代わりに, よく知

Algorithm 1: Gathering differences method

input : distance and the set Hist of pairs $(x, f(x))$ observed so far
output: An inferred value of $g(\text{distance})$

initialize Diff as a map from \mathcal{R} to \mathcal{R} ;
foreach key1, value1 *in* Hist **do**
 foreach key2, value2 *in* Hist **do**
 Diff [|key1 - key2|] \leftarrow |value1 - value2|;
 end
end
sum_diff_key \leftarrow 0;
sum_diff_val \leftarrow 0;
foreach key, value *in* Diff *in ascending order w.r.t.* key **do**
 sum_diff_key \leftarrow sum_diff_key + key ;
 sum_diff_val \leftarrow sum_diff_val + value ;
 if sum_diff_key \geq distance **then**
 return sum_diff_val ;
 end
end
return ∞ ;

られたメタヒューリスティクスの一つである疑似焼き鈍し法を (Simulated Annealing: SA) を実験に加えた . GA, HC, PG に関しては Kohl と Stone のアルゴリズムと同じものを使用するため, SA のアルゴリズムの詳細に関してだけ述べる .

疑似焼き鈍し法 (Simulated Annealing)

SA は, Kirkpatrick ら [3] によって提案された金属工学における焼き鈍し処理を模した最適化手法である . Algorithm 2 に, 間引きを用いた SA のアルゴリズムを示す . 我々の提案する間引きはメタ戦略であるため, 他の手法に関しても, 同様の方法で簡単に間引きを適用することができる .

4 実験

4.1 シュートモーションの学習

本節では, 間引きが身体学習に適応できることを確かめるために, 物理シミュレーションによるシュートモーションの学習を行う . シミュレータとして, Zaratti ら [9] が開発した 3D シミュレータを拡張して使用した . このシミュレータは, 非常に使いやすいユーザインターフェイスを持っており, 実ロボットと同じように各関節角度を送り込むことで簡単に仮想ロボットを動かすことができる . このシミュレータは完全な実環境を実現するものではないが, ロボットを傷つけないだけでなく, 実ノイズに悩まされることなく再現性のある実験ができるため, 新しい手法の性能を検証するには敵した環境である .

Algorithm 2: Simulated annealing with thinning-out

input : num_trials, initial_motion, initial_temperature, cooling_factor
output: an optimized motion

state \leftarrow initial_motion ;
Hist [state] \leftarrow Evaluate(state);
temperature \leftarrow initial_temperature ;
while true *in* num_trials **do**
 new_state \leftarrow a random perturbation of state ;
 while new_state *satisfies the thinning-out condition* **do**
 new_state \leftarrow a random perturbation of state ;
 end
 new_score \leftarrow Evaluate(new_state);
 if new_score < score **then**
 if true *with a probability of*
 $\exp\left(-\frac{\text{score}-\text{new_score}}{\text{temperature}}\right)$ **then**
 state \leftarrow new_state ;
 score \leftarrow new_score ;
 end
end
temperature \leftarrow temperature \cdot cooling_factor ;
Hist [new_state] \leftarrow new_score ;
end
return Max(Hist);

本研究では, 右前足で左斜め方向にボールを打つシュートモーションを初期モーションとして用いた . このモーションは 5 個のキーフレームを使用するため, 探索空間は 75 次元となる . 実験はメタヒューリスティクスにより 50 個の候補点を選ぶことを 10 回行い, 間引き率とその失敗率の平均を出した (今回は失敗率を出すために間引かれた候補点も評価した .) 本実験はシミュレータ上で行うが, 物理シミュレーションには複雑な計算が必要であるため, それでもかなりの時間が必要となる . 実際, 今回の一回の実験には数十時間を要した . それゆえ, 時間のかかる試行自体を削減する間引きは, 実環境だけではなく, 仮想環境においても有効に働く .

表 1 に, 2 種類の間引き (MG, GD) を適用した各種メタヒューリスティクス (GA, SA, HC, PG) による実験結果を示した . 表の MG と GD を比較すると, MG は比較的安全だがあまり候補点を間引けず, GD は失敗を恐れずに多くの候補点を間引けることが分かる . また, GD は 9 割以上の成功率で試行回数を通常の半分程度にできることが分かる . 今回は 50 個の候補点での結果であるため間引き率はそれほど高くないが, 候補点が多いほど間引き率を高くできることが分かっている .

MG と GD の性質を比較するために, SA にそれぞれの

手法を適用した実験結果のグラフを図 2 に示す。青い丸は実際に試行された候補，黄色い四角は正しく間引かれた候補，赤い十字は間違えて間引かれた候補を表す。表 1 と同様に，MG の結果 (a) では高い確率でスコアが低い候補点だけが間引かれており，GD の結果 (b) ではある程度高いスコアを持つ候補点も間引かれていることが分かる。GD の結果 (b) に表れているように，誤って間引いたとしても，未来の試行でより高いスコアを持つ候補点を見つける可能性がある。

図 3 に，初期モーションと学習の結果得られたモーションを示す。初期モーション (a) はほとんど左前足だけしか使っていなかったが，学習したモーション (b) は体全体を使っている。また，モーション (c) は体重を乗せて，足を降り下ろすモーションである。これらの学習後のモーションはどちらも体全体を動かしているため，次元数を減らすためにモデル化するような既存の学習方法では獲得することが困難である。学習前，学習後それぞれのモーションの動画は，我々のサイト (<http://www.jollypochie.org/papers/>) で閲覧可能である。

5 まとめと今後の課題

本論文では，それぞれの試行に多くの評価時間を要する問題に対して有効な間引きを提案した。また，間引きで利用する関数 g を推定するための 2 つの手法 (MG と GD) を提案した。シュートモーション学習の実験により，本手法がさまざまなメタヒューリスティクスに関して有効であり，汎用的なメタ戦略であることを確かめた。

今後は，間引きが実ノイズに耐えうるかどうかを確かめるために，実ロボットでの実験が必要である。しかし，実環境でのシュートモーション学習は，毎回の試行後に転がったボールの距離や角度などを測定し，それを注意深く元の場所に戻す作業を行わなければならないため，非現実的である。それゆえ我々は，Kobayashi ら [4] のように，ロボットに自律学習 (autonomous learning) を行わせることを検討している。前方シュートの自律学習は，彼らの手法をそのまま利用することができる。他の方向のシュートに関しても，天井カメラなどを用いれば可能である。自律学習という概念は，通常は現実的ではなかったシュートモーション学習などを実現するだけでなく，試合中に，つまりオンラインで身体動作を学習し，向上させる可能性を秘めている。実機リーグにおいては，戦略のオンライン学習はいくつか報告されているが，身体動作のオンライン学習はいまだ行われていない。オンライン学習能力は，未来の RoboCup に必要とされる重要な能力であるため，取り組みがいのある問題だといえる。

また，本章で提案した 2 つの手法は一長一短であったため，我々はより精度の高い関数 g の推定方法を考える必要がある。我々は，重みつき平均を用いた手法など，2

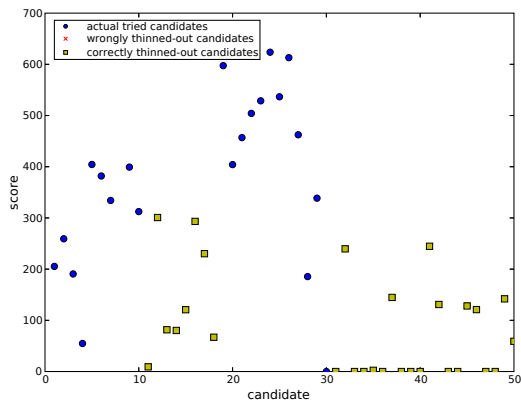
つの手法以外にもいくつかの推定方法を思いついたが，どれももうまくいかなかった。今後は，個々の問題に依存したヒューリスティクスをうまく活用することによって，もっとよい推定ができないかを考えたい。

参考文献

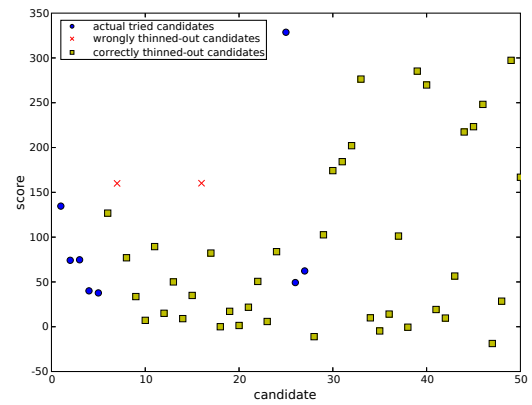
- [1] P. Fiedelman and P. Stone. The Chin Pinch: A Case Study in Skill Learning on a Legged Robot. In *RoboCup 2006: Robot Soccer World Cup X*, LNAI. Springer-Verlag, 2007. to appear.
- [2] M. S. Kim and W. Uther. Automatic Gait Optimisation for Quadruped Robots. In *Proceedings of 2003 Australasian Conference on Robotics and Automation*, pp. 1–9, 2003.
- [3] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by Simulated Annealing. *Science*, 220(4598):671–680, May 1983.
- [4] H. Kobayashi, T. Osaki, E. Williams, A. Ishino, and A. Shinohara. Autonomous Learning of Ball Trapping in the Four-legged Robot League. In *RoboCup 2006: Robot Soccer World Cup X*, LNAI. Springer-Verlag, 2007. to appear.
- [5] N. Kohl and P. Stone. Machine Learning for Fast Quadrupedal Locomotion. In *The Nineteenth National Conference on Artificial Intelligence*, pp. 611–616, 2004.
- [6] M. Saggat, T. D’Silva, N. Kohl, and P. Stone. Autonomous Learning of Stable Quadruped Locomotion. In *RoboCup 2006: Robot Soccer World Cup X*, LNAI. Springer-Verlag, 2007. to appear.
- [7] Y. Sano and H. Kita. Optimization of Noisy Fitness Functions by means of Genetic Algorithms using History of Search with Test of Estimation. In *Proceedings of the Congress on Evolutionary Computation (CEC2002)*, pp. 360–365, 2002.
- [8] J. C. Zagal and J. R. del Solar. Learning to Kick the Ball Using Back to Reality. In *RoboCup 2004: Robot Soccer World Cup VIII*, Vol. 3276 of LNAI, pp. 335–347. Springer-Verlag, 2005.
- [9] M. Zaratti, M. Fratarcangeli, and L. Iocchi. A 3D Simulator of Multiple Legged Robots based on US-ARSim. In *RoboCup 2006: Robot Soccer World Cup X*, LNAI. Springer-Verlag, 2007. to appear.

Table 1: シュートモーション学習の実験結果．actual trial は間引き率，error rate は失敗率を表す．失敗率は，間引かれた候補の中で，学習の性能を落とす原因となった候補の率を表す．

	actual trial		error rate	
	MG	GD	MG	GD
GA	37.6%	49%	0.56%	7.69%
SA	66.2%	76.2%	0.75%	7.00%
HC	7.2%	57.4%	0.00%	2.88%
PG	10.6%	47.6%	0.71%	3.54%

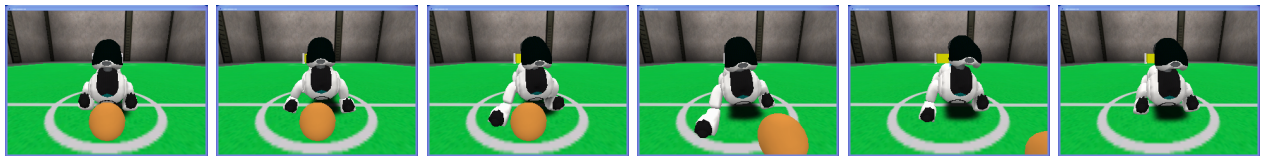


(a) 最大傾斜法



(b) 差分収集法

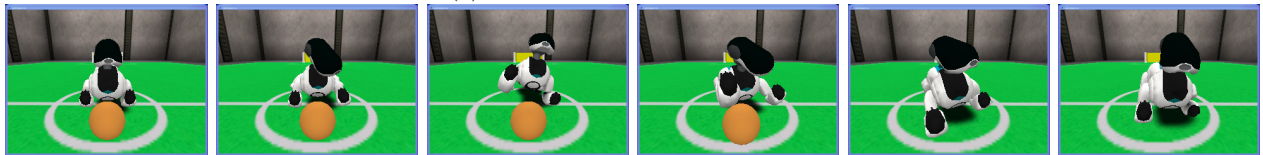
Figure 2: 疑似焼き鈍し法における最大傾斜法と差分収集法の実験結果



(a) 初期モーション



(b) 学習後：体全体を使ったモーション



(c) 学習後：体重を使ったモーション

Figure 3: 初期モーションと学習後のモーション