GAにより探索空間の動的生成を行うQ学習による 実多自由度ロボットの制御

—階層構造の拡張と蛇型ロボットへの適用—

伊藤 一之^{*1} 松野 文俊^{*2*3}

Control of Real Robot Using QDSEGA

—Extension of Layered Structure and Application to Snake-like Robot—

Kazuyuki Ito^{*1} and Fumitoshi Matsuno^{*2*3}

Reinforcement learning is very effective for robot learning. Because it does not need priori knowledge and has higher capability of reactive and adaptive behaviors. In our previous works, we proposed new reinforce learning algorithm: “Q-learning with Dynamic Structuring of Exploration Space Based on Genetic Algorithm (QDSEGA)”. It is designed for complicated systems with large action-state space like a robot with many redundant degrees of freedom. However the application of QDSEGA is restricted to static systems. We extend the layered structure of QDSEGA so that it could be applicable to the dynamical system. A snake-like robot has many redundant degrees of freedom and the dynamics of the system are very important to complete the locomotion task. For this task, application of usual reinforcement learning is difficult. In this paper, we extend layered structure of QDSEGA for applying real robot. We apply it to acquiring of locomotion pattern of the snake-like robot and demonstrate the validity of QDSEGA with the extended layered structure by simulation and experiment.

Key Words: Snake-like Robot, QDSEGA, Reinforcement Learning, Q-learning, Genetic Algorithm, Redundant Robot

1. はじめに

近年、強化学習 [1] がロボットの行動獲得法として注目されている [2]~[6]。強化学習は学習に際し予備知識を必要とせず、高い環境適応性を持つ学習法である。しかし、学習速度や行動・状態空間の大きさなどの問題により、複雑な実ロボットへの適用は困難であり [7]~[10]、多自由度を有するロボットへの適用例は少ない [11]~[13]。

強化学習の代表例である Q 学習 [14] は、最適性に関する理論的枠組みを持ち、小規模な問題に対しては非常に有効であるが、多自由度ロボットなどの大規模な問題においては、行動・状態空間の増加が学習を困難にしまい、事実上適用不可能である。

分類システム [15] は、GA によりルールを生成するため、探索空間が広いタスクに対して学習を試みることは可能であるが、条件部が示す状態の集合と、行動部によって遷移可能な状態の集

合との整合を取ることが難しく、多自由度ロボットなど、探索空間の大きなシステムに対する有効な政策の獲得は困難であると考えられる。また、“don't care”は対象を部分観測としてしまうという問題を持つこと、さらに、汎用的なルールと特殊なルールの共存が困難であることなどが指摘されている [7] [8] [10]。

このような問題をふまえ、我々は、Q-learning with Dynamic Structuring of Exploration Space Based on Genetic Algorithm (QDSEGA) を提案してきた [16]。QDSEGA は、多くの自由度を有する冗長ロボットに対して設計されたものであり、冗長自由度により実現される冗長な運動の中からタスクの達成に必要な動きを自律的に抜き出し、これらを組み合わせることによってタスクを達成することができる。文献 [16] では、50 リンクを有する多自由度マニピュレータの障害物回避問題を例に、大規模な探索空間を持つ対象に対する適用可能性、環境変化に対する頑強性など、その有用性が示されている。また、文献 [17] では、12 本の足を持つ多足ロボットを例に、最適な歩行運動の獲得が可能であることが示されている。しかし、上述の文献では、QDSEGA による階層構造において、十分な能力を持った低層を仮定しており、適用可能なロボットのクラスを制限した場合での有効性の検討にとどまっている。また、これらの文献におけるすべての例題において、学習は、静的なモデルを対象

原稿受付 2002 年 3 月 27 日

^{*1}岡山大学

^{*2}電気通信大学

^{*3}国際レスキューシステム研究機構

^{*1}Okayama University

^{*2}Tokyo Institute of Technology

^{*3}International Rescue System Institute

Fig. 1 Outline of QDSEGA

にシミュレーションにより行われたものであり、実ロボットのようなダイナミクスを有するシステムに対する有効性や適用における妥当性の検討には至っていない。

一方、蛇型ロボットは、地面とボディーの間の摩擦を利用して推進することが知られている。したがって、蛇型ロボットの推進運動は、地面とロボットのダイナミクスがタスク達成のかぎとなるタスクである。また、多自由度を有し、探索空間が膨大となることから、従来の強化学習の適用が困難な対象の一つである。

本論文では、多自由度を有する実ロボットへの適用を視野に入れ、従来のQDSEGAにおける、「十分な能力を持った低層の仮定」をなくし、ダイナミクスや、アクチュエータの能力の限界を考慮した、より現実的な新しい問題領域に対して適用可能な自律的学習手法の実現を目標に、QDSEGAにおける階層構造を拡張する。また、実蛇型ロボットにおける推進運動の獲得問題を対象として、シミュレーションによる学習と、学習により獲得された推進パターンを実ロボットに実装することで、ダイナミクスを有する対象に対する提案手法の有用性、獲得された運動の実ロボットへの適用に対する妥当性を検討する。

2. QDSEGAの概要と階層構造の拡張

本章では、我々が提案したGAにより探索空間の動的生成を行うQ学習(QDSEGA) [16]について概説するとともに、階層構造の機能を拡張する。

2.1 QDSEGAの概要

Fig. 1にQDSEGAにおける学習の流れを示す。QDSEGAでは、ロボットの身体を学習器に取り込むことにより、探索空間を限定する機能を学習器に付加する。これにより、学習器は探索空間の部分集合を抜き出すことが可能となる。次に、抜き出した探索空間の部分集合に対し強化学習を適用する。強化学習で得られた知識を基に、その部分集合を再構成する。これを繰り返すことにより、有効な部分集合と強化学習における政策が獲得される。

具体的には、探索空間を限定する機能については、学習器を階層構造にすることにより実現している。また、強化学習についてはQ学習で実装し、探索空間の更新はGAにより行っている。

2.2 階層構造の拡張と探索空間の動的構成

2.2.1 内部状態と外部状態

本論文では、内部状態と外部状態を次のように定義する。内

Fig. 2 Layered structure for a manipulator

部状態とは、ローカルに直接操作可能な状態の集合であるとし、全状態集合から、内部状態の集合を除いたものを外部状態とする。例えば、マニピュレータの障害物回避を考えると、障害物の位置などが外部状態であり、マニピュレータの各関節角度などが内部状態である。

なお、本論文における操作可能とは、被制御量を任意の応答に制御可能なことを意味しているものではなく、単に、被制御量を変化させるための入力を操作することが可能であることを意味する。

2.2.2 階層構造

QDSEGAでは学習器は2階層からなる階層構造を持つ。Fig. 2にマニピュレータへの適用例を示す。低層のエージェントは、操作対象の各内部状態をローカルに直接操作する。上層のエージェントは、全内部状態の目標値のプランニングを行い、低層のエージェントへ指令する。階層構造が導入可能な条件は、状態を外部状態と内部状態に分離可能であることであり、これは、システムの中にローカルに操作可能な状態が含まれるという条件に相当する。例えば、多自由度ロボットの場合については、アクチュエータを備えた関節の関節角度などが内部状態となり、障害物などの環境が外部状態となる。

本論文では、低層のエージェントについては、「与えられた目標値に向けて対応する被制御量を制御し、制御の成否を上層に返す」という仕様を満たしていれば、どのような手法により実現されてもよいものとし、ステッピングモータなどの位置制御が可能なアクチュエータや、制御工学、ニューラルネットワーク、強化学習などにより実現され、上層の学習に先立って実装されているものとする。また、低層のエージェントを学習によって実装する場合には、事前に学習が行われているものとし、上層との同時学習については扱わないものとする。

従来のQDSEGAでは、静的な環境を考え、低層のエージェント(アクチュエータの能力を含む)は十分な性能を持ち、各状態は各ステップ間に目標値に追従するものと仮定していた。本論文では、階層構造の拡張を行うことで、この仮定を不要なものとし、より現実的な問題の取り扱いが可能となるよう、適用可能な問題領域の拡大を図る。つまり、ダイナミクスを考慮し、低層のエージェントが達成不可能な状態を含む環境を考え、このような環境においても学習を継続し、実際に実現可能な行動と状態のみを用いてタスクを実現する振る舞いの獲得が可能となるように階層構造を拡張する。ここで、低層のエージェント

Fig. 3 Extraction of closed-subset

が目標値を達成不可能な場合とは、アクチュエータの能力の限界から物理的に実現不可能な場合や、他のエージェントの制御による干渉が強い場合である。例えば、前者は、重力や、環境との摩擦、加速度などによりアクチュエータの能力を超えたトルクが必要とされる場合であり、後者は、低層のエージェントの制御系の能力の限界を超えた外乱が他のエージェントからの干渉として与えられる場合である。

2.2.3 探索空間の動的構成

本項では、低層のエージェントが目標値に追従可能な場合について考え、探索空間の動的構成法について述べる。低層のエージェントが目標値に追従不可能な場合については、次項の階層構造の拡張で述べる。

Fig. 3 に探索空間の限定と部分空間の抽出の概要を示す。学習器の階層構造から、低層のエージェントの働きにより、システム全体の内部状態は、上層のエージェントの指令した目標値と等しくなる。ここで、上層のエージェントが目標値を出力することを行動 (action) と考えると、システムが遷移可能な内部状態の集合は、その行動が示す内部状態の集合と等しくなる。したがって、行動空間を限定することで (Fig. 3 STEP1) 内部状態空間も限定され (Fig. 3 STEP2)、結果として探索空間が限定される。ここで我々は、行動に着目し、行動を動的に生成することを考える。この場合、状態空間を構成する際に、すべての内部状態を必要とせず、初期状態と、動的に生成された行動に対応する内部状態のみで構成すればよいこととなる。したがって、行動の動的生成とともに、状態空間も限定され、閉じた探索空間を抜き出すことができる (Fig. 3 STEP3)。

2.2.4 階層構造の拡張

本項では、提案手法の階層構造を拡張する。まず、上層のエージェントと低層のエージェントとの情報伝達を双方向とする。従来の QDSEGA では、上層のエージェントから低層のエージェントへの一方的な情報伝達により低層のエージェントの目標値が与えられていた。本論文では、これに加え、低層のエージェントから上層のエージェントに対し、低層での制御結果を返す通信を付加する。低層のエージェントは、上層から目標値が与えられると、その目標値に向けて対応する内部状態の制御を行い、一定時間が経過した段階で制御を終了し、制御結果の成否を上層へ返す。これにより、双方向の通信となり、上層のエージェントは、低層へ指令した目標値の妥当性を判断することが可能となる。

Fig. 4 Learning algorithm of upper agent

次に、上層のエージェントの学習において、低層から送られた制御結果を報酬 (ペナルティ) として設定する。特に、低層のエージェントのうち一つでも目標値に追従できない場合には、これをペナルティとし、試行を打ち切り、初期状態から再び次の試行を行うこととする。これにより、ある状態において低層のエージェントが達成不可能な目標値を含む行動は、上層の学習により、その状態において使用されることはなくなり、各状態において、低層のエージェントが実現可能な行動のみを用いた政策が獲得されることとなる。

次に、探索空間の動的構成について考える。低層のエージェントが目標値に追従可能な場合には、拡張以前と同様に 2.2.3 項で述べたように、探索空間の動的構成が可能である。次に、低層のエージェントのどれか一つでも目標値に追従しない場合について考える。この場合には、追従が不可能であった低層のエージェントから、上層のエージェントに対しペナルティが返され、初期状態から再び次の試行が行われる。したがって、上述の動的に生成された状態空間の中で学習を行うことが可能となり、階層構造の拡張により、低層のエージェントが達成不可能な状態を含む問題に対しても適用が可能となっていることが分かる。

行動の動的生成法および学習アルゴリズムについては次節で述べる。本論文では以降、すべての行動・状態によって構成される空間を全探索空間と呼び、動的に構成された行動をもとに動的に構成された行動・状態空間を単に探索空間と呼ぶこととする。

2.3 上層における学習アルゴリズムの概要

本節では、上層での学習アルゴリズムについて説明する。Fig. 4 にフローチャートを示す。上層の学習アルゴリズムは、GA による“行動 (action) 生成ダイナミクス”と Q 学習による“振る舞い (behavior) 獲得ダイナミクス”からなる。各行動 (action) は、遺伝子としてコーディングされ GA により生成される。まず初期個体がランダムに生成され、生成された個体の表現型である行動とその行動によって遷移可能な状態を要素とする Q-table が生成される。この Q-table を用いて Q 学習が行われ、Q-table が強化される。Q 学習が十分行われた段階で Q 学習を終了し、強化された Q 値をもとに各行動 (個体) の適合度が計算される。計算された適合度をもとに選択が行われ、選

ばれた親に対し、交叉・突然変異が行われ、新たな個体集合が生成される。以上を繰り返すことにより、有効な行動の集合とその行動を用いた政策が獲得される。

ここで、用語を次のように定義しておく。Fig. 4におけるstartからendまでを、1学習と呼び、学習1回につき、一つのタスクを達成する行動が獲得される。GAにおける繁殖回数を世代と呼び、初期集団を第1世代とする。Q学習において、初期状態からゴールまたはペナルティーを得て再び初期状態に戻るまでを1試行と呼ぶ。1試行において、Q値が更新された回数をステップと呼ぶ。また、初期状態からゴールに至るまでに遷移した状態とその状態で使われた行動の一連を系列と呼ぶ。以下に上層における学習則の詳細を述べる。

2.4 遺伝子のコーディングと個体の生成

QDSEGAで扱うGAでは、各個体は、振る舞い獲得ダイナミクス（Q学習）における実行可能な行動を表し、染色体は各行動と対応するようにコーディングされている。

QDSEGAでは、行動の全集合のなかから、GAにより部分集合を構成し、その部分集合のなかでQ学習を行う。Q学習によって部分集合の各行動が評価され、GAによって部分集合が再構築される。GAにおける個体数は、Q学習における行動の最大数に一致する。

2.5 個体から行動への変換とQ-tableの構成

各個体の表現型が行動である。しかし、各個体すべての表現型をそのまま行動としてしまうと、同一の表現型を持つ個体が複数存在した場合、Q-tableが冗長となってしまう効率が悪い。QDSEGAでは、冗長性を排除し、同一の表現型を持つ個体は一つとみなして行動空間を構成する。GAにおける繁殖を繰り返すことにより同一の表現型を持った個体が増加し、Q-tableの構成においてこれらは同一とみなされるので、その結果、行動数が減少し、世代を重ねるごとに学習効率が改善される。したがって、QDSEGAでは、設計者は、タスクの複雑さに対し十分に大きな数の行動数を個体数というかたちで与えるだけでよく、実際に必要な行動数は学習によって自動的に得られる。

また、QDSEGAでは状態空間は次のように構成される。外部状態については、環境がマルコフ決定過程として観測されるよう適切に構成されるものとする。内部状態は、低層のエージェントの働きにより、行動が示す目標状態に遷移する。したがって、内部状態は、GAにより作られた行動に対応する状態のみで構成すればよく、行動の動的構成に伴って動的に構成されるものとする。したがって、探索空間のうち内部状態と行動からなる空間は、GAにより動的に生成されることとなり、世代を重ねるにつれて、タスク遂行に必要な大きさまで減少していく。

2.6 Q学習

振る舞い獲得ダイナミクスでは、現在提案されているQ学習に関する研究成果の多くが適用可能である。本論文では、簡単のため、従来の典型的なQ学習を用いる。以下に更新式を示す。

$$Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha\{r(s, a) + \gamma \max_{a'} Q(s', a')\} \quad (1)$$

s: 状態, a: 行動, s': aによってsから遷移した状態, r(s, a):

状態sで行動aを実行したときに得られる報酬, α : 学習率, γ : 割引率

なお、Q値の更新は、低層のエージェントの運動がすべて終了した段階で行うものとする。

2.7 適合度の算出

強化されたQ値をもとに、各個体の適合度を算出する。適合度の算出は、Q値による適合度、使用頻度による適合度の二つの適合度からなる。以下に詳細を記す。

2.7.1 Q値による適合度の算出

Q値による適合度は、Q値の規格化とQ値による適合度の算出の2手順から計算される。

<Q値の規格化>

規格化は以下の手順で行う。まず、各状態の価値の最大値 V_{max} 、最小値 V_{min} を次のように定義する。

$$V_{max}(s) = \max_{a'} (Q(s, a'))$$

$$V_{min}(s) = \min_{a'} (Q(s, a'))$$

次に以下の式により、規格化されたQ値である Q' を求める。

$$\text{if } Q(s, a) \geq 0 \text{ then} \\ Q'(s, a) = \frac{1 - p}{V_{max}(s)} Q(s, a) + p \quad (2)$$

$$\text{else } (Q(s, a) < 0) \\ Q'(s, a) = -\frac{p}{V_{min}(s)} Q(s, a) + p \quad (3)$$

ここで、 p は報酬とペナルティーの比重を決める定数である。

この規格化により、次の二つのことが行われている。第一に、状態空間についての規格化である。Q学習では、報酬に近い状態ほど状態の価値が高く評価される。Q値をそのまま行動の価値とした場合、報酬に近い状態で実行された行動が高く評価されてしまい、その状態に至るまでの行動は淘汰されてしまう可能性があり、タスクの獲得が困難となる。このため、状態についての規格化が必要となる。

第二に、ペナルティーについての扱いである。Q学習では、負の値を持つペナルティーを設定することが可能であり、Q値が負の値を持つ場合がある。一方、GAでは適合度は正の値である必要がある。QDSEGAでは、報酬とペナルティーの比重を決める定数 $p(0 \leq p \leq 1)$ を導入し、正のQ値は式(2)のように、規格化されたQ値の1から p の間に、負のQ値は式(3)のように p から0の間に変換している。

<Q値による適合度の算出>

規格化されたQ値から適合度を算出する。まず、規格化されたQ値 ($Q'(s, a)$) を行動を固定し、状態に対してソートし、これを Q'_s とする。例えば、行動 a_i について表すと、 $Q'_s(1, a_i)$ は、 a_i における $Q'(s, a_i)$ の最大値であり、 $Q'_s(N_s, a_i)$ は最小値である。ここで、 N_s とは各世代のQ学習における状態の総数である。行動 a_i を表現型とする遺伝子の適合度は以下の式により与えるものとする。ここで、 a_i はQ-tableを構成する行動集合の任意の要素である。

$$fit_Q(a_i) = \sum_{j=1}^{N_s} \left(w_j \frac{\sum_{k=1}^j Q'_s(k, a_i)}{j} \right) \quad (4)$$

$w_j (j = 1, \dots, N_s)$ は重み係数であり, w_j の値により, 一般的な行動と特殊な行動の価値の重みを決めることができる. 例えば, 第一項は規格化された Q 値の最大値を示しており, これは, 行動 a_i が最も高く評価される状態での a_i の価値を示している. したがって, w_1 を大きくすることにより, 特殊な状態で有効な特殊な行動が高く評価される. 一方, 第 N_s 項は平均値を示しており, これは, 様々な状態においての a_i の価値の平均を意味し, w_{N_s} を大きくすることにより, 様々な状態で有効な一般的な行動が高く評価される. また, 複数の重みを同時に設定することで, 汎用的な行動と特殊な行動の共存をはじめ, 異なる性質を持った複数の行動を同時に評価することが可能となる.

2.7.2 使用頻度による適合度の算出

使用頻度による適合度を導入することにより, 有効な系列の維持を考える. 2.7.1 項で述べた Q-table による適合度の算出では, 各状態に対して規格化された Q 値を用いることで, 各状態における行動の価値を評価し, その和を適合度としている. したがって, 系列における行動の重要性は評価されていない. ここで, 使用頻度の適合度を導入すると, 系列の核となる行動, つまり, 複数の系列で使われる行動は他の行動に比べ使用頻度が高く, 適合度が高く評価され, 系列の破壊を防ぐことができる.

QDSEGA では, 使用頻度による適合度を以下のように定義する.

$$fit_u(a_i) = \frac{N_u(a_i)}{\sum_{j=1}^{N_a} N_u(a_j)} \quad (5)$$

ここで, N_a はある世代での行動の総数であり, $N_u(a_i)$ は, その世代での振る舞い獲得ダイナミクス (Q 学習) において行動 a_i が使われた試行回数である. 式 (5) は, 行動獲得ダイナミクスにおいて実行された全試行回数に対してその行動が少なくとも 1 回は使用された試行の回数の割合, つまり, 使用頻度を表している.

ここで, 式 (5) は振る舞い獲得ダイナミクスが終了した後に一括して計算され, 行動生成ダイナミクスにおいてのみ使われるため, 振る舞い獲得ダイナミクスに影響を与えないという点が重要である. Q 学習の性質が保たれ, 前回の振る舞い獲得ダイナミクスにおける使用頻度とは無関係に, 現在の行動集合のなかで最適な行動に向かって Q 学習が行われ, 新たな系列が作られる.

2.7.3 適合度の算出

2.7.1 項および 2.7.2 項で導入した適合度の和をとって各遺伝子の適合度を以下の式で与える.

$$fit(a_i) = fit_Q(a_i) + k_f \cdot fit_u(a_i) \quad (6)$$

ここで, $k_f (k_f \geq 0)$ は, fit_Q と fit_u の影響力の割合を決めるための定数であり, k_f を大きな値に設定することで, 系列の保存が重視された適合度となり, 逆に小さく設定することで, 各状態での行動の価値が重視される適合度となる.

なお, w_j および, k_f の設定が学習結果に与える影響については, 文献 [16] においてシミュレーションにより検討されている. 詳細は, こちらを参照されたい.

Fig. 5 Task of snake-like robot

2.8 選択・繁殖

計算された適合度をもとに, 選択・繁殖を行う. これらは, 現在提案されている様々な手法が適用可能であり, タスクや, コーディングされた遺伝子をもとに, 適切に選択する必要がある. 本論文では, 従来のシンプル GA を基本とする. 詳しい実装法についてはシミュレーションによる学習の章で述べる.

3. 蛇型ロボットによる推進運動の獲得

蛇型ロボットにおける推進運動の獲得を例に, 提案手法をダイナミクスを有するシステムである実ロボットへ適用した場合の有効性, 妥当性について検討する. また, 本タスクの設定は, 文献 [17] における多足ロボットによる歩行運動の獲得と同じであり, 文献 [17] の結果と本論文における結果とに基づき, 異なる形態のロボットに同一のタスクを課した場合の学習器の汎用性についても検討する. ここで, ダイナミクスを有するとは, ダイナミクスを通じて低層同士が干渉する場合や, タスク自体がロボットや環境のダイナミクスを考慮しないと実現できないようなものを意味している.

最初に, ダイナミクスを考慮したモデルを用いシミュレーションによる学習を行い, ダイナミクスを有するシステムに適用した場合の有効性を検討する.

次に, シミュレーションにより獲得された行動を実機に実装することで, 提案手法により獲得された運動は実ロボットで実現可能な妥当なものであるかを検討する.

本タスクは, 多自由度を有するロボットを対象にしており, 探索空間が膨大であるという点において, 従来の Q 学習では実現が事実上不可能であり, また, ダイナミクスや低層の制御限界を考慮している点において, 従来の QDSEGA の適用が不可能なタスクである.

3.1 タスク

Fig. 5 に示すように, 蛇型ロボットの初期状態は, x 軸上に直線状に置かれた状態であり, x 軸上に置かれた目標位置への接近をタスクとする. ただし, 目標は蛇ロボットの全長に比べて十分遠くにあるものとし, 本実験では, 目標位置の x 座標を 100 とする.

3.2 蛇型ロボット

水平方向に運動可能な 5 リンクを有する蛇型ロボットを考える. Fig. 6 に実験に用いたロボットを示す. 各リンクの大きさは, 80 [mm] × 80 [mm] × 13 [mm] であり, 重量は 600 [g] である. 各関節にはステッピングモータが搭載されている. また, 各リンクには, リンクと平行する方向に回転するタイヤ (受動

Fig. 6 Snake-like robot

Fig. 7 Simulation model of snake-like robot

車輪)が取り付けられており、リンクと水平な方向への摩擦は小さく、リンクに対し垂直な方向には摩擦が大きくなるような構成となっている。

3.3 蛇型ロボットのモデル

蛇型ロボットのモデルには、Saitoらによって、提案されている動力学モデル [18] を用いる。このモデルは、蛇型ロボットと地面との摩擦を考慮したモデルであり、地面との摩擦を利用した推進運動をシミュレートすることができる。Fig. 7 および式 (7) にモデルの概要を示す。

$$\begin{bmatrix} \mathcal{J} & 0 \\ 0 & mI \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{w} \end{bmatrix} + \begin{bmatrix} C\dot{\theta}^2 \\ 0 \end{bmatrix} + \begin{bmatrix} \mathcal{R} & S \\ S' & Q \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} D' \\ 0 \end{bmatrix} u \quad (7)$$

式 (7) において、 w は、ロボットの重心の位置ベクトルであり、 θ および u は、それぞれ、Fig. 7 に示す、関節角度ベクトルと入力トルクベクトルであり、 $\dot{\theta}^2$ は、 $\dot{\theta}_i^2$ を要素とするベクトルである。なお、シミュレーションにおいては関節角は絶対角で定義している。式 (7) を解くことにより、各関節へ入力したトルクにより生成されるロボットの運動を求めることができる。詳しくは、文献 [18] を参照されたい。本シミュレーションでは、実機に合わせ、リンクと水平な方向への動摩擦係数を 0.1 に、リンクに対し垂直な方向への動摩擦係数を 10 に設定した。

次に、ロボットのアクチュエータであり、また同時に、低層のエージェントとしての機能を果たす、ステッピングモータのモデルについて説明する。一般的にステッピングモータの入力はパルス状の信号であり、ステッピングモータの能力の範囲内であれば、入力されたパルスの数に角度が比例し、角速度がパル

スの周波数に比例する運動が行われる。したがって、必要なトルクはアクチュエータによって自動的に生成されることとなり、この機構をモデル化する必要がある。本論文のシミュレーションでは、角速度と角加速度を入力として与えることにより、各関節に必要なトルクを示す関係式を求め、それを式 (7) のトルクに代入することにより運動の計算を行う。これは、目標とする角加速度および角速度を達成するために必要なトルクを計算し、そのトルクを発生させることで、目標の角加速度、角速度を達成する枠組みとなり、ステッピングモータの性質をよく表していると考えられる。入力する角加速度の大きさは、ステッピングモータの能力をもとに一定値に設定する。入力される角速度は、低層のエージェントにより計算され、各ステップにおいて、一定の角速度で運動するものとする。ただし、角速度が目標値に至るまでの間は、上述の一定の角加速度で加速するものとする。角速度の設定方法については、3.6 節で述べる。また、アクチュエータの能力の上限を角速度を用いて設定し、上限値を超える大きさの角速度を入力した場合には、ステッピングモータは追従できないものとして、ステッピングモータは動作しないものと設定する。

3.4 行動生成ダイナミクス (GA) の構成

QDSEGA における GA 部を次のように構成する。最初にコーディングについて述べる。

一つの染色体は関節数と同数の遺伝子を持つ。遺伝子は、関節の角度を表し、遺伝子と関節は、1 対 1 に対応している。遺伝子は 0 から 8 までの 9 種類をとるものとする。各関節の可動範囲を ± 20 [deg] とし、遺伝子の 0 は -20 [deg] を表し、以降、5 [deg] ずつ増加し、8 は $+20$ [deg] を表すものとする。

初期集団はランダムに生成され、個体数を 100 とする。選択はエリート戦略を用い、上位 20% の種類の個体を残し、残りをルーレット選択を用いて初期集団と同数の個体を選択すると設定した。

交叉は一様交叉を用る。交叉確率 $P_c = 0.5$ の確率で、個体集合から親を選び、その中からランダムにペアを作り交叉を行う。交叉により作られた子個体の全ビットに対し、突然変異確率 $P_m = 0.02$ で変異させる。交叉・突然変異によって作られた子個体は親個体との交換は行わず、元の個体集合に加えるものとする。

3.5 振る舞い獲得ダイナミクス (Q 学習) の構成

内部状態は、蛇型ロボットの各関節の相対角度とする。状態空間は、動的に生成された行動によって遷移可能な内部状態と初期状態からなり、動的に構成される。報酬は、時刻 t [step] におけるロボットの重心から目標位置までの距離を $d(t)$ とし、 $\text{reward} = 100 \times (d(t-1) - d(t))$ で与えるものとする。選択にはボルツマン分布による確率的選択を用い、温度定数を $T = 150.0 \times \exp(-0.002 \times \text{trial}) + 1$ とする。ここで、 trial は、試行回数を示す。学習パラメータは、学習係数 α を 0.5、割引率 γ を 0.9 とし、10,000 回の試行を行う。1 試行の最大ステップ数を 40 とし、40 ステップとなった時点で、初期状態に戻し、次の試行を行う。

3.6 低層のエージェントの構成

低層のエージェントは、ステッピングモータ自身による角速

Fig. 8 Transitions of gain and number of actions

Fig. 9 Acquired patterns of each joint

度制御と、とステッピングモータへの入力となる角速度を求める機構とから構成される。

低層のエージェントは、上層から目標値が与えられた場合、対応する関節の現時刻 t_i における相対角度 $\phi(t_i)$ と上層から与えられた目標値 $\phi_d(t_{i+1})$ から、式 (8) に従って目標角速度 $\dot{\phi}_d(t_{i+1})$ を計算する。次に、得られた一定の角速度で $t_{i+1} - t_i$ 時間の間ステッピングモータを回転させるように、ステッピングモータに入力を加えることとする。計算された角速度がステッピングモータの性能の限界を超える場合には、ステッピングモータは動作しないため、これにより制御の失敗を検出し、上層に対しペナルティーを返すこととする。

$$\dot{\phi}_d(t_{i+1}) = \frac{\phi(t_i) - \phi_d(t_{i+1})}{t_{i+1} - t_i} \quad (8)$$

3.7 シミュレーションによる学習

5リンクの蛇型ロボットを対象として、シミュレーションにより学習を行った。終了条件を世代数で設定し、30世代とした。その他のパラメータを以下のように設定した。

$p = 0.01$, $w_1 = 0.5$, $w_i = 0$ ($i = 2, \dots, N-1$), $w_N = 0.5$, $k_f = 200$.

Fig. 8 にゲインと行動数の推移を、Fig. 9 に獲得された推進

Fig. 10 Acquired locomotion

パターンを示す。ここで、ゲインとは、各世代における Q 学習の最終 100 試行間に得られた報酬の 1 ステップ当たりの大きさである。Fig. 9 における最上段の図は、進行方向一番目のジョイントに対応し、以下順に後方のジョイントに対応する。Fig. 9 の○印は、上層の学習により獲得された各時刻の目標値であり、実線は低層により実現される運動を示す。次に、6 ステップごとの運動の様子を Fig. 10 に示す。各図の上の数字は、遷移順序を示す。

Fig. 8 から分かるように、初期世代においてゲインが正となり、移動パターンの獲得に成功していることが分かる。これは、蛇が推進可能な冗長なパターンが複数存在し、ランダムに選んだ初期集団の中に推進可能なパターンを形成可能な行動の集合が含まれていたためであると考えられる。一方、行動数は世代を重ねるごとに減少しており、ゲインはほぼ一定に保たれていることから、推進運動に関与しない行動がのぞかれ、必要な行動の集合が抜き出されていることが分かる。QDSEGA では、内部状態も行動に伴って生成されるため、内部状態の数は行動数と等しくなり、最終世代における探索空間の部分空間の大きさは、 20×20 程度である。一方、従来の Q 学習を用いた場合には、状態数および行動数はそれぞれ、 9^4 となり、探索空間 (Q-table の大きさ) は、 9^8 となることから、実装に必要な記憶容量および、学習に必要な試行回数の面から、従来の Q 学習では困難であり、QDSEGA では、この点が改善されている。

Fig. 11 Transient responses in experiment

ことが分かる。次に、獲得された運動を見てみると、Fig. 9およびFig. 10から、周期的なうねり運動により推進していることが分かる。また、本論文で用いたタスクは、文献[17]において多足ロボットに対して用いたものと同じ設定であるが、獲得された運動は、多足ロボットの場合とは異なる。これは、QDSEGAでは、タスクが同一であっても、ロボットのボディーの形態に合わせた運動が自律的に獲得されることを示している。

蛇型ロボットのうねり運動の獲得については、我々は、Q-learning with propagation of motion (QPM) [11]により、自律的な獲得に成功している。QPMでは、先頭の関節の運動が後方の関節に伝播していくという拘束条件を与えることにより探索空間を限定し、Q学習の適用を可能としている。これは、生物の蛇の運動をもとに設計者が与えたものであり、設計者の知識による探索空間の限定である。一方、QDSEGAでは、設計者の知識を用いず、単に、行動となる各関節の相対角度の目標値を限定することで探索空間を限定しており、限定された探索空間は学習によって改善され、より自律的、適応的な手法であるといえる。獲得された運動について考えると、両者による探索空間の限定法は異なるものの、獲得された運動は両者とも、うねり運動であり、また、これは事前にプログラムされていた運動ではなく、環境との相互作用と学習によって獲得されたものであり、この意味において創発した運動であると考えられる。また、生物の蛇が行う推進運動もうねり運動であることを考えると、うねり運動は、ボディーの形状に内在した運動であると考えられ、これが、一切の事前知識を用いず、自律的に獲得されたことは、非常に興味深い。

4. 実機への実装

シミュレーションにより獲得された推進パターンを実機に実装し運動を行わせることで、提案手法により獲得された推進運動が実ロボットにより実現可能な妥当なものであるか検討する。

最初に実装法について説明する。実機の運動は、シミュレーションにより得られた周期的な推進パターンにおける各関節の角度の時系列を、実機の目標値とすることで実現した。本タスクでは、状態空間は、内部状態のみであるから、シミュレーショ

Fig. 12 Locomotion of snake-like robot

ンにおける低層のモデルが十分に実機と近い場合には、実際の状態遷移とシミュレーションによる状態遷移が一致する。したがって、推進パターンを直接実装する方法と、シミュレーションによる学習によって得られたQ-tableを実機に実装し、Q-tableに基づいて各状態ごとに行動を選択することで実装する方法とが等価となる。また、実装において必要とされる計算時間の面では、仮にQ-tableを実装した場合を考えると、Q-tableから行動を選ぶために要する計算時間は、ロボットの制御のサンプリングタイム2[s]に比べて十分に小さく、計算に必要な時間の面からも、今回の実装方法は妥当であると考えられる。

最初に、低層の働きについて検討する。Fig. 11に、実機の各関節の時間応答の様子を示す。Fig. 11の○印は、学習によって獲得された上層の示す目標値であり、点線が低層により生成された各関節の目標軌道である。実線は、各関節の実時間応答である。Fig. 11において、10[s]が制御を開始した時刻である。Fig. 11から分かるように、学習により得られた運動パターンである目標値に対し、若干の遅れがあるものの、よく追従していることが分かる。遅れ時間は、目標値が与えられてから、各関節をローカルに制御するのに要した時間であると考えられる。この目標値に対し追従しているという結果は、低層のエージェントが有効に機能していることを示している。低層のエージェントであるステッピングモータが有効に働いた理由は、シミュレーションにおいて、低層のエージェントが実現不可能な状態をペナルティーとすることにより、ステッピングモータが実現可能な目標値のみを用いて推進パターンを形成したためであると考えられる。また、この結果は、ステッピングモータのモデルの精度が十分であったことも示している。

次に、実現された運動について検討する。Fig. 12に実現された運動の様子を示す。Fig. 12から分かるように、うねり運動

による推進が行われ、目標のタスクを達成していることが分かる。50秒間に推進した距離は、ばらつきがあるものの、おおむね180[cm]前後であり、シミュレーションによる値との誤差は、大きいもので50%程度であった。推進距離のばらつきや、シミュレーションとの誤差は、床の状態にばらつきがあること、また、実際の摩擦係数とシミュレーションに用いた摩擦係数との誤差のためであると考えられる。このほか、シミュレーションにより獲得された異なる5パターンについて同様に実験を行った結果、すべてのパターンについて推進運動が可能であった。これは、提案手法が架空の理想化された対象のみでなく、ダイナミクスを有するような実際に存在する対象に対しても有効であることを示している。

5. 結 論

本論文では、多自由度を有する実ロボットへの適用を視野に入れ、従来のQDSEGAにおける仮定を緩和し、ダイナミクスや、アクチュエータの能力の限界を考慮した、新しい問題領域に対して適用可能な自律的学習手法の実現を目標に、QDSEGAにおける階層構造を拡張した。また、多自由度を有し、ダイナミクスがタスクの達成のカギとなる蛇型ロボットによる推進運動の獲得問題に適用することで、提案手法のダイナミクスを考慮したタスクへの適用の可能性、実ロボットに対する実装の妥当性を検討した。階層構造の拡張により、低層が実現不可能な目標値を含む対象に対してのQDSEGAの適用が可能となり、実際にステッピングモータの性能を考慮したモデルを用いることにより、ステッピングモータの実現可能な目標値のみを用いた推進運動が獲得されることを確認した。また、これは、ダイナミクスを有する対象に対しても提案手法が有効に働いていることを示している。実ロボットへの適用については、シミュレーションによる学習により獲得された推進パターンを実ロボットに実装し、実際に推進運動が行われることを確認した。したがって、提案手法は、ダイナミクスを有する実ロボットに対しても有効であると考えられる。今後の課題は、学習の高速化とロボットによる実学習にむけての拡張である。

参 考 文 献

- [1] R.S. Sutton, Reinforcement Learning: An Introduction. The MIT Press, 1988.

- [2] 木村元, 小林重信, “Actorに適正度の履歴を用いた actor-critic アルゴリズム”, 人工知能学会誌, vol.15, no.2, pp.267-275, 2000.
- [3] K. Doya, H. Kimura, and M. Kawato, “Neural mechanisms of learning and control,” IEEE Control Systems Magazine, vol.21, no.4, pp.42-44, 2001.
- [4] 高橋泰岳, 浅田稔, “実ロボットによる行動学習のための状態空間の漸次的構成”, 日本ロボット学会誌, vol.17, no.1, pp.118-123, 1999.
- [5] 中村陽一郎, 黒山和宏, 山田和明, 大倉和博, 上田完次, “Instance-based classifier generator による自律移動ロボットの行動獲得”, 日本ロボット学会誌, vol.17, no.3, pp.371-379, 1999.
- [6] K. Yamada, K. Ohkura, M. Svinin and K. Ueda, “Adaptive segmentation of the state space based on bayesian discrimination in reinforcement learning,” In Proc. of the 6th Int. Symp. on Artificial life and Robotics, pp.68-171, 2001.
- [7] 林朗, 末松伸朗, “POMDP 研究に基づいたハイブリッド分類システム”, 人工知能学会誌, vol.14, no.3, pp.538-546, 1999.
- [8] S.W. Wilson and D.E. Goldberg, “A critical review of classifier systems,” In Proc. of the Third Int. Conf. on Genetic Algorithms, pp.244-255, 1989.
- [9] T. Jakkola, S.P. Singh and M.I. Jordan, “Reinforcement learning algorithm for partially observable markov decision problems,” In Advances of Neural Information Processing Systems 7, pp.345-352, 1994.
- [10] T.H. Westerdales, “Classifier systems—no wonder they don't work,” In Proc. of the Second Annual Genetic Programming Conf., pp.529-537, 1997.
- [11] K. Ito and F. Matsuno, “Application of reinforcement learning to hyper-redundant system—acquisition of locomotion pattern of snake like robot—,” In Proc. The Pacific Asian Conference on Intelligent Systems, pp.65-70, 2001.
- [12] M. Svinin, S. Ushio, K. Yamada and K. Ueda, “Emergent systems of motion patterns for locomotion robots,” In Proc. of Int. Workshop on Emergent Synthesis, pp.119-126, 1999.
- [13] S. Ushio, M. Svinin, K. Ueda and S. Hosoe, “An evolutionary approach to decentralized reinforcement learning for walking robots,” In Proc. of the 6th Int. Symp. on Artificial life and Robotics, pp.176-179, 2001.
- [14] C.J.C.H. Watkins and P. Dayan, “Technical note Q-learning,” Machine Learning, vol.8, pp.279-292, 1992.
- [15] J.H. Holland, “Escaping brittleness: the possibilities of general purpose learning algorithms applied to parallel rule-based system,” In Machine Learning II, pp.593-623, 1986.
- [16] 伊藤一之, 松野文俊, “GA により探索空間の動的生成を行う Q 学習”, 人工知能学会誌, vol.16, no.6, pp.510-520, 2001.
- [17] 伊藤一之, 松野文俊, “QDSEGA による多足ロボットの歩行運動の獲得”, 人工知能学会誌, vol.17, no.4, pp.363-372, 2002.
- [18] M. Saito, M. Fukaya, and T. Iwasaki, “Serpentine locomotion with robotic snakes,” IEEE Control Systems Magazine, vol.22, no.1, pp.64-81, 2002.

伊藤一之 (Kazuyuki Ito)

1973年10月4日生。1996年3月法政大学工学部電気工学科卒業。1998年3月同大学大学院工学研究科修士課程システム工学専攻修了。日清紡績株式会社を経て、1999年4月東京工業大学大学院総合理工学研究科博士後期課程知能システム科学専攻入学。2002年3月同課程修了。現在、岡山大学工学部システム工学科助手。主に知能ロボットに関する研究に従事。1997年 The 5th International Ground Robotics Competitionにおいて優勝。人工知能学会正会員。博士(工学)。(日本ロボット学会正会員)

松野文俊 (Fumitoshi Matsuno)

1957年7月26日生。1986年大阪大学大学院基礎工学研究科博士課程修了。大阪大学基礎工学部助手、神戸大学工学部講師、助教授、東京工業大学大学院総合理工学研究科助教授を経て、2003年電気通信大学教授(知能機械工学科)となり、現在に至る。主に、ロボティクス・制御理論・地理情報システムに関する研究に従事。最近は、地理情報システムを基盤としたレスキュー活動支援システムに関する研究を行っている。2002年より(NPO)国際レスキューシステム研究機構川崎ラボリード。1989年度日本ロボット学会研究奨励賞、1993年度システム制御情報学会論文賞、1997年度消防防災科学論文賞(消防庁長官表彰)、RoboCup Japan Open 2000 JSAI賞、2001年度計測自動制御学会論文賞・武田賞などを受賞。IEEE、計測自動制御学会、GIS学会、情報処理学会などの会員。工学博士。(日本ロボット学会正会員)