

# 論文内容の要旨

## 論文題目 Evolutionary Robotics by Means of Layered Genetic Programming

(階層的遺伝的プログラミングを用いた進化型ロボットに関する研究)

氏名：劉宏偉

ロボット制御の立場から見れば、プログラムはロボットの制御に対して、最も一般的、直観的な方法であるが、手でプログラミングするのは極めて困難なことである。故に、進化的な手法が広く応用されてきた。遺伝的プログラミング (GP) は、進化型計算手法の1つ、与えられた問題を解決するため、自動的にコンピュータプログラムを探索する手法である。多くの研究は GP がロボットのプログラム設計に有効性と証明した。

しかし、これに関する研究はほとんどシミュレーションで行われた。筆者はシミュレーションで構想を検証することではなく、実機のロボットを制御して実環境の問題を解決することを注目する。GP を用いた実機ロボットを制御することは困難であり、今まで、有効な手法を報告されていない。

GP を使って実機のロボットの制御プログラムを生成する時に、膨大な数の個体を実機で評価しなければならない、この過程には通常膨大な時間を要する。この問題を回避するため、シミュレーションで進化させ、最終的に得られた個体を実機に移植するアプローチは直観的な考えである。しかしながら、シミュレーションから獲得した結果は実機での有効性に異論があり、普通、シミュレーション上で進化させた個体を実環境に実装しても、適切に行動することが難しい、すなわち、シミュレーションと実環境の間に **gap (reality gap)** がある。

その **gap** を克服する努力とし、多くの研究者らは実機を用いて直接進化を行なわせ、有望な結果を報告した。これらの研究はほとんどニューラル・ネットワークに基づく手法であるが、少数の研究者は GP を使い、実機で制御プログラムを進化させた。代表的な研究として、Banzhaf らは線型 GP (linear-GP) を用いて、実機でロボットの制御プログラムを進化させる実験を報告した。彼らは Khepera ロボットを使い、障害回避と目標追従の行動を進化させ、良好な結果を得た。これらの結果は GP を用いて実機で直接進化の可能性を示した。

しかしながら、上記の研究は簡単なロボットを用いて簡単なタスクを達成できたが、複雑システムに対する拡張性は未解決の問題である。

本論文はこれに対して、階層的遺伝的プログラミングを提案し、実機ロボットを進化させる。

Brooks が提唱しているサブサンプションアーキテクチャ (Subsumption Architecture, SSA) は目標タスクを達成するために、ロボットの行動をセンシングとアクションを持った

基本的な動作(primitive behaviors)に分割し，低レベルから高レベルに至る各々の基本的な動作が確実に実世界で動くようにする，その上で，高レベル低レベルを包摂する手法である。

サブサンプションアーキテクチャ手法の最大の利点は実環境に頑健な行動ができる。しかし，基本的な動作は手で試行錯誤によってデザインされ，複雑なタスクに対し，制御システムの設計は大変難しい問題である。それに対して，筆者は進化的なサブサンプションアーキテクチャ(Evolutionary Subsumption Architecture, ESSA)を提案する(図 1 進化的なサブサンプションアーキテクチャ)。

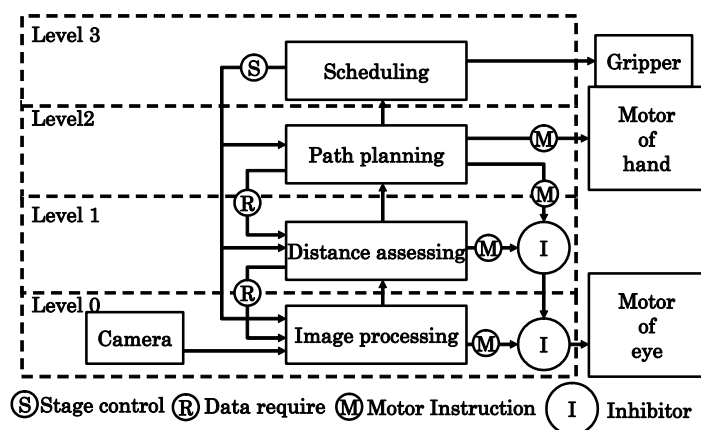


図 1 進化的なサブサンプションアーキテクチャ

この手法はサブサンプションアーキテクチャによって，制御システムを複数のレベルに分け，GP を用いて低レベルから高レベル順，自動的に生成する。GP はサブサンプションアーキテクチャの基本的な動作を生成することによって，デザインの重荷を軽減し，一方，サブサンプションアーキテクチャは制御システムを分けることによって，GP の探索空間が大幅に削減した。

この方法は「手-目協調」という二台ロボットの協調動作問題に検証された。「手-目協調」問題は，二つの移動ロボット，カメラを装備されたロボットは「目」とし，グリッパーを装備されたのは「手」として行動する。目標タスクは「目」が「手」を指揮し，物体を指定したところに搬送することである。

実験はまずシミュレーションで行い，進化させた個体を実機ロボットに実装する。サブサンプションアーキテクチャ手法によって，獲得した結果は十分に頑健であるため，シミュレーションと実環境の gap を克服し，実環境に高い確率で目標タスクを完成することができる。

ヒューマノイドロボットは高自由度の複雑システムであり，移動ロボットより制御が困難である。ヒューマノイドロボットの複雑さを考えると実機を用いて直接進化を行なわせるのは極めて困難，或いは事実上不可能である。

一方，シミュレーションで進化させる方法では，ヒューマノイドロボットのシミュレーシ

オン自身も大変計算負荷が大きく、進化させるのも困難である。ヒューマノイドロボットの複雑性により、シミュレーションと実環境の間の **gap** が一層大きいため、移植することも困難である。さらに、シミュレーション環境が用意されていないヒューマノイドロボットもある。それゆえに、シミュレーションで進化させ、最終的に得られた個体を実機に移植するアプローチはヒューマノイドロボットに対しても困難、もしくは、不可能である。それに対して、筆者は **GP** と **CBR** の統合手法を提案し、自律的な搬送タスクにおいて、提案する手法の有効性を検証する。

提案する手法はロボットの制御システムをハイレベルの行動計画レイヤー(**planning layer**)とローレベルの反射行動レイヤー(**reactive layer**)に分ける。筆者らは“抽象動作要素”(abstract behavior)という新たな概念を提案する。abstract behavior はヒューマノイドロボットの似ている多様な要素動作を代表する抽象的なシンボルである。提案する手法は abstract behavior に基づいて遺伝的プログラミング(**Genetic Programming, GP**)により高度に単純化したシミュレーションの中に行動計画レイヤーを生成する。そうして、事例に基づく推論(**Case-Based Reasoning, CBR**)をオンライン適応手段として用い、実環境で、abstract behavior を具体的なモーター指令に変換する。システムは図 2 **GP** と **CBR** の統合手法に示している。

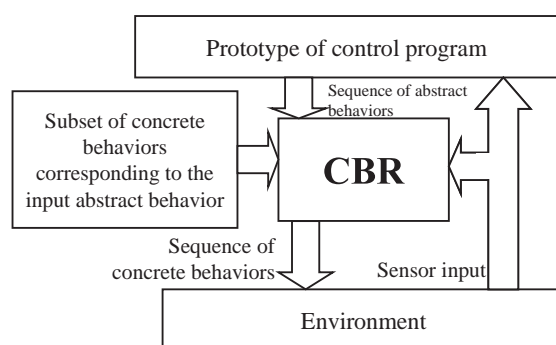


図 2 **GP** と **CBR** の統合手法

実験の結果により、提案する手法は効率的にヒューマノイドロボットの制御システムを生成し、環境に柔軟的に適応できることを示す。さらに、本手法はヒューマノイドロボットの制御だけではなく、進化的な手法を複雑システムへ応用する、一般的なフレームワークを提案する。

**GP** と **CBR** の統合手法に事例に基づく推論をオンライン適応手段として用い、実環境で、abstract behavior を具体的なモーター指令に変換する。しかし、事例ベースの定義は目標も達成に対して大変決定的なことである。しかし、手で行うため、困難で、実機ロボットの性能に大きい影響を与える。この問題を軽減するため、筆者はオンライン進化的な抽象動作要素を提案した(図 3 オンライン進化的な抽象動作要素)。

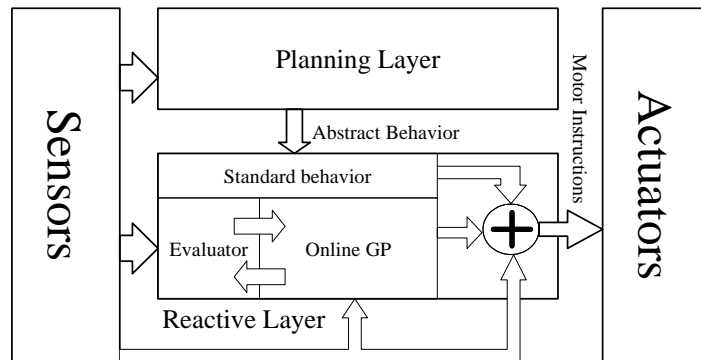


図 3 オンライン進化的な抽象動作要素

この手法において、制御システムを高レベルと低レベルに分け、高レベルは **abstract behavior** に基づいて GP により単純化したシミュレーションの中に生成する。低レベルは図 3 オンライン進化的な抽象動作要素のした部分に示すように、オンライン GP を用いて、実環境の状況に応じてダイナミックに **abstract behavior** を具体的なモーター指令に変換する。

提案する方法はナビゲーションタスクにおいて評価される。ロボットは障害物がある環境において、障害物を回避しながら、目標に接近する。

低レベルには、標準的な動作、オンライン GP システム(**online GP system**)、行動評価システム(**Evaluator system**)とコーディネーターシステム四つ部分で構成される。

オンライン GP システムは現場の状況に基づいて、適切な行動をダイナミックに生成する。普通、実機で **Online GP** を使う時に、進化過程は通常膨大な時間を要するため、実機の進化は不可能になる。提案する手法はオンライン GP システムの評価は実環境を用いらずに、行動評価システムを採用し、進化時間はきわめて削減され、リアルタイムに進化することができる。

行動評価システムはオンライン GP システムから出力行動の効果を評価し、適応度を計算し、オンライン GP システムへフィードバックする。

コーディネーターシステムは標準的な動作とオンライン GP システムの出力を結合するメカニズムである。

実験の結果によって、提案する手法は実機で低レベル動作を生成し、反射行動を行っながら、高レベル制御プログラムの計画的な動作を達成する。

本論文は実機ロボットに対して、階層的遺伝的プログラミング手法を提案し、三つのタスクにおいて検証した。複雑なロボットシステムに対して、制御プログラムを効率的に生成できる。実験結果を通して、提案手法の有効性を示した。