

# 実環境における進化ロボティクスと対話的設計

片上大輔<sup>\*</sup> 山田誠二<sup>\*\*</sup>

<sup>\*</sup> 東京工業大学大学院総合理工学研究科 横浜市緑区長津田町 4259

<sup>\*\*</sup> 国立情報学研究所 東京都千代田区一ツ橋 2-1-2

<sup>\*</sup> Interdisciplinary Graduate School of Science and Engineering Tokyo Institute of Technology, Nagatsuta-cho 4259, Midori-ku, Yokohama, Japan

<sup>\*\*</sup> National Institute of Informatics, 2-1-2 Hitotsubashi, Chioda-ku, Tokyo, Japan

\* E-mail: katagami@ntt.dis.titech.ac.jp

キーワード: ロボット学習 (robot learning), 進化計算 (evolutionary computation), 実環境 (real environment), 進化ロボティクス (evolutionary robotics), 対話型進化計算 (interactive evolutionary computation), 対話型進化ロボティクス (interactive evolutionary robotics)  
JL 002/02/4202-0086 ©2002 SICE

## 1. はじめに

実環境における, 状態 (センサなどの知覚) から行為へのマッピングである行動を扱うロボットの学習や適応において, 知能はロボットのおかれる環境の状態や複雑さ, ロボットの身体性に強く依存するため, ロボットの行動獲得における環境設計の重要性は様々な分野で示唆されてきた。しかしながら, これらを考慮して効率よく妥当な制御器を構築するための方法論は未だ存在しないのが現状である。

このような問題に対して, 強化学習や進化計算を用いてロボットに自律的に制御器を獲得させる研究が行われてきた。これらの手法の一つの目的は, 制御器設計におけるトップダウンな設計手続きをなるべく排除することである。これは同時に身体性や環境との相互作用を制御器の構築に反映させる際に, 設計者による不適切なバイアスを排除することにつながる。

この中でも, 特に進化計算 (Evolutionary Computation; EC) を用いてロボットを設計するアプローチは, 進化ロボティクス (Evolutionary Robotics; ER) と呼ばれている。進化ロボティクスは, 遺伝的アルゴリズム (Genetic Algorithm; GA), 遺伝的プログラミング (Genetic Programming; GP) などの進化計算手法を用いて, ロボットを設計するアプローチの総称であり, その有効性は多くの研究で確認されてきている<sup>1), 2)</sup>。しかしながら, その一方で進化ロボティクスに関するいくつかの問題点や限界についても明らかにされてきた<sup>1), 3)</sup>。

筆者らは, 自律ロボットが人間とのインタラクションを通して, 世界をどのように構築し行動を生成していくかといった過程に興味がある。このようなアプローチは, 前述のバイアスの問題を意識せずに, 人間の設計をシステムに組み込める点に特徴がある。本稿では, ロボットの設計論として, 人間とロボット間のインタラクションの設計, 特に人間からのインタラクションを明示的な教示として扱った対話的設計について議論する。ここでは, 対話的設計のポイントとして, 1) 教示者の負荷の問題, 2) 教示者である人間と学習者であるロボットの認知のずれの問題について言及し, 具体例を交えながら, 対話的設計のあり方を考えていく。

## 2. 進化ロボティクスとその問題

ER では, 行動主体の身体性や環境とのインタラクションダイナミクスを設計者が意識しなくても制御器の設計に反映することができるという優れた特徴を持っている。ER では, ロボットの制御器を設計の対象とする研究が多いが, その設計過程では制御器の詳細な仕様をトップダウンに決定するのではなく, 評価関数 (もしくは適応度) の下で進化過程を通してボトムアップに設計する。このような間接的な設計手法により, 高い創造性を期待しているところにこのアプローチの特徴がある。ER の主な研究としては, Koza は, 行動ベース移動ロボット<sup>6)</sup> の行動学習に GP を適用している<sup>7)</sup>。また, GP で生成されたロボットプログラムの頑健性を検証した研究<sup>8)</sup> や, 環境認識に適した行動を GA を使って学習する研究<sup>9)</sup>, 共進化によりサッカーロボットの行動を学習する研究<sup>10)</sup> など, さまざまな進化ロボティクスの研究が行なわれている。

これらの研究を通して, 以下のような問題点が指摘されている。

1. ギャップ問題 (gap problem): 進化過程は通常膨大な計算時間を必要とするため, シミュレータを用いたオフライン進化を採用することが多い。しかし, 進化は与えられた環境に過度な特化 (over-adapting) をする傾向があるため, シミュレータ上で獲得された最良個体を実機に実装しても, シミュレータ上と同様な行動が困難であるという問題。
2. ブートストラップ問題 (bootstrap problem): 複雑なタスクを実現する制御器を進化的に獲得しようとする時, 初期世代に部分的にでもタスクを達成し, ある程度の評価値を獲得できる個体が発現する確率が極めて低くなるという問題。

特に実環境においては, 学習の収束に時間がかかり, 報酬を得るまでの初期学習の立ち上がりに大きな時間コストがかかる。しかも 1 回の行動に必要な多くの時間は, ロボットの感覚, 行動系の処理時間に費やされるので, 高速に学習するためには, 学習試行数の削減が必要である。また, 実環境を十分な精度でシミュレートすることは困難であるた

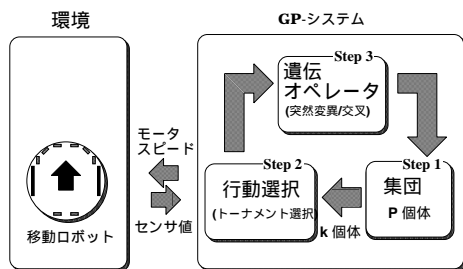


図1 オンライン進化アルゴリズム

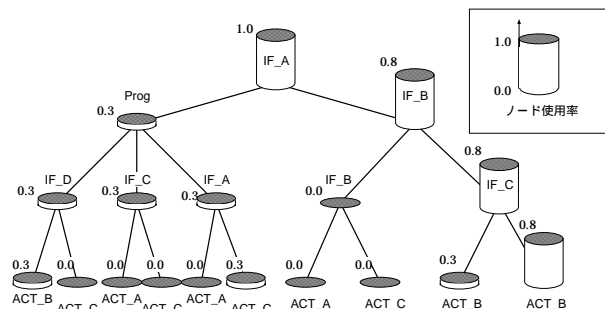


図2 ロボットプログラムと使用率の偏り

め、オンラインで直接進化を行う、オンライン進化<sup>(注1)</sup>が注目されるが、ER で用いている EC では、多数の個体を数百世代にわたり実行させ適応度を評価する必要があり、その高速化が求められる。実際には、その多大な実験コストが原因となって、シミュレーションによりオフライン進化を行うことが多い。

### 3. 実環境での進化ロボティクスの適用

実環境におけるオンライン進化の適用は、前述のギャップ問題を考えなくてもよいという利点があるが、ブートストラップ問題をどう対処するかが大きな鍵となる。そこで、EC 自体の性質と ER における環境とのバイアスに注目してオンライン進化そのものを高速化することにより、実環境での学習に応用した例を紹介する<sup>(12)</sup>。

この研究のポイントは、ER を GP で行なう場合に、GP におけるノードの使用率に環境特有の偏りがあることに注目し、使用頻度に依存した交叉点の設定方法を考えている点である。この手法を、GP を用いたロボットの行動学習の枠組としてよく使われる下記の進化アルゴリズム<sup>(13)</sup> (図1)に適用する。

**Step1** 初期集団の生成：初期の個体集団 ( $P$  個体) を生成する。

**Step2** 選択と適合度計算：個体集団からトーナメント選択により選ばれた  $k$  個体について、現在のセンサ値を入力として個体を実行し、モータスピードの値を決定して結果をセンシングする。得られたセンサ値の系列から個体の適合度を計算し、適合度の上位 2 個体を親の候補とする。

**Step3** 終了条件が満たされれば、終了。

**Step4** GP オペレータの適用：親の候補に対して、遺伝オペレータ (交叉, 突然変異) を適用して子をつくる。

**Step5** 得られた 2 つの子を個体集団の最も適合度の悪い 2 個体と入れ換え、Step2 へ。

Step4 で用いられるオペレータには、交叉、突然変異、交換などがあるが、ここでは、交叉に注目する。一般的に、GP の遺伝子は木構造なので、交叉は、Step2 の選択で選ばれた親の候補から 2 つの個体を選び、それらの遺伝子の部分木を入れ換える操作となる。ここで、一般に交換される部分木の根ノード (交叉点と呼ばれる) は、ランダムに選ばれるが、本研究ではノードの使用率に基づき交叉点を決定する。

例えば、図2は、実環境において移動ロボットが箱押し行動を学習する実験において獲得したロボットプログラム (木構造) の例である。ここで、各個体の使用頻度の遺伝子中のノードの使用回数の初期世代からの累積をそのノードの使用頻度とし、使用頻度をその木の最大使用頻度で割って正規化した値をノードの使用率とすると、このような偏りが生じやすい<sup>(12)</sup>。

ノード使用率によって交叉確率を変えるために、使用率の高いノードと低いノードで高い値をとる交叉確率関数  $H1$  (図3)、使用率の高いノード同士の高い交叉確率関数  $H2$  (図4)、使用率の低いノード同士の高い交叉確率関数  $H3$  (図5) の 3 つを用意した。ここで、図中の  $H_{mn}^{ij}$  は個体  $i, j$  のノード  $m, n$  が交叉の点となる交叉確率、 $R_i^m$  は個体  $i$  のノード  $m$  の使用率である。この確率関数より交叉する 2 つのノードの使用率から交叉確率を変えている。

このようなバイアスのかかった交叉確率による交叉により、オンライン進化において、ER の環境特性に基づいた EC 自体の高速化により従来法に比べて 1.4 ~ 1.6 倍の高速化がはかられている。

GP を用いたオンライン進化の研究としては、移動ロボットが障害物回避を学習する研究がある<sup>(13)</sup>。この研究は、障害物回避といった簡単なタスクではあるが、実時間で学習が行なえることを示した。しかし、GP 自体が高速化されているのではなく、より難しいタスクへの適用に問題がある。同様に Mondada ら<sup>(14)</sup> も実ロボットを用いて進化学習を行わせたが、その際数十時間もかかったことが報告されている。

一方、オフライン進化では、近藤ら<sup>(11)</sup> が、ニューラルネットワークを用いた行動選択を行う進化ロボティクスで、シナプスの

(注1)ここでは、訓練過程と評価過程を区別せず、環境とのインタラクションを通して試行錯誤的に進化・学習し続けることをオンライン進化・学習と呼ぶ

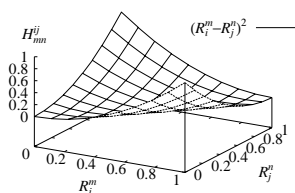


図3 交叉確率関数  $H1$

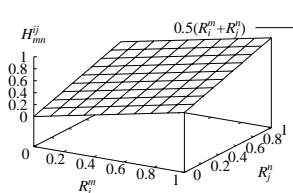


図4 交叉確率関数  $H2$

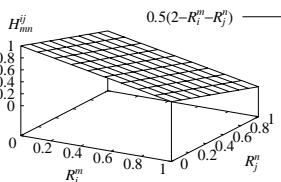


図5 交叉確率関数  $H3$

重みをコーディングするのではなく、どのような学習をどのタイミングで行うかを記述するパラメータをコーディングすることにより、進化ロボティクスにおけるシミュレーションと実世界のギャップを解消する方法を提案している。最終的に進化により得られた結果を実機に実装するアプローチをとっている。

#### 4. 教示学習と対話型進化計算

前章のように、これまで多くのアプローチによる試みが行われてきた。しかし、いずれの研究もオンライン進化の実現という困難な課題に対して、様々な視点からのアプローチを行っているが、学習収束時間やタスクの複雑性や達成度など、これで十分であるとは言えない。

そのなかでも、その対策へのアプローチまたは、今後の研究の発展として人間を含む実環境を対象とした研究の必要性・重要性が指摘されている<sup>4), 5)</sup>。これまで、人間要素をシステムに積極的に取り入れる試みは、多くの研究でなされてきた。

人間の主観的評価を取り入れるといえば、対話型進化計算法 (Interactive Evolutionary Computation; IEC) が有名である。人間を評価系に組み込むというアプローチで進化的に探索を行う手法であり、今までにデザインや音楽、医療など様々な分野で研究が行われてきた。IEC 研究の詳しい説明や研究動向は<sup>17)</sup>を参照されたい。

対話型手法の最大の問題は、人間の疲労をどう軽減するのかがあり、これが実用上の大きな問題になっている。人間が疲れを知らないコンピュータと協調して多くの評価を行うには限界がある。この結果、評価の際の肉体的および心理的疲労軽減のため、評価数を、通常のアフライン探索と比較して非常に少なくせざるを得ないことがあり、学習の収束悪化につながっている。さらに、対話型手法の出力が音や動画像、ロボットの行動などの場合、解候補を時系列提示せざるを得ないことである。評価する人間にとって、

記憶にある過去の解候補との比較を強いられることになるので、心理的負担が大きい。

一方、対話型手法のタスクの場合、人間が適合関数となるため、最適化問題とは異なり対極的最適点というものがない。そのため、少ない世代数でも最適領域に早く到達できれば、実用上十分な場合が多いことが指摘されている<sup>17)</sup>。

#### 5. 進化ロボティクスと対話的設計

ER において、対話的手法によりによりルールを獲得することで進化を促進させることができると考えられる。筆者らは、この枠組みによるアプローチが未知な環境に対する学習、特に試行錯誤の初期段階の学習に効果があると考ええる。また、人間とロボットのインタラクションにより自律ロボットだけでは解けないような局所解に対して解を得るといった効果も期待できる。

この代表的な例として、Lundらは、ロボットの制御器の構築に対話型 GA を適用し、LEGO ロボットの制御器を構築する進化ロボティクスアプローチを提案した<sup>18)</sup>。ロボットコントローラとしてはニューラルネットワークを使用し、対話型 GA と一種の強化学習を融合することで構築にかかる時間を劇的に減らしている。特に興味深いのは、プログラムの知識を持たない子供達による実験である。画面に表示される9つのロボットの動作に対する子供達の嗜好による評価・選択によってロボットは制御器を構築していく。これにより、障害物回避行動や壁つたい行動や、それらの複合行動など簡単に(数分で)獲得をすることができている。これらは、対話型進化手法の主観的評価をうまく利用しているといえる。

##### 5.1 教示者の負荷

一般に、対話型学習の場合、教示を行えば行うほどいいパフォーマンスが得られるが、教示コストは低ければ低いほどよいというトレードオフの関係になっていることは明らかである。この教示者の負担をどれだけ考慮できるかが、対話型設計の大きなポイントになる。

Crandallらは、ヒューマンロボットシステムの設計を目的としてインタラクション効率 (interaction efficiency) の理論的なアプローチを行った<sup>16)</sup>。この研究の興味ある点は、ユーザの認知的負荷を考慮していることである。彼らはユーザの認知的負荷を計測するために、ロボットを操作するメインのタスクと平行して簡単な数値計算を行わせてその効果(消費時間、解答数、正解率)を調べ、ユーザのみが操作する direct control が自律ロボットとの shared-control よりも、認知的負荷が高い事を示している。

Crandallらの例のように、対話型学習を実ロボット環境に適用する場合、ロボットは教示指令が無い場合は自律的に学習を行い、教示者が任意のタイミングで用意した入力装置から直感的にロボットを操作をした場合には、自動的にルールが作成される方法が考えられる。この方法により、

教示者の負荷が軽減される。例えば、これらは入力装置にジョイスティックを用いてロボットの直接操作を行い、その操作とその時の環境情報からルールの自動生成を行うことで教示とする。この枠組みにおいては、多くの個体を比較評価、または評価値の入力を行う必要はなく、肉体的および心理的疲労が大幅に軽減されることが期待される。

堀口らは、この操作型のシステムにより、人間とロボットのインタラクションの設計として相互主導型インタラクションの概念を用い、力覚フィードバックを利用した移動ロボットの自動化プロセスと人間の操作の協調行動を実現<sup>15)</sup>している。

筆者らは、ロボットが動作する際に人間から適切な行為としての教示情報を受け取って、タスクを解決する状態認識・行為ルールの獲得を実現する操作型の手法を提案し<sup>19)</sup>、このような枠組みを対話型進化ロボティクスと呼んでいる。

## 5.2 教示者とロボットとの認識のずれ

ロボットの認識（状態空間）と設計者の認識との間には一般に大きな差異がある。例えば実環境上でロボットを動かした場合、センサ情報の獲得によって形成されるロボットの内部モデルは、その状況および観測の履歴に依存して大きく異なったものになってしまう。その結果、ロボットはあらかじめ設計者が設計した行動規則に従うものの、実際には設計者の意図する行動を生成しないことがある。この問題は、学習ロボットと教示者との間の認識のずれにも当てはまる。

三島らは、学習ロボットと教示者との間の内部状態の違いを Cross Perceptual Aliasing<sup>20)</sup> と呼び、これによって起こる学習者の状態空間に一致する教示データを見つけ、強化学習によってタスクをうまく達成するためにクラスタリングに基づいてその状態空間をうまく修正している。

筆者らは、教示システムの設計問題として教示者とロボットの認識のずれ問題をとりあげた<sup>19)</sup>。この研究では、クラシファイアシステムにより行動学習を行う移動ロボットに対し、次にとるべき行為を人間がジョイスティックで直接教示することにより、一般には非常に効率の悪い進化ロボティクスによる学習の高速化を実現している。ロボットは、教師の教示をクラシファイア（またはルール）として取り込み、学習に役立てる。この場合のロボットのタスクは、障害物回避行動を含む自律移動である。

- 学習者の視点：ロボットの得るセンサ情報と同じものを人間の教師も見て、環境の情報を獲得し、教示を行う。ロボットのセンサ情報を表示するインタフェース（図7）を通して教示がなされる。
- 教師の視点：人間は俯瞰的に環境を見て、環境の情報を獲得する。図6のような環境全体を見渡せる視点から教示がなされる。

比較実験の結果、学習者の視点による教示の方が、学習効率がよかった。これは、学習者の視点の方が教師の視点よ

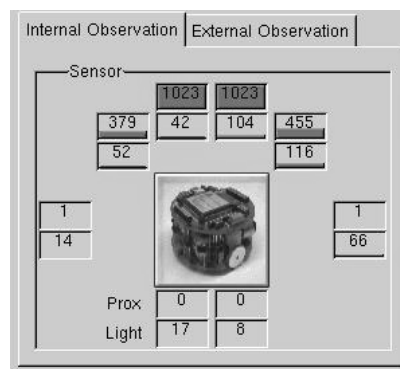


図6 ロボットと同じ視点



図7 教師の視点

りもロボットの環境認識能力とのギャップが少なく、そのためより適切なクラシファイアが学習できたことによる。一方、人間にとっては教師の視点のほうが、自然に環境を認識できるために認知的負荷が少ないため、環境によって視点を自由に変えられる仕組みが必要である。

## 6. おわりに

誌面の都合により、他の興味ある例を紹介することができなかったが、特にヒューマン-コンピュータインタラクション (HCI) の分野では、多くの研究がなされており、重要な知見が得られるだろう。近年では、ヒューマン-エージェントインタラクションの分野で、人間とエージェント間のインタラクション特に相互適応について研究が行われている<sup>21)</sup>。

対話型設計は、実環境で動くシステムには非常に相性がいいが、人間要素を含むため、計算機上とは違い幾つかの扱いにくい問題点が存在する。それに対し、現状では、明確な指針がでていないわけではなく、今後の研究の進展が期待される分野である。本稿であげた対話型設計のアプローチが新たな議論の一助となれば幸いである。

(2003年2月28日受付)

### 参考文献

- 1) P. J. Meyer, P. Husbands and I. Harvey: Evolutionary Robotics: A Survey of Applications and Problems, Proc. of the 1st European Workshop Evolutionary Robotics, pp.1-

21, 1998.

- 2) P. Nordin and W. Banzhaf: Genetic Programming Controlling a Miniature Robot, Working Notes for the AAAI Symposium on Genetic Programming, pp.61-67, 1995.
- 3) 石黒: 進化ロボティクス - その意義,そして今後 -, 第 13 回自律分散システム・シンポジウム, pp.289-294, 2001.
- 4) 浅田: 実環境におけるロボットの学習・進化的手法の適用と課題, 計測自動制御学会誌, Vol.38, No.10, pp.650-653, 1999.
- 5) 久保田: 進化的ロボティクスと適応, システム制御情報学会誌, Vol.47, No.12, pp.565-570, 2003.
- 6) R. Brooks: A robust layered control system for a mobile robot, IEEE Journal of Robotics and Automation, Vol.RA-2, No.1, pp.14-23, 1986.
- 7) J. R. Koza and J. P. Rice: Automatic Programming of Robots using Genetic Programming, Proc. of Tenth National Conference on Artificial Intelligence, AAAI Press/MIT Press, pp.194-201, 1992.
- 8) C. W. Reynolds: Evolution of Obstacle Avoidance Behavior: Using Noise to Promote Robust Solutions, *Advances in Genetic Programming*, pp. 221-241, MIT Press, 1994.
- 9) 山田: 行為に基づく環境モデリングのための移動ロボットの進化的行動獲得, 人工知能学会誌, 14 巻, 5 号, pp. 870-878, 1999.
- 10) E. Uchibe, M. Nakamura and M. Asada: Co-evolution for Cooperative Behavior Acquisition in a Multiple Mobile Robot Environment, IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol.1, pp.425-430, 1999.
- 11) 近藤, 石黒, 内川, Eggenberger: 進化ロボティクスにおける制御器の頑健性の実現: 動的再編成機構を有する神経回路モデルの提案, 計測自動制御学会論文誌, Vol.35, No.11, pp. 1407-1414, 1999.
- 12) 片上, 山田: ノード使用頻度に依存した交叉による進化ロボティクスの高速化, 人工知能学会誌, Vol.16, No.4, pp.392-399, 2001.
- 13) P. Nordin and W. Banzhaf: An on-line method to evolve behavior and to control a miniature robot in real time with genetic programming, *Adaptive Behavior*, Vol.5, No.2, pp. 107-140, 1996.
- 14) F. Mondada and D. Floreano: Evolution and mobile autonomous robotics, In Proc. of the Evolutionary Engineering Approach, pp.221-249, 1995.
- 15) 堀口, 榎木: 遠隔操作環境における共有自律の実現を指向した人間 ロボット間の相互主導型インタラクションの設計, 計測自動制御学会論文誌, Vol.38, No.12, pp.1097-1106, 2002.
- 16) J. W. Crandall and M. A. Goodrich: Characterizing Efficiency of Human Robot Interaction: A Case Study of Shared-Control Teleoperation, IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.1290-1295, 2002.
- 17) 高木, 畝見, 寺野: 対話型進化計算法の研究動向, 人工知能学会誌, Vol.13, No.5, pp.692-703, 1998.
- 18) H. H. Lund, O. Miglino, L. Pagliarini, A. Billard and A. Ijspeert: Evolutionary Robotics - A Children's Game, IEEE 5th International Conference on Evolutionary Computation, IEEE press, 1998.
- 19) 片上, 山田: 対話的進化ロボティクスの観察に基づく教示の設計, システム制御情報学会誌, Vol.16 No.6, pp.279-286, 2003.
- 20) C. Mishima and M. Asada: Active Learning from Cross Perceptual Aliasing Caused by Direct Teaching IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.1420-1425, 1999.
- 21) 山田, 角所, 新田ほか: 特集 「HAI: ヒューマンエージェントインタラクション」にあたって, 人工知能学会誌, Vol.17, No.6, pp.657-700, 2002.

## [ 著 者 紹 介 ]

かた がみ だい すけ  
片 上 大 輔 君 (正会員)



1973 年 9 月 3 日生 . 1996 年 , 電気通信大学電気通信学部機械制御工学科卒業 . 2002 年東京工業大学大学院博士課程了 . 同年 , 東京工業大学大学院総合理工学研究科助手 , 現在に至る . 博士 (工学) 1999 年電気学会電子・情報・システム部門大会優秀論文発表賞受賞 . 知的ロボット , ヒューマンエージェントインタラクションなどの研究に従事 . 人工知能学会 , 日本ロボット学会 , システム

制御情報学会 , IEEE 各会員 .

やま だ せい じ  
山 田 誠 二 君 (正会員)



1960 年 10 月 11 日生 . 1984 年大阪大学基礎工学部卒業 . 1989 年同大学院博士課程修了 . 博士 (工学) . 同年大阪大学基礎工学部助手 . 1991 年同大学産業科学研究所講師 . 1996 年東京工業大学大学院総合理工学研究科助教授 . 2002 年国立情報学研究所教授 , 現在に至る . 工学博士 . 人工知能 , 特に , 知的 Web , ヒューマンエージェントインタラクションに興味をもつ . 情報処理学会 ,

日本ロボット学会 , 電子情報通信学会 , AAAI , IEEE , ACM 各会員 .