

インタラクティブ性のある適応型 行動選択ネットワークの提案

小山 宗三^{1,a)} 高屋 英知^{2,b)} 加藤 慶彦^{2,c)} 覚井 悠生^{2,d)} 栗原 聡^{1,e)}

概要：将来的に活躍するであろう車の自動運転技術，介護ロボットは，人と同じ空間に常にいることが考えられる．そうした時，人を傷つけず与えられた目標を達成するには，環境が動的に変化することによって備えた機能が不可欠である．本研究では，人と共同して目的を達成する適応型行動選択ネットワークを提案する．目的を効率的に達成する古典的人工知能の枠組みに注目し，この枠組みに動的に環境を取得し即座に反応する枠組みを統合する事で，熟考性と即応性を持ったプランニングを実現した．それにより人とのインタラクションに対応が可能な行動選択ネットワークを実装した．さらに，提案手法を実環境での実験を行うことでその有効性を検証した．本実験を通し，提案手法だけではなくまた別の枠組みを統合する必要性や，提案手法に対するパラメータ調整を行う機構の確立といった課題を確認できた．

Proposal of an Interactive Action Selection Network

1. はじめに

我々が生活する実社会は人とロボットが共存する社会へと向かいつつある．家の中を自動で掃除してくれるルンバや，受付の自動化をしてくれる Pepper の浸透などがその実例である．その背景にはセンサや機械の処理能力の発達，急速に進む IoT 化，ロボットの労働力としての利用などが挙げられる．しかし，未だにこれらのロボットは動的に変化し続ける実環境で有効に機能しているとは言えない．例えば，ルンバは人がいるいないに関わらず部屋を限なく掃除し，Pepper は人のいない環境でも話しかけることがある．これらのロボットは人とのインタラクションが少なく，環境に適応していないことは大きな問題とは言えない．しかし，将来的に活躍するであろう車の自動運転技術，介護ロボットは，人と同じ空間に常にいることが考えられる．そうした時，人を傷つけず与えられた目標を達成

するには，環境が動的に変化することによって備えた機能が不可欠である．

本研究の目的は，人とのインタラクションに対応しつつ，人と共同して目的を達成する適応型行動選択ネットワークの提案にある．目的を如何にして達成するかというプランニングに関しては，環境をはじめに取得し，その後いくつかの処理を施しプランを決定するといった熟考的なアプローチとしてこれまでも多くの手法が提案されてきた．その代表例が STRIPS[1] や階層型タスクネットワーク (HTN) プランニング [2]，Agent Network Architecture[3][4] (以降 ANA) といった手法である．だが，これらは熟考性はあるつつも急に環境が変化したことに対応する即応性を持ち合わせていなかった．そのためこれまでのロボットは環境が理想どおりに変化する，もしくは急激な変化が最初から予期されることを前提としていた．

即応性をもつ手法としては Brooks に提案されたサブサンプリングアーキテクチャ [5][6] が存在する．しかし，即応性のモデルだけではあくまで反射的の反応となるため，全体で一連の行動を見た時に無駄な動きが多く，最適行動とはかけ離れた行動を取ってしまうという欠点がある．

現実社会の常に変化する環境に対応するには，即応性と熟考性の両方を持ったプランニングモデルを構築しなければならない．この即応性と熟考性というものにもある程度

¹ 慶應義塾大学理工学部
Faculty of Science and Technology, Keio University
² 慶應義塾大学大学院理工学研究科
Graduate School of Science and Technology, Keio University
a) k.rex829@keio.jp
b) etakaya@keio.jp
c) yoshihikokato@keio.jp
d) samei@keio.jp
e) satoishi@keio.jp

の正確性が求められる。

先行研究として、人とのインタラクション性をもつプランニングがいくつか提案されている。近年はコンピュータ技術の発達もあり、Chin[7]らの述べたような分散並列かつ同時に計算可能なマルチエージェントシステムをもつプランニング[8][9]が提案されている。さらには熟考型プランニングにおける再プランニングなどの計算効率を上げた研究[10][11]や、熟考層と即応層を分けた階層型にすることでインタラクションに対応した研究[12][13][14]も存在する。しかし、これらのプランニングは限られた環境でしか有効とは言えない。様々な種類の環境の変化や場面に対応するためには栗原らの研究[15]のように複数の代替プランを用意しておくことや、新たな動きを追加できる拡張性を持ったモデルでなければならない。

そこで、本論文では目的を効率的に達成する古典的人工知能の枠組みに注目し、この枠組みに動的に環境を取得し即座に反応する枠組みを統合する事で、熟考性と即応性を持ったプランニングを提案した。そして人とのインタラクションに対応が可能な行動選択ネットワークを実現した。さらに実環境での実験を行うことでその有効性を確認した。

本論文の構成について述べる。第2節で提案するシステムについて述べ、第3節で提案するシステムの有効性を検証するための実験環境、手法について述べる。第4節でその実験結果について述べ、第5節で実験結果を元に、本研究における提案システムの有効性、および欠点などを評価、考察する。最後に第6節で全体の流れをまとめ、今後の課題について述べる。

2. 提案システム

本節では提案するシステムについて述べる。提案システムには、古典的人工知能の枠組みの中に、動的に環境を取得し即座に反応する枠組みを統合した。これにより熟考性と即応性を持ったプランニングができるようにした。具体的には古典的枠組みとしては前節で述べたANAを用い、そのアルゴリズムを改良することで常に環境を取得し、即応性を持てるようにした。

2.1 ANA

ANAは分散した competence モジュール (以降モジュール) によって構成されており、モジュールはそれぞれネットワークを通して繋がっている。

モジュールはロボットに実際の行動をさせる action モジュール、「観察する」といった実際の行動ではない belief モジュールに分けられる。カップを認識するといったことは belief モジュールにあたり、カップを掴むといったことは action モジュールに当たる。それぞれのモジュールが condition-list, add-list, delete-list, 活性レベル, 閾値の5

つを持つ。condition-list はそのモジュールが実行される環境の条件を指し、add-list はモジュールの実行により環境に新たに追加される要素、delete-list は逆に新たに削除される要素を示す。モジュールは活性値が閾値を超えた時に実行される。閾値を小さく決めておけば、即応的な行動を取るようになり、逆に大きくすれば熟考的な行動を取るようになる。

ネットワークは predecessor, successor, confictor の3つのリンクから構成される。successor は、あるモジュールの add-list の要素を実行に必要とするモジュール、つまりそれが実行されれば自分も次に実行されるといったモジュール間に張られ、その逆方向に predecessor が張られる。confictor は condition-list に同じ要素は含むが、add-list に別の要素をもつ競合関係のモジュール間に張られる。モジュールはこのネットワークを通して活性値が上げられていく。まず最終的な目的(goal)と、現在の環境(sensor-data)から実行が可能なモジュールの活性値をあげる。その後活性値が上がったモジュールは、さらに successor を通して他のモジュールの活性値を上げ、最終的な目的(goal)のモジュールから predecessor を通して他のモジュールの活性値を上げる。最後に全てのモジュールが confictor を通して競合関係のモジュールの活性レベルを下げ、すでに達成され、最終的にはそれも達成されていなければならないゴール(protected-goal)の要素を delete-list に含むモジュールの活性値を下げる。これら一連の活性値の電波はセクションと呼ばれる。

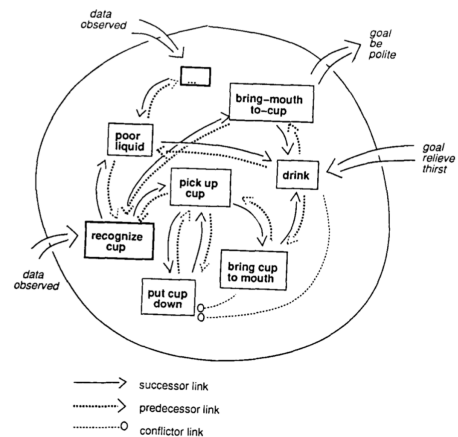


図1 モジュール間に張られたネットワーク [4]

図1はモジュール間のネットワークの例である。この図1には喉の渇きを癒す(relieve thirst)を達成するためのモジュールが描かれている。

実行するモジュールに関してはセクションを経て活性値が閾値を超えたものが実行される。具体的な活性値の値は以下のように計算される[4]。

$$\begin{aligned} activation(module, t) = & D \sum_{j=1}^n \frac{1}{d(module, sensor_data_m_j)} \\ & + G \sum_{k=1}^n \frac{1}{d(module, goal_k)} \\ & - C \sum_{l=1}^n \frac{1}{d(module, protected_goal_l)} \\ & + M \cdot activation(module, t-1) \end{aligned}$$

$d(a, b)$: distance(a から b まで何ホップでたどり着くか)

T : threshold(閾値)

D : data_orientedness

(現状の環境との関連度をどれだけ考慮するか)

G : goal_orientedness

(ゴールとの関連度をどれだけ考慮するか)

C : conflict_sensitivity(競合関係をどれだけ考慮するか)

M : mean_activation_level(過去の値をどれだけ考慮するか)

ANA は以下のようなアルゴリズムとなっている。モジュールと初期の環境、物の位置などは、事前に定義しておく必要がある。

ANA のアルゴリズム

- (1) モジュールの活性値が閾値を超えるまでセクションを繰り返す。
- (2) 活性値が閾値を超え、初期の環境 (現環境) から実行可能なモジュールを実行する。
- (3) モジュールの add-list を初期の環境 (現環境) に加え、delete-list を初期の環境 (現環境) から削除する。
- (4) 実行したモジュールの活性値をゼロにする。
- (5) 1~4 を繰り返す。

以上のアルゴリズムが目的が達成されるまで実行される。

2.2 環境取得型 ANA

古典的枠組みとして ANA を用いたことには 3 つの大きな理由がある。

1 つ目は現環境から即座にモジュールが選択され実行されることである。例えば STRIPS[1] は現環境からゴールまで導く一連の動きが見つからなければ行動が始まらない。対して ANA はゴールまでの動きが見つからなくても、現環境からゴールまで導くための動きに近いモジュールが場当たり的に選択される。例えば現環境でカップが存在していることを把握したならば、場当たり的にカップを手にとるといったモジュールが選択される。そのため環境が変化したとしても即応的な行動ができるプランニングアーキテ

クチャであると考えられる。

2 つ目はセクションが他のモデルとは違い、ANA では活性値、パラメーターの数値によって決まることである。ANA は場当たり的にモジュールが選択されるため、最適ではないモデルが選択されることもある。このセクションを最適に近づけるために ANA では活性値、パラメーターの調整が必要になるが、これらの数値は近年人気が高まりつつある強化学習と絡めることができると考えられる。

3 つ目はモジュール群に拡張性があることである。将来的に、人とのインタラクションに対応したロボットが活躍する際には自身で新たな動き (モジュール) を与えられるのではなく自ら獲得する必要がある。ANA の活性値、パラメーターはモジュール間のリンクではなくモジュールそのものに与えられる。そのため ANA は実行途中にモジュールが追加、もしくは取り除かれたとしても冗長性がある。つまり ANA は自己の動きの獲得にも対応できると考えられる。

しかし、従来の ANA には Pfeifer ら [16] が指摘した通り、従来の熟考型のプランニングと同じくモジュールの実行中に環境が変化した場合に対応することができない。現実環境でモジュールの開始から終了までは時間がかかり、環境が変化する可能性は大いに考えられる。

そこで、ANA のアルゴリズムを改良し、以下のようなアルゴリズムの環境取得型 ANA を実装した。

環境取得型 ANA のアルゴリズム

- (1) 現環境を取得する。
- (2) セクションを一度行う。
- (3) 活性値が閾値を超え、初期の環境 (現環境) から実行可能なモジュールがあるなら実行する。
- (4) 実行したモジュールの add-list を初期の環境 (現環境) に加え、delete-list を初期の環境 (現環境) から削除する。
- (5) 実行したモジュールの活性値をゼロにする。
- (6) 1~5 を繰り返す。

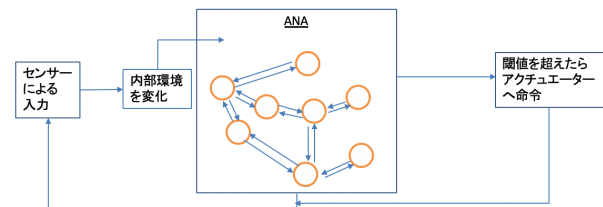


図 2 環境取得型 ANA

大きな変化としてはセクションを行う前に毎回現環境を取得していることである。これにより実環境にそぐわないモジュールが選択されるということがなくなる。さらに、

実環境に大きな変化が起きた際、現在動いているモジュールを停止させもう一度セクションを行うこととした。こうすることでモジュールが実行されている途中に実環境に大きな変化が起きた際にもそれに対応した動きが可能となると考えられる。

2.3 実験環境

本節では提案システムである環境取得型 ANA の有効性について検証した実験環境および手法について述べる。

2.4 想定したストーリー

本実験では環境取得型 ANA の有効性を検証するために、ミルク入りコーヒーを入れるというストーリーを用いた実験を行った。実際に人とのインタラクションを織り交ぜるため、モジュールの動きをロボットアームに行わせることにした。目的を達成するためのロボットの最短の動きは以下の通りである。

- (1) 空のコップを設置する。
- (2) 空のコップにコーヒーを注ぐ。
- (3) コーヒーを入れたコップにミルクを注ぐ。
- (4) お客さんに提供する。

モジュールは以下のように実装した。

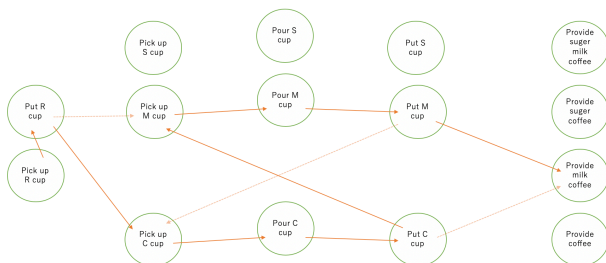


図 3 ストーリーを実行するためのモジュール

全 4 種類のカップの意味は以下の通りである。

- R cup: 空のコップ, 提供するカップ
- C cup: コーヒーが入っているコップ
- M cup: ミルクの入っているコップ
- S cup: 砂糖の入っているコップ

図 3 の矢印が目的を達成するための動きであり、点線の矢印は矢印の向いた先のモジュールを実行できなかった際に実行するであろうモジュールを指している。図 3 にある通り、本実験では目的とは関係ない砂糖を注ぐ動きのモジュールも用意した。これらのモジュールが実行されてしまうと、本来目的とするミルク入りコーヒーを提供することができなくなり、目的が達成できなくなってしまう。

本実験では、環境取得型 ANA が熟考性と即応性を持つ

た枠組みであること、特に従来の ANA にはない即応性を持つことを示すため、ストーリーを行っている最中に人が介入した時の変化を観察した。本実験では 8 つの人が介入方法を行い、その時のロボットアームの動きを観察した。

8 つの実験

- (1) 空のコップをセットする (Pick up R cup) 最中に掴もうとしたコップの位置をずらした。
- (2) 空のコップにコーヒーを注ぐ (pour C cup) 最中に空のコップの位置をずらした。
- (3) コーヒー入りのコップを掴む (Pick up C cup) 最中に掴もうとしたコップの位置をずらした。
- (4) コーヒーを注いだ後、ミルク入りのコップを掴む (Pick up M cup) 最中に掴もうとしたコップの位置をずらした。
- (5) 空のコップにミルクを注ぐ (pour M cup) 最中に空のコップの位置をずらした。
- (6) コーヒー入りのコップを掴む (Pick up C cup) 最中に人が代わりにコーヒーを注いだ。
- (7) コーヒーが空のコップに注がれた後、ロボットアームではなく、人がミルクを注いだ。
- (8) コーヒーが空のコップに注がれた後、人がミルク入りのコップをどこかに隠した。

実験 1~5 の 5 つの場面に関しては、環境取得型 ANA が動的に環境を取得し、動くことができる即応性を持っているかを判断するために行った。6,7 に関しては、ロボットの作業途中に人が介入し、作業を手伝うといったインタラクションが行われた際にも対応が可能かということを確認するために行った。8 に関しては、目的の達成が不可能となってしまう場合にも、それに近い目標を達成することができるかを確認するために行った。

2.5 ストーリー実行環境

これらのストーリーを実行するために以下のような環境を整えた。

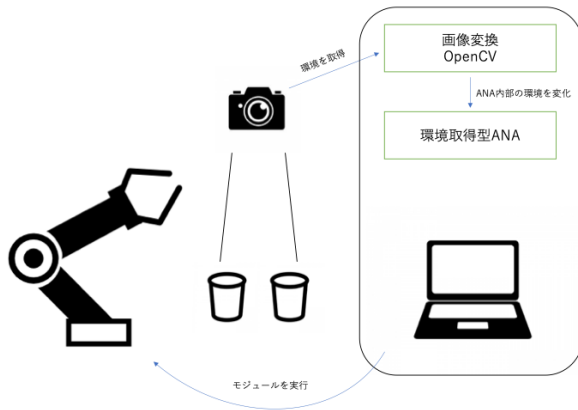


図 4 ストーリーを実行するための環境

まず、外部の環境を常に取得する方法として、本実験では色検知により設置されているカップの存在や、カップの位置を取得した。カメラ (Logitech HD Pro Webcam C920) から作業する場所の画像を取得し、OpenCV の色検知を用いることによってこれを実現した。人の介入に関しても、肌の色の検知により環境として取得している。

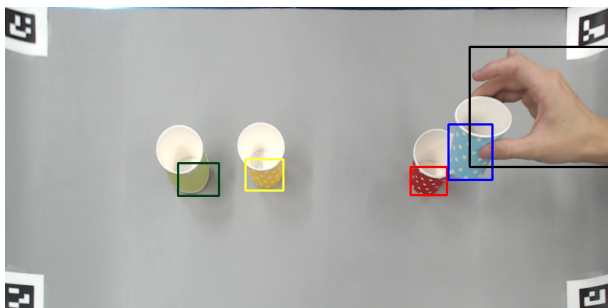


図 5 OpenCV による手の検知

本実験ではコーヒーやミルクを注ぎ終わったカップは撮影している範囲外に置くようにした。そして色検出により特定のカップを検出できなかった場合には、それはすでに注ぎ終わっていると環境取得型 ANA の内部環境を変化するようにした。こうすることによって、人が代わりにコーヒーやミルクを注いであげた場合に、それを検知することができるようになった。

以上によって、取得した情報から環境取得型 ANA の内部環境を変更した。モジュールを実行する際には、アクチュエーターとしてロボットアーム (NIRYO ONE) を用い、実行したいモジュールの動きの情報をロボットアームに送り、指定した動きをさせた。人が介入した時の処理としては、ロボットアームが動作途中で環境が変化 (人が介入) した場合にはロボットアームを初期位置へ一度戻すようにした。これは階層構造による反射とも取れる。その後、モジュールが実行する前、アルゴリズムの“現環境を取得する”に戻るようにした。これによって変化した環境をもう一度捉え直すことができると考えた。

3. 実験結果

ミルク入りコーヒーを作る最中に、人が介入した 8 つの場面での変化を観察した。結果を表 1 に示す。

表 1 実験結果

実験番号	実験結果
実験 1	ロボットアームを空のコップをずらした位置に移動させ、再び空のコップを手に取り所定の場所へセットしたのちに作業を続けた。
実験 2	ずらした位置に持っているコーヒー入りのカップを選び、注いだ。
実験 3	ロボットアームはコーヒー入りのカップを掴むことをやめ、ミルク入りのカップに手を伸ばし、ミルクを注ぎ終えた後にコーヒー入りのカップを掴み注いだ。
実験 4	ロボットアームを再びミルク入りのカップをずらした位置に移動させ、ミルク入りのカップを掴み作業を続けた。
実験 5	コーヒー入りのカップを掴む最中に掴もうとしたカップの位置をずらした時と同じように、ずらした位置に持っているミルク入りのカップを選び、注いだ。
実験 6	ロボットアームはコーヒー入りのカップを掴むことをやめ、ミルク入りのカップを掴み注いだのちにミルクコーヒーを提供した。
実験 7	ロボットアームはミルク入りのカップを掴まず、ミルクコーヒーを提供した。
実験 8	ロボットアームはミルク入りのカップではなく、砂糖入りのカップを掴み注いだのちに砂糖入りコーヒーを提供した。

4. 考察

本節では実験の結果について考察する。

実験 1～5 から環境取得型 ANA がセレクション毎に環境を取得していることによって、人の介入や邪魔に対し、対応が可能になったことがわかった。特に実験 3 では一度失敗したモジュールではなく、同じ条件下で実行可能な別のモジュールが選択された。これは最初コーヒーを掴もうとした時に、コーヒー入りのカップを掴むモジュールの活性値が 0 になり、もう一度その状態からセレクションが行われた際に最初の時にある程度、活性値の上昇していたミルク入りのカップを掴むモジュールの活性値が閾値を超えたからである。

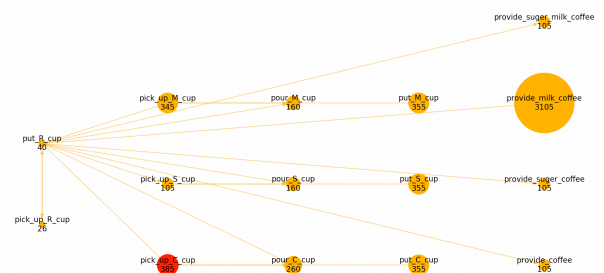


図 6 コーヒー入りカップを掴む前の活性値

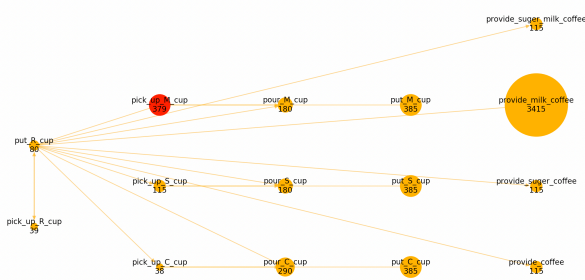


図 7 コーヒー入りカップを掴むモジュール失敗後の活性値

図 6, 7 のモジュール名の下の数値は各モジュールの活性値を表しており、赤くなっているのはそのモジュールが選ばれたことを示す。図 6 からコーヒー入りカップを掴む前にもミルク入りのカップを掴むモジュールの活性値が他に比べて高められていることがわかる。このことから環境取得型 ANA がプランから外れても、目的のために方法を変えてプランを実行できることが確認できた。

実験 6, 7 では人間がやってくれた作業を飛ばし、ロボットアームが次の作業へと進んだことがわかる。ANA の説明や実験 4 からわかる通り、ANA ではすでに満たされた条件を満たそうとするモジュールは選択されない。従来の ANA のように環境取得型 ANA はセクション前に毎回環境を取得し内部環境を更新している。そのため人が作業をしてくれたことを検知し、次の作業へと進むことができたと考えられる。これを応用すれば人がある作業を進め、その間にロボットも別の作業を同時に進めるといった半順序プランニング、共同作業が可能となるだろう。

実験 8 ではロボットアームが目的を達成することが不可能な状況で、それに近い目標を達成することができていた。これは ANA のもつ、場当たりの行動が可能な特徴からであると考えられる。ANA では活性値の上昇によって、環境に適した行動を取ることはもちろんであるが、その間にもそれに近い行動や現環境で可能なものの、ふさわしくない行動も活性値が少しずつではあるが上がっていく。環境取得型 ANA は最適解を見つけられなくても、そこで立ち止まらずに、本来の目標に近い別の目的を達成するように行動することが可能だと考えられる。しかし今回の場合、人がミルク入りコーヒーが出せない場合、砂糖入りコーヒーではなくブラックのコーヒーを求めるかもしれない。それを解決するのが ANA の持つ、現状の環境との関連度やゴールとの関連度などを表すパラメーターであるだろう。本実験では目的に応じてコーヒーを提供できるように手でパラメータ調整を行ったが、調整を変えることにより実験 8 のような状況でブラックのコーヒーを提供できるようにもなると考えられる。

本実験では人が介入した際に初期位置へ一度戻すようにした。もしこの仕組みがなかった場合、実験 2 では人の動きに合わせて注ぐ前に、何も無いところに注いでこぼして

しまっていたかもしれない。実社会においては急にボールが飛んでくるといってもありえる。そのような状況では反射的に避けるといった枠組みが必要になる。今後、環境取得型 ANA を階層構造にするなど現状の枠組みから外れたシステムの構築も必要となるだろう。

5. 終わりに

本研究では、提案システムである環境取得型 ANA が人とのインタラクションに対応しつつ、人と共同して目的を達成する適応型行動選択ネットワークであることを確認する実験を行なった。実験 1~5 により、環境取得型 ANA が突然の環境の変化にも対応し、臨機応変に行動を取ることができるとわかった。実験 6,7 から、環境取得型 ANA が人とのインタラクション、共同作業に対しても有効にプランニングが可能なが確認できた。実験 8 から、環境取得型 ANA が目的を達成できない状態でもそれに近い目標を達成しようとする行動選択をできることが確認できた。実験を通し、環境取得型 ANA だけではなくまた別の枠組みを統合する必要性についても確認できた。

5.1 今後の課題

本研究で行なった実験 8 から、環境取得型 ANA のパラメータ調整の重要性が確認できた。本研究の実験ではモジュールの数が少なかったため、手でパラメーターの調整を行なった。しかし、実社会で運用することを考えた時、モジュールの数は無数に作られることが想定される。その際、パラメータを予め決めておくという方法では複雑に変化する環境変化に対応できるとは考えられない。今後社会で環境取得型 ANA が活躍するにあたり、与えられた目的に対して過去の経験や知識から自動でパラメータを調整する機構の確立を今後の課題としたい。

参考文献

- [1] Richard E.Fikes,Nils J. Nilsson. Strips: A new approach to the application of theorem proving to problem solving, *The 2nd IJCAI, Imperial College, London, England*, September 1-3, (1971).
- [2] Pascal Bercher , Ron Alford2 and Daniel Holler: A Survey on Hierarchical Planning – One Abstract Idea, Many Concrete Realizations, *International Joint Conference on Artificial Intelligence Survey track*, pp.6267-6275, (2019).
- [3] Pattie Maes: How to do the right thing, *Connection Science*, Journal 1, pp291-323, (1990).
- [4] Pattie Maes: The agent network architecture (ANA), *ACM SIGART Bulletin*, Volume 2 Issue 4, Aug. 1991 pp.115-120, (1991).
- [5] R. Brooks: A robust layered control system for a mobile robot, *IEEE Journal on Robotics and Automation*, Volume:2 , Issue:1, (1986).
- [6] 五味 隆志: サブサンプリング・アーキテクチャ, 日本ファジィ学会誌 Vol7, No.5, pp.909 – 930, (1995).
- [7] Kim On CHIN, Kim Soon GAN, Rayner ALFRED, Pa-

- tricia ANTHONY, Dickson LUKOSE: Agent Architecture: An Overview, *Transactions on science and technology 1.1*, pp.18-35, (2014).
- [8] Satoshi Kurihara, Shigemi Aoyagi, Rikio Onai, Toshiharu Sugawara: Adaptive selection of reactive/deliberate planning for a dynamic environment, *Robotics and Autonomous Systems*, Volume 24, Issues 3–4, 30 September 1998, pp.183-195, (1998).
- [9] 林 久志, 十倉 征司, 尾崎 文夫, 長谷川 哲夫: Dynagent: 割込み HTN プランニングエージェント, 知能と情報 (日本知能情報ファジィ学会誌), Vol. 21, No.5, pp.859-869, (2009).
- [10] Michael Cashmore, Andrew Coles, Bence Cserna, Erez Karpas, Daniele Magazzeni, Wheeler Ruml: Replanning for Situated Robots, *Twenty-Ninth International Conference on Automated Planning and Scheduling*, Vol 29, pp.665-673, (2019).
- [11] Ishani Chatterjee, Maxim Likhachev, Ashwin Khadke, Manuela Veloso: Speeding Up Search-Based Motion Planning via Conservative Heuristics, *Twenty-Ninth International Conference on Automated Planning and Scheduling*, Vol 29, pp.674-679, (2019).
- [12] Philipp Allgeuer, Sven Behnke: Hierarchical and State-based Architectures for Robot Behavior Planning and Control, *Proceedings of 8th Workshop on Humanoid Soccer Robots, International Conference on Humanoid Robots (Humanoids)*, (2013).
- [13] 松岡 諒, 榎木 哲夫, 堀口 由貴男, 中西 弘明, 階層型プランニングシステムによる産業用ロボットの自律的エラーリカバリー. 計測自動制御学会論文集, Vol.53, No.1, pp.80-89, (2017)
- [14] Yuqian Jiang, Nick Walker, Justin Hart, Peter Stone: Open-World Reasoning for Service Robots, *Twenty-Ninth International Conference on Automated Planning and Scheduling*, Vol 29, pp.725-733, (2019).
- [15] 栗原 聡, 高屋 英知, 橋 良暢, 芦原 佑太: 汎用 AI 実現のための鍵となる自律性とマルチモーダル性についての考察, *The 31st Annual Conference of the Japanese Society for Artificial Intelligence*, (2017).
- [16] Rolf Pfeifer, Bruce Blumberg, Jean-Arcady Meyer, Stewart W. Wilson: An extension of Maes Action Selection Mechanism for Animats, *Proceedings of Fifth International Conference on Simulation of Adaptive Behavior*, pp.153-158, (1998).