

マルチロボットによる箱押しのための 明示的通信を用いない適応的行為選択

山田 誠二* 齋藤淳也*

Adaptive action selection without explicit communication for multi-robot box-pushing

Seiji Yamada* and Jun'ya Saito*

This paper describes a novel action selection method for multiple mobile robots box-pushing in a dynamic environment. The robots are designed to need no explicit communication, and be adaptive to a dynamic environments by changing modules of behaviors. Though it is a significant problem to deal with adaptive action selection for multiple mobile-robots in a dynamic environment, few studies have been done. Decentralized control of robots without explicit communication is also practical and important for robustness. Thus we propose adaptive action selection without explicit communication for multi-robot box-pushing, which changes an available behavior set depending on a situation. First four situations are defined with two parameters: existence of other robots and task difficulty. Next we design a set of behaviors for each situations, and mobile robots are programmed to act with behavior-based approach. We fully implement our method on four real mobile robots, and make experiments in dynamic environments.

Key Words: Multi-robot box-pushing, adaptive action selection, behavior-based robots

1. はじめに

単独のロボットでは達成できないタスクを扱うために、複数の移動ロボットによる協調作業の研究が活発に行われてきた。それらは、大きく分けて集中制御 (centralized control) [12] [21] [22] と分散制御 (decentralized control) [1] [3] [6] [7] [10] [11] [15] [20] に分類できる。

集中制御では、中央システムが、センシングや通信により、環境および全てのロボットに関する大局的な情報を獲得し、全てのロボットの行動を決定する。そして、その行動の命令を通信により各ロボットに送り、ロボットはその命令に従って行動する。このアプローチは、大局的な情報を用いて全体的な計画を立てることができるため、各ロボットが合理的かつ効率的に行動できるという利点がある。しかし反面、中央システムが停止すると、全てのロボットが止まってしまうという意味での頑健性の欠如、さらに大局的な情報を得ることの現実的困難さなどの問題がある。

一方、分散制御のマルチロボットシステムも、集中制御とともに活発に研究されてきた。中央システムが存在しないため、一部のロボットが停止したとしても、その影響でシステム全体が停止することを避けることができるという意味で、集中制御

よりも頑健である。また、大局的な情報とそれに基づく最適な行動の計画を必要としないため、行動決定が高速である。

本研究では、マルチロボットシステムの構築に際し、実際的なメリットが大きいと考えられる分散制御を採用する。しかし、従来の中央制御、分散制御の研究は、以下の重要な問題をもつと考えられる。

- (1) 明示的通信の必要性：中央制御におけるほとんどのマルチロボットシステム [6] [12] [21] [22] が、無線などによる明示的通信を必要とする。また、分散制御でも、明示的通信を必要とするもの [2] がある。しかし、このような明示的通信は、実際にはかなりのコストがかかり、実環境によっては非常に不安定でさえある。よって、明示的通信を必要としないマルチロボットシステムが望まれる。
- (2) 動的環境への適応：実際の環境では、ロボットの故障、新しいロボットの導入、タスクの変更等により、しばしば環境は変化する。しかし、中央制御、分散制御に関わらず、ほとんどのマルチロボットシステム [1] [3] [6] [7] [10] ~ [12] [15] [20] ~ [22] が、そのような動的環境に対して適応するメカニズムを持っていない。

これらの問題に対処するために、我々は、動的環境におけるマルチロボット箱押し問題を目標タスクとして、適応的行為選択メカニズムを提案し、それを実ロボット上に実装する [18]。この行為選択手法は、明示的通信を必要とせず、ロボット数の変化とタスクの難易度の変化による環境の変化に対して適応可

原稿受付 1998年5月11日

*東京工業大学大学院総合理工学研究科知能システム科学専攻

*CISS, IGSSSE, Tokyo Institute of Technology, 4259 Nagatsuta-cho, Midori-ku, Yokohama 226-8502, JAPAN

能である。

本論文では、最初に、他のロボットの存在の有無とタスクの難易度をパラメータとして、4つの状況を定義する。この状況は、環境のクラスであり、本研究において動的環境は、状況の遷移としてモデル化される。そして、それぞれの状況に対し、適切な行動の集合を設計し、複数の移動ロボットが行動ベースアプローチで構成される。最後に、我々の適応的行動選択を実機の移動ロボット上に実装し、いくつかの動的環境において実験を行い、評価する。

人工知能の分野では、まず機械学習において、通信なしの箱押し行動を強化学習で実現する研究 [19] がある。しかし、シミュレーションによる実験だけであり、また本研究で扱っているような箱を見つける作業や動的環境への適応は、考慮されていない。また、分散人工知能の分野では、契約ネットを用いてマルチロボットを制御し、さらに事例ベース推論により効率化をはかる研究 [14] を始め様々な研究があるが、残念ながらそれらはすべてシミュレーションに留まっている。

行動ネットワーク [8] や服属アーキテクチャ [4] でも、常に環境を監視し、その変化に反応して、行動選択を行う機能が実現されている。しかし、本研究では、環境を適切に特徴づけ、かつ観測可能なパラメータを用いて変化する環境を記述し、それぞれの環境に適した行動の集合を用意しており、動的環境に適応するためのより具体的な枠組になっている。

また、マルチロボットによる協調的箱押しに関しても先行研究がある。ACTRESS [2] は、箱が一台で押せないときに他のロボットと協調する枠組みである。ここでは、他のロボットへの明示的な通信による協調の要請が行われる。本研究では、そのような通信は必要ない。Noreils とも、複数台の移動ロボットによる協調的箱押しを実現している [13]。しかし、彼らの研究は、問題の分割、ロボットへのサブタスク割り当てなどが中央集権的に行われる点が、本研究との違いである。

さらに、Parker は、適応的でフォールトトレラントなマルチロボットシステムである ALLIANCE アーキテクチャを提案している [16]。ALLIANCE は、一台のロボットが複数の行動集合をもち、どの行動集合により行動選択および行動の実行を行うかの決定を、行動集合の活性化により行う。センサ入力、他の行動集合の活性度、ロボット間通信などによって計算された動機が閾値を超えると、活性化が行われる。行動集合の活性度間には、抑制性があるので、一度に一つの行動集合だけが活性化される。このようなメカニズムは、本研究と類似点が多い。しかし、実装された ALLIANCE では、ロボット間のインタラクションとして、ある行動集合が活性化しているロボットが、回りのロボットにブロードキャストによる明示的通信を行っており、その伝達情報も現在自分の取り組んでいるタスクという抽象的な記号である。これに対し、本研究では、他のロボットのタスクを知る必要なく協調行動をとることができるし、また明示的通信を必要としない。

これまで、本研究で扱っているように、タスクの難しさの変化と他のロボットの有無による変化により、マルチロボットの動的環境を記述した研究は、筆者の知る限りではない。よって、動的環境への適応に対し、新しいパラメータにより動的環境を

記述し、適応できるシステムを実機を使って実現したことには、意義がある。

2. 動的環境を記述するための状況の定義づけ

2.1 環境とタスク

まず最初に、ロボットのタスクを設定する。本研究における複数移動ロボットのタスクは、「目的地まで箱をおすこと」である。また、本研究における環境は、「110cm×90cm で、周囲を白いプラスチック板で囲って壁とした平坦な長方形のテーブル」である (Fig.1)。光源がテーブルの横に設置されており、移動ロボットの目標地点は、その光源に最も近い壁である。タスクは、箱を壁に接触するまで押すことにより達成される。なお、本論文における実験では、障害物は設置していない。我々の行動選択手法は、障害物の有無に依存しないため、障害物がないことは、本研究を本質的に制約するものではない。

小型移動ロボットとして、Khepera (Fig.2) を用いる。Khepera は、モトローラ 68331 (16MHz)、RAM 256Kbyte、ROM 512Kbyte を搭載し、C によりプログラム可能である。Fig.3 に示すように、Khepera は、インクリメンタルエンコーダ付き DC モータ (ロボットの移動 1mm あたり 10 パルス) を 2 つ、距離を計測できる赤外線近接センサと光の強度を測る光センサが一体になったセンサを 8 つ装備している。なお、バッテリー持続時間は、約 20 分である。しかし、近接センサの精度は悪く、また近接センサはせいぜい数センチ先の物体にしか反応しない。一方、光センサも精度は悪いが、環境の任意の位置において、光源に反応するようになっている。また、箱は、光を十分に通す透明なプラスチック板で作られている。よって、移動ロボットは、箱越しに光源方向を計測可能である。

2.2 環境に対する仮定

通信を行わず、かつ局所的な情報だけを基に移動ロボットが行動するためには、環境に対する仮定をおく必要がある。本研究では、以下のような仮定を環境に対し設定している。後述す

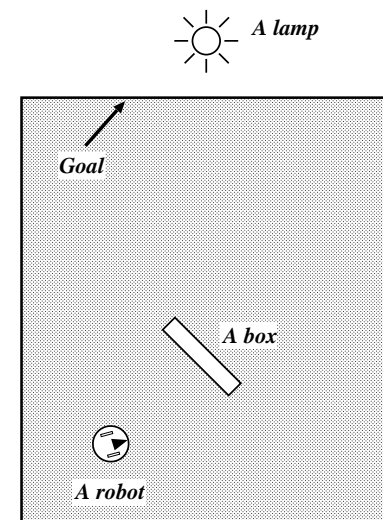


Fig. 1 An experimental environment

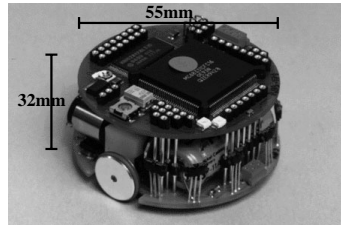


Fig. 2 Khepera

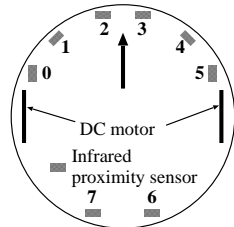


Fig. 3 Sensor positions

るように、これらの仮定は、主に他のロボットの存在を判定する際に使われる。

AS-1: ロボット以外に移動物体は存在しない。

AS-2: ロボットが、押ししても動かない物体(壁, 重い箱)を押すとき, 車輪は回転しない。つまり, ロボットの車輪はスリップしない。

2.3 状況の定義

マルチ移動ロボットの箱押しにおける動的環境を記述するために、我々は他のロボットの存在の有無とタスクの難易度という2つのパラメータを用いる。他のロボットの存在の有無は、環境に他のロボットがいるか否かを意味し、タスクの難易度は、一つのロボットが押しして移動させることができる箱が存在するか否かを意味する。これらのパラメータにより、ロボットの故障、ロボットの導入/削除、重い/軽い箱の導入/削除などによる環境の変化を記述できる。

この2つのパラメータは、それぞれ2値をとるので、論理式 M と T とそれらの否定で2つのパラメータを表すことにする。 M は、環境に他のロボットがいる状態を表し、 $\neg M$ は、他のロボットがいない状態を表す。また、 T は、環境に一つのロボットで押すことのできる箱がある状態を表し、 $\neg T$ は、そのような箱がない状態を表す。

本研究では、 $\{M \wedge T, \neg M \wedge T, M \wedge \neg T, \neg M \wedge \neg T\}$ の4つの論理積で動的環境を記述し、それぞれの環境を状況と呼ぶ。また、状況を記述するパラメータを状況パラメータと呼ぶことにする。ここで、状況とは、「ロボットが一つの行動集合を用いて行動することでタスクを達成可能である環境のクラス」を意味する。よって、各移動ロボットは、自分が今置かれている状況のために設計された一つの状況行動集合を用いて、タスクを達成可能である。

以下にそれぞれの状況とその状況に適した行動を示す。これらの行動は、次節で、IF-THEN 規則で具体的に記述される。

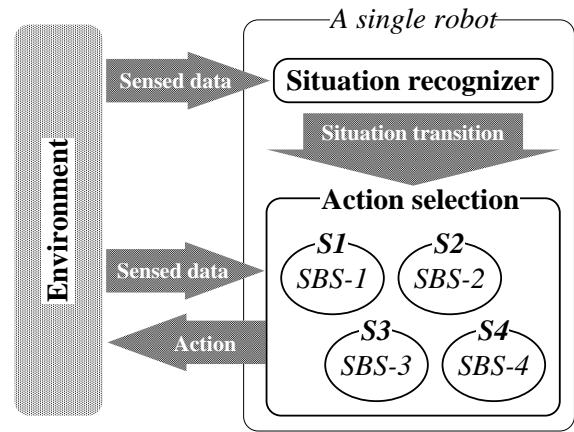


Fig. 4 Adaptive action selection architecture

注意して欲しいのは、本研究では、それぞれのロボットが、明示的通信なしに、自分自身で得ることのできる局所的情報から自分の置かれている状況を判断することである。よって、その判断は必ずしも大局的に正しいものとは限らない。

- 状況 $S1 = \neg M \wedge T$ (環境にロボットが1台だけで、かつ1台で押せる箱がある状況): 1台で箱押しを行う。
- 状況 $S2 = M \wedge T$ (環境に複数のロボットでいて、かつ1台で押せる箱がある状況): 協力の必要がないので、それぞれのロボットが独立に箱を押す。この行動により、複数の箱がある場合は、それぞれの箱を個々のロボットが並列に押ししていくことが可能になり、効率向上が期待できる。
- 状況 $S3 = M \wedge \neg T$ (環境に複数のロボットでいて、かつ1台で押せる箱がない状況): 1台では箱を押せないで、複数ロボットが群をなして協力することにより一つの箱を押す。
- 状況 $S4 = \neg M \wedge \neg T$ (環境にロボットが1台だけで、かつ1台で押せる箱がない状況): この状況では、箱押しは不可能である。しかし、大局的には、他の状況かも知れない[†]、また時間が経つと環境が変化して大局的に他の状況に遷移するかもしれない。よって、ロボットは、他のロボットや単独で押せる箱が見つかるまで徘徊する。

2.4 行為選択アーキテクチャ

本研究では、すべてのロボットは、均質に Fig.4のようなアーキテクチャをもつ。状況認識器 (situation recognizer) が、常にセンサからのデータを監視しており、現在ロボットがどの状況にあるのかを決定する。そして、現在の状況に適した行動の集合である SBS (状況行動集合) を活性化させる。ロボットは、活性化された SBS に基づき実際に行動する。

状況の認識と状況遷移

動的環境に適応するために、ロボットは自分自身で現在の状況を認識し、適切な SBS を変更しなければならない。よって、そのロボットの状況認識器は、 M, T の真偽を決定するために

[†]つまり、本当は環境に他のロボットや単独で押すことのできる箱があるのにもかかわらず、まだそれらを見つけていないだけの場合が考えられる。

絶えず環境からのデータを監視する． M , T をどのように決定するかを以下に示す．なお，環境認識器は，後述する行動のような IF-THEN 形式ではなく，普通の手続き的プログラムで書かれている．

- M の判定条件 C_M : 状況認識器は，ロボットが止まっているときに，センサデータの変化がないかを調べる．もし変化があれば，環境内に他のロボットが存在すると判断し， M が成り立つことになる．この判定には，§2.2の仮説 AS-1 が使われている．なお，他のロボットに押された物体は， M の判定に影響を与えない．
- $\neg M$ の判定条件 $C_{\neg M}$: 最後に M が成り立ってから，一定時間 t_M の間に C_M が全く満たされなかった場合に， $\neg M$ が成り立つ．
- T の判定条件 C_T : ロボットが物体に接触して，かつ車輪が回転している場合は，1 台のロボットで押せる箱があると判断し， T が成り立つ．この判定条件には，§2.2の AS-2 が使われている．
- $\neg T$ の判定条件 $C_{\neg T}$: ロボットが，押すことのできない物体に連続して t_T 回以上ぶつかった時に， $\neg T$ が成り立つ．このとき，ロボットは，連続して衝突した押せない物体が同一物体であることを判定できない．よって，ある程度周囲を探索した結果，その間衝突したものがすべて押すことができなかった場合に，1 台で押せる箱はないと判定することになる．

3. SBS : 状況行動集合

我々は，行動と呼ばれるセンサ情報から行為への直結したマッピングを IF-THEN 規則で記述し，それに従って移動ロボットを制御する行動ベースアプローチ [4] を用いる．この方法は，プランニングなどを用いる方法に較べると，高精度かつ最適な制御することは難しい反面，センシング行為のサイクルを迅速に行うことができ，環境の変化に対しても追従性が高い．実際，これまでの我々の研究 [24] における経験からも，壁沿い行動などの基本的タスクについては，行動ベースアプローチで容易に実装可能なことがわかっている．また本研究では，行動が状況によりモジュール化されているため，個々の状況に対し，単純な行動の集合を記述できればよい．

行動は，“IF 状態 THEN 行為”の IF-THEN 規則で記述される．ここで，状態とは，センサ情報によって直接的に決定される記述であり，本研究では，センサ情報の値の範囲で定義される．また，行為とは，モータコマンドである．以下で，我々は，先に示した各状況に適切な行動の集合を，具体的に IF-TEHN 規則で記述していく．そのような行動の集合は，状況行動集合 *SBS* と呼ばれ，*SBS*- i は，状況 S_i に適切な状況行動集合を意味する．

なお，設計方針として，設計者がトップダウンに状況を分類して，それぞれの状況に適切な行動集合 *SBS* を設定していく．よって，異なる状況の *SBS* に，同じ行動が含まれることもあり，また *SBS* 間に包含あるいは等しい関係が生じることもありえる．しかし，状況認識器により状況が決定され，ある時点

で一つの *SBS* しか活性化されないもので，*SBS* 間の行動の重複は，タスク達成にとって障害にはならない．

また，ここで与えられた行動により，一般に困難とされる「点接触で，箱との接触点の位置を測定または推測，制御することなしに，一台のロボットが物理的に確実かつ安定的に，箱を直進あるいは特定な方向に回転させる」ことが必ずできるわけではなく，失敗する場合もある．さらに，本研究における行動ベースの箱押しは，タスクが達成されることを保障する解析的証明をもっていない．よって，解析的研究 [9] [5] との比較は難しいと考える．

3.1 状態と行為の記述

まず，センサをカテゴライズする．*forward-sensors* と *back-sensors* は，それぞれ Fig.3のセンサ 1~4 とセンサ 6, 7 を表す．また，*left-sensor* と *right-sensor* は，Fig.3のセンサ 0 とセンサ 5 を意味する．これらのセンサを用いて，以下のように状態と行為が定義される．行動を実行するために，明示的通信は一切必要としない点に注意して欲しい．

状態

- *forward/back/left/right-object* : ロボットの前方 / 後方 / 左方向 / 右方向のロボットから 20mm 以内[†]に物体がある状態．つまり，ロボットの *forward/back/left/right-sensors* に反応がある状態である．
- *forward/back/left/right-light* : ロボットの *forward/back/left/right-sensors* が，最大の光強度をもつ状態．
- *no-light* : 全ての方向のセンサの光強度がほぼ同じ状態．
- *no-rotation* : モータに回転するコマンドを送っているにも関わらず，エンコーダでモータの回転が計測されない状態．

行為

- *direction-change* : ロボットが，180° 回転する行為．
- *push-clockwise/counterclockwise* : ロボットが押すことにより，物体（箱）を時計回り / 反時計回りに回転される行為．
- *push-straight* : 真っ直に箱を押す行為．
- *turn-left/right* : その場での左 / 右への回転．
- *go-ahead* : 直進．
- *stop* : 停止．

以上の状態と行動をそれぞれ IF-THEN 規則の条件部と結論部として，以降でそれぞれの状況に適した *SBS* を記述していく．なお，同一 *SBS* 内で，複数の行動が適用可能な場合に，行動間の競合が生じる．そのときは，下記の規則により競合解消を行い，実行すべき行動を 1 つに決定する．

- (1) より多くの条件をもつ行動を優先する．
- (2) 番号の若い行動を優先する．

また，以降で設定される行動により，原理的に起こり得るすべての環境の状態をカバーできる保証はない．もしどの行動も適用できないときは，移動ロボットは単に直進する．この行動は，全状況で共通とする．

[†]Khepera の赤外線近接センサは，20mm 前後までの物体にしか反応しない．

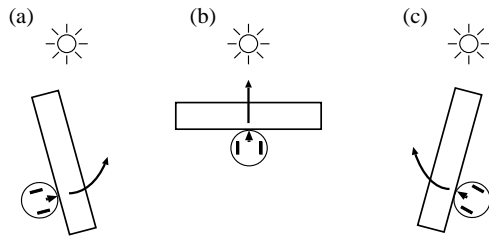


Fig. 5 Executions of B-3 ~ B-5

3.2 SBS-1: 単独ロボットの箱押し行動

状況 $S1$ では、1 台で箱を目標地点まで押すために、以下の行動を用いる。Fig.5が、B-3~B-5 を実行したときのロボットの挙動を示している。

B-1 IF \neg forward-object \wedge \neg left-object \wedge \neg right-object
THEN go-ahead .

(前方、左右ともに何もなければ、前進する。)

B-2 IF forward-object \wedge no-rotation
THEN direction-change .

(前方に物体があり、車輪が回転しなければ、180 度方向転換する。この行動は、ロボットが壁に衝突したときに実行される。)

B-3 IF forward-object \wedge left-light
THEN push-clockwise.

(前方に物体があり、左方向に光源があるとき、物体を押して時計周りに回転させる。Fig.5(a) が、この行動の挙動を示す。)

B-4 IF forward-object \wedge forward-light
THEN push-straight .

(前方に物体があり、前方に光源があるとき、物体をまっすぐ押す。Fig.5(b) が、この行動の挙動を示す。)

B-5 IF forward-object \wedge right-light
THEN push-counterclockwise .

(前方に物体があり、右方向にランプがあるとき、物体を押して反時計周りに回転させる。Fig.5(c) が、この行動の挙動を示す。)

3.3 SBS-2: 分散箱押し行動

状況 $S2$ のための状況行動集合 $SBS-2$ は、 $SBS-1$ と基本的には同じである。ただし、複数のロボットが存在するので、そのロボット間の干渉を扱う必要がある。まず、 $SBS-1$ をそのまま状況 $S2$ に適用した結果、Fig.6のような望ましくない干渉が観察された。

Fig.6(a) は、2 台のロボットが、同じ箱を両側から反対方向に押ししまい、その結果その物体は押ししても動かない壁が重い箱であると判定し、2 台とも箱から離れてしまう場合である。本来なら目標地点に背を向けている 1 台だけが箱から離れ、もう 1 台はそのまま箱を押すことが望ましい。Fig.6(b) は、1 台のロボット (R2) が別のロボット (R1) を押している場合である。この状態は、2 台で直接箱を押すよりも効率が悪いので避けるべきである。Fig.6(c) は、箱を押している 2 台のロボッ

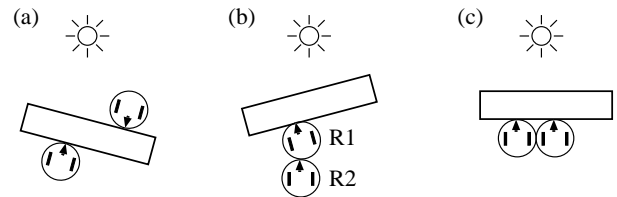


Fig. 6 Interaction between robots

トが接触している場合である。この状態は、片方のロボットがもう一方のロボットに接触しながら背後に移動してしまい、結果的に Fig.6(b) の状態になる場合が多い。

これらの干渉を避けるために、以下の行動を $SBS-1$ に追加して、 $SBS-2$ を B-1~B-9 で構成した。Fig.6(a) の干渉に対し、B-6 を適用することにより、目標地点の反対を向いているロボットが方向転換し、目的地点方向を向いている他方のロボットは箱を押し続けることができる。Fig.6(b) の干渉に対し、B-7 を適用することにより、Fig.6(b) のロボット R1 は背後に物体を観測した場合は停止する。すると、ロボット R2 は停止した R1 を壁と認識し、去って行くことにより干渉をさけることができる。また、Fig.6(c) の干渉に対して B-8 と B-9 を適用することにより、ロボットは左または右に物体がセンシングされたときに、それと反対方向に少し回転する。そして、その直後に B-4 により直進して、隣のロボットと離れることができる。この行動は、[17] における鳥の群の行動を参考に考えたものである。

B-6 IF forward-object \wedge back-light
THEN direction-change.

(前方に物体があり、後方に光源があるとき、180 度方向転換する。)

B-7 IF forward-object \wedge back-object
THEN stop.

(前方に物体があり、後方にも物体があるとき、停止する。)

B-8 IF forward-object \wedge forward-light \wedge left-object
THEN turn-right.

(前方に物体と光源があり、左に物体があるとき、右回転する。)

B-9 IF forward-object \wedge forward-light \wedge right-object
THEN turn-left.

(前方に物体と光源があり、右に物体があるとき、左回転する。)

状況 $S2$ において、複数の押ししている箱が衝突することにより干渉する場合が起こり得る。その場合、もしそのまま光源方向に押し進めることが可能なら、ロボットは押し進める。また、不可能なら、行動 B-2 が実行され、箱を押していたロボットは、方向転換して箱から離れていく。ただし、この段階では、 t_T 回以上連続して押せない物体にぶつかっていないため、 T は否定されず、ロボットは 1 台でも押せる他の箱を探しに行くことになる。そして、接触したまま残された複数の箱は、1 台では押せない 1 つの箱として扱われる。

3.4 SBS-3: 群れによる箱押し行動

状況 S_3 では、1 台のロボットでは、箱押しをすることができない。よって、ロボットは群をなして箱を探し、見つければ複数台のロボットが協力して 1 つの箱を押しことにする。一般に群といっても、列、円形等のさまざまな形状が考えられるが、本研究では、ロボット間の干渉が少ないこと、構成が容易であることから列の群を採用する。

群のための状況行動は、ロボットが他のロボットを認識する必要があるため、これまでの行動より複雑になる。よって、新たに以下に示すような状態 *forward/right/left/back-robot*, *forward/back-robot-leaving* と、行為 *following*, *side&push* を導入する。

状態

- *forward/right/left/back-robot*: 他のロボットが、前方、右方向、左方向、または後方に感知されたことを意味する。§ 2.4 の M の判定条件 C_M を用いて決定される。
- *forward/back-robot-leaving*: 前方、または後方に感知されていた他のロボットが感知されなくなったことを意味する。

行為

- *following*: 他のロボットが感知された時、あるいは感知されていた他のロボットが感知されなくなった時に、その方向へ進む行為。
- *side&push*: 列をなしていたロボットが、Fig.7の波線矢印のように横に出て、協力して箱を押し一連の行為。ロボットは、ランダムに左右どちらかに方向転換し、一定距離進んでから元の進行方向に向かって、何かに衝突するまで直進する。

上記の付加的な状態と行為を用いて、状態 S_3 の状況行動集合 *SBS-3* は、以下の 3 つの部分から構成される。

(1) 群の形成

基本的に、群は、まず他のロボットを探すために徘徊し、見つけたらそのロボットに追従することにより形成される。以下の行動 B-10 を *SBS-1* に追加することにより、このような群の形成が実現される。

B-10 IF *left-robot* \vee *right-robot*

THEN *following*.

(左方向、または右方向にロボットが感知されたとき、それを追いかける。)

(2) 列の維持

列の群を維持するために、列の先頭ロボットとそれ以外のロボットについて、以下の行動を設計した。基本的な行動として、先頭ロボットは、背後に何かを感知したときに前進し、背後に何も感知しなくなったときに停止する。また、先頭でないロボットは、前方のロボットが感知されなくなったときに、それを追いかけて、感知されると停止する。

先頭のロボットのための行動

B-11 IF \neg *forward-robot* \wedge *back-robot*

THEN *go-ahead*.

(前にロボットが感知されず、かつ後ろにロボットが感知されたとき、前進する。)

B-12 IF \neg *forward-robot* \wedge *back-robot-leaving*

THEN *stop*.

(前にロボットが感知されず、かつ後ろの感知されていたロボットが感知されなくなったとき、停止する。)

先頭以外のロボットのための行動

B-13 IF *forward-robot-leaving*

THEN *following*. (感知されていた前方のロボットが感知されなくなったとき、それを追いかける。)

B-14 IF *forward-robot*

THEN *stop*.

(前方にロボットが感知されているとき、停止する。)

(3) 群での箱押し

列をなしているロボット群が箱を見つけたとき、それらは、Fig.7のように協力して箱押しを行う必要がある。また、群が壁にぶつかったときは、列をなしたまま壁から離れていくのが望ましい。これらの挙動は、以下の行動により実現される。

先頭のロボットのための行動

B-15 IF *forward-object* \wedge *back-robot* \wedge *forward-light*

THEN *push-straight*.

(前方に物体があり、後ろにロボットが感知されていて、かつ前方に光源があるとき、真っ直ぐに箱を押し。)

B-16 IF *forward-object* \wedge *back-robot-leaving*

THEN *following*.

(前方に物体があり、後方の感知されていたロボットが感知されなくなったとき、そのロボットを追いかける。)

先頭以外のロボットのための行動

B-17 IF *forward-robot* \wedge *back-robot-leaving*

THEN *following*.

(前方にロボットが感知され、後方のロボットが感知されなくなったとき、その後方のロボットを追いかける。)

B-18 IF *forward-robot* \wedge \neg *forward-light*

THEN *direction-change*.

(前方にロボットが感知され、前方に光源がないとき、180度方向転換する。)

B-19 IF *forward-robot* \wedge *forward-light*

THEN *side&push*.

(前方にロボットが感知され、前方に光源があるとき、横に出て箱に接触する。)

最終的に、*SBS-3* は、*SBS-1* と B-10~B-19 から構成される。また、状況 S_3 で個々のロボットは自分が先頭なのか否かを判定せずに、単に *SBS-3* の行動を実行する。上では、説明の都合上、「先頭のロボットのための行動」と「先頭以外のロボットのための行動」を分けて示したが、これらは *SBS-3* の中では区別されていない。

3.5 SBS-4: 状況遷移の行動

状況 S_4 においては、ロボットは他のロボットも、単独で押せる箱も見つけていないため、箱押しタスクを達成することは不可能である。しかし、状況 S_4 にいるという認識は、あくまでロボットが自分で得られる局所的な情報から判断しているに過ぎず、大局的に見れば本当はロボットは他の状況にいるかも知

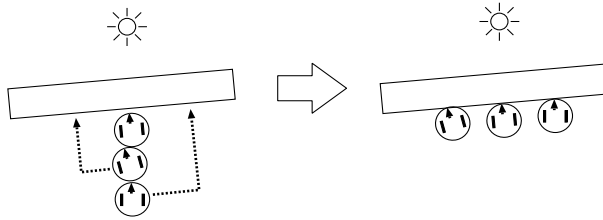


Fig. 7 Cooperative box-pushing

れない。また、時間とともにロボットや軽い箱の追加などの環境変化が起こり、大局的状況が他の状況に変化する可能性もある。よって、他のロボットや軽い箱を見つける目的と、状況の変化を待つ意味もこめて、状況 S_4 ではロボットを徘徊させることにする。具体的には、状況が変化するまで $SBS-1$ を用いて徘徊する。

4. 複数移動ロボットによる実験

我々は、ここまで述べてきた適応的行動選択メカニズムを、4台の Khepera それぞれに実装した。C言語を用いたプログラミングをホスト計算機である IBM-PC 互換機で行ない、そのプログラムを Khepera にダウンロードした後は、それぞれの移動ロボットは自律的に行動する。Khepera のセンサは、個々の特性にばらつきがあるので、プログラムによりそれらを補正した。

また、実験において、§2.4で述べた状況パラメータ t_M と t_T は、経験的にそれぞれ 300 秒と 10 回に設定した。すべての実験において、目標地点は、右側の壁であり、ロボットは箱を右側の壁の方向に、その壁と接触するまで押していく。行為の実行を含む行為選択の 1 周期は、約 100m 秒である。実際の実装においては、環境認識器と行為選択は、ハードウェア的に並列になっているのではなく、Khepera 上の一つのマイクロプロセッサにより、時分割で処理される。

我々の方法の有効性を調べるため、静的環境と動的環境の両方において実験を行った。まず、最初に状況の変化のない静的環境において実験を行い、次に状況変化を伴う動的環境における実験も行った。

4.1 それぞれの状況における実験

状況 S_1 と S_2 での実験

まず、単独で押せる箱一つと単独の移動ロボットを環境にランダムに配置して、実験を行った。この環境は、大局的には、状況 S_1 である。しかし、最初ロボットは、自分は状況 S_4 にいると認識して徘徊を行う。そのうち、箱を見つけ、状況 S_1 に認識を変更し、その箱を光源方向に押していく。その箱押しの軌跡を Fig.8に示す。図からわかるように、ロボットは状況 S_1 において箱押しを達成している。

また、環境に 2 台のロボットと単独で押せる 2 つの箱があるような状況 S_2 では、状況 S_3 のように 2 台のロボットが協力して一つの箱を押すのではなく、それぞれのロボットが独立して箱を押すというより効率的な行動をとることが確認された。Fig.9が、その軌跡を示している。

状況 S_3 と S_4 での実験

一つの単独では押せない箱と、4台のロボットを環境にランダムに配置して、状況 S_3 を構成して実験を行った。その結果、時間の経過とともに 4台のロボットは、それぞれが独立に環境が状況 S_3 であることを認識し、群をつくり始めた。Fig.10(a)~(d)にその様子を示す。そして、群で列をなして行動し、箱を見つけて協力して箱押しを行うことが確認された。Fig.10(e)~(h)が、行為 *side&push* を実行している様子である。

次に、状況 S_4 における実験を行った結果、ロボットは他のロボット、あるいは単独で押せる箱を探して徘徊を行った。

4.2 動的環境への適応

環境にロボットや単独では押せない箱を追加したり、取り除いたりすることにより、状況を変化させてロボットの挙動を観察した。その結果、 $S_1 \sim S_4$ の任意の 2 つの状況間での変化において、それぞれのロボットが独立に状況を遷移させ、適切な SBS を活性化することが確認できた。なお、複数のロボットが同一の環境で行動するとき、個々のロボットの状況遷移は非同期におこり、すべてのロボットの認識した状況が同一である保証はない。

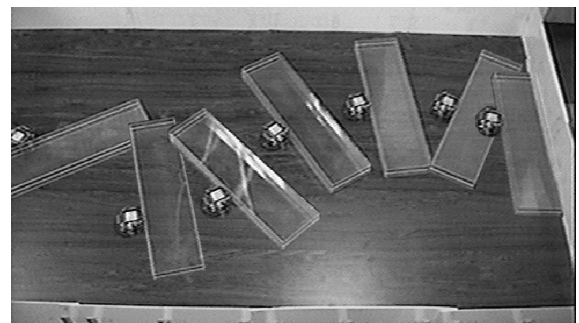
一つの例を示す。Fig.11は、単独で押せない箱の存在しない環境において、2台の移動ロボットが遭遇する前後の軌跡である。その 2台のロボットは、現在の状況は、状況 S_2 と認識しているので、 $SBS-2$ が活性化されている。よって、Fig.11のように他のロボットと遭遇しても、互いに離れていってしまう。

その後、単独では押せない箱を環境に追加した。すると、時間の経過により、その重い箱が観測されて 2台のロボットは、状況 S_3 へ遷移する。よって、 $SBS-3$ を活性化して、群をつくって行動しようとする。このような状況において、2台のロボットが遭遇する前後の行動の軌跡が、Fig.12である。先程の状況 S_2 と認識されている場合とは異なり、同じように遭遇しても、今度はお互いに近づいて群になるってことが見てとれる。

また、すべての実験を通して、どの行動も適用できないような状態が環境において生じることが、観測されなかった。このことから、我々の行動の設定は、妥当であったと考えられる。

5. 考 察

本節では、我々の提案した行為選択メカニズムの有用性、そして限界について考察していく。

Fig. 8 Box-pushing in S_1

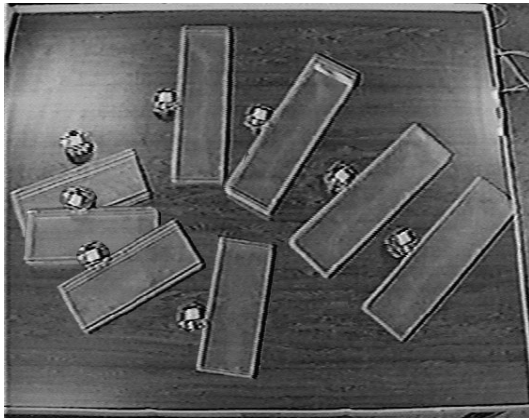


Fig. 9 Box-pushing in S2

5.1 枠組としての一般性

本研究は、マルチロボットシステムのタスクを「箱押し」に限定して適応的行動選択メカニズムを設計し、実ロボットに実装している。しかし、本研究で提案している枠組は、対象とするタスクと環境において、設計者が状況、状況パラメータ、状況行動集合を記述できれば適用可能である。ただし、本質的に、他のロボットとの通信や大局的な情報を必要とするタスクは扱えない。

5.2 問題点と課題

タスク遂行可能性

当然ながら、ロボット及び箱の任意の初期位置に対して、箱押しを達成できるわけではない。箱押しの達成できない典型的な例を以下に示す。

- 箱が壁に密着した場合：ロボットは箱を押すだけであり、引くことはできないため、壁に密着した箱に対処できない。
- 箱あるいは他のロボットの探索の失敗：箱や他のロボットの探索が偶然に依っているため、長時間経ってもそれらを見つけれない場合がある。例えば、大局的には、状況 $S1$ 、 $S4$ 、 $S3$ のいずれかであるにも関わらず、1台で押せる箱または他のロボットを見つけることができず、局所的には $S4$ であり続ける場合は、箱押しを達成できない。

上記の場合には、アクチュエータの機能の追加、状況行動の追加・修正などによりある程度は対応できると考えられる。

状況判断の非同一・非同期性

個々のロボットが独立に状況の判定を行ない、明示的通信による個々の状況の共有もしないため、複数の移動ロボットが同期して状況判定を行なうわけではない。また、いつも全てのロボットの認識している状況が同じである保障もない。このことが、システム全体にどのような影響をあたえるかは今のところ明らかでないが、興味深いテーマと考えられる。

学習の必要性

本研究は、設計者が、動的環境においてあらかじめ予想される状況を調べ上げ、それぞれの状況に適切な行動の集合、状況の記述、そして状況判定の手続きをすべて記述するという、トップダウン的なアプローチである。よって、あらかじめ予想

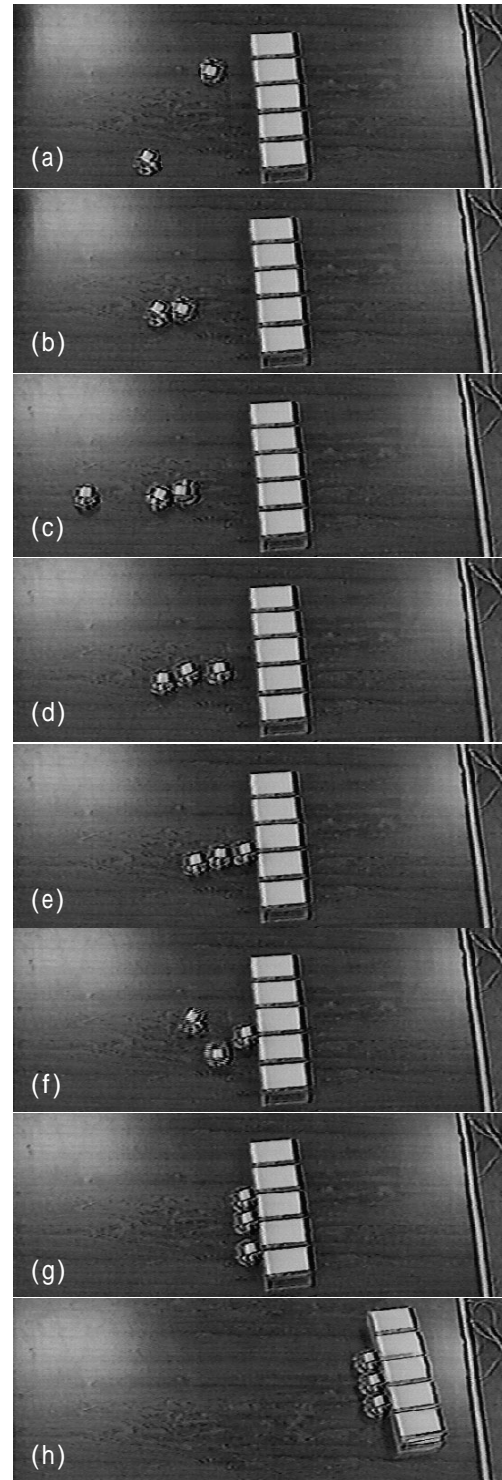


Fig. 10 Box-pushing in S3

されていなかった状況、つまり用意したいずれの行動集合を用いても適切な行動をとれない状況にロボットが遭遇した場合には、ロボットはタスクを達成できないかも知れない。この意味で、本研究のアプローチは、時間的に変化するという意味での動的環境には適応していると言えるが、エージェントの環境モ

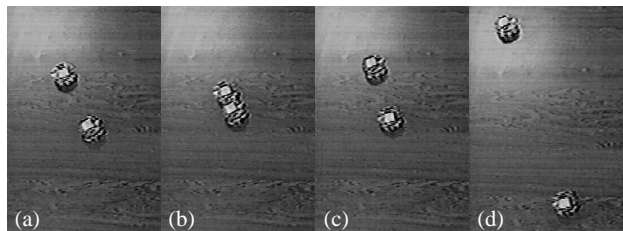


Fig. 11 Encounter of robots in S2

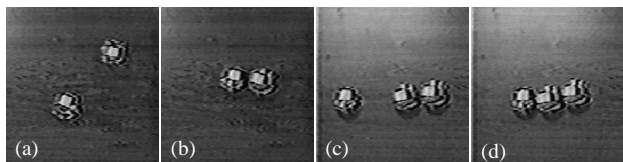


Fig. 12 Encounter of robots in S3

デルでは予測不能であるエージェントにとっての動的環境 [23] には、適応できてない。

この問題に対処するためにまず考えられるのが、ロボットに学習機能を持たせることである。本研究では、状況の獲得、状況行動の学習、状況行動間の競合解消の学習、状況判別条件の学習など様々なレベルでの学習が考えられる。ただし、実機を動かしながらのオンライン学習を目指した場合、Q 学習、ニューラルネット学習、帰納学習等の学習アルゴリズムを用いることは、その学習効率の悪さのため有効ではないと考える。実ロボットのオンライン学習に適した学習効率のよい学習アルゴリズムを開発することは、今後の課題である。

環境変化への追従性

我々の提案した行為選択メカニズムは、環境変化に追従することが実験的に確認された。しかし、大局的に状況が変化した場合に、状況認識器がその状況変化を認識するため要する時間は、環境の広さに大きく依存するが、我々の実験環境では、5～10 分の場合が多く、かなり遅いものだった。よって、この時間よりも速い周期で起こる環境変化には追従できない。この追従の遅さは、§ 2.4 で述べた移動ロボットが用いる状況の判定条件が、局所的な情報に基づいたアドホックなものであることが原因の一つである。しかし、明示的通信をせずに局所情報だけを用いてマルチロボットを制御するという本研究の方針の下では、この状況認識の遅さはやむを得ないと考える。

6. ま と め

マルチロボットによる箱押しのために、動的環境に適応できる行為選択メカニズムを提案した。本手法では、他のロボットの存在の有無とタスクの難しさの 2 つの状況パラメータを用いて動的環境が記述される。そして、記述された個々の状況に対し、適切な行動のモジュールを設計した。常に環境を監視して、現在おかれた状況を個々のロボットが認識することにより、適切な行動モジュールを切替え、動的環境への適応が実現される。

最後に、4 台の小型の移動ロボット上に、我々の提案した行為選択メカニズムを実装し、静的環境及び動的環境において実験を行うことにより、我々のアプローチの有効性を確認した。謝辞 本研究は、日本学術振興会 未来開拓学術研究推進事業研究プロジェクト「生物的適応システム」の支援のもとに行なわれた。記して、感謝いたします。

参 考 文 献

- [1] Y. Aiyama, M. Hara, T. Yabuki, J. Ota, and T. Arai: "Cooperative Transportation by Two 4-legged Robots with Implicit Communication", *Distributed Autonomous Robotics Systems* 3, pp. 131–140, Springer, 1998.
- [2] H. Asama, A. Matsumoto and T. Ishida: "Design of an Autonomous and Distributed Robot System", *Proc. of The 1989 IEEE/RSJ Int. Workshop on Intelligent Robots and Systems*, pp. 283–290, 1989.
- [3] R. Beekers, O.E. Holland, and J.L. Deneubourg: "From Local Actions to Global Tasks: Stigmergy and Collective Robotics", *Artificial Life IV*, pp. 181–189, 1994.
- [4] R.A. Brooks: "A Robust Layered Control System for a Mobile Robot", *IEEE Transaction on Robotics and Automation*, vol. 2, no. 1, pp.14–23, 1986.
- [5] B.R. Donald, J. Jennings and D. Rus: "Analyzing Teams of Cooperating Mobile Robots", *Proc. of The 1994 IEEE Int. Conf. on Robotics and Automation*, pp. 1896–1903, 1994.
- [6] K. Kosuge and T. Osumi: "Decentralized Control of Multiple Robots Handling and Object", *Proc. of The 1996 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 318–323, 1996.
- [7] C.R. Kube and H. Zhang: "The Use of Perceptual Cues in Multi-Robot Box-Pushing", *Proc. of The 1996 IEEE Int. Conf. on Robotics and Automation*, pp. 2085–2090, 1996.
- [8] P. Maes: "Learning Behavior Networks from Experience", *Proc. of The 1st European Conf. on Artificial Life*, pp.48–57, 1991.
- [9] M.T. Mason and J.K. Salisbury, Jr: *Robot Hands and the Mechanics of Manipulation*, MIT Press, 1985.
- [10] M. J. Mataric: "Learning in Multi-Robot Systems", *Adaption and Learning in Multi-agent Systems*, Springer, pp. 152–163, 1996.
- [11] M. J. Mataric, M. Nilson, and K. T. Simsarian: "Cooperative Multi-Robot Box-Pushing", *Proc. of The 1995 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 556–561, 1995.
- [12] N. Miyata, J. Ota, T. Arai, E. Yoshida, D. Kurabayashi, J. Sakaki and Y. Aiyama: "Cooperative Transport with Re-grasping of Torque-Limited Mobile Robots", *Proc. of The 1996 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 304–309, 1996.
- [13] F.R. Noreils and R.De Nozay: "An Architecture for Cooperative and Autonomous Mobile Robots", *Proc. of The 1992 IEEE Int. Conf. on Robotics and Automation*, pp. 2703–2710, 1992.
- [14] T. Ohko, K. Hikaki and Y. Anzai: "Learning to Reduce Communication Cost on Task Negotiation among Multiple Autonomous Mobile Robots", *Adaption and Learning in Multi-Agent Systems*, Springer, pp. 177–190, 1996.
- [15] H. Osumi: "Cooperative strategy for multiple mobile manipulators", *Proc. of The 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 554–559, 1996.
- [16] L.E. Parker: "ALLIANCE: an architecture for fault tolerant multirobot cooperation", *IEEE Transaction on Robotics and Automation*, vol. 14, no. 2, pp. 220–240, 1998.
- [17] C. W. Reynolds: "Flocks, Herds, and Schools: A Distributed Behavioral Model", *ACM Computer Graphics*, vol. 21 No. 4, pp.25–34, 1987.

- [18] 齋藤, 山田: “局所情報を用いた複数の実ロボットによる箱押し行動の実現”, 第15回日本ロボット学会学術講演会, pp. 883-884, 1997.
- [19] S. Sen and M. Sekaran and J. Hale: “Learning to Coordinate Without Sharing Information”, Proc. of The 12th National Conf. on Artificial Intelligence, pp. 426-431, 1994.
- [20] D. J. Stilwell and J. S. Bay: “Toward the development of a material transport system using swarms of ant-like robots”, Proc. of The 1993 IEEE Int. Conf. on Robotics and Automation, p. 766-771, 1995.
- [21] H. Sugie, Y. Inagaki, S. Ono, H. Aisu, and T. Unemi: “Placing objects with multiple mobile robots – mutual help using intention inference”, Proc. of The 1995 IEEE Int. Conf. on Robotics and Automation, pp. 2181-2186, 1995.
- [22] Z. Wang, E. Nakano, and T. Matsukawa: “Realizing cooperative object manipulation using multiple behavior-based robots”, Proc. of The 1996 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 310-317, 1996.
- [23] 山田 誠二: 適応エージェント, p. 43, 共立出版, 1997.
- [24] S. Yamada and M. Murota: “Unsupervised Learning to Recognize Environments from Behavior Sequences in a Mobile Robot”, Proc. of The 1998 IEEE Int. Conf. on Robotics and Automation, pp. 1871-1876, 1998.

山田 誠二

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。同年大阪大学基礎工学部助手。1991年同大学産業科学研究科講師。1996年4月より、東京工業大学大学院総合理工学研究科助教授、現在に至る。工学博士。人工知能、特に、プランニング、WWWでの知的情報

収集、ロボット学習に興味をもつ。人工知能学会、情報処理学会、AAAI、IEEE各会員。(日本ロボット学会正会員)

齋藤 淳也

1973年2月4日生。1996年東京理科大学理工学部経営工学科卒業。1998年東京工業大学総合理工学研究科修士修了。同年より(株)富士通に入社、現在に至る。