

# Tiny H8/3694Fによる接触通信システムのモジュール化

— multi-CPU化による処理能力の向上へ —

高井 裕紀\* 高田 宗樹\* 平田 隆幸\*

**Module of the contact communication system by Tiny H8/3694F  
— Improvement of the information processing ability by multi-CPU —**

**Hiroki TAKAI\*, Hiroki TAKADA\* and Takayuki HIRATA\***

(Receive January 23, 2014)

Swarm intelligence as an autonomous distributed system has been attracted much attention of researchers in various fields. A swarm robot is one of the research fields of the swarm intelligence. However, to design swarm robot is difficult because of too much degree of freedom. The social insects, such as ants, are good examples of sophisticated autonomous distributed system in nature. A cooperative behavior of ant colony gives us a hint in designing a swarm robot. We design an ant-type small robot as a prototype general purpose swarm robot. The behavior of ants is too complicated one to simulate by a single micro CPU. We try to solve this hard problem by using the multi-CPU system instead of the use of high-performance CPU. The some functions of the ant-type small robot are modularized by adopting sub-CPU system. In this study, we focus on modularization of the contact communication system.

**Key Words :** multi-CPU, Swarm Robot, Ant, Communication module.

## 1. はじめに

複数のロボットエージェントを用いて、ある仕事をおこなうシステムを設計しようとしたとき2通りの方法がある。1つは、情報を1か所に集め、中央集権的に制御をおこなう集中管理システムである。この方法では各ロボットへの作業の割り当てや相互調整などを一箇所でまとめておこなうことができるため、統一的な方針を立てることができ、作業効率も高く、完成した仕事の質もよいと言われている。しかし、情報の欠落や遅延の発生などによる環境変動に対しては柔軟な制御が困難である。また、システムが巨大化したときに中枢への負担が大きくなる。近年のコンピュータは高速化しているが1台ですべてを管理する集中管理システムでは制御できる系の大きさに限界がある。また、情報伝達が不完全な場合には、集

中管理システムはうまく機能しない場合がある。

一方、別の方法として挙げられるのが自律分散システムである。自律分散システムはエージェントが並列分散的に機能しながらシステム全体として当初の目的に合うように振舞うシステムである。ひとつひとつのエージェントが自律性をもち、環境の認識や他のエージェントとの相互作用を通じて状況を判断し、自分自身の行動を決定する。自律分散システムは個々のエージェントが状況を認識し、行動を決定することにより、環境変動に迅速に対応することが可能である。

群ロボットは、自律分散システムによる群知能(Swarm Intelligence)の研究分野の中の一つである。群知能とは、要素間の相互作用によって創発現象が発生し、より高度な知的システムが現れることに注目したものである。群知能の研究においては、コンピュータ・シミュレーションを用いた研究が多い。これは、様々な相互作用を試すことができるためである。しかし、コンピュータ・シミュレーションによる

\* 大学院工学研究科知能システム工学専攻

\* Department of Human & Artificial Intelligent Systems

研究には、モデル化する過程で本質的な部分を落としてしまう危険性が常に存在する。

群ロボットの研究では、実機を使うことはロボットの身体性を含めて、当初予想もしなかったことが起こる<sup>[1]</sup>。群ロボットには、相互作用する小型ロボットが用いられる。個々のロボットに高度な能力を要求しない群ロボットには、次のような特徴がある。1) ロボットを安価に製作できる。2) 安価ゆえに、多数のロボットを製作できる。3) 多数のロボットが協調行動をおこない、複雑な仕事を達成する (Emergence)。4) 数台のロボットが故障しても群ロボット全体の仕事は影響を受けない (Robustness)。5) 用途を限定し特殊化したロボットではないため、プログラム変更で、様々な作業をこなせる柔軟性がある (Flexibility)。

群ロボットは前述のように多くの利点を持っているが、設計の自由度が大きいいため、かえって設計が難しい。そこで、蟻や蜂などの社会性昆虫 (social insect) の行動をヒントにして、群知能の設計をおこなうという研究がなされている<sup>[2]</sup>。例えば、蟻の群れの行動は、経路の最適化問題などに応用されている (Ant Colony Optimization (ACO))<sup>[3]</sup>。蟻は餌を見つけると、餌の一部をくわえて巣へ戻る。このとき、腹部から餌を見つけたことをあらわす「道しるべフェロモン」を少しずつ地面につけながら戻ってくる。途中で出会った他の蟻は、このフェロモンの匂いを辿って餌場へ集まり、帰りには同じフェロモンをつけながら巣へ戻る。多数の蟻が餌場と巣を行き来することで自己組織的に巣と餌場間の運搬路が最適化されていく。これは、群ロボットによる複雑な仕事の達成例である。

ここでは、社会性昆虫を手本とした群ロボットの設計と開発についてみていく。

## 2. 蟻型群ロボットの multi-CPU 化

蟻型群ロボットは、メカニカル動作部として、移動のための2つのステッピングモータや擬似フェロモン散布用水筆を上下させるためのサーボモータをもつ<sup>[4]</sup>。main-CPUのみでこれらの制御をおこなう場合、ひとつひとつの制御は複雑なものではないが、常にタスクを管理する必要があるため、main-CPUの処理を専有してしまうことになる。そのため、我々は sub-CPU を用いて、モジュール化している。蟻型群ロボットの写真を Fig. 1 に、制御構成を Fig. 2 に示す。

さて、蟻型群ロボットのセンサ群についてみると、センサはライトセンサ、タッチセンサ、接触通信部を持っている。蟻型ロボットは、このセンサ群の反応を

常に確認している。なかでも接触通信部は300[ $\mu$ s]毎に他の蟻型ロボットと接触したかどうかを確認しているため、main-CPUの負荷が大きい。そこで、今回は接触通信部のモジュール化を試みる。

自律分散システムとしての蟻型群ロボットにおいて相互作用、つまりマップ情報や目的地を発見したかどうかの情報を共有することは非常に重要なことである。蟻型群ロボットには、蟻が直接接することで情報交換をおこなう情報伝達方法を手本とした接触通信システムが搭載されている。この接触通信システムを新たな sub-CPU に移植し、接触通信が成功した場合のみ情報を main-CPU に送信することで、main-CPUの負荷を大幅に軽減させることが可能である。

接触通信システムのモジュール化のための sub-CPU として、秋月電子社製の Tiny H8/3694F を用いた<sup>[5]</sup>。Tiny H8/3694F は、周波数20[MHz]の16bit高速CPUで、小型にも関わらず十分に高い処理能力を持っている。メモリはROM 32[KB]、RAM 2[KB]を内蔵しており、同じ Tiny シリーズのなかでも大きい方である。Fig. 3 に、蟻型群ロボットの接触通信システムをモジュール化した制御構成を示す。

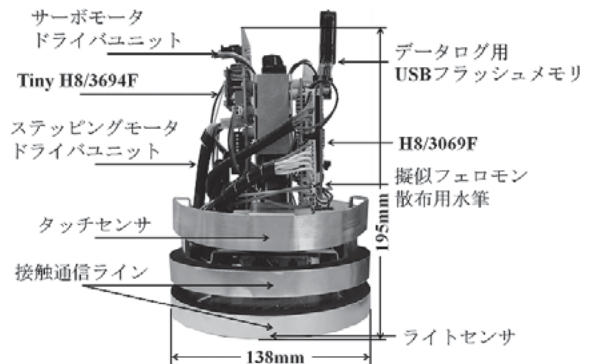


Fig. 1 蟻型群ロボット。

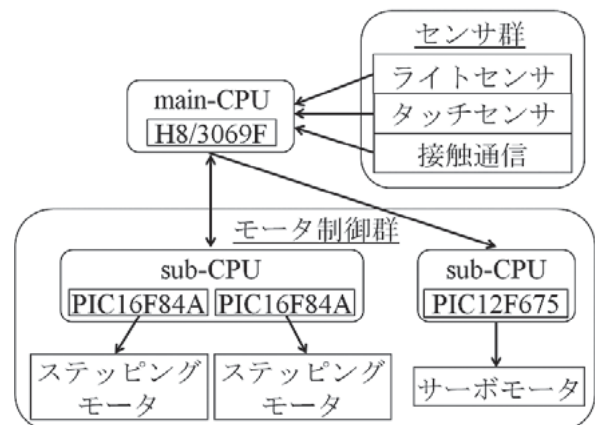


Fig. 2 蟻型群ロボットの multi-CPU 化によるシステム制御構成。矢印の方向はデータの出力方向を示す。

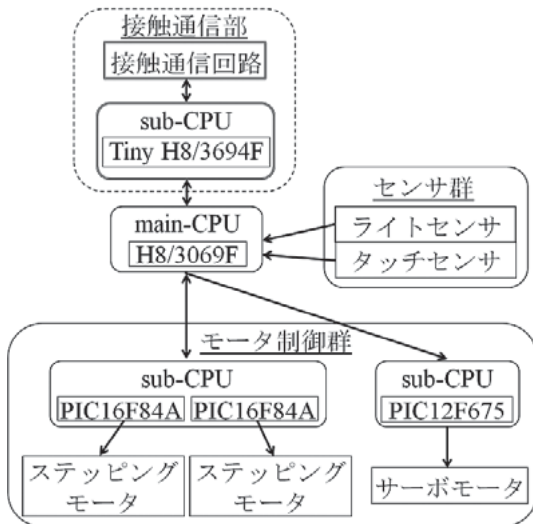


Fig. 3 蟻型群ロボットの接触通信システムをモジュール化したシステム制御構成. 矢印方向はデータの出力方向を示す.

### 3. 接触通信システム

#### 3.1. 接触通信とは

接触通信システムとは、蟻の触覚を用いた情報共有手段を手本として、接触通信ラインが物理的に接触しているときのみデータ通信がおこなえるシステムである<sup>[6]</sup>。蟻のコロニーにおける情報伝達方法を考えたとき、蟻の足跡フェロモンによる運搬路形成はよく知られているが、フェロモンの中には、揮発性で、仲間が直接接触することで伝えられるコンタクト・フェロモンも存在することが分かっている<sup>[7]</sup>。蟻は、他の蟻に出会うと頻りに触覚でお互いを探りあい、体表のワックス(炭化水素)を検出して仲間かどうかの確認をする。また、蟻の触覚には同じ巣の仲間を発見する受容器があることも発見されている。群ロボットに置き換えると、移動するロボット同士が出くわしたとき、物理的に接触している時間は長くない。物理的に接触している間のみ情報交換をおこなうシステムであるため、データ通信中に離れてしまうと、通信に失敗するという問題点がある。しかし、数byteの情報量のデータ通信を想定した場合、物理的に接触している間に、通信をおこなうことは可能である。

接触通信ラインは、一本の通信ラインとGNDから構成されている。双方向の通信を完了させるには1本の通信ラインで送信、受信をおこなう必要がある。したがって接触通信のアルゴリズムにおいて、接触した際にどちらが先に送信、受信をおこなうかの順番を決める必要がある。送信側、受信側を決める際、最初に接触を感知したロボットは、相手の出力状態

の継続時間を計測する。接触通信システムでは、通信ラインの状態を一定の搬送周波数  $T$  で出力・入力を切り替えている<sup>[8]</sup>。計測した出力状態の継続時間が周期  $T$  よりも短い場合、相手のロボットは、こちらのロボットの接触を感知していないことがわかる。その直後に、通信ラインを出力状態へとスイッチし、出力状態を  $2T$  以上保持することにする。そのときの相手のロボットの通信ラインの状態は入力状態であるので、接触を感知しており、出力状態の継続時間を計測している。そして、計測した相手のロボットの出力状態の継続時間が  $T$  より長いことから、相手のロボットは既に接触を感知していると判断できる。通信時には、 $2T$  以上の出力をおこなった側が先に送信を、 $2T$  以上の出力を感知した側が先に受信をおこなう。受信したデータは Start Bit, End Bit の両方が正しかった場合、その間のデータも正しいものとして、その後の蟻型群ロボットの行動アルゴリズムに反映させる。通信データの詳細を Fig. 4 に示す。

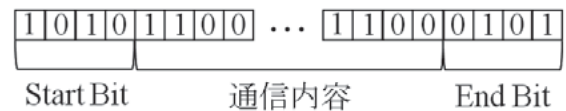
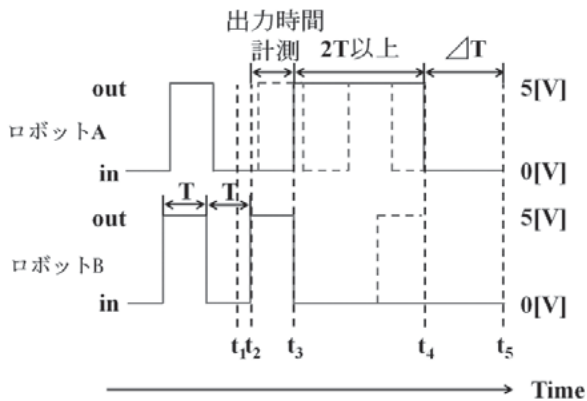


Fig. 4 通信データの詳細。通信データは、情報の正確性確認のための Start Bit, End Bit, 通信内容として敵味方識別 bit やマップ情報などで構成される。

接触通信のプロトコルをタイムチャート (Fig. 5) とタイムチャート進行表 (Table 1) を通して詳しくみていく。まず、ロボットAの視点に立ってタイムチャートをみしてみる。時刻  $t_1$  でロボット同士が接触したとして、時刻  $t_2$  でロボットAはロボットBの存在に気付く。この時点ではロボットBが先にロボットAを感知しているかは不明である。そこで、 $t_2 \sim t_3$  の時間間隔でロボットAは、ロボットBが自身に気付いているか出力状態の継続時間を計測する。計測した結果、ロボットBの出力状態は  $T$  以下であるため、ロボットBはロボットAの存在にまだ気付いていないことがわかる。そこで、ロボットAは先にロボットBの存在に気付いていることを知らせるため  $2T$  以上の出力をおこなう。一方、ロボットBは時刻  $t_3$  でロボットAの存在に気付く。先ほどと同様にして、ロボットBはロボットAが自身の存在に気付いているか出力状態の継続時間を計測する。前述したようにロボットAは先にロボットBの存在に気付いているため、ロボットBは  $2T$  以上の出力状態の継続時間を計測すること

で、ロボットAが先に気付いていることを知り、時刻 $t_4$ で送信、受信をどちらが先におこなうかの順序が決定する。そして、時刻 $t_5$ で $\Delta T$ の待ち時間の後、通信が開始される。この場合はロボットAが先に送信側、ロボットBが先に受信側となる。通信後に送信側、受信側をスイッチする。我々の蟻型群ロボットを用いた実験では、搬送周波数 $T$ を $300[\mu\text{s}]$ とした。

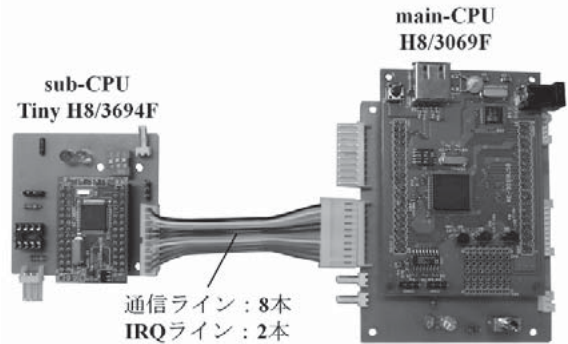


**Fig. 5** 2台のロボット間で、どちらが先に受信側、送信側なのかを決定するタイムチャート。  $t_1$ でロボット同士が接触、  $t_2$ でロボットAがロボットBの存在に気付く、  $t_2 \sim t_3$ でロボットBが気付いているか出力時間を計測、  $t_3$ でロボットBも気付く、  $t_4$ で送信・受信の順番が決定、  $t_5$ で $\Delta T$ の待ち時間の後、通信を開始。

### 3.2. main-CPUとTiny H8/3694Fの通信方式

接触通信システムをsub-CPUを用いてモジュール化した場合、main-CPUと接触通信モジュール(sub-CPU)間のデータ通信が必要となる。main-CPUと接触通信モジュール(sub-CPU)の通信方式は、8本の通信ラインを使って1[byte]単位でのパラレル通信方式を用いた。main-CPUとsub-CPUを接続した写真をFig. 6に示す。また、main-CPUとsub-CPU間の通信は、IRQ(Interrupt ReQuest)割り込み処理を用いてmain-CPUとsub-CPUが柔軟に対応できるような通信方式を設計した。こうしてsub-CPUからmain-

CPU、main-CPUからsub-CPUと、IRQ割り込みを用いた通信を互いに可能とした。また、接触通信に用いる通信byte数はStart Bit, End Bit, 敵味方識別bitやマップ情報など、必要に応じて設定可能である。



**Fig. 6** sub-CPUとmain-CPUを接続した写真。

sub-CPUとmain-CPU間のデータ転送が2[byte]以上の場合について説明する。1[byte]の場合は、一度の送信でデータ転送が完了するが、2[byte]以上の場合は、データ転送において、ハンドシェイクが必要となる。この通信方式を詳しくみていく。通信をおこなう際、送信側はIRQ割り込み信号を送信後、1[byte]のデータを出力し、受信側が認識するまでの待ち時間 $t$ 秒の出力を続ける。この処理を通信byte数を超えるまで繰り返す。受信側は、送信側からのIRQ割り込み信号を受信後、IRQ割り込みルーティーンに入り、出力された値を格納し、カウント変数を加算する。その際、カウント変数があらかじめ設定した通信byte数よりも少ない場合は通常ルーティーンに戻り、通信byte数と等しくなった場合は、カウント変数をクリアし、通信は完了と判断する。最初にデータ長を送信し、転送するデータの長さをデータ転送毎に変えることも可能であるが、ここでは転送データ長は固定とした。

**Table 1** 接触通信タイムチャート進行表。

| Time           | ロボットA            | I/O状態  | ロボットB            | I/O状態  |
|----------------|------------------|--------|------------------|--------|
| $t_2$          | ロボットBの存在に気付く     | input  |                  | output |
| $t_2 \sim t_3$ | ロボットBの出力時間を計測    | input  |                  | output |
| $t_3$          |                  | output | ロボットBの存在に気付く     | input  |
| $t_3 \sim t_4$ | 2T以上の出力          | output | ロボットAの出力時間を計測    | input  |
| $t_4 \sim t_5$ | $\Delta T$ の待ち時間 | input  | $\Delta T$ の待ち時間 | input  |

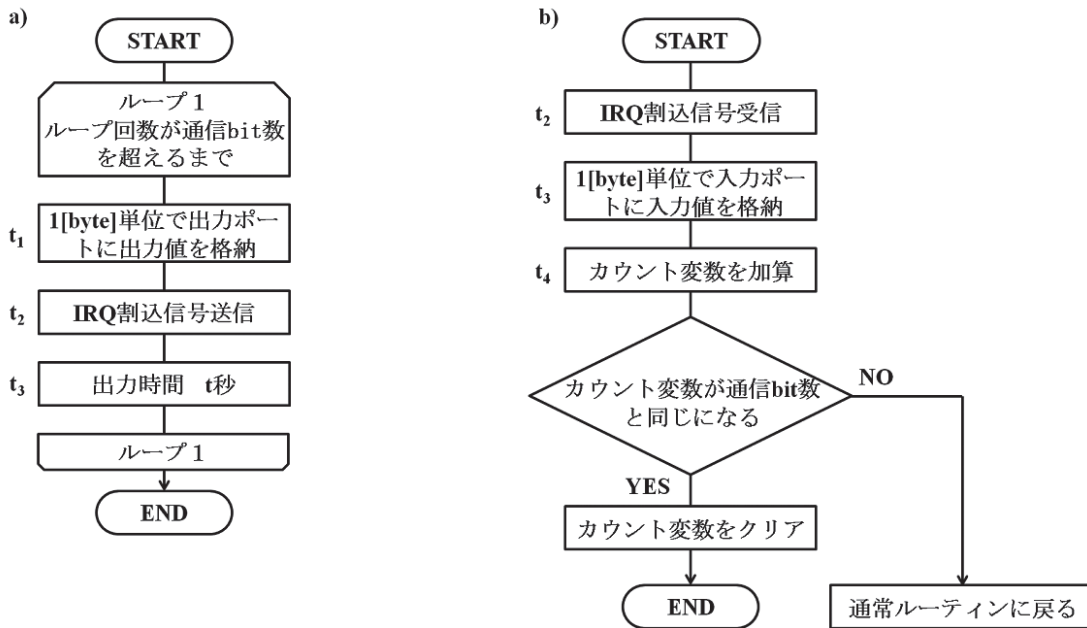


Fig. 7 main-CPUとsub-CPUの通信フローチャート. a)は出力用データを準備し, 出力するための送信側のフローチャート. b)は出力側から出力された値を格納するための受信側のフローチャート.

Table 2 main-CPUとTiny H8/3694Fの通信タイムチャート進行表.

| Time  | sub-CPU                                  | main-CPU  |
|-------|--|---|
|       | 接触通信によって情報を得た                            |   |
| $t_1$ | 1[byte]単位で出力ポートに出力値を格納                   |   |
| $t_2$ | IRQ割込信号送信                                | IRQ割込信号受信   |
| $t_3$ | 出力時間 $t$ 秒                               | 1[byte]単位で入力ポートに入力値を格納  |
| $t_4$ |  | カウント変数を加算   |
|       | ループ回数が通信byte数を超えるまで $t_1 \sim t_3$ を繰り返す | カウント変数が通信byte数と同じならカウント変数をクリアし, 通信は完了とする. カウント変数が通信byte数よりも少ないならば, 通常ルーティンに戻る |

main-CPUとsub-CPU間の通信システムをsub-CPUが接触通信によって得た情報をmain-CPUへ送信すると想定して通信フローチャート(Fig. 7), 通信タイムチャート進行表(Table 2)を通して解説する. まず, あるロボット同士が接触し, 前述したアルゴリズム通り通信が完了したとする. すると, sub-CPUは時刻 $t_1$ において, 1[byte]単位で出力ポートに出力値を格納する. そして, 時刻 $t_2$ でmain-CPUにIRQ割込信号を送信する. こうしてmain-CPUはIRQルーティンに入る. 時刻 $t_3$ でsub-CPUはmain-CPUが出力情報を認識するまで,  $t$ 秒間の出力を継続させる. sub-CPUは $t_1 \sim t_3$ の過程を通信byte数を超えるまで繰り返す. main-CPUは時刻 $t_3$ でsub-CPUから出力された情報を格納した後, 時刻 $t_4$ でカウント変数を加算し, このカウント変数が通信byte数と等しくなれば, 通信を完了とする. また, カウント変数が通信byte数に満たない場合は, 通常ルーティンに戻る.

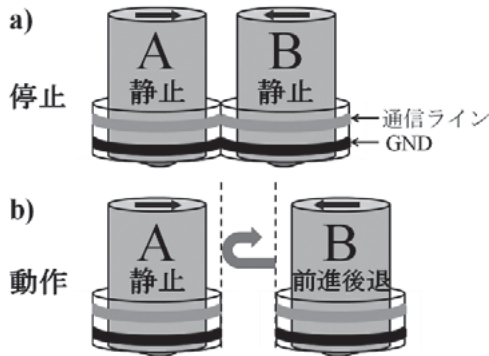
## 4. 実験

### 4.1. 接触通信成功確率の検証実験

Tiny H8/3694Fを搭載した蟻型群ロボットの接触通信の能力を測るため, 停止時と動作時の2つの条件下で接触通信実験をおこなった. この際, ロボット同士は正面から接触させた(側面からの場合については, 後述する). 停止時の実験では, 2台のロボット(ロボットA, ロボットBとする)を静止状態で接触させることで, 接触通信の成功確率を調べた. また, 動作時の実験では, ロボットAを停止させ, ロボットBには, 前進し, 接触を感知したら後退を繰り返させ, 接触通信の成功確率を求める. 通信byte数は1, 2, 4, 8, 16, 19, 32[byte]の7パターンで試行をおこなった. 試行回数は各パターンで10回とし, ロボットA, ロボットBが共に通信に成功したときのみ成功と判断する. Fig. 8に実験の模式図, Table 3に通

通信 byte 毎の接触通信成功確率を示す。

実験結果から、停止時の接触通信は 100 % 成功していたにもかかわらず、動的に接触させた動作時の環境下においては、通信 byte 数の増加に伴い成功確率の減少が示された。これは、動的に接触させたことにより、接触している時間が短くなった可能性がある。そのため通信 byte 数の増加により、実際に接触している時間内に通信の完了が困難になったと考えられる。



**Fig. 8** 接触通信成功確率の検証実験の模式図。a)は2台のロボットを静止させて接触させた。b)はロボットAは静止させ、ロボットBは前進し、ロボットAに接触後、後退することを示す。図中ロボット上部の矢印はロボットの進行方向を示し、ロボット間の矢印はロボットBの動作(前進し、ロボットAに接触後、後退)を示す。

**Table 3** 通信 byte 毎の接触通信成功確率。

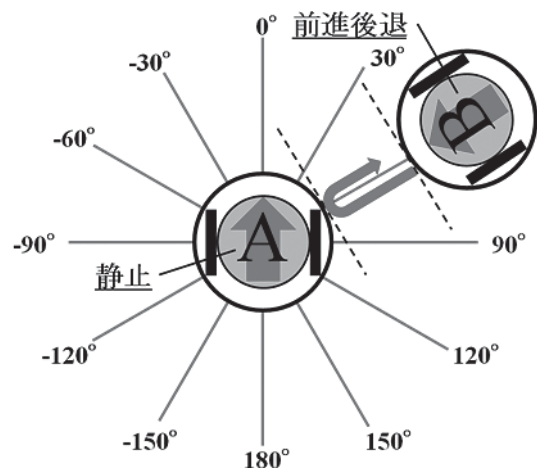
|          | 停止時  | 動作時     |         |
|----------|------|---------|---------|
|          |      | Tiny搭載後 | Tiny搭載前 |
| 1[byte]  | 100% | 100%    | 90%     |
| 2[byte]  | 100% | 90%     | 90%     |
| 4[byte]  | 100% | 80%     | 90%     |
| 8[byte]  | 100% | 70%     | 80%     |
| 16[byte] | 100% | 60%     | 60%     |
| 19[byte] | 100% | 60%     |         |
| 32[byte] | 100% | 0%      | 20%     |

#### 4.2. 角度別通信成功確率の測定実験

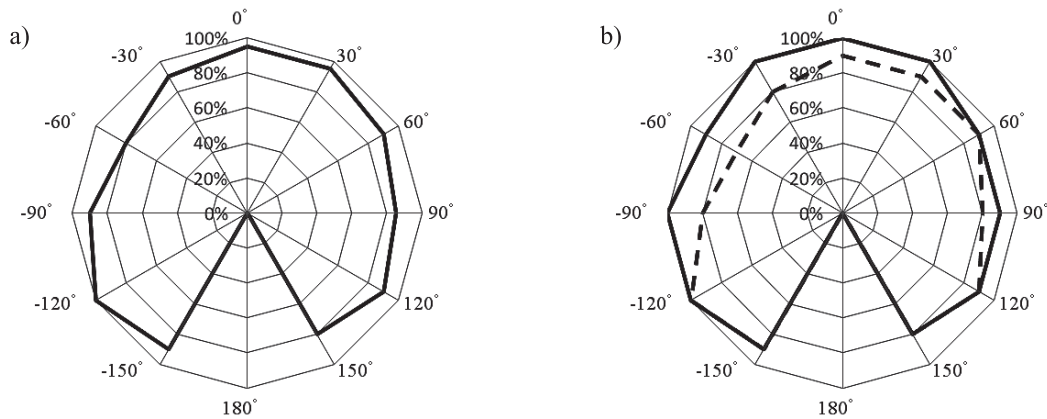
2台のロボットが正面衝突し、接触するのではなく、斜めから衝突する、つまり接触角度が異なる場合について、接触通信成功確率を調べる。停止させたロボットAの正面を0°として30°毎にロボットBを直進させ、接触させた。接触する角度を変えることで、成功確率にどのような変化が起こるのかを調べた(Fig. 9 参照)。通信 byte 数は、1[byte]とした。試行回数は各角度で10回とし、ロボットA、ロボット

Bが共に通信に成功したときのみ成功と判断する。接触通信システムは、接触時に通信をおこなうため、通信ラインとGNDが同時に一定時間接触していないと通信がおこなえないという必要条件が課せられる。このため、通信ラインの微細な形状などが通信成功に寄与してくる。そこで、ロボットによる通信成功確率の差異を調べるため、ロボットAは同じロボットを使用し、ロボットBを異なるロボットを使用して実験をおこなった。Fig. 10に角度毎に通信成功確率を示したローズダイヤグラムを示す。Fig. 10のa)は2台のロボットの平均であり、b)は個々のロボットの成功確率をプロットしたものである。

Fig. 10から、蟻型群ロボットの擬似フェロモン散布用水筆を搭載する後方-150°から150°を除いて高い成功確率で通信がおこなえていることがわかる。角度による接触通信成功確率のばらつきは、蟻型群ロボットのハード面である、通信ラインが原因であると考えられる。蟻型群ロボットは、ロボットにより些細な違い、個性が存在する。これは通信ラインにも影響するため、接触通信の成功確率にも反映される。したがって、各ロボットの通信ラインの調整によって確率が向上する可能性がある。



**Fig. 9** 角度別通信成功確率の測定実験の模式図。角度は、ロボットの正面を0°として、時計りに0°～180°まで、反時計回りに0°～-150°まで30°刻みで実験をおこなった。図中、ロボット内部の矢印はロボットの進行方向を示し、ロボット間の矢印はロボットBの動作(前進し、ロボットAに接触後、後退)を示す。



**Fig. 10** 角度毎に通信成功確率を示したローズダイヤグラム. a)は2台の平均成功確率, b)は個別の成功確率を実線と点線で示した. ロボットの向きは, Fig. 9のロボットAと対応している. なお, 蟻型群ロボットには擬似フェロモン散布用水筆を搭載しているため, 後方 $-150^{\circ}$ から $150^{\circ}$ は0%となる.

## 5. 議論

接触通信システムを Tiny H8/3694F を用いて, モジュール化することに成功した. main-CPU と sub-CPU 間でのデータ転送は, IRQ(Interrupt ReQuest) 割り込み処理を用いておこなった. このアルゴリズムにより, ロボット間の接触がなく, 通信がおこなわれていないときは, main-CPU は行動アルゴリズムの処理に集中できるようになった.

自律分散システムを用いた群ロボットにおいて, 通信システムは重要な機能である. 接触通信システムは, 電磁波ノイズが大きい極限的な環境下においても通信ができるメリットがある. 接触通信の応用例として, 極限下でのレスキューロボットへの応用を挙げることができる. 災害現場での要救助者の発見など, 効率的な目的地探索をおこなうためには, 探索フィールドの探索済エリア, 未探索エリアの情報を共有することが大切である.

接触通信システムは, 電磁波ノイズに強いというメリットがある反面, 接触時の通信成功確率が100%ではないというデメリットがある. 通信ラインの改良で改善できる可能性があるが, 通信成功確率が100%でなくても定用的な意味を持つと考えられる. これは, 集中管理システムとは異なり, 自律分散型システムでは, 100%の通信成功でなくてもシステムが機能するからである. 今回のモジュール化により, main-CPU の負荷が軽減され, ニューラルネットワークなど, より生物的な知能を持った群ロボットの研究が可能となった.

## 謝辞

本研究をおこなう上で, 福井大学大学院工学研究科知能システム工学専攻非線形科学研究室のメンバーには, 実験および議論において大変お世話になりました. ここに感謝の意を表させていただきます.

## 参考文献

- [1] F. Mondada, M. Gambardella, D. Flireano, S. Nolfi, J-L Deneubourg, M. Dorigo. IEEE Robotics & Automation Magazine, 12-2,21(2005).
- [2] 川村秀憲:生命複雑系からの計算パラダイム, 北出版, pp. 1-81(2003).
- [3] M. Dorigo and T. Stützle: Ant Colony Optimization, MIT Press, Cambridge(2004).
- [4] 高井裕紀:汎用蟻ロボットの開発-マルチCPU化によるインテリジェント化-, 福井大学工学部知能システム工学科卒業論文(2011).
- [5] Tiny H8/3694F, Renesas Electronics 社, <http://japan.renesas.com/>
- [6] 平田隆幸, 川地秀幸:福井大工報, 55, 61(2007).
- [7] 水波誠:昆虫-驚異の微小脳, 中央公論新社, pp. 8-25(2006).
- [8] 平田隆幸, 大場公隆:福井大工報, 57, 21(2009).

