

Title	自律移動ロボット群による環境に適応した編隊移動制御に関する研究
Author(s)	小松, 有樹
Citation	
Issue Date	2005-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1862
Rights	
Description	Supervisor: 丁 洛榮, 情報科学研究科, 修士

修 士 論 文

自律移動ロボット群による環境に適応した
編隊移動制御に関する研究

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

小松 有樹

2005 年 3 月

修 士 論 文

自律移動ロボット群による環境に適応した 編隊移動制御に関する研究

指導教官 丁洛榮 助教授

審査委員主査 丁洛榮 助教授

審査委員 松澤照男 教授

審査委員 Defago Xavier 助教授

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

310038 小松 有樹

提出年月: 2005 年 2 月

概要

本稿では、ランダムに位置した自律移動ロボット群によって、フォーメーション制御に必要なパターンの形成を行うアルゴリズムを提案する。形成可能なパターンの種類は多数あり、また、あるパターンからほかのパターンへの再編成や、形成したパターンを維持した移動が可能であることをシミュレーションによって示す。シミュレーションはCで構築したアルゴリズムを Open-GL を利用して表示した。最後にフォーメーション制御を利用した狭隘領域の通過とロボットの故障に対するフォーメーションのロバスト性を示し、本稿で提案するアルゴリズムが環境に適応した編隊移動制御に対して有効であることを確認する。

目次

第1章	序論	1
1.1	研究の背景	1
1.2	研究の目的	3
第2章	フォーメーション	6
2.1	移動ロボット群の定義と仮定	6
2.2	フォーメーション形成方法	7
第3章	leader	12
3.1	leader 決定の目的	12
3.2	leader の決定方法	12
3.3	共通地点の決定	13
3.4	leader の決定	14
3.5	シミュレーション	16
3.6	candidate 重複問題	20
3.6.1	candidate の選定	20
3.7	結果	25
第4章	共通座標	27
4.1	共通座標を創造する目的	27
4.2	共通座標の創造方法	27
4.3	共通方向の創造	28
4.4	共通座標の創造	30
4.5	結果	32
第5章	ID 番号	33
5.1	ID を決定する目的	33
5.2	ID の決定方法	34
5.3	ID 決定方法への制約	34
5.4	ID の決定	35
5.5	結果	39

第 6 章	フォーメーションパターン	42
6.1	フォーメーションパターンの形成	42
6.1.1	各ロボットのフォーメーション形成位置	42
6.1.2	フォーメーション形成位置への移動	56
6.1.3	シミュレーション	57
6.2	フォーメーションの再編成	69
6.3	フォーメーションを維持した移動	72
6.4	フォーメーションのロバスト性	74
6.5	結果	76
第 7 章	結論	78
7.1	本研究のまとめ	78
7.2	本研究のアドバンテージ	78
7.3	今後の展望	79

第1章 序論

1.1 研究の背景

従来、作業ロボットは所定の位置に固定されて単純作業を繰り返すものであったが、ロボット技術の発展に伴い、近年では自らが判断して適切な作業を行う自律移動ロボットが多く開発されている。このような中で、1つの作業を複数台自律移動ロボットの協力によって達成するフォーメーションの研究が注目されている。フォーメーションを組んで作業を行うことで、効率性や耐故障性、機能の分担による1台あたりのコスト減などが期待できる。

アプリケーションとして、地雷探査、清掃、荷物運搬、斥候、警備、レスキューなどが挙げられる。例えば、地雷探査ロボットや清掃ロボットなど、ある領域を網羅する必要がある作業においては、フォーメーションによって探索領域の広い1台のロボットのように振舞うことで作業効率を上げることができる(図1.1)。斥候ロボットはフォーメーションによって役割分担をすることで、単独で行動する際に比べて耐故障性やロボット1台あたりのコスト減が可能となる(図1.2)。また、ロボット群がオフィスや工場内、街中など限られた場所を移動する際は、その環境にいる他の作業ロボットや通行人の妨げにならないように列形のフォーメーションによってロボット群全体が占める領域を少なくすることが必要である(図1.3)。

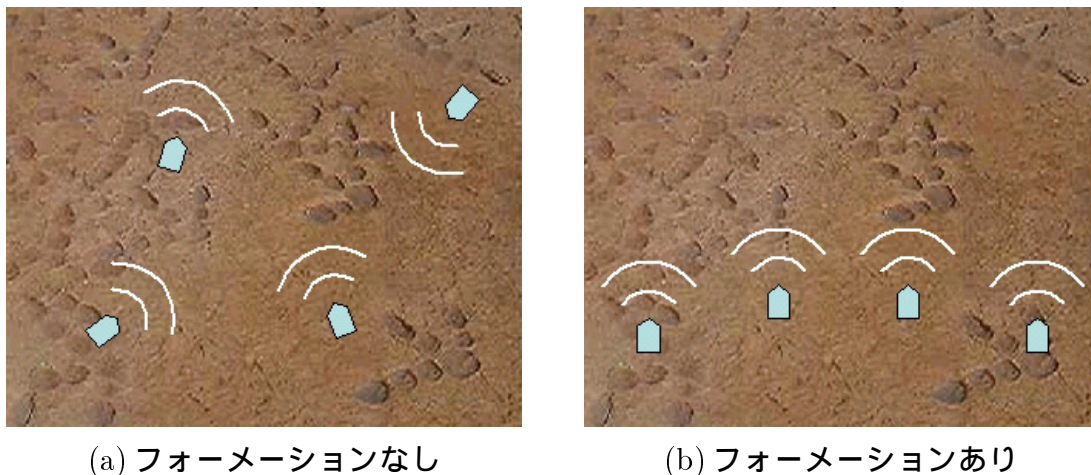
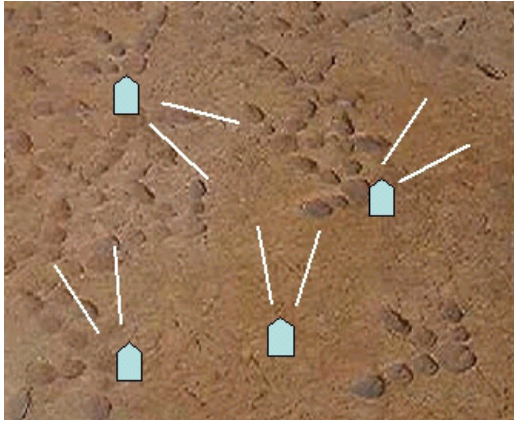
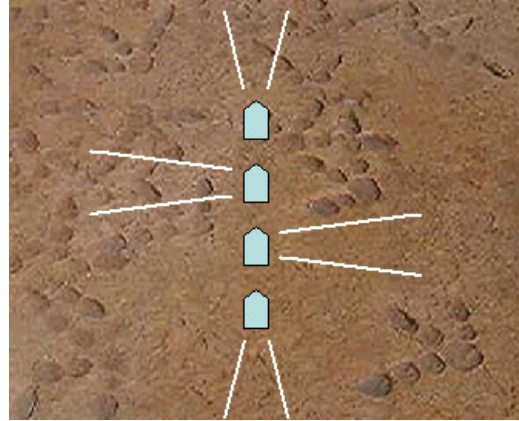


図 1.1: 地雷探査

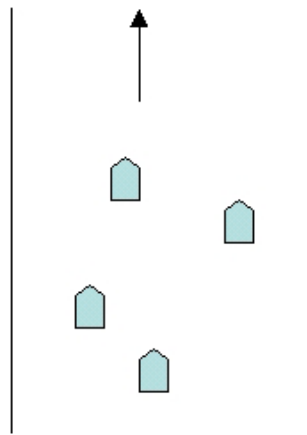


(a) フォーメーションなし

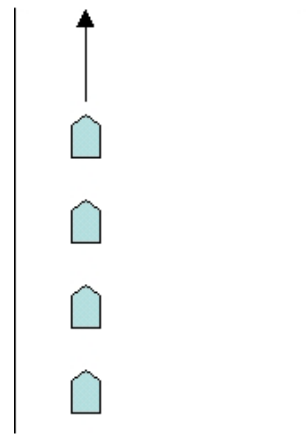


(b) フォーメーションあり

図 1.2: 斥候



(a) フォーメーションなし



(b) フォーメーションあり

図 1.3: 移動

これまで行われた自律移動ロボット群のフォーメーションに関する研究はさまざまなものがある。

ランダムに配置された状態にあるロボットがある 1 点に集まる convergence problem を解決する方法を提案している研究は多くある [3][7][13][14][25]。全ロボットは 1 点に集まることでその地点を共通した基点としてフォーメーション形成に利用できるため、convergence problem は重要な意味を持っている。これらの研究には、ロボットのセンサ範囲に制限を設けていないものと [3][13][25]、センサの範囲に制限を設けているものがある [7][14]。センサ範囲に制限を設けている研究 [7][14] ではこの制限を補うために、全てのロボットは予め共通する方向などを与えられていることが必要となっている。このように convergence

problem の解決を行う研究は多くあるものの、これらの研究はロボット同士の衝突を無視しなければならず、現実的ではないといえる。

フォーメーションの形成を行う研究に [3][8] などがある。Suzuki & Yamashita は各ロボットが相互の位置への移動を繰返すことで [3]、Defago & Konagaya はボロネイ線図を利用することでフォーメーション形成を行っている [8]。他の研究には [6][5][9] などがある。これらの研究はロボットの配置がランダムである状態から、anonymous などの制約を設けた弱い機能を持つロボット群によってフォーメーション形成を行っている¹。このため、形成できるフォーメーションパターンの種類が非常に少ない。形成可能なパターンの種類は多いほど、さまざまな状況に適応できる一般性のあるアルゴリズムといえる。

多様なフォーメーションパターンの形成を可能としている研究に [10][11][26] などがある。これらの研究では、外装の違いと相互に違いを確認するセンサや [11][26]、赤外線や相互通信を利用している [10]。このように、多様なフォーメーションパターンを形成している反面、特別な装置への依存を前提としている。また、範囲に制限のあるセンサを用いてフォーメーションの形成を行う研究に [10][11] があるが、これらの研究ではセンサ範囲の制限を補うために初期状態を人の手によって意図的に操作しなければならず、ロボット群はフォーメーションを自律的に形成することができていない。

ロボット群の移動に関する研究はそのほとんどが leader-followers 方式を用いている。leader-followers 方式では群れを形成するロボットが 1 台の leader とその他の follower に分かれ、主に leader が経路探索を、follower が群れの維持に努める。群れの舵取りを 1 台のロボットに委託することで、統率のある群れの移動を可能としている (図 1.4)。leader-followers 方式を利用した群れの移動に関する研究に以下がある。

中川はドロネー線図を利用して群れの維持を行っている [12]。狭隘領域の通過が可能であるが、特定のパターンを維持することができない。群れの移動時はパターンを維持することが重要である。

Balch & Arkin は群れの維持について Unit-center-referenced、Leader-referenced、Neighbor-referenced の 3 種類の方法を提案し、behavior-based を利用した障害物回避によってそれぞれを検証している [21]。

その他の研究に [11][23][24][26][27] などがある。これらの研究には、leader や neighbor からの距離と角度を維持するものと [11][24][26][27]、leader からのみの距離と角度を維持するものがある [23]。また、障害物の回避を行う研究もある [24][27]。

1.2 研究の目的

フォーメーションを組んで協調作業を行うロボット群は、以下が実行可能であることが必要である。

(a) ランダムなロボット配置からのフォーメーション形成

¹anonymous とは、外見の区別がない、同じアルゴリズムを実行するロボット群のことである [3]。

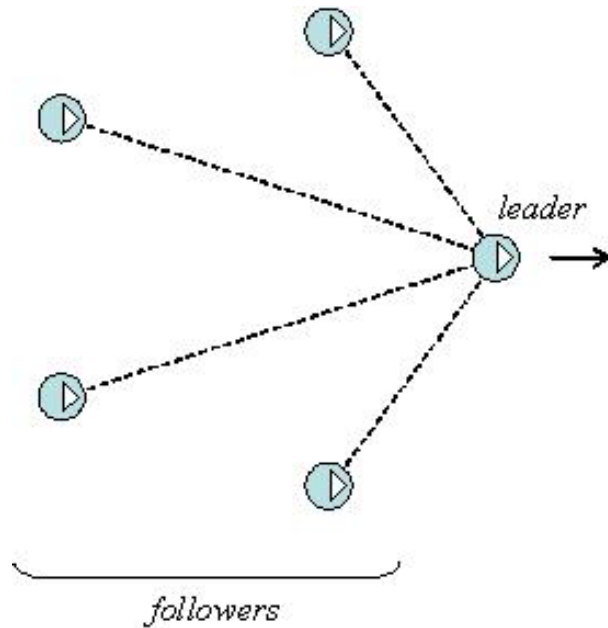


図 1.4: leader-followers 方式による移動

- (b) フォーメーションを維持した移動
- (c) フォーメーションの再編成
- (d) 狭隘領域の通過や障害物回避

関連研究を考察すると、それらは (a) を行う研究と (b)(d) を行う研究に大別することができる。このため、(a)(b) 間のメカニズムが考慮されることはほとんどない。また、(a) を行う研究において多様な種類のフォーメーションを形成可能にしているものは少ない。本研究ではこれらの点に着目した。

(a)(b) 間のメカニズムを考慮し、多様なフォーメーションパターンの形成からフォーメーションを維持した移動までを行う研究に [11][26] などがあるが、これらの研究は各ロボットの外見の違いや相互の違いを確認するセンサなどの特別な装置を必要としている。

そこで本研究では、初期状態において外見に区別のない同じアルゴリズムを実行する anonymous ロボット群を用い、ランダムな配置からの多様な種類のフォーメーションパターンの形成を可能とするアルゴリズムの提案を行う。このアルゴリズムを利用することで、フォーメーションを維持した移動やフォーメーションの再編成についても達成可能であることを示す (図 1.5)。また、本研究で提案するアルゴリズムはフォーメーションが fault-tolerance であり、ロボット群は群れを構成するメンバが故障した際に正常に動くロボットでフォーメーションを維持することが可能であることを示す。

最後に、ロボット群はフォーメーションを維持した移動とフォーメーションの再編成によって狭隘領域の通過が行えることをシミュレーションで表し、自律移動ロボット群による環境に適応した編隊移動制御に対するアルゴリズムの実現可能性を示す。

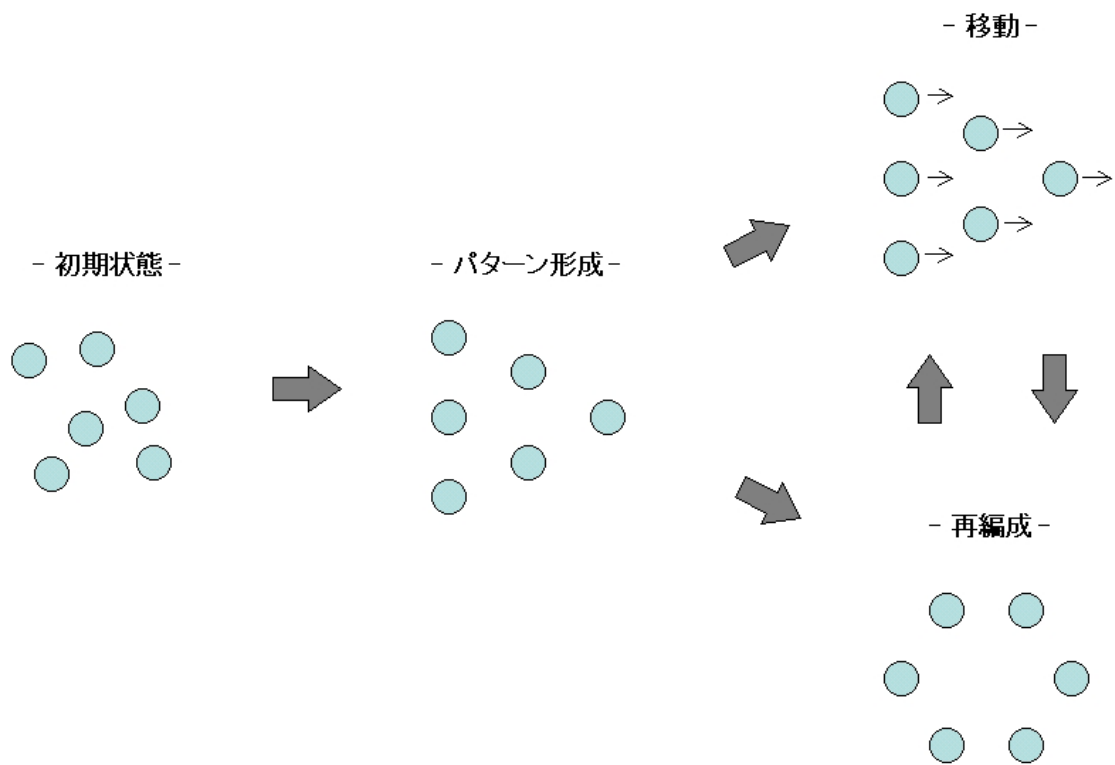


図 1.5: モチベーション

第2章 フォーメーション

2.1 移動ロボット群の定義と仮定

各ロボット r_i は移動能力とセンシング機能を持つ自律ロボットであり、初期状態において障害物のない平面上にランダムに位置している (図 2.1)。複数のロボットが同地点にいることはなく、全てのロボットはそれぞれ離れた場所に位置している。また、ロボット同士は衝突を起こすため、初期状態以降においても複数のロボットが同時に同位置にいることはない。

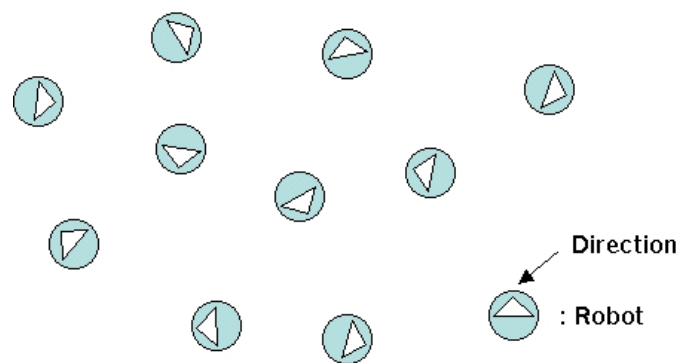


図 2.1: 初期状態

各ロボット r_i のローカル座標は、ロボット r_i の中心位置 p_i が原点 $(0, 0)$ 、ロボット r_i の進行方向が Y 軸である (図 2.2)。このため、ロボット r_i のローカル座標における他のロボット r_j の位置座標 $L_i[p_j]$ はロボット r_j の移動のみならずロボット r_i の移動によっても変化する。

初期状態において全てのロボットに共通する地点、方向、座標等の共通知識は与えられておらず、これら共通知識を得るための flag や lighthouses など存在しない。また、compass 等の各ロボットが持つ特別な装置も存在しない。それぞれに固有の識別番号や外見の違いもなく、全てのロボットは同じアルゴリズムを実行する。このため、ロボットは anonymous であるといえる。

センサによって他のロボットの進行方向や座標軸方向の取得、外見による各ロボットの区別をすることはできない。センサによって直接得られる情報は以下の通りである (図 2.2)。

- ロボット r_i の中心位置 p_i から他のロボット r_j の中心位置 p_j までの距離 $dist(p_i, p_j)$
- ロボット r_i の X 軸方向 $\vec{x}_i$ から p_j までの角度 $angle(\vec{x}_i, p_j)$

これらの計測にかかる時間と計測範囲の限度は考慮せず、計測結果は常に正しいものとする。

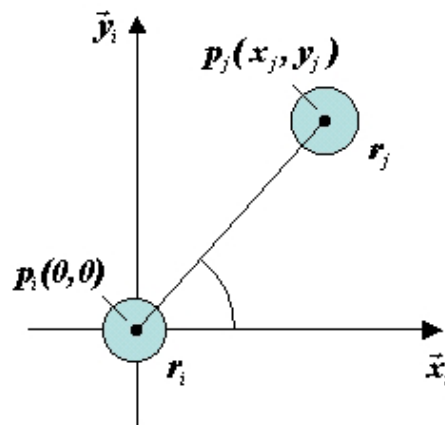


図 2.2: センサ

2.2 フォーメーション形成方法

全ての研究は何らかの共通知識を利用することでフォーメーションの形成を行っている。

ランダムに配置された状態にあるロボットがある 1 点に集まることで、その地点を共通した基点としてフォーメーション形成に利用できるとしている研究に [3][7][13][14][16][17][25] がある。この方法を利用し、各ロボットが相互の位置への移動を繰り返すことによってフォーメーションの形成を行う研究に [3][5] が、共通座標を生成する研究に [4] がある。

共通地点を得るその他の方法に circle の形成がある。circle を形成することで全てのロボットは circle の中心点を共通する地点として認識することが可能である。このような研究に [6][8][9][10] などがある。また、池本らは circle の形成後に数台のロボットが移動によって中心点からの距離を変えることで、triangle などの形成を行っている [10]。

以下の研究は全てのロボットが何らかの共通知識を得るための特別な装置を予め所持していることを前提としている。compass などの利用によって全てのロボットに共通する方向を得る研究に [1][7][14] が、lighthouse などの利用によって共通する地点を得る研究

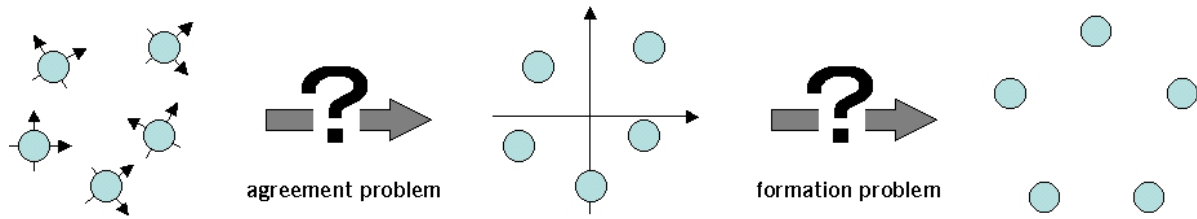


図 2.3: agreement problem と formation problem

に [2] がある。また、環境の依存によって共通知識を獲得する研究に [22] がある。下村らは格子状のフィールドを利用することでフォーメーション形成を行っている [22]。また、Flocchini らはさまざまな共通知識を前提として各ロボットに与え、フォーメーション形成との関係を示している [15]。

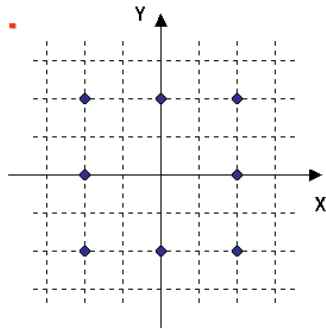
センサの範囲に制限を設けないことで、各ロボットは相互の位置を共通知識とすることができる。これは多くの研究において行われている。全てのロボットが孤立した状態においてフォーメーションの形成を確実に行うことはほとんど不可能であり、センサの範囲に制限を設ける研究では [11] のように初期状態におけるロボット配置の操作や、[14] のように予め共通座標があるとする仮定が必要となっている。

共通知識のほかに、各ロボットがフォーメーションパターンにおける辺や頂点などの役割を得ることでフォーメーション形成問題を緩和することができる。役割を利用する研究に [10][11][26] などがあり、フォーメーションの形成前にロボット間で辺や頂点などのフォーメーションパターンにおける役割を取り決めている。これによって各ロボットがそれぞれ異なる行動をすることができるため、フォーメーション形成問題を容易にすることができる。

これらの関連研究から、あるフォーメーションをどのように形成するかという formation problem と全ロボットに共通する知識や各ロボットに与えられた役割をどのように得るかという agreement problem は密接に関係しており、agreement problem の解決が formation problem の解決のために非常に重要であることがわかる (図 2.3)。

本研究では、各ロボットが共通知識として共通座標を、役割として可変 ID 番号を持つことで、formation problem を解決する手段とする。これらを利用すると図 2.4 のように共通座標からフォーメーション形成に必要な位置を、ID 番号からそれぞれがどの位置に移動するかを決定することが可能となる。これによってフォーメーションごとにロボットの配置と役割分担を定めておくことで多様なフォーメーションパターンの形成が可能となる。フォーメーションを維持した移動には leader-followers 方式を用い、統率のある群れの移動を行う (図 1.4)。以上のようにフォーメーションの形成とフォーメーションを維持した移動を行うため、agreement problem における共通座標と ID 番号の取得、leader の決定が必要である。

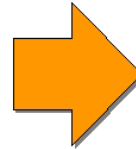
・ 共通座標 ・



・ ID番号 ・

頂点	右上	右下	左上	左下
ID	①	③	⑤	⑦

辺	上	下	右	左
ID	②	④	⑥	⑧



ex) Quadrilateral

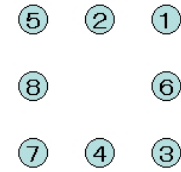


図 2.4: 共通座標と ID を利用した quadrilateral 形成

ロボット群はこれらの知識を獲得するため、まず、共通する地点を創造し、その地点を利用して他の共通知識の獲得を行う。つまり、比較的達成しやすい共通知識の獲得と、その共通知識を利用した他の共通知識の獲得を繰り返すことにより、達成困難な共通知識の獲得を可能とする。これにより、フォーメーション形成やフォーメーションを維持した移動を実現する (図 2.5)。フォーメーション形成までのフローチャートは図 2.6 の通りである。

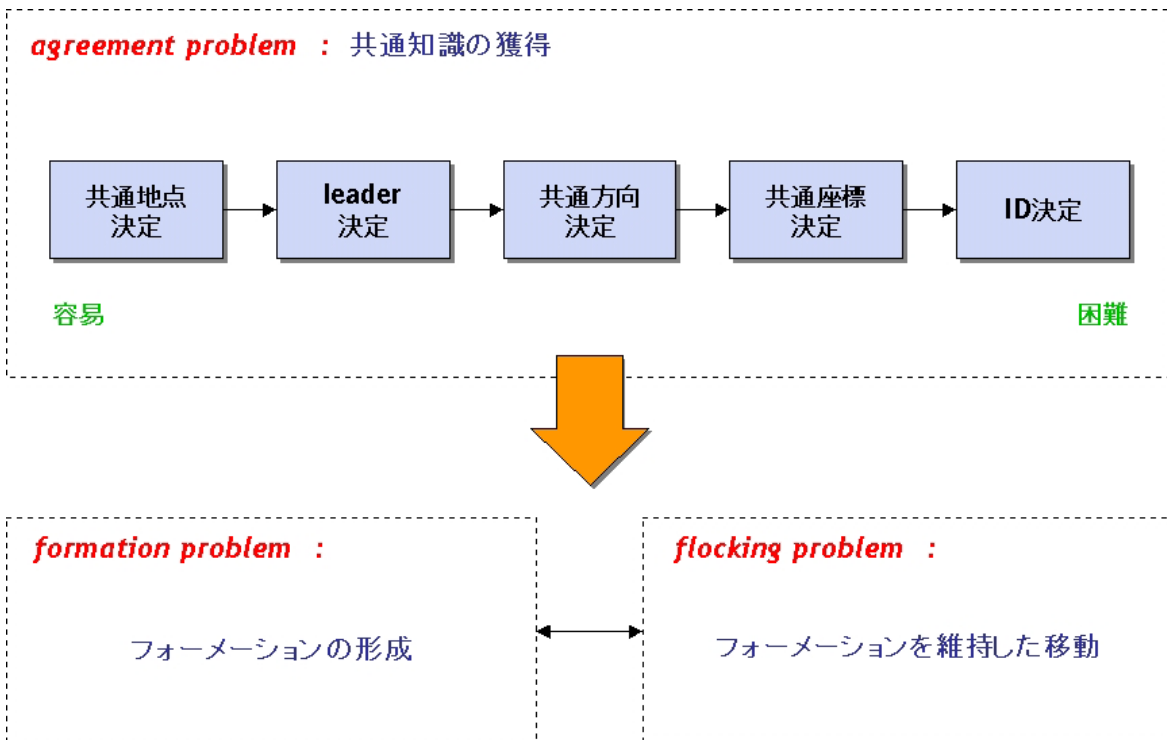


図 2.5: 知識の獲得とフォーメーション形成

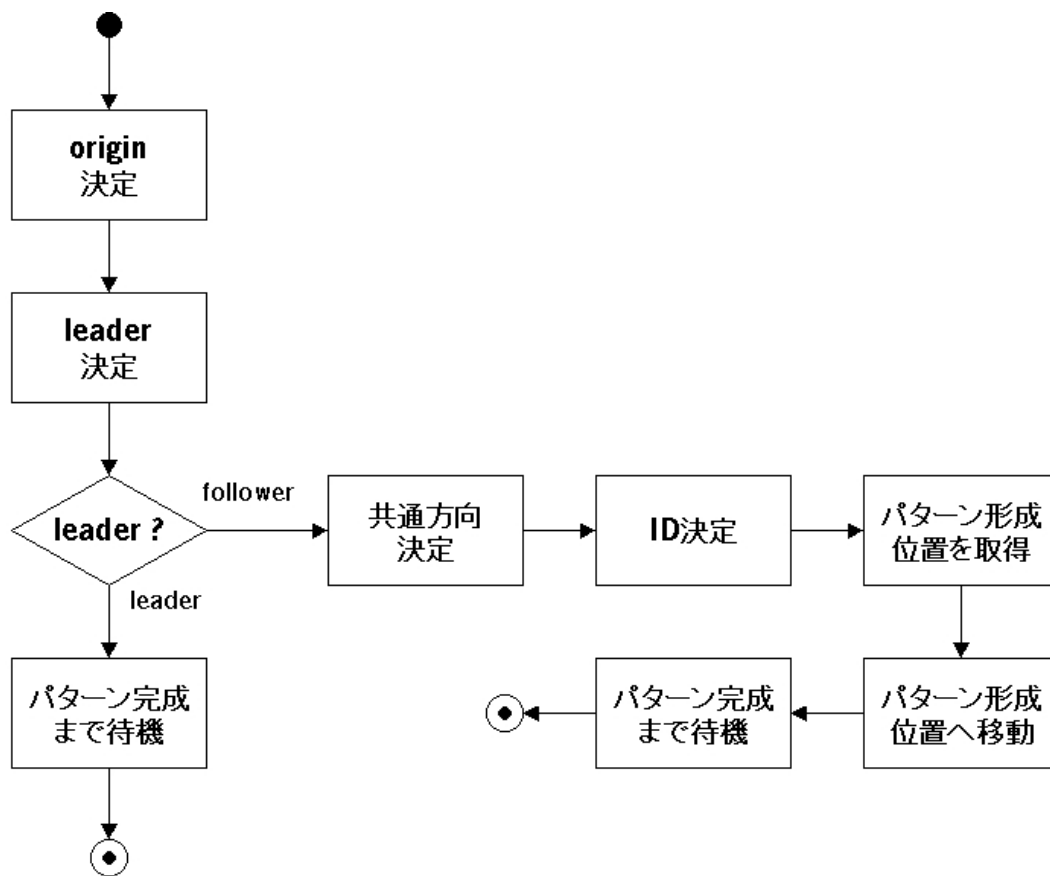


図 2.6: Scenario

第3章 leader

3.1 leader 決定の目的

leader を決定する目的の 1 つは、フォーメーション形成後にフォーメーションを維持したロボット群の移動を leader-followers 方式で行うためである (図 1.4)。leader-followers 方式では 1 台のロボットが leader となり、leader 以外のロボットは leader に追従する follower として振る舞う。leader が群れの先頭に立って移動し、各 follower がフォーメーションの維持に努めることで、結果としてフォーメーションを維持したロボット群全体の移動が可能となる [18]。leader-followers 方式ではロボット群全体の行動を 1 台の leader が決定するため、複数台のロボットがそれぞれの行動を主張し合うことなく統率の取れた移動ができるという利点がある。このため、共通知識の獲得によって leader の決定を行い、その leader が全てのフォーメーションパターンの先頭に位置することが必要である。

leader を決定するもう 1 つの目的は、leader が位置する地点を他の共通知識の獲得やフォーメーション形成に利用するためである。全てのロボットがある共通する 1 台を leader として認識することでその leader がいる地点を共通地点として認識し、この地点を共通座標の生成 (4 章を参照) やフォーメーション形成 (6 章を参照) に利用することが可能である。

leader の存在は群れの統率において重要な要素であり、leader の決定方法のみに注力した研究もある [18][19][20]。これらの研究が無線通信を用いているのに対し、本章では無線通信を用いずに leader を決定する方法の提案を行う。

3.2 leader の決定方法

全てのロボットが共通して認識する地点があれば、その地点からの距離の近さが 1 番、2 番、3 番 $\dots$ N 番と、各ロボットを順位づけすることができる。何番目のロボットを leader にするかを予め定めておくことで leader の決定が可能である (図 3.1)。

しかし、本研究では flag や lighthouses などその環境にある特別な装置を共通知識獲得の手段として利用しない。これらを利用するとフォーメーション形成が環境に依存することになり、一般性にかけることになる。このため、ロボット群は leader の決定を行うために、全てのロボットに共通となる地点を創造することが必要である。3.3 節でその方法を述べる。

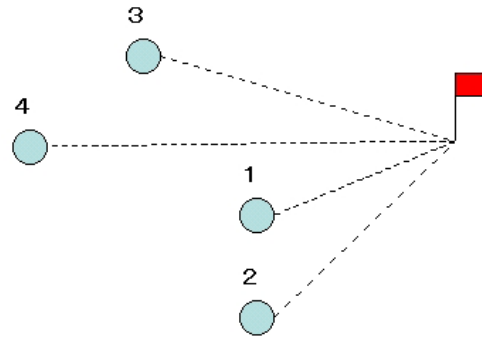


図 3.1: 共通地点からの距離によるロボットの順位付け

3.3 共通地点の決定

本節で述べる共通地点が、ロボット群が最初に獲得する共通知識である。これは、共通地点が他の共通知識に比べて容易に獲得することができ、また、共通方向や共通座標、leader などその他の共通知識の獲得に利用しやすいからである（4 章を参照）。

関連研究

共通地点はその他の共通知識の獲得やフォーメーションの形成に利用可能であるため、共通地点の存在はフォーメーション制御において重要な意味を持つ。このため、多くの研究において共通地点を獲得する方法が提案されている。

全ロボットがある 1 点に集まることで、その地点を共通地点とすることができるとしている研究に [3][7][13][14][16][25] がある。しかし、これらは 1 点に集まるまでに多くの移動距離と時間を必要としている。また、この方法はロボット同士の衝突を考慮しないという前提が必要である。この前提は技術の発展によっても解決することが難しく、1 点に集まることによる共通地点の創造は現実的ではないといえる。Donald は beacons や lighthouses などを用いることで共通地点を得ることができるとしている [2]。しかし、このような特別な装置の存在は一般的環境において期待することはできない。circle を形成することで circle の中心位置を共通地点とすることができるとしている研究に [6][8][9][10] がある。しかし、この方法では 1 点に集まる方法と同様に、circle の形成までに多くの移動距離と時間が必要となってしまう。

本研究では全ロボットの配置から重心点を算出し、その重心点を共通地点とする。この方法には各ロボットの移動が必要ではなく、共通地点を得るための時間が少なくすむ利点がある。

アルゴリズム

各ロボットはそれぞれのローカル座標における全ロボットの位置座標を加算し、群れのロボット数 N （以下 N と表記する）で割ることで全ロボット配置の分布の重心点を算出することができ、その位置を共通地点とすることができる。

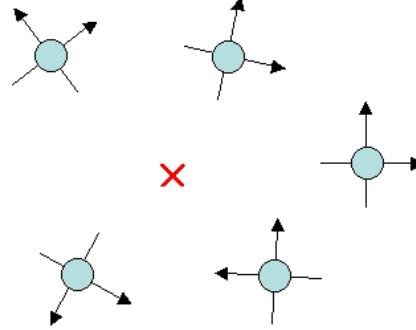


図 3.2: ロボット群と重心点

まず、ロボット r_i はそのローカル座標における他ロボット r_j の位置 p_j の座標 $L_i[p_j] = (L_i[x_j], L_i[y_j])$ を式 (3.1) で得る (図 2.2)。

$$\begin{aligned} L_i[x_j] &= \cos(\text{angle}(\vec{x}_i, p_j)) \times \text{dist}(p_i, p_j) \\ L_i[y_j] &= \sin(\text{angle}(\vec{x}_i, p_j)) \times \text{dist}(p_i, p_j) \end{aligned} \quad (3.1)$$

式 (3.1) を利用し、ローカル座標における重心点の位置 p_c の座標 $L_i[p_c] = (L_i[x_c], L_i[y_c])$ は式 (3.2) で得る。 $L_i[p_c]$ はロボットによって異なる値となるが、その座標の示す位置は同じ地点となる。これによって共通地点の獲得が可能となる。

$$\begin{aligned} L_i[x_c] &= \frac{\sum L_i[x_j]}{N} \\ L_i[y_c] &= \frac{\sum L_i[y_j]}{N} \end{aligned} \quad (3.2)$$

ここで得られた共通地点は共通座標の原点とすることができる (4 章を参照)。そのため、各ロボットはそれぞれのローカル座標における重心点の位置座標 $L_i[p_c]$ を origin の位置座標 $L_i[p_o]$ として記憶する。以下、共通地点を origin と表記する。

3.4 leader の決定

leader 決定のフローチャートは図 3.3 のとおりである。

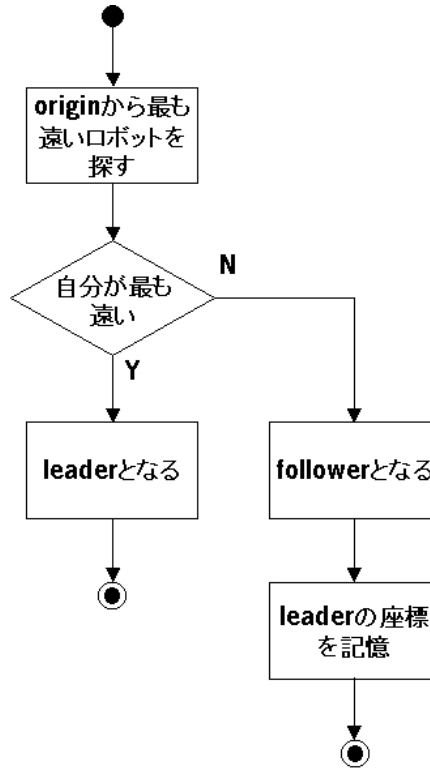


図 3.3: leader 決定フローチャート

まず、各ロボットは全てのロボットから origin までの距離を測る。

各ロボット r_i は origin の位置 $p_o = (x_o, y_o)$ までの距離 $dist(p_o, p_i)$ を式 (3.3) で得る (図 3.4)。

$$dist(p_o, p_i) = \sqrt{L_i[x_o]^2 + L_i[y_o]^2} \quad (3.3)$$

各ロボット r_i は origin の位置 p_o から他のロボット r_j の位置 p_j までの距離 $dist(p_o, p_j)$ を式 (3.4) で得る (図 3.4)。

$$dist(p_o, p_j) = \sqrt{(L_i[x_o] - L_i[x_j])^2 + (L_i[y_o] - L_i[y_j])^2} \quad (3.4)$$

式 (3.3) と式 (3.4) から、各ロボット r_i は origin から最も遠い地点にいるロボットを leader とし (図 3.5) その位置 p_l のローカル座標における位置座標 $L_i[p_l]$ を記憶する。

origin から最も遠いロボットとした理由は leader-followers 方式による移動のためである。leader-followers 方式では leader は群れの先頭になって群れを先導する。初期状態において origin から最も遠いロボットは群れの先頭にあると見ることができるので、origin から最も遠いロボットを leader とする。これにより、leader が群れの先頭に出るための移

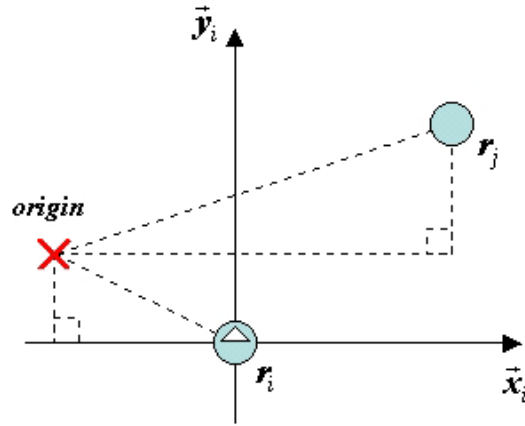


図 3.4: origin から各ロボットまでの距離の算出

動や各 follower が leader の後ろに回りこむための移動、それらに伴う衝突回避や時間の浪費を回避することができる。

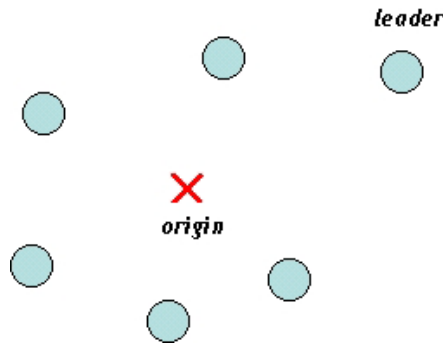


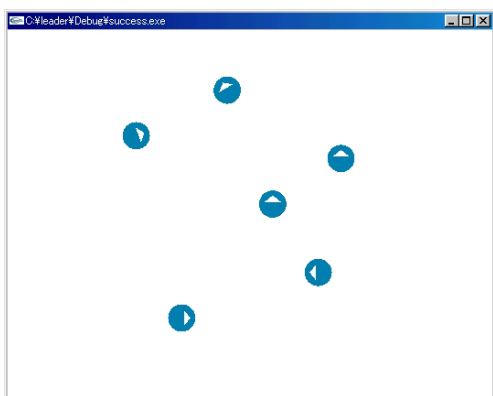
図 3.5: origin からの距離と leader

3.5 シミュレーション

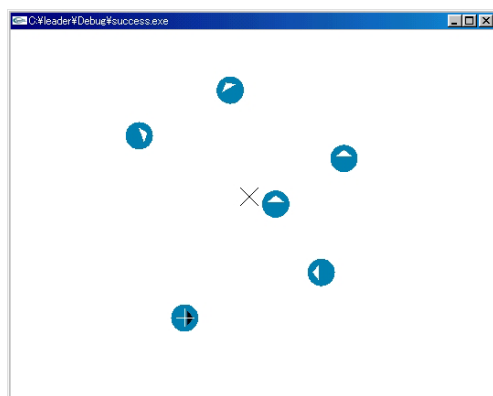
それぞれが固有のローカル座標を持つランダムに位置した自律移動ロボット群によって、群れの中から 1 台の leader を決定するシミュレーションを行った。シミュレーション画像（図 3.6）において、丸型模様はそれぞれロボットである。ロボットの中の三角模様はそのロボットの進行方向を表している。×（黒）は各ロボット r_i のローカル座標における origin の位置 $L_i[p_o]$ である。+（白）は各ロボット r_i のローカル座標における leader の位置 $L_i[p_l]$ である。また、leader 決定アルゴリズムの結果、自身が leader であると判断したロボットは進行方向を表す三角模様を白から黒へと変化させる。

シミュレーション結果を図 3.6 と図 3.7 に示す。(a) が初期状態であり、(b) が leader を決定した結果である。

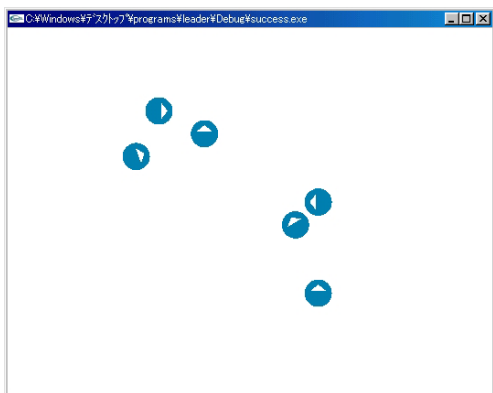
leader 決定アルゴリズムの結果、sim01、sim02、sim03 の各状態では一意に leader を決定することができた。しかし、sim04、sim05 では leader を一意に決定することができず、重複していることがわかる。これは、origin からの距離が同じロボットが複数あり、それらのロボットより origin からの距離が遠いロボットが存在しないとき、leader となる条件を満たしたロボットが 1 台にならないためである。このため、sim04、sim05 のような状態においても leader 決定を可能とするアルゴリズムが必要となる。



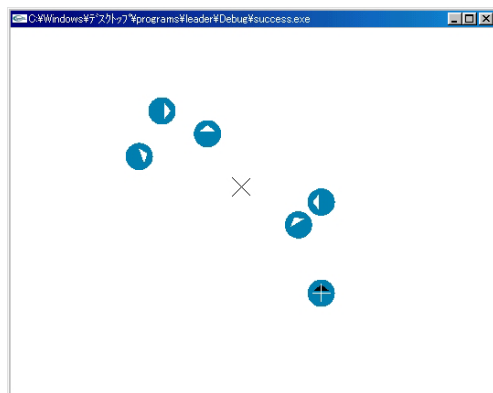
sim01 (a)



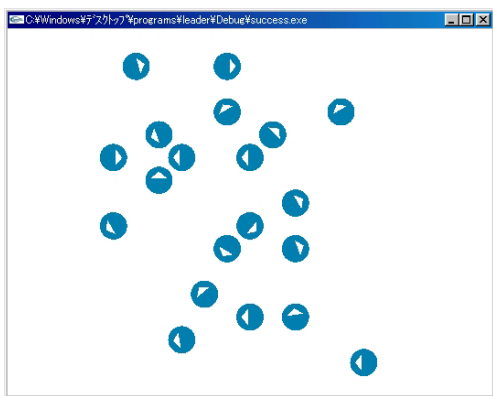
sim01 (b)



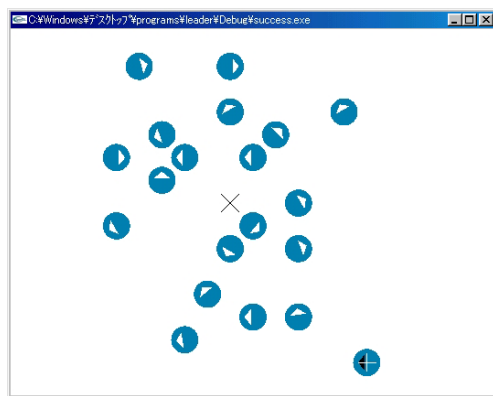
sim02 (a)



sim02 (b)

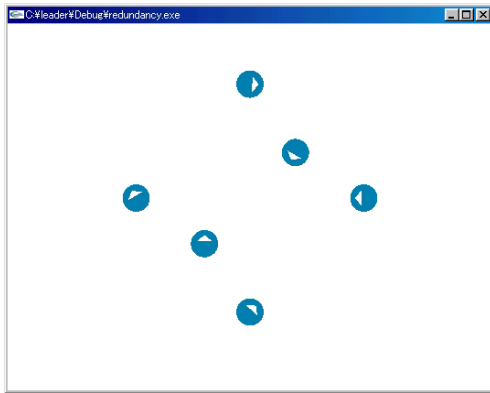


sim03 (a)

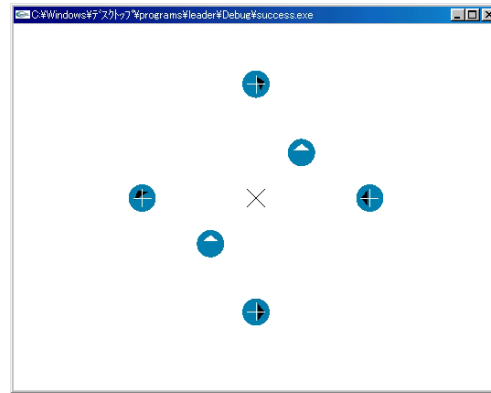


sim03 (b)

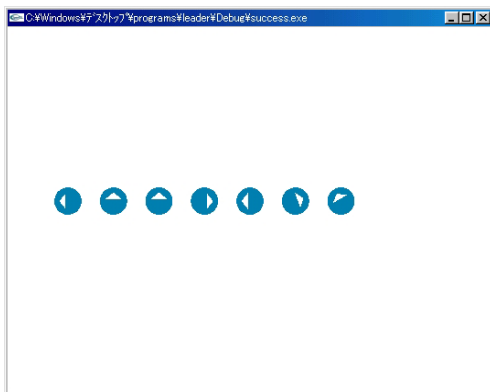
図 3.6: leader 決定の成功 (a) 初期状態 (b) leader 決定後



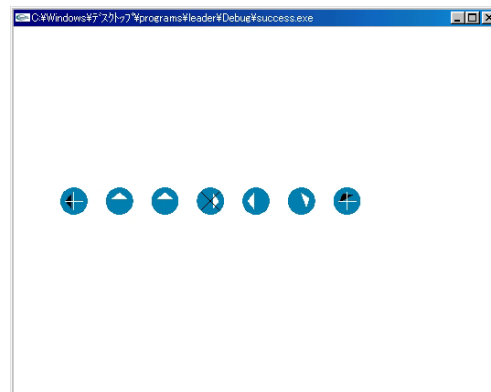
sim04 (a)



sim04 (b)



sim05 (a)



sim05 (b)

図 3.7: leader 決定の失敗 (a) 初期状態 (b) leader 決定後

3.6 candidate 重複問題

シミュレーション結果図 3.6 のロボット配置のように、origin から全てのロボットまでの距離が異なるとき leader は一意に決定される。しかし、図 3.7 のロボット配置のように、origin からの距離が同じロボットが複数あり、それらのロボットよりも origin からの距離が遠いロボットが存在しないとき、leader となりうる candidate ロボット (以下 candidate と記述) が 1 台ではないため leader を一意に決定することができない (図 3.8)。

この問題を解決するため、重複が起きた場合に candidate はランダムな方向に一定距離だけ移動し、origin から各 candidate までの距離を変化させる。candidate 以外は全ロボットが停止状態になるまで待機し、再び leader 決定アルゴリズムを実行する。candidate が 1 台になるまでこれを繰り返すことで leader を一意に決定する。図 3.9 に candidate 重複問題を考慮した leader 決定フローチャートを示す。

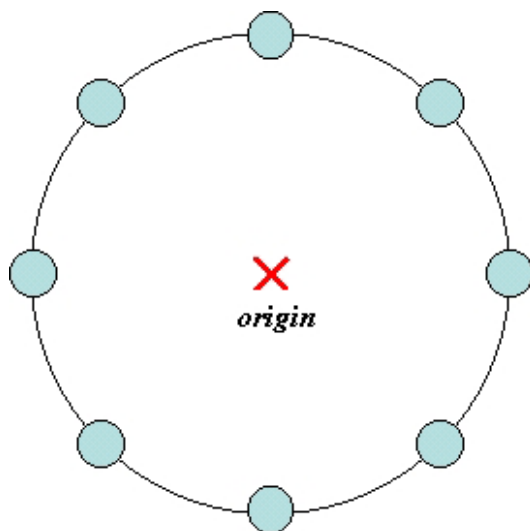


図 3.8: 全ロボットが candidate となる状態 (circle)

3.6.1 candidate の選定

candidate 重複後、各 candidate はそれぞれから origin までの距離が異なるようにランダムな移動方向に一定距離だけ移動する。candidate が 1 台になるまでこれを行うことで leader を一意に定める。

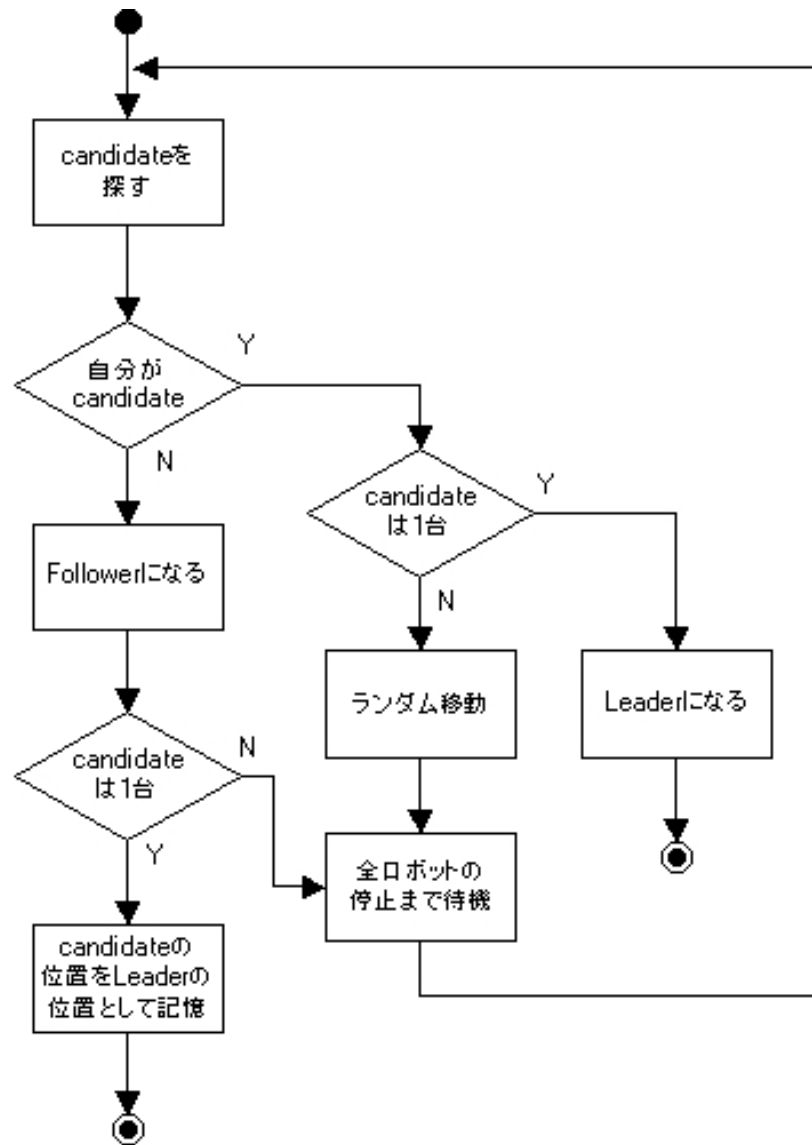


図 3.9: candidate 重複問題を考慮した leader 決定フローチャート

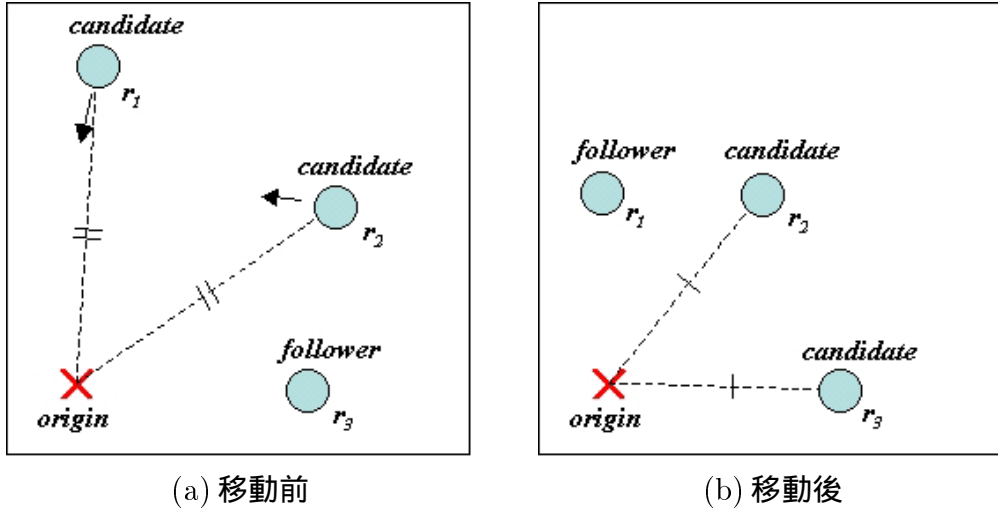


図 3.10: candidate の再重複

candidate の移動方向

candidate が再重複する機会を減少させるため、candidate の移動方向に制限をつける。これは、candidate が origin までの距離の近くなる範囲に移動した場合、一度 follower に決定したロボットが candidate になる可能性があるためである。図 3.10 において、candidate である r_1 と r_2 （番号は図の説明のために表記）が origin に近づく範囲に移動したことにより、 r_2 と follower であった r_3 が candidate の再重複を起こしている。origin に近づかない範囲に移動することで、leader となるロボットを candidate となったロボットに絞ることができる。

candidate となったロボット r_i は重複時の位置 p_i と移動後の位置 p'_i において、 $dist(p_i, p_o) < dist(p'_i, p_o)$ となる移動方向をランダムに定めて移動する (図 3.11)。

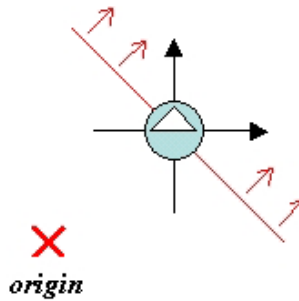


図 3.11: candidate の移動範囲

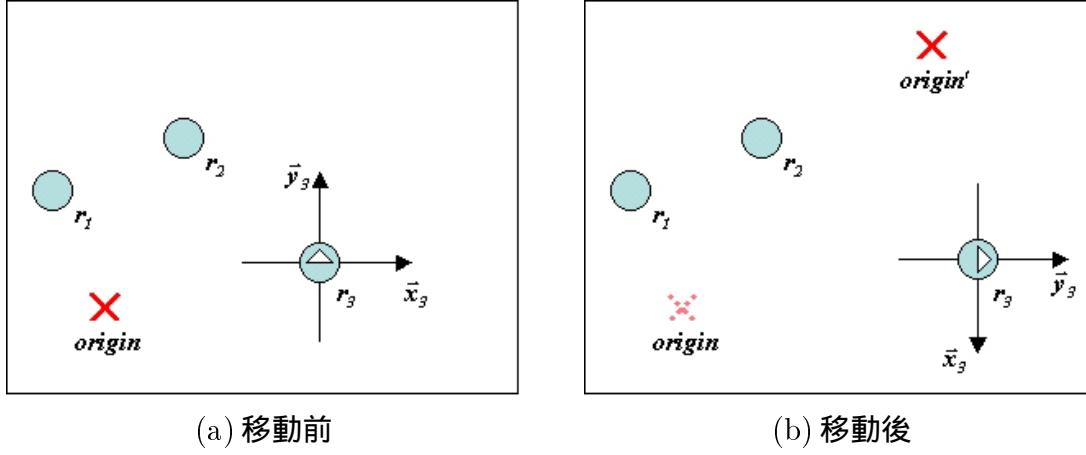


図 3.12: 移動に伴う origin の位置の変化

origin の位置の修正

各ロボット r_i は origin の位置 p_o をそれぞれのローカル座標における位置座標 $L_i[p_o]$ で記憶しているため、 r_i が移動をした際に $L_i[p_o]$ の示す位置は異なる地点となってしまふ。図 3.12 において、ロボット r_3 が記憶する origin の位置座標 $L_3[p_o]$ は、 r_3 の移動に伴う p_3 と $\vec{x}_3$ の変化によって本来の地点 p_o とは異なる地点を示してしまう。このため、方向転換角度と移動距離に応じて本来の origin の位置 になるよう修正する必要がある。式 (3.5) で方向転換によるズレを、式 (3.6) で移動によるズレを修正する (図 3.13)。

$$\begin{aligned}
 L_i[x_o] &= \cos(\text{angle}(\vec{x}_i, L_i[p_o]) - a) \times \text{dist}(p_i, L_i[p_o]) \\
 L_i[y_o] &= \sin(\text{angle}(\vec{x}_i, L_i[p_o]) - a) \times \text{dist}(p_i, L_i[p_o])
 \end{aligned} \tag{3.5}$$

$$\begin{aligned}
 L_i[x_o] &= L_i[x'_o] - \cos(-a) \times d \\
 L_i[y_o] &= L_i[y'_o] - \sin(-a) \times d
 \end{aligned} \tag{3.6}$$

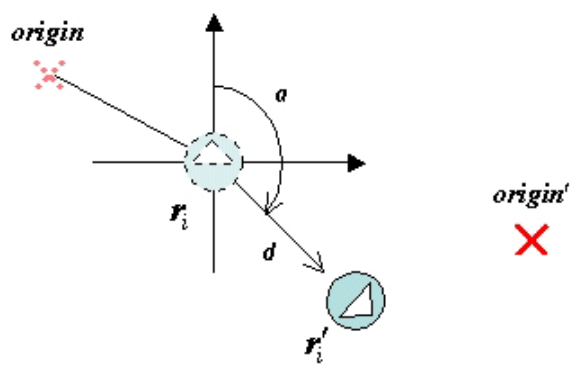


図 3.13: origin の位置の修正

3.7 結果

candidate 重複問題を考慮した leader 決定アルゴリズムのシミュレーション結果を図 3.14 に示す。図 3.14 の (a) は 4 台のロボットが candidate として重複する初期状態である。(b) において各ロボットは candidate の重複を発見し、(c) における candidate の移動を経て、(d) における leader の再決定によって leader の決定に成功している。

これにより、candidate が重複する際も leader を一意に決定することが可能である。

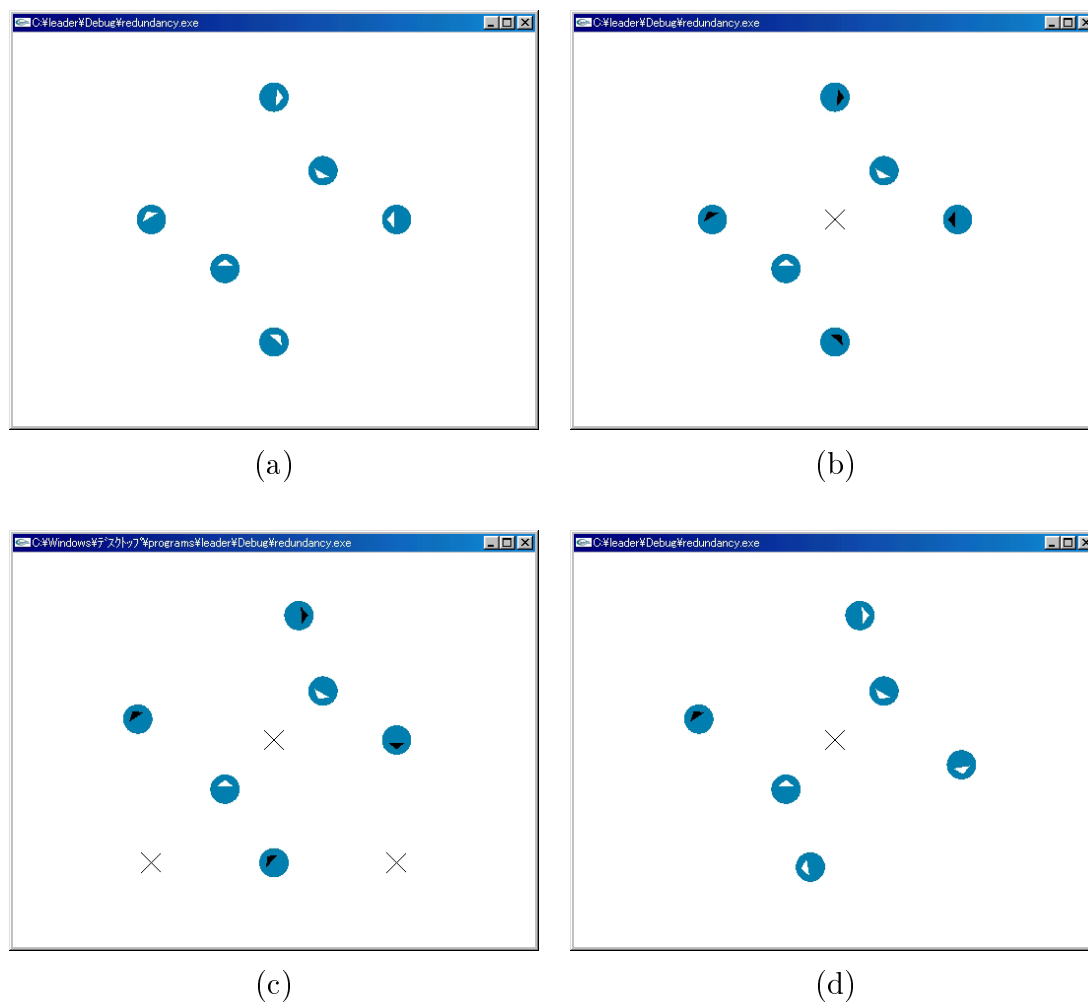


図 3.14: candidate 重複問題の解消

このように、本章で提案するアルゴリズムを用いることで、ランダムに配置された anonymous ロボット群の中から 1 台のロボットを leader として決定することが可能となった。また、leader は動的に決定することができるため、leader が故障した際は他のロボットの中から leader を再決定することが可能である。このように、ロボット群は leader の故障に対してロバストであるといえる。

しかし、全ロボットのセンサの計測結果が正確でなければ leader を一意に決定することは難しく、このアルゴリズムはセンサ能力に依存していることがわかる。今後はセンサの計測結果が正しいとは限らないと仮定し、無線通信によってセンサの誤差を補うアルゴリズムの開発が考えられる。

第4章 共通座標

4.1 共通座標を創造する目的

もし、30人の小学生がグラウンドで円形に並ぶとき、彼らはそれぞれの位置を相互に観察することで自らの移動距離と移動方向を決定し、円形に並ぶことができるであろう。しかし、5角形や6角形など形が複雑な場合、均質のとれた並び方をするのは困難となる。このとき、地面に形成目標とする形に沿って引かれた白線が予めあれば、子供たちはその白線上に移動するだけでどのような形の並びでも簡単に形成することができる。

このように、どのようにフォーメーションを形成するかという formation problem は、どのようにフォーメーションを形成するための位置を共通して認識するかという agreement problem と密接な関係にあり、agreement problem を解決することで formation problem を大幅に容易にすることができる [3]。このため、多くの研究において共通知識を得るための方法や、予め与えられた装置、共通知識の利用方法について提案されている。

多様なフォーメーションを形成するための最も効果的な共通知識の1つは、全ロボットに共通する座標である。図 4.1 に8台のロボットが quadrilateral を形成するために必要な座標上の位置の例を示した。図 4.1 において、座標上の点は quadrilateral を形成するために必要なロボット配置であり、これらの点 $(2, 2)$ 、 $(2, 0)$ 、 $(2, -2) \cdots (2, 0)$ に各ロボットが1台ずつ位置することで quadrilateral を形成することができる。全てのロボットが同じ座標を持つことで、これらの点が位置するフィールド上の地点はどのロボットにとっても全く同じとなるため、フォーメーションの形成が可能となる。このように、共通座標を持つことで様々なパターンを形成するために必要な位置の同意をロボット間で容易に行うことができる。

しかし、本研究のロボット群のように、各ロボットがそれぞれのローカル座標のみを持つ場合は、共通座標によるフォーメーション形成位置の同意ができない。本章においてこの問題を解決し、ローカル座標の原点、方向が異なるロボット群の共通座標の創造方法を提案する。

4.2 共通座標の創造方法

ロボットは座標の原点となる地点とある1方向によってその環境にある物体の位置座標を取得することができる (式 3.1)。同様に、全てのロボットが共通する原点と共通する方向を持つことで、その環境にある物体の座標は全てのロボットで一致する。

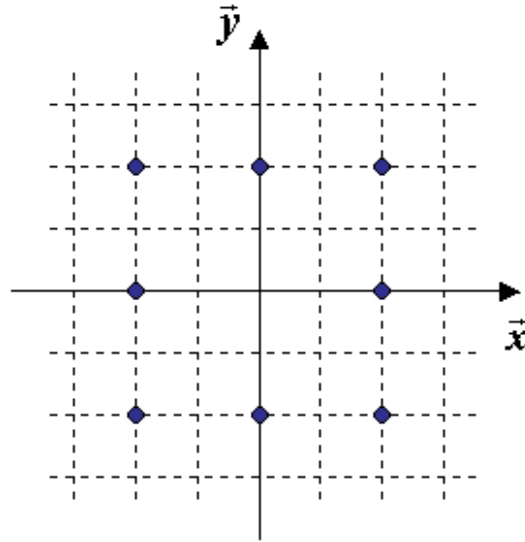


図 4.1: 8 台のロボットによる quadrilateral 形成に必要な位置

共通座標を創造するために必要となる共通原点は共通知識 origin としてその位置を得ることができている。共通方向の創造方法は次節においてその方法を述べる。

4.3 共通方向の創造

各ロボットが共通知識として全てのロボットに共通の方向を持っていることで、その方向を群れの移動時の移動方向や共通座標の軸方向として利用することができる。

共通方向を創造する簡単な方法として、ある特定のロボットが 1 方向に直線に動き、その移動した方向をその他のロボットが観察することで共通方向とすることができる。しかし、この方法は特定のロボットが決定されている必要がある。また、その特定ロボットの移動が必要であり、移動に伴う時間の浪費や特定ロボットが他のロボットに囲まれているなどロボットの配置に応じた高度な衝突回避能力が予測される。

また、その他の方法として compass などの装置がある。このような特別な装置は agreement problem の解決を大幅に容易にすることができるが、本研究ではこのような特別な装置に頼ることなくフォーメーションの形成を行う。

本研究では 2 つの共通地点を利用した共通方向の創造方法を提案する。図 4.2 において、ロボット群はその置かれた環境に $\text{flag}\#a$ と $\text{flag}\#b$ を持ち、全てのロボットは両方の flag のある位置を認識しているとする。このとき $\text{flag}\#a$ を始点、 $\text{flag}\#b$ を終点という取り決めが行われていれば、 $\text{flag}\#a$ から $\text{flag}\#b$ を結ぶ直線方向を全ロボットは共通方向として同意することが可能である。このように、双方が区別された共通地点が 2 つあるとき、共通方向を創造することができる。

これまでに各ロボットは origin と leader の位置を共通知識として持っているため、origin

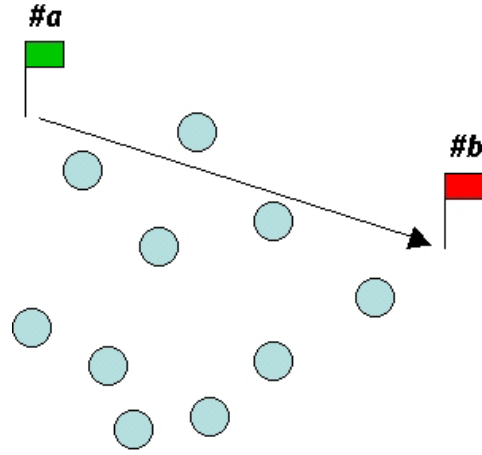


図 4.2: 2つの地点を利用した共通方向の創造

から leader を結んだ仮想直線を共通方向 (以下 $\vec{u}$ と表記) とすることが可能である (図 4.3)。

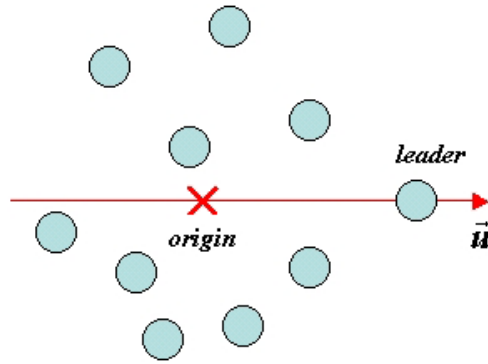


図 4.3: 共通原点と共通軸方向

共通方向を得るために、まず、各ロボットはそれぞれのローカル座標における origin と leader の位置座標 $L_i[p_o]$ 、 $L_i[p_l]$ から式 (3.4) を利用して $dist(p_o, p_l)$ を得る。そして、ローカル座標における $\vec{x}_i$ と $\vec{u}$ との角度差 $angle(\vec{x}_i, \vec{u})$ を式 (4.1) で得る (図 4.4)。これによって、 $\vec{u}$ がどの方向であるのかを認識し、全てのロボットに共通する方向とすることができる。

$$angle(\vec{x}_i, \vec{u}) = \arccos\left(\frac{L_i[x_i] - L_i[x_o]}{dist(p_o, p_l)}\right) \quad (4.1)$$

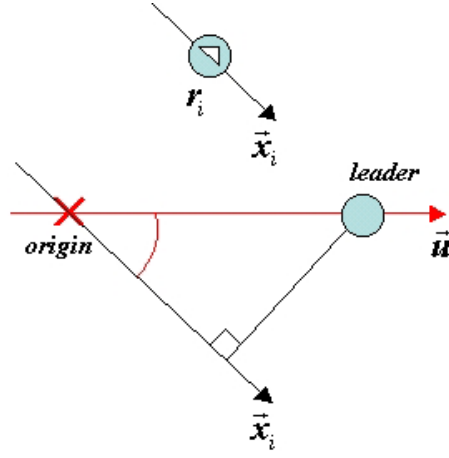


図 4.4: 共通方向への角度

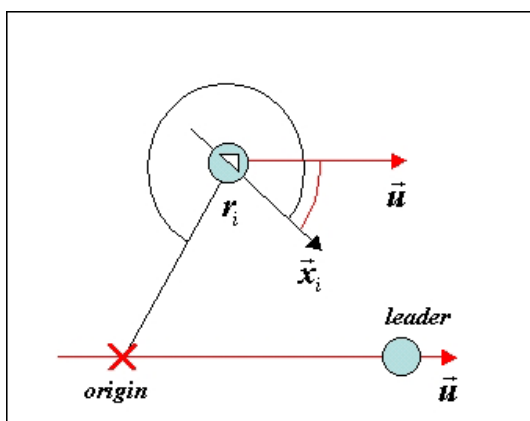
4.4 共通座標の創造

ロボット群は共通原点 *origin* と共通方向 $\vec{u}$ を利用することにより、共通座標を創造することが可能である。

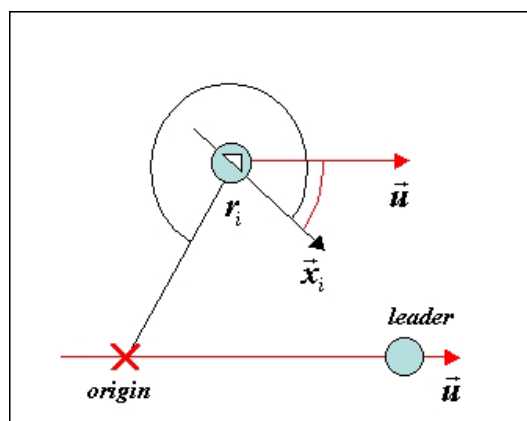
各ロボット r_i は $\vec{x}_i$ から *origin* までの角度 $angle(\vec{x}_i, p_o)$ と $\vec{x}_i$ から $\vec{u}$ までの角度 $angle(\vec{x}_i, \vec{u})$ 、*origin* の位置 p_o を利用し、式 (4.2) によって共通座標における自身の位置 p_i の位置座標 $W_i[p_i] = (u_i, v_i)$ を得る (図 4.5 (a))。また、式 (4.3) によって共通座標における他ロボット r_j の位置座標 $W_i[p_j] = (u_j, v_j)$ を得る (図 4.5 (b))。

$$\begin{aligned} u_i &= -\cos(angle(\vec{x}_i, p_o) - angle(\vec{x}_i, \vec{u})) \times dist(p_i, p_o) \\ v_i &= -\sin(angle(\vec{x}_i, p_o) - angle(\vec{x}_i, \vec{u})) \times dist(p_i, p_o) \end{aligned} \quad (4.2)$$

$$\begin{aligned} u_j &= \cos(angle(\vec{x}_i, p_j) - angle(\vec{x}_i, \vec{u})) \times dist(p_i, p_j) - u_i \\ v_j &= \sin(angle(\vec{x}_i, p_j) - angle(\vec{x}_i, \vec{u})) \times dist(p_i, p_j) - v_i \end{aligned} \quad (4.3)$$



(a) r_i の位置座標の算出



(b) r_j の位置座標の算出

図 4.5: 共通座標の算出

4.5 結果

図 4.6 に示す 3 台の自律移動ロボットがそれぞれ共通座標の創造を行った結果を表 4.1 に示す。表 4.1 における A、B、C は各ロボットを表しており、図 4.6 における左上のロボットが A、左下のロボットが B、右のロボットが C である。× (黒) は各ロボットのローカル座標における origin の位置を、+ (白) は leader の位置を表している。

表 4.1 に示した結果から、全てのロボットは共通座標におけるそれぞれの位置座標を共通知識として同意していることがわかる。このように、本章で提案したアルゴリズムによってロボット群は共通座標を創造することが可能となった。

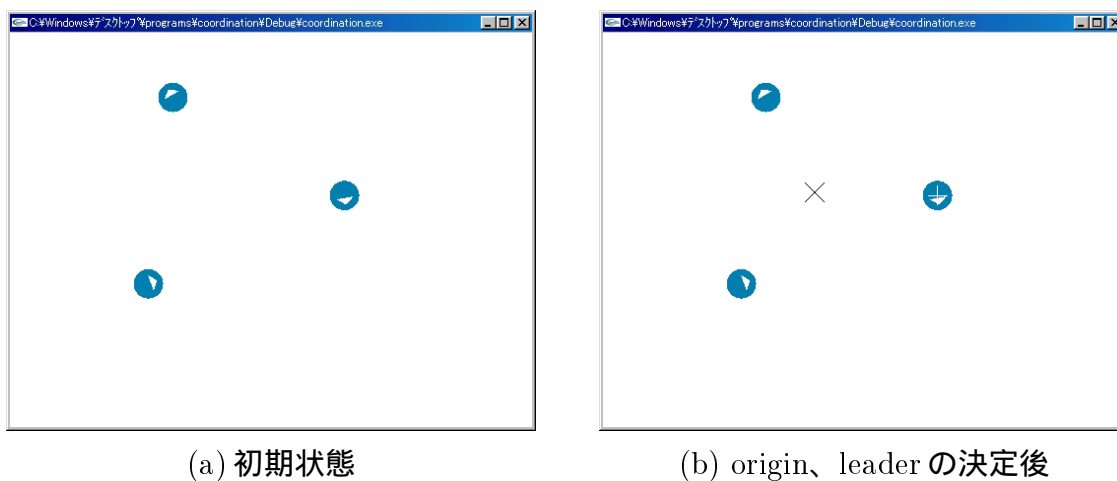


図 4.6: 共通座標の創造シミュレーション

	ローカル座標					共通座標		
	A	B	C	origin	$\text{angle}(\vec{x}, \vec{u})$	A	B	C
A	0.0000, 0.0000	-4.6602, -6.0817	4.0621, -6.9641	-0.2012, 4.3486	328.4724	-2.1023, 3.8119	-2.8994, -3.8119	5.0017, 0.0000
B	-6.7996, 3.5390	0.0000, 0.0000	-0.6467, 8.7488	-2.4821, 4.0959	68.4724	-2.1023, 3.8119	-2.8994, -3.8119	5.0017, 0.0000
C	6.1990, -5.1547	8.5035, 2.1561	0.0000, 0.0000	4.9008, 0.9995	168.4724	-2.1023, 3.8119	-2.8994, -3.8119	5.0017, 0.0000

表 4.1: 共通座標の創造結果

第5章 ID 番号

5.1 ID を決定する目的

全校生徒が校庭で六角形など複雑な形状に並ぶことを考えたとき、地面に引かれた六角形状のラインを共通知識とすることでこの問題を大幅に容易にすることが可能であることを4章で述べた。しかしこのとき、ライン上のどの地点に誰が位置するかという問題が起きる。人間であればそれぞれの動作の観察による自身の動作の決定やコミュニケーションによる位置の主張、譲渡を通してこの問題を解決することができる。全校生徒が何かしらの形状に並ぶときは、問題を容易にするために移動する位置をクラス単位で予め取り決めておくことが行われる。

フォーメーション形成を行うロボット群についても同様に、フォーメーションパターンにおける頂点や辺などの役割をフォーメーション形成前に予め決定しておくことで各ロボットは役割に応じた行動をすることが可能となり、フォーメーション形成を容易にすることができる。図5.1は各ロボットが役割に応じた位置に移動することで quadrilateral を形成する例である。

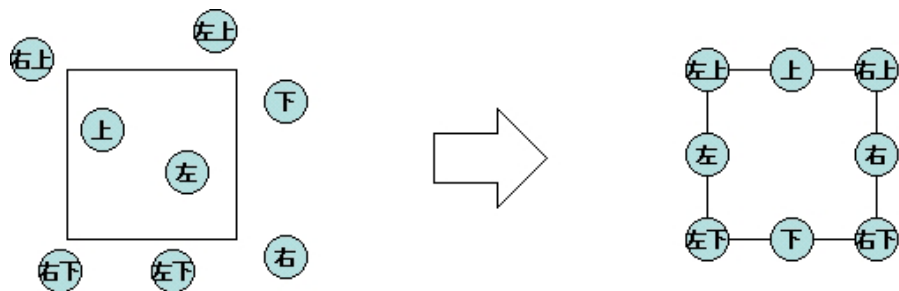


図 5.1: ロボットごとの役割とフォーメーション形成

共通知識と役割を利用してフォーメーション形成を行う研究に [10][11] がある。池本らは circle を形成することで circle の中心点からそれぞれまでの距離を共通知識とし、その状態からの相互通信によってフォーメーションパターンにおける頂点、辺の役割をロボット毎に決定している。各ロボットは役割に応じて移動して circle の中心点からの距離を変化させることで triangle, pentagon などのフォーメーションパターンを形成している [10]。

Fredslund & Mataric は conductor と呼ばれる特定ロボットの移動方向を共通方向とし、予め各ロボットが持つ固有の ID とその ID に応じた friend と呼ばれる特定ロボットとの角

度を移動によって変化させることで line, column, wedge などのフォーメーションパターンを形成する [11]。

しかし、池本らの研究では circle パターンを形成した状態でなければ役割を決定することができないため、circle を形成するための別のアルゴリズムが必要である。また、あるフォーメーションパターンから他のパターンにダイナミックに再編成することができない [10]。Fredslund & Mataric の研究では各ロボットの ID が固定されているため、ロボットの故障に伴う ID の欠如に対する耐故障性が必要である。また、各ロボットは相互の ID を把握するための特別な装置を必要としている [11]。

本研究では、図 2.4 のように、共通座標を ID と共に用いることでフォーメーションの形成を行う。共通座標があることで、どの地点にロボットが位置すればよいかをフォーメーションパターンに応じて決定することができる。ID があることで、各ロボットがフォーメーション形成に必要な地点のそれぞれどこに位置するかを ID に応じて決定することができる。これにより、多様なフォーメーションパターンの形成が可能である。ID を決定する方法を次節で述べる。

5.2 ID の決定方法

ID の決定は、共通座標における各ロボットの位置座標を利用して行う。まず、各ロボットは以下によってそれぞれの順位を獲得する。

- ロボット r_i は共通座標 $\vec{u}$ 軸方向における自身の位置 $W_i[u_i]$ を他のロボット r_j の位置座標 $\{W_i[u_j] \mid 1 \leq j \leq N\}$ と比較し、値が大きい順位を得る。
- $u_i = u_j$ のときは v_i と v_j を比較し、値の大きい方が優先して順位を得る。

この順位から 1 だけ減算した値を ID とする (図 5.2)。これは、leader が ID の獲得を行わず (図 2.6)、また、leader は u_l の値が最大となっているからである (3 章を参照)。

5.3 ID 決定方法への制約

初期状態からのフォーメーション形成や、あるフォーメーションパターンから別のパターンへの再編成を行う際は、これらにかかる時間や各ロボットの移動距離、ロボット間の衝突機会が少ないほど望ましい。このようなとき、フォーメーションの形成とフォーメーションの再編成は容易であるといえる。

本研究ではフォーメーションの再編成を容易に行うため、各ロボットの ID に応じたフォーメーション形成位置が全フォーメーションパターンで類似するよう以下を全フォーメーションパターンに共通した ID の役割としている (6 章を参照)。

- ID が偶数のロボットは共通座標 $\vec{v}$ 軸プラス上に、ID が奇数のロボットは $\vec{v}$ 軸マイナス上に位置する。

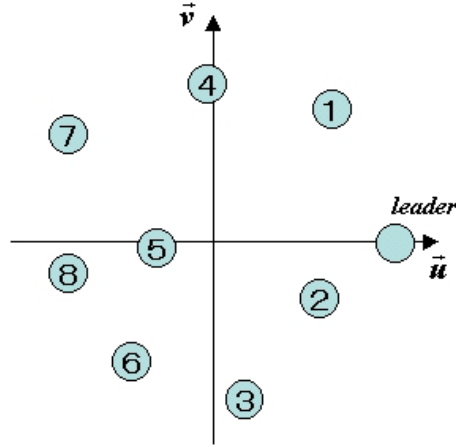


図 5.2: ロボット配置と ID

- ID の小さいロボットほど共通座標 $\vec{u}$ 軸上の値が大きくなるよう位置する。

図 5.3 に circle と quadrilateral の ID に応じたロボット配置を示した。全てのフォーメーションパターンに共通した ID の役割によって、両フォーメーションパターンにおける ID 配置は類似している。このため、フォーメーションの再編成を容易に行うことが可能となる。

各ロボットがランダムに位置している初期状態からのフォーメーション形成においても同様に、容易に行うためにはランダム状態にあるロボットの ID を図 5.3 に示した各フォーメーションパターンの ID 配置と類似させて決定する必要がある。つまり、各ロボットの ID は図 5.2 のような配置ではなく、図 5.4 のような配置になるよう決定することで図 5.3 に示したフォーメーションの形成を容易に行うことが可能である。このため、ランダムに位置したロボット群の ID 決定方法に以下の制約をつける。

- 共通座標 $\vec{v}$ 軸プラス上のロボットは偶数の ID を、 $\vec{v}$ 軸マイナス上にあるロボットは奇数の ID を取得する。
- 共通座標 $\vec{u}$ 軸上の値が大きいロボットほど低い ID を取得する。

5.4 ID の決定

各ロボットは以下のようにして ID の決定を行う。

- ロボット r_i は共通座標における自身の位置座標を取得し、 $\vec{v}$ 軸プラス上に位置していれば $\vec{v}$ 軸プラス上にある全ロボットの中での、 $\vec{v}$ 軸マイナス上に位置していれば $\vec{v}$ 軸マイナス上にある全ロボットの中での自分の順位を算出する (図 5.5)。

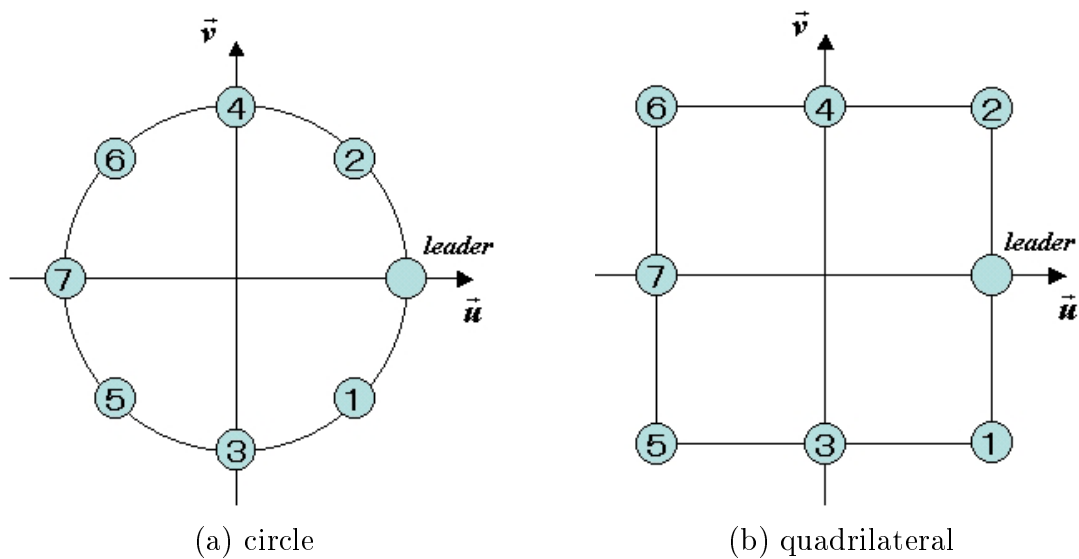


図 5.3: フォーメーションパターンと ID の配置

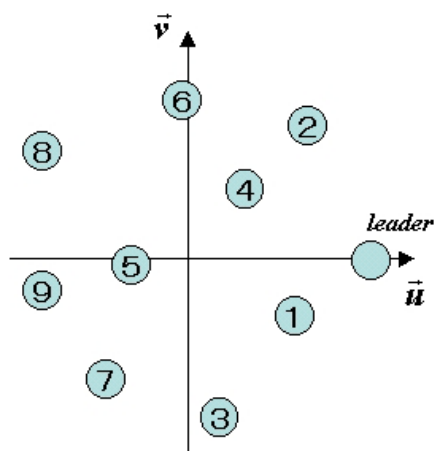


図 5.4: 望ましいロボット配置と ID

- $\vec{v}$ 軸プラス上のロボットは順位 $\times 2 - 2$ 、 $\vec{v}$ 軸マイナス上のロボットは順位 $\times 2 - 1$ によって ID を得る。

順位の決め方は 5.2 節で述べた方法と同じである。これによって、ランダムに配置されたロボットに付与される ID の分布は図 5.4 の状態ようになる。

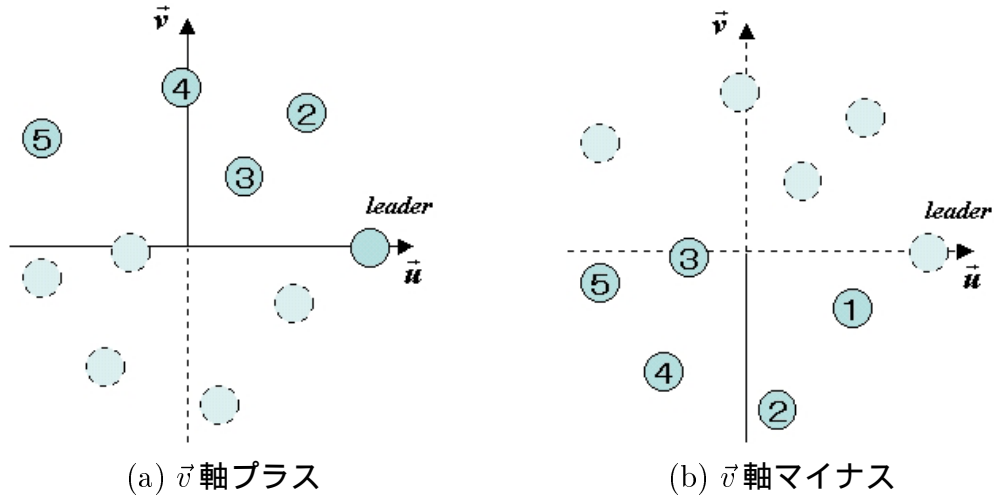


図 5.5: 順位

しかし、実際には図 5.5 のように $\vec{v}$ 軸プラス上のロボット数と $\vec{v}$ 軸マイナス上のロボット数が一致するとは限らず、図 5.6 のような状態がありえる。この状態において上述した方法で ID の決定を行うと ID は 1, 2, 3, 4, 6, 8, 10 となり、連番とならない。これを回避するため、以下を付け加える。

- $\vec{v}$ 軸プラス上に位置し、順位が $\vec{v}$ 軸マイナス上のロボットの数+2 以上のロボットは 順位+1 を ID とする (図 5.7 の点線内)。

これ以外のロボットは通常の ID 決定方法を用いることで、各ロボットはそれぞれに固有の連番となった ID を得ることができる (図 5.7)。 $\vec{v}$ 軸マイナス上のロボット数が $\vec{v}$ 軸プラス上のロボット数よりも多い場合においても同様の方法によって ID の決定を行う。

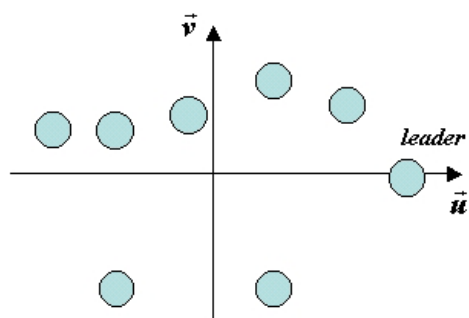


図 5.6: ロボット配置の不均衡

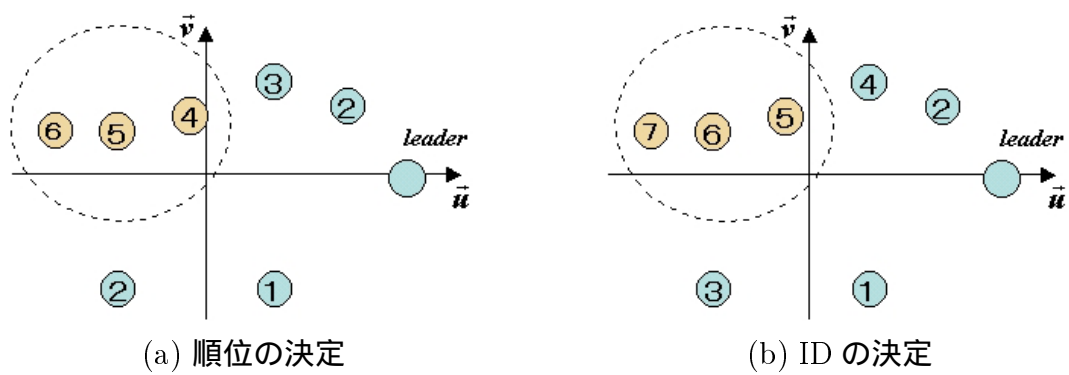


図 5.7: ロボット配置と ID 決定

5.5 結果

図 5.8 に示すランダムな初期状態にある 15 台のロボットを用い、ID 決定アルゴリズムのシミュレーションを行った。シミュレーションにおける各パラメータを表 5.1 に示す。

ID の決定は 5.2 節で述べた改良前のアルゴリズムと、5.4 節で述べた改良後のアルゴリズムを実行し、circle の形成を行うことで両者の比較を行った。表 5.2 にその結果を示す。また、改良前のアルゴリズムによる circle の形成過程を図 5.9 に、改良後のアルゴリズムによる形成過程を図 5.10 に示す。circle の形成は 6 章で述べるアルゴリズムに従っている。図中において origin 位置を示す × が多数存在することがあるが (図 5.9 の 1.00(s) など)、これは各ロボットの方向転換に伴う origin 位置の移動を示すものである (3.4.1 節を参照)。

表 5.2 に示した結果から、各フォーメーションパターンに共通して定められた ID 配置の制約と類似するように ID 決定を行った改良後のアルゴリズムにより、フォーメーションの形成を大幅に容易にすることが可能となったことがわかる。

このように、本章で提案したアルゴリズムによって 1 から N までの重複のない ID 番号を各ロボットがそれぞれ獲得することが可能である。この ID 番号は動的に決定できる可変 ID であり、ロボットの故障によって ID の欠如が起きる問題を ID の再決定によって回避することができる。このため、ロボットの故障に対してロバストであるといえる。

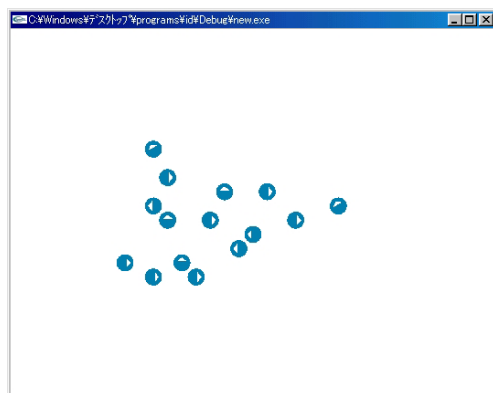


図 5.8: 初期状態

直径	直進速度	回転速度	衝突回避距離
1.2	0.03	2	1.6

表 5.1: シミュレーションパラメータ

N=15	総移動距離	平均移動距離	時間 (s)	衝突回避機会
改良前	93.330	6.222	7.75	19
改良後	57.498	3.833	4.26	0

表 5.2: circle 形成における ID 決定アルゴリズムの比較

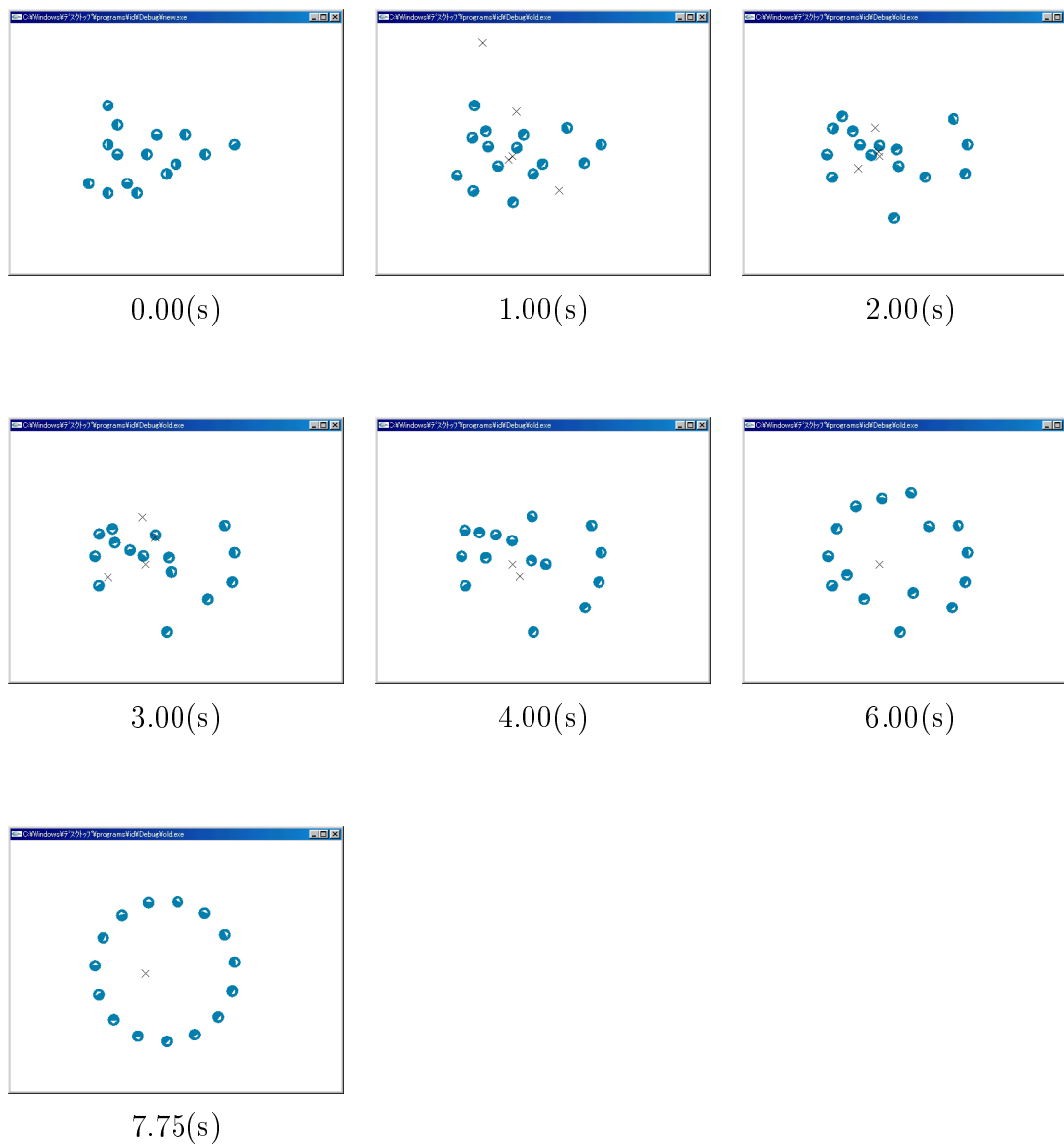


図 5.9: 改良前の ID 決定アルゴリズムによる circle 形成

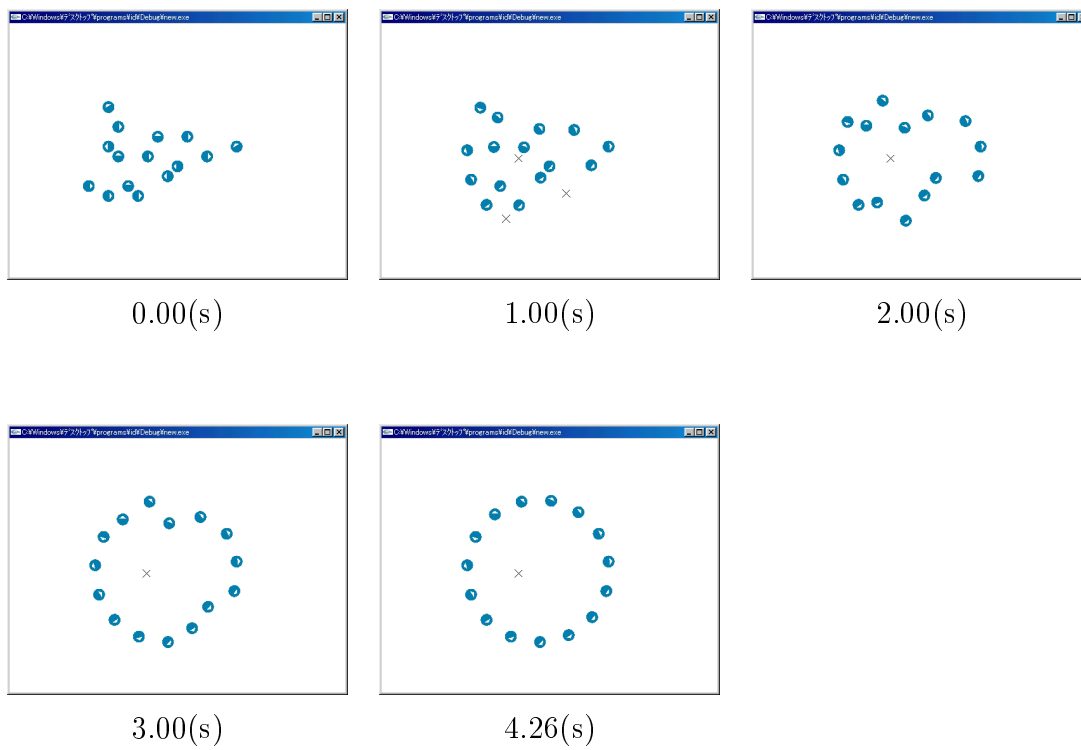


図 5.10: 改良後の ID 決定アルゴリズムによる circle 形成

第6章 フォーメーションパターン

6.1 フォーメーションパターンの形成

6.1.1 各ロボットのフォーメーション形成位置

各ロボットはフォーメーション形成位置を共通座標と ID 番号を利用して決定する。

各ロボット r_i は形成するフォーメーションパターン、群れのロボット数 N 、ID 番号 ID_i に応じて共通座標におけるフォーメーション形成のための位置 p'_i を取得し、その位置に移動 (6.1.2 節を参照) することでフォーメーションの形成を達成する。

図 5.4 に示したように全フォーメーションパターンに共通して ID の配置を類似させることで、フォーメーション形成における各ロボットの移動距離や衝突機会を減らすことができ、フォーメーションの形成と再編成を容易に行うことが可能となる。

このため、フォーメーションにおける ID 配置は全てのフォーメーションパターンにおいて以下の共通した制約を設ける。

- ID が偶数のロボットは共通座標 $\vec{v}$ 軸プラス上に、ID 番号が奇数のロボットは共通座標 $\vec{v}$ 軸マイナス上に位置する。
- ID が低いロボットほど共通座標 $\vec{u}$ 軸上の値が大きい地点に位置する。

また、フォーメーションを維持した移動 (6.3 節) を leader-followers 方式で行うため、leader は群れの先頭に位置しなければならない。このため、全フォーメーションパターンにおいて以下の共通した制約を設ける。

- leader は共通座標 $\vec{u}$ 軸上の値が最大となる地点に位置する。

以上の制約のもと、本研究では多様なフォーメーションパターンの形成を可能とした。本稿において wedge, circle, arc, triangle, diamond, pentagon, hexagon, arrow, cross, line, cross の 11 種類のフォーメーションパターンの形成方法を述べる。それぞれのフォーメーションパターンを形成する ID に応じた各ロボットの配置を記す。

Wedge

5 台のロボットによる wedge の形成位置を図 6.1 に示す。

式 (6.1) において u_l は各ロボット r_i が認識する leader の共通座標における位置座標 $W_i[p_l]$ の $\vec{u}$ 軸上の値、DIST は予め与えられているロボット同士の間隔である。

$$\begin{aligned} u'_i &= u_l - \cos(\theta) \times DIST \times factor \\ v'_i &= \sin(\theta) \times DIST \times factor \times mark \end{aligned} \quad (6.1)$$

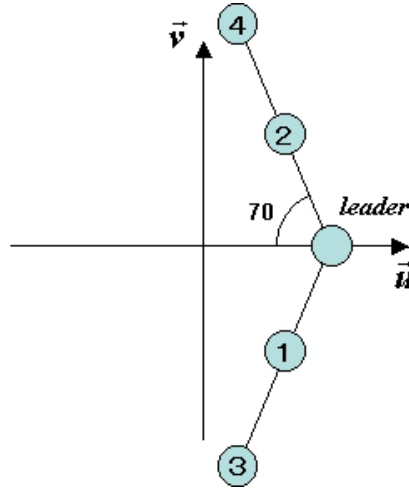


図 6.1: Wedge

Algorithm 6.1 wedge におけるロボット r_i の位置 $p'_i = (u'_i, v'_i)$

- 1: **if** $ID_i \in$ 偶数 **then** $mark := 1$
 - 2: **else** $mark := -1$
 - 3: $factor := \lfloor (ID_i + 1)/2 \rfloor$
 - 4: $\theta :=$ wedge の頂点角度/2 にあたる 70 とする。
 - 5: $u'_i, v'_i :=$ 式 (6.1) を利用して得る。
-

Circle

8 台のロボットによる circle の形成位置を図 6.2 に示す。

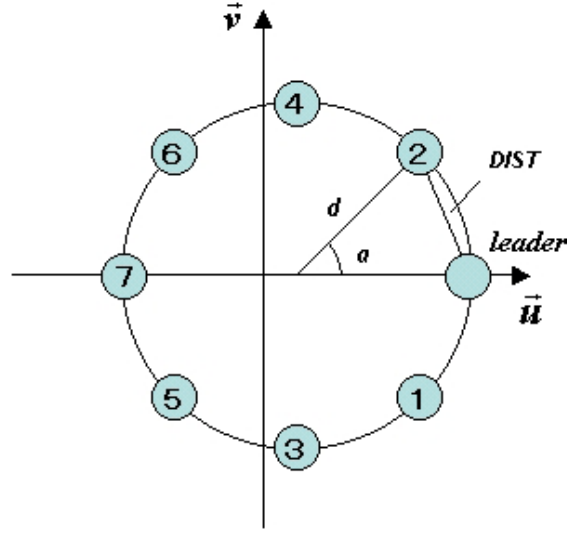


図 6.2: Circle

Algorithm 6.2 circle におけるロボット r_i の位置 $p'_i = (u'_i, v'_i)$

- 1: if $ID_i \in$ 偶数 then $mark := 1$
 - 2: else $mark := -1$
 - 3: $\vec{u}$ 軸方向からの角度 (図 6.2 の a) $:= 360/N \times \lfloor (ID_i+1)/2 \rfloor \times mark$
 - 4: 半径 (図 6.2 の d) $:= (DIST/2) / \cos((180 - (360/N))/2)$
 - 5: フォーメーションは leader の位置を基準とするため、circle は $u_l - d$ だけ $\vec{u}$ 軸方向にずれる。このズレを c とする。
 - 6: $u'_i := \cos(a) \times d + c$
 - 7: $v'_i := \sin(a) \times d$
-

Arc

各ロボット r_i は arc の形成位置 p'_i の座標を、 N の値を数倍した仮想ロボット数を用いた circle アルゴリズムによって得る。図 6.3 に 5 台のロボットによる arc の形成位置を示す。ここでは仮想ロボット数を N の 3 倍としており、それぞれのロボットの arc 形成位置は $N = 15$ のときの circle 形成位置と一致している。

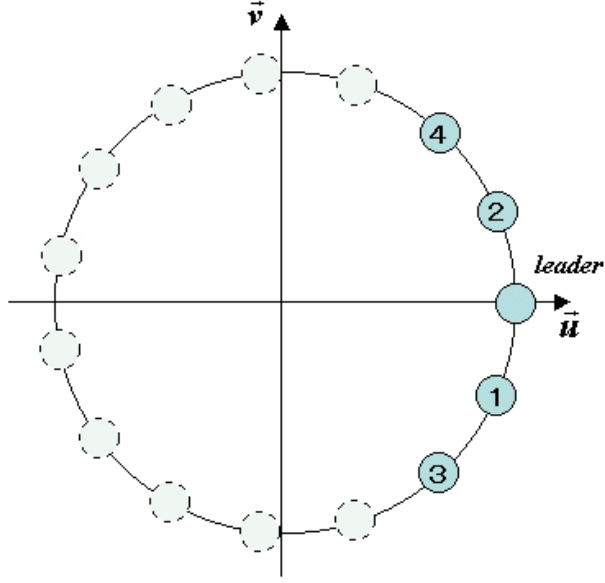


図 6.3: Arc

Algorithm 6.3 arc におけるロボット r_i の位置 $p_i' = (u_i', v_i')$

- 1: $N \times 3$ を仮想ロボット数とする。
 - 2: Algorithm 6.2 を実行する。
-

Triangle

図 6.4 に 6 台のロボット数による triangle 形成位置を示す。図 6.4 からわかるように、triangle の形成では各ロボットの位置する辺の方向が end の位置の ID である ID_e を境に変化することがわかる。このため、各ロボット r_i は ID_i が $ID_i \leq ID_e$ か $ID_i > ID_e$ に応じて triangle を形成する位置座標の決定を行う必要がある。

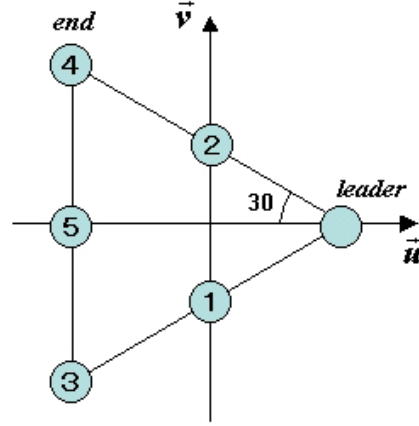


図 6.4: Triangle(1)

Algorithm 6.4 triangle におけるロボット r_i の位置 $p'_i = (u'_i, v'_i)$

- 1: if $ID_i \in$ 偶数 then $mark := 1$
 - 2: else $mark := -1$
 - 3: $\theta :=$ 頂点角度の半分である 30 とする。
 - 4: $factor := \lfloor (ID_i + 1)/2 \rfloor$
 - 5: end となる ID 番号 $ID_e := N - N/3$
 - 6: if $ID_i \leq ID_e$ then
 - 7: $u'_i, v'_i :=$ 式 6.1 を利用して得る。
 - 8: else
 - 9: $u'_i := u'_e$
 - 10: $v'_i := v'_e - (ID_e - ID_i) \times DIST$
-

上述した triangle 形成位置の算出方法は $N/3$ の余りが 0 となるときのみ有効である。余りが 0 以外のときにおいても均整のとれた triangle を形成する ID 配置となるよう座標取得式を与える必要がある。図 6.5 は $N/3$ の余りが 1 や 2 となるときの ID 配置である。本研究では end の算出式と end 前後の辺のロボット間隔を $N/3$ の余りに応じて変化させることで均整のとれた triangle の形成を行っている。

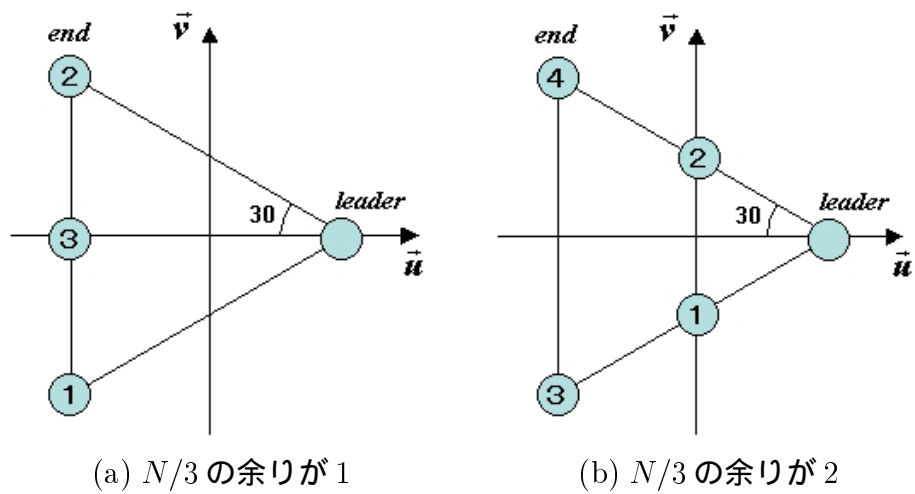


図 6.5: Triangle(2)

Diamond

図 6.6 に 8 台のロボットによって diamond を形成する際の ID の配置を示す。triangle と同様に、各ロボットの位置する辺の方向が end の位置の ID である ID_e を境に変化することがわかる。このため、各ロボット r_i は ID_i が $ID_i \leq ID_e$ か $ID_i > ID_e$ に応じて diamond を形成する位置座標の決定を行う必要がある。

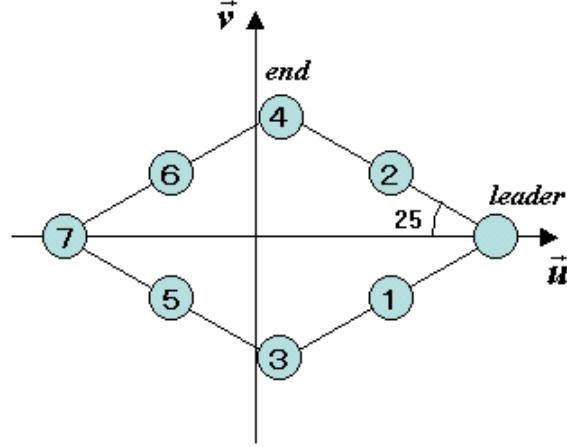


図 6.6: Diamond

Algorithm 6.5 diamond におけるロボット r_i の位置 $p_i' = (u_i', v_i')$

- 1: if $ID_i \in$ 偶数 then $mark := 1$
 - 2: else $mark := -1$
 - 3: $\theta :=$ 頂点角度の半分である 25 とする。
 - 4: $factor := \lfloor (ID_i + 1)/2 \rfloor$
 - 5: end となる ID 番号 $ID_e := N/2$
 - 6: $u_i' :=$ 式 6.1 を利用して得る。
 - 7: if $ID_i \leq ID_e$ then
 - 8: $v_i' :=$ 式 (6.1) を利用して得る。
 - 9: else
 - 10: $factor := ID_e/2 - \lfloor ((ID_i + 1) - ID_e)/2 \rfloor$
 - 11: $v_i' :=$ 式 (6.1) を利用して得る。
-

diamond の ID 配置においても triangle と同様、 $N/4$ の余りに応じて均整のとれた形になるよう end と各辺のロボット間隔を調整する必要がある。

/clearpage

Pengaton

図 6.7 は 10 のロボットにおける pentagon の ID 配置である。図 6.7 から、pentagon の形成では ID は half 位置の ID である ID_h と end 位置の ID である ID_e を境にして所属する辺の方向が変わることがわかる。このため、各ロボット r_i は ID_i の所属が $ID_i \leq ID_h$ 、 $ID_h < ID_i \leq ID_e$ 、 $ID_e < ID_i$ のいずれであるかに応じて pentagon を形成する位置座標の決定を行う必要がある。

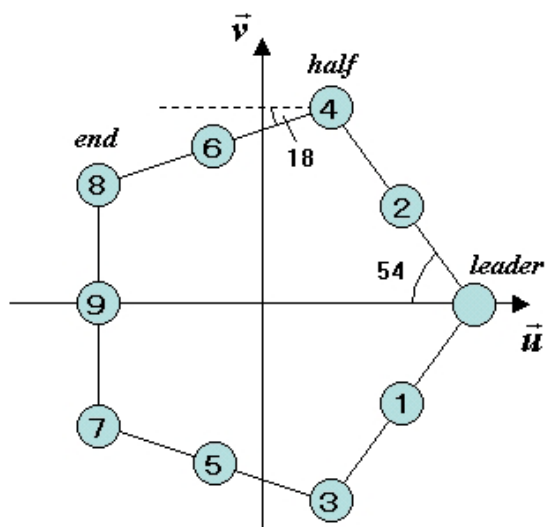


図 6.7: Pentagon

Algorithm 6.6 pentagon におけるロボット r_i の位置 $p'_i = (u'_i, v'_i)$

```
1: if  $ID_i \in$  偶数 then  $mark := 1$ 
2: else  $mark := -1$ 
3: half 位置の ID 番号  $ID_h := N/5 \times 2$ 
4: end 位置の ID 番号  $ID_e := N/5 \times 4$ 
5: if  $ID_i \leq ID_h$  then
6:    $factor := \lfloor (ID_i + 1)/2 \rfloor$ 
7:    $\theta :=$  頂点角度の半分である  $54$  とする。
8:    $u'_i, v'_i :=$  式 6.1 を利用して得る。
9: else if  $ID_i \leq ID_e$  then
10:   $factor := (ID_i + 1 - ID_h)/2$ 
11:   $\theta := 18$ 
12:   $u'_i :=$  式 6.1 を利用し、基準点となっている  $u_l$  を half の  $\vec{u}$  軸
    上の位置  $u_h$  に変えた結果を得る。
13:   $v'_i := v_h \times mark$  に式 (6.1) を利用して得た  $v'_i \times -mark$  を加
    算した結果を得る。
14: else
15:   $u'_i :=$  end の  $\vec{u}$  軸上の位置  $u_e$ 
16:   $v'_i :=$  end の  $\vec{v}$  軸上の位置  $v_e + ((ID_i - ID_e + 1)/2) \times DIST \times$ 
     $-mark$ 
```

上記は $N/5$ の余りが 0 となる場合の p'_i の計算である。余りが 0 とならないときも、その余りの数に応じて half、end の値を変えることで均整のとれた pentagon となるようにする必要がある。

Hexagon

図 6.8 は 12 台のロボットによる hexagon の ID 配置である。図 6.8 から、hexagon の形成では pentagon と同様、ID は half の ID である ID_h と end の ID である ID_e を境にして所属する辺の方向が変わることがわかる。このため、所属する辺に応じて hexagon を形成する位置座標の決定を行う必要がある。

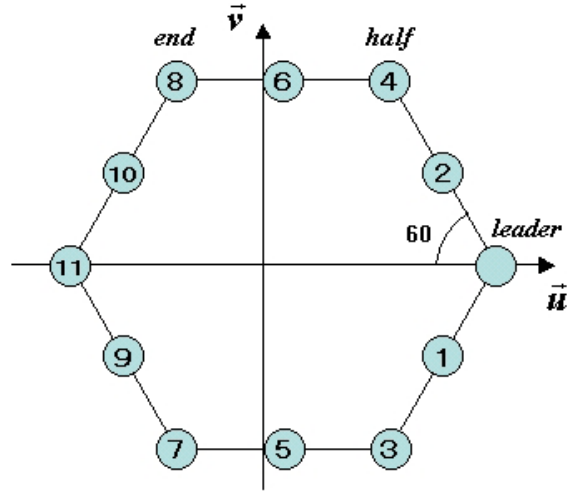


図 6.8: Hexagon

Algorithm 6.7 hexagon におけるロボット r_i の位置 $p'_i = (u'_i, v'_i)$

- 1: if $ID_i \in$ 偶数 then $mark := 1$
 - 2: else $mark := -1$
 - 3: half となる ID 番号 $ID_h := N/3$
 - 4: end となる ID 番号 $ID_e := N \times 2/3$
 - 5: if $ID_i \leq ID_h$ then
 - 6: $\theta :=$ 頂点角度の半分である 60 とする。
 - 7: $factor := \lfloor (ID_i + 1)/2 \rfloor$
 - 8: $u'_i, v'_i :=$ 式 (6.1) を利用して得る。
 - 9: else if $ID_i \leq ID_e$ then
 - 10: $factor := \lfloor (ID_i + 1 - ID_h)/2 \rfloor$
 - 11: $\theta := 18$
 - 12: $u'_i := u_h - DIST \times factor$
 - 13: $v'_i := v_h$
 - 14: else
 - 15: $u'_i := factor$ を $\lfloor (ID_i + 1 - ID_e)/2 \rfloor$ とし、式 (6.1) で得た u'_i を u_e から引く。
 - 16: $v'_i := factor$ を $\lfloor (N - ID_i)/2 \rfloor$ とし、式 (6.1) によって得る。
-

Arrow

図 6.9 は 10 台のロボットによる arrow の ID 配置である。triangle と同様に end を設け、ロボット r_i は自身の ID である ID_i と end に位置する ID_e から、以下のように arrow を形成する位置 p'_i を得る。

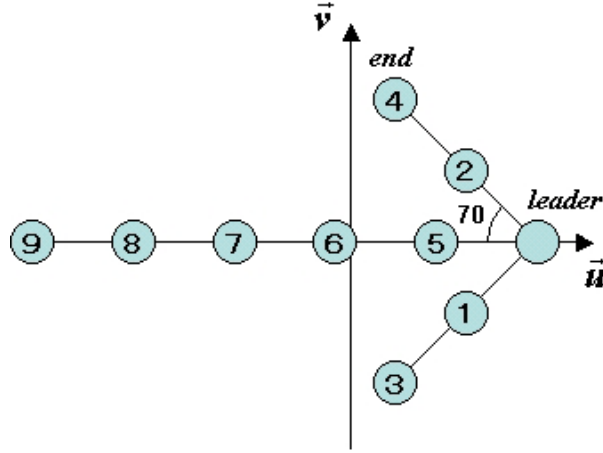


図 6.9: Arrow

Algorithm 6.8 arrow におけるロボット r_i の位置 $p'_i = (u'_i, v'_i)$

- 1: if $ID_i \in$ 偶数 then $mark := 1$
 - 2: else $mark := -1$
 - 4: end となる ID 番号 $ID_e := N \times 2/3$
 - 5: if $ID_i \leq ID_e$ then
 - 6: $\theta :=$ 頂点角度の半分である 45 とする。
 - 7: $factor := \lfloor (ID_i + 1)/2 \rfloor$
 - 8: $u'_i, v'_i :=$ 式 (6.1) を利用して得る。
 - 9: else
 - 10: $factor := ID_i - ID_h$
 - 12: u'_i 式 (6.1) を利用して得る。
 - 13: $v'_i := 0$
-

Cross

図6.10 に9 台のロボットにおける cross の ID 配置を示す。cross では pentagon、hexagon と同様に N に応じて end となる ID である ID_e と half となる ID_h を求め、各ロボットは自身の ID である ID_i とこれらの ID の大小関係によって cross を形成する位置 p'_i を求める。

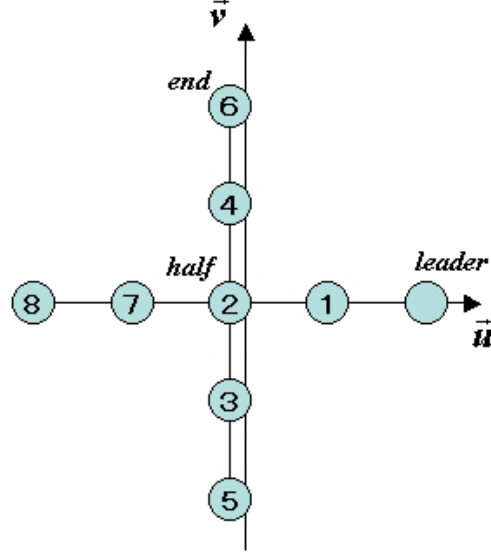


図 6.10: Cross

Algorithm 6.9 cross におけるロボット r_i の位置 $p'_i = (u'_i, v'_i)$

- 1: if $ID_i \in$ 偶数 then $mark := 1$
 - 2: else $mark := -1$
 - 3: half となる ID 番号 $ID_h := \lfloor N/4 \rfloor$
 - 4: end となる ID 番号 $ID_e := ID_h \times 3$
 - 5: if $ID_i \leq ID_h$ then
 - 6: $u'_i := u_l - ID_i \times DIST$
 - 7: $v'_i := 0$
 - 8: else if $ID_i \leq ID_e$ then
 - 9: $u'_i := u_e$
 - 10: $v'_i := (ID_i - ID_h + 1)/2 \times DIST \times mark$
 - 11: else
 - 12: $u'_i := u_e - (ID_i - ID_e + ID_h) \times DIST$
 - 13: $v'_i := 0$
-

Line

図 6.11 において、(a) は 8 台のロボットによる 2 列 line の形成位置、(b) は 12 台のロボットによる 3 列 line の形成位置である。line では形成列数 $\leq N$ であれば何列でも形成可能である。

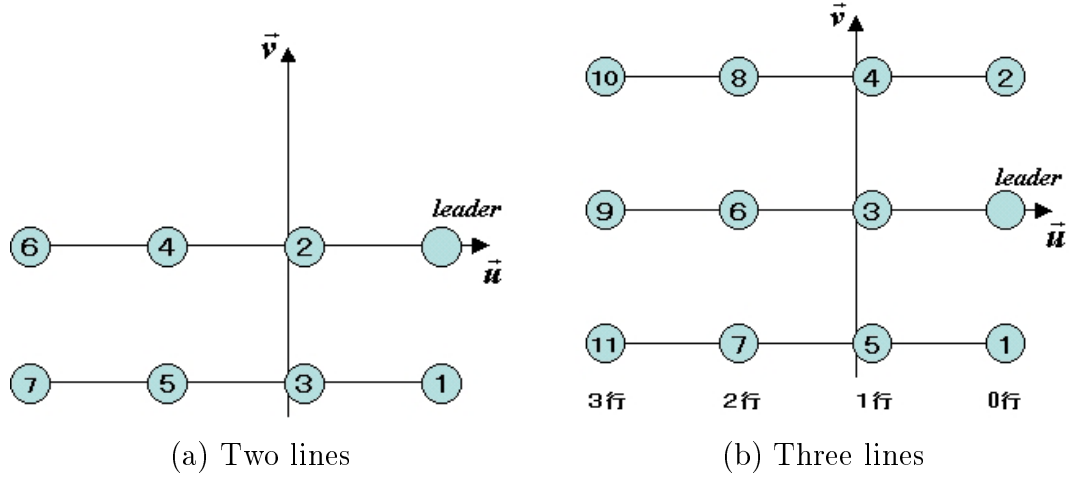


図 6.11: Line

Algorithm 6.10 line におけるロボット r_i の位置 $p'_i = (u'_i, v'_i)$

- 1: if $ID_i \in \text{偶数}$ then $mark := 1$
 - 2: else $mark := -1$
 - 3: $factor := ID_i \% \text{列数} + 1$
 - 4: if 列数 = 奇数 & $ID_i \in \text{奇数行}$ then
 - 5: if $ID_i \in \text{偶数}$
 - 6: $factor := \lfloor (ID_i \% \text{列数} + 2) / 2 \rfloor$
 - 7: else
 - 8: $factor := \lfloor (ID_i \% \text{列数} + 1) / 2 \rfloor$
 - 9: else $factor := \lfloor (ID_i \% \text{列数} + 1) / 2 \rfloor$
 - 10: $u'_i := ID_i / \text{列数} \times DIST$
 - 11: $v'_i := factor \times DIST \times mark$
-

Column

図 6.12 は 8 台のロボットにおける 2 行 column の形成位置である。

column は line のアルゴリズムを利用することで何行でも形成することが可能となっている。

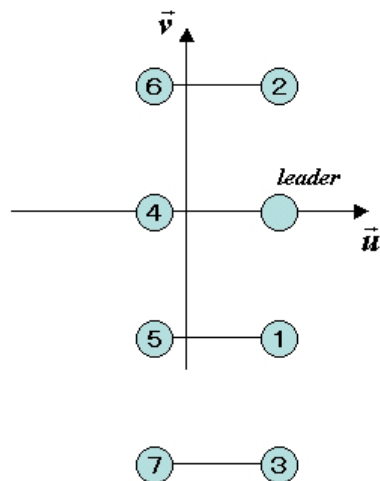


図 6.12: Two columns

Algorithm 6.11 column におけるロボット r_i の位置 $p_i' = (u_i', v_i')$

- 1: **if** $N \% \text{行数} = 0$ **then**
 - 2: 列数 $:= \lfloor N / \text{行数} \rfloor$
 - 3: **else**
 - 4: 列数 $:= \lfloor N / \text{行数} \rfloor + 1$
 - 5: **Algorithm 6.10** を実行する。
-

6.1.2 フォーメーション形成位置への移動

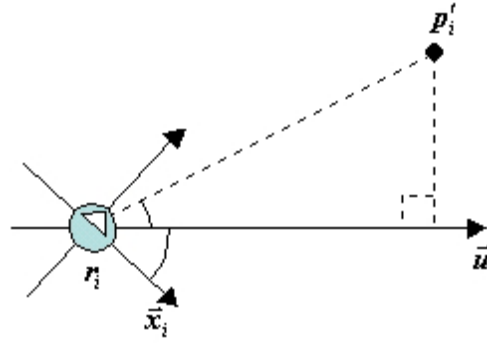


図 6.13: フォーメーション形成位置への移動方向と距離

各ロボット r_i は共通座標におけるそれぞれのフォーメーションパターン形成位置 p'_i の位置座標 $W_i[p'_i]$ を 6.1.1 節から得ることができる。 $W_i[p'_i]$ と r_i の共通座標における現在位置 $W_i[p_i]$ を用いて、 p_i から p'_i までの移動距離を式 (3.3) から、移動方向を式 (3.1) を利用して得ることができる。これによって得られた $angle(\vec{u}, p'_i)$ に $angle(\vec{x}_i, \vec{u})$ を加えた角度だけ方向転換し、 $dist(p_i, p'_i)$ だけ前進することで各ロボット r_i はフォーメーション形成位置へと移動することができる。

また、移動時は式 (3.5) と式 (3.6) によって origin と leader の位置座標を正しい値に維持することが可能である。

6.1.3 シミュレーション

図 6.14 に示す 12 台のロボットがランダムに配置された初期状態から各フォーメーションパターンを形成するシミュレーションを行った。シミュレーションの結果を表 6.1 に示す。表 6.1 におけるフォーメーションパターンの並びの順番は、ロボットがフォーメーション形成時に移動した距離の短い順である。フォーメーションパターンの形成過程は図 6.15 から図 6.25 までの通りである。各パラメータは表 5.1 に従っている。

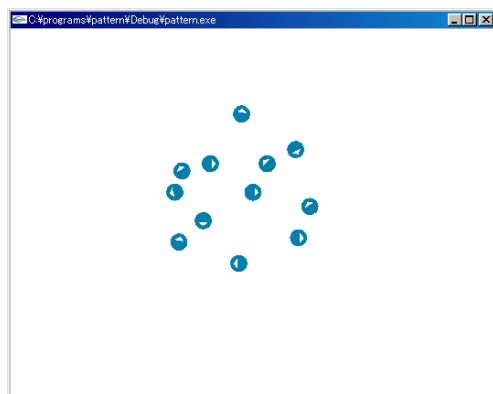


図 6.14: 初期状態

N=12	総移動距離	平均移動距離	時間 (s)	衝突回避機会
pentagon	22.426	1.869	3.070	2
hexagon	25.689	2.141	2.541	0
triangle	26.421	2.202	3.705	2
circle	26.814	2.235	2.389	0
line(3)	31.772	2.648	4.306	3
diamond	40.759	3.397	6.017	4
column(3)	42.179	3.515	4.347	2
line(2)	43.102	3.592	4.181	0
cross	43.572	3.631	5.333	3
arrow	51.844	4.320	5.431	0
arc	76.870	6.406	7.902	1
wedge	78.065	6.505	9.045	1

表 6.1: シミュレーション結果

Wedge

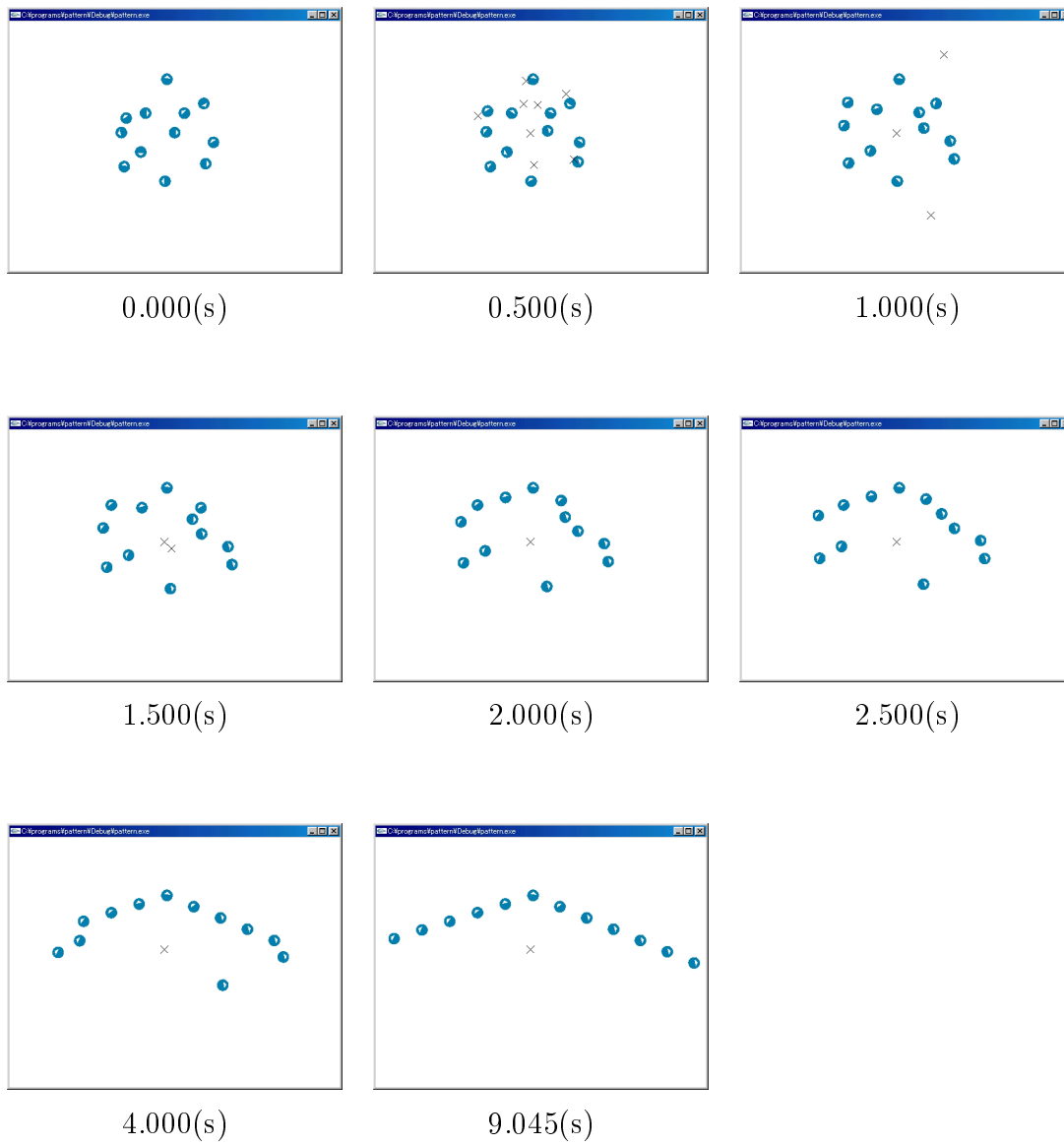


図 6.15: Wedge の形成過程

Circle

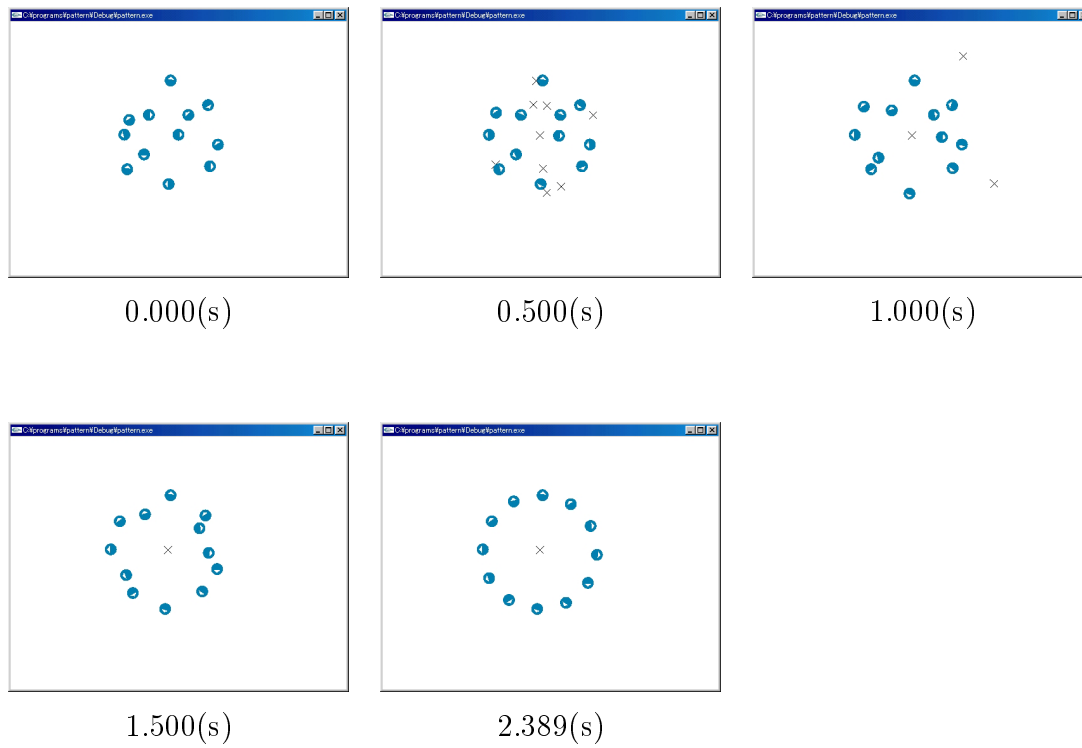


図 6.16: Circle の形成過程

Arc

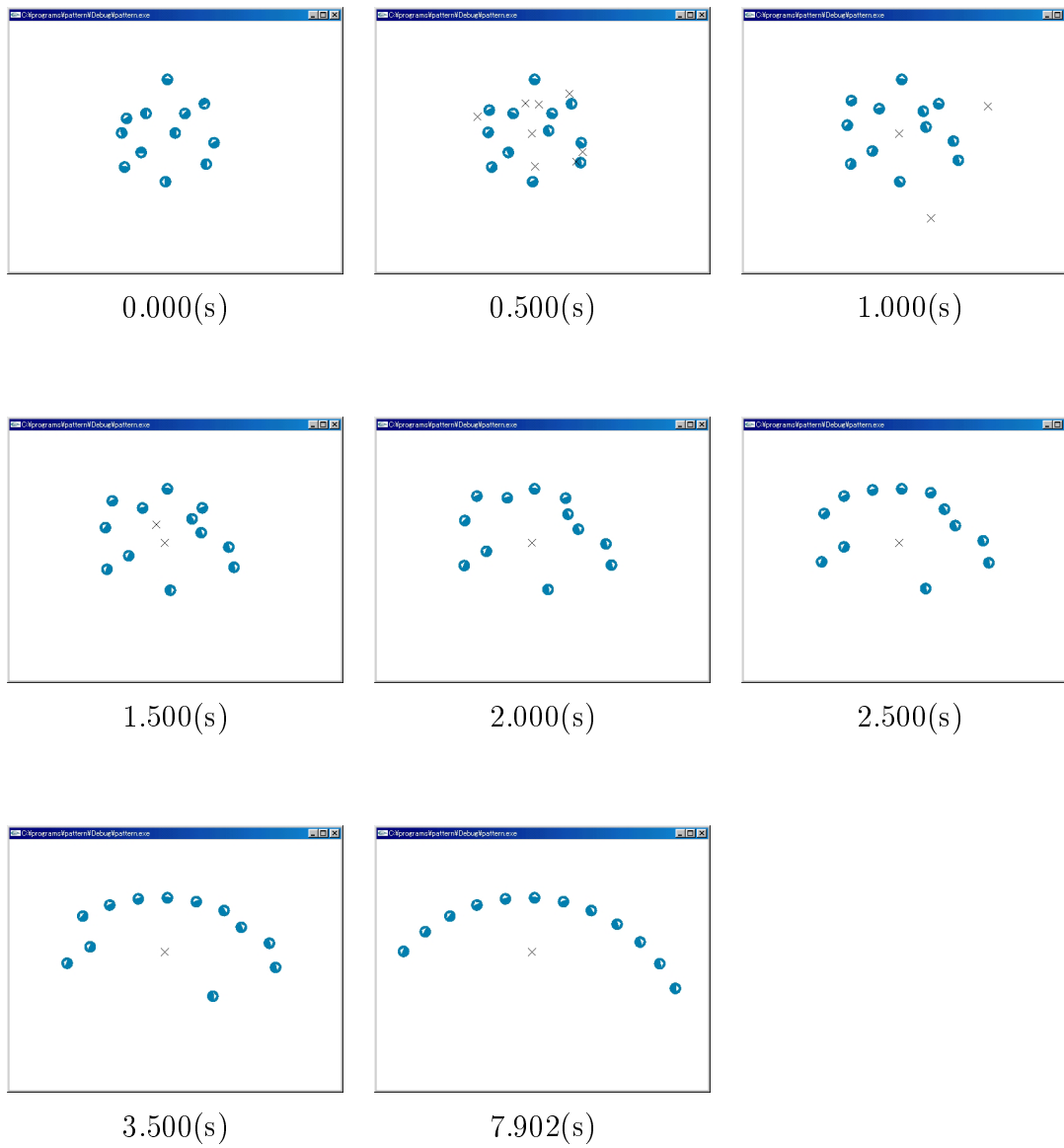


図 6.17: Arc の形成過程

Triangle

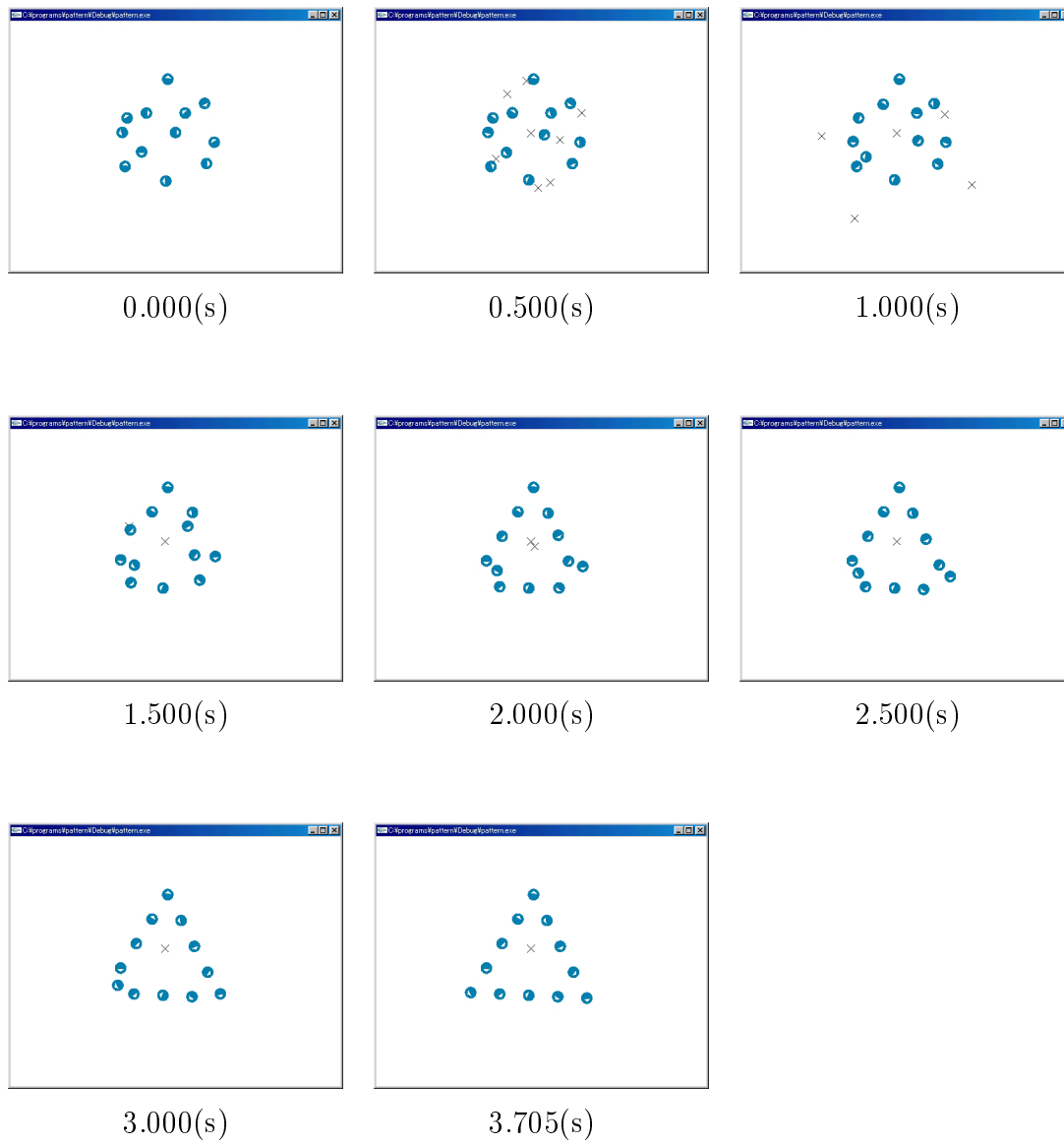


図 6.18: Triangle の形成過程

Diamond

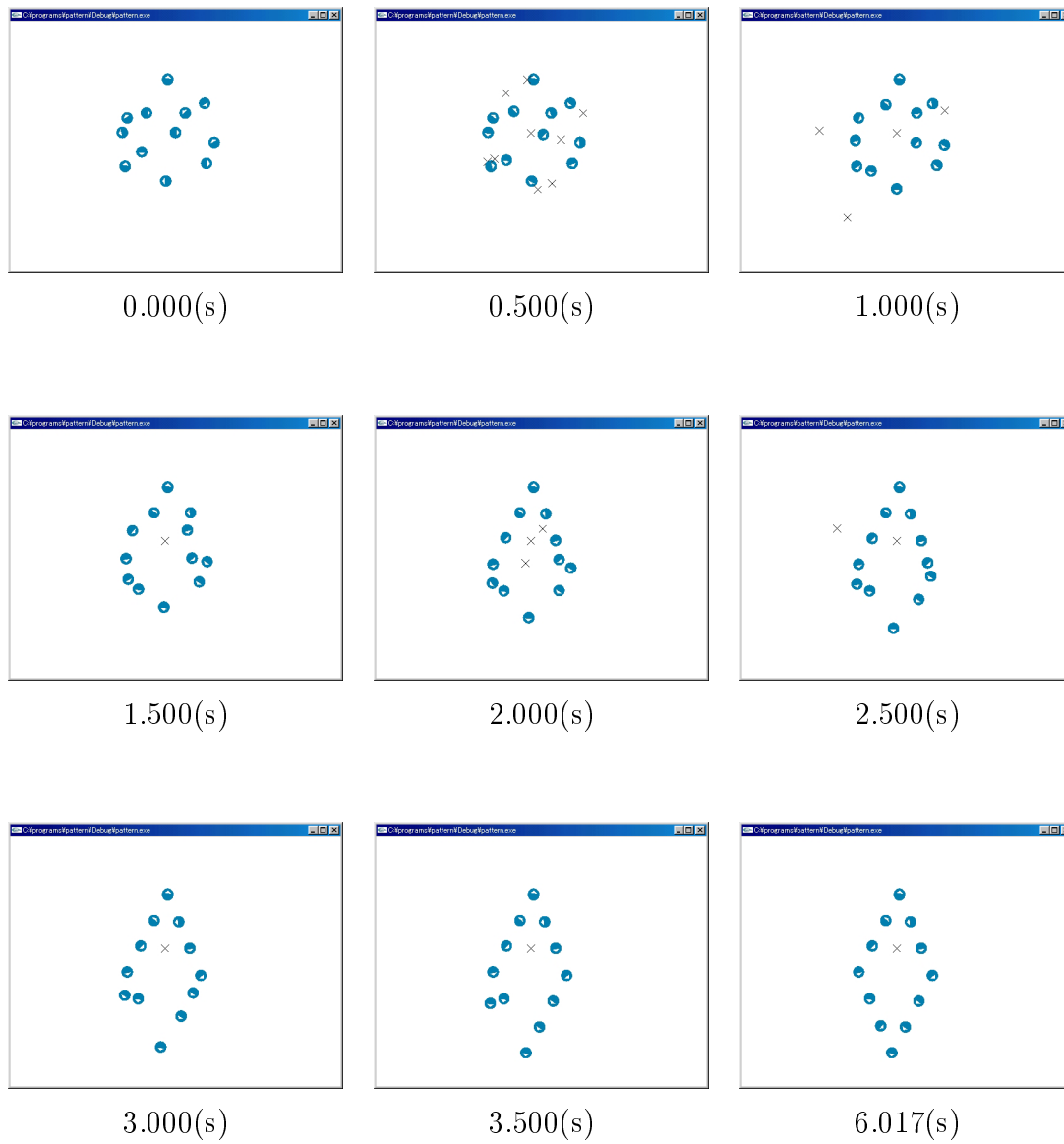


図 6.19: Diamond の形成過程

Pentagon

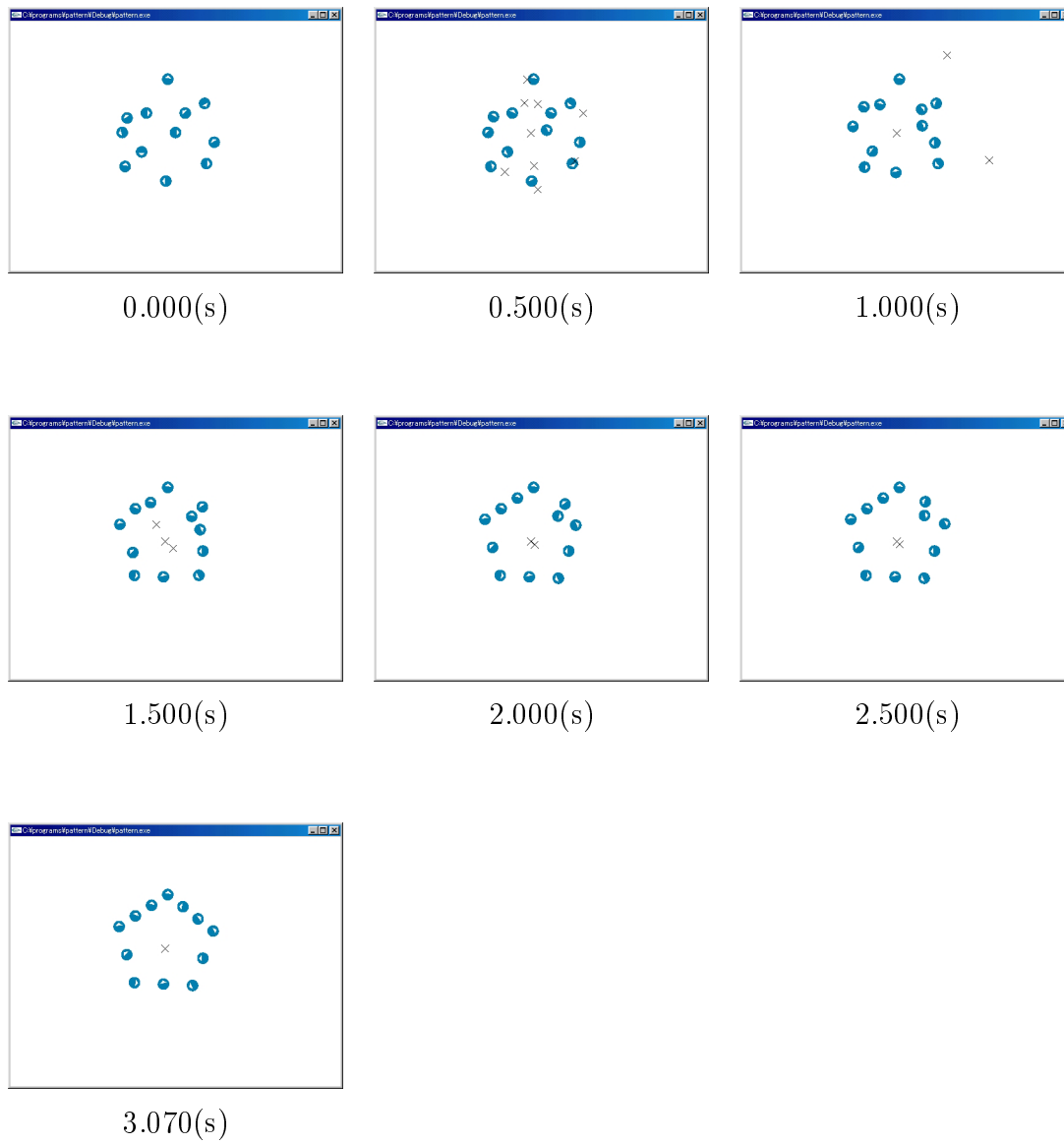


図 6.20: Pentagon の形成過程

Hexagon

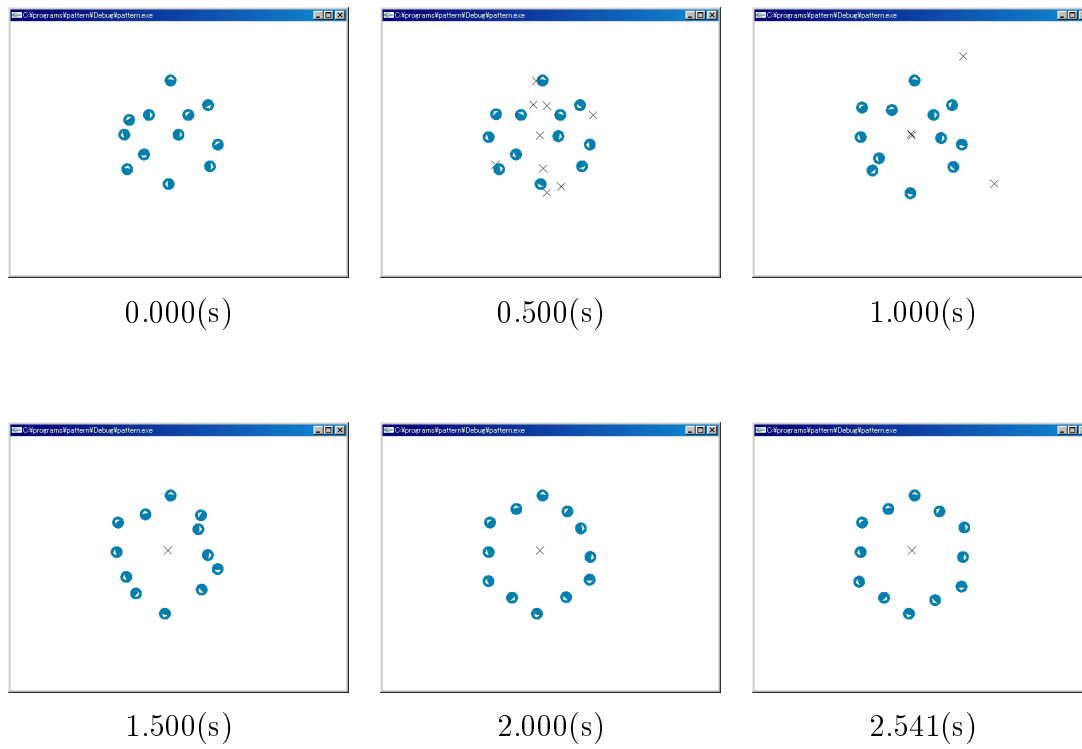


図 6.21: Hexagon の形成過程

Arrow

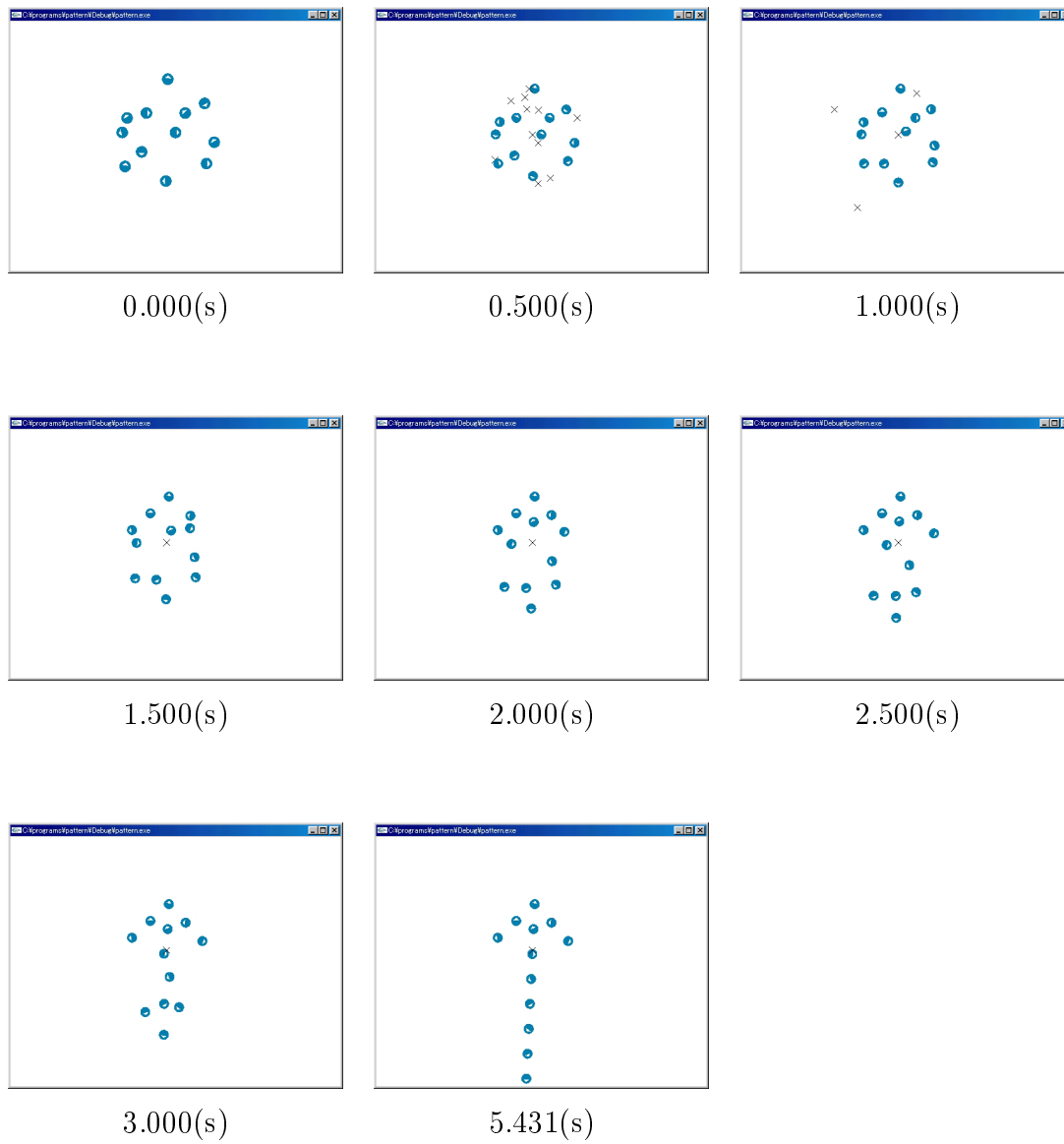


図 6.22: Arrow の形成過程

Two lines

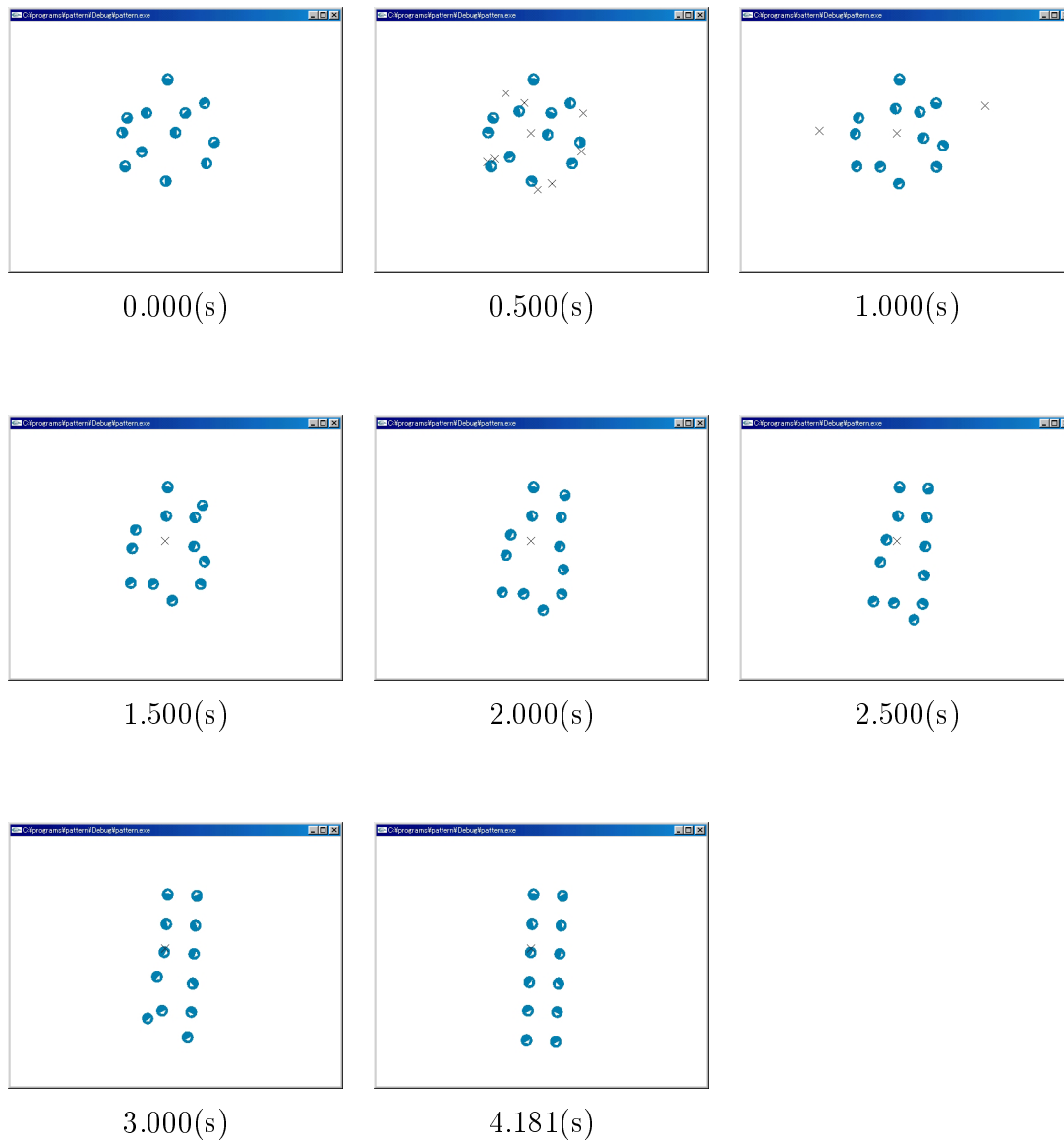


図 6.23: Two lines の形成過程

Three lines

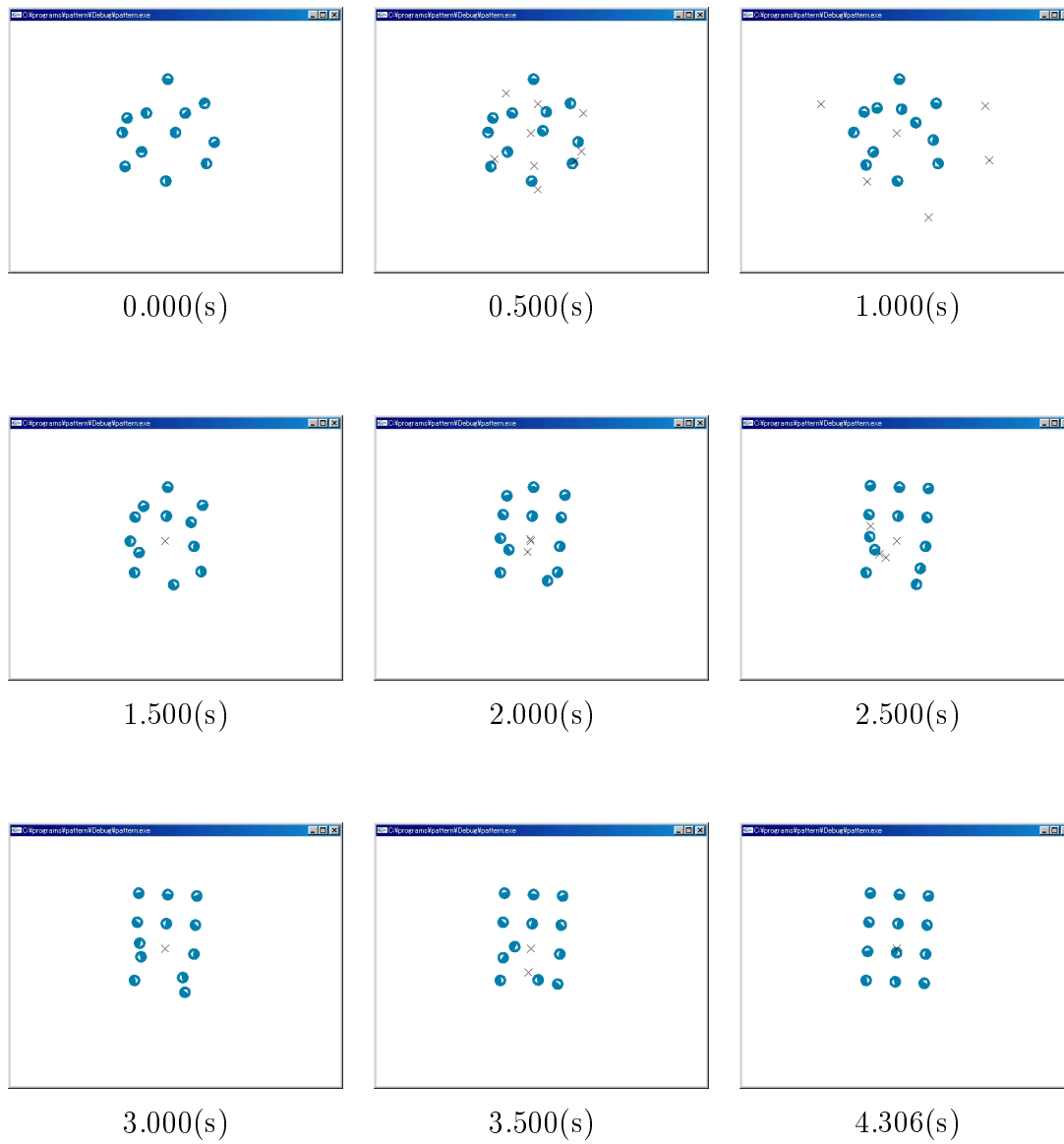


図 6.24: Three lines の形成過程

Three columns

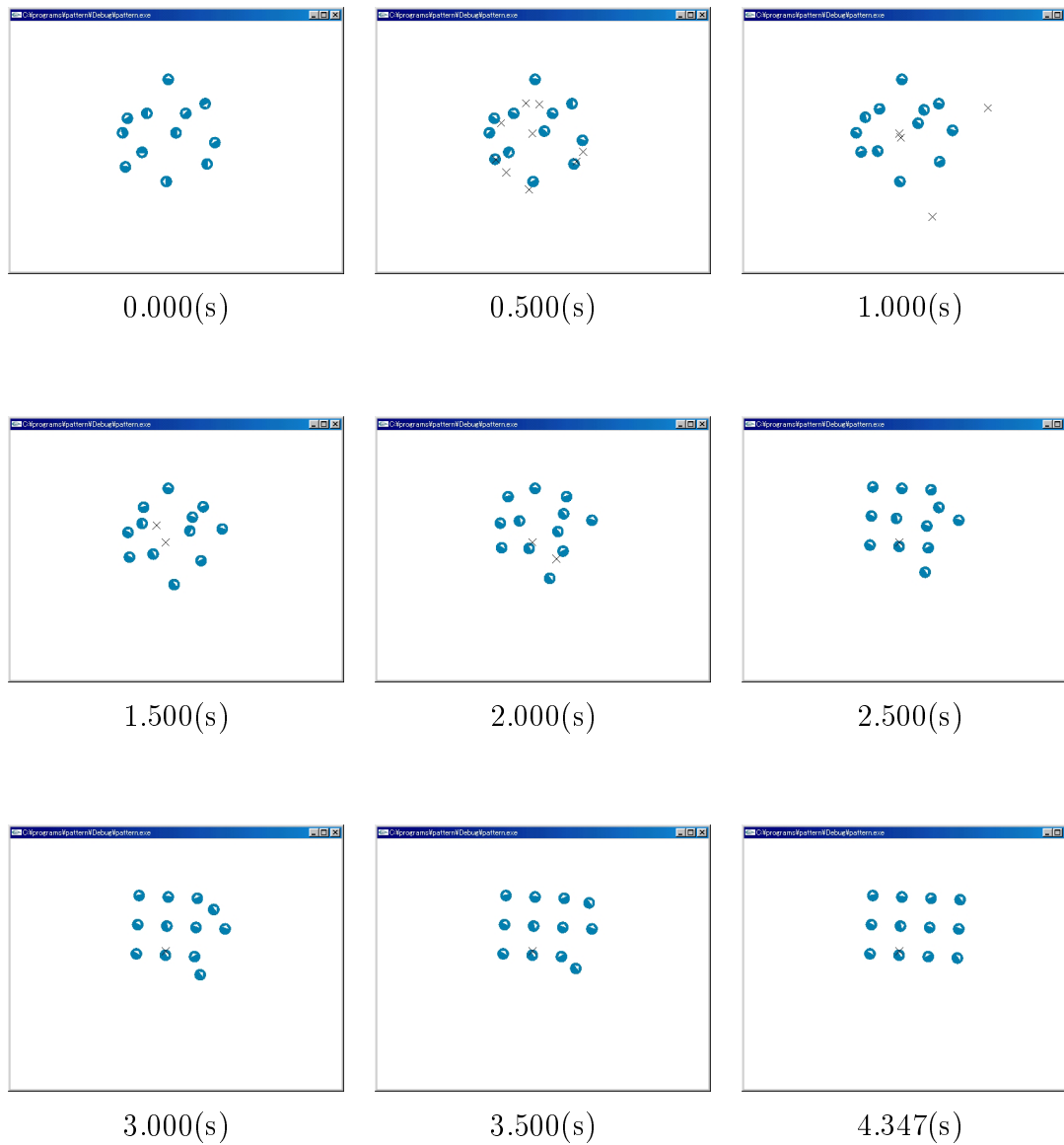


図 6.25: Three columns の形成過程

6.2 フォーメーションの再編成

ロボット群はフォーメーションの再編成を行う能力を持つことで、群れの移動に伴う環境の変化に応じてフォーメーションパターンを変形させることが可能となる。

フォーメーションの再編成を行う際は leader と ID の決定を再び行う必要はない。これらの各ロボットが持つ群れの役割を決定するための知識は群れの移動によって変化することはない。再編成をするために新たに獲得しなければならない知識は共通座標である。これは共通原点である $origin$ と共通方向である $\vec{u}$ が群れの移動に伴うフォーメーションのズレによってロボット間で共有できなくなるからである。

以上から、ロボット群は再編成をする際は次の点を行えばよい。

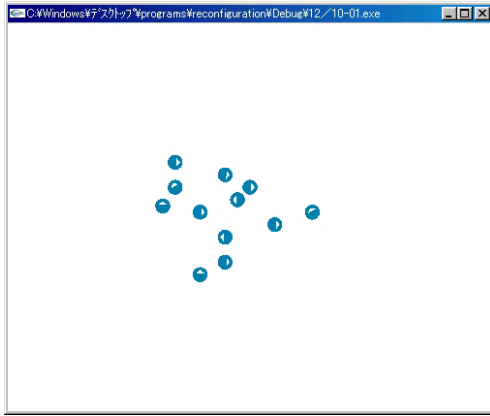
1. leader 位置の確認
2. $origin$ の決定
3. $\vec{u}$ の決定
4. 再編成するフォーメーションパターンの形成位置の獲得
5. フォーメーション形成位置への移動

シミュレーション

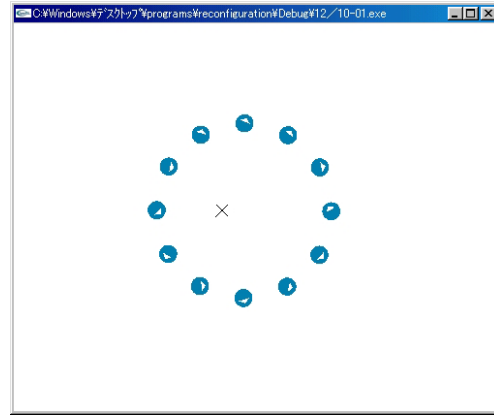
図 6.26(a) に示すランダムな配置にある 12 台のロボットが circle を形成した状態 (図 6.26(b)) から他の各フォーメーションパターンへと再編成するシミュレーションを行った。表 6.2 にその結果を示す。表 6.2 におけるフォーメーションパターンの並びの順番は、再編成においてロボットの移動距離が短かった順である。図 6.26 には circle から two lines に再編成する過程を示す。

表 6.2 から、circle から各フォーメーションパターンへの再編成においては hexagon、triangle などの circle と同様に閉じた形態への再編成は容易であることがわかる。また、wedge や arc などの開いた形態への再編成の際も、少ない衝突機会で行うことが可能であることがわかる。

このように全フォーメーションパターンにおいて ID 配置が類似しているため、再編成におけるロボットの移動距離や衝突機会、必要時間を減少させることができ、再編成を容易に行うことが可能となっている。



(a) 初期状態



(b) circle の形成後

図 6.26: circle の形成

N=12	総移動距離	平均移動距離	時間 (s)	衝突回避機会
hexagon	5.228	0.436	1.383	0
triangle	21.466	1.789	2.694	0
pentagon	21.968	1.831	1.944	0
diamond	36.622	3.052	3.181	0
cross	39.794	3.316	4.434	1
line(2)	40.657	3.388	4.306	0
arrow	53.988	4.499	5.306	2
column(2)	62.182	5.182	6.310	2
arc	67.174	5.598	7.041	0
wedge	68.628	5.719	7.332	0

表 6.2: circle からの再編成結果

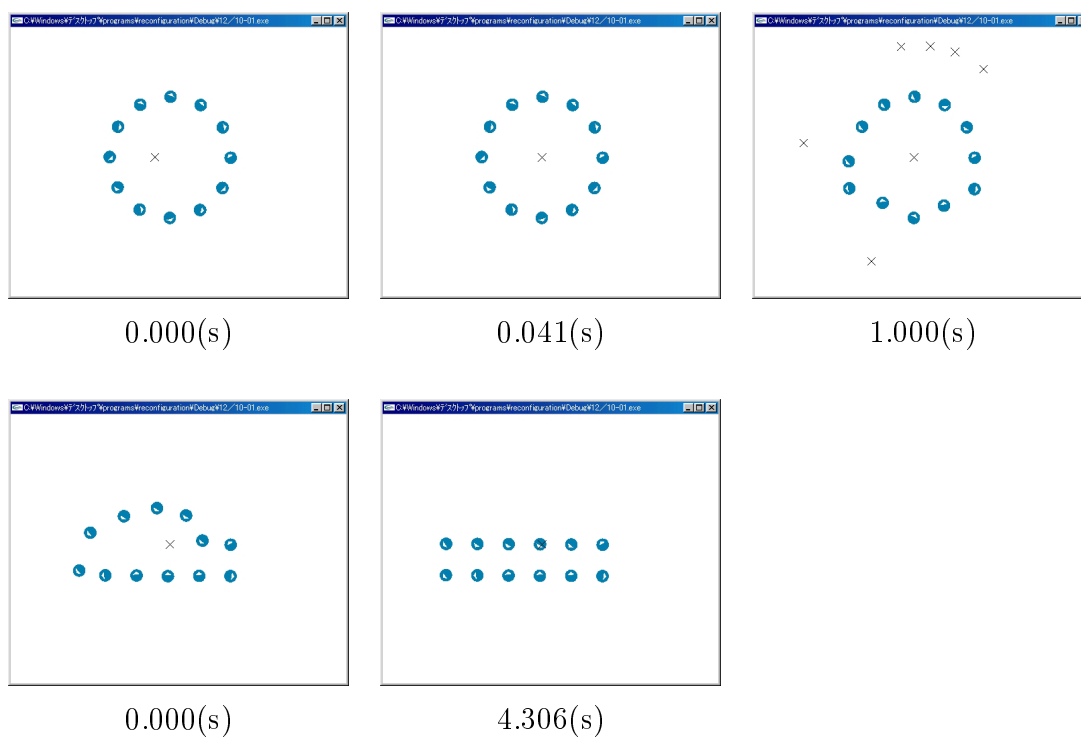


図 6.27: circle から two lines への再編成過程

6.3 フォーメーションを維持した移動

群れで行動するロボット群はフォーメーションを維持した移動を行うことで地雷探査、斥候、警備などの各種アプリケーションにおける協調した作業を行うことができる。

フォーメーションを維持した移動に関する研究は多くあるが、最も一般的なものは leader-followers 方式による移動である。leader-followers 方式では 1 台のロボットが leader となり、経路探索など群れの引率に関わる役割を担う。その他の follower は leader との距離や角度、もしくは隣接するロボットとの距離や角度の維持を行う。これにより、群れ全体の統率のとれた移動を行うことが可能となる。

本研究で提案したフォーメーションの形成では全てのフォーメーションパターンにおいてその先頭に leader がいるため、これによって leader-followers 方式によるフォーメーションを維持した移動が可能である。フォーメーションを形成した全ロボットは leader の位置と origin の位置を共通して理解しているため、共通方向に方向転換し、leader との距離、角度を維持することで leader の動きに合わせたフォーメーションの維持を行うことが可能となる (図 6.28)。

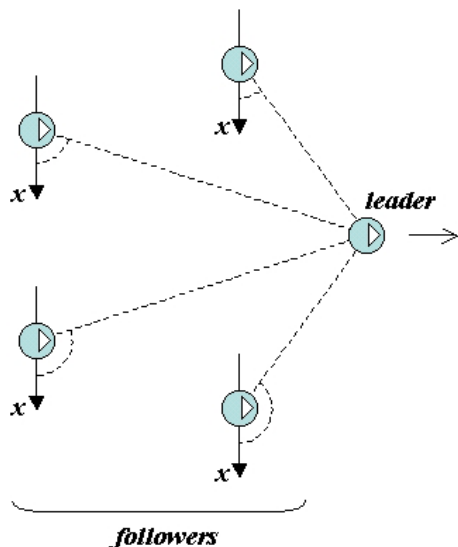


図 6.28: leader-followers 方式

subsubsection シミュレーション

triangle を形成したロボット群において、leader-followers 方式を利用した移動を行った (図 6.29)。leader は全 follower の移動の停止によってフォーメーション形成の完了を確認し、ランダムな移動を開始する。follower はフォーメーション形成後に leader 位置 p_l の地点に存在するロボットを leader として認識し、leader との角度、距離を維持することに努める。これにより、leader-followers 方式を利用した移動が可能となっている。

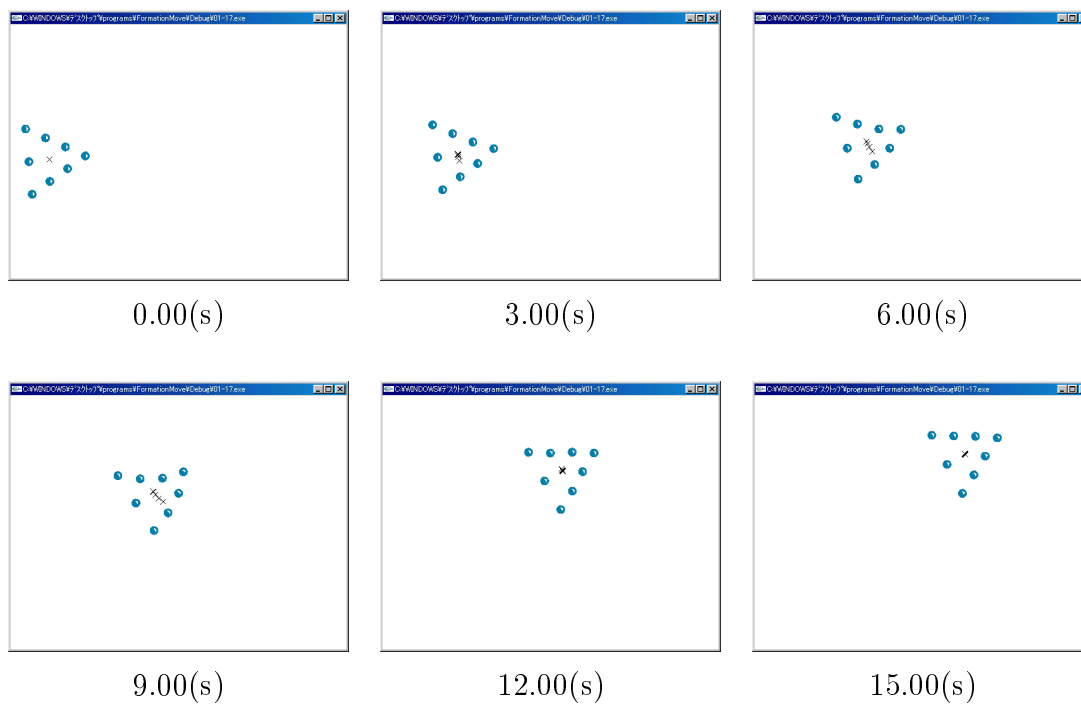


図 6.29: leader-followers 方式による移動

6.4 フォーメーションのロバスト性

本研究において定義したロボット群は初期状態において anonymous であり、共通知識の獲得を通して leader の決定や ID 番号の獲得を行う。このため、leader や群れを構成するロボットメンバに故障が発生した際においても leader や ID 番号の再決定をすることができ、これによって正常に作動するロボットにおいてフォーメーションの再形成が可能となる。このため、本研究で提案するアルゴリズムは群れを構成するロボットの故障に対してロバストである。

図 6.30 にロバスト性能を表すシミュレーション結果を示す。シミュレーションにおいて 9 台のロボットが circle を形成し、維持しながら目的地まで進む。このとき、2 台のロボットに一定距離を進行後に停止するよう予め定めた。ロボット群はこれらのロボットの停止を故障とみなして ID の再決定とフォーメーションの再編成を行い、circle の維持が可能となっている。

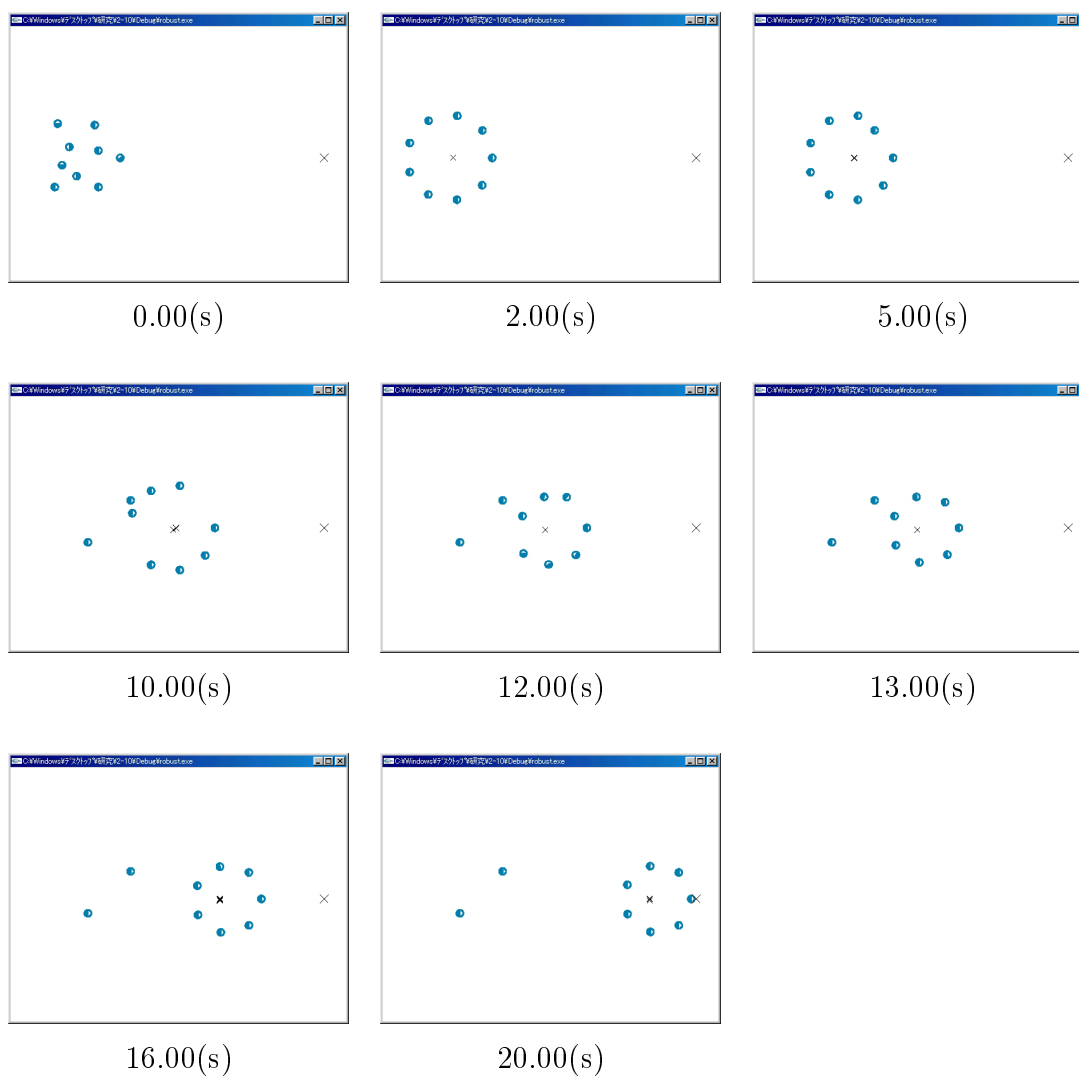


図 6.30: 群れのメンバの故障に対するロバスト性

6.5 結果

ロボット群がフォーメーションを組んだ協調作業を行うとき、以下の項目が達成可能であることが必要である。

- (a) ランダムな配置からのフォーメーション形成
- (b) フォーメーションを維持した移動
- (c) フォーメーションの再編成
- (d) 狭隘領域通過や障害物回避

本章では、ランダムな配置からのフォーメーションの形成 (a)、あるフォーメーションパターンから他のパターンへの再編成 (c)、leader-followers 方式を利用したフォーメーションを維持した移動 (b) のそれぞれを可能とし、シミュレーションを行った。これにより、フォーメーションの再編成とフォーメーションを維持した移動を状況に応じて行うことで狭隘領域の通過 (d) を可能とした。図 6.31 は狭隘領域の通過を行うシミュレーションである。leader が通過領域の幅とフォーメーションの幅を計算し、各 follower にフォーメーションの再編成を命令することで狭隘領域の通過を行った。

この結果、本章においてロボット群における協調作業に必要な (a) から (d) までの項目を達成し、本稿で提案するアルゴリズムの有効性を示した。

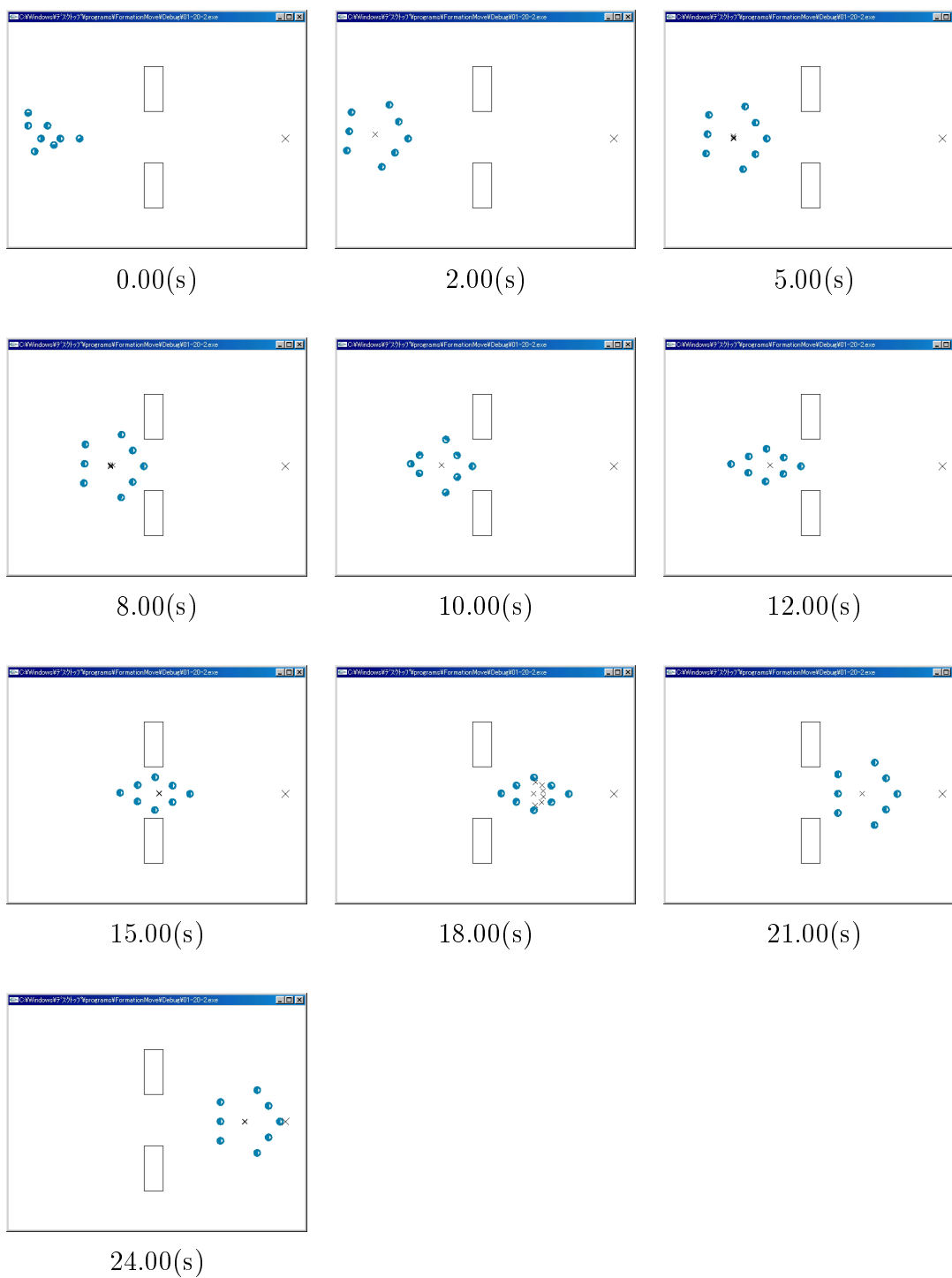


図 6.31: 環境に応じた移動

第7章 結論

7.1 本研究のまとめ

本研究では、anonymous ロボット群によるランダムな配置からのフォーメーション形成は困難であることに着目した。また、フォーメーション形成を扱った研究の多くがフォーメーションを維持した移動まで考慮していないことに着目した。これらの点から、ロボット群がどのような状態からでもフォーメーションを形成し、そのフォーメーションを維持した移動を可能とする方法の考察を行った。これを実現するため、共通知識を段階的に獲得するアルゴリズムと、その知識を利用した移動可能なフォーメーションの形成を行うアルゴリズムを提案した。この結果、anonymous ロボット群はランダムに配置された状態から leader や ID 番号、共通座標の決定を行い、これらを利用したフォーメーションの形成が可能となった。形成可能なフォーメーションの種類は多様であり、本稿において 11 種類のフォーメーションパターンの形成方法を紹介した。これらのフォーメーションパターンにおいて、あるフォーメーションパターンから他のパターンへの再編成は ID 番号の配置が類似しているため非常に容易となっている。フォーメーションを維持した移動は leader-followers 方式によって行った。ロボット群はフォーメーションを維持した移動とフォーメーションの再編成の両機能を利用することで編隊移動制御を達成し、シミュレーションでその有効性を確認した。

7.2 本研究のアドバンテージ

本研究で提案したアルゴリズムを利用することにより、ロボット群はどのようなランダムに配置された状態からでもフォーメーションを形成することが可能である。これにより、ロボットは人の手に頼ることなく自律的にフォーメーションを形成することができる。形成できるフォーメーションパターンの種類は多様であり、さまざまな状況に応じたフォーメーションの形成を行うことが可能である。さらに、フォーメーションはあるフォーメーションパターンから他のパターンへの再編成を容易に行うことができるため、ロボット群は作業目的や周囲の環境の変化に応じて迅速にフォーメーションの組み替えを行うことが可能である。また、ロボット群は leader や ID を動的に決定することが可能であり、ロボットの故障に伴う群れの構成員の欠如に対してロバストにフォーメーションを維持することが可能となっている。

7.3 今後の展望

本研究のアルゴリズムはセンサの遅延と誤差を考慮していないため、今後はこれらを考慮した共通知識の獲得とフォーメーションの形成を行う。また、周囲の環境やロボットの故障をどのようにして認識するかという問題についても考慮する必要がある。最終的にはアルゴリズムの有効性を実機を用いて示す。

謝辞

本研究を進めるにあたり、ご指導賜りました丁洛榮助教授に深く感謝致します。また、細部にわたり熱心に助言を下さった李根浩さんをはじめとするモバイルロボットチームのメンバー、そしてロボティクス講座の皆様に厚くお礼を申し上げます。最後に、私のよき理解者である両親と大学院での生活を支えてくれた妻、気晴らしの相手をしてくれた娘に心から感謝致します。

参考文献

- [1] B. R. Donald, Information invariants in robotics: Part I-State, Communication, and Side-Effects, ICRA(3), pp.276-283, 1993.
- [2] B. R. Donald, Information Invariants in Robotics: Part II - Sensors and Computation, ICRA(3), pp284-290, 1993.
- [3] I. Suzuki and M. Yamashita, Distributed anonymous mobile robots: Formation of geometric Patterns, SIAM Journal of Computing, 28(4), pp1347-1363, 1999.
- [4] I. Suzuki and M. Yamashita, Agreement on a common x-y coordinate system by a group of mobile robots, In Proc. Dagstuhl Seminar on Modeling and Planning for Sensor-Based Intelligent Robots, 1996.
- [5] K. Sugihara and I. Suzuki, Distributed Algorithms for Formation of Geometric Patterns with Many Mobile Robots, Journal of Robotics Systems, 13, pp127-139, 1996.
- [6] K.Sugihara and I.Suzuki, Distrivuted Motion Coordination of Multiple Mobile Robots, In Proc. IEEE Int. Symp. on Intelligent Control, pp138-143, 1990.
- [7] M.Cieliebak and G.Prencipe, Gathering autonomous mobile robots, In Proc. 9th Colloquium on Structural Information and Communication Complexity(SIROCCO'02),2002.
- [8] X. Defago and A. Konagaya, Circle formation for oblivious anonymous mobile robots with no common sense of orientation, In Proc. 2nd ACM Intl. Workshop on Principles of Mobile Computing, pp.97-104, 2002.
- [9] O.Tanaka, Forming a Circle by Distributed Anonymous Mobile Robots, Bachelor Thesis, Department of Electrical Enginnering, Hiroshima University, 1992.
- [10] 池本有助, 長谷川泰久, 福田敏男, 松田一彦, ロボット群による段階的空間パターン形成に関する研究, JRSJ, Vol.22 No.07, pp.97-105, 2004.

- [11] J. Fredslund and M. J. Mataric, A general algorithm for robot formations using local sensing and minimal communication, *IEEE Transactions on Robotics and Automation*, 18(5), pp837-846, 2002.
- [12] 中川研太, 複数台移動ロボットの編隊移動制御、東京工業大学修士論文, 2002
- [13] H.Ando, Y.Oasa, I.Suzuki, and M. Yamashita, Distributed memoryless point convergence algorithm for mobile robots with limited visibility, *IEEE Trans.on Robotics and Automation*, 15(5), pp818-828, 1999.
- [14] P.Flocchini, G.Prencipe, N.Santoro, and P. Widmayer, Pattern formation by autonomous robots without chirality, In *Proc. VIII Intl. Colloquium on Structural Information and Communication Complexity(SIROCCO 2001)*, pp147-162, 2001.
- [15] P. Flocchini, G.Prencipe, N. Santoro and P. Widmayer, Hard Tasks for Weak Robots: The Role of Common Knowledge in Pattern Formation by Autonomous Mobile Robots, In *Proc. 10th Annual International Symposium on Algorithms and Computation (ISAAC 99)*, LNCS 1741, pp93-102, 1999.
- [16] M. Cielibak, P. Flocchini, G. Prencipe, N. Santoro, Solving the robots gathering problem, In *Proc. of 30th International Colloquium on Automata, Languages and Programming (ICALP 2003)*, LNCS 2719, pp1181-1196, 2003.
- [17] P.Flocchini, G.Prencipe N.Santoro, P.Widmayer, Distributed Coordination of a Set of Autonomous Mobile Robots, In *Proc. IEEE Intelligent Vehicle Symposium (IV 2000)*, pp480-485, 2000.
- [18] N.Malpani, J.Welch, and N.Vaidya, Leader election algorithms for mobile ad hoc networks, *Proceeding of the 4 'th International Workshop on Discrete Algorithms and Methods for Mobile Computing & Communication*, 2000
- [19] Cristian tapus and Lars Cremean, Real-Time Leader Election for Cooperative Control of Dynamic Vehicle Formations, *10th IEEE Real-Time and Embedded Technology and Applications Symposium(RTAS)*, 2004.
- [20] L. Barriere, P. Flocchini, P. Fraigniaud and N. Santoro, Electing a leader among anonymous mobile agents in anonymous networks with sense-of-direction, *Technical Report LRI-1310*, 2002.
- [21] T. Balch and R. Arkin, Behavior-based formation control for multi-robot teams, *IEEE Transactions on Robotics and Automation*, 1998.

- [22] 下村芳樹, 中野康夫, 大築康生, マイクロマシーン群のための作業パターン形成技術, 電機学会誌 E, vol.122-E, no.5, pp267-273, 2002.
- [23] 藤井正範, 北陸先端科学技術大学院大学修士論文, 2003.
- [24] P. K. C. Wang, Navigation Strategies for Multiple Autonomous Mobile Robots Moving in Formation, Journal of Robotic Systems, 8(2), pp177-195, 1991.
- [25] G.Prencipe, CODA: Distributed coordination of a set of autonomous mobile robots, In Proc.4th European Research Seminar on Advances in Distributed Systems(ERSADS'01), pp185-190, 2001.
- [26] Lemay, M., Michaud, F., Letourneau, D., Valin, J.-M., Autonomous initialization of robot formations, IEEE International Conference on Robotics and Automation, 2004.
- [27] A. K. Das, R. Fierro, V. Kumar, J. P. Ostrowski, J. Spletzer, and C. J. Taylor, A vision-based formation control framework, IEEE Trans. Robotics and Automation, vol.18, no.5, pp813-825, 2002.