

Title	局所的相互作用に基づいた群ロボットの未知作業環境への適応的移動
Author(s)	花田, 洋輔
Citation	
Issue Date	2007-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/3621
Rights	
Description	Supervisor: 丁 洛榮, 情報科学研究科, 修士

修 士 論 文

局所的相互作用に基づいた
群ロボットの未知作業環境への適応的移動

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

花田 洋輔

2007 年 3 月

修 士 論 文

局所的相互作用に基づいた 群ロボットの未知作業環境への適応的移動

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 丹康雄 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

510081 花田 洋輔

提出年月: 2007 年 2 月

概 要

本稿では，自律移動ロボット群が障害物の存在している環境内を適応的に移動をすることが可能な Adaptive Flocking アルゴリズムを提案する．提案手法はロボット間の局所的な相互作用を基に Team Maintenance，Team Partition，Team Unification の 3 つのアルゴリズムで構成される．提案手法のシミュレーションを行った結果，分散制御，自己組織化，自己安定化，決定論的手法の特徴が明らかになった．分析結果は提案手法が環境変化に対しても適応的に群移動が可能で，シンプルかつ効果的なアプローチであることを示した．

目次

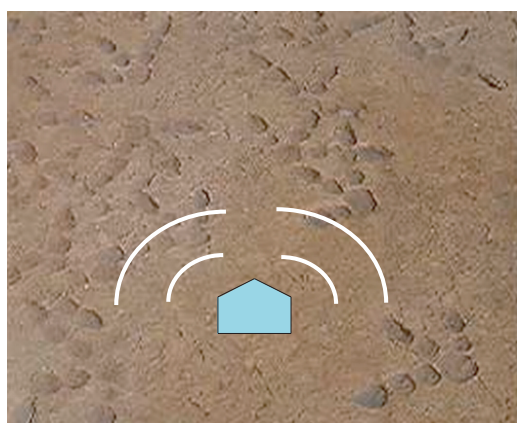
第1章	はじめに	1
1.1	研究の背景	1
1.2	関連研究	4
1.3	研究の目的	5
第2章	問題の定式化	7
2.1	魚の群泳	7
2.2	問題定義	9
2.3	ロボットモデル	10
2.4	まとめ	11
第3章	Flocking Framework	12
3.1	アルゴリズム	12
3.2	Local Interaction の収束性	14
第4章	Decomposed problems	17
4.1	Team Maintenance	17
4.1.1	アルゴリズム	17
4.1.2	シミュレーション	19
4.2	Team Partition	22
4.2.1	<i>favorite vector</i>	22
4.2.2	アルゴリズム	22
4.2.3	シミュレーション	24
4.3	Team Unification	26
4.3.1	アルゴリズム	26
4.3.2	シミュレーション	28
第5章	Adaptive Flocking	29
5.1	アルゴリズム	29
5.2	シミュレーション	30
5.3	提案手法の検証	36
5.3.1	アルゴリズムの拡張性	36

5.3.2	各アルゴリズムの効果	38
第 6 章	Tracking Multiple Moving Goals	41
6.1	アルゴリズム	41
6.1.1	複数のゴールに対する Team Partition	41
6.1.2	${}^g\vec{f}_{\max}$ と ${}^s\vec{f}_{\max}$ の両立	42
6.2	シミュレーション	43
第 7 章	まとめ	47
7.1	研究のまとめ	47
7.2	研究の利点	47
7.3	今後の展望	48

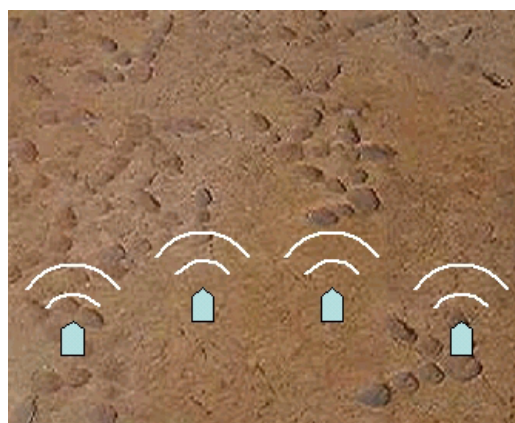
第1章 はじめに

1.1 研究の背景

従来，作業ロボットは所定の位置において予め定められた動作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ロボットが多く開発されている．このような中で，一つのタスクを一台の高性能ロボットで行うのではなく，低機能な複数台の自律移動ロボットの協調動作によって達成するマルチロボットシステムに関する研究が注目されている（図．1.1，1.2）自律移動ロボットを複数台用いタスクを達成することで，効率性や耐故障性，一台の高性能ロボットからの脱却によるコスト減や汎用性などのアドバンテージを得ることができる．



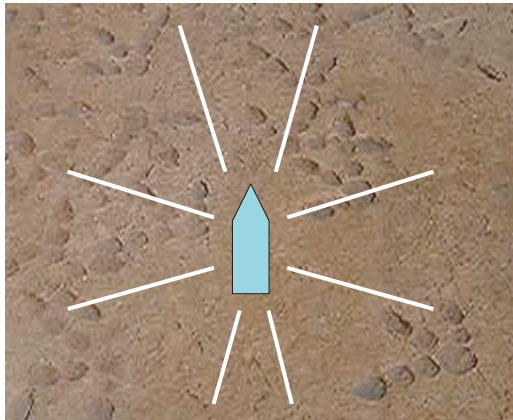
(a) 1 台での探査



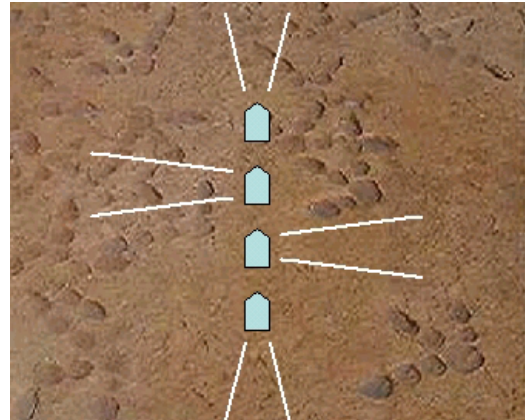
(b) 複数台での探査

図 1.1: 地雷探査

マルチロボットシステムの研究は少数台を扱った研究と多数台を扱った研究に大きく大別できる．前者の少数台の場合，互いに全ロボットを観測することが容易なため，チームを統括するリーダーロボットの事前設定などによる制御手法を用いる傾向が強い．一方で，後者の多数台を用いる場合，一台が他の全ロボットを観測できることを必ずしも期待できない．そのため，リーダーロボットの設定による制御手法の適用を試みたとき，大域的な観測者や大域座標による全ロボットの位置情報などが必要になる．しかし，実環境ではそれらの存在は必ずしも期待できない．



(a) 1 台での斥候



(b) 複数台での斥候

図 1.2: 斥候

これを解決するために自律ロボットを完全に分散的に制御し、近傍のロボットとの局所的な相互作用によってシステム全体に目的に応じた挙動を発現させる手法が注目されている．これは群ロボット工学 (swarm robotics) [1] という分野に位置づけられ、シンプルな均質個体を多数用いることで、拡張性、柔軟性、耐故障性などの利点を得ることができ、近年多大な期待を寄せられている．しかし、このようなアプローチはロボットが置かれるであろう状況の予測とそれに対する適切な行動をあらかじめ規定することや、それによる各ロボットの適切な入出力関係を事前に規定することの困難性が存在している．それらの問題を解決するために、個体間での相互作用や、環境との相互作用を通じて自己組織化を行うことができる生物学・物理学・化学等の創発システムを模倣対象とし、制御則を設計することが必要となってくる．例えば、葉を紡いで営巣するツムギアリは葉を引っ張るために協力し合う (図 1.3) [2]．このように、一個体では効率性を低下させてしまうタスクの効率化、あるいは一個体では実行不可能なタスクさえも可能になる．この応用として、運搬 [3] や探査、あるいは化学物質の漏洩源特定 [16]、マイクロ・ナノロボット群を利用した体内診療活動 [4]、モバイルセンサネットワーク [5]、監視・警備活動、搜索・救助活動 [6] などの多岐にわたる分野における群ロボットの協調作業に応用することが期待できる．それらのタスクを実行するために、基本的な機能として群移動 (flocking) は重要であり、本研究ではこの問題について取り組んだ．

群移動はこれまでで多くの研究者が取り組んできた問題である．この問題を最初に取り扱った Craig W. Reynolds はコンピュータグラフィックや人工知能の分野に貢献した人物である．参考文献 [7] では一見複雑に見える鳥や魚、あるいは地上に生息する野生動物の群移動は各エージェントに collision avoidance, velocity matching, flock centering の三つの簡単なルールを与えることでコンピュータ上に再現できることを示した．

群移動は群ロボット工学のみならず、様々な分野で研究されてきた問題である．しかし、群ロボットが基本的な機能として備えておくべき群移動には環境への適応能力が必要不可



図 1.3: 協力して葉を引っ張るツムギアリ

欠であると考えられる．例えば，惑星の探査や斥候などの未知領域での活動，体内診療活動のような狭隘領域，被災地での搜索・救助の危険領域での活動など，ロボットにマップ情報を与えることが困難な上に，環境に事前にランドマークを立てることも不可能な状況が多く考えられる．そのようなとき，ロボットはその場で得た情報を基に自律的かつ協調的に未知の環境に適応的に群移動をする必要がある．しかしながら，これまでに環境の幾何条件に対する適応を目的として行われた研究は少ない．群ロボットは様々な応用例において環境条件の変化に適応しながら目標地点へ向かうことを必要とされる．

1.2 関連研究

環境に適応した群移動の研究は，リーダー・フォロワ手法 [8, 9, 10, 11, 12, 13] とリーダーレス手法 [14, 15, 16, 17, 18, 19, 20, 21] に大別される．前者の例として，倉林ら [8] はドローン線図により隣接ロボットの関係を定義することで群れを維持した群移動を行った．この論文によるとゴールターゲットまでの経路計画を与えられたリーダーを見失わないようにフォロワが追従することで群れ全体が目標地まで到達することができる．Parker ら [9] は多様なセンサを搭載したリーダーロボットの主導による群移動手法を提案した．Lee ら [10, 11] は分散的な方法で anonymous ロボット群がリーダー選抜や ID の取得を経て自己組織化を行うフォーメーション制御を提案した．また，これを基にリーダー参照手法とネイバー参照手法を比較，分析した．これらの群移動手法ではリーダーの行動や判断に強く依存する．また，リーダーロボットの故障に対する耐故障性の欠如や，拡張性の欠如などの問題がある．

前者に対して後者のリーダーレス手法は特別な役割を持たないロボット群を用いることを意味している．この手法は現在多くの研究が報告されており，特徴的なアプローチとしていくつかに分類することができる．

第一は Behavior-based アプローチであり，Balch らが文献 [14] で報告している．各ロボットは事前に決まった "attachment site" と呼ばれるフォーメーション形成の目標位置を設定され結晶生成を模倣した物理的な群移動手法である．群れは移動中に障害物に遭遇するとそれを回避するように分裂，その後統合を行った．この論文によると behavior-based によるロボットの動作は引力・反発力を組み合わせることで決定される．そのため，環境やタスクによって各力のパラメータとゲインの事前調整という問題が必ず存在する．

第二に，Physical-based アプローチは物理法則を群れに適用することでフォーメーション形成や移動を達成させる．Zarzhitsky らは重力に似た人工物理 (artificial physics) [15] を基にして，化学物質の流れの追従手法 [16] を提案した．この方法は 6 台の近接ロボットと流体力学の流れ場のパラメータを共有することで移動目標位置を算出するために，六角形状にロボットを配列した．Ye ら [17] はポテンシャル流れから導出した経路計画で障害物を回避しながら目的地まで達する障害物を回避する適応的な移動手法を報告している．この論文ではさらに dynamic association と singular association と呼ばれるロボット間の隣接関係の構築法により環境への適応性の向上を示した．

第三は群知能 (Swarm Intelligence) に準じた創発システムのアプローチである。これの代表的な研究例として清水ら [18] はロボット同士が流体力学的に相互作用することで水の流れに乗るような環境に適応できる形態制御手法を提案した。この論文によるとモジュラーロボット同士の力学的特性を積極活用し、振動子間の位相勾配の発生により粘菌などの生物にみられる原形質流動の創発を実現した。Folino ら [19] が提案した SPARROW と呼ばれる空間的な群制御アルゴリズムは人工生命分野に属す。このアルゴリズムは空間データにおいて形状と規模がバラバラの群れを発見するために鳥の群れに基づいた探査方法と密度情報に基づいた群制御アルゴリズムを結合した。このアルゴリズムでは、ロボットは他の各ロボットに群れを捜しながら空間データを探らせる採餌行動を行うハンターに変わる。

さらに、Communication-Based 手法では次のような報告がある。Nembrini ら [20] は局所的な情報伝達を用いたが、互いの位置情報を必要としない適応的な群移動を提案した。この手法では環境に適応した群れ形状に変形したときであっても大域的な通信接続が維持できることに主眼を置いた。Esposito ら [21] は各ロボットに目的地を与えられた群れが障害物環境下でワイヤレス通信距離と視野内に他ロボットを維持する群移動制御法を報告した。この論文では進行可能な方向集合を表す複数のポテンシャル関数からロボットの状態ベクトルのポテンシャル関数が最小となるような条件を満たす一つの適した進行方向を構成する方法を提案した。この方法は局所解の存在のために、群れの確かな接続維持が保障されない。

1.3 研究の目的

本稿は多数台の群ロボットが障害物の存在する環境内を自律的かつ協調的に適応した移動ができる完全分散型の群移動制御手法を提案する。複数台のロボットが群移動を行う際には未知環境にどのような制約が存在していても互いの近傍のロボットと局所的に相互作用し目的地まで走破する必要があるからである。したがって、本稿は「与えられた任務を達成するために群ロボットはいかにして局所的な相互作用によって未知環境に適応した群移動を行うことができるか」を問題定義し、この問題に取り組んだ。この問題解決のために、本稿はマグロの群泳行動で観察される環境に適応した秀逸な群泳形状を群ロボットの自律的適応方法として適用する。そのため、本アプローチではロボットと環境に関する仮定を次のように設定した。識別番号を持たない、リーダーロボットの事前選任はしない、共通の大域座標系は設定しない、過去の知覚・動作情報は記憶しない、相互情報伝達を行わない、作業環境情報は事前に与えない。これらの制限下で、初期状態で任意に配置された多数台の群ロボットが局所的に近傍のロボットと相互作用しながら目標位置に向かって移動する制御手法を考察した。マグロの群泳は相互作用を通じ局所的な幾何学形状を形成する。この局所的相互作用に基づきチームの維持・チームの分裂・チームの統合の三つの行動により環境に適応した群泳を可能にしている。本稿はそれぞれの行動を可能にするアルゴリズムを提案することで未知環境に適応可能な群移動を可能にした。具体的には、

各ロボットはセンサ範囲内で制御周期毎に二台のネイバーロボットを選択し、それらと一定距離を維持する。これは三台の隣接しているロボットが正三角形を形成でき、障害物が存在しているときも安定状態を保つことが可能である。これにより、群れは環境条件に応じて複数のグループに分かれること、あるいは、再び一団として復帰することができる。特に、本研究は単純な衝突の回避による適応ではなく、チーム単位での適応に主眼を置いたことに注意されたい。本稿ではチームとして活動する群ロボットにとって環境に適応するために各ロボットが自由に回避することでチームを失うことは好ましくないと考えた。そのため、チームの自己組織化の明確化とこれを維持したままの環境への適応的移動を重要視した。これにより、環境条件に対する群れのサイズや形状を自律的に適応できる。また、移動中においても一定距離を維持できるために、モバイルセンサネットワークモデルへの応用としても考えられる。それゆえ、定義した問題は大きな重要性を含んでいると考える。

本稿で提案するアルゴリズムの正当性はシミュレーションを通して確かめられた。結果は、群ロボットが局所的なフォーメーション形状を維持しながら複数の隘路を通過するために分裂と統合の過程を繰り返した。また、本稿で提案したフレームワークを用い、複数の移動するゴールの追従アルゴリズムを提案し、本研究の様々な応用の可能性を示した。この提案手法の応用として、UGV, UAV, UGV などを含む移動ロボット群、また、MEMS などによる体内診療を行うマイクロ・ナノロボット群まで幅広い自律ロボットに活用できることが考えられる。

第2章 問題の定式化

2.1 魚の群泳

魚の群泳は海面下の様々な環境条件に適応を可能にしている群行動の適例である（図 2.1）[22]．一般に，魚の群れは近接した魚との流体力学的な相互作用に基づき幾何学的なフォーメーションを形成・維持しながら泳ぐことが観察される．また，これを維持しながら，巨大な岩石や捕食者などの障害に直面した際にはグループを複数に分裂し回避，その後，再び1つの群れに統合することができる．例えば，マグロの群泳は局所的幾何学モデルであるダイヤモンド形状を形成することを Stocker は文献 [23] で示した．マグロはこのモデルを用いながら3つの興味深い群泳行動が観察される．

- 群泳中の幾何学的なフォーメーションの維持
- 捕食者や障害物等に遭遇した時の群れの分裂
- 捕食者や障害物等を回避した後の分裂した群れの統合

である．本論文では，環境に適応可能な群移動アルゴリズムの提案は前章で挙げたロボットモデルに多くの点で合致するマグロに観察されるこれらの群泳行動を模倣することで可能にする．これより，群ロボットの Adaptive Flocking 問題を次のように定義した．



図 2.1: 魚の群泳

2.2 問題定義

本稿では以降，Adaptive Flocking 問題を (図．2.2) に示すように 3 つの sub-problem に分解して各問題に取り組んだ．

Problem (Adaptive Flocking)

“ それぞれ異なった任意の位置に配置したロボット $r_1, \dots, r_n$ の群れは局所的に相互作用しながら与えられた任務を達成するために未知環境に適応可能な群移動を行う ”

Problem-1 (Team Maintenance)

“ それぞれ異なった任意の位置に配置したロボット $r_1, \dots, r_n$ の群れは移動しながら近接ロボットと局所的な幾何学的フォーメーションを形成，維持することができる． ”

Problem-2 (Team Partition)

“ 障害物などの環境の制約を検出したロボットは，局所的な幾何学的フォーメーションを維持しながら環境に適応した複数のチームに分裂することができる． ”

Problem-3 (Team Unification)

“ 接近している分裂した複数のチームが存在しているとき，局所的な幾何学的フォーメーションを維持しながら統合することができる． ”

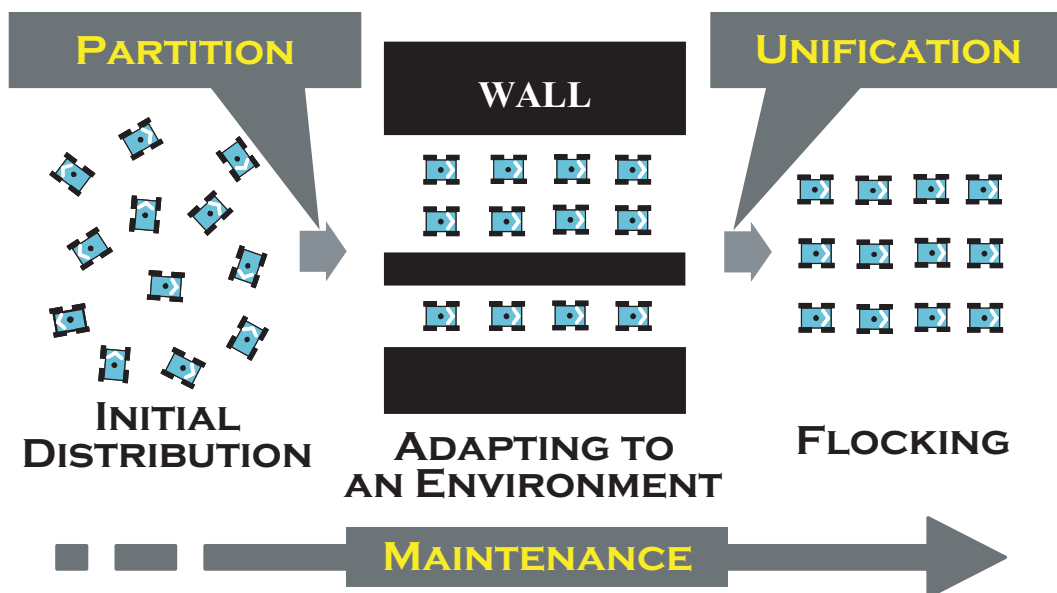


図 2.2: Adaptive Flocking Problem のコンセプト

2.3 ロボットモデル

定義した問題に取り組むに当たり，前章で簡単に述べたロボットモデルに基づき付加的な仮定を明確にする．

- ロボットは計算能力を持った点であり 2 次元平面上を自由に移動可能である．
- 全ロボットが同じアルゴリズムを実行する．
- 互いに独立にかつ非同期で動く．
- 局所的な有限のセンサ範囲を持ち，毎時刻における現在のエラーなしのセンシング情報のみを持ちうる．
- 与えられるゴールを事前に知らない（ゴールは光や化学物質などとして考え，これをロボットがセンシング可能であることを仮定する．）
- 壁や障害物のような環境を認識でき，それらとロボットを識別できる．（但し，センサ範囲に対して大きいものは測定できないこともある．）
- 群れ内部でメンバーの遅れが発生しないよう時刻 $t_{\max}$ を設定することで移動速度を制限する（ $t_{\max} = d_{\max}/v_{\max} = d_u/(\sqrt{3} \times v_{\max})$ ）

本研究の非同期の時刻 (time instant) は全ロボットにおいて $t, t+1, t+2, \dots$ の無限数列として表される．各時刻 t においてロボットはセンシングで取得した他ロボットの位置を入力とし，提案したアルゴリズムにより目標位置を計算，その位置に向かって動作という過程を 1 周期として，与えた仕事が達成されるまで無限に繰り返す．ロボットは上に述べた仮定より過去の行動や判断を記憶しない．この前提はメモリなしで提案されたアルゴリズムが本質的に自己安定化 (self-stabilization[24]¹) の特徴を持っていることを裏付ける [25]．ロボットのアルゴリズムは各周期で実行される function φ で構成される．この φ の引数は毎時刻で観測されたロボットの位置集合と各ロボットの現在の位置であり，戻り値は次のターゲット位置である．ロボットは過去の行動や判断を一切記憶しないため， φ の引数は現在の観測情報のみを使用することから，提案するアルゴリズムは自己安定化である結論に帰着する．

¹任意の配置から開始したシステムが所望する振る舞いに向かい，必ず収束する，という性質である

表 2.1: 表記のまとめ

表記	説明
d_u	ロボット間の一定距離
SB	ロボットのセンサ範囲 (Sensing Boundary)
O_i	SB 内で観測したロボットの位置集合
r_i	ロボット (i は説明のための仮想 index)
p_i	r_i の位置
$s1$	r_i によって選出された第 1 ネイバーにつけられる index
$s2$	r_i によって選出された第 2 ネイバーにつけられる index
p_t	r_i の移動目標位置
p_b	p_i, p_{s1}, p_{s2} の 3 台のロボットの重心点
p_v	r_i によって設定された仮想ロボットの位置

2.4 まとめ

仮定したのロボットモデルを用い, 定義した Team Maintenance, Team Partition, Team Unification の各問題に取り組み, それぞれに対してアルゴリズムを提案する. 最後にそれらを結合することで Adaptive Flocking が可能となるアルゴリズムを確立する. 次章より各問題に対してアルゴリズムを提案する. その際に本稿で多様に用いられる表記とその説明を表. 2.1 に示す.

第3章 Flocking Framework

魚は極めて低い能力しか持ち合わせていない．しかしながら，得ることができる数少ない情報量のみで緻密なフォーメーションを形成することができる．これは，ある魚が近傍の魚との局所的な相互作用をもとに自己の位置を調整することで全体として緻密なフォーメーションを形成している．本研究ではこの魚の群泳で観察される相互作用を群ロボットに適用し，群行動の基本的フレームワークとして提案した．

3.1 アルゴリズム

本章では，ロボット間の局所的な相互作用 (*LI: Local Interaction*) を説明する．2.2 節より，すべての sub-problem において根本的に共通しているものは局所的な幾何学的フォーメーションの形成である．それゆえ，幾何学的形状形成のためのロボット間の局所的相互作用を提案する．本稿で提案する *LI* は， r_i が 2 台の“ネイバー”と呼ばれるロボットを選出し d_u を一辺の長さとする正三角形のフォーメーションを形成する．そのために，図 3.1-(a) に示すように， r_i は SB 内で観測した全ロボットの位置集合 O_i を取得する．図で示されるように， r_i は r_{s1} を p_i から最短距離に位置しているロボットを選択し，この位置を p_{s1} とする． r_{s2} は p_i から p_{s1} までの総距離が最小となる経由点に存在しているロボットを選択することで p_{s2} を取得する．本稿では，任意の時刻において，ロボットは SB 内で他ロボットを 2 台以上センシング可能であることを仮定する．これら 2 台のネイバーに対して図 3.1-(b) に表すように r_i が正三角形を形成するための相互作用として Algorithm-1 を実行する． r_i は p_i, p_{s1}, p_{s2} の 3 点の位置座標から目標位置を計算する． r_i は三角形の重心 p_b を計算し (Line 1)，自身のローカル座標の水平軸に対する線分 $\overline{p_{s1}p_{s2}}$ の角度を測定する (Line 2)． r_i はこの角度と垂直を成す直線上で p_b から $d_u/\sqrt{3}$ 離れた 2 地点のうちの最近点を目標位置 p_t として算出する (Line 3~4)．これにより， r_i は， $\{p_{s1}, p_{s2}\}$ に対して二等辺三角形を生成するように動作する．これを全ロボットが行うことによって，群れは正三角形を格子構造的に繋ぎ合わせたフォーメーションを形成する．このアルゴリズムの実行により形成される正三角形のフォーメーションを *TC (triangular geometry configuration)* と定義する．次に，このアルゴリズムによる *TC* への収束性について述べる．

ALGORITHM-1 LOCAL INTERACTION (code executed by a robot r_i which occupies a point p_i)

constant $d_u :=$ uniform range distance

FUNCTION $\varphi_{interaction}(\{p_{s1}, p_{s2}\}, p_i)$

1 $(b_x, b_y) := \text{barycenter}(\{p_{s1}, p_{s2}, p_i\})$

2 $\theta :=$ angle between $\overline{p_{s1}p_{s2}}$ and r_i 's local horizontal axis

3 $target_x := b_x + d_u \cos(\theta \pm \pi/2)/\sqrt{3}$

4 $target_y := b_y + d_u \sin(\theta \pm \pi/2)/\sqrt{3}$

5 $p_t := (target_x, target_y)$

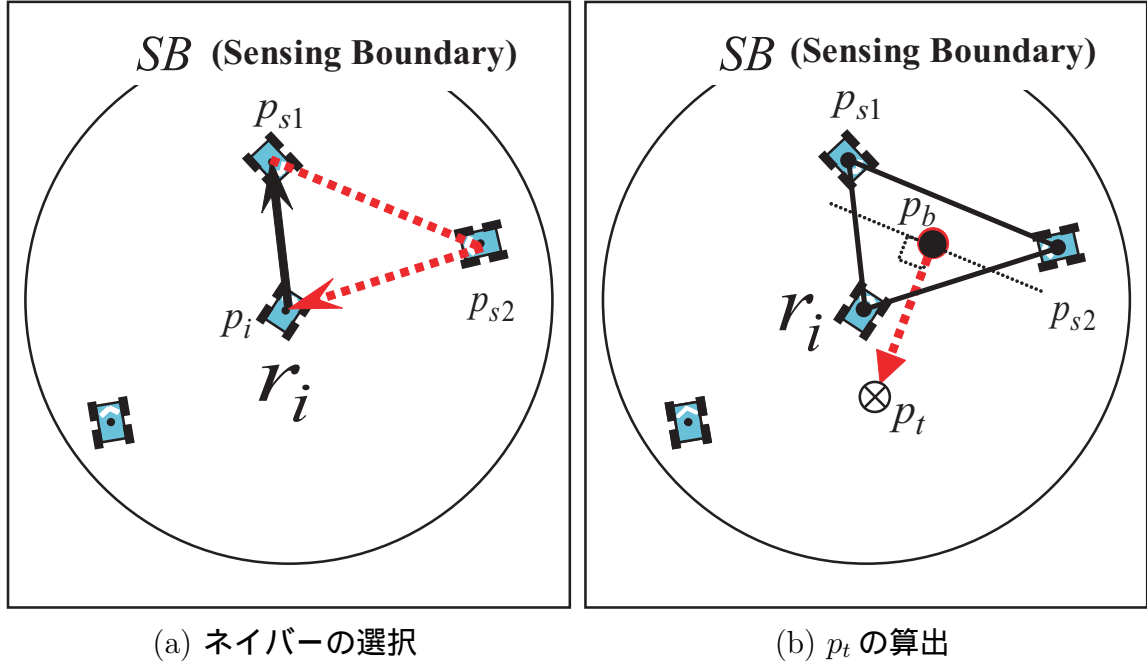


図 3.1: LIによる正三角形の形成

3.2 Local Interaction の収束性

図 3.2-(a) で示されるように 3 台のロボットによって形成される三角形 ABC を考える．図に示すように角度をそれぞれ α, β, γ とする．各ロボットは $\triangle ABC$ の頂点に位置し，次の時刻で新しい位置 A', B', C' に動く． $\triangle A'B'C'$ の角度は α', β', γ' ， O は $\triangle ABC$ の重心位置を表す．また， $\overline{AB}$ 上の P は O から $\overline{AB}$ に下ろした，同様に， $\overline{AC}$ 上の Q は O から $\overline{AC}$ に下ろした垂線の足である．四角形 $APOQ$ について考えると， P と Q の角度は直角である．したがって， $\angle POQ$ は $180^\circ - \alpha$ である．図から， $\angle B'OC'$ と $\angle POQ$ は対頂角のため等しい．また，Algorithm-1 (Line3~4) より $\overline{OB'}$ と $\overline{OC'}$ は $d_u/\sqrt{3}$ で等しく， $\triangle B'OC'$ は二等辺三角形である．したがって， α', β', γ' は図 3.2-(a) で示すようにそれぞれ $\alpha' = (\beta + \gamma)/2$ ， $\beta' = (\gamma + \alpha)/2$ ， $\gamma' = (\alpha + \beta)/2$ である．

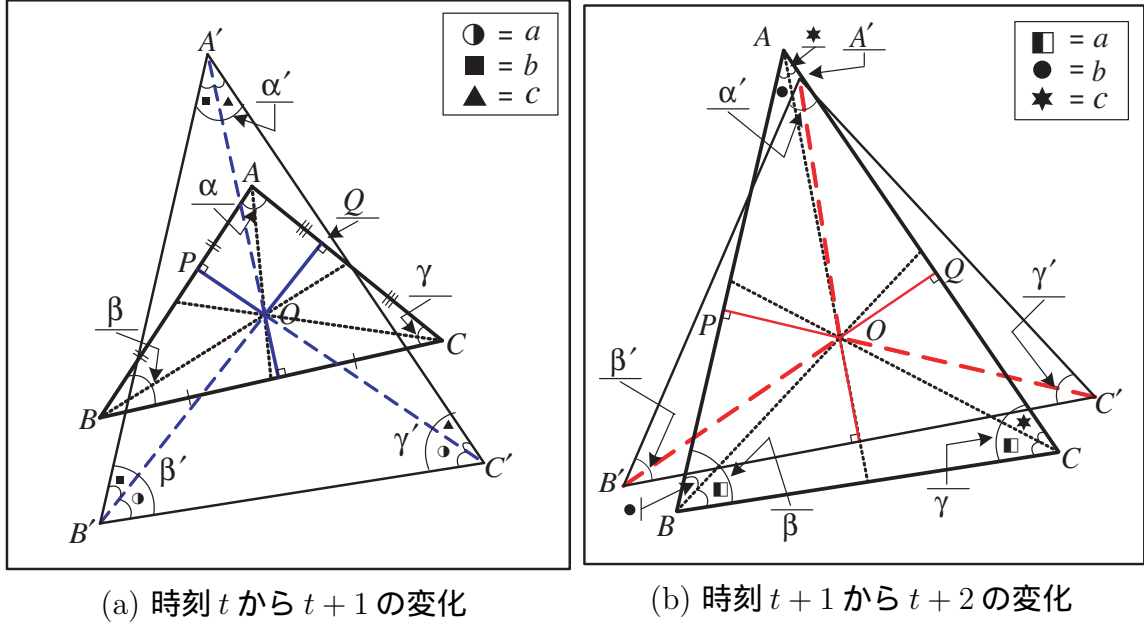


図 3.2: TC の形成過程

上述した角度 α, β, γ の関係に時間表現を導入すると，

$$\alpha' = (\beta + \gamma)/2 \rightarrow \alpha(t+1) = (\beta(t) + \gamma(t))/2 \quad (3.1)$$

として表される． $\beta(t+1), \gamma(t+1)$ においても同様に表現できる．図 3.2-(a) から (b) までの一連の流れにおける角度関係を簡略的に表現すると，次のような関係が導き出される．

$$\alpha(t+2) = \frac{\beta(t+1) + \gamma(t+1)}{2} = \frac{\alpha(t)}{2} + \frac{\beta(t) + \gamma(t)}{4} \quad (3.2)$$

$\beta(t+2)$ と $\gamma(t+2)$ においても同様である．(3.2) 式は更に (3.1) 式の関係式を用いて

$$\alpha(t+2) = \alpha(t+1)/2 + \alpha(t)/2 \quad (3.3)$$

のように表され， $\beta(t+2)$ $\gamma(t+2)$ も同様に表現可能である．ここで，(3.1) 式は β γ と共に

$$\begin{bmatrix} \alpha(t+1) \\ \beta(t+1) \\ \gamma(t+1) \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha(t) \\ \beta(t) \\ \gamma(t) \end{bmatrix} \quad (3.4)$$

のように行列記述ができる．同様に，(3.3) 式も

$$\begin{bmatrix} \alpha(t+2) \\ \beta(t+2) \\ \gamma(t+2) \end{bmatrix} = \frac{1}{2} \left(\begin{bmatrix} \alpha(t) \\ \beta(t) \\ \gamma(t) \end{bmatrix} + \begin{bmatrix} \alpha(t+1) \\ \beta(t+1) \\ \gamma(t+1) \end{bmatrix} \right) \quad (3.5)$$

として表される．(3.5) 式に (3.4) 式を代入すると $t+2$ と t の間の関係が

$$\begin{aligned} \begin{bmatrix} \alpha(t+2) \\ \beta(t+2) \\ \gamma(t+2) \end{bmatrix} &= \frac{1}{2} \begin{bmatrix} \alpha(t) \\ \beta(t) \\ \gamma(t) \end{bmatrix} + \frac{1}{2^2} \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha(t) \\ \beta(t) \\ \gamma(t) \end{bmatrix} \\ &= \frac{1}{2^2} \begin{bmatrix} \alpha(t) \\ \beta(t) \\ \gamma(t) \end{bmatrix} + \frac{3}{2^2} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \frac{\alpha(t) + \beta(t) + \gamma(t)}{3} \end{aligned} \quad (3.6)$$

のように表現できる．ここで，(3.6) 式より， $t+n$ と t の間の関係に拡張すると以下のような関係式が導出される．

$$\begin{aligned} \begin{bmatrix} \alpha(t+n) \\ \beta(t+n) \\ \gamma(t+n) \end{bmatrix} &= \frac{1}{2^n} \begin{bmatrix} \alpha(t) \\ \beta(t) \\ \gamma(t) \end{bmatrix} \\ &\quad + \frac{\sum_{k=1}^n 3 \cdot 2^{k-2}}{2^n} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \frac{\alpha(t) + \beta(t) + \gamma(t)}{3} \end{aligned} \quad (3.7)$$

これより， $n \rightarrow \infty$ として時間の極限を取ると，形成される三角形の3つの角度は

$$\begin{aligned} \lim_{n \rightarrow \infty} \begin{bmatrix} \alpha(t+n) \\ \beta(t+n) \\ \gamma(t+n) \end{bmatrix} &= 0 \times \begin{bmatrix} \alpha(t) \\ \beta(t) \\ \gamma(t) \end{bmatrix} + 1 \times \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \frac{\alpha(t) + \beta(t) + \gamma(t)}{3} \\ &= \begin{bmatrix} 60^\circ \\ 60^\circ \\ 60^\circ \end{bmatrix} \end{aligned} \quad (3.8)$$

のように算出され，正三角形状に収まっていくことが分かった．

多数のロボットの場合を考える． d_u の正三角形の集合構造の形成を証明するためにフェルマー点を使用した [26]．フェルマー点は任意の $\triangle ABC$ の外側における正三角形の建造を意味する．正三角形形成の過程で，もし $\triangle ABC$ が正三角形を生成したら，他の全ロボットがその正三角形に従い，同様に正三角形を形成していく．ここで， $\triangle ABC$ の正三角形形成は上記により証明された．したがって，複数の正三角形格子の形成は上記の証明とフェルマー点によって示されることがわかる．

一般にロボットは通信やセンシング，あるいは直接接触などによって LI の設計が可能である．しかし，本稿では上述したロボットモデルは SB の制限とその SB 内で観測したロボットの位置集合で可能なり，特に， $\varphi_{interaction}$ は現在のセンシング情報のみを用いる方法で正三角形のフォーメーション形成を達成させた．

第4章 Decomposed problems

本章は 2.2 節において定義した Adaptive Flocking 問題における sub-problem について取り組む．各節で Team Maintenance, Team Partition, Team Unification の各アルゴリズムを提案し, シミュレーションにより有効性を確かめる．本稿において各ロボットの p_t は前章で提案した LI によって算出される．そのため, 本章で提案するアルゴリズムはネイバーの選出手法であると考えられる．本章で提案する各アルゴリズムにより選択された 2 台のネイバーの位置情報を $\varphi_{interaction}$ に引数として受け渡すことにより最終的な p_t が算出されるからである．

4.1 Team Maintenance

本節は Problem-1 の Team Maintenance 問題に対するアルゴリズムについて述べる．Team Maintenance アルゴリズムの目的は群れがゴールに向かって移動しながら隣接するロボットと TC を維持することである．これに基づき Algorithm-2 を提案した．

4.1.1 アルゴリズム

このアルゴリズムの入力は SB の現在の観測情報 O_i と p_i である．初めに, r_i は任意時刻 t において, 自身のローカル座標における $\vec{G}$ を調整し (Line 1), SB と $\vec{G}$ から $\pm 90^\circ$ 内の領域の共通部分領域を $A(\vec{G})$ として定義する (Line 2)． r_i は r_{s1} を $A(\vec{G})$ 内で選択するため, $A(\vec{G})$ 内のロボットの集合を A_g として定義する (Line 3)． A_g にロボットが存在するとき, その中で最短距離に位置するロボットを r_{s1} に選択する (Line 4~5) (図．4.1-(a))．逆に, A_g にロボットが存在しないとき, r_i から距離 $k \times d_u$ だけ離れた p_v を $\vec{G}$ 上に設定し, ここに r_{s1} の存在を仮定する (Line 6~7)．これにより, r_{s1} とその位置 p_{s1} を取得する． r_{s2} は p_i から p_{s1} までの総距離が最小となる経由点に存在しているロボットを選択する (Line 9) (図．4.1-(b))． $\{p_{s1}, p_{s2}\}$ を選出して, r_i は TC を形成するよう $\varphi_{interaction}$ を使用し移動目標位置を取得する (Line 10)．このアルゴリズムにより $\vec{G}$ 方向に向かうネイバー関係が構築される．したがって, 同じチームとしてのつながりとそれらとの TC の維持が可能になる．よって, 提案したアルゴリズムによりロボット 3 台で TC を形成し, さらに $\vec{G}$ 方向へ移動しながら群れが全体のつながりを維持することが可能となる．

ALGORITHM-2 TEAM MAINTENANCE (code executed by a robot r_i which occupies a point p_i)

constant $d_u :=$ uniform range distance

$O_i := \{p_1, \dots, p_n\}$ //observation set

FUNCTION $\varphi_{maintenance}(O_i, p_i)$

1 $\vec{G} :=$ goal direction

2 $A(\vec{G}) :=$ goal directional area

3 $A_g := \{ \text{set of positions of robots located in } A(\vec{G}) \}$

4 **IF** $\{\exists p \in A_g\}$ **THEN**

5 $p_{s1} := \min_{p \in A_g} [dist(p_i, p)]$ //1st neighbor

6 **ELSE** $\{\forall p \notin A_g\}$

7 $p_{s1} := p_v$ located on $\vec{G}$ apart from $k \times d_u$ //1st neighbor

8 **END IF**

9 $p_{s2} := \min_{p \in O_i - \{p_{s1}\}} [dist(p_{s1}, p) + dist(p, p_i)]$ //2nd neighbor

10 $p_t := \varphi_{interaction}(\{p_{s1}, p_{s2}\}, p_i)$

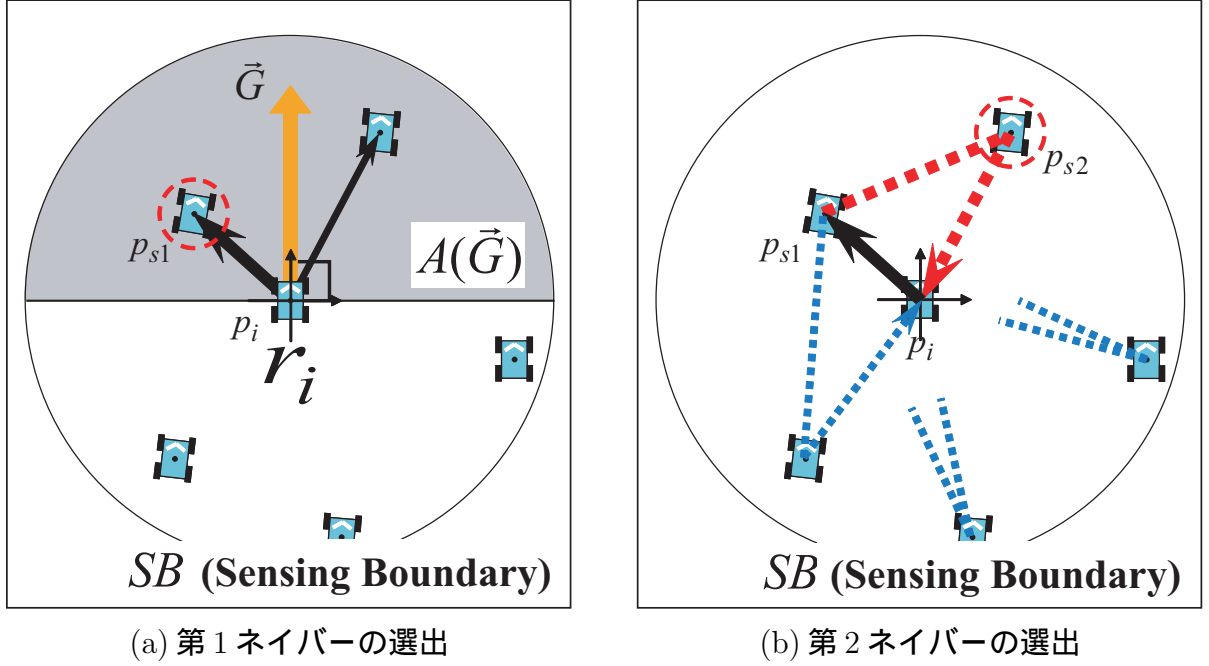
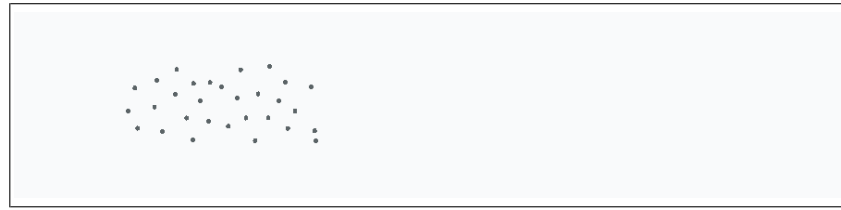


図 4.1: Team Maintenance

4.1.2 シミュレーション

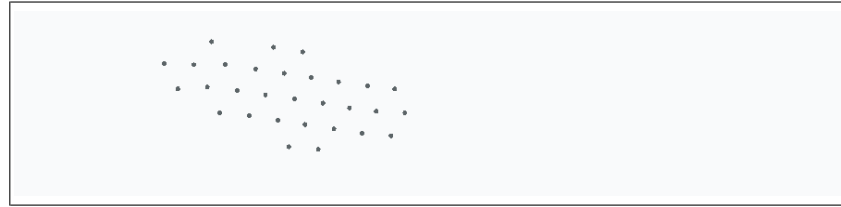
図 4.2, 4.3 は Team Maintenance アルゴリズムのシミュレーション結果である．シミュレーションにおいて必要となるパラメータとして p_v を設定するための係数 k を 1.2 とし，各ロボットが d_u を保ち TC を維持しながら，図中右側のゴールに向かって移動させた．ロボットは初めに図 4.2-(a) に示したように平面上の異なる任意の位置に配置した．図 4.2-(b),(c) では，各ロボットはゴールに向かって動きながら TC を生成した．また，図 4.2-(d) ~ (f) では，群れが TC を維持しながら移動できることを示した．図 4.3 は図 4.2 の結果を基に， TC を形成しながら，各ロボットの位置を中心としてその周辺に距離 d_u の地点に配置しているロボット台数の時間遷移に対する変化を表している．例えば，提案した LI によると自身が群れの内部に位置しているとき，距離 d_u に配置しているロボット台数は 6 台である．これを基に，図 4.2-(d) と 4.3 の 10[sec] 時の値を比較すると d_u 配置台数が 6 台のロボット台数は 8 台である．つまり，群れの内部に位置しているロボットが 8 台いることを意味しており，図 4.2-(d) を見ればその様子わかる．本稿では，これをロボットの接続性 (connectivity) と定義する．シミュレーション結果から群れはおよそ 3[sec] 後以降は変化がなく，全ロボットが TC を維持しながら移動ができたことを確認できた．このシミュレーションにおいて，いずれのロボットも $\varphi_{maintenance}$ によってゴールに向かって TC を形成し，群れを維持しながらゴールまで群移動することが可能であることを確認できた．



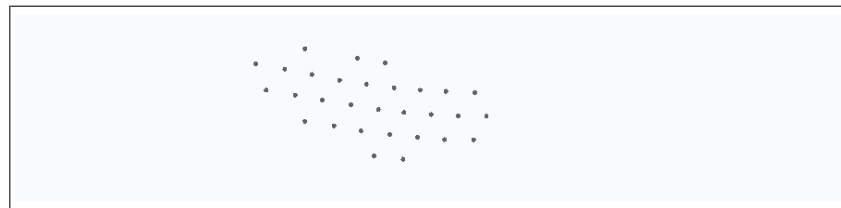
(a)0[sec]



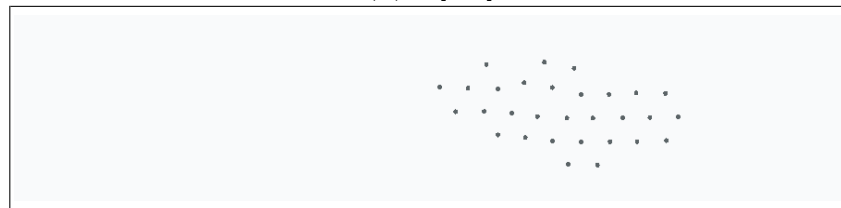
(b)2[sec]



(c)5[sec]



(d)10[sec]



(e)20[sec]



(f)30[sec]

図 4.2: Team Maintenance のシミュレーション結果

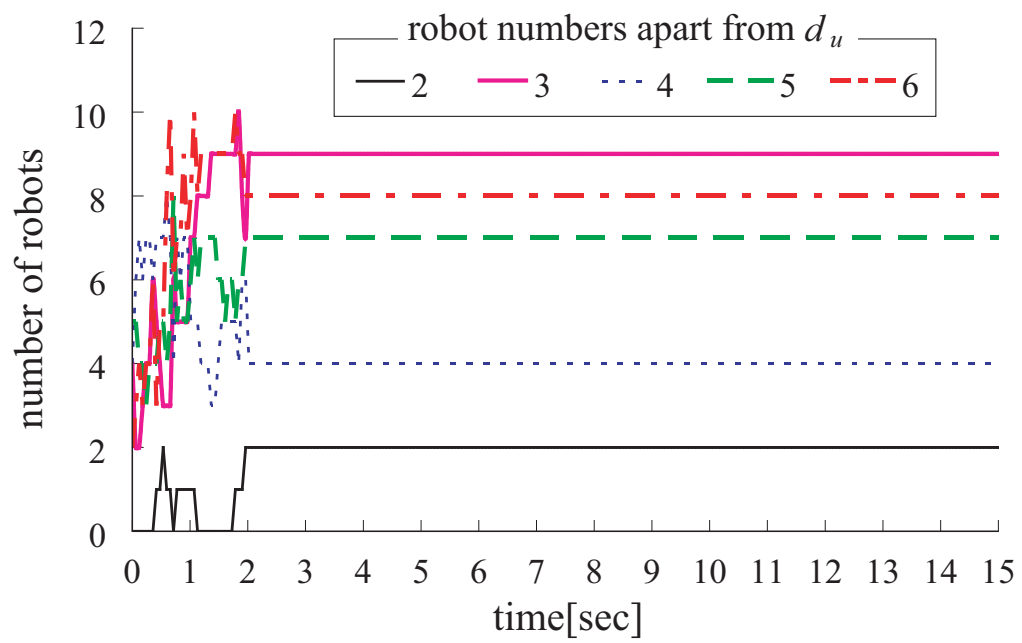


図 4.3: 図 4.2 のシミュレーションにおける時間変化に対する接続性の変化

4.2 Team Partition

本節は Problem-2 として定義した Team Partition 問題のアルゴリズムについて述べる．このアルゴリズムは各ロボットを SB で発見した環境条件に適応させることで群れ全体として分裂を発生させる．このアルゴリズムの主要部分は，群れを環境に適応させるための各ロボット単位での判断である．つまり，与えられたゴールへの移動任務を遂行するために進行する方向を各ロボットによって判断させる必要がある．これは全てのロボットを維持したままでは分裂を起こすことができないため，通常の全ロボットを維持した状態からネイバー関係を崩さなければならないからである．また，そのネイバー関係は各隘路に対してチームを組織できるようにしなければならない．そのため，各ロボットは進行する隘路を明確に定め，ネイバーの選択領域を確保し，隘路ごとにチームを形成することができる．

4.2.1 *favorite vector*

本稿で提案した Team Partition の方法は様々な環境条件に対する進行方向の判断を助けるために万有引力の法則 [27] を応用した *favorite vector* $\vec{f}$ と称される判断指標と成るベクトルを導入した．このベクトルは本来の物理的な意味の引力としてではなく複数の隘路に対する相対的な誘引強度として使用する．特に，移動経路上に障害物によって形成された隘路は群れを複数の小チームに分裂するように促す通路としてモデル化した（図．4.4）．各ロボットは $\vec{f}$ を用い，最大の $\vec{f}$ を選択することで r_i の通る隘路を決定する．ロボットが図．4.4 で示されるような隘路に遭遇したときを考える． d_j をロボットから隘路 s_j の中心位置までの距離， w_j を隘路幅とすると，隘路 s_j に対するロボットの $\vec{f}_j$ は

$$|\vec{f}_j| = w_j/d_j^2 \quad (4.1)$$

と定義する．各ロボットには限られたセンサ範囲があり，時折，隘路片側の障害物の端面しか見えないために隘路として認識できず幅 w_j を測定できないことも考えられる．この場合，ロボットはセンサ範囲を知っているため，その端面からセンサの限界線上までの距離を s_j の幅と見なす（この詳細については次章で扱うため，ここでの議論は行わないこととする．）． $\vec{f}$ の集合を $\{\vec{f}_j | 1 \leq j \leq m\}$ と表現すると，ロボットはこの集合の中で最大の $|\vec{f}_j|$ を探す．最大の $|\vec{f}_j|$ として判断されたベクトルを特に， $\vec{f}_{\max}$ として表す．この方向を各ロボットが進行したい方向とすることで各々が環境に適応しながら進行することができる．

4.2.2 アルゴリズム

前項で説明したようなコンセプトに基づき Team Partition アルゴリズムを提案する．Algorithm-3 は隘路を発見した条件のときに実行される．ロボットは各隘路に対する $\vec{f}_j$ を計算，その中で最大の $|\vec{f}_j|$ を探し，そのベクトルを r_i が進行したい方向 $\vec{f}_{\max}$ とする (Line

1) . r_{s1} を $A(\vec{f}_{\max})$ 内で選択するため (Line 4~8) , $\vec{f}_{\max}$ 方向に $A(\vec{f}_{\max})$ を取得し (Line 2) .
 さらに , $A(\vec{f}_{\max})$ 内のロボットの集合 A_p を定義する (Line 3) . A_p にロボットが存在する
 ときは最短距離に位置するロボット r_{s1} から , 逆に , A_p にロボットが存在しないときは r_i
 から距離 $k \times d_u$ だけ離れた $\vec{f}_{\max}$ 上の仮想地点 p_v に存在を仮定したロボット r_{s1} により ,
 p_{s1} を得る (Line 4~8) . r_{s2} を p_i から p_{s1} までの総距離が最小となる経路点に存在してい
 るロボットから選択することで 3 点 p_i, p_{s1}, p_{s2} を取得し (Line 9) , $\varphi_{interaction}$ を使用し移動
 目標位置を取得する (Line 10) . このアルゴリズムにより各ロボットが進行すべき $\vec{f}_{\max}$ 方
 向を取得することができ , その方向へ進行するためのネイバーを適切に選ぶことができる .
 したがって , このアルゴリズムにより $\vec{f}_{\max}$ 方向に向かうネイバー関係が構築される .
 従って , 同じ隘路を選択したロボット同士でチームとしてのつながりとそれらとの TC の
 維持が可能になる . このようにして群れの分裂を発生させることができる . このように群
 れは LI により , ロボット 3 台での TC を形成し , さらに $\vec{f}_{\max}$ 方向へ移動しながら群れが
 分裂したチーム間でのつながりを維持することが可能になる .

ALGORITHM-3 TEAM PARTITION (code executed by a robot r_i which occupies
 a point p_i)

constant $d_u :=$ uniform range distance

$O_i := \{p_1, \dots, p_n\}$ //observation set

$S := \{s_1, \dots, s_m\}$ //observed strait set

FUNCTION $\varphi_{partition}(O_i, p_i)$

1 $\vec{f}_{\max} := \max_{s_j \in S} [|\vec{f}_j|]$

2 $A(\vec{f}_{\max}) :=$ favorite area

3 $A_p := \{ \text{set of positions of robots located in } A(\vec{f}_{\max}) \}$

4 **IF** $\{\exists p \in A_p\}$ **THEN**

5 $p_{s1} := \min_{p \in A_p} [dist(p_i, p)]$ //1st neighbor

6 **ELSE** $\{\forall p \notin A_p\}$

7 $p_{s1} := p_v$ located on $\vec{f}_{\max}$ apart from d_u //1st neighbor

8 **END IF**

9 $p_{s2} := \min_{p \in O_i - \{p_{s1}\}} [dist(p_{s1}, p) + dist(p, p_i)]$ //2nd neighbor

10 $p_t := \varphi_{interaction}(\{p_{s1}, p_{s2}\}, p_i)$

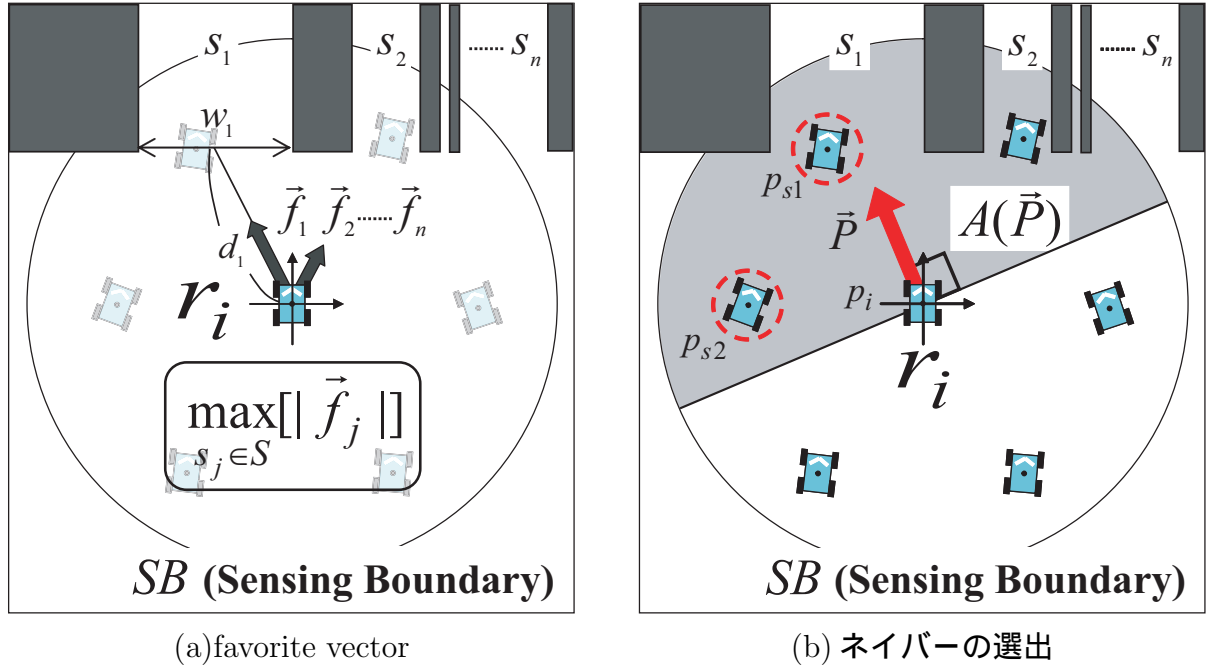
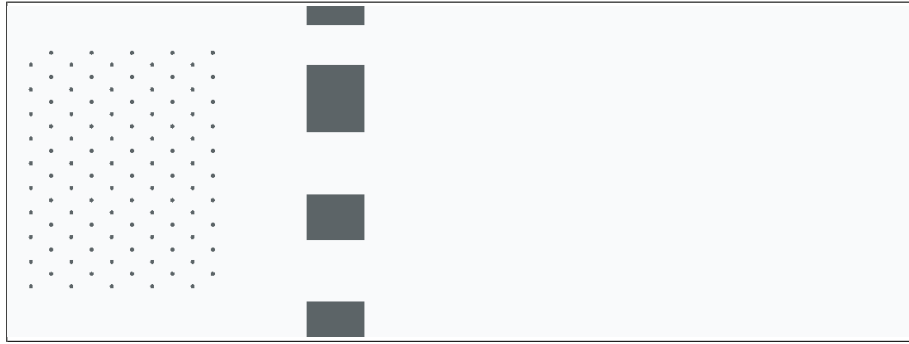


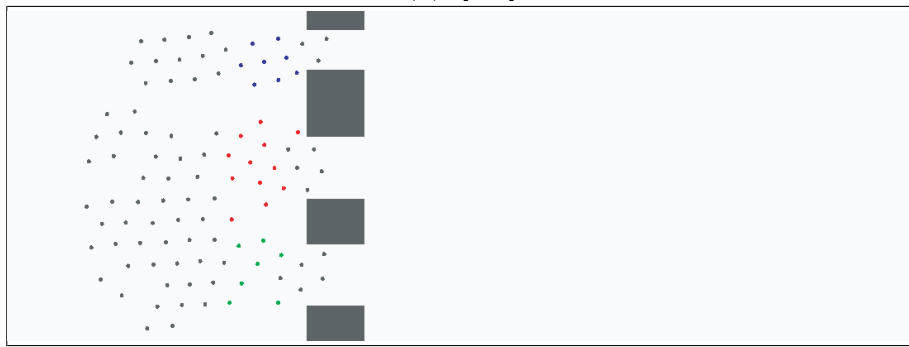
図 4.4: Team Partition

4.2.3 シミュレーション

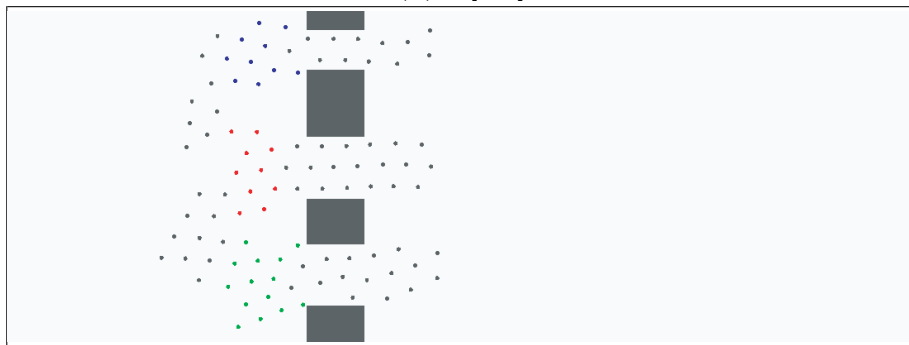
図 4.5 で示したシミュレーションでは，提案した Algorithm-3 によって群れを環境条件に応じた複数のチームに分裂させた．Maintenance のときと同様で p_v を設定するための係数 k は 1.2 とした．ここでは隘路は 3 本とし，群れは初期状態において一様に幾何形状を形成させた状態で配置した．群移動の途中で，障害物環境に遭遇すると，各ロボットは最大の $\vec{f}_j$ を選択することで分裂判断を行い， TC を維持しながら 3 チームに分かれることができた．また，隘路が SB にないロボットは Algorithm-3 を実行せず，Algorithm-2 の Team Maintenance を実行しているのであるが，隘路が見えていなくとも，ロボットは前方のネイバーの動きに従うことで自然に環境条件に適応することができることもまた，このシミュレーションは示している．ゆえに，群れは LI により環境の制約に対しても適応的に TC を形成し，チームを維持することができた．



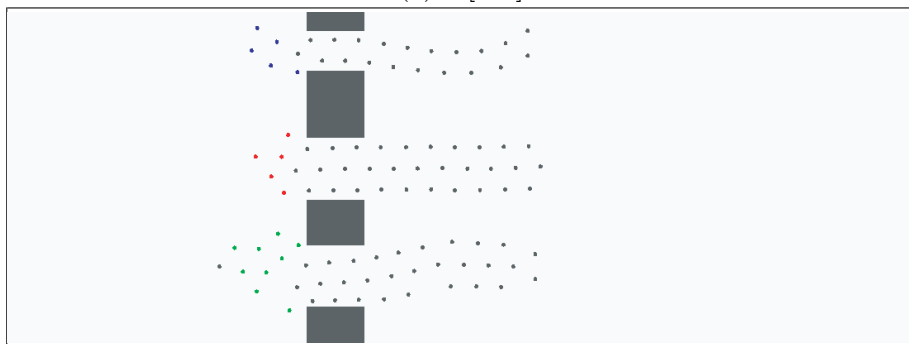
(a)0[sec]



(b)10[sec]



(c)20[sec]



(d)30[sec]

図 4.5: Team Partition のシミュレーション結果

4.3 Team Unification

本章は Problem-3 の Team Unification 問題に対するアルゴリズムについて述べる．これはロボット r_i が分裂した他チームを発見したとき，再び群れを統合させることができることを目的とする．Team Unification によって分かれたつながりを一つにするようにネイバーを選択する必要がある．解決手法として 2 台のネイバーのうち 1 台を離れたロボットから，もう 1 台を現在の所属したチームと考えられるロボットから選択することで全ロボットのつながりを一つにしようと働きかける手法を考案した．しかし，ロボットは SB の局所性の制約があるため分かれたロボットを発見できないときは Team Unification の働きを期待できない．離れたロボットが発見できた場合にのみ統合を試みる．

4.3.1 アルゴリズム

はじめに r_i はローカル座標における $\vec{G}$ を調整する (Line 1)．次に，距離 d_u 以内に配置しているロボットの位置集合を D_u として取得する (Line 2)． $\vec{G}$ に対して最も角度が最小になるベクトル $\overrightarrow{p_i p_u}$ を形成している D_u のロボット p_u から p_{ref} を選ぶ (Line 3)．図 4.6-(a) に表したように，この p_{ref} を基点として $\overrightarrow{p_i p_{ref}}$ から最も右回りに位置しているロボットを p_{rn} (Line 4)，逆に最も左回りに位置しているロボットを p_{ln} (Line 5) として $\overrightarrow{p_i p_{ref}}$ から 60° 刻みで探す．ここで， r_{s1} を探すためにこれまでの領域と異なる特殊な領域 $A(U)$ を取得する (Line 6)．この領域は p_{rn} と p_{ln} によって挟まれる D_u に所属するロボットが存在しない側の SB 領域と， $A(\vec{G})$ との共通部分領域として定義される (図 4.6-(a))．そして，図 4.6-(b) に示されるように， $A(U)$ に位置するロボットの集合を A_u として定義し (Line 7)，この集合から最短距離のロボットを p_{s1} として選択する (Line 8)．さらに， $\{p_{rn}, p_{ln}\}$ の 2 台のうち p_{s1} への総距離が最短となる経路点に位置しているロボットを p_{s2} として選択する (Line 9)．最終的な出力として $\varphi_{interaction}$ に 3 点 p_i, p_{s1}, p_{s2} を入力することで r_i は移動目標位置を取得する (Line 10)．このアルゴリズムの実行により，全ロボットが TC を形成しながら群れを統合し，群移動を行う．要約すると，このアルゴリズムは各ロボットが自分が群れの外側に位置しているか否か，また，さらに外側にロボットがいるか否かの両条件を認識した時に実行され，認識したロボットから順次統合が行われる．

前述したように，Team Unification は必ずしも実行されない．これは Algorithm-4 の中に明記しなかった IF 文が潜在していることを意味する．つまり，Line7 と Line8 の間に以下のような処理を行っている．

```
IF {  $A_u = \phi$  }   Team Unification を break;
ELSE              Team Unification を続行
```

つまり，他チームのメンバーであると判断されたロボットの集合 A_u が空集合の場合は実行されない．全体の流れは次章でフローチャートと共に説明を行うためここでは深く言及しない．

ALGORITHM-4 TEAM UNIFICATION (code executed by a robot r_i which occupies a point p_i)

constant $d_u :=$ uniform range distance

$O_i := \{p_1, \dots, p_n\}$ // observation set

FUNCTION $\varphi_{unification}(O_i, p_i)$

1 $\vec{G} :=$ goal direction

2 $D_u := \{ \text{position set of robots located in the range } d_u \}$

3 $p_{ref} := \min_{p_u \in D_u} [|\text{angle}(\vec{G}, \overrightarrow{p_i p_u})|]$

4 $p_{rn} :=$ farthest position in the right-hand direction of $\overrightarrow{p_i p_{ref}}$

5 $p_{ln} :=$ farthest position in the left-hand direction of $\overrightarrow{p_i p_{ref}}$

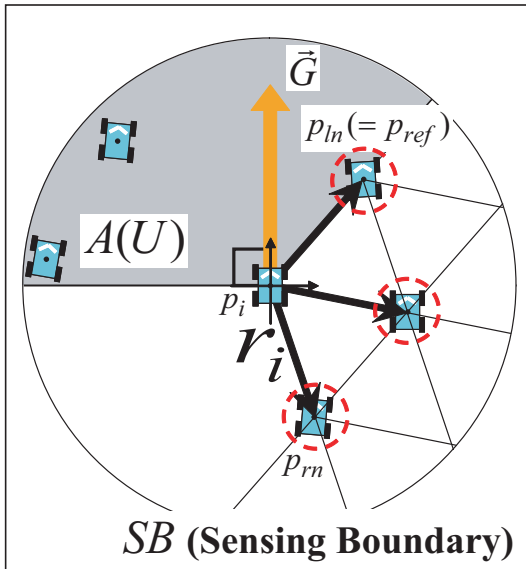
6 $A(U) :=$ unification area

7 $A_u := \{ \text{set of positions of robots located in } A(U) \}$

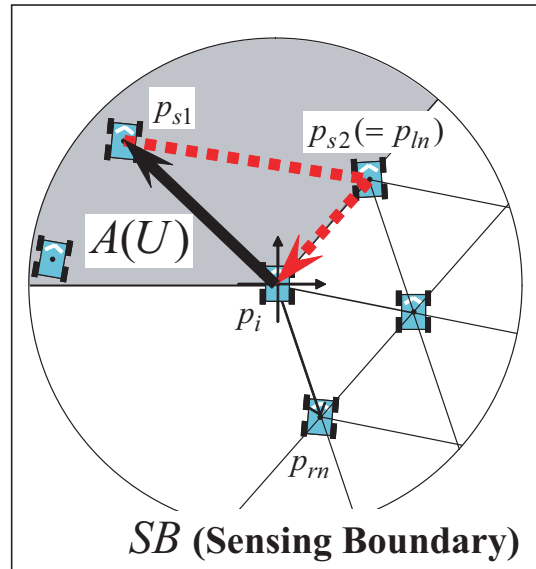
8 $p_{s1} := \min_{p \in A_u - \{p_{rn}, p_{ln}\}} [\text{dist}(p_i, p)]$ //1st neighbor

9 $p_{s2} := \min_{p \in \{p_{rn}, p_{ln}\}} [\text{dist}(p_{s1}, p) + \text{dist}(p, p_i)]$ //2nd neighbor

10 $p_t := \varphi_{interaction}(\{p_{s1}, p_{s2}\}, p_i)$



(a) unification area



(b) ネイバーの選出

図 4.6: Team Unification

4.3.2 シミュレーション

図 4.7 は Algorithm-3 によるシミュレーション結果を示す．このシミュレーションでは，100 台のロボットに統合を行なわせた．はじめに 2 チームに分かれていたロボット群は引き合い，一様に TC を形成した．このアルゴリズムは SB に別の分裂したチームのメンバーを発見したときに実行される．このアルゴリズムにより，ロボットは互いに歩み寄ることで統合が達成される．提案した Team Unification アルゴリズムにより，群れが統合し一様に TC を形成することが確認できた．

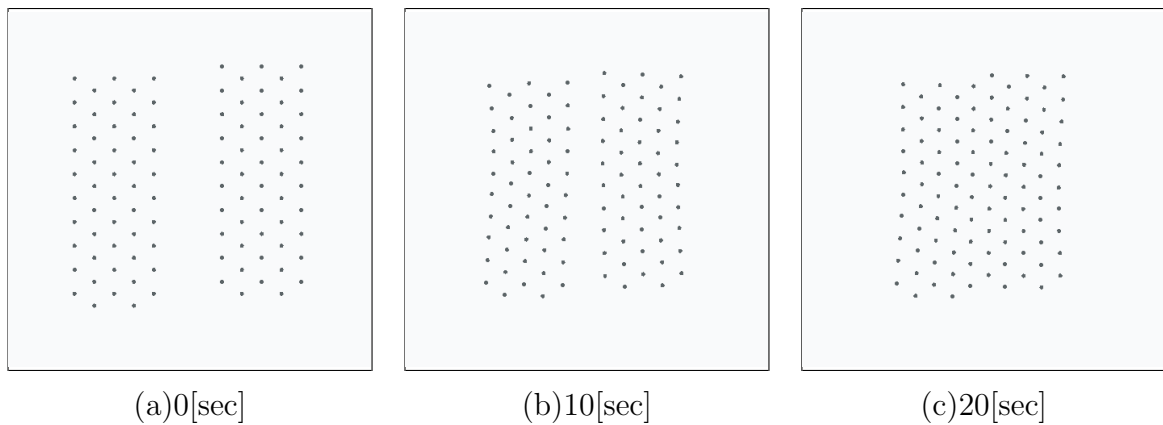


図 4.7: Team Unification のシミュレーション結果

第5章 Adaptive Flocking

本章では Team Maintenance , Team Partition , Team Unification の3つのアルゴリズムを結合させることで、群ロボットが環境に適応しながら TC を維持し群移動を行なうことができる Adaptive Flocking アルゴリズムを確立する．次節から結合したアルゴリズムを簡単に説明し、シミュレーション結果を通して有効性を示す．また、3つのアルゴリズムの組み合わせを変えることで各アルゴリズムの役割の分析と考察を行なう．

5.1 アルゴリズム

Adaptive Flocking 問題は3つの sub-problem に分解して扱うことでそれぞれのアルゴリズムを提案した．Adaptive Flocking アルゴリズムは図．5.1 に示したような流れを毎時刻に全ロボットが独立にかつ非同期で実行する．このアルゴリズムの INPUT は毎時刻に各ロボットのローカル座標で得た他ロボットの位置集合 O_i と p_i , また、観測された環境情報である．OUTPUT として各ロボットは次の移動目標位置を取得する．ロボットは Adaptive Flocking アルゴリズムを毎時刻に実行する際には、はじめに $A(\vec{G})$ 内に隘路が存在するか否かを確認する．そこに隘路が存在した場合、ロボットは Team Paritition アルゴリズムを実行し分裂を行う． $A(\vec{G})$ 内に隘路を確認しなかったロボットは次に、ロボットの配置状況から Team Unification を行うか否かの判断を行う．群れの外側に位置しているロボットがさらに外側にロボットを認識した時、Team Unification アルゴリズムを実行し、統合を行う．このいずれの状況も認識しなかったロボットは Team Maintenance アルゴリズムを実行し、 TC を維持し移動を行う．これが本稿で提案する Adaptive Flocking の一連の流れである．

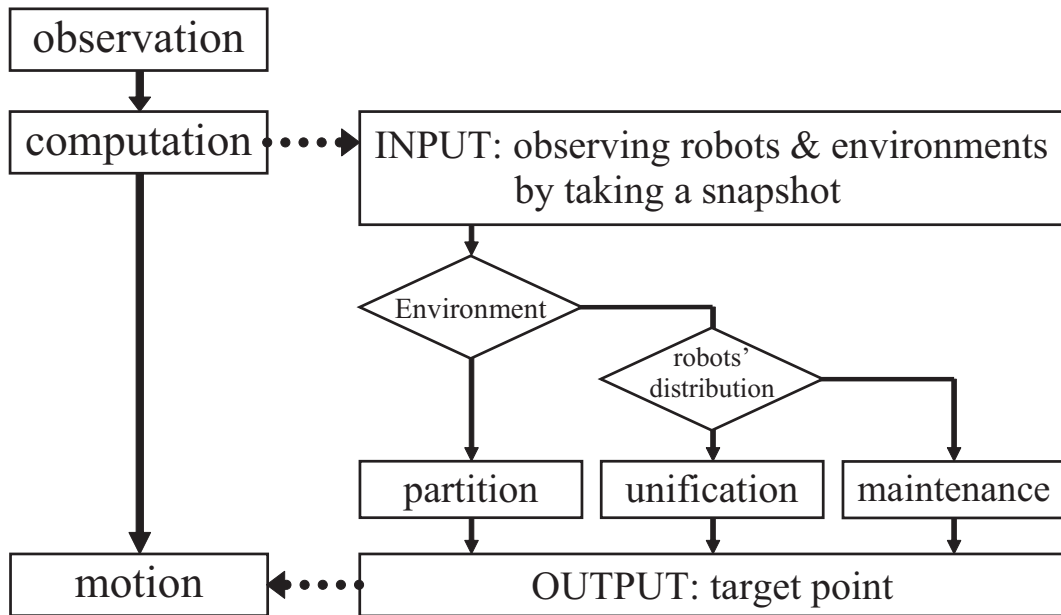


図 5.1: 各ロボットが実行する Adaptive Flocking のフローチャート

5.2 シミュレーション

組み合わせた Adaptive Flocking アルゴリズムを用いてシミュレーションを行った。はじめに，図 5.2 に示したようなシミュレーション環境で Adaptive Flocking アルゴリズムを実行した。これは図 4.5 と同じ環境であるが，紙面の関係上，90° 回転させて表示している。また，図 4.5 と異なり，初期状態においてバラバラに配置した 100 台のロボットを用いた。バラバラに配置された群れは *TC* を形成しながら移動を開始した。図 5.2-(c) で，群れは Team Partition により，三隘路に対して適応した分裂を始めた。各チームは自律的にチームとしての組織を維持したまま進行することも確認できた。図 5.2 と異なる点は，隘路通過後の Team Unification である。図 5.2-(f) では環境に適応するために分裂した三つのチームは Team Unification により初期の 1 チームとして一様に *TC* を形成し，下側に設定したゴールに向かって移動を続けることができた。以上より，提案した Adaptive Flocking アルゴリズムは初期状態でバラバラに配置した群ロボットが *TC* を形成，維持しながら環境に適応可能な複数のチームに分裂，統合が可能であることが確認できた。

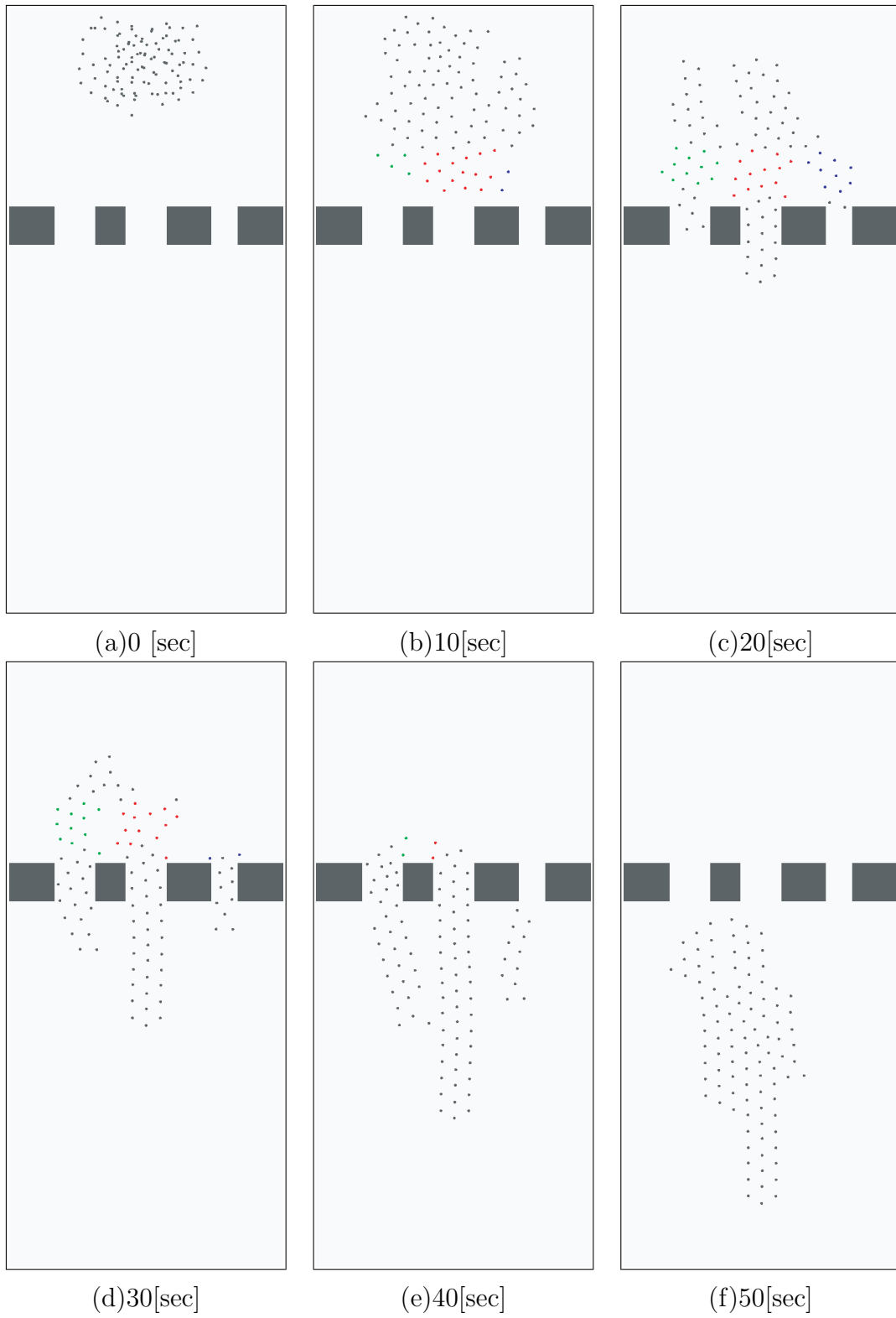


図 5.2: Adaptive Flocking のシミュレーション結果

次に提案した Adaptive Flocking を用い、図 5.4 ようなシミュレーションを行った。同様に紙面の都合上 90° 回転させて表示している。これは提案した手法が隘路が存在している場合にのみ有効に働くのではなく障害物の回避行動をも可能となることを示すためにを行った。これは障害物の両端面が見えていないときの認識方法により障害物の両側に隘路があることを仮想的に設定させることで可能としている。図 5.3 にこの認識方法の説明を示した。SB は局所的で限界があるためにこのような状況は多くの場面で起こっている。ロボットは自身の SB を知っている。そのため、障害物の片側は見えているが逆側が見えていないときは自身の SB の限界線上に仮想的に隘路の逆側端面の存在を仮定する。例えば、図 5.3 の場合はロボットから見て、最左端に幅 w_1 の仮想隘路 s_1 が存在していることを認識している。この情報を基に Team Partition により進行する隘路を判断する。図 5.4 はこの認識方法を積極的に活用した代表的な例として示した。初期状態でバラバラに配置した 100 台のロボットは下側のゴールを目指して TC を形成しながらチームを組織し、群移動を始めた。障害物を発見したロボットはその両側に仮想的に隘路を創造し、適応的に分裂を行った (図 5.4-(b) ~ (d))。また、障害物を通過後は再び統合を行い一様に TC を形成した。この結果から提案した Adaptive Flocking は隘路の存在の有無に関わらず環境に適応できることが明確に確認できた。

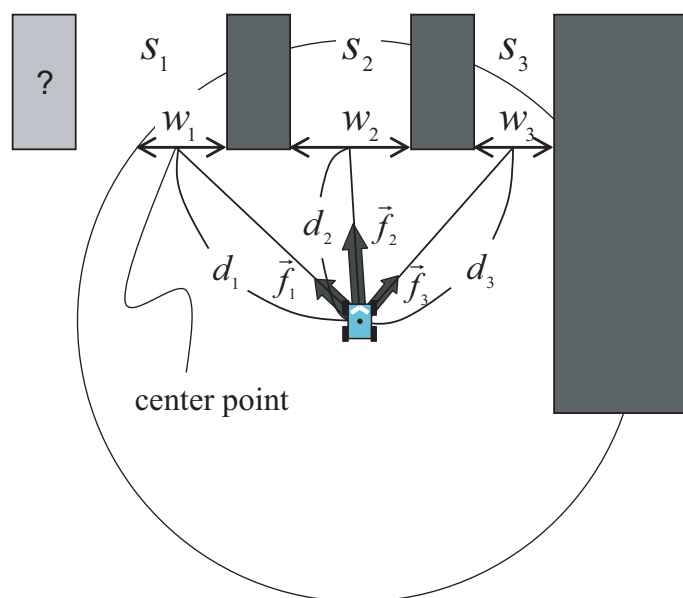


図 5.3: 不完全な隘路の認識方法



図 5.4: 単純な障害物の回避群移動

さらに、この発展として群ロボットに図. 5.5 のような未知環境下で適応的な群移動を行わせた。初期状態でバラバラに配置していた群れは局所的に相互作用しながら図中下側のゴールに向かって群移動を行った。群れは $\vec{G}$ 方向に群移動している途中にはじめに観測した障害物に遭遇すると、Team Partition によりその環境に対して効果的に適応できる2つのチームに分裂した。SB が局所的であるため後方のロボットは隘路が見えていないにも関わらず、前方のネイバーに続くことで環境に適応するように分裂できた。そして、2つのチームに分裂した後、各チームは群移動しながら TC を維持した。図. 5.5-(d) では、分裂したチームが Team Unification によって再び統合し、新たに発見した障害物によって更に分裂を行った。このようにアルゴリズムを繰り返すことで、群ロボットは与えられたゴールに達することができた。したがって、提案したアルゴリズムは様々な障害物が存在している未知環境においても群ロボットが TC を形成、維持しながら環境に適応可能な複数のチームに分裂、統合することができるアプローチであることが確認できた。この未知環境に適応的に群移動を行うシミュレーション結果から、提案した Adaptive Flocking アルゴリズムの有効性が確認できた。

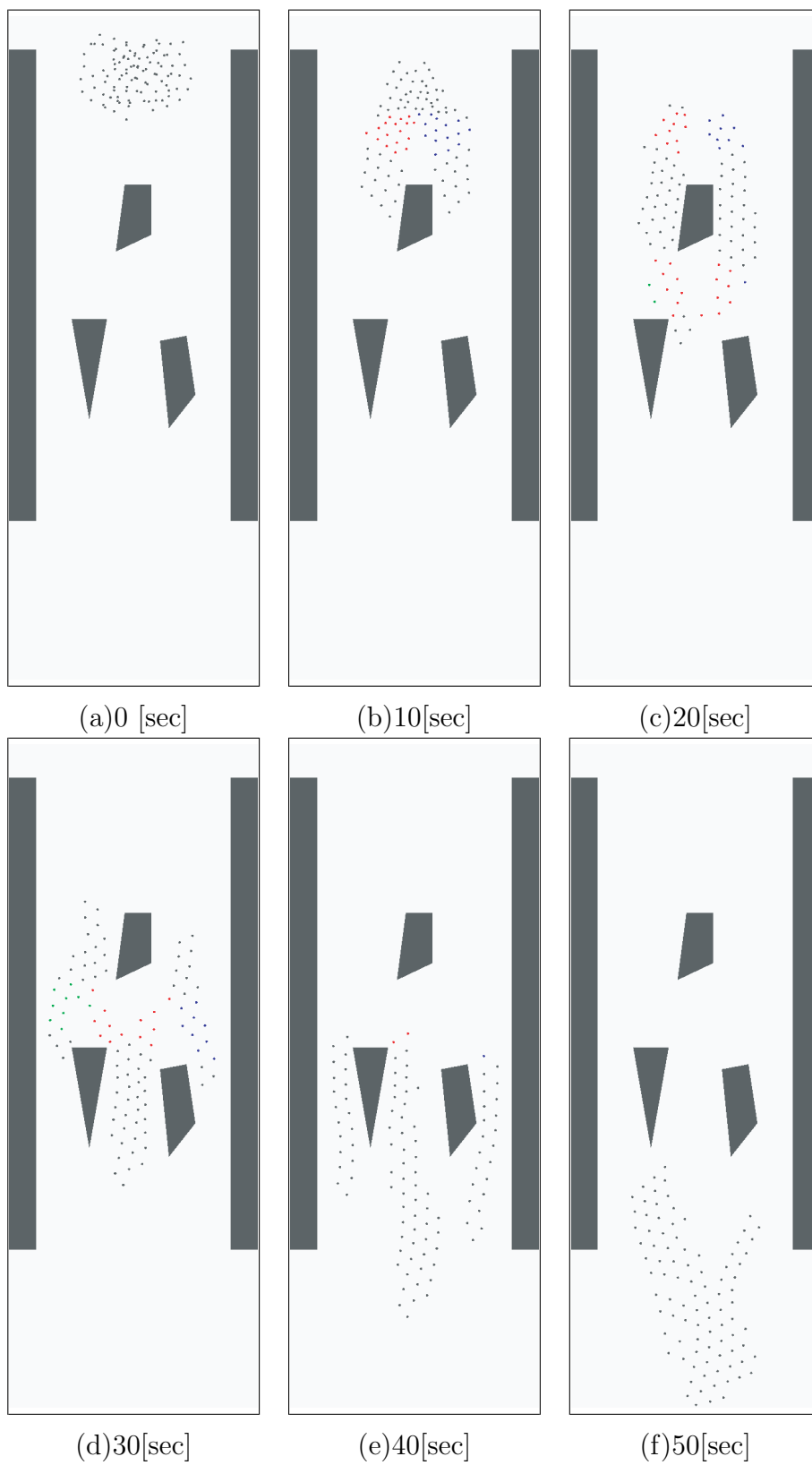
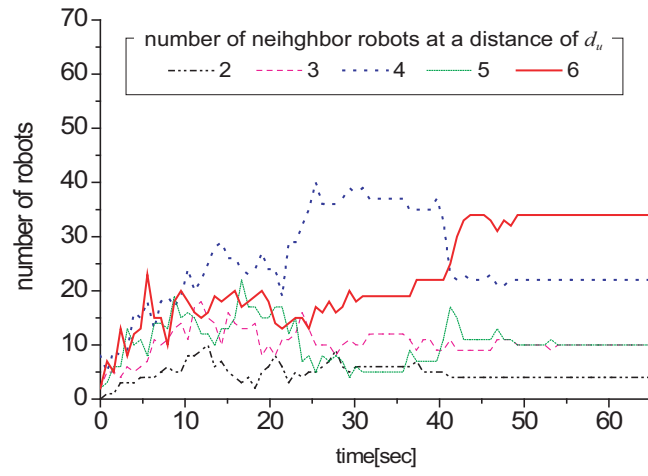


図 5.5: 複雑な未知環境下での Adaptive Flocking のシミュレーション結果

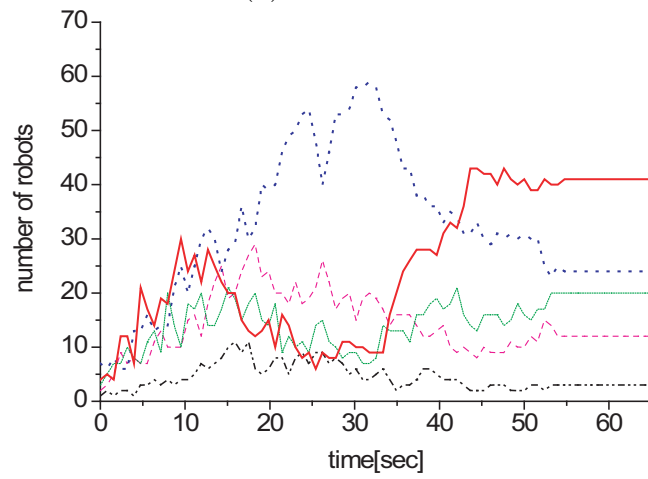
5.3 提案手法の検証

5.3.1 アルゴリズムの拡張性

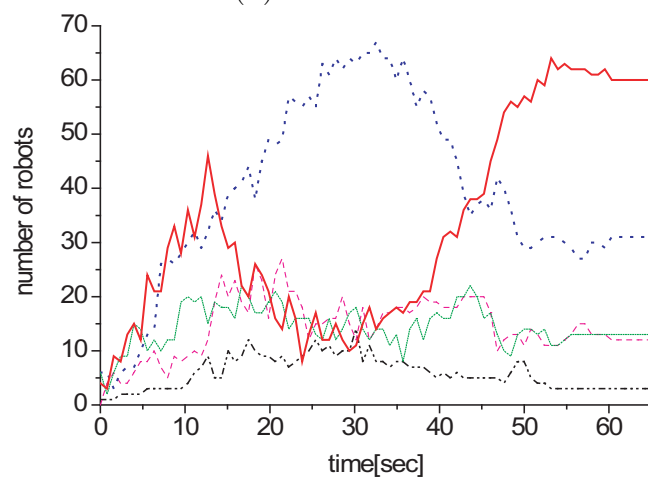
図 5.6 は図 5.5 の結果を基に、4 章で定義した接続性を時間推移に対しての変化を表した。(a), (b), (c) はそれぞれ 80, 100, 120 台で行ったシミュレーション結果である。この図で図 5.5 におけるロボット群の振る舞いが三区間（第一区間:0 ~ 15[sec] , 第二区間:15 ~ 45[sec] , 第三区間:45 ~ 60[sec]）に分けて考えられる。第一区間では *TC* を形成しながら移動を開始することによって接続性が 6 台の値が増加した。第二区間に入ると同時に 6 台の値が減少し、逆に 4 台の値が増加したことが確認された。図 5.5 を参照すると、この区間で隘路によって Team Partition が実行されることで接続性が減少したことがわかった。第三区間では 4 台と 6 台の値が再び逆転し、4 台が減少、6 台が増加した。再び図 5.5 を参照すると、この区間で隘路通過後に Team Unification が実行されていることがわかる。さらに、台数別による時間差があるものの群れは最終的に一定の接続性を示し、維持しながら移動していることがわかる。区間によって接続性の変動が各アルゴリズムの実行により顕著に現れた。(a), (b), (c) の各結果で三区間の時間的な誤差が存在するが、Partition, Unification, Maintenance の実行によって起こった現象から台数に関係なく動作可能なアルゴリズムであることが確認できた。この結果から、提案したアルゴリズムは環境に対して適応的な移動が台数に関係なく行われていることが明確にわかった。よって、提案した Adaptive Flocking アルゴリズムは拡張性を有していると判断できる。



(a) 80 robots



(b) 100 robots



(c) 120 robots

図 5.6: 図. 5.5 のシミュレーションにおける時間変化に対する接続性の変化

5.3.2 各アルゴリズムの効果

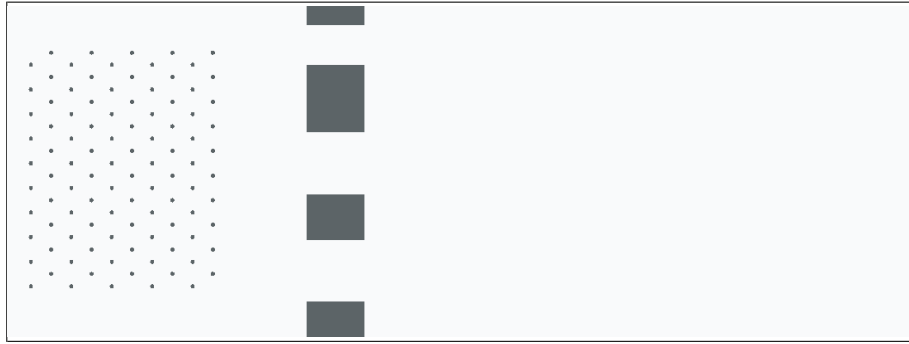
提案した Adaptive Flocking 問題は 3 つのアルゴリズムを結合することで解決することができた．本項では，提案したこれら 3 つのアルゴリズムの組み合わせを変更することでそれぞれのアルゴリズムが Adaptive Flocking に対してどのように影響を及ぼしたか分析を行なう．

はじめに，Team Partition の能力を持たない群れの振舞いを図．5.7 のシミュレーションで確認した．複数のチームに分裂できない群れは隘路を通り抜けるのにおよそ 90[sec] を要した (図．5.7 および図．4.5 参照)．図．4.5 と同条件に設定した上で要した時間はおよそ 40[sec] であり Team Partition の能力を持たせなかった図．5.7 の結果よりおよそ 1/2 に短縮できた．この結果から，Team Partition アルゴリズムは群移動に対して効率的な能力を提供することが確認できた．

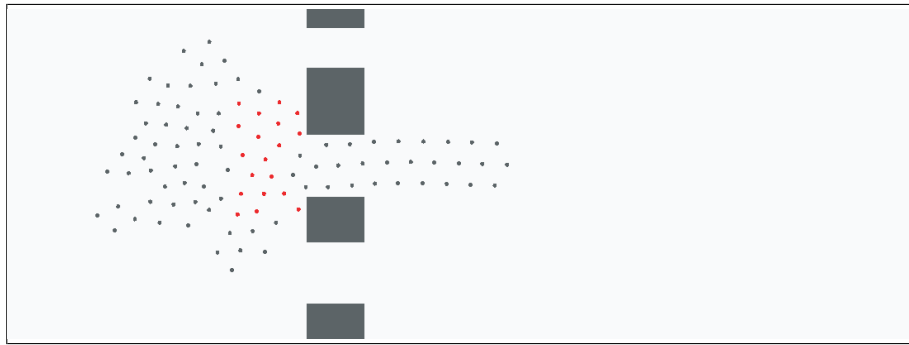
さらに，Team Unification の能力の必要性について考察する．例えば，ロボットの通信距離におよそ d_u の限界があるモバイルセンサネットワークを利用した監視活動への応用を考える．この群れがセンサネットワーク網を張り巡らせたとき，TC の形成による最大の接続性は 6 台である． d_u 内に配置している通信可能な相手ロボットが多いほど情報伝達の即効性，共有性が高いとして評価すると Team Unification 能力が群れに多大な影響を与えていることが表 5.1 は示した．これは図．5.5 の第二区間の 3 チームの分裂時を右から A,B,C チームと仮定し，3 チーム時の接続性と第 3 区間の接続性をを比較したものである．表 5.1 からわかるように分裂時は 1 割ほどしかいなかった 6 台の接続性を持ったロボットが統合後はおよそ半数のロボットが確保することができた．これは Team Unification が Adaptive Flocking に与えた利点である．また，表 5.1 より，隘路の通過という接続性の確保が困難な状況にも関わらず多くのロボットが 4 台を維持していることは Team Maintenance が有効に働いている結果である．したがって，この Team Unification は群移動の際のチームメンバーの離散防止としても重要な役割を果たすことも期待できる．

表 5.1: 図．5.2 における統合前と統合後の connectivity の変化

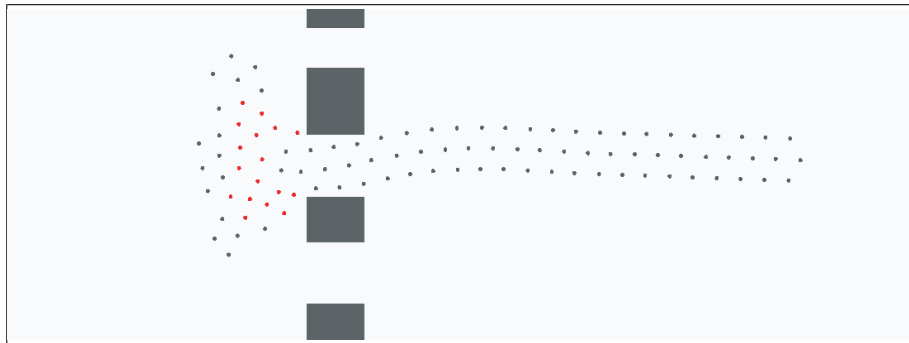
number of robots	before Team Unification			after Team Unification
	A	B	C	a re-unified team
2	2	6	4	3
3	2	11	4	13
4	19	27	21	21
5	0	11	2	13
6	0	11	0	62



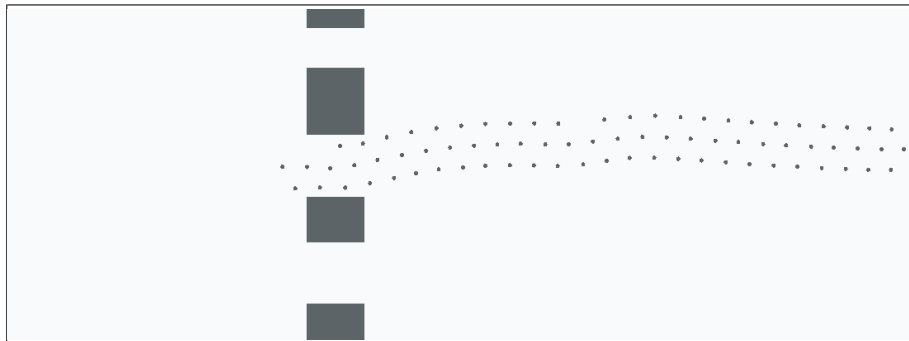
(a)0[sec]



(b)30[sec]

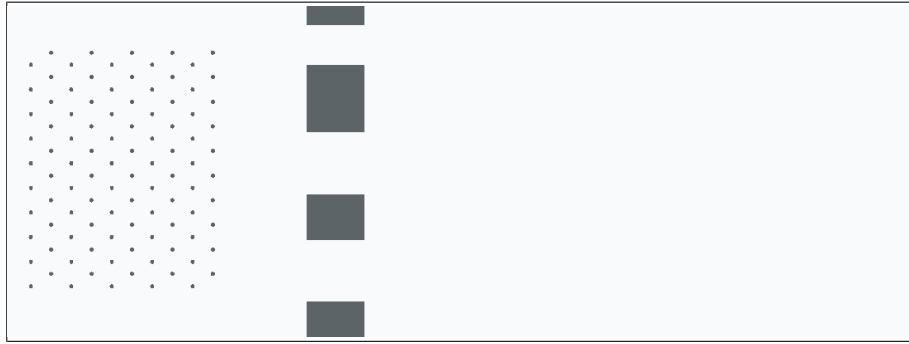


(c)60[sec]

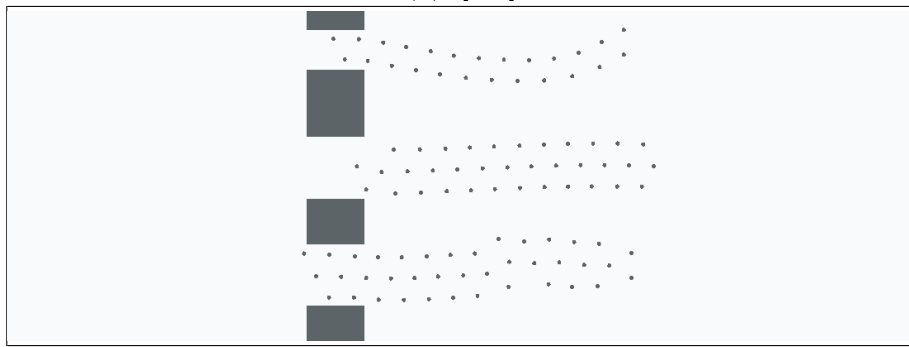


(d)90[sec]

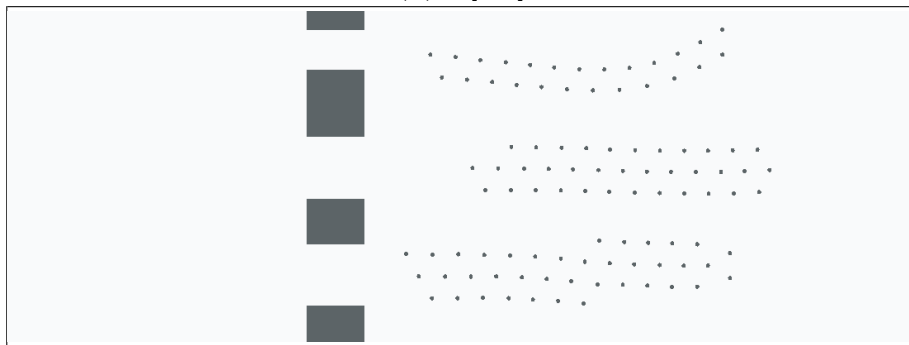
図 5.7: Team Partition 能力が欠如した Adaptive Flocking



(a) 0[sec]



(b) 40[sec]



(c) 50[sec]



(d) 60[sec]

図 5.8: Team Unification 能力が欠如した Adaptive Flocking

第6章 Tracking Multiple Moving Goals

本章は Adaptive Flocking アルゴリズムの応用として複数の移動するゴールの追従群移動について解説する．群ロボットの応用において複数のするゴールを追従しながら群移動を行うことが必要とされることが考えられる．例えば，複数個所の地点から化学物質が漏洩したときに，漏洩規模に適したチーム配分を行いながら漏洩地点を特定し，現場でチームとして対処に当たらなければならない．あるいは，監視活動中の群れが複数の侵入者を発見したとき，書く侵入者に対しチームを構成し，追従することが必要である．よって，複数の移動するゴールに対して群れを分裂させメンバーを配分することでゴールに対してかつ環境に対して適応した群移動を行う手法を提案する．本章では Team Maintenance と環境に対する Team Partition およびゴールに対する Team Partition の三つで目的を達成する．よって，本章ではゴールに対する Team Partition アルゴリズムを 4 章で記述した手法を拡張することで提案する．

6.1 アルゴリズム

6.1.1 複数のゴールに対する Team Partition

複数の移動するゴールの追従方法は，複数のゴールから追従するゴールの判断から始まる．自分がどのゴールを追従するかを選択しなければならないからである．複数のゴールに対する判断は 4 章で提案した Team Partition と同様で万有引力の法則を用いる．ここで，混乱を避けるために環境に対する Team Partition による *favorite vector* $\vec{f}$ を本節では sf_j として改めて定義をしておす．これは環境に対する分裂判断のベクトルとゴールに対する分裂判断の区別を明確にするための措置であり，アルゴリズムに対して影響を与えるものではない．複数のゴールを発見したロボット r_i について考える．このとき， d_k を r_i からゴール g_k までの距離， m_k を g_k の優先度や，必要労働力などを表す値とすると，ゴール g_k に対するロボットの $|^gf_k|$ は

$$|^gf_k| = m_k/d_k^2 \quad (6.1)$$

と定義する．複数のゴールに対する分裂の手順は 4 章の Team Partition と同様である． r_i は発見した全ゴールに対して $|^gf_k|$ を算出する．その中で最大となるベクトルを $|^gf_{\max}|$

とする．ここで算出されたベクトルはこれまではひとつだけしか与えられていなかったゴールに対応する $\vec{G}$ にとってかわる．つまり，ゴールを選択した後に実行される Team Maintenance でのゴールエリアの設定は $A(\vec{g}_{f_{\max}})$ となり，これにより，複数のゴールに対して分裂が発生する．

6.1.2 $\vec{g}_{f_{\max}}$ と $\vec{s}_{f_{\max}}$ の両立

前節では複数のゴールに対しての分裂を説明した．次に問題となることが障害物が存在している環境内での複数の移動するゴールの追従である．ここで，判断した $\vec{g}_{f_{\max}}$ と相反するような $\vec{s}_{f_{\max}}$ を算出しかねないという問題が存在する．つまり，環境への適応とゴール追従のジレンマである．例えば，図．6.1-(a) において r_i はゴール g_1 を選択した．次にゴールに対する *favorite vector* を考慮せずに環境に対する分裂を行った場合，図．6.1-(b) における広い隘路 s_2 を選択する恐れがある．この場合，ロボットはゴールに対し遠回りをしてしまうために追従性が劣ってしまうことが考えられる．環境に適応したい一方でゴール追従のために最短経路を経由したいという考えが自然に発生する．本稿ではこの問題を解決する環境へ適応した移動するゴール追従の両立を実現するために二つのパラメータを導入する．二つの異なる判断指標をひとつの指標として捉えるためにパラメータにより補正した二つのベクトルの総和が最大となる隘路を選択させるのである．これは，先に判断したゴール方向のベクトルを用いて，隘路を判断する方法である．概要は以下の通りである．

Step1. $\max_{g_{j_2} \in G} [|\vec{g}_{f_{j_2}}|]$ を選択．

Step2. $\max_{s_j \in S} [E_k \times \vec{s}_{f_j} + G_k \times \vec{g}_{f_{\max}}]$ となる隘路 s_j を判断し，進行する．

ここで， E_k は環境係数， G_k はゴール係数としてそれぞれのベクトルをパラメータである．

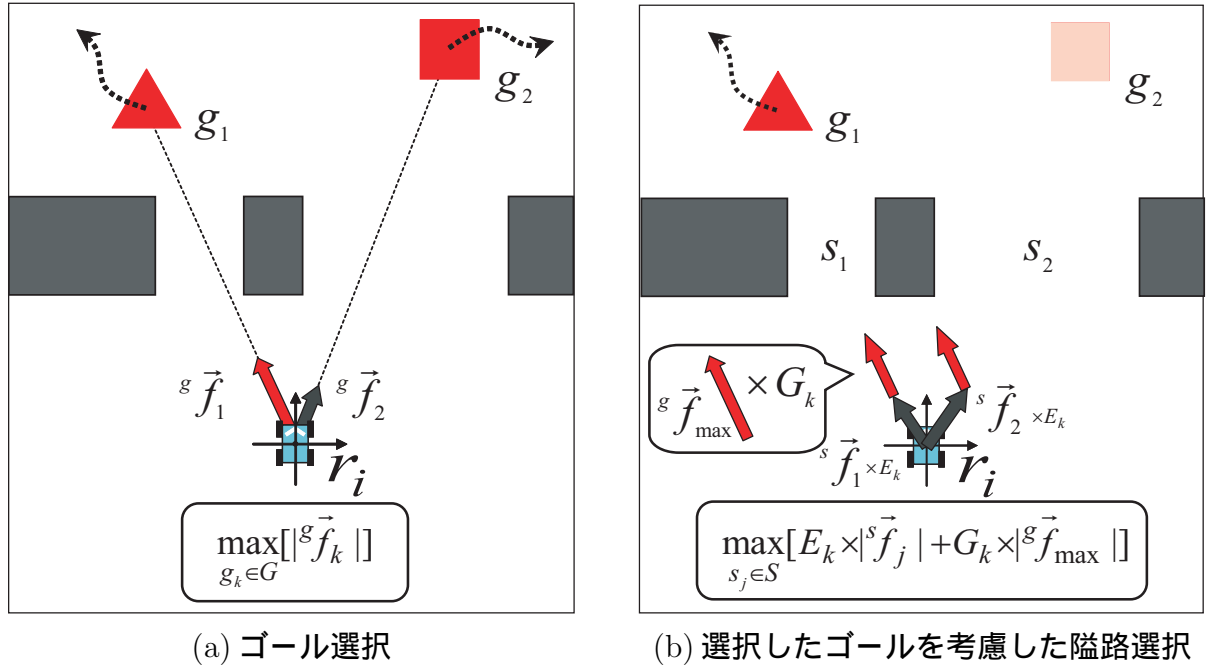


図 6.1: 複数のゴールからの選択とそれによる複数の隘路からの選択

6.2 シミュレーション

はじめに，シミュレーション条件としてパラメータを $E_k = 0.1$ ， $G_k = 1.0$ と設定し，シミュレーションを行った．はじめは移動する単体のゴールに向かう Adaptive Flocking の結果を図. 6.2 に示した．シミュレーション条件は図. 5.5 と同じで，移動するゴールはランダムな軌道を与え，ロボットに比べ大きな正方形として表示した．このとき，どのロボットにも移動するゴールの軌道を事前に与えなかった．ゴールを動かすことで群れは追従を開始した．追従の際に群れが障害物に遭遇すると，各ロボットは TC を維持しながら複数のチームに分かれ，通過後は 1 つに統合した．図. 5.5 と図. 6.2 の (e) の分裂を比較するとロボットの軌道が異なっていることがわかる．ゴールが移動することによって $\vec{G}$ が刻々と変化するために，ロボットの軌道も異なる結果を示した．しかし，この変化する条件においても群れは環境に適応しながらゴールまで達することができた．

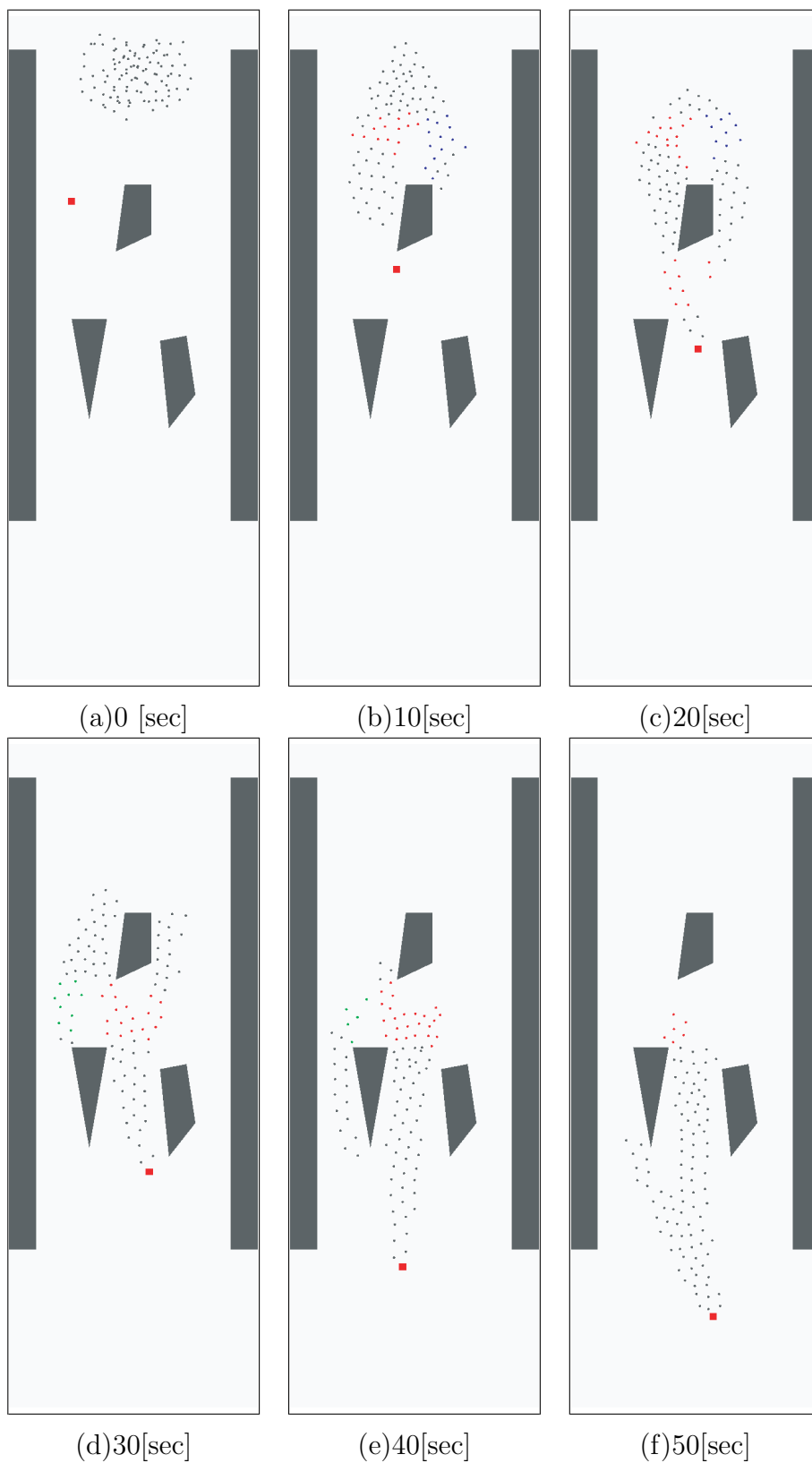


図 6.2: 複雑な未知環境下を移動する単体のゴールの追従のシミュレーション結果

次に障害物環境内を移動する複数のゴールの追従を行った．図．6.3 に示すような環境で移動するゴールを二つ設定し，それぞれを三角形，正方形として描いている．移動するゴールは初期状態で同一地点に配置し，右側に浮遊するように軌道を与えた．この時，ロボットに軌道情報を与えることは一切行わなかった．ゴールの軌道は障害物をすり抜けるように描き，途中二つを交差させ，最終的にそれぞれ右側に静止させた．これまでと同様に初期状態において 100 台のロボットをランダムに配置された群れは二つの移動するゴールを選択し追従を行った．ランダムに配置していた群れは図．6.3-(a) から (b) にかけて TC を形成しながら少しずつチームを組織した．また，それぞれの移動するゴールに対応したチームを自律的に構成していった．図．6.3-(c) から (d) にかけて下側のゴールを追従していた群れの一部はより狭い中央の隘路を選択し，ゴールの追従を行った．また，図．6.3-(f) から (h) においては下側，上側のそれぞれチームで同様の追従形態を示した．これは前節で提案した複数のゴールと複数の隘路に対する分裂判断を行った結果である．ゴールに対する *favorite vector* を考慮せずに環境に対する分裂を行った場合，群れは広い隘路を求めて障害物下側の進行を希望する．この場合，ロボットは移動するゴールに対し遠回りをしてしまうためにゴールへの追従性が劣ってしまうと考えられる．これに対し，シミュレーション結果は狭い通路も追従のために積極的に通過を行った．従って，前節で提案した手法により，ゴールの追従性と環境への適応を両立させることができた．さらに，図．6.3-(d) から (f) にかけて群れは交差したゴールに対してチームを自律的に再編成した．この結果は，提案手法ではゴールに対して固定したチームを作るのではなく，その時その場に応じたチームを自律的に構成する手法を取っていることから起こった事象である．このとき，各ゴールに対して追従するロボットが決定されていた場合，群れはそれぞれの移動するゴールを目指すために衝突を回避するための複雑な交渉や制御が付加的に必要なってしまう．また，これにより移動するゴールを見失ってしまう危険性も考えられる．この結果から，提案手法は追従に適応した分裂を可能にしていることもまた含有している．ゆえに，群ロボットは前節で提案した手法により障害物が存在する環境に対して適応的に群移動を行いながら移動する複数のゴールを追従することが可能である．



図 6.3: 障害物が存在する環境下を移動する二つのゴールの追従

第7章 まとめ

7.1 研究のまとめ

本稿は群ロボットにおける問題として未知環境への適応的群移動について問題定義を行った．Adaptive Flocking 問題として定義した問題に対して魚の群泳を模倣対象に 3 つの sub-problem に分解し，各問題に取り組んだ．sub-problem-1.Team Maintenance，sub-problem-2.Team Partition，sub-problem-3.Team Unification に対してそれぞれアルゴリズムを提案した．提案したそれぞれ 3 つのアルゴリズムを結合することで群ロボットが移動しながら未知作業環境に適応したチーム分裂，あるいはチーム統合ができる Adaptive Flocking アルゴリズムを提案した．提案した手法はシミュレーションにより検証し，さまざまな分析を行った．Adaptive Flocking をさらに拡張することで環境内を移動する複数のゴールの追従を可能としたことを通して提案手法の様々な応用の可能性を示した．Adaptive Flocking は群ロボットを実世界へ適用するため第一歩である．例えば，体内の薬物送達や探索活動，あるいは，巡視活動などに応用が可能である．

7.2 研究の利点

本論文では，未知環境に適応しながらゴールまで群移動を行う Adaptive Flocking 問題を定義した．この問題ではロボットは 1) anonymous，2) 特定のリーダーロボットの不在，3) 共通座標なし，4) ロボット間での通信なし，5) 作業環境は未知，6) メモリなし (oblivious) といった単純なモデルに基づいて問題を分析した．提案した Adaptive Flocking アルゴリズムで全ロボットは現在のセンシング情報のみを用いてするゴール位置を計算した．これにより群ロボットは TC を維持しながら柔軟に環境に適応することが可能となった．また，提案手法はシミュレーションにより有効性を示した．各分析とシミュレーション結果より，Adaptive Flocking は変動する環境においても適応的に群移動が可能な単純かつ効果的なアプローチである．さらに，定義した Adaptive Flocking 問題の解決手法は分散制御，自己組織化，自己安定化，決定論的手法の特徴が明らかになった．提案した Adaptive Flocking は TC を維持しながら群移動を行うことができるため，群ロボットを用いた mobile sensor networking の応用も期待される．

7.3 今後の展望

本研究の主な課題として二点が考えられる．

- センシングやモーターなどのエラーを考慮した制御：本研究はシミュレーションにおいてセンサ情報にエラーを組み込むことは一切行わなかった．他ロボットの位置を完全に取得できるセンサ，また，計算により算出された移動目標地点には誤差なく必ず到達することを仮定した．今後は提案手法の耐エラー性の検証，あるいはエラーに対する耐性を持ったアルゴリズムの提案が必要である．
- 2D から 3D への拡張：近年，提案されている群ロボットの制御手法は 3 次元空間に対応できるものが増加傾向にある．これは水中で活動する群ロボットや空中を編隊飛行するロボットなどに応用可能である．本研究で提案した LI は正三角形を形成することで二次元空間上に秀逸な格子構造的フォーメーションを形成できた．これを三次元に拡張することで正四面体の格子構造フォーメーションの形成の可能性を含んでいるアプローチであると考えられる．よって，提案手法を 3 次元空間に拡張することが本研究の第二の課題であると考ええる．

謝辞

本研究を進めるにあたり，ご懇篤なご指導とご高配を賜りました丁洛榮助教授に心から感謝の意を表します．そしてロボティクス講座の皆様に心よりお礼申し上げます．

ポストドクターの金亨旭さん，短い期間でしたが本当にありがとうございました．

細部にわたり熱心に御助言を賜りました李根浩さんをはじめとする SWARM ROBOT チームのメンバの清水大輔さん，尹錫勲さん，本当にありがとうございました．

英語ができない私のために日本語で話しかけてくれた金明植さん，梁佑誠さん，柳在官さん，TRUONG Manh Cuong さん，本当にありがとうございました．

2年間の苦楽を共にした竹内常哲さん，久保貴史さん，南孝さん，村井真也さん，本当に感謝いたします．

頼りない先輩である私を頼りにしてくれた鵜飼真広さん，須永聡さん，瀬尾若葉，村上享平さん，本当にありがとうございました．

参考文献

- [1] E. Sahin, Swarm robotics: from sources of inspiration to domains of application, Swarm Robotics, SAB 2004 International Workshop, E. Sahin and W. M. Spears (Eds.), Lecture Notes in Computer Science 3342, Springer, 2005.
- [2] バート・ヘルドブラー , エドワード・O・ウィルソン [著], 辻和希 , 松本忠夫 [訳] , 蟻の自然誌, 朝日新聞社, 1997.
- [3] A. Yamashita, T. Arai, J. Ota, and H. Asama, Motion planning of multiple mobile robots for cooperative manipulation and transportation, IEEE Transactions on Robotics and Automation, Vol. 19, No. 2, pp1–15, 2003.
- [4] J. Seyfried, M. Szymanski, N. Bender, R. Estana, M. Thiel, and H. Worn, The I-SWARM project: intelligent small world autonomous robots for micro-manipulation, Swarm Robotics, SAB 2004 International Workshop, E. Sahin and W. M. Spear (Eds.), Lecture Notes in Computer Science 3342, Springer, 2005.
- [5] M. Lam and Y. Liu, ISOGIRD: an efficient algorithm for coverage enhancement in mobile sensor networks Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006.
- [6] M. Long, A. Gage, R. Murphy, and K. Valavanis, Application of the distributed field robot architecture to a simulated demining task, Proc. of IEEE International Conference on Robotics and Automation, 2005.
- [7] C. W. Reynolds, Flocks, herds, and schools: a distributed behavioral model Computer Graphics, Vol. 21, No. 4, pp25–34, 1987.
- [8] 倉林大輔 , 長川研太 , 幾何条件による自律移動ロボット群の編隊構造遷移 , 日本ロボット学会誌 , Vol. 23, No. 6, pp78–84, 2005.
- [9] L. E. Parker, B. Kannan, F. Tang, and M. Bailey, Tightly-coupled navigation assistance in heterogeneous multi-robot teams, Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004.

- [10] Geunho Lee, Yuki Komatsu, Shunsuke Sone, and Nak Young Chong, Formation navigation of a team of mobile robots adapting an environment, Proc. of 36th International Symposium on Robotics, 2005.
- [11] Geunho Lee, Yosuke Hanada, and Nak Young Chong, Decentralized formation control for small-scale mobile robot teams, Proc. of Robotics and Mechatronics Conference, 2006.
- [12] J. Fredslund and M. J. Mataric, A general algorithm for robot formations using local sensing and minimal communication, IEEE Transactions on Robotics and Automation, Special issue on Advances in Multi-Robot Systems, Vol. 18 No.5, pp837–846, 2002.
- [13] T. Balch and R. C. Arkin, Behavior-based formation control for multi-robot teams, IEEE Transactions on Robotics and Automation, Vol. 14, No. 6, pp926–939, 1998.
- [14] T. Balch and M. Hybinette, Social potentials for scalable multi-robot formations, Proc. of IEEE International Conference on Robotics and Automation, 2000.
- [15] W. Spears, D. Spears, J. Hamann, and R. Heil, Distributed, physics-based control of swarms of vehicles, Autonomous Robots, Vol. 17, No.2-3, pp137–162, 2004.
- [16] D. Zarzhitsky, D. F. Spears, and W. M. Spears, Distributed robotics approach to chemical plume tracing, Proc. of IEEE/RSJ International Conference on Intelligent Robotics and Systems, 2005.
- [17] Guiha Ye, H. O. Wang, K. Tanaka, and Z. Guan, Managing group behaviors in swarm systems by associations, Proc. of American Control Conference, 2006.
- [18] 清水正宏, 高橋正康, 川勝年洋, 石黒章夫, 制御系と機構系の有機的カップリングを活用したモジュラーロボットの創発的形態制御, 日本ロボット学会誌, Vol. 24 No. 7, pp107–114, 2005.
- [19] G. Folino and G. Spezzano, An adaptive flocking algorithm for spatial clustering, Proc. 7th International Conference on Parallel Problem Solving from Nature, 2002.
- [20] J. Nembrini, A. Winfield, and C. Melhuish, Minimalist coherent swarming of wireless networked autonomous mobile robots, Proc. of Simulation of Artificial Behavior, 2002.
- [21] J. M. Esposito and T. W. Dunbar, Maintaining wireless connectivity constraints for swarms in the presence of obstacles, Proc. of IEEE International Conference on Robotics and Automation, 2006.
- [22] E. O. Wilson, Sociobiology: The new synthesis, Harvard University Press, 1976

- [23] S. Stocker, Models for tuna school formation, *Mathematical Biosciences*, Vol. 156, pp167–190, 1999.
- [24] S. Dolev, *Self-Stabilization*, MIT Press, 2000.
- [25] I. Suzuki and M. Yamashita, Distributed anonymous mobile robot: Formation of geometric patterns, *SIAM Journal of Computing*, Vol. 28 No. 4, pp1347–1363, 1999.
- [26] A. Johnson, *Famous Problems and Their Mathematicians*, Teacher Ideas Press, 1999.
- [27] D. Halliday, R. Resnick, and J. Walker, *Fundamentals of Physics*, Wiley, 5th Ed. 1997.