アドホックネットワークを形成する群ロボットの優先度付き分散 Receding Horizon Control

村 山 暢*

Distributed Receding Horizon Control with Priority for Multi Robot System Including Ad-hoc Network

Toru Murayama*

This paper presents a multi-robot control method considering a mobile ad-hoc network connectivity and inter-robot collision avoidance. A method I studied previously was based on a receding horizon control (RHC) method, and was decentralized one by some kind of serialization. However, the previous method remains some disadvantages; especially, no-scalability and huge computations. Because these disadvantages are critical flaws for multi-vehicle systems, I strive to improve the method in this paper. By a distributed priority variable, several vehicles (but not all vehicles) can make their decisions at same time in the proposed method. We show the proposed method in this paper is less computation than the previous method.

Key Words: Multi Robot System, Mobile Ad-hoc Network, Receding Horizon Control, Network Topology Control

1. は じ め に

群ロボットシステムは広範な作業空間で効率よく作業を行うことや1台のロボットのみでは不可能なタスクを行うことを目標に研究開発が進められている[1][2]. 想定される主な作業としては協調的な監視や観測, 把持や運搬などが挙げられる. このような協調作業を自律分散的に行うために, 各ロボットは他のロボットと通信, 観測, 接触といった何らかの相互作用を行う必要がある.

群ロボット間での通信については, 多くの場合, 移動上の制約の少ない無線通信の利用が想定される. 特に近年では, モバイルアドホックネットワーク(MANET)のような近傍のロボット同士でメッシュネットワークを構成する手法が研究されている. アドホックネットワークを用いた多数のロボットからの映像通信や, 多数のロボットの遠隔制御も実現されている[3].

これらの無線通信では, 無線の送信電力や周波数帯域の干渉などの制限により, 送受信が可能なロボット間距離に上限があると考えられる. またその一方, ロボット間の距離が近くなりすぎることによる衝突も避けるべきである. したがって群ロボットの協調移動制御を考える際には, これらの移動可能範囲制約

を考慮することが重要である.

以上のことから, 群ロボットシステムにおけるロボット間のネットワーク連結性維持を主目的とした移動制御手法の研究が行われている. 特に, 2台以上のロボットが同時に任意のネットワークリンクを切断する場合にネットワークの連結性が崩れる恐れがある, という固有の課題があるため, これを解決する手法が重要となる. Mengらは群ロボットを目標フォーメーションへ移動させる問題に対し, 一旦全ロボットを一箇所へランデブーさせ, ネットワーク構造を完全グラフとした後, 目標のグラフ構造へ変化させて目標フォーメーションへ移動させるという手法を提案している[4]. これは局所的な情報のみで制御できる分散的な手法ではあるが, ロボット同士の衝突を考慮しておらず, ロボットの体積が大きく通信可能範囲が小さい場合には完全グラフを経由することができないという問題がある. Zavlanosらは複数のロボットで同じ切断リンクを選択できるような分散オークションアルゴリズムを提案している[5]. これは各ロボットが切断したいリンクを一つ候補に上げ, 全ロボットの切断要求リストが集まったところでオークションにより切断を決定する手法である. 移動制御についてはポテンシャル法を用いており, 目標フォーメーションへの収束性は保証されていない.

また, ロボット間ネットワークでは群ロボットシステムに行わせるタスクに応じて様々な情報が送受信されると考えられるため, 通信量や通信回数は少ないほうが望ましい. 例えば, Fig. 1のように群ロボットシステムによって操作者から遠隔操作ロボットまでの通信を補助・維持する場合, 操作者-遠隔操作ロボット

原稿受付 2014年12月26日

*和歌山工業高等専門学校知能機械工学科

*Dept. of Intelligent Mechanical Engineering, National Institute of Technology, Wakayama College.

■ 本論文は新規性(要素分野)で評価されました.

Fig.1 Application: autonomous robots preserve network connection from operator to teleoperated robot

間の通信が主であり、群ロボット間での情報のやりとりは少ないほうが良いと考えられる。特にここではマルチホップネットワークを想定しているため、大量の転送データによって自身の通信を行えない際も、ネットワーク全体の連結性を保つことが重要である。

筆者らはこの問題に対して、制御入力の上限や速度制約といった移動に関する制約や、ネットワークの連結性や通信できるロボット数（グラフの頂点次数）の制約など多様な制約を考慮できるような、Receding Horizon Control (RHC) をベースとした制御手法を提案してきた [6] [7]。RHC は一般に有限時間最適制御問題を解くことによって未来を予測しながら制御入力をオンラインで決定する手法であり、最適制御問題としてネットワークの連結性制約を導入することで連結性維持制御を達成している。また、最適制御の目的関数を適切に設計することにより、目標フォーメーションへの収束性も保証している [6]。さらに、根らの提案したロボット間の衝突回避制約を含む RHC 手法 [8] を導入し、衝突回避とネットワークの連結性維持を同時に達成している [7]。RHC はその時刻における制御入力だけではなく有限区間長先までの制御入力列を決定するため、通信に消失がある場合の制御手法としても有効である [9] との提案もあり、本研究のようなロボット間通信においても有効であると考えられる。

手法の分散化に関して、各ロボットが同時に独立して意思決定を行うと、制約を満たさない制御入力列を生成する恐れがある。特に衝突回避や連結性維持といったロボット間距離にかかわる複数台のロボットに跨る制約条件を含む問題は、独立した部分問題に分解できないことから、過去に求めた予測制御入力列を通信によって知らせること、現在計算する予測制御入力列を過去のもの大きく変化させない、といった手法によって制約を満たした解を得る分散 RHC が提案されている [10]。

先に示した筆者らの提案手法 [6] [7] では、各時刻にロボット 1 台ずつ順に制御入力列を求めることによって制約の充足性を保っている。このため、ロボット数の増加に伴い制御入力列計算の機会が減り、ロボットが制御入力列を計算できるまで停止する時間が多くなってしまふ。また、各ロボットは通信により得たシステム全体の情報を利用した最適制御問題により、制約を満たす制御入力列を得ている。このため、考慮すべき制約の数がロボット数に比例して多くなり、最適制御問題の求解に時間がかかることが課題となっている。本研究ではこれらの課題を解決した、改善制御手法について提案する。

本手法の特徴は次のとおりである。まず、リンク切断の対象と衝突回避の対象を局所的な物に限定した。これによって、最適制御問題の変数と制約の数は、全ロボット数から局所的な近

傍ロボットの数へと減少した。また、衝突の恐れがあるロボット同士が同時に制御入力列を決定しないよう、隣接ロボット同士で値の異なる優先度変数を導入した。これによって、各ロボットは局所的な情報で最適制御問題を解くことが可能となり、制御入力列計算の機会も増加した。

本論文の構成は次のとおりである。まず 2 章では本研究が対象とする群ロボットシステムの定式化を行う。3 章で提案手法の詳細について述べ、4 章にて手法の有用性を示す。4 章では計算機シミュレーションにより連結性維持、衝突回避、フォーメーション制御が行えているかを確認すると共に、先行研究との比較を行う。5 章では本論文のまとめと、今後の課題を述べる。

2. 対象とする群ロボットシステム

本研究では、 N 台で構成される $d \in \{2, 3\}$ 次元空間を移動する群ロボットシステムを考える。各ロボットはそれぞれラベル $i \in \{1, 2, \dots, N\}$ を持ち、 i 番目のロボットをロボット i と呼ぶ。ロボット i の中心位置座標を $q_i \in \mathbb{R}^d$ とし、位置 q_i を含むロボット i の状態を $x_i \in \mathcal{X}_i$ と定義する。例えば、三次元空間を移動するロボットの状態として $x_i = (q_i^x, q_i^y, q_i^z, \theta_i, \phi_i, \psi_i, \dot{q}_i^x, \dot{q}_i^y, \dot{q}_i^z, \dot{\theta}_i, \dot{\phi}_i, \dot{\psi}_i)^T$ をとる場合などが考えられる。ここで θ_i, ϕ_i, ψ_i はそれぞれロール角、ピッチ角、ヨー角であり、 $\dot{q}_i^x, \dot{\theta}_i$ などはそれぞれ速度、角速度である。また、姿勢を考慮しない二次元平面を移動するロボットの場合、状態を中心位置座標のみ $x_i = (q_i^x, q_i^y)^T$ と考えることも可能である。

ロボットは次の式で表されるような離散時間 $t \in \mathbb{Z}$ でのダイナミクスを持つとする。

$$x_i(t+1) = f_i(x_i(t), u_i(t)), \quad (1)$$

ここで $u_i \in \mathcal{U}_i$ はロボット i への制御入力を意味する。本研究では、集合 $\mathcal{X}_i^f = \{x_i^f | \text{s.t. } x_i^f = f_i(x_i^f, u), \exists u \in \mathcal{U}_i\}$ が存在すると仮定する。これはロボットの能力として状態 $x_i^f \in \mathcal{X}_i^f$ にとどまり続けることができることを想定している。ロボット群が並進移動を行う隊列制御などを行いたい場合も、状態 x_i や制御入力 u_i の座標変換によって対応可能である。

ロボット i は d_i の車体半径を持つとし、ロボット i とロボット j の間の距離が $d_i + d_j$ 以下となる場合、ロボット i とロボット j は衝突しているとする。このとき、次の条件をすべての時刻 t で満たす場合、ロボット i はロボット間の衝突を回避できていると考えられる。

$$\|q_i(t) - q_j(t)\| > d_i + d_j, \quad \forall j \in \{1, 2, \dots, N\} \setminus \{i\}. \quad (2)$$

ここで $\|\cdot\|$ はユークリッドノルムを意味する。

また、各ロボットは無線通信により他のロボットに情報を送信でき、他のロボットから情報を受信できるとする。ロボット i は通信半径 r を持ち、ロボット間距離が r 以下のすべてのロボットに対して情報の送受信ができると考える。つまり、ロボット i は時刻 t において

$$\|q_i(t) - q_j(t)\| \leq r, \quad (3)$$

となるすべてのロボット $j \in \{1, 2, \dots, N\} \setminus \{i\}$ へ情報を送る

Fig. 2 System architecture of robots

ことができるとする。本研究では上式が成り立つとき、ロボット i から j への通信リンク l_{ij} が存在すると考え、このときの通信ネットワーク全体の構造をグラフ $\mathcal{G}(t) = \mathcal{G}(\mathcal{V}, \mathcal{E}(t))$ で表す。ここで、 $\mathcal{V} = \{1, 2, \dots, N\}$ はグラフの頂点集合（ロボットのラベル集合）、 $\mathcal{E}(t)$ はグラフのリンク集合であり、条件式 (3) を満たすとき $l_{ij} \in \mathcal{E}(t)$ である。また、ロボット i がグラフ上で隣接するロボットの集合を $\mathcal{N}_i(t) = \{j \mid \text{s.t. } \|q_i(t) - q_j(t)\| \leq r\}$ で表す。本研究におけるロボットは他のロボットを観測することを想定しておらず、隣接ロボットからの通信によってのみ他のロボットの情報を得ることができるとする。

各ロボットは Fig. 2 のように通信制御系と移動制御系から成っているとする。通信制御系はマルチホップ無線通信にかかわる送受信制御やルーティングなどを行う部分であり、移動制御系はロボットの移動に関する各種アクチュエータ・センサ（ジャイロセンサやエンコーダなど）による計測制御や自己位置推定などを行う部分である。本研究では、通信制御系モジュールからルーティングに用いるためのネットワーク構造情報 $\hat{\mathcal{G}}$ を得ることができるとする。もしルーティング情報などからネットワーク構造を得ることができない場合は、3 章で定義する *message* にネットワーク構造情報を含ませて利用する。これらの両モジュールから得られる情報を基にして、ネットワーク構造 $\mathcal{G}(t)$ が常に連結であるように各ロボットを目標位置まで移動させるための分散制御手法を提案することが本研究の目的である。

以上のシステムに対して、各ロボットを任意の初期状態 $x_i(t_0)$ から目標状態 x_i^d へ制御するための、衝突回避制約 (2) およびネットワーク連結制約を満たすような制御入力 u_i を求めることが本研究の制御目標である。ここで t_0 は初期時刻であり、初期時刻 t_0 において初期状態 $x_i(t_0)$ は衝突回避条件を満たしネットワーク $\mathcal{G}(t_0)$ は連結であると仮定する。同様に目標状態においても、各ロボットは衝突回避条件を満たしネットワーク構造が連結であるとする。本論文では表記を簡単にするため、目標状態 x_i^d は制御入力 $u_i = 0$ に対して安定 ($x_i^d = f_i(x_i^d, 0)$) であるとする。また、本研究ではロボットの制御入力を決定する手法を議論するが、先行研究 [7] のように軌道生成手法として扱うことも可能である。

3. 提案制御手法

本研究では、各ロボット i は現在時刻 t よりも以前の時刻 $t_i^{\text{last}} \leq t$ に未来の予測制御入力列

$$\hat{u}_i^*(\cdot; t_i^{\text{last}}) = \{\hat{u}_i^*(t_i^{\text{last}}; t_i^{\text{last}}), \hat{u}_i^*(t_i^{\text{last}} + 1; t_i^{\text{last}}), \dots\}$$

をあらかじめ計算し保存しているものとする。ここで、 $\hat{u}_i^*(t; t_i^{\text{last}})$

Algorithm 1 Compute priority value

Input: $c_i(t-1), c_{j \in \mathcal{N}_i(t)}(t-1)$

```

1: if  $c_i(t-1) = c_j(t-1), \exists j \in \mathcal{N}_i(t)$  then
2:    $c_i(t-1) \leftarrow \max_{j \in \mathcal{N}_i(t)} c_j(t-1) + \text{random}$ 
3: end if
4: if  $c_i(t-1) = 0$  then
5:    $c_i(t-1) \leftarrow \max_{j \in \mathcal{N}_i(t)} c_j(t-1) + 1$ 
6: end if
7: if  $c_i(t-1) < c_j(t-1), \forall j \in \mathcal{N}_i(t)$  then
8:    $c_i(t) \leftarrow 0$ 
9: else
10:   $c_i(t) \leftarrow \max_{\{j \in \mathcal{N}_i(t), \text{s.t. } c_j(t-1) < c_i(t-1)\}} \{c_j(t-1)\}$ 
11: end if

```

Algorithm 2 Send network topology control authority

Input: *message* = {*list*}

```

1: ロボットのラベル  $i$  を list の最後尾へ移動させる
2: グラフ  $\hat{\mathcal{G}}$  上での  $i$  から list の先頭への最短経路 path を求める
3: message を path における  $i$  の次の頂点へ送る

```

は時刻 t_i^{last} に計算した時刻 t における予測制御入力を表し、 $\hat{u}_i^*(\cdot; t_i^{\text{last}})$ は時刻 t_i^{last} に計算した予測制御入力列全体を表す。この予測制御入力列に従って現在時刻 t での制御入力 $u_i(t)$ を $u_i(t) = \hat{u}_i^*(t; t_i^{\text{last}})$ と決定する。この予測制御入力列の計算方法および更新時刻 t_i^{last} の決定方法が本研究の主な内容となる。以下、提案手法の具体的な内容を記述する。

隣接するロボット同士が同時に予測制御入力を計算する場合、衝突回避などの制約に相反する制御入力を決定してしまう恐れがある。これを回避するため、本研究では各ロボットに優先度変数 $c_i \geq 0$ を与え、各時刻 t において $c_i(t) = 0$ となったロボット i のみが制御入力列 $\{\hat{u}_i^*(\cdot; t)\}$ を計算する。変数 c_i は Algorithm 1 に従って各ステップごとに値を更新する。ロボット i は隣接する全ロボット $\forall j \in \mathcal{N}_i$ と変数 c_i の値を比較し、隣接するロボット j の値 c_j と一致した場合、乱数を用いてリセットするように設定している (2~4 行め)。隣接するロボット j の値 c_j よりも自身の変数値 c_i が最も小さかった場合は $c_i = 0$ とする (8~9 行め)。

Algorithm 1 においてロボット i は、隣接ロボットと c_i の値を比較し、 c_i の値が最も小さかった場合に制御入力列の決定機会を得る。これによって隣接ロボット数が少ないロボットが制御入力列の決定機会を多く得ることとなり、各時刻においてなるべく多くのロボットが同時に制御入力列の決定を行うことを実現している。

リンクの切断についても、複数のロボットが同時に意思決定することによってネットワークが非連結になる恐れがある。本研究ではネットワーク構造の制御権 *message* をネットワークに巡回させ、各時刻 t で制御権 *message* を持っているロボットのみがネットワークの切断を決定できるものと限定する。ネットワーク構造の制御権 *message* は、過去にネットワーク構造を制御したロボット番号の逆順序 *list* を含み、Algorithm 2 に従って転送されるとする。以下に例を示す。

例 1. $N = 5$ 台の群ロボットシステムの例を考える。グラフ構造が $\mathcal{E} = \{e_{12}, e_{23}, e_{35}, e_{45}\}$ である時、ロボット 3 が *list* = 1, 2, 3, 4, 5 を含むネットワーク構造制御権 *message* を

Algorithm 3 Compute control input $u_i(t)$

Output: $u_i(t)$

```

1: if  $c_i(t) = 0$  then
2:   if ロボット  $i$  が message を保持している then
3:      $\mathcal{P}_i(x_i(t), x_{j \in \mathcal{N}_i(t)}^*(\cdot), \tilde{\mathcal{G}})$  から  $\hat{u}_i^*(\cdot; t), \hat{x}_i^*(\cdot; t), \hat{Adj}^*$  を求める
4:     message を送信する (Algorithm 3)
5:   else
6:      $\mathcal{P}_i'(x_i(t), x_{j \in \mathcal{N}_i(t)}^*(\cdot), \tilde{\mathcal{G}})$  から  $\hat{u}_i^*(\cdot; t), \hat{x}_i^*(\cdot; t)$  を求める
7:   end if
8:    $t_i^{\text{last}} \leftarrow t$ 
9:    $\hat{x}_i^*(\cdot; t_i^{\text{last}})$  を  $\forall j \in \mathcal{N}_i(t)$  へ送信する
10: end if
11: if  $\hat{u}_i^*(t; t_i^{\text{last}})$  が未定義 then
12:   return  $u_i(t) \in \{u | s.t. x_i(t) = f_i(x_i(t), u)\}$ 
13: else
14:   return  $u_i(t) = \hat{u}_i^*(t; t_i^{\text{last}})$ 
15: end if

```

受け取り、制御入力列を計算したとする。この場合、以下ののように *message* の送信先を決定する。まず、*list* のうち 3 を最後尾へ移動させ、 $list = 1, 2, 4, 5, 3$ とする (1 行め)。次に、グラフ $\mathcal{G}$ における頂点 3 から *list* の先頭すなわち頂点 1 への最短経路を求める (2 行め)。このとき、最短経路は $\{e_{23}, e_{12}\}$ となるので、経路上の次の頂点 2 へ *message* を送る (3 行め)。

このように、*list* は次に制御権 *message* を送る宛先を決定する指針として利用される。これはネットワークの半径が大きく *message* の宛先が遠い場合に、ネットワーク構造制御が長時間行われないことを防ぐ目的がある。

時刻 t において $c_i(t) = 0$ であり、かつネットワーク制御権 *message* を持つロボット i は、以下の最適化問題 $\mathcal{P}_i(x_i(t), x_{j \in \mathcal{N}_i(t)}^*(\cdot), \tilde{\mathcal{G}})$ を解くことで予測制御入力列 $\hat{u}_i^*(\cdot; t)$ を決定する。この最適化問題は変数としてネットワークリンクを含んでおり、リンクの維持/切断を決定することのできる最適制御問題となっている。

問題 1 ($\mathcal{P}_i(x_i(t), x_{j \in \mathcal{N}_i(t)}^*(\cdot), \tilde{\mathcal{G}})$).

$$\min_{\hat{x}_i(k), \hat{u}_i(k), \hat{adj}_{ij}} \sum_{k=t}^{t+T-1} \left(\|\hat{x}_i(k) - x_i^d\|_Q^2 + \|\hat{u}_i(k)\|_R^2 \right) + \|\hat{x}_i(t+T) - x_i^d\|_{Q_f}^2, \quad (4)$$

subject to

$$\hat{x}_i(k+1) = f_i(\hat{x}_i(k), \hat{u}_i(k)), \quad \hat{x}_i(t) = x_i(t), \quad (5)$$

$$\|\hat{q}_i(k) - \hat{q}_j^*(k)\| > d_i + d_j, \quad j \in \mathcal{N}_i(t), \quad (6)$$

$$\|\hat{q}_i(k) - \hat{q}_j^*(k)\| \leq r + M(1 - \hat{adj}_{ij}), \quad j \in \mathcal{N}_i(t), \quad (7)$$

$$\text{Network } \hat{Adj} \text{ is } \kappa \text{ connected}, \quad (8)$$

$$\hat{adj}_{ij} = 0, \text{ if } j \notin \mathcal{N}_i(t), \quad (9)$$

$$\hat{adj}_{lm} = \hat{adj}_{lm}, \text{ if } l \neq i, m \neq i, \quad (10)$$

$$\hat{x}_i(t+T) \in \mathcal{X}_i^f, \quad (11)$$

$$\hat{x}_i(k) \in \mathcal{X}_i, \quad \hat{u}_i(k) \in \mathcal{U}_i, \quad k = t, \dots, t+T.$$

ここで、 Adj はグラフ $\mathcal{G}$ の隣接行列を、 adj_{ij} は隣接行列 Adj の ij 要素を表す。 $\hat{x}_j^*$ はロボット i の近傍ロボット $j \in \mathcal{N}_i(t)$ が過去の時刻 t_j^{last} に最適化問題 $\mathcal{P}_j(x_j(t_j^{\text{last}}), \tilde{\mathcal{G}}(t_j^{\text{last}}))$ あるいは

後に示す最適化問題 $\mathcal{P}_j'(x_j(t_j^{\text{last}}), \mathcal{N}(t_j^{\text{last}}))$ を解くことで得た予測軌道であり、1-hop 通信によって得られるものとする。ただし、過去に予測軌道を求めていない場合は $\hat{x}_j^* = x_j(t_0)$ とする。また、変数 $\hat{adj}_{ij}$ の最適解 $\hat{adj}_{ij}^*$ より、通信リンクの保存を選択された隣接ロボット $j \in \mathcal{N}_i(t)$ の集合 $\tilde{\mathcal{N}}_i = \{j \mid s.t. \hat{adj}_{ij}^* = 1\}$ を保存しておく。制約条件 (8) の κ は点連結度を表す。

一方、時刻 t において $c_i(t) = 0$ であるがネットワーク制御権 *message* を持っていないロボットは、次の最適化問題 $\mathcal{P}_i'(x_i(t), x_{j \in \mathcal{N}_i(t)}^*(\cdot), \tilde{\mathcal{G}})$ を解くことで予測制御入力列 $\hat{u}_i^*(\cdot; t) = \hat{u}_i^*(\cdot)$ を決定する。この最適化問題は前述の $\mathcal{P}_i$ とは異なり、変数としてネットワーク構造を含んでおらず、現状の通信リンクを維持するような移動制御入力列を決定するものである。

問題 2 ($\mathcal{P}_i'(x_i(t), x_{j \in \mathcal{N}_i(t)}^*(\cdot), \tilde{\mathcal{G}})$).

$$\min_{\hat{x}_i(k), \hat{u}_i(k)} \sum_{k=t}^{t+T-1} \left(\|\hat{x}_i(k) - x_i^d\|_Q^2 + \|\hat{u}_i(k)\|_R^2 \right) + \|\hat{x}_i(t+T) - x_i^d\|_{Q_f}^2, \quad (12)$$

subject to

$$\hat{x}_i(k+1) = f_i(\hat{x}_i(k), \hat{u}_i(k)), \quad \hat{x}_i(t) = x_i(t) \quad (13)$$

$$\|\hat{q}_i(k) - \hat{q}_j^*(k)\| > d_i + d_j, \quad j \in \mathcal{N}_i(t), \quad (14)$$

$$\|\hat{q}_i(k) - \hat{q}_j^*(k)\| \leq r, \quad j \in \tilde{\mathcal{N}}_i, \quad (15)$$

$$\hat{x}_i(t+T) \in \mathcal{X}_i^f, \quad (16)$$

$$\hat{x}_i(k) \in \mathcal{X}_i, \quad \hat{u}_i(k) \in \mathcal{U}_i, \quad k = t, \dots, t+T.$$

以上をまとめたものが、Algorithm 3 である。各ロボットはこれらのアルゴリズムに従って分散的に各時刻 t における制御入力 $u_i(t)$ を決定する。

本研究では、近傍のロボット間 (ロボット i とロボット $j \in \mathcal{N}_i(t)$) のみの衝突回避制約 (6) (14) を考えているので、非近傍の 2 ロボット (ロボット i とロボット $j \notin \mathcal{N}_i(t)$) が決定した制御入力列ではロボット間の衝突を回避できない可能性がある。保守的ではあるが、各ロボットの移動を通信半径の半分からロボットの半径を減じたものに制限する、つまり移動制約

$$\|\hat{x}_i(t+T) - \hat{x}_i(t)\| \leq r/2 - d_i, \quad (17)$$

を加えることで非近傍のロボット同士の衝突を防ぐことができる。

ある 1 台のロボット j が故障したとき、他のすべてのロボットがロボット j の故障を検知することができると仮定する。このとき本提案手法は破綻せず、システムを $\kappa - 1$ 連結に保つ。これを以下に示す。システム中のある 1 台のロボット j が時刻 $t - 1$ に故障した際、

- (1) ロボット j が *message* を保持しておりネットワークから *message* が消失する。
- (2) *message* の送り先がロボット j であり、*message* を送信することができない。
- (3) *message* は次のノード (ロボット i) へ届けられるが、ノード (故障ロボット j) の欠落によりネットワークの連結度が一つ減少し連結度制約 (8) が満たされない。

の三つの事象が起こる可能性がある。事象 (1) が発生し *message* がネットワークから失われたとき、各ロボットは時

刻 t 以降 $\mathcal{P}_i'$ (問題 2) のみを解くこととなり, 全ロボットはその時点でネットワーク構造を維持したまま移動制御を続ける. このネットワーク構造は連結度 κ のグラフから故障ロボット j が欠落したものとなっており, 連結度は少なくとも $\kappa - 1$ となる. 事象 (2) が発生したとき, Algorithm 2 の挙動は未定義であるが, *message* を失うと仮定すると事象 (1) の場合と同様に考えることができ, *message* を保持しているロボットが再度 $\mathcal{P}_i$ (問題 1) を解くと仮定すると事象 (3) の場合と同様に考えることができる. 事象 (3) が発生したとき, $\mathcal{P}_i$ (問題 1) は実行可能解が存在しない. この場合, ロボット i は前回の解 $\hat{x}_i^*(\cdot; t_i^{last})$ に従って移動するが, この予測軌道は時刻 t_i^{last} 時点でロボット i が決定したネットワーク構造を維持するものとなっている. 従って同様に, 連結度は少なくとも $\kappa - 1$ となる. 以上のことから, 提案手法を用いた群ロボットシステムのうち 1 台のロボットが故障したとき, 群ロボットシステムはネットワーク構造を $\kappa - 1$ 連結に保ち続ける.

本研究では移動制御に外乱が存在する場合を考慮していないが, 外乱の大きさが既知である場合には通信範囲制約および衝突回避制約に対して

$$\|\hat{q}_i(k) - \hat{q}_j^*(k)\| + \delta_{ij}(k) \leq r + M(1 - \hat{a}d_{ij}), \quad (18)$$

$$\|\hat{q}_i(k) - \hat{q}_j^*(k)\| - \delta_{ij}(k) > d_i + d_j, \quad (19)$$

のように最大誤差 $\delta_{ij}(k)$ の余裕を持った制約条件を与えることで, 外乱が存在する場合に対しても制約充足性を保証できる.

本研究では目標状態 x_i^f が時間変化しないと仮定したが, 時間変化する場合 $x_i^f(t)$ に対しても自然に拡張できる. ただし有限時間最適制御問題の定義上, ロボット i の予測制御列更新時刻 $t_i \in \{t | s.t. c_i(t) = 0\}$ において, 区間 T まで先の目標状態列 $\{x_i^f(\cdot; t_i)\} = \{x_i^f(t_i; t_i), \dots, x_i^f(t_i + T; t_i)\}$ が定義されている必要がある. またこの影響により, 各ロボットの予測状態列 $\{\hat{x}_i^*(\cdot; t_i)\}$ は現時刻 t 以前の目標状態列 $\{x_i^f(\cdot; t_i)\}$ に収束するよう計算されているため, 目標状態列 $\{x_i^f(\cdot; t)\}$ そのものが動的に変化する場合の収束が遅くなる可能性がある.

4. 計算機シミュレーション

本節では計算機シミュレーションによって本論文の提案手法の有用性を示す. まず, 本手法によって各ロボットがネットワーク維持・衝突回避を行いながら目標位置へ移動できることを確認する. 次に, 筆者の先行研究における提案手法との比較を行い, 本研究の提案手法による改善点を示す.

本節では計算の簡略化のため, 二次元平面を移動するロボットを考え, $x_i = (q_i^x, q_i^y)^T$ と設定し, ダイナミクスを $x_i(t+1) = x_i(t) + u_i(t)$ とした.

4.1 提案手法の検証

提案手法を計算機シミュレーションにて実行した際の結果を示す. ロボット数は $N = 10$, ロボットの車体半径と通信可能半径はそれぞれ $d_i = 0.8$, $r = 10$ とし, 初期位置を Fig. 3 (a), 目標位置を Fig. 3 の (b) とした. Fig. 3 が示すようにこの例では, 各ロボットは最短経路を辿ると衝突する恐れがあり, またそれを回避する際に通信リンクが切断される可能性もある. 最

Fig. 3 Initial and objective positions of robots, and shortest paths

適化問題 $\mathcal{P}_i, \mathcal{P}_i'$ のパラメータは $Q = I, R = I, Q_f = 50I$ (I は単位行列), $\mathcal{X}_i = \mathbb{R}^2, \mathcal{U}_i = \{u | \|u\| \leq 1\}$ とした. さらにここでは, 最適化問題 $\mathcal{P}_i, \mathcal{P}_i'$ を混合整数二次計画問題 (MIQP) で近似し, CVXGEN [11] による内点法と MATLAB で実装した分枝限定法 [8] を用いて解を求めた. また本シミュレーションでは予測制御入力 $\hat{u}_i^*$ の代わりに予測軌道 $\hat{x}_i^*$ を求め, 位置と姿勢を持つロボット

$$\frac{d}{dt} \begin{pmatrix} q_i^x(t) \\ q_i^y(t) \\ \theta_i(t) \end{pmatrix} = \begin{pmatrix} \cos(\theta_i(t)) & 0 \\ \sin(\theta_i(t)) & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v_i(t) \\ w_i(t) \end{pmatrix} \quad (20)$$

を追従させた. ここで q_i^x, q_i^y, θ_i はそれぞれロボット i の x 座標, y 座標, 姿勢であり, v_i, w_i はそれぞれ並進速度, 旋回速度となる制御入力である.

上記の設定における計算機シミュレーションの結果として, $t = 5, 10, 15, 20, 25, 30, 35, 40$ における各ロボットの位置とネットワーク構造, 移動軌跡を Fig. 4 に示す. 各時刻においてロボットは衝突を回避し, ネットワーク構造を連結に保っていることが分かる. また, 各ロボットが目標位置へ収束していることが確認できる.

4.2 先行研究との比較

ここでは本研究の提案手法と先行研究 [7] で提案した手法について, 最適化問題求解時間と制御入力決定周期の 2 点を比較する.

提案手法は本研究と先行研究のいずれも最適制御問題の数値解を求める必要があり, その求解が最も計算時間のかかる部分である. 本研究では 2 種類の最適化問題 $\mathcal{P}_i, \mathcal{P}_i'$ があるが, リンクの切断を決定する $\mathcal{P}_i$ のほうがリンク切断に関する整数変数を含んでいる分だけ計算時間が大きい. よってこちらを先行研究の計算時間と比較する. 先行研究では最適制御問題の整数変数の次元はロボット数 N のオーダー $\mathcal{O}(N)$ だったのに対して, 本研究の最適制御問題の整数変数の次元は $\mathcal{O}(|N_i|)$ となる. ここで $|N_i|$ はロボット i がグラフ上で隣接するロボットの数, つまり頂点 i の次数である. この数はグラフ構造によって変化するが, たかだか $N - 1$ である. またロボット数が大きい群ロボットシステムにおいて, 頂点次数が多くなることは通信範囲の問題や処理能力の問題から避けることが望ましいと考えられる. 従って本研究の提案手法のほうが, 最適化問題求解の計算

Fig. 4 Snapshot of simulation

時間が少ないと考えられる。

Fig. 3 の制御シミュレーションを 1 回行った際の、最適制御問題 $\mathcal{P}_i$ の求解時間の比較を Table 1 に示す。各時刻 t において最適化問題 $\mathcal{P}_i$ を解くロボットはただだか 1 台であるため、ここではシミュレーション内の時間ステップ t における $\mathcal{P}_i$ の求解時間を $Comp(t)[s]$ と定義し、 $Comp(t)$ の区間 $t \in [0, 80]$ での最大値 MAXTIME[s] と平均値 MEANTIME[s] を示している。 $N = 6$ の場合、先行研究手法の計算時間のほうが本研究の提案手法より少なくなっている。これは後に示す制御入力決定機会が増えたことによって複数のロボット間で衝突の可能性が増え、分枝限定法の分枝操作が多くなったことが原因であると考えられる。一方、 $N = 12$ の場合、先行研究手法は全ロボットに対するリンク切断と衝突回避の計算を行うため計算時間が

Table 1 Computational time to solve optimization problem

(a) Number of robots $N = 6$		
	MAXTIME[s]	MEANTIME[s]
Previous method	1.49	0.13
Proposed method	3.17	0.22

(b) Number of robots $N = 12$		
	MAXTIME[s]	MEANTIME[s]
Previous method	≥ 60	
Proposed method	10.00	1.10

多くなっている。先行研究手法はロボット数が増えるに従って確実に計算時間は増加するが、本研究の提案手法はロボット数が増えても頂点次数が増えなければ計算時間は増加しないと予想される。

次に、制御入力列決定の周期について比較する。先行研究の手法では各ステップに 1 台のロボットのみが制御入力列の決定を行っていたため、各ロボットは N ステップに 1 度決定できる、つまり決定周期は $\mathcal{O}(N)$ であった。一方、本研究の提案手法では各ロボットは隣接するロボットと同時に制御入力列を決定しなければよいため、決定周期は $\mathcal{O}(|N_i| + 1)$ となる。計算時間における議論と同様に、頂点次数は大きくならないことが望ましいため、ロボット数が増加しても決定周期は大きく増加しないと考えられる。ただし、ネットワーク構造の変化に伴い頂点次数も変化するため、ネットワーク構造が頻繁に変化する場合に制御入力列の決定機会を得られないロボットが出る可能性がある。この問題については優先度変数 c_i の決定アルゴリズムを改善する必要がある、今後の課題となる。

$N = 6$ 台のロボットで同様のシミュレーションを行った際の、制御入力列を計算した時刻を示したものが Fig. 5 である。横軸がシミュレーション内の時間ステップ t を表し、縦軸が各ロボットの番号 i を表す。Fig. 5 (a) が先行研究の手法を用いた際に各ロボットが制御入力列を得た時間ステップを示しており、Fig. 5 (b) が本研究の提案手法によって各ロボットが制御入力列を得た時間ステップを示している。図中の星形マーカー * は通信リンク切断を考慮した制御入力列を決定した時間ステップ、菱形マーカー $\diamond$ はリンク切断を考慮しないネットワーク構造維持の制御入力列を決定した時間ステップを表す。Fig. 5 より、先行研究手法と比較して本提案手法では制御入力列の決定機会が増加していることが分かる。その一方で、リンク切断を考慮した制御入力列の決定時刻のみに着目すると、本提案手法のほうが決定回数が少なく決定間隔が長いことが確認できる。この点を改善するための優先度の決定方法について検討することが今後の課題となる。

5. おわりに

本論文ではアドホックネットワークを形成する群ロボットシステムが連結性維持と衝突回避を達成するための、動的な優先順位設定による分散制御手法を提案した。本研究では、ネットワークの連結性維持や衝突回避を含む様々な制約を考慮し予測制御入力を生成できる RHC 手法を基に制御手法を組み立てた。RHC 手法を分散化する際は、各ロボットが独立に計算した制御

Fig. 5 Decision timing (when solved optimal control problems) of all robots. Star marker * denotes when robot i decided its control input with network topology control (solved $\mathcal{P}_i$), and diamond marker ◇ denotes without network topology control (solved $\mathcal{P}_i'$). Horizontal axis is time step t and vertical axis is robot index number i (label)

入力列がネットワーク連結性や衝突回避制約を満たしているか注意する必要がある。本提案手法は近傍のロボット同士が同時に制御入力列を決定しないことにより制約の充足性を保証している。また、分散的に計算できる優先度変数によって各ロボットが制御入力列を決定できるタイミングを与えている。提案手法によって、最適化問題の計算時間と制御入力列の決定周期を筆者らの先行研究[7]から改善した。これによってロボット数が増加しても計算時間に大きく影響が出ないと考えられる。

本提案手法では確率的分散優先度決定によって隣接するロボット同士が同時に制御入力列を更新しないことを実現した。この手法は、ネットワーク構造が完全グラフに近い場合に、多数のロボットが制御入力決定の機会を長時間得ることができない恐れを含んでいる。全ロボットの制御入力列更新周期を適切に管理できる分散手法の提案と解析が今後の研究課題となる。

本研究では群ロボットシステム内のロボット数を固定値としたが、ロボットの故障／新規投入を想定するとロボット数が可変の場合を考慮した手法が望ましい。本提案手法ではネットワーク構造の連結度計算にロボット数を利用しており、システム内

のロボット数ならびにネットワーク構造を分散的に得ることのできる分散アルゴリズムを研究することによって、ロボット数可変の場合にも対応できると考えられる。

その他、本提案手法では通信リンクを切断できるロボット数は先行研究から改善されておらず、各時刻において1台のみに限定されている。したがってロボット数と通信リンク数の増加に伴い、多くのロボットがリンク切断のために待機する時間が増加してしまう恐れがある。各時刻において複数台のロボットが同時にリンクの切断を行えるような手法を研究し、この問題を解決することが今後の課題となる。また、より大規模な群ロボットシステムにおけるメッシュネットワーク構造の頂点次数と計算時間との関係性の考察や、実機ロボットによる提案手法の検証実験も今後の課題である。

参 考 文 献

- [1] C.A. Rabbath, C.-Y. Su and A. Tsourdos: "Special issue on multivehicle systems cooperative control with application," IEEE Trans. on Control Systems Technology, vol.15, no.4, 2007.
- [2] R.M. Murray: "Recent research in cooperative control of multi-vehicle systems," Journal of Dynamic Systems, Measurement, and Control, vol.129, no.5, pp.569-754, 2007.
- [3] 田所諭: "閉鎖空間内高速走行探索群ロボット", 日本ロボット学会誌, vol.27, no.10, pp.1107-1110, 2009.
- [4] Ji Meng and M. Egerstedt: "Distributed Coordination Control of Multiagent Systems While Preserving Connectedness", IEEE Trans. on Robotics, vol.23, no.4, pp.693-703, 2007.
- [5] M.M. Zavlanos and G.J. Pappas: "Distributed connectivity control of mobile networks," IEEE Trans. on Robotics, vol.24, no.6, pp.1416-1428, 2008.
- [6] 村山暢, 長野明紀, 羅志偉: "通信範囲制約を考慮するネットワーク群ロボットシステムの分散制御手法", システム制御情報学会論文誌, vol.25, no.9, pp.239-246, 2012.
- [7] T. Murayama: "Trajectory Planning for Multi-vehicle System Considering Network Connectivity and Collision Avoidance," Proc. IEEE/SICE International Symposium on System Integration (SII2013), pp.889-894, 2013.
- [8] 根和幸, 福島宏明, 松野文俊: "衝突回避問題に適合した分枝限定法に基づく複数移動体の編隊制御", 計測自動制御学会論文集, vol.44, no.1, pp.36-43, 2008.
- [9] M. Nagahara, D.E. Quevedo and J. Ostergaard: "Sparse Pack- etized Predictive Control for Networked Control Over Erasure Channels," IEEE Trans. on Automatic Control, vol.59, no.7, pp.1899-1905, 2014.
- [10] W.B. Dunbar and R.M. Murray: "Distributed receding horizon control for multi-vehicle formation stabilization," Automatica, vol.42, no.4, pp.549-558, 2006.
- [11] J. Mattingley and S. Boyd: "CVXGEN: A Code Generator for Embedded Convex Optimization," Optimization and Engineering, vol.13, no.1, pp.1-27, 2012.

村山 暢 (Toru Murayama)

2008年神戸大学工学部情報知能工学科卒業。2009年9月神戸大学大学院工学研究科情報知能学専攻博士前期課程修了。2012年9月神戸大学大学院工学研究科情報知能学専攻博士後期課程修了。同大学大学院システム情報工学科学術推進研究員を経て、2013年4月より和歌山工業高等専門学校知能機械工学科助教、現在に至る。(日本ロボット学会正会員)