マルチエージェントシステムに対するすべての勾配型分散制御器の パラメータ表現

桜 間 一 徳*・東 俊 一**・杉 江 俊 治**

Parametrization of All Distributed Controllers for Networked Multi-agent Systems

Kazunori SAKURAMA*, Shun-ichi AZUMA** and Toshiharu SUGIE**

In this paper, a parametrization of all distributed controllers based on the gradient-flow method is presented for networked multi-agent systems. By using the proposed parametrization, the first step in designing controllers will be to present a class of distributed controllers. Then, desired controllers are designed by determining the rest of freedom of controllers, which is represented by free parameters, according to tasks. The new design procedure removes the trial-and-error to keep the distributedness of controllers from conventional design methods. The proposed parametrization will be a strong tool to assist in designing distributed controllers for various kinds of tasks.

Key Words: multi-agent system, distributed controller, formation control

1. はじめに

近年、マルチエージェントシステムの自律分散制御に関する研究が盛んに行なわれている¹⁾。自律分散制御とは、局所情報を基に自律制御を行なうエージェントが、ネットワーク通信などの相互作用を通じて協調動作を実現することである。このような制御方式は複数の移動体やプラントを含むさまざまな場面で要求されるものであり、人工衛星のランデブや姿勢同期、飛行機のフォーメーションフライト、警備ロボットによる不審者包囲、移動体によるセンサネットワークの構築などの産業利用が期待されている。

マルチエージェントシステムの代表的な問題としては、合意²⁾、姿勢同期³⁾、フォーメーション⁴⁾、囲い込み⁵⁾、捕獲⁶⁾、被覆⁷⁾などが、精力的に研究されている。このような研究では、タスクに応じた目的関数を最大化(あるいは最小化)するような制御器を設計する、いわゆる勾配フロー法⁸⁾が背景にある。たとえば、合意やフォーメーションなどの配置問題では、エージェント同士の位置偏差を目的関数とした制御器が設計されている。それ以外のタスクに対する目的関数については、文献 9) を参照されたい。勾配フロー法は目的関数の勾配を制御入力とする明瞭な手法であるため、マルチエージェ

ントシステムの制御器設計における基礎的なツールとして位置づけることができる。

その一方で、マルチエージェントシステムでは、各エージェントに対してネットワークを通じて得られる局所情報のみを利用する、いわゆる分散制御器を設計することが重要である。したがって、勾配フロー法では、目的関数の勾配が分散的になることに留意しなければならない。そのため、多くの既存研究では、i) タスクに応じて目的関数を設計、ii) 勾配を計算して分散性を確認、iii) 分散的でなければ再設計、という手順が取られている。このような試行錯誤が必要となる理由として、勾配が分散的になる目的関数のクラスが不明瞭であることが挙げられる。

そこで、本論文では、勾配が分散的になるようなすべての目的関数に対するパラメータ表現を与える。提案するパラメータ表現を用いると、制御器設計における最初の手順が分散的な制御器のクラスを与えることになる。あとは、自由度として残されたパラメータをタスクに応じて設計することで、所望の分散制御器を設計することができる。この新しい設計法では、従来法における制御器の分散性を確保するための試行錯誤が必要なくなる。本パラメータ表現はあらゆるタスクに対して有効であるため、マルチエージェントシステムの分散制御器の設計を支援する強力なツールとなる。

本論文で用いる記号を定義する。自然数の部分集合 $\mathcal{N} = \{1, 2, \dots, n\}$ に対して、べき集合 (すべての部分集合の集合族) を $\text{pow}(\mathcal{N})$ と表記する。 $\mathcal{N}$ の部分集合 $\mathcal{I} = \{i_1, i_2, \dots, i_m\} \in \text{pow}(\mathcal{N})$ 、ベクトル $x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$ 、十分滑らかな関数 $V: \mathbb{R}^n \rightarrow \mathbb{R}$ に対して、以下の表記を用いる。

* 鳥取大学大学院工学研究科 鳥取市湖山町南 4 丁目 101

** 京都大学大学院情報学研究科 宇治市五ヶ庄

* Graduate School of Engineering, Tottori University, 4-101 Koyama-Minami, Tottori

** Graduate School of Informatics, Kyoto University, Gokasho, Uji

(Received August 23, 2011)

(Revised March 28, 2012)

- ベクトル x の添字集合 $\mathcal{I}$ に対応する成分ベクトル

$$(x_i)_{i \in \mathcal{I}} := (x_{i_1}, x_{i_2}, \dots, x_{i_m}) \in \mathbb{R}^m$$

- 変数 x_i , $i \in \mathcal{I}$ による関数 $V(x)$ の m 階偏導関数

$$\partial_{\mathcal{I}} V(x) = \partial_{i_1 i_2 \dots i_m} V(x) := \frac{\partial^m V(x)}{\partial x_{i_1} \partial x_{i_2} \dots \partial x_{i_m}}$$

2. 問題設定

2.1 対象システム

本論文では、以下の1階微分方程式にしたがう n エージェントからなるシステムを考える。

$$\dot{x}_i = u_i, \quad i \in \mathcal{N} \quad (1)$$

ただし、 $x_i, u_i \in \mathbb{R}$ はエージェント i の状態と入力を、 $\mathcal{N} := \{1, 2, \dots, n\}$ はエージェント集合を表わす。状態をまとめて $x := (x_i)_{i \in \mathcal{N}}$ と表記する。

エージェントが情報交換を行なうネットワークの通信経路を、エージェントのペア集合 $\mathcal{E}$ で与える。

$$\mathcal{E} := \{\{i, j\} : i, j \in \mathcal{N}, i, j \text{ connected}\}$$

以降では、 $\mathcal{N}$ をノード集合、 $\mathcal{E}$ をエッジ集合とし、ネットワークグラフを $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ と表記する。なお、エッジは2つのノードの集合として表わしているため、 $\{i, j\} = \{j, i\}$ が成り立つ。ここで、ネットワークは無向であることを仮定している。

このネットワーク上でエージェント i が通信可能なエージェントの集合を隣接集合と呼び、

$$\mathcal{N}_i := \{j \in \mathcal{N} : \{i, j\} \in \mathcal{E}\}$$

と定義する。各エージェントが利用できる情報は、自分自身の状態とネットワークで直接つながったエージェントの状態であるとする^(注1)。したがって、エージェント i の制御器を実装するためには、入力 u_i が適当な関数 $f_i : \mathbb{R}^{1+|\mathcal{N}_i|} \rightarrow \mathbb{R}$ に対して、

$$u_i = f_i(x_i, (x_j)_{j \in \mathcal{N}_i}) \quad (2)$$

という形をしていなければならない。このような形をした制御器は分散的であるといわれる。

つぎに、ネットワークでつながっていないエージェントのペアの集合をつぎで定義する。

$$\bar{\mathcal{E}} := \{\{i, j\} : \{i, j\} \notin \mathcal{E}, i, j \in \mathcal{N}, i \neq j\} \quad (3)$$

これをエッジ集合としたグラフ $\bar{\mathcal{G}} = (\mathcal{N}, \bar{\mathcal{E}})$ は、元のグラフ $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ の補グラフと呼ばれる。これに応じて、エージェ

(注1) 情報通信の形態によっては、エージェントを媒介することで、直接つながっていないエージェントの情報も取得できる。しかし、過大な参照情報による計算負荷の増加、および情報の遅れによる安定性の損失を防ぐため、本論文では、このようなネットワーク的に離れたエージェントの情報は扱わないこととする。

Fig. 1 Example of a network

ント i が通信不可能なエージェントの集合を非隣接集合と呼ぶことにし、以下のように表記する。

$$\bar{\mathcal{N}}_i := \{j \in \mathcal{N} : \{i, j\} \in \bar{\mathcal{E}}\} \quad (4)$$

〔例題1〕 Fig. 1 のネットワークを取り上げる。実線が通信経路を表わす。このネットワークのグラフ $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ および補グラフ $\bar{\mathcal{G}} = (\mathcal{N}, \bar{\mathcal{E}})$ に対する各集合は以下で与えられる。

$$\mathcal{N} = \{1, 2, 3, 4, 5, 6, 7\}$$

$$\mathcal{E} = \{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{3, 4\}, \dots, \{6, 7\}\}$$

$$\bar{\mathcal{E}} = \{\{1, 5\}, \{1, 6\}, \{1, 7\}, \{2, 3\}, \dots, \{3, 7\}\}$$

$$\mathcal{N}_1 = \{2, 3, 4\}, \quad \bar{\mathcal{N}}_1 = \{5, 6, 7\}$$

$$\mathcal{N}_2 = \{1\}, \quad \bar{\mathcal{N}}_2 = \{3, 4, 5, 6, 7\}$$

2.2 勾配フロー法による制御器設計

システム (1)、ネットワーク $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ に対して、合意、姿勢同期、被覆などのタスクが与えられたとする。ここでは、これらのタスクを実現するために、勾配フロー法を利用して制御器を設計することを考える。設計された制御器が (2) 式を満たし分散的であるか否かが、この設計問題の争点となる。

勾配フロー法とは、タスクの達成度を評価する目的関数を設計し、その勾配によってエージェントを制御する方法である。ここでは、タスクの詳細には触れずに、タスクに応じた目的関数 $V(x) \in \mathbb{R}$ が設計されたとする。このとき、勾配入力

$$u_i = \partial_i V(x), \quad i \in \mathcal{N} \quad (5)$$

によってエージェントを制御することが勾配フロー法である。この制御入力によって、目的関数 $V(x)$ を局所最大化することができる。実際、(1)、(5) 式より目的関数の時間微分は

$$\dot{V}(x) = \sum_{i=1}^n \partial_i V(x) \dot{x}_i = \sum_{i=1}^n (\partial_i V(x))^2$$

となるため、 $V(x)$ は非減少である。

ここで問題となるのは、勾配入力 (5) がネットワーク $\mathcal{G}$ 上で実装できる、すなわち、(2) 式を満たし分散的であるとは限らないことである。勾配入力が分散的となるように目的関数を設計することが、勾配フロー法による分散制御器の設計問題の争点となる。本論文では、このような目的関数は分散勾配をもつと定義する。

【定義】 グラフ $\mathcal{G}$ において関数 $V : \mathbb{R}^n \rightarrow \mathbb{R}$ が分散勾配をもつとは、任意の $i \in \mathcal{N}$ に対して、以下を満たす関数 $f_i : \mathbb{R}^{1+|\mathcal{N}_i|} \rightarrow \mathbb{R}$ が存在することである。

$$\partial_i V(x) = f_i(x_i, (x_j)_{j \in \mathcal{N}_i}) \quad (6)$$

本論文の主題は、勾配フロー法によるすべての分散制御のパラメータ表現を行なうことである。パラメータ表現とは、不変な構造部分と自由パラメータ部分に分けた表現形式である。これによって、与えられたタスクに応じて自由パラメータを設計することで、分散性を崩さずに制御器を設計・調整することが可能となる。ここで、(6) 式における関数 $f_i(x_i, (x_j)_{j \in \mathcal{N}_i})$ によって、分散制御器がパラメータ表現されているわけではない。なぜなら、関数 $f_i(x_i, (x_j)_{j \in \mathcal{N}_i})$ を適当に与えても、すべての i に対して (6) 式を満たす共通の目的関数 $V(x)$ が存在するとは限らないためである。本論文では、目的関数 $V(x)$ 自身のパラメータ化によって、勾配フロー法による分散制御器のパラメータ表現を行なう。

以上をまとめ、本論文で扱う問題を与える。

[問題 1] 与えられたグラフ $\mathcal{G}$ に対して、分散勾配をもつすべての目的関数 $V(x)$ のパラメータ表現を与えよ。

3. 分散勾配をもつ目的関数のパラメータ表現

ここでは、問題 1 を考えるために、まず、定義式 (6) と等価な未知関数 $f_i(x_i, (x_j)_{j \in \mathcal{N}_i})$ を含まない条件式を導く。この定義式は、勾配成分 $\partial_i V(x)$ が非隣接集合 $\bar{\mathcal{N}}_i$ には依存しないことを意味する。したがって、両辺を変数 $x_j, j \in \bar{\mathcal{N}}_i$ によって偏微分すると、

$$\partial_{ij} V(x) = \partial_j(\partial_i V(x)) = \partial_j f_i(x_i, (x_k)_{k \in \mathcal{N}_i}) \equiv 0$$

を得る。これが任意のエージェントのペア $i \in \mathcal{N}, j \in \bar{\mathcal{N}}_i$ について成り立つことは、(4) 式より補グラフのエッジ集合 $\bar{\mathcal{E}}$ を用いると次式で表わされる。

$$\partial_{ij} V(x) \equiv 0, \forall \{i, j\} \in \bar{\mathcal{E}} \quad (7)$$

この式は目的関数 $V(x)$ が分散勾配をもつことと等価である。

[補題 1] C^2 級関数 $V(x)$ が、グラフ $\mathcal{G} = (\mathcal{E}, \mathcal{N})$ において分散勾配をもつための必要十分条件は、(7) 式を満たすことである。

証明 必要性は補題の前に示したため、十分性を示す。(7) 式を仮定する。あるエージェント $i \in \mathcal{N}$ とその非隣接集合にある $j \in \bar{\mathcal{N}}_i$ について考える。このとき (7) 式が成り立つ。この式を変数 x_i, x_j に対する偏微分方程式として捉えると、その一般解は、任意関数 $W_1((x_k)_{k \in \mathcal{N} \setminus \{i\}}), W_2((x_k)_{k \in \mathcal{N} \setminus \{j\}})$ に対して以下のように得られる (付録の補題 4 を参照のこと)。

$$V(x) = W_1((x_k)_{k \in \mathcal{N} \setminus \{i\}}) + W_2((x_k)_{k \in \mathcal{N} \setminus \{j\}})$$

この式を x_i について偏微分すると

$$\partial_i V(x) = \partial_i W_2((x_k)_{k \in \mathcal{N} \setminus \{j\}}) = f_{ij}((x_k)_{k \in \mathcal{N} \setminus \{j\}})$$

を得る。最後は適当な関数 $f_{ij}: \mathbb{R}^{n-1} \rightarrow \mathbb{R}$ によって置き換えた。これが任意の非隣接集合の要素 $j \in \bar{\mathcal{N}}_i$ について成り立つため、ある関数 $f_i: \mathbb{R}^{1+|\mathcal{N}_i|} \rightarrow \mathbb{R}$ が存在し

$$\partial_i V(x) = f_i((x_k)_{k \in \mathcal{N} \setminus \bar{\mathcal{N}}_i}) = f_i((x_k)_{k \in \mathcal{N}_i \cup \{i\}})$$

が成り立つ。ただし、最後の等式は (3), (4) 式から導いた。これより、(6) 式を得る。□

補題 1 が示唆することは、目的関数 $V(x)$ が分散勾配をもつためには、通信経路のないエージェント i, j が、(7) 式の意味で相関がないようにしなければならないことである。したがって、「通信経路のあるエージェントのみが相関をもつ関数」を素にした目的関数は、分散勾配をもつことが期待される。以下では、実際に、このようにパラメータ表現された目的関数が分散勾配をもつことを示す。

このアイディアに基づいた目的関数を表記するために、ネットワーク上で通信経路のあるエージェント同士の集合である、クリークの定義を与える¹⁰⁾。

[定義] グラフ $\mathcal{G}$ におけるノード集合の部分集合 $\mathcal{I} \subset \mathcal{N}$ がクリークであるとは、それによって誘導される部分グラフが完全である、すなわち $\{i, j\} \in \mathcal{E}, \forall i, j \in \mathcal{I}, i \neq j$ を満たすことである。また、クリーク $\mathcal{I}$ の要素数 $|\mathcal{I}|$ を位数という。グラフ $\mathcal{G}$ における全クリークの集合族をクリーク族と呼び、 $\text{clq}(\mathcal{G}) \subset \text{pow}(\mathcal{N})$ で表わす。つまり、以下のように定義する。

$$\text{clq}(\mathcal{G}) := \{\mathcal{I} \subset \mathcal{N} : \{i, j\} \in \mathcal{E}, \forall i, j \in \mathcal{I}, i \neq j\} \quad (8)$$

[例題 2] Fig. 1 のネットワークにおいて、鎖線の枠内のグループは、それぞれ位数 3 と 4 のクリークである。このネットワーク中のクリーク族は以下で与えられる。

$$\begin{aligned} \text{clq}(\mathcal{G}) = & \{\{4, 5, 6, 7\}, \{1, 3, 4\}, \{4, 5, 6\}, \{4, 5, 7\}, \\ & \{4, 6, 7\}, \{5, 6, 7\}\} \cup \mathcal{E} \cup \{\{1\}, \{2\}, \dots, \{n\}\} \end{aligned}$$

通信経路のあるエージェントの集合がクリークであることに注意すると、上述のアイディアは「クリークに属するエージェントの状態のみに依存する関数」を素にした目的関数を考えることになる。この関数をクリーク $\mathcal{I}$ に対応して $W_{\mathcal{I}}((x_k)_{k \in \mathcal{I}})$ で与えると、このような目的関数は以下のように表現される。

$$V(x) = \sum_{\mathcal{I} \in \text{clq}(\mathcal{G})} W_{\mathcal{I}}((x_k)_{k \in \mathcal{I}}) \quad (9)$$

以下では、実際にこの考えは正しく、目的関数 (9) が分散勾配をもつことを示す。さらに、(9) 式を満たす関数 $W_{\mathcal{I}}((x_k)_{k \in \mathcal{I}})$ が存在しないとき、目的関数 $V(x)$ は分散勾配をもたないという強い結果を示す。これより、分散勾配をもつすべての目的関数 $V(x)$ は、関数族 $W_{\mathcal{I}}((x_k)_{k \in \mathcal{I}})$ によって (9) 式でパラメータ表現されることになる。

まず、準備のために、任意の十分滑らかな関数 $V(x)$ が、ある性質をもつ関数族の和で表わすことができることを示す。なお、この結果は分散勾配とは無関係である。

[補題 2] 十分滑らかな関数 $V(x)$ に対して、 $\mathcal{N}$ の部分集合族 $\mathcal{J} \subset \text{pow}(\mathcal{N})$ とその要素に対応する関数族 $W_{\mathcal{I}}: \mathbb{R}^{|\mathcal{I}|} \rightarrow \mathbb{R}, \mathcal{I} \in \mathcal{J}$ および定数 $c \in \mathbb{R}$ が存在し、以下が満たされる。

$$V(x) = \sum_{I \in \mathfrak{I}} W_I((x_k)_{k \in I}) + c \quad (10)$$

$$\partial_I W_I((x_k)_{k \in I}) \neq 0, \forall I \in \mathfrak{I} \quad (11)$$

$$\mathcal{J} \not\subset \mathcal{I}, \forall \mathcal{I}, \mathcal{J} \in \mathfrak{I}, \mathcal{I} \neq \mathcal{J} \quad (12)$$

証明 付録 A を参照されたい. $\square$

さらに準備のため, 関数 $W_I((x_k)_{k \in I})$ に関する偏微分の計算式を与える.

[補題 3] 関数 $W_I: \mathbb{R}^{|I|} \rightarrow \mathbb{R}$ に対して以下が成り立つ.

$$\partial_{\mathcal{J}} W_I((x_k)_{k \in I}) \equiv 0, \text{ for } \mathcal{I}, \mathcal{J} \subset \mathcal{N} \text{ s.t. } \mathcal{J} \not\subset \mathcal{I} \quad (13)$$

特別な場合として, 次式が成り立つ.

$$\partial_{ij} W_I((x_k)_{k \in I}) \equiv 0, \text{ for } \mathcal{I} \in \text{clq}(\mathcal{G}), \{i, j\} \in \bar{\mathcal{E}} \quad (14)$$

証明 (13) 式の条件を満たす $\mathcal{I}, \mathcal{J}$ に対して, $j \in \mathcal{J} \cup \mathcal{I}^c$ なる要素が存在する. これより (13) 式の左辺を計算すると,

$$\partial_{\mathcal{J}} W_I((x_k)_{k \in I}) = \partial_{\mathcal{J} \setminus \{j\}} \partial_j W_I((x_k)_{k \in I}) \equiv 0$$

を得る. つぎに, ある $\mathcal{I} \in \text{clq}(\mathcal{G})$ と $\{i, j\} \in \bar{\mathcal{E}}$ を考える. このとき, $\{i, j\} \not\subset \mathcal{I}$ が成り立つ. なぜなら, $\{i, j\} \subset \mathcal{I}$ ならば, (8) 式より $\{i, j\} \in \mathcal{E}$ となり, 矛盾するためである. よって, $\mathcal{J} = \{i, j\}$ に対し (13) 式が成り立ち, (14) 式を得る. $\square$

最後に, 本論文の主結果を示す. つぎの定理で示すことは, 分散勾配をもつ目的関数を (10) 式のように分解すると, 必ず (9) 式の形に帰着することである. この結果は, 分散勾配をもつすべての目的関数が (9) 式のパラメータ表現で与えられることを示す. ただし, この表現において, 構造部分は関数の和 $\sum_{\mathcal{I} \in \text{clq}(\mathcal{G})}$, 自由パラメータはクリークに依存する関数 $W_I((x_k)_{k \in I})$ である.

《定理 1》 十分滑らかな関数 $V(x)$ がグラフ $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ において分散勾配をもつための必要十分条件は, クリーク $\mathcal{I} \in \text{clq}(\mathcal{G})$ に対応した関数族 $W_I: \mathbb{R}^{|I|} \rightarrow \mathbb{R}$ が存在し, (9) 式の形で表わされることである.

証明 十分性: (9) 式を $\{i, j\} \in \bar{\mathcal{E}}$ に対する変数 x_i, x_j によって偏微分する. このとき, (14) 式より (7) 式を得る. したがって, 補題 1 より $V(x)$ は分散勾配をもつ.

必要性: 分散勾配をもつ関数 $V(x)$ が, (9) 式の形をしていることを示す. 補題 2 より, 関数 $V(x)$ を (10)–(12) 式を満たすように分解できる. このとき, (10) 式が (9) 式の形状に帰着することを証明する. (ここで, 定数 c を 0 としても, 一般性を失わない.) そのためには, 包含関係 $\mathfrak{I} \subset \text{clq}(\mathcal{G})$ を示せばよい. この関係式を背理法によって示すために, $\mathcal{J} \in \mathfrak{I}$ かつ $\mathcal{J} \not\subset \text{clq}(\mathcal{G})$ を満たす集合 $\mathcal{J} \subset \mathcal{N}$ が存在することを仮定し, 矛盾を導く.

背理法の仮定 ($\mathcal{J} \in \mathfrak{I}$) と (12) 式より, 任意の $\mathcal{I} \in \mathfrak{I} \setminus \{\mathcal{J}\}$ に対して, $\mathcal{J} \not\subset \mathcal{I}$ を得る. したがって, (13) 式より, $\partial_{\mathcal{J}} W_I((x_k)_{k \in I}) \equiv 0$ が成り立つ. よって, (10) 式から

$$\partial_{\mathcal{J}} V(x) = \sum_{\mathcal{I} \in \mathfrak{I}} \partial_{\mathcal{J}} W_I((x_k)_{k \in I}) = \partial_{\mathcal{J}} W_{\mathcal{J}}((x_k)_{k \in \mathcal{J}}) \quad (15)$$

が成り立つ. つぎに, 背理法の仮定 ($\mathcal{J} \not\subset \text{clq}(\mathcal{G})$) と (8) 式より, $\{i, j\} \in \bar{\mathcal{E}}$ なる $i, j \in \mathcal{J}$ が存在する. ここで, $V(x)$ が分散勾配をもつという仮定から補題 1 より (7) 式が満たされるため, このような i, j に対して以下が成り立つ.

$$\partial_{\mathcal{J}} V(x) = \partial_{\mathcal{J} \setminus \{i, j\}} \partial_{ij} V(x) \equiv 0 \quad (16)$$

ここで, $\mathcal{J} \in \mathfrak{I}$ であることより, (15), (16) 式は (11) 式に矛盾する. これより, 題意は示された. $\square$

定理 1 を用いた分散制御器の設計について考察する. 目的関数 (9) の勾配入力 (5) を計算すると, つぎを得る.

$$u_i = \partial_i V(x) = \sum_{\mathcal{I} \in \text{clq}_i(\mathcal{G})} \partial_i W_I((x_k)_{k \in I}) \quad (17)$$

ここで, $\text{clq}_i(\mathcal{G})$ は i を含むクリークの集合族である.

$$\text{clq}_i(\mathcal{G}) := \{\mathcal{I} \in \text{clq}(\mathcal{G}) : i \in \mathcal{I}\}$$

したがって, 各エージェントは, 自分の属するクリークを把握し, 各クリーク $\mathcal{I}$ に所属するエージェントと共通で自由パラメータ $W_I((x_k)_{k \in I})$ を設計すればよい. すなわち, 各エージェントは, 制御器を設計する際に, ネットワーク全体を把握する必要はない.

(注意 1) ロボットが他機をカメラで認識する場合には各ロボットは他機との相対距離のみが取得できる. このような相対情報を制御入力 (17) で利用するためには, 目的関数 (9) の各パラメータを $W_I((x_i - x_j)_{i, j \in \mathcal{I}})$ と, クリーク単位で状態差を取るようにすればよい.

(注意 2) 分散制御器によって勾配系を実現するの研究として, 文献 11) が挙げられる. この文献では, 状態差 $x_i - x_j, \{i, j\} \in \mathcal{E}$ に対する勾配系を実現する制御入力 $u_i = g_i(\sum_{j \in \mathcal{N}_i} (x_j - x_i))$ という形で設計されている. この形は, 制御器に使える情報が, 隣接集合内の全エージェントとの状態差の総和であることを意味しており, 設計できる制御器のクラスが限定されている. 一方, 本論文では, 入力 (5) によって, 状態 $x_i, i \in \mathcal{N}$ についての勾配系を実現している. 導出される制御入力 (17) のクラスはより広く, 無向グラフに限れば文献 11) のものを含んでいる. その結果, 4 章の設計例を含め, より多くのタスクに対応することが可能となる.

4. 等間隔フォーメーションによる設計例

ここでは, 等間隔フォーメーション, すなわち, Fig. 2 のようにエージェントを番号順に等間隔に配置するタスクを考える. このタスクは, 次式で定式化される.

$$\lim_{t \rightarrow \infty} \{(x_i(t) - x_{i-1}(t)) - (x_{i+1}(t) - x_i(t))\} = 0, \quad i = 2, 3, \dots, n-1 \quad (18)$$

ここでは, 所望の間隔が事前には決められていないため, 各

Fig. 2 Equally spaced formation

Fig. 3 Network for simulation

エージェントはほかのエージェントのようすを窺いながら間隔を合わせなくてはならない。したがって、事前に所望の間隔が決まっている、既存のフォーメーション制御とは異なる問題設定である。

本タスクの達成度を評価する目的関数を (9) 式の形で設計する。(18) 式を任意のエージェントに対する所望の位置関係に書き直すと、以下を得る。

$$\lim_{t \rightarrow \infty} \left(\frac{x_i(t) - x_j(t)}{i - j} - \frac{x_i(t) - x_k(t)}{i - k} \right) = 0, \quad i, j, k \in \mathcal{N}, i \neq j, i \neq k, j \neq k \quad (19)$$

ここでは、(19) 式の 2 乗偏差である次式を、目的関数の自由パラメータ $W_{\mathcal{I}}((x_k)_{k \in \mathcal{I}})$, $\mathcal{I} = \{i, j, k\}$ として与える。

$$W_{\mathcal{I}}((x_k)_{k \in \mathcal{I}}) = -c_{\mathcal{I}} \{ (j-k)x_i + (k-i)x_j + (i-j)x_k \}^2 \quad (20)$$

ただし、 $c_{\mathcal{I}} > 0$ は定数である。ここで、 $\mathcal{I}$ は位数 3 のクリークであり、それ以外のクリークに対しては、 $W_{\mathcal{I}} = 0$ とする。以上より、本タスクを実現する分散制御器 u_i が、(17), (20) 式によって得られる。

最後に、設計した分散制御器の有効性を確認するために、Fig. 3 のネットワークを対象にしたシミュレーションを行なう。ここでは、 $c_{\mathcal{I}} = 1$ とする。適当な初期座標からの各エージェントの座標 $x_i(t)$ の時間変化を Fig. 4 に示す。これより、初期時刻では揃っていないエージェントの間隔が、終端時刻ではほぼ揃っていることがわかる。実際、すべての間隔は約 2.2 に収束している。以上より、等間隔フォーメーション (18) が実現されていることが確認された。

類似の整列問題を取り扱っている文献 [12] と比較して、本論文の等間隔フォーメーションの意義について検討する。この文献の整列問題では、事前に、両端のエージェントの位置 $x_a^*, x_b^* \in \mathbb{R}$ が与えられている。このような場合、つぎの目的関数を用いることで、タスクを達成することができる。

Fig. 4 Motions of 20 agents applying distributed controllers

Fig. 5 Line network

$$V(x) = \sum_{i=1}^{n-1} (x_{i+1} - x_i - d)^2 + (x_1 - x_a^* - d)^2 + (x_b^* - x_n - d)^2 \quad (21)$$

ただし、 $d = (x_b^* - x_a^*) / (n-1)$ はエージェントの間隔を表す。定理 1 によると、目的関数 (21) による分散制御器を実現するためのネットワークは、Fig. 5 のような直線型でよい。この整列問題に比べると、本論文の等間隔フォーメーションでは、エージェントの位置・間隔が事前に決められておらず、エージェントの収束先の自由度が高い。このようなタスクを柔軟に達成するためには、エージェントはネットワークからの情報を基に、移動しながら適切な間隔を周囲と調整して決める必要がある。定理 1 を用いると、柔軟なタスクを表現する目的関数の自由パラメータ (20) を分散制御器として実装するためには、Fig. 3 のような比較的高い密度のネットワーク構造が必要になることがわかる。

(注意 3) 本論文の大きな成果のひとつとして、位数の高いクリークに依存する関数を用いても、目的関数が分散匀配をもつことが明らかになったことが挙げられる。上述の例のように、位数の高いクリークを用いることで、収束先の自由度が高いタスクを柔軟に達成できるようになる。たとえば、エージェントがいくつかのフォーメーション形状の中から分散的にパターンを選び出し、形成するという問題 (分散のパターン選択) が、クリークを利用することで効果的に実現できることが示されている^[13]。この問題は、設計者はいくつかのフォーメーション形状を与えるものの、どの形状を実現すればよいのかを決めておらず、収束先が複数あるという点で自由度が高い。また、既存のタスクである合意^[2]、協調捕獲^[6]、被覆^[7]などは、適切な目的関数があれば、本論文の枠組みで捉えることが可能である。既存の研究では、エッジ、すなわち位数 2 のクリーク単位で制御器を設計しているが、位数の高いク

リークを利用することで、捕獲半径が決まっていらないなどの自由度の高いタスクを実現できる可能性がある。

5. おわりに

本論文では、勾配フロー法に基づくすべての分散制御器に対するパラメータ表現を与えた。提案するパラメータ表現はあらゆるタスクに対して有効な基礎的なものであり、分散制御器の設計を支援する強力なツールとなる。本論文が明らかにしたことは、分散勾配をもつすべての目的関数がクリークに依存する関数族によってパラメータ表現されることである。グラフ理論において重要な概念であるクリークが分散制御器の設計に結びついたことは、注目すべき結果である。今後の課題は、位数の高いクリークを用いることで、分散制御器によって実現できるタスクの幅を広げることである。

参考文献

- 1) 滑川 徹 (編): ミニ特集 協調とフォーメーションの制御理論, 計測と制御, **46**-11, 823/869 (2007)
- 2) R. Olfati-Saber and R.M. Murray: Consensus problems in networks of agents with switching topology and time-delays, *IEEE Trans. Automatic Control*, **49**-9, 1520/1533 (2004)
- 3) Y. Igarashi, T. Hatanaka, M. Fujita and M.W. Spong: Passivity-based attitude synchronization in $SE(3)$, *IEEE Trans. Control Systems Technology*, **17**-5, 1119/1134 (2009)
- 4) A. Fax and R.M. Murray: Information flow and cooperative control of vehicle formations, *IEEE Trans. Automatic Control*, **49**-9, 1465/1476 (2004)
- 5) T.-H. Kim and T. Sugie: Cooperative control for target-capturing task based on a cyclic pursuit strategy, *Automatica*, **43**-8, 1426/1431 (2007)
- 6) 小林, 大坪, 細江: 群ロボットによる協調捕獲行動の自律分散制御, 計測自動制御学会論文集, **43**-8, 663/671 (2007)
- 7) J. Cortés, S. Martínez, T. Karatas and F. Bullo: Coverage control for mobile sensing networks, *IEEE Trans. Robotics and Automation*, **20**-2, 243/255 (2004)
- 8) W.M. Hirsch and S. Smale: Differential equations, dynamical systems and linear algebra, Academic Press (1974)
- 9) S. Martínez, J. Cortés and F. Bullo: Motion coordination with distributed information, *IEEE Control Systems Magazine*, **27**-4, 75/88 (2007)
- 10) 恵羅, 土屋: グラフ理論, 産業図書株式会社 (1996)
- 11) 湯浅, 伊藤: 自律分散システムの構造理論, 計測自動制御学会論文集, **25**-12, 1355/1362 (1989)
- 12) I. Wagner and A.M. Bruckstein: Row straightening by local interactions, *Circuits, Systems and Signal Processing*, **16**-3, 287/305 (1997)
- 13) 桜間, 東, 杉江: 分散的パターン選択によるマルチエージェントシステムのフォーメーション制御, システム制御情報学会論文誌, **25**-3, 45/53 (2011)

《付 録》

A. 補題 2 の証明

準備のため, 高階偏微分方程式に関する補題を与える。

[補題 4] 添字集合 $\mathcal{I} = \{i_1, i_2, \dots, i_m\} \in \text{pow}(\mathcal{N})$ に対する関数 $V(x)$ の m 階偏微分方程式

$$\partial_{\mathcal{I}} V(x) = 0 \quad (\text{A.1})$$

の一般解は, 任意関数 $W_l: \mathbb{R}^{n-1} \rightarrow \mathbb{R}$, $l = 1, 2, \dots, m$ によって, 以下で与えられる。

$$V(x) = \sum_{l=1}^m W_l((x_k)_{k \in \mathcal{N} \setminus \{i_l\}}) \quad (\text{A.2})$$

証明 帰納法を用いて示す。(A.2) 式が $m = 1$ の場合に成り立つことは明らかである。つぎに $m = p - 1$ に対して題意が成り立つことを仮定し, $m = p$ の場合を考える。 p 個の自然数の集合を $\mathcal{I} = \{i_1, i_2, \dots, i_p\}$ とおく。(A.1) 式より

$$\partial_{\mathcal{I} \setminus \{i_p\}} (\partial_{i_p} V(x)) = \partial_{\mathcal{I}} V(x) = 0$$

が成り立つ。この式を関数 $\partial_{i_p} V(x)$ に対する添字集合 $\mathcal{I} \setminus \{i_p\}$ に関する $p - 1$ 階偏微分方程式と捉える。すると, 帰納法の仮定より, 任意関数 $\bar{W}_l: \mathbb{R}^{n-1} \rightarrow \mathbb{R}$, $l = 1, 2, \dots, p - 1$ に対して以下を得る。

$$\partial_{i_p} V(x) = \sum_{l=1}^{p-1} \bar{W}_l((x_k)_{k \in \mathcal{N} \setminus \{i_l\}})$$

この式を x_{i_p} に関して積分すると以下を得る。

$$\begin{aligned} V(x) &= \int \partial_{i_p} V(x) dx_{i_p} + W_p((x_k)_{k \in \mathcal{N} \setminus \{i_p\}}) \\ &= \sum_{l=1}^{p-1} \int \bar{W}_l((x_k)_{k \in \mathcal{N} \setminus \{i_l\}}) dx_{i_p} + W_p \end{aligned}$$

ここで, $W_p((x_k)_{k \in \mathcal{N} \setminus \{i_p\}})$ は任意関数である。また, $i_l \neq i_p$ より $\int \bar{W}_l((x_k)_{k \in \mathcal{N} \setminus \{i_k\}}) dx_{i_p} = W_l((x_k)_{k \in \mathcal{N} \setminus \{i_l\}})$ と置き換えられ, これらも任意関数である。これより, $m = p$ に対して (A.2) 式を得る。□

補題 2 の証明を行なうため, エージェント数 n に関する帰納法を用いる。まず, $n = 1$ のときを考える。 $x = x_1$ である。もし $\partial_1 V(x) \neq 0$ であれば, $\mathcal{I} = \{1\}$, $W_{\{1\}}(x) = V(x)$, $c = 0$ に対して (10)–(12) 式が成り立つ。 $\partial_1 V(x) \equiv 0$ であれば, 微分方程式を解くと, 積分定数 c に対して $V(x) = c$ を得る。この c と $\mathcal{I} = \{\emptyset\}$ に対して (10)–(12) 式が成り立つ。

つぎに $n \leq p - 1$ に対して題意が成り立つことを仮定し, $n = p$ の場合を考える。 $\mathcal{P} = \{1, 2, \dots, p\}$ とおく。もし $\partial_{\mathcal{P}} V(x) \neq 0$ であれば, $\mathcal{I} = \{\mathcal{P}\}$, $W_{\mathcal{P}}(x) = V(x)$, $c = 0$ に対して (10)–(12) 式が成り立つ。 $\partial_{\mathcal{P}} V(x) \equiv 0$ であれば, 補題 4 より $V(x)$ は適当な関数 $W_l: \mathbb{R}^{p-1} \rightarrow \mathbb{R}$, $l = 1, 2, \dots, p$ に対して以下のように分解される。

$$V(x) = \sum_{l=1}^p W_l((x_k)_{k \in \mathcal{P} \setminus \{l\}}) \quad (\text{A.3})$$

本式を構成する関数のうち $\partial_{\mathcal{P} \setminus \{l\}} W_l((x_k)_{k \in \mathcal{P} \setminus \{l\}}) \neq 0$ であるものは (11) 式を満たす。そうでないものは, さらなる分解操作によって (11) 式を満たすようにすることができる。以下, その分解操作を示す。

(A.3) 式における分解操作の途中経過として, 一般的に, あ

る集合族 $\mathcal{I}_\alpha, \mathcal{I}_\beta \subset \text{pow}(\mathcal{P})$ と関数族 $W_{\mathcal{I}} : \mathbb{R}^{|\mathcal{I}|} \rightarrow \mathbb{R}$ が存在し、以下が得られる。

$$V(x) = \sum_{\mathcal{I} \in \mathcal{I}_\alpha} W_{\mathcal{I}}((x_k)_{k \in \mathcal{I}}) + \sum_{\mathcal{I} \in \mathcal{I}_\beta} W_{\mathcal{I}}((x_k)_{k \in \mathcal{I}}) \quad (\text{A.4})$$

$$\partial_{\mathcal{I}} W_{\mathcal{I}}((x_k)_{k \in \mathcal{I}}) \begin{cases} \neq 0, & \mathcal{I} \in \mathcal{I}_\alpha \\ \equiv 0, & \mathcal{I} \in \mathcal{I}_\beta \end{cases} \quad (\text{A.5})$$

$$\mathcal{I} \notin \mathcal{J}, \forall \mathcal{I}, \mathcal{J} \in \mathcal{I}_\alpha \cup \mathcal{I}_\beta, \mathcal{I} \neq \mathcal{J} \quad (\text{A.6})$$

分解の出発点である (A.3) 式も、各部分集合 $\mathcal{P} \setminus \{l\}$, $l \in \mathcal{P}$ が互いに包含関係にないため、この形で表現される。ここで、集合族 $\mathcal{I}_\beta$ に属する集合に対して、(11) 式が成立していない。以下では、この $\mathcal{I}_\beta$ の要素数をひとつ減らす操作を示す。この操作を繰り返すことで $\mathcal{I}_\beta$ は空集合となり、所望の分解 (10)–(12) 式が得られる。

(A.4) 式において、ひとつの $\mathcal{I}_0 \in \mathcal{I}_\beta$ に着目する。(A.4) 式は (A.3) 式からの分解操作を経て得られているため、いずれかの $l \in \mathcal{P}$ に対して包含関係 $\mathcal{I}_0 \subset \mathcal{P} \setminus \{l\}$ が成り立つ。したがって、 $|\mathcal{I}_0| \leq p - 1$ である。これより、帰納法の仮定から、 $W_{\mathcal{I}_0}((x_k)_{k \in \mathcal{I}_0})$ は (10)–(12) 式を満たすように分解される。すなわち、ある集合族 $\mathcal{I}_0 \subset \text{pow}(\mathcal{I}_0)$ と関数 $W_{\mathcal{I}_0, \mathcal{I}'} : \mathbb{R}^{|\mathcal{I}'|} \rightarrow \mathbb{R}$, 定数 c_0 により、つぎのように分解される。

$$W_{\mathcal{I}_0}((x_k)_{k \in \mathcal{I}_0}) = \sum_{\mathcal{I}' \in \mathcal{I}_0} W_{\mathcal{I}_0, \mathcal{I}'}((x_k)_{k \in \mathcal{I}'}) + c_0 \quad (\text{A.7})$$

$$\partial_{\mathcal{I}'} W_{\mathcal{I}_0, \mathcal{I}'}((x_k)_{k \in \mathcal{I}'}) \neq 0, \mathcal{I}' \in \mathcal{I}_0 \quad (\text{A.8})$$

$$\mathcal{I}' \notin \mathcal{J}', \forall \mathcal{I}', \mathcal{J}' \in \mathcal{I}_0, \mathcal{I}' \neq \mathcal{J}' \quad (\text{A.9})$$

つぎに、(A.7) 式で分解された関数 $W_{\mathcal{I}_0}((x_k)_{k \in \mathcal{I}_0})$ によって、(A.4) 式を再構成する。まず、 $\mathcal{I}' \in \mathcal{I}_\alpha \cup \mathcal{I}_\beta$ に対する関数 $W_{\mathcal{I}_0, \mathcal{I}'}((x_k)_{k \in \mathcal{I}'})$ を (A.4) 式のサメイションの中に入れる。つぎに、 $\mathcal{I}' \notin \mathcal{I}_\alpha \cup \mathcal{I}_\beta$ を満たす要素からなる集合を $\mathcal{I}_\gamma$ とおき、対応する関数 $W_{\mathcal{I}_0, \mathcal{I}'}((x_k)_{k \in \mathcal{I}'}), \mathcal{I}' \in \mathcal{I}_\gamma$ をサメイションの外においておく。以上の関数 $W_{\mathcal{I}_0}((x_k)_{k \in \mathcal{I}_0})$ の分解と分類によって、(A.4) 式の各集合族はつぎのように置き換えられる。

$$\mathcal{I}_\alpha \rightarrow \mathcal{I}_\alpha \cup \mathcal{I}_\gamma, \mathcal{I}_\beta \rightarrow \mathcal{I}_\beta \setminus \{\mathcal{I}_0\}$$

ここで (A.8), (A.9) 式より、それぞれ (A.5), (A.6) 式に相当する式が成り立つ。以上の操作より、 $\mathcal{I}_\beta$ に相当する集合族の要素がひとつ減った。したがって、題意は得られた。

[著者紹介]

桜間 一徳 (正会員)

2004 年京都大学大学院情報学研究科博士後期課程修了。2003 年日本学術振興会特別研究員、2004 年電気通信大学大学院電気通信学研究科助手、2007 年同助教、2011 年京都大学大学院情報学研究科特定研究員、2011 年鳥取大学大学院工学研究科准教授となり、現在に至る。博士 (情報学)。マルチエージェントシステム、非線形システムの制御に関する研究に従事。システム制御情報学会、日本ロボット学会、日本機械学会の会員。

東 俊一 (正会員)

2004 年東京工業大学大学院情報理工学研究科博士後期課程修了。同年日本学術振興会特別研究員 (PD)、2005 年京都大学大学院情報学研究科助手、2007 年同助教、2011 年同准教授となり現在に至る。この間、2004 年日本学術振興会特別研究員 (DC2)、2004~2005 年ジョージア工科大学客員研究員、2009~2010 年ペンシルバニア大学客員研究員。ハイブリッドシステム制御論の研究に従事。博士 (工学)。2005、2008 年計測自動制御学会論文賞、2011 年度同学会制御部門バイオニア賞などを受賞。IEEE、システム制御情報学会などの会員。

杉江 俊治 (正会員・フェロー)

1978 年京都大学大学院修士課程精密工学専攻修了。84 年同博士後期課程研究指導認定退学。78~80 年日本電信電話公社勤務。84 年大阪府立大学工学部助手。88 年京都大学工学部助教授を経て、現在情報学研究科教授。ロバスト制御、システム同定、非線形制御などの研究に従事。工学博士。91 年システム制御情報学会榎木記念論文賞。98、2008 年同学会論文賞。94、2000、2003、2007、2008 年計測自動制御学会論文賞を受賞。システム制御情報学会、日本ロボット学会などの会員。IEEE のフェロー。