

移動ロボット群における連結性

山本, 薫
九州大学大学院システム情報科学研究所

<https://hdl.handle.net/2324/7170249>

出版情報 : SYSTEMS, CONTROL AND INFORMATION. 67 (1), pp.12-17, 2023. THE INSTITUTE OF SYSTEMS, CONTROL AND INFORMATION ENGINEERS (ISCIE)

バージョン :

権利関係 : © 2023 一般社団法人 システム制御情報学会



移動ロボット群における連結性

山本 薫*

1. はじめに

移動ロボット群制御は、マルチエージェントシステム (MAS) の制御の応用例として、代表的なものの一つである。ほかの MAS 同様に、情報伝達構造をグラフによりモデル化し、グラフ理論の知見を用いた解析や制御系設計に関するさまざまな研究が行われている。移動ロボット群における大きな特徴は、ロボット間の情報伝達の可否が距離ベースで決まり、移動の際に情報伝達構造が変化する可能性があることである。本稿では、移動ロボット群において、連結性を維持しながら制御する手法について、いくつかの重要な結果と、筆者らの最近の取り組みを紹介する。

2. グラフによるロボットネットワークの表現と距離ベースの連結性

本稿では、最も単純なロボットモデルである、質点ロボットモデルを扱う。つまり、 d 次元空間を移動するロボット i の時刻 $t \geq 0$ における位置 $x_i(t) \in \mathbb{R}^d$ は、時刻 t におけるロボット i への制御入力 $u_i(t) \in \mathbb{R}^d$ を用いて、以下のような積分系

$$\dot{x}_i(t) = u_i(t)$$

あるいは、二重積分系

$$\dot{x}_i(t) = v_i(t)$$

$$\dot{v}_i(t) = u_i(t)$$

で表される。ここで、 $v_i(t)$ はロボット i の時刻 $t \geq 0$ における速度である。

各ロボットは、通信やセンシングにより、周辺のロボットの状態を取得できるものとする。情報を取得する範囲としては、第 1 図に示すように、ロボットを中心として全方位に広がるもののほか、指向性のあるもの考えることも多い。これは、使用するセンサの特性により情報取得可能な範囲が物理的に制限される場合のほか、進行方向にいるロボットの状態をより重視したい場合など、制御系設計の方針によって決まる場合も含む [10]。

各ロボットが他のどのロボットから影響を受けるかを表すために、グラフの概念が一般に用いられる。ロボット i は頂点 i として表され、ロボット i がロボット j の状態を取得することを、頂点 j から頂点 i への矢印で図示し、辺とよばれる順序対 (j, i) で定義する^{1,2}。

以上より、 n 台のロボットで構成される群は、頂点集合 $V := \{1, 2, \dots, n\}$ と、群内部での情報の流れを表す辺集合 $E \subseteq V \times V$ により定義されるグラフ $G = (V, E)$ で表現される。さらに、ロボット i がロボット j の状態を取得するとき、 j は i の neighbour であるといい、そのようなロボットの集合を $N_i(t) = \{j \in V \mid (j, i) \in E\}$ と表す。移動ロボット群を分散制御する場合、ロボット i は、自身の状態とロボット $j \in N_i(t)$ の状態のみを制御に用いる。

辺 (j, i) にはロボット間距離 $d_{ij}(t) := \|x_i(t) - x_j(t)\|$ に応じたつぎのような重みを関連づけることも多い：

$$w_{ij}(t) = f(d_{ij}(t)) \quad (1)$$

文献 [15] では、関数 f として、ロボット間距離に対する減少関数のうちつぎのような性質を満たすものを挙げている。

$$\begin{cases} 1 - \epsilon < f(d_{ij}(t)) \leq 1 & (d_{ij}(t) < \rho_1 \text{ のとき}) \\ 0 \leq f(d_{ij}(t)) < \epsilon & (d_{ij}(t) > \rho_2 \text{ のとき}) \end{cases}$$



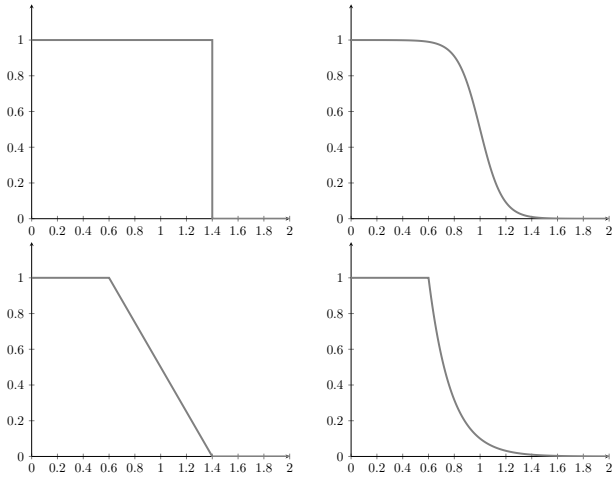
第 1 図 ロボットの情報取得範囲。2次元の場合は、ロボットを中心とした円（左）や、進行方向に扇状に広がる範囲（右）を考えることが多い。

¹本稿では情報の流れの向きを矢印の向きとしているが、「ロボット i がロボット j をセンシングする」という意味で、 i から j への矢印として定義する文献も多く、注意が必要である。

²辺 (edge) は矢印が双方向である場合のみを指すことも多い。このとき、辺は異なるノード i, j の非順序対 $\{i, j\}$ として定義される。これに対し、方向をもつ場合は弧 (arc) とよばれる。

* 九州大学 大学院 システム情報科学研究院

Key Words: mobile robot networks, connectivity, formation control.



第2図 関数 f の例. 左上: $d \leq \rho_2$ のとき $f(d) = 1$, それ以外で $f(d) = 0$. 右上: $f(d) = 1/(1 + e^{-\alpha(d-\rho)})$, $\alpha = (2/(\rho_2 - \rho_1)) \log((1-\epsilon)/\epsilon)$, $\rho = (\rho_1 + \rho_2)/2$. 左下: $d < \rho_1$ で $f(d) = 1$, $\rho_1 \leq d < \rho_2$ で $f(d) = d/(\rho_1 - \rho_2) - \rho_2/(\rho_1 - \rho_2)$, $\rho_2 \leq d$ で $f(d) = 0$. 右下: $d < \rho_1$ で $f(d) = 1$, $d \geq \rho_1$ で $f(d) = e^{-\alpha(d-\rho_1)}$, $\alpha = (1/(\rho_2 - \rho_1)) \log(1/\epsilon)$. 図では $\rho_1 = 0.6, \rho_2 = 1.4, \epsilon = 0.01$ としている. 文献 [15] より.

ただし, $0 < \rho_1 < \rho_2$ であり, ϵ は $0 < \epsilon < 1$ を満たす十分小さい定数とする. このような性質を満たすいくつかの関数を第2図に示す. なお, 文献 [15] においては, この定義の説明として, ロボット i, j が無線通信により情報共有をする場合に, シグナルが距離 ρ_1 までは十分に強いが, そこから急激に弱まり, 距離 ρ_2 でほぼ通信が途切れてしまうという特性を表している, としているが, 取得する情報の重要度を表している, というようにも解釈できるだろう.

このとき, 重み付きグラフ G^w のグラフラプラシアンをつぎのように定義する:

$$[L(t)]_{ij} = \begin{cases} \sum_{k \neq i} w_{ik}(t) & (i=j) \\ -w_{ij}(t) & (i \neq j) \end{cases} \quad (2)$$

ただし, $(j, i) \notin E$ のとき, $w_{ij} = 0$ とする. L の任意の行の和が0となることから, グラフラプラシアン L が固有値0を少なくとも一つもち, $\mathbf{1} = [1, \dots, 1]^T$ が対応する固有ベクトルとなることがわかる. さらに, ゲルシュゴリンの定理より, L の0以外の固有値はすべて開右半平面に存在することが容易に示される. とくに, 任意の $i, j \in V(G^w)$ において $w_{ij}(t) = w_{ji}(t)$ が成り立つとき, $L \in \mathbb{R}^{n \times n}$ は実対称行列となり, 固有値は $0 = \lambda_1(L(t)) \leq \lambda_2(L(t)) \leq \dots \leq \lambda_n(L(t))$ と書ける. このとき, グラフ G^w が連結であるための必要十分条件は, $\lambda_2(L(t)) > 0$ であることが知られている [5]. この2番目に小さい固有値 $\lambda_2(L(t))$ は, グラフの連結性のほか, 頂点や辺の欠落に対するロバスト性 [5,6] や合意アルゴリズムの収束速度 [11–13] を表す重要な指標であり, 代数

的連結度, あるいはFiedler値ともよばれる [4].

以降では, 重みが対称, すなわち, 任意の $i, j \in V(G^w)$ において $w_{ij}(t) = w_{ji}(t)$ が成り立つ場合を考える. 移動ロボット群の分散制御においては, もともと連結なネットワークであったとしても, 移動途中に連結性が失われる可能性がある. 基本的な問題として, 局所的な情報交換のみに基づきロボットを1点に集合させる「ランデブー問題」を挙げる. グラフの構造が時不変である場合, 初期状態において連結であれば, 線形分散制御器

$$\dot{x}_i = - \sum_{j \in N_i} (x_i - x_j)$$

はランデブーを達成する. しかし, ロボット間距離に応じた重みを考える場合, 上記の線形分散制御器では, 初期状態で連結であったとしても, 収束までの過程において連結性は保証されないことが知られている [7].

そこで, 移動途中の連結性を保証するためのさまざまな手法が提案されている. 次節以降では, 代表的なものをいくつか紹介する.

3. バイナリ制御信号と人工ポテンシャル法に基づく方法

積分系を考える. このとき, すでに連結しているロボット間の連結性を保つような分散制御器として, つぎのようなものを用いることが考えられる.

$$u_i(t) = - \sum_{j \in N_i(t)} \nabla_{x_i} g_{ij}(t) \quad (3)$$

$$g_{ij}(t) := \frac{1}{\rho_2^2 - d_{ij}^2}$$

関数 g_{ij} は人工ポテンシャル関数であり, 初期状態でロボット i, j 間の距離が ρ_2 未満であれば, 以後も ρ_2 未満であり続けるよう強制するような制御となっていることがわかる.

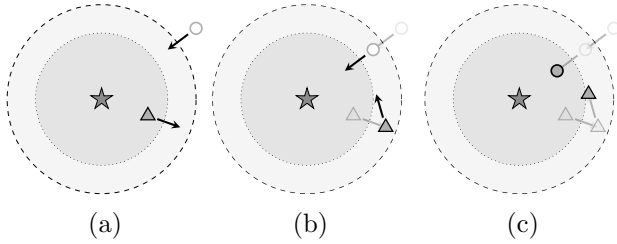
しかし, この場合, 初期状態で ρ_2 以上離れている場合は, 近づくことができず, 新たにつながりを持つことができない. この問題を解消するため, バイナリ制御信号 $\sigma_{ij} \in \{0, 1\}$ を導入し,

$$\begin{cases} j \in N_i(t) & (\sigma_{ij}(t) = 1 \text{ のとき}) \\ j \notin N_i(t) & (\sigma_{ij}(t) = 0 \text{ のとき}) \end{cases}$$

と定義して, 近づく場合と遠ざかる場合とを区別することを考える. 文献 [7] では, つぎのようなバイナリ制御信号を導入している.

$$\sigma_{ij}(t) = \begin{cases} 0 & (\sigma_{ij}(t^-) = 0 \text{ かつ } d_{ij} \geq \rho_1 \text{ のとき}) \\ 1 & (\text{それ以外の場合}) \end{cases} \quad (4)$$

ここで, $0 < \rho_1 < \rho_2$ であり, t^- は t の直前の時刻を表す. ロボット i からの距離が $\rho_1 < d < \rho_2$ のロボットにつ



第 3 図 (4) 式に示すバイナリ制御信号の作用の一例を示す。ここでは、時間の経過とともに (a)→(b)→(c) と状況が移行する様子を表している。星印のロボット★とその他のロボット(丸印○と三角印△)の連結性について考える。ロボット○とロボット△について、有色はロボット★と連結している状態を表し、無色は連結していない状態を表す。ロボット★を中心に、半径 ρ_1 の内側の円と、半径 ρ_2 の外側の円を考える。ロボット○は、(a) の時点でロボット★と連結していないため、 $\rho_1 < d_{*o} < \rho_2$ の領域では制御がはたらかず、(b) に示すように、ロボット○はロボット★に近づくことができる。(c) のように $d_{*o} \leq \rho_1$ となった時点で (○, ★) 間に辺が形成され、制御力 (3) が作用するようになる。一方、ロボット△は (a) の時点で連結しているため、ロボット★から遠ざかる方向に進み (b) のように $\rho_1 < d_{*△} < \rho_2$ となったとしても、辺は残ったままであり、(c) に示すように、制御力 (3) によりロボット★側に引き戻されることになる。

いて、たとえば、遠くから近づいてくるなどでまだ連結していない場合には、近づくことを妨げず、距離が ρ_1 以下となった時点ではじめて辺を形成し、連結させるようになっている。一方、すでに連結しているロボットに関しては、距離 $\rho_1 < d < \rho_2$ においても制御入力 (3) が作用し、連結性を維持する。第 3 図にその様子を示す。

4. 最適化に基づく方法

つぎに最適化に基づく手法を紹介する。

基本的なアイデアは、 $\lambda_2(L(x))$ を最大化するロボットの配置を求めることである。すなわち、

$$\max_x \lambda_2(L(x)) \quad (5)$$

と書ける。ただし、 $x = [x_1^\top, x_2^\top, \dots, x_n^\top]^\top \in \mathbb{R}^{dn}$ 。

ここで、 L が対称であることから、固有ベクトルが直交することと、 L が常に固有値 0 に対応する固有ベクトルとして $\mathbf{1}$ をもつことに注意すると、 $\lambda_2(L)$ は以下のように表すことができる。

$$\lambda_2(L) = \inf_{z \in \mathbf{1}^\perp} \frac{z^\top L z}{z^\top z} \quad (6)$$

ただし、 $\mathbf{1}^\perp$ は直交補空間 $\mathbf{1}^\perp := \{z \in \mathbb{R}^n \mid \mathbf{1}^\top z = 0\}$ を表すものとする。これより、任意の $\alpha \in (0, 1)$ に対し、

$$\lambda_2(\alpha L + (1 - \alpha)\tilde{L}) = \inf_{z \in \mathbf{1}^\perp} \frac{z^\top (\alpha L + (1 - \alpha)\tilde{L}) z}{z^\top z}$$

$$\begin{aligned} &\geq \alpha \inf_{z \in \mathbf{1}^\perp} \frac{z^\top L z}{z^\top z} + (1 - \alpha) \inf_{z \in \mathbf{1}^\perp} \frac{z^\top \tilde{L} z}{z^\top z} \\ &= \alpha \lambda_2(L) + (1 - \alpha) \lambda_2(\tilde{L}) \end{aligned}$$

が成り立ち、 $\lambda_2(L)$ は空間 $\mathbf{1}^\perp$ において凹関数であることがわかる。

この性質を利用し、等価な凸最適化問題を解く集中アルゴリズムや分散アルゴリズムが提案されている。ここで、 $\lambda_2(L)$ が L に関する凹関数であっても、問題 (5) は状態 $x \in \mathbb{R}^{dn}$ に依存するため非凸であることに注意が必要である。このため、反復法による求解が必要となる。以下に、集中アルゴリズムによる解法と、分散アルゴリズムによる解法を紹介する。

集中アルゴリズム

まず、(6) 式より、つぎの関係が成り立つことがわかる。

$$\lambda_2(L) > 0 \iff P^\top L P \succ 0$$

ただし、 P は部分空間 $\mathbf{1}^\perp$ を張るベクトルで構成される行列である。すなわち、

$$\begin{aligned} P &:= [p_1, p_2, \dots, p_{n-1}], \\ p_i^\top \mathbf{1} &= 0 \quad \forall i \in \{1, 2, \dots, n-1\}, \\ p_i^\top p_j &= 0 \quad \forall i \neq j \end{aligned}$$

これを用いれば、問題 (5) はつぎのように書き直すことができる。

$$\begin{aligned} &\max_{x \in \mathbb{R}^{dn}} \gamma \\ &\text{s.t.} \quad P^\top L(x) P \succeq \gamma I_{n-1} \end{aligned}$$

文献 [9] では、ここにさらにロボット同士の衝突回避条件 $d_{ij} := \|x_i - x_j\|^2 \geq \rho_1$ ($i \neq j$) を制約に加えた問題を、以下のような反復アルゴリズムにより解く手法を提案している。

$$\begin{aligned} &\max_{x(k+1) \in \mathbb{R}^{dn}} \gamma \\ &\text{s.t.} \quad P^\top L(x(k+1)) P \succeq \gamma I_{n-1}, \\ &\quad 2(x_i(k+1) - x_j(k+1))^\top (x_i(k) - x_j(k)) \\ &\quad = d_{ij}(k+1) + d_{ij}(k), \\ &\quad d_{ij}(k+1) \geq \rho_1 \end{aligned}$$

ただし、 $i < j$ 。つまり、 $\lambda_2(L(x(k)))$ を各ステップで最大化することを目標としている。なお、2 番目と 3 番目の制約条件は、 $d_{ij} := \|x_i - x_j\|^2 \geq \rho_1$ ($i \neq j$) の線形近似であるため、ロボットの位置とロボット間距離に整合性がない可能性があるが、さらに追加の制約を加えることで整合性が保証されることが文献 [9] に述べられている。

分散アルゴリズム

ここでは、文献 [3] で提案されている、supergradient を用いて問題 (5) を分散的に解く手法を紹介する。二つ

の異なるグラフラプラシアン $L \neq \tilde{L}$ と、 $\lambda_2(L)$ に対応する単位固有ベクトル $z \in \mathbb{1}^\perp$ について、

$$\begin{aligned} \lambda_2(\tilde{L}) z^\top z &\leq z^\top \tilde{L} z \\ &= z^\top L z + z^\top (\tilde{L} - L) z \\ &= \lambda_2(L) + \langle z z^\top, (\tilde{L} - L) \rangle \end{aligned}$$

が成り立つことから、行列 $G := z z^\top$ は $\lambda_2(L)$ の supergradient¹ となっていることがわかる。したがって、反復計算

$$L_{k+1}^* = L_k^* + \alpha_k G_k \quad (7)$$

において、ステップ幅 α_k を適切に定めることで、最適解 L^* に収束する。なお、ステップ幅 α_k はつぎの二つの条件を満たすものであればよいことが知られている [1,2,16] (square summable but not summable series):

$$\sum_{k=0}^{\infty} \alpha_k = \infty, \quad \sum_{k=0}^{\infty} \alpha_k^2 < \infty$$

これを満たすものとして、たとえば、 $\alpha_k = 1/(k+1)$ が挙げられる。上述の supergradient 法に、主分解あるいは双対分解と組み合わせることで、分散アルゴリズムを構築することが可能である [1,2,16]。ただし、このためには、ロボット i は各ステップで、 L のすべての固有ベクトルの第 i 要素、および近傍のエージェント $j \in N_i$ に対応する第 j 要素を知っている必要がある。これが通信などで達成できる場合は、Decentralized Orthogonal Iteration Algorithm[8] 等を用いることで、分散的に L の固有ベクトルを計算することが可能である。

このような前提のもと、文献 [3] では、(7) 式で得られる各ステップでの L_k^* とロボットの配置で決まる実際のグラフラプラシアン $L(x)$ の誤差を小さくするような制御則として、ポテンシャル関数を用いた以下のような分散制御器が提案されている。

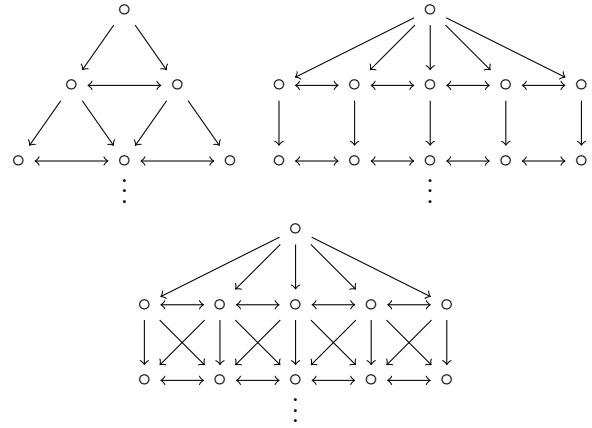
$$\begin{aligned} u_i(t) &= - \sum_{j \in N_i} \nabla_{x_i} V_{ij}(t) \\ V_{ij}(t) &= \begin{cases} (d_{ij}^2 - [L_k^*]_{ij}^{-1})^2 & (d_{ij} \leq \rho_2 \text{ のとき}) \\ (\rho_2 - [L_k^*]_{ij}^{-1})^2 & (d_{ij} > \rho_2 \text{ のとき}) \end{cases} \end{aligned}$$

ただし、 $[L_k^*]_{ij}^{-1}$ は L_k^* の (i,j) 要素の逆写像であり、(1)、(2) 式より $[L_k^*]_{ij} = -f(d_{ij}^*)$ であることから、ロボット i, j 間の最適距離と捉えることができる。

5. 階層型パスグラフにおけるロボットの群離脱と代数的連結度

最後に、移動ロボット群における代数的連結度に関する筆者らの最近の取り組みを紹介する。

¹subgradient は劣勾配と訳されるが、supergradient の (広く採用されている) 和訳はないようである。



第 4 図 階層構造を持つ編隊移動フォーメーション

ロボット群の編隊移動として、第 4 図のような、1 台のリーダーを先頭として階層的に構成されたロボットのフォーメーションを考える。図の矢印の向きは情報の流れの向きを表し、各層はパスグラフ（葉の数がちょうど 2 である無向木）、あるいはパスグラフの非交和²であるとする。このようなグラフを「階層型パスグラフ」とよぶこととし、あるロボットが群を離脱した際に、代数的連結度がどのように変化するかを調べる。ここで、ロボットの離脱は、対応する頂点および辺の除去を意味する。

以下に、階層型グラフと階層型パスグラフの定義を述べる [14]。

【定義 1】 (階層型グラフ) 有向グラフ G の頂点集合 $V(G)$ が頂点集合 $\{V(G_k)\}_{1 \leq k \leq m}$ の非交和であり、かつ、辺集合 $E(G)$ に属する任意の辺 e において、 $e \in E(G_k)$ または $e \in V(G_k) \times V(G_{k+1})$ となるような k が存在するとき、 G は層 $(G_k)_{1 \leq k \leq m}$ をもつ階層型グラフであるという。

【定義 2】 (階層型パスグラフ) 以下の性質を満たす階層型グラフを層 $(G_k)_{1 \leq k \leq m}$ をもつ階層型パスグラフという。

- (1) $V(G_1)$ は単集合である。
- (2) 各 G_k はパスグラフの非交和である。

さらに、層 G_k 内の頂点 i について、from-above degree $d^\downarrow(i)$ をつぎのように定義する。

$$\begin{aligned} d^\downarrow(i) &:= \#N^\downarrow(v) \\ N^\downarrow(i) &:= \{j \in V(G_{k-1}) \mid (j, i) \in E(G)\} \end{aligned}$$

ただし、 $\#A$ は集合 A の濃度を表す。すなわち、 $N^\downarrow(i)$ はエージェント i の neighbour のうち上の層に属するものの集合を、 $d^\downarrow(i)$ はその要素数を表す。このとき、代数的連結度に関し、以下の結果が得られる（証明は文献 [14] を参照されたい）。

²有向グラフ G がグラフ $\{G_k\}_{1 \leq k \leq m}$ の非交和であるとは、頂点集合 $V(G)$ と辺集合 $E(G)$ がそれぞれ頂点集合 $\{V(G_k)\}_{1 \leq k \leq m}$ と辺集合 $\{E(G_k)\}_{1 \leq k \leq m}$ の非交和であることを意味する。

【命題 1】 任意の階層型パスグラフ G において,

$$0 \leq \lambda_2(L_G) \leq 1$$

【定理 1】 層 $(G_k)_{1 \leq k \leq m}$ をもつ階層型パスグラフ G が, $k=1, \dots, n$ において以下の性質を満たすものとする.

$$\forall (i, j) \in E(G_k), \#(N^\downarrow(i) \setminus N^\downarrow(j)) \leq 1 \quad (8)$$

このとき, つぎの (i), (ii) は等価である.

- (i) $0 \leq \lambda_2(L_G) < 1$;
- (ii) $d^\downarrow(i) = 0$ となるような $i \in V(G) \setminus V(G_1)$ が存在する.

条件 (8) は, ロボット群の編隊移動でよくみられるフォーメーションにおいては自然に満たされる場合が多い. さらに, 先頭のリーダーが離脱しない限り, このような階層型パスグラフの誘導部分グラフもまたこの条件を満たす階層型パスグラフとなる. つまり, もとの階層型パスグラフが条件 (8) を満たす場合, G_1 に属する頂点以外のどの頂点を取り除いても, 条件 (8) を満たす階層型パスグラフとなる.

命題 1 と定理 1 より, 条件 (8) を満たす任意の階層型パスグラフにおいて, ある頂点を取り除いたときに, 残りのすべての頂点において, 少なくとも 1 本の辺が上の層から入っている場合, 代数的連結度 λ_2 は最大値 1 をとり, それ以外の場合にはすべて悪化することがわかる. 第 5 図に数値例を示す.

フォーメーション制御

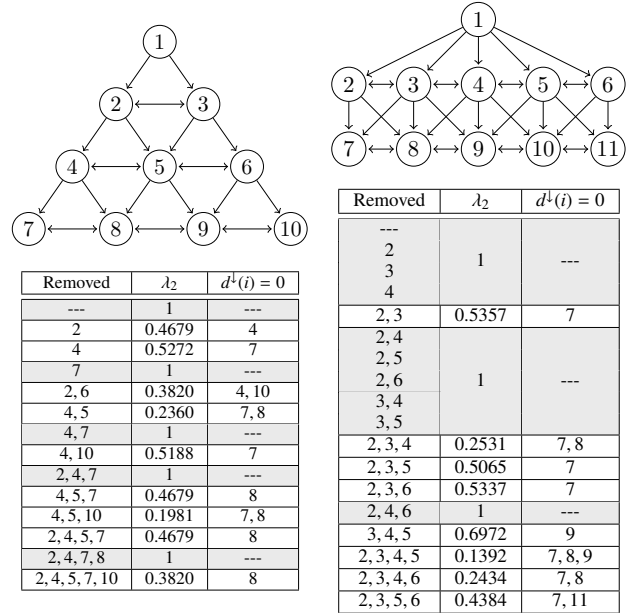
ロボット i のダイナミクスがつぎのように表される, 基本的な合意プロトコルによるフォーメーション制御を考える.

$$\begin{aligned} \dot{x}_i(t) &= v_i \\ \dot{v}_i(t) &= - \sum_{j=1}^n [L]_{ij} [(p_i - p_j) + (v_i - v_j)] \quad i = 1, \dots, n \end{aligned}$$

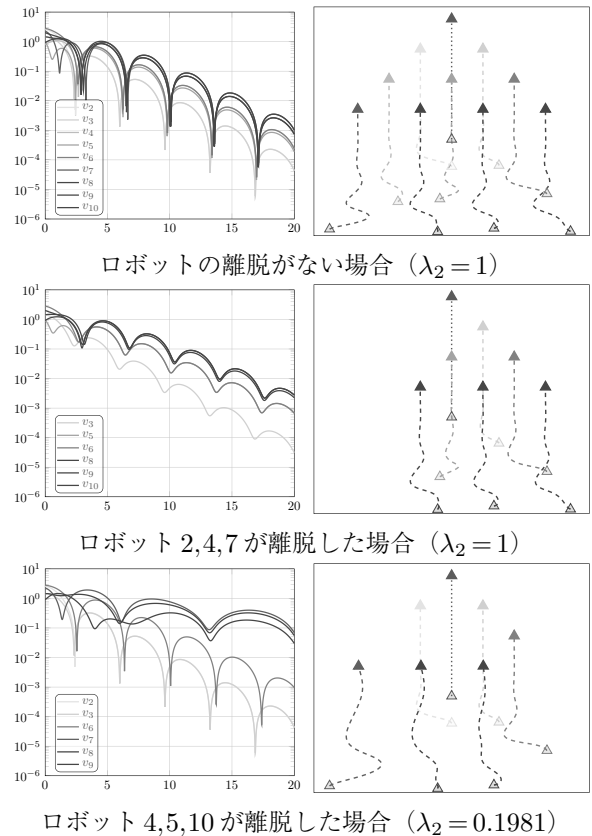
ここで, n はロボットの数, $x_i \in \mathbb{R}^2, v_i \in \mathbb{R}^2$ はそれぞれ 2 次元空間におけるロボット i の位置および速度とする. ただし, 目標フォーメーションにおける位置をロボット i の原点とする. 第 5 図の左に示すような三角形の階層型パスグラフを目標フォーメーションとする編隊移動で, ロボットの離脱の影響がどのように現れるかを第 6 図に示す. $v_1 = (0, 1)$ m/s とし, ある初期状態から目標フォーメーションに収束するまでを図示している. 中段および下段においては, どちらも 3 台のロボットが離脱したネットワークであるが, 中段のケースでは目標フォーメーションに速やかに収束しているのに比べ, 下段の場合では収束に時間がかかっていることがわかる.

6. おわりに

本稿では, 移動ロボット群における連結性を維持するためのいくつかの制御手法について紹介した. 移動ロ



第 5 図 階層型パスグラフにおける頂点の除去と代数的連結度 λ_2 の関係



第 6 図 左: 目標位置からの偏差の時間推移, 右: 各ロボットの軌跡.

ット群における連結性に関する研究は非常に活発に行われており, それらを網羅的かつ俯瞰的に述べることは, 紙面の都合以上に筆者の力量に余るため, 内容を限定し, そのベースとなるアイデアに焦点をあてた解説を試みた. 移動ロボット群制御は, さまざまな分野の考え方が

導入された，非常に興味深く魅力的な分野であると筆者は感じている．本稿が本分野に興味をもつ読者の一助となれば幸いである．

謝 辞

本稿をまとめるための調査研究の一部は JSPS 科研費 JP20K14766 の助成を受けている．

(2022 年 7 月 11 日受付)

参 考 文 献

- [1] D. Bertsekas: *Nonlinear Programming*, Athena Scientific, 3rd edition (2016)
- [2] S. Boyd, L. Xiao and A. Mutapcic: Subgradient methods, lecture notes of EE392o, Stanford University, Autumn Quarter; http://www.stanford.edu/class/ee392o/subgrad_method.pdf (2003, Accessed: 2022-6-20)
- [3] M. C. De Gennaro and A. Jadbabaie: Decentralized control of connectivity for multi-agent systems; *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 3628–3633 (2006)
- [4] M. Fiedler: Algebraic connectivity of graphs; *Czechoslovak Mathematical Journal*, Vol. 23, No. 2, pp. 298–305 (1973)
- [5] C. Godsil and G. F. Royle: *Algebraic Graph Theory*, Vol. 207, Springer Science & Business Media (2001)
- [6] A. Jamakovic and S. Uhlig: On the relationship between the algebraic connectivity and graph's robustness to node and link failures; *2007 Next Generation Internet Networks*, pp. 96–102 (2007)
- [7] M. Ji and M. Egerstedt: Distributed coordination control of multiagent systems while preserving connectedness; *IEEE Transactions on Robotics*, Vol. 23, No. 4, pp. 693–703 (2007)
- [8] D. Kempe and F. McSherry: A decentralized algorithm for spectral analysis; *Journal of Computer and System Sciences*, Vol. 74, No. 1, pp. 70–83 (2008)
- [9] Y. Kim and M. Mesbahi: On maximizing the second smallest eigenvalue of a state-dependent graph Laplacian; *Proceedings of the 2005 American Control Conference*, pp. 99–103 (2005)
- [10] A. Nag, S. Huang, A. Themelis and K. Yamamoto: Flock navigation with dynamic hierarchy and subjective weights using nonlinear MPC; *2022 IEEE Conference on Control Technology and Applications (CCTA)* (To appear)
- [11] R. Olfati-Saber, J. A. Fax and R. M. Murray: Consensus and cooperation in networked multi-agent systems; *Proceedings of the IEEE*, Vol. 95, No. 1, pp. 215–233 (2007)
- [12] R. Olfati-Saber and R. M. Murray: Consensus problems in networks of agents with switching topology and time-delays; *IEEE Transactions on Automatic Control*, Vol. 49, No. 9, pp. 1520–1533 (2004)
- [13] A. Olshevsky and J. N. Tsitsiklis: Convergence speed in distributed consensus and averaging; *SIAM Journal on Control and Optimization*, Vol. 48, No. 1, pp. 33–55 (2009)
- [14] R. Yoshise and K. Yamamoto: Algebraic connectivity of layered path graphs under node deletion; *2022 61st IEEE Conference on Decision and Control (CDC)* (To appear)
- [15] M. M. Zavlanos, M. B. Egerstedt and G. J. Pappas: Graph-theoretic connectivity control of mobile robot networks; *Proceedings of the IEEE*, Vol. 99, No. 9, pp. 1525–1540 (2011)
- [16] 東, 永原, 石井, 林, 桜間, 畑中: マルチエージェントシステムの制御, コロナ社 (2015)

著 者 略 歴

やま もと 薫
山 本



2009 年京都大学工学部建築学科卒業, 2016 年 1 月 University of Cambridge, Department of Engineering, Information Engineering, Control Group PhD 課程修了. University of Minnesota, Twin Cities および Lund University でのポスドクを経て, 2018 年 8 月九州大学大学院システム情報科学研究院准教授となり現在に至る. 自律移動ロボット群制御や機械系の振動制御などの研究に従事. IEEE, 計測自動制御学会などの会員.