自己釣り合い状態を利用したマルチエージェントシステムの フォーメーション制御

増 田 容 一*・長 瀬 賢 二**

Formation Control for Multi-agent Systems by Using Self-equilibrium Condition

Yoichi MASUDA* and Kenji NAGASE**

This paper proposes a cooperative formation control of multi-agent systems by using self-equilibrium condition based on the force-density method. The force-density method is widely used in structural mechanics, which enables us to design the equilibrium of axially loaded structures (tensegrity structures) with non-zero control force. The control law is derived from a potential function of linear springs connecting the agents. Replacing the design variables in linear springs by force density (control force per unit length) and member constant (product of spring constant and rest length), the design problems can be described as convex optimization problems. As an application of the proposed strategy, a control design method for multi-agent systems with communication time-delays is provided. Permissible time-delay is maximized by minimizing the maximum eigen value of the tangent rigidity matrix. Numerical examples show that the permissible time-delay is increased by introducing the non-zero internal force.

Key Words: multi-agent system, formation control, self-equilibrium condition, tensegrity structure, time-delay system

1. はじめに

マルチエージェントシステムに対する協調制御の研究は、その応用分野の広さからさまざまな視点からの研究がなされてきた¹⁾。中でも近年、フォーメーション制御に関する研究が活発に行なわれており、航空宇宙分野やロボティクス分野への応用が期待されている^{2), 3)}。

従来提案されているフォーメーション制御の手法は、コントローラの構造により、合意ベース手法と人工ポテンシャル法の二種類に大別できる。合意ベース手法は、エージェントの接続関係（結合行列）により定まるグラフラプリアンをフィードバックゲインに用いる手法である^{4), 5)}。この手法は、コントローラの形がグラフの結合関係のみから定まるため単純で扱いやすい反面、設計の自由度が少なく安定性以外の系の収束特性を陽に指定するのが難しい。

一方、人工ポテンシャル法は、人工的なポテンシャル関数を設定し、その負の勾配を入力として利用することで、ポテン

シャル関数が極小値をとる位置に各エージェントを導く制御手法である^{6), 7)}。特にフォーメーション制御においては、各エージェント間の距離に応じたポテンシャルを設定することが多い。この手法の利点として、設定するポテンシャル関数の形状によって、各エージェントの接近時や離反時における、距離に応じた収束特性を設計できることがある。これを利用することで、エージェント同士の衝突回避^{8), 9)}や、センサ性能や通信可能距離の考慮¹⁰⁾などが可能となる。また、エージェントの接続関係が動的に変化する場合に対しても、グラフラプリアンや、対称剛性行列の固有値に関するポテンシャル関数を設定する手法^{11)~13)}を用いることで、接続関係の自動選択が可能となるなど、有用な応用が多い。

人工ポテンシャル法におけるポテンシャル関数は、通常、目標のフォーメーション形状において、その総和の勾配が零となるよう設定される。このようなポテンシャルの設計手法としては、各エージェント間に設定される（和をとる前の）ポテンシャルの勾配自体が、目標形状でそれぞれ零となるよう設計する方法が簡易であり、広く用いられている。この場合、平衡状態において、個々のポテンシャルから生じる制御力も零となる。一方、容易に想像できるように、ポテンシャル関数の選び方によっては、個々のポテンシャルの勾配から計算される非零の制御力が互いに釣り合うことで平衡状態を生み出すことも可能である。このように、非零の要素間力を利用する構造は、構造力学の分野において「テンセグリティ構造」と

* 和歌山大学大学院システム工学研究科 和歌山市栄谷 930

** 和歌山大学システム工学部 和歌山市栄谷 930

* Graduate School of Systems Engineering, Wakayama University, 930 Sakaedani, Wakayama

** Department of Systems Engineering, Wakayama University, 930 Sakaedani, Wakayama

(Received July 3, 2015)

(Revised July 30, 2015)

呼ばれ、多くの研究^{14),15)}がなされてきた。本研究では、この分野にならない、非零の要素間力を有する平衡状態を自己釣り合い状態、また、このとき系の内部で釣り合う力を自己釣り合い力と呼ぶ。文献16)~20)では自己釣り合い力が系の安定性や剛性に影響を与えることが指摘されており、フォーメーション制御においても、自己釣り合い力を許容した、より広いクラスでコントローラを探索することで、制御性能の向上が期待できる。これまでに自己釣り合い状態を考慮したフォーメーション制御則^{21)~23)}はいくつか提案されているが、これら設計法は、自己釣り合い力の設計自由度の一部しか利用しておらず、また、自己釣り合い力がシステム特性に与える影響も考慮されていない。特に、自己釣り合い力を許容することによる設計自由度を制御性能の向上のために積極利用した例は著者らの知る限り存在しない。

本研究では、新たに自己釣り合い力の設計自由度を導入・利用したフォーメーション制御系の設計法について考える。基本的な制御系設計の枠組みは、従来研究と同様とする。すなわち、制御入力は、エージェント間の距離を変数としたポテンシャル関数の負の勾配として与え、ポテンシャル関数の生成要素には、最も基本的な線形ばねを考える。このとき、設計対象の制御パラメータは、ばね定数および自然長となる。上記の制御パラメータはモデル中に積の形で存在するため、これらを直接設計に用いるのは好ましくない。そこで本研究では、軸力密度法と呼ばれる変数変換法を導入し、制御パラメータを単位長さあたりの制御力（軸力密度）、ならびに、ばね定数と部材長さの積（部材特性）に置き換える。この変換により、モデルがパラメータに関して線形となり、以降の解析・設計が容易となる。上記システム表現を用いると、安定な自己釣り合い状態の設計問題は、軸力密度と部材特性に関する凸最適化問題として定式化できる。目的関数の選択にはさまざまなものが考えられるが、本研究ではその一例として、エージェント間の通信に遅延が生じる状況を考え、その許容むだ時間を最大化する問題を考える。上記最適化問題は半正定値計画問題（SDP: Semidefinite Programming）となり、主双対内点法などを用いて、数値的に安定に最適解を求めることができる。最適解が一意に定まる凸最適化問題として定式化したことで、自己釣り合い力の有無に対する解の比較・有効性評価が可能となる。

本論文の構成は、以下のとおりである。第2章では、制御問題の設定を行ない、フォーメーション制御における一般的な人工ポテンシャル法について述べる。このとき、従来法では考慮されなかった自己釣り合い力に関する設計自由度の存在について述べる。第3章では、軸力密度法と呼ばれる変数変換法を導入し、非零の制御力が目標フォーメーション近傍でシステムに与える影響について述べる。第4章では、安定な自己釣り合い状態のもとでの目的関数の最適化による制御系設計法を提案する。提案手法の応用例の1つとして、第5章では、むだ時間を含むシステムに対するフォーメーション制御系設計法を提案する。最後に、第6章にて、通信遅延時間

Fig. 1 Multi-agent system

を有するシステムに対する提案法の有効性をシミュレーションにより検証する。

2. 問題設定

2.1 システムの記述

N 個のエージェントが、近くのエージェントとの相対位置情報のみを利用して、指定したフォーメーションを形成するタスクを考える。対象システムを Fig. 1 に示す。 i 番目のエージェントの位置を $x_i \in \mathbb{R}^3$ 、これらを並べたものを $x = [x_1^T, x_2^T, \dots, x_N^T]^T \in \mathbb{R}^{3N}$ とする。エージェント x_i と x_j 間で相対位置情報が取得可能であるとき、 x_i と x_j を結ぶ辺ベクトル $e_k = x_j - x_i \in \mathbb{R}^3$ を定義し、これら M 本の辺ベクトルを並べて $e = [e_1^T, e_2^T, \dots, e_M^T]^T \in \mathbb{R}^{3M}$ とする。また、エージェントの番号を要素にもつ集合を $\mathcal{N} = \{1, 2, \dots, N\}$ 、さらに、辺番号を要素にもつ集合を $\mathcal{E} = \{1, 2, \dots, M\}$ とする。

頂点ベクトル x と辺ベクトル e の関係は、クロネッカ積 $\otimes$ を用いて以下のように書ける。

$$e = (C \otimes I_3)x \quad (1)$$

$C \in \mathbb{R}^{M \times N}$ は頂点と辺の接続関係を表わす行列であり、接続行列と呼ばれる。ここで C の (k, i) 要素 C_{ki} は以下のように定義される。

$$C_{ki} = \begin{cases} -1 & (\text{if } i \text{ is the start node of element } k) \\ 1 & (\text{if } i \text{ is the end node of element } k) \\ 0 & (\text{otherwise}) \end{cases} \quad (2)$$

本研究では最も基本的な状況として、各エージェントが速度制御できる場合を考える。

$$\dot{x}(t) = u(t) \quad (3)$$

$u(t) = [u_1^T, u_2^T, \dots, u_N^T]^T \in \mathbb{R}^{3N}$ は制御入力である。

本研究における制御目的を以下に定める。

【定義】 目標フォーメーションを表わすフォーメーション集合 $\Phi(d)$ を以下のように定義する.

$$\Phi(d) = \{x \mid \|e_k\| = d_k, \forall k \in \mathcal{E}\} \quad (4)$$

ただし, $d_k > 0$ は辺ベクトル e_k の目標長さを表わし, $d = [d_1, d_2, \dots, d_M]^\top \in \mathbb{R}^M$ とする. $\|e_k(x)\|$ は辺ベクトル e_k の長さである.

システムがフォーメーション集合 $\Phi(d)$ 上の点 x^* に収束したとき, つまり, $\lim_{t \rightarrow \infty} x(t) = x^* \in \Phi(d)$ となったとき, システムはフォーメーションを形成したと呼ぶ.

本研究の目的は, N 個のエージェントが, フォーメーション集合 $\Phi(d)$ で与えられる目標フォーメーションを形成することである. このとき, 各エージェント i は接続するエージェントとの相対位置 $e_k = x_j - x_i$ のみを利用可能とする. これは, 収束場所を事前に指定できない場合や, フォーメーション総体が移動するようなタスクを考える際に, 絶対位置に依存する制御法が好ましくないためである. 本問題設定は, エージェントが通信により互いの位置を交換可能な場合や, センサを用いて他エージェントとの相対位置を計測可能な場合を含む.

(4) 式より, $\Phi(d)$ は, 目標の辺長さ d_k をもつエージェント位置の集合である. $\Phi(d)$ はフォーメーション形状に対して, 剛体運動 (並進・回転) の自由度を許容しており, x^* の候補は無数に存在することに注意する.

接続行列 C , および x^* , $\Phi(d)$ に関して, 以下を仮定する. ただし, $e^* = (C \otimes I_3)x^*$ は, システムがフォーメーション集合 (4) 内の点 x^* にあるときの辺ベクトルを表わし, $R(e) = \hat{E}^\top(e) (C \otimes I_3)$, ただし, $\hat{E}(e) = \text{blockdiag}\{e_1(x), e_2(x), \dots, e_M(x)\}$ とする.

[仮定 1] $\text{rank}(R(e^*)) = 3N - 6$.

[仮定 2] $\Phi(d) \neq \emptyset$.

仮定 1 は目標フォーメーションが剛体運動以外の自由度をもたないための条件²⁴⁾を, 仮定 2 は目標長さ d からなるフォーメーションが存在するための条件を表わしている.

本研究では, 位置ベクトル x に対するスカラー関数 $f(x)$ の勾配ベクトルを横ベクトルとして, $\nabla f(x) = \frac{\partial f(x)}{\partial x} = [\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \dots, \frac{\partial f(x)}{\partial x_N}]$ と定義する.

2.2 人工ポテンシャル法

本節では, 従来よりフォーメーション制御に多く用いられている人工ポテンシャル法について述べる. 系全体のポテンシャルは, 各辺 $k \in \mathcal{E}$ に設定するポテンシャル関数 V_k の和として設定する.

$$V(\bar{e}(x)) = \sum_{k \in \mathcal{E}} V_k(\|e_k(x)\|) \quad (5)$$

また, 制御入力, ポテンシャル関数の負の勾配より,

$$u(t) = -\nabla V^\top(\bar{e}(x)) \quad (6)$$

と与える.

ポテンシャル関数の構成要素には, さまざまなものが考えられるが, 本研究では最も基本的な線形ばねを用いる. 各辺 e_k に設定されるポテンシャル $V_k(\|e_k(x)\|)$ を線形ばねにより構成するとき, (5) 式は以下で表わされる.

$$V(\bar{e}(x)) = \sum_{k \in \mathcal{E}} \frac{1}{2} k_k (\|e_k(x)\| - l_k)^2 \quad (7)$$

$$= \frac{1}{2} (\bar{e}(x) - l)^\top K (\bar{e}(x) - l) \quad (8)$$

ここで, $k_k > 0$, l_k , ($k \in \mathcal{E}$) はそれぞれ線形ばねのばね定数と自然長であり, $K = \text{diag}\{k_1, k_2, \dots, k_M\} \in \mathbb{R}^{M \times M}$, $l = [l_1, l_2, \dots, l_M]^\top \in \mathbb{R}^M$,

$$\bar{e}(x) = [\|e_1(x)\|, \|e_2(x)\|, \dots, \|e_M(x)\|]^\top \in \mathbb{R}^M \quad (9)$$

とした. ばねの自然長 l は物理的には正であるべきだが, 系の安定性には直接影響しないため, 負の自然長を許容している.

$V(\bar{e}(x))$ の勾配 $\nabla V(\bar{e}(x))$ はチェーンルール, および, (1), (8), (9) 式より,

$$(\nabla V(\bar{e}(x)))^\top = \left(\frac{\partial V(\bar{e}(x))}{\partial x} \right)^\top \quad (10)$$

$$= \left(\frac{\partial V(\bar{e}(x))}{\partial \bar{e}(x)} \frac{\partial \bar{e}(x)}{\partial e(x)} \frac{\partial e(x)}{\partial x} \right)^\top \quad (11)$$

$$= (C^\top \otimes I_3)t(e) \quad (12)$$

と求まる. ただし,

$$\frac{\partial V(\bar{e}(x))}{\partial \bar{e}(x)} = [\|t_1\|, \|t_2\|, \dots, \|t_M\|] \quad (13)$$

$$\frac{\partial \bar{e}(x)}{\partial e(x)} = \begin{bmatrix} \frac{e_1^\top(x)}{\|e_1(x)\|} & & \\ & \ddots & \\ & & \frac{e_M^\top(x)}{\|e_M(x)\|} \end{bmatrix} \quad (14)$$

$$\frac{\partial e(x)}{\partial x} = C \otimes I_3 \quad (15)$$

を用いた. (12) 式における $t(e) = [t_1^\top, t_2^\top, \dots, t_M^\top]^\top \in \mathbb{R}^{3M}$ は, ばねから生じる制御力ベクトルであり, その要素ベクトルは,

$$t_k(e_k) = k_k \left(1 - \frac{l_k}{\|e_k(x)\|} \right) e_k(x) \quad (16)$$

と表わせる.

(12) 式より, この場合の制御入力 (6) は,

$$u = -(C^\top \otimes I_3)t(e) \quad (17)$$

であり, 対応する閉ループ系は以下となる.

$$\dot{x} = -(C^\top \otimes I_3)t(e) \quad (18)$$

ここで, エージェント i の制御入力を計算すると,

$$u_i = \sum_{k \in \mathcal{E}} C_{ki} t_k(e_k) \quad (19)$$

$$= \sum_{k \in \mathcal{E}} C_{ki} k_k \left(1 - \frac{l_k}{\|e_k(x)\|} \right) e_k(x) \quad (20)$$

である。(1)式より、 C_{ki} は、 x_i が e_k の始点ノードのとき -1 、終点ノードのとき $+1$ 、それ以外で 0 であるため、エージェント i の制御入力 u_i が、自身に接続されている辺 e_k (エージェントの相対位置)のみの観測により計算できることが確認できる。また、制御入力 u_i が、ばね定数 k_k 、および、ばねの自然長 l_k の2つのパラメータで定まることに注意する。

2.3 自己釣り合い状態と新たな設計自由度

人工ポテンシャル法における制御パラメータ (k, l) は、目標フォーメーション $\Phi(d)$ においてその勾配が零 $(\nabla V^\top(d) = 0)$ 、すなわち、

$$(C^\top \otimes I_3)t(e^*) = 0 \quad (21)$$

を満たすように選ぶ必要がある。このようなパラメータの設計法として、従来、各辺 e_k に設定される個々のポテンシャルの勾配が零 $(\nabla V_k^\top(d_k) = 0)$ 、すなわち、 $t_k(e_k^*) = 0$ となるように、パラメータ $k > 0$ 、 $l = d$ を選ぶ方法が広く用いられてきた。

上記のパラメータ設計法は非常に簡便であるが、一般には目標フォーメーションにおいて $t_k(e_k^*) = 0$ を満たす必要はなく、非零の制御力 $t_k(e_k^*) \neq 0$ が存在しながらも等式(21)が成立する場合がある。本研究では、(21)式を満たす非零の制御力 $t(e^*) \neq 0$ を自己釣り合い力、各辺の制御力が互いに拮抗するような平衡状態を自己釣り合い状態と呼ぶ。自己釣り合い状態を許容した、より広いクラスにおけるパラメータ探索により、設計自由度の拡大と、それに伴う制御性能の向上が期待できる。自己釣り合い状態の例をFig. 2に示す。図中の丸印はエージェントを、エージェント間の実線は情報交換可能な辺を表わす。図のエージェント配置において、黒色の実線に引張りに相当する制御力が、灰色の実線に圧縮力に相当する制御力が適切な大きさで存在するとき、システムは自己釣り合い状態となる。このとき、たとえば図中左上のエージェントに注目すると、引張りと圧縮力が互いに釣り合い、平衡状態となっている。

自己釣り合い力を許容することは、コントローラの設計自

Fig. 2 Self-equilibrium condition

由度を拡大させる反面、制御力の釣り合いを考慮しつつ目標を達成するパラメータ (k, l) の設計が難しくなる。以降では、自己釣り合い力に関する新たな設計自由度がシステムへ与える影響を考察し、さらに、その自由度を利用した一設計法を提案する。

3. 自己釣り合い状態のシステムへの影響

前述したように、自己釣り合い状態を考慮することで、従来利用されてこなかった、新たな設計自由度を利用することができる。文献16)~20)においては自己釣り合い力が系の安定性や剛性に影響を与えることが指摘されており、拡大した設計自由度を有効に利用することで、制御性能の向上が期待できる。本章では、自己釣り合い力に関する設計自由度を軸力密度法と呼ばれる変数変換法を用いて記述する。これにより、自己釣り合い力に関する新たな設計自由度が、目標フォーメーション近傍のシステムへ与える影響を考察する。

3.1 軸力密度法による変数変換

本節では、自己釣り合い力が目標フォーメーション近傍の特性に与える影響を調べるために、軸力密度法¹⁹⁾と呼ばれる変数変換法を導入する。この手法は、構造力学の分野において古くから用いられ、特に内力をもつ構造の解析・設計に利用されている。

システムが目的の集合 $\Phi(d)$ 上の点 x^* にあるとき、変数変換、

$$t(e^*) = [q_1 e_1^{\top}, \dots, q_M e_M^{\top}]^{\top} = (Q \otimes I_3) e^* \quad (22)$$

を施すと、制御力の釣り合い条件(21)は、

$$(C^\top \otimes I_3) \hat{E}(e^*) q = 0 \quad (23)$$

と書ける。ここで、 $q = [q_1, q_2, \dots, q_M]^{\top}$ 、 $Q = \text{diag}\{q\}$ であり、その要素 q_k は、

$$q_k = \frac{\|t_k(e_k^*)\|}{\|e_k^*\|} = k_k \left(1 - \frac{l_k}{\|e_k^*\|} \right) \quad k \in \mathcal{E} \quad (24)$$

により与えられる。 q_k は、ばねの発生力とばねの長さの比(単位長さあたりの発生力)を表わす定数であり、軸力密度と呼ばれる。(24)式からわかるように、 q_k は制御力の大きさ $\|t_k(e_k^*)\|$ と直接対応しており、元々の設計変数 k_k 、 l_k が、新たな設計変数 q_k で置き換えられていることに注意する。

(22)式より、 $t(e^*)$ は q に対して線形である。この変換により、システム(18)を、 M 個の設計変数 q に対する線形関数(23)として表わすことが可能となった。この性質は、以下の影響解析や、次章の制御系設計において重要となる。以下では、軸力密度 $q_k \neq 0$ が閉ループ系の特性に与える影響について考察する。

3.2 接線剛性行列と構造の安定性

フォーメーション集合 $\Phi(d)$ 上の任意の点 x^* の近傍、 $x = x^* + \delta x$ におけるシステム(18)の線形近似は、

$$\delta \dot{x}(t) = -K_N(e^*) \delta x(t) \quad (25)$$

と表わせる．ここで， $K_N(e(x)) = \frac{\partial \nabla V(\bar{e}(x))}{\partial x} = \frac{\partial^2 V(\bar{e}(x))}{\partial x^2} \in \mathbb{R}^{3N \times 3N}$ は接線剛性行列と呼ばれる．同様に，(6) 式を x^* 近傍で線形近似すると，

$$\delta u = -K_N(e^*)\delta x \quad (26)$$

となる．上式より， $K_N(e^*)$ は，システムが平衡点にあるときに，エージェントに生じた微小変位 δx とそのとき発生する制御力 δu 間の関係（剛性）を表わしていることがわかる． $K_N(e^*)$ はその定義から点 $x = x^*$ におけるポテンシャル関数 $V(\bar{e}(x))$ のヘッセ行列である．構造力学的には， $K_N(e^*)$ から自明な運動（剛体モード）を除いた接線剛性行列 $\bar{K}_N(e^*)$ が正定，すなわち， $\bar{K}_N(e^*) \succ 0$ であれば，構造が安定であることが知られている¹⁴⁾．

接線剛性行列 $K_N(e^*)$ は，

$$K_N(e^*) = K_G(q) + K_E(c) \quad (27)$$

と表わすことができることが知られている¹⁸⁾．ここで $K_G(q)$ は幾何剛性行列， $K_E(c)$ は線形剛性行列と呼ばれる行列であり，それぞれ，

$$K_G(q) = (C^\top \otimes I_3) \tilde{Q}(q) (C \otimes I_3) \quad (28)$$

$$K_E(c) = (C^\top \otimes I_3) \hat{E}(e^*) D(e^*, c) \hat{E}^\top(e^*) (C \otimes I_3) \quad (29)$$

ただし，

$$\tilde{Q}(q) = \text{blockdiag} \{q_1 I_3, q_2 I_3, \dots, q_M I_3\} \quad (30)$$

$$D(e^*, c) = \text{diag} \left\{ \frac{c_1}{\|e_1^*\|^3}, \frac{c_2}{\|e_2^*\|^3}, \dots, \frac{c_M}{\|e_M^*\|^3} \right\} \quad (31)$$

で定義される^{14), 17), 18)}．ただし， c は部材特性と呼ばれ，以下で書ける．

$$c = [c_1, \dots, c_M] = [k_1 l_1, k_2 l_2, \dots, k_M l_M] \quad (32)$$

(27)～(29) 式からわかるように， $K_N(e^*)$ は，軸力密度 q により構成される $K_G(q)$ と，部材特性 c から構成される $K_E(c)$ の和で表わされる．目標フォーメーションにて $t(e^*) = 0$ ，つまり $q = 0$ とする従来の方法は， $K_G(q) = O$ と限定し， c に関する $K_E(c)$ の設計自由度のみを用いてコントローラを設計する状況に相当することがわかる．

$q \neq 0$ の自由度が， $K_N(e^*)$ に与える影響をより詳細に調べるため，(27)～(29) 式を整理すると， $K_N(e^*)$ は，

$$K_N(e^*) = (C^\top \otimes I_3) \text{blockdiag} \{ \bar{K}_1, \bar{K}_2, \dots, \bar{K}_M \} (C \otimes I_3) \quad (33)$$

ただし，

$$\bar{K}_k(e_k^*, q_k, c_k) = q_k I_3 + \frac{c_k}{\|e_k^*\|^3} e_k^* e_k^{*\top} \in \mathbb{R}^{3 \times 3} \quad (34)$$

と表わせる．(34) 式の構造より， $\bar{K}_k$ は，軸力密度 q_k から構成される行列 $q_k I_3$ ，および，部材特性 c_k から構成される行列 $\frac{c_k}{\|e_k^*\|^3} e_k^* e_k^{*\top}$ の線形結合で表わされる．また，

$$e_k e_k^\top = \begin{bmatrix} e_{kx}^2 & e_{kx} e_{ky} & e_{kx} e_{kz} \\ e_{kx} e_{ky} & e_{ky}^2 & e_{ky} e_{kz} \\ e_{kx} e_{kz} & e_{ky} e_{kz} & e_{kz}^2 \end{bmatrix} \neq \alpha I_3, \forall e_k (\alpha \neq 0) \quad (35)$$

であることに注意する．

(34)，(35) 式より，実現可能な $\bar{K}_k$ の集合は， $q_k \neq 0$ ，すなわち $q_k I_3$ を利用することで必ず広がることがわかる． $q \neq 0$ ，すなわち自己釣り合い力を利用することは， c に関する $\frac{c_k}{\|e_k^*\|^3} e_k^* e_k^{*\top}$ に加え， q に関する $q_k I_3$ の自由度を含めたより広いクラスで， $\bar{K}_N(e^*) \succ 0$ を満たすコントローラを設計することに相当する．

4. 自己釣り合い状態の設計法

本章ではフォーメーション制御のための安定な自己釣り合い状態の設計法を考える．自己釣り合い力の自由度を考慮する場合，力の釣り合い (23) を保ちつつ，ポテンシャル関数 (のヘッセ行列 $\bar{K}_N(e^*)$) のパラメータ (q, c) を直接調整することは困難である．そこで本研究では，最適化手法を用い，制御性能に直接関連した目的関数を最小化（または最大化）することでパラメータの設計を行なう．

4.1 剛体モードの除去

フォーメーション制御では，まず，フォーメーション集合 $\Phi(d)$ 内の目標フォーメーション x^* が安定となるよう制御することが求められる．前述したように， x^* 近傍におけるシステムの安定性は，接線剛性行列 $K_N(e^*)$ より自明な剛体運動を除いた接線剛性行列 $\bar{K}_N(e^*)$ の正定性により判定できる． $\bar{K}_N(e^*)$ は， $K_N(e^*)$ を用いて以下のように求めることができる．

まず，並進・回転運動を表わす基底ベクトルを並べた行列 P を定義する． $P = [p_1, p_2, \dots, p_6] \in \mathbb{R}^{3N \times 6}$ ，ただし， $p_1 = 1_N \otimes e_x$ ， $p_2 = 1_N \otimes e_y$ ， $p_3 = 1_N \otimes e_z$ ， $p_4 = [(I_n \otimes \text{skew}(e_x))x^*]$ ， $p_5 = [(I_n \otimes \text{skew}(e_y))x^*]$ ， $p_6 = [(I_n \otimes \text{skew}(e_z))x^*]$ である． e_x, e_y, e_z は， x, y, z 軸方向の単位ベクトルであり， $\text{skew}(\cdot)$ は，外積と等価な歪対称行列を生成する関数である． $p_1 \sim p_3$ が並進運動を， $p_4 \sim p_6$ が回転運動の自由度を表わす¹⁴⁾．

P の特異値分解を， $P = U_P \Sigma_P V_P^\top = U_{P1} \Sigma_r V_{P1}^\top$ ，ただし， $V_P = [V_{P1}, V_{P2}]$ ， $\Sigma_P = \text{blockdiag}\{\Sigma_r, O\}$ ， $U_P = [U_{P1}, U_{P2}]$ とする．このとき，剛体モードを除いた接線剛性行列 $\bar{K}_N$ は， $U_{P2} \in \mathbb{R}^{3N \times (3N-6)}$ を用いて，

$$\bar{K}_N = U_{P2}^\top K_N U_{P2} \quad (36)$$

と書ける¹⁴⁾．なお， K_N における設計変数の線形性は，この変換において保存されることに注意する．

4.2 設計変数の置き換えと設計手順

前述したように， k_k, l_k は辺 e_k に対応するばね定数，および，ばねの自然長であり，ポテンシャル関数 $V_k(\|e_k^*(x^*)\|)$ (もしくは，そのヘッセ行列 $\bar{K}_N(e^*)$) の形状を決める直接の物理定数である．しかしながら，(24)，(31) 式からわかるよ

うに、これらのパラメータは、剛性行列 (28), (29) 中に積の形で存在するため、設計変数としてそのまま用いることは好ましくない。そこで本手法では、本来の設計変数 (k, l) の代わりに、(システムが目標フォーメーション x^* にあるときの) 軸力密度および部材特性を求めたのち、これらを用いて (k, l) を導出する。幸い、(23), (28), (29) 式は、軸力密度 q 、および、部材特性 c に関して線形であり、(24) 式より、 $k_k \neq 0$ のもとで、 (k, l) と (q, c) は一対一対応である ($k_k = 0$ のとき、 l_k は任意になることに注意)。

$$\begin{cases} q_k = k_k \left(1 - \frac{l_k}{\|e_k^*\|}\right) \\ c_k = k_k l_k \end{cases} \iff \begin{cases} k_k = q_k + \frac{c_k}{\|e_k^*\|} \\ l_k = \frac{c_k}{k_k} \end{cases} \quad (37)$$

以上の変換を用いることで、以降の設計に用いる制約式 (23), および、 $\bar{K}_N(e^*) \succ 0$ を凸関数として表わすことが可能となる。新たな設計変数 (q, c) について、以下の性質が示される。

[性質 1] 閉ループ系 (18) を考え、仮定 1 および仮定 2 が成り立つとする。任意の目標フォーメーション $x^* \in \Phi(d)$ において、自己釣り合いの条件 (23) および $\bar{K}_N \succ 0$ を満たすパラメータ (q, c) が少なくとも 1 つ存在する。

(証明) $q = 0$ かつ $c > 0$ と選んだとき、任意の x^* において $\bar{K}_N(e^*) \succ 0$ を満たすことを示す。 $q = 0$ のとき、(28) 式により、 $K_G(q) = O$ となる。また、(29) 式は、 $K_E(c) = R^T(e^*)D(e^*, c)R(e^*) \succeq 0$ と書ける。 $c > 0$ より $D \succ 0$ であるため、仮定 1 より、 $\text{rank}(K_E(c)) = \text{rank}(R(e^*)) = 3N-6$ となる。以上より、 $\text{rank}(\bar{K}_N(e^*)) = \text{rank}(K_E(c)) = 3N-6$ となり、 $\bar{K}_N(e^*)$ から 6 つの自明な零モードを除いた $\bar{K}_N$ は、 $\bar{K}_N(e^*) \succ 0$ を満たす。□

以降のコントローラ設計の手順を以下に示す。

1. フォーメーション集合 $\Phi(d)$ 内の任意の点から目標フォーメーション x^* を選ぶ。
2. 目標フォーメーションにてシステムが安定となる ((23) 式と $\bar{K}_N(e^*) \succ 0$ を満たす) ような (q, c) を求める。
3. 求めた (q, c) から、(37) より (k, l) を導出し、コントローラ (16), (17) を決定する。

制御時にコントローラで用いるパラメータは (k, l) のみであり、 (q, c) および目標フォーメーション x^* は設計時にのみ用いられることに注意する。以降では、 (q, c) を設計変数とした制御系設計問題について考える。

4.3 自己釣り合い状態の最適設計

前章の議論より、 $\bar{K}_N(e^*)$ から剛体モードを除いた $\bar{K}_N(e^*)$ の最小固有値は、目標フォーメーション x^* の安定性 (剛体モードを除く) を表わす良い指標となる。そこで本研究では、自己釣り合い状態の設計問題を、 $\bar{K}_N(e^*)$ の固有値の下限を制約した、パラメータ (q, c) の最適化問題として定式化する。制御目的に応じて定められる目的関数を $f(q, c) \in \mathbb{R}$ とすれ

ば、最適化問題は以下で与えられる。

[設計問題 1] (3) 式で与えられるシステムを考える。安定なフォーメーション $x^* \in \Phi(d)$ が形成されるような (q, c) の設計問題は、以下で表わされる。

$$\begin{aligned} \min_{q, c} \quad & f(q, c) \\ \text{s.t.} \quad & (C^T \otimes I_3) \hat{E}(e^*) q = 0 \\ & q_k + \frac{c_k}{\|e_k^*\|} \geq 0 \quad (k \in \mathcal{E}) \\ & \bar{K}_N(e^*, q, c) \succeq \varepsilon_{\min} I \end{aligned} \quad (38)$$

ただし、 $\varepsilon_{\min} > 0$ とする。

求めた (q, c) より、(37) を用いてパラメータ (k, l) を導出し、(17) に適用することでコントローラを得る。

制約条件の第 1 式は、制御力の釣り合い条件 (23) である。制約式の第 2 式は、辺 k のばね係数 $k_k = q_k + \frac{c_k}{\|e_k^*\|}$ が正であることを保証する条件である ((37) 式参照)。また、第 3 式は、接線剛性行列 $\bar{K}_N$ の正定条件であり、これにより、システムの安定性が保証される。(27)~(29) および (36) 式より、この制約式は、 q, c に関する線形行列不等式である。なお、性質 1 より、仮定 1 および仮定 2 が満たされているとき、(38) の解は少なくとも 1 つ存在することに注意する。

以上からわかるように、自己釣り合い状態設計のための最適化問題は、目的関数 $f(q, c)$ を線形関数とすることで、軸力密度 q と部材特性 c に関する半正定値計画問題 (SDP: Semidefinite Programming) として定式化できる。

4.4 系の安定性と収束先

前述したように、上記の最適化問題において、 x^* を目標フォーメーションとして設計したシステムは、必ずしも x^* そのものに収束するわけではない。 x^* 近傍 $x(0) = x^* + \delta x$ から出発した解は、以下に示すように、 x^* に剛体変換 (並進・回転運動) を施した点に収束する。

[性質 2] 設計問題 1 で設計されたシステム (18) を考える。また、 $P \in \mathbb{R}^{3N \times 6}$ を並進・回転運動を表わす基底ベクトルを並べた行列とし、その特異値分解を $P = U_P \Sigma_P V_P^T = U_{P1} \Sigma_r V_{P1}^T$ 、ただし、 $V_P = [V_{P1}, V_{P2}]$, $\Sigma_P = \text{blockdiag}\{\Sigma_r, O\}$, $U_P = [U_{P1}, U_{P2}]$ とする (4.1 節と同様)。目標フォーメーション x^* 近傍の初期値 $x(0) = x^* + \delta x_0$ から出発した解は、 x^* に剛体変換 $U_{P1} U_{P1}^T \delta x_0$ を施した点 $x^* + U_{P1} U_{P1}^T \delta x_0 \in \Phi(d)$ に収束する。

(証明) x^* 近傍の初期値 $x_0 = x^* + \delta x_0$ から出発する解の挙動を調べるため、(18) 式の線形化システム (25) を考える。 P の特異値分解 $P = U_P \Sigma_P V_P^T$ より求まる U_P を用いて、正則変換、

$$U_P^T \delta x = \begin{bmatrix} U_{P1}^T \delta x \\ U_{P2}^T \delta x \end{bmatrix} = \begin{bmatrix} \delta \bar{x}_1 \\ \delta \bar{x}_2 \end{bmatrix} \quad (39)$$

を行なうと, (25) 式は以下のように変換される.

$$\begin{bmatrix} \delta \dot{x}_1 \\ \delta \dot{x}_2 \end{bmatrix} = - \begin{bmatrix} O & O \\ O & \bar{K}_N \end{bmatrix} \begin{bmatrix} \delta x_1 \\ \delta x_2 \end{bmatrix} \quad (40)$$

ただし, $\text{Ker}(K_N) = \text{Im}(P) = \text{Im}(U_{P1})$ より $K_N U_{P1} = O$ であること, および K_N が対称であることを用いた. (40) 式と $\bar{K}_N \succ 0$ より, 部分空間 $\text{Im}(U_{P2}^\top)$ 内の成分 $\delta x_2(t)$ は原点に収束するが, 部分空間 $\text{Im}(U_{P1}^\top)$ 内の成分 $\delta x_1(t)$ は初期値 $\delta x_1(0)$ から動かないことがわかる. 任意の δx_0 は, $\text{Im}(U_{P1}^\top)$ と $\text{Im}(U_{P2}^\top)$ の要素を用いた直和分解

$$\delta x_0 = U_{P1} U_{P1}^\top \delta x_0 + U_{P2} U_{P2}^\top \delta x_0 \quad (41)$$

により表わすことができ, δx_0 から出発した (25) 式の解は, $x^* + U_{P1} U_{P1}^\top \delta x_0$ に収束する. $\text{Im}(P) = \text{Im}(U_{P1})$ より, $U_{P1} U_{P1}^\top \delta x_0$ は剛体変換の自由度を表わしている. よって性質は示された. $\square$

(注意) 非零の要素間力が釣り合い, 平衡状態を実現するという本手法の性質上, あるエージェントに故障が発生し, 故障エージェントを切り離した場合, 力の釣り合いが満たされず, 目標フォーメーションが形成されない可能性がある. この場合の対応策として, 従来パラメータへの切り替え (自然長に目標長を設定する $l = d$) が考えられるが, 本手法で付与した制御性能が失われてしまうなど, 対故障性に課題が残る.

$f(q, c)$ の設定方法としては, 剛性の最大化や, 外乱影響の最小化など, さまざまなものが考えられるが, 次章ではその一例として, エージェント間の通信遅延を有するフォーメーション制御における, 許容遅延時間の最大化について考える.

5. 通信遅延を有するシステムに対するフォーメーション制御

分散エージェントの協調制御が抱える問題の1つとして, 通信遅延に起因するむだ時間の影響がある. そこで前章までの結果の応用例の1つとして, 通信遅延を有するマルチエージェントシステムに対するフォーメーション制御系設計法について考える. 本研究では簡単のため, すべての辺に発生する遅延時間は等しく $\tau > 0$ とする. なお, 本来ならば, エージェント i から見た自己の状態 $x_i(t)$ には遅延時間が生じないが, (16) 式より, 本研究の制御則は相対位置情報 $e_k(t) = x_j(t) - x_i(t)$ のみを利用するため, エージェント $x_i(t), x_j(t)$ に生じる遅延時間を τ で揃えていることに注意する.

5.1 問題設定

時間 τ の通信遅延が存在する場合, 制御入力,

$$u(t) = -(C^\top \otimes I_3)t(e(t - \tau)) \quad (42)$$

となる. また, 対応する閉ループ系は以下となる.

$$\dot{x}(t) = -(C^\top \otimes I_3)t(e(t - \tau)) \quad (43)$$

点 $x^* \in \Phi(d)$ 近傍における遅延時間の影響を調べるため, むだ時間システム (43) を線形化すると, (25) 式と同様に,

$$\delta \dot{x}(t) = -K_N(e^*)\delta x(t - \tau) \quad (44)$$

と書ける. さらに, (39) 式の変数変換を適用すると, (40) 式と同様にして, (42) 式のシステムは以下で表わせる.

$$\delta \dot{x}_2(t) = -\bar{K}_N(e^*)\delta x_2(t - \tau) \quad (45)$$

許容むだ時間の最大化問題は, $\bar{K}_N(e^*)$ の最大固有値最小化問題と等価であることが知られている²⁵⁾.

[性質 3] (45) 式のシステムを考える. $\bar{K}_N(e^*) \succ 0$ であり, また, $\bar{K}_N(e^*)$ の最大固有値を $\lambda_{\max}$ とする. このときシステムが漸近安定であるための必要十分条件は

$$0 \leq \tau \leq \frac{\pi}{2\lambda_{\max}} \quad (46)$$

である.

証明は, 文献 25) を参照されたい.

5.2 許容むだ時間の最大化設計

性質 3 より, むだ時間システム (44) の許容むだ時間最大化のためには, $\bar{K}_N(e^*)$ の最大固有値の最小化を行なえばよい. この問題は, 最適化問題 (38) を用いて, 以下のように定式化できる. 最適化問題を以下に示す.

[設計問題 2] 入力むだ時間を含むマルチエージェントシステム (45) を考える. 許容むだ時間の最大化を目的としたとき, 安定なフォーメーション $x^* \in \Phi(d)$ が形成されるような (q, c) の設計問題は, 以下で表わされる.

$$\begin{aligned} \min_{q, c, \varepsilon_{\max}} \quad & \varepsilon_{\max} \\ \text{s.t.} \quad & (C^\top \otimes I_3)\hat{E}(e^*)q = 0 \\ & q_k + \frac{c_k}{\|e_k^*\|} \geq 0 \quad (k \in \mathcal{E}) \\ & \bar{K}_N(e^*, q, c) \succeq \varepsilon_{\min} I \\ & \bar{K}_N(e^*, q, c) \preceq \varepsilon_{\max} I \end{aligned} \quad (47)$$

ただし, $\varepsilon_{\min} > 0$ とする.

求めた (q, c) より, (37) を用いてパラメータ (k, l) を導出し, (42) に適用することでコントローラを得る.

(47) 式の制約条件第 4 式より, $\varepsilon_{\max}$ は $\bar{K}_N$ の最大固有値の上限を与える. (47) 式は $(q, c, \varepsilon_{\max})$ に対する SDP となっていることに注意する. 上で述べているように, 最適化問題より得られた (q, c) より, (37) 式から (k, l) を求めると, (8) 式よりポテンシャル関数 $V(\bar{e})$ が定まり, (42) 式より, 制御入力 $u(t)$ が決定される.

6. 数 値 例

本章では, 前章の最適化法 (47) の有効性を数値例により検証する. Case 1 では, 提案法の基本的な性質を調べるため, 二次元平面上におけるフォーメーション制御系の設計を行なう. 自己釣り合い力の設計自由度を用いた提案法と, 自由度

を用いない ($q = 0$ とした) 従来法との比較を行ない, 提案手法により最大許容遅延時間が改善されることを示す. Case 2 では, より一般的な三次元空間内のフォーメーション制御系にて, 提案法の有効性を検証する. また, Case 3 では, エージェント数の増大が最適化の計算時間へ及ぼす影響を検証する.

6.1 Case 1: 二次元平面上のフォーメーション制御

本節では, 前章の最適化手法 (47) の有効性を, エージェント数 $N = 5$, 情報交換可能な辺の数 $M = 8$ とした場合の数値例により検証する. 簡単のため, 各エージェントの運動は二次元平面上にあるとする. Fig. 3 に目標フォーメーション x^* における, エージェント x_i と辺 e_k の配置を示す. 図中 a, b はフォーメーションの縦横比であり, 図には $a = b = 1$ の場合を示している. 最適化問題 (47) 中の剛性の下限値は $\varepsilon_{\min} = 1$ とし, ソルバには主双対内点法を実装した SDP ソルバ SeDuMi²⁶⁾を用いた. また, 比較のため, $q = 0$ すなわち, $\bar{K}_N$ 中の幾何剛性行列 ((28) 式の $K_G(q)$) を O として最適化を行なった場合 (従来法と呼ぶ) についても同時に調べた.

結果を Fig. 4 に示す. 図には, 目標フォーメーション形状

Fig. 3 Target formation

Fig. 4 Permissible time delays (solid line: proposed, dotted line: conventional)

Fig. 5 Optimization result $a = 5$ ((a) (b): proposal, (c) (d): conventional)

の横の長さを $a = 1 \sim 10$ として最適化を行なった場合の最大許容遅延時間 $\tau_{\max} = \frac{\pi}{2\varepsilon_{\max}} = \frac{\pi}{2\lambda_{\max}}$ が示しており, 実線が提案手法, 破線が従来手法の結果である. 両者とも, a が大きく横長のフォーメーション形状になるほど許容遅延時間が短くなっているが, いずれの場合でも, 提案手法のほうが大きな遅延時間を許容している. $a = 10$ のとき, 提案手法の許容遅延時間は, 従来手法の約 18 倍となっている.

また, Fig. 5 に $a = 5$ の場合の最適化結果を示す. 上段 (a), (b) が提案手法, 下段 (c), (d) が従来手法の結果であり, それぞれ, 左側が各辺に設定されたばね定数 k (制御ゲインに相当), 右側が軸力密度 q であり, 値の大きさを線の太さで表わしている (黒色は正の値, 灰色は負の値). なお, 従来手法と提案手法の剛性行列の最小固有値は, ともに $\varepsilon_{\min} = 1$ と一致していた.

Fig. 5 (a), (c) より, 従来手法と提案手法は同じ剛性の下限値を有するにもかかわらず, 制御ゲイン k は提案手法のほうが小さく抑えられていることが確認できる. 一般に制御ゲインの増大は, 遅延時間に対する安定性 (ロバスト性) を損なう方向に作用する. 提案法は, Fig. 5 (b) に示すように, 平衡点における軸力 (自己釣り合い力) の自由度を利用することで, 制御ゲインの小さなコントローラの設計を可能としている.

つぎに, 目標フォーメーション形状を Fig. 5, エージェントの初期位置 x_0 を Fig. 6 としたときのシステムの時間応答を示す. Fig. 7 は, 各辺の長さの誤差 $\Delta e_k = \|e_k\| - l_k$ の時間応答であり, 上段 (a), (b) が提案手法, 下段 (c), (d) が従来手法である. それぞれ, 左側が遅延時間なし ($\tau = 0$) の場合, 右側が遅延時間あり ($\tau = 0.1$) の場合である. 図より, 従来法 (下段) は, 遅延時間の存在しない, $\tau = 0$ における収

Fig. 6 Initial configuration $x(0)$ Fig. 7 Relative distance error Δe_k ((a) (b): proposed, (c) (d): conventional)

Fig. 8 Initial and target configuration

束は早いものの、 $\tau = 0.1$ の場合において発散している。一方、提案法（上段）は、いずれの許容遅延時間の場合にも良好な特性を示している。この結果は、Fig. 4 とも整合している。

6.2 Case 2: 三次元空間内のフォーメーション制御

本節では、より一般的な三次元空間内のフォーメーションに対する設計例を示す。エージェント数を $N = 6$ 、情報交換可能な辺の数を $M = 12$ とし、そのほかの条件は Case 1 と同じとした。エージェントの初期位置 x_0 、および、目標フォーメーション x^* を Fig. 8 (a), (b) に示す。

Fig. 9 Agent trajectories in three dimensional space

Fig. 10 Grid formation

この場合、最適化により得られる最大許容遅延時間は約 $\tau = 0.58$ となった。従来法で設計を行なった場合、最大許容時間は約 $\tau = 0.22$ であり、提案手法は従来手法に比べて 2 倍以上の遅延時間を許容していた。遅延時間を $\tau = 0.22$ としたときの各エージェントの応答軌道を Fig. 9 に示す。図の×印はエージェントの初期位置を、丸印が最終位置を表わし、エージェントの軌道を点線で示している。図より、各エージェントは、目標フォーメーションに対して剛体運動を許容した位置に収束しており、二次元平面内の場合同様、良好な特性が得られていることが確認できる。

6.3 Case 3: エージェント数の増大の影響

本手法を適用する際、数値最適化の計算コストを考慮する必要がある。そこで本節では、エージェント数の増大が計算時間へ及ぼす影響について調査する。目標フォーメーション x^* を Fig. 10 に示す。図のように網目状に多数 ($N = n^2$) 配置されたエージェントが、周囲八方向のエージェントと情報交換可能な状況を考える。エージェント数が $N = 4$ ($n = 2$) から $N = 100$ ($n = 10$) の場合について最適化問題 (47) を解き、計算時間を計測した。なお、最適化計算には 3.2 GHz の Intel Core i5-4570 CPU、8 GB の RAM をもつ計算機を使

Table 1 Optimization time

Agent number	—	$4(n=2)$	$9(n=3)$	$16(n=4)$	$25(n=5)$
Optimization time (s)	—	1.13×10^{-1}	1.31×10^{-1}	3.64×10^{-1}	1.64
Agent number	$36(n=6)$	$49(n=7)$	$64(n=8)$	$81(n=9)$	$100(n=10)$
Optimization time (s)	7.49	2.34×10^1	8.22×10^1	1.87×10^2	4.75×10^2

用, MATLAB7.5.0.342 上で SDP ソルバ SeDuMi をデフォルト設定で用いた.

各エージェント数に対して 10 回ずつの計測を行なった平均値を Table 1 に示す. 表より, エージェント数 $N = 100$ の場合には計算終了までに約 475 秒を要する結果となり, エージェント数に対する計算時間の増大が課題となる.

7. おわりに

本研究では, マルチエージェントシステムのフォーメーション制御において, 自己釣り合い力 (互いに釣り合う非零の制御力) の自由度を積極的に利用した制御系設計法について考えた. 制御入力, エージェント間のばねにより構成されるポテンシャル関数の負の勾配とし, そのばね特性の設計法を示した. 設計変数には, 構造設計の分野で用いられる軸力密度を導入し, 安定なフォーメーションが形成されるような制御パラメータの設計問題を, 半正定値計画問題 (凸計画問題) として定式化した. 自己釣り合い力の自由度を含めた設計問題を凸計画問題として定式化したことにより, 本自由度を用いた場合と用いない場合の比較・評価が可能となった. 目標フォーメーションにおいて制御力を零とする従来の設計法は, 接線剛性行列の構成要素の観点でみると, 軸力密度に起因する幾何剛性行列の自由度を利用せず, 部材特性に起因する線形剛性行列の自由度のみで設計を行なっていることに相当する. 上記最適化手法の適用例として, 通信遅延を有するシステムにおける許容むだ時間の最大化問題を考えた. 手法の有効性は, 通信遅延を有するシステムに対する数値例により検証し, 自己釣り合い状態を許容したより広いクラスで設計を行なうことで, 従来手法に対して大幅な許容むだ時間の増加が確認できた.

一部のエージェントが故障した場合の対策や, 多数のエージェントを有するシステムに対する効率的な計算アルゴリズムの開発が今後の課題である.

参 考 文 献

- 1) Y. Cao, W. Yu, W. Ren and G. Chen: An overview of recent progress in the study of distributed multi-agent coordination, *IEEE Transactions on Industrial Informatics*, **9**-1, 427/438 (2013)
- 2) R.M. Murray: Recent research in cooperative control of multivehicle systems, *Journal of Dynamic Systems, Measurement, and Control*, **129**-5, 571/583 (2007)
- 3) Y.Q. Chen and Z. Wang: Formation control: a review and a new consideration, *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 3181/3186 (2005)
- 4) R. Olfati-Saber, J.A. Fax and R.M. Murray: Consensus and cooperation in networked multi-agent systems, *Proceedings of the IEEE*, **95**-1, 215/233 (2007)
- 5) W. Ren and R. Beard: *Distributed Consensus in Multi-Vehicle Cooperative Control: Theory and Applications*, Springer (2007)
- 6) R. Olfati-Saber and R.M. Murray: Distributed cooperative control of multiple vehicle formations using structural potential functions, *IFAC World Congress*, 346/352 (2002)
- 7) D.V. Dimarogonas and K.H. Johansson: On the stability of distance-based formation control, *Proc. 47th IEEE Conference on Decision and Control*, 1200/1205 (2008)
- 8) H.G. Tanner, A. Jadbabaie and G.J. Pappas: Stable flocking of mobile agents, part i: Fixed topology, *Proc. 42nd IEEE Conference on Decision and Control*, **2**, 2010/2015 (2003)
- 9) H.G. Tanner, A. Jadbabaie and G.J. Pappas: Stable flocking of mobile agents part II: Dynamic topology, *Proc. 42nd IEEE Conference on Decision and Control*, **2**, 2016/2021 (2003)
- 10) M. Ji and M.B. Egerstedt: *Distributed Coordination Control of Multi-Agent Systems While Preserving Connectedness*, Georgia Institute of Technology (2007)
- 11) P.R. Giordano, A. Franchi, C. Secchi and H.H. Bühlhoff: A passivity-based decentralized strategy for generalized connectivity maintenance, *The International Journal of Robotics Research*, **32**-3, 299/323 (2013)
- 12) D. Zelazo, A. Franchi, H.H. Bühlhoff and P.R. Giordano: Decentralized rigidity maintenance control with range measurements for multi-robot systems, *The International Journal of Robotics Research*, 0278364914546173 (2014)
- 13) D. Zelazo, A. Franchi, F. Allgöwer, H.H. Bühlhoff and P.R. Giordano: Rigidity maintenance control for multi-robot systems, *Robotics: Science and Systems*, 473/480 (2012)
- 14) R.E. Skelton and M.C. de Oliveira: *Tensegrity Systems*, Springer (2009)
- 15) S.H. Juan and J.M. Mirats Tur: Tensegrity frameworks: Static analysis review, *Mechanism and Machine Theory*, **43**-7, 859/881 (2008)
- 16) R. Connelly: Tensegrity structures: Why are they stable?, *Rigidity Theory and Applications*, 47/54 (2002)
- 17) R. Connelly and W. Whiteley: Second-order rigidity and prestress stability for tensegrity frameworks, *SIAM Journal on Discrete Mathematics*, **9**-3, 453/491 (1996)
- 18) J. Zhang and M. Ohsaki: Stability conditions for tensegrity structures, *International Journal of Solids and Structures*, **44**-11, 3875/3886 (2007)
- 19) A. Tibert and S. Pellegrino: Review of form-finding methods for tensegrity structures, *International Journal of Space Structures*, **18**-4, 209/223 (2003)
- 20) K. Nagase and R. Skelton: Minimal mass design of tensegrity structures, *Proc. SPIE Conference on Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems*, 90610W (2014)
- 21) B. Nabet and N.E. Leonard: Shape control of a multi-agent system using tensegrity structures, *Lagrangian and Hamiltonian Methods for Nonlinear Control*, 329/339, Springer (2007)
- 22) D. Pais, M. Cao and N.E. Leonard: Formation shape and orientation control using projected collinear tensegrity

- structures, *Proc. American Control Conference*, 610/615 (2009)
- 23) S.Y. Lau and W. Naeem: Tensegrity-based formation control of unmanned vehicles, *Proc. UKACC International Conference on Control (CONTROL)*, 1/6 (2012)
- 24) M. Mesbahi and M. Egerstedt: *Graph Theoretic Methods in Multiagent Networks*, Princeton University Press (2010)
- 25) R. Olfati-Saber and R.M. Murray: Consensus problems in networks of agents with switching topology and time-delays, *IEEE Transactions on Automatic Control*, **49**–9, 1520/1533 (2004)
- 26) J.F. Sturm: Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones, *Optimization Methods and Software*, **11**–1-4, 625/653 (1999)

=====

[著 者 紹 介]

増 田 容 一

2013 年和歌山大学システム工学部光メカトロニクス学科卒業。同年同大学大学院システム工学研究科に入学，現在に至る。張力構造の設計，マルチエージェントシステムの制御に関する研究に従事。

長 瀬 賢 二（正会員）

1998 年名古屋大学大学院工学研究科博士後期課程修了。同年名古屋大学工学研究科助手。2005 年和歌山大学システム工学部助教授，2007 年より同准教授，2015 年より同教授。振動の制御，ロボットの制御などの研究に従事。日本機械学会，日本ロボット学会などの会員（工学博士）。

=====