衝突回避を考慮した複数ロボットのデッドロックフリー軌道追跡制御

桜 間 一 徳*・中 野 和 司*

Deadlock-free Path Following Control with Collision Avoidance for Multiple Robots

Kazunori SAKURAMA* and Kazushi NAKANO*

This paper deals with a path following problem with collision avoidance for multiple robots. The path following aims to move the robots along reference paths with assigned velocities. When there are geometric errors between the robots' positions and the reference paths, or when the differences between their velocities and assigned velocities are not zero, we expect to reduce these errors. Unfortunately, if the multiple robots try to realize the exact path following, they may collide with one another in areas where the reference paths intersect. In this case, the robots have to avoid collision at the expense of the original paths. This paper introduces a value function including geometric and velocity errors, and proposes a new online collision avoidance method which constrains the value function. The proposed method minimizes the time derivative of the value function in each instance. Moreover, this method prevents deadlocks of the robots with the following strategy: design a time-varying function which moves slowly along the reference path for each robot, and append a penalty function to the value function which increases when the position of the robot becomes less than the time-varying function.

Key Words: multiple robots, collision avoidance, path following, deadlock-free

1. はじめに

近年、ロボットの自律化に向けて、複数ロボットの動作計画法の研究が盛んに行なわれている¹⁾。このような研究で目的とされているのは、各ロボットが衝突を起こさないように目標点に到達する軌道を設計することである。LaValle ら²⁾はロボット同士の相対距離を評価関数に導入することで、衝突回避を考慮した最適軌道を設計するアルゴリズムを提案している。Cassandras ら³⁾はこのような最適化をオンラインで行なう Receding Horizon による手法を提案した。この方法では計算コストが問題となることがあるが、Richards ら⁴⁾や Hao ら⁵⁾は衝突回避問題を比較的、計算コストが小さい混合整数計画問題に帰着させている。しかし、制御性能を上げるためにはサンプリング時間を小さくし予測ホライズンを大きくとる必要があるため、依然として計算コストの問題が残る。新井ら⁶⁾や Loizou ら⁷⁾はロボット同士の相対距離を用いたポテンシャルを考えることで、計算コストの小さい静的フィードバックによる衝突回避法を提案している。このような衝突回避に関する研究では、ロボットを静止した目標点へ移動させることを考えている。

一方、ある与えられた目標軌道上を指定速度でロボットを

走行させる軌道追跡、あるいはライントレースも重要である。単体のロボットに対して軌道追跡を考える問題は文献 8)～11)などで議論されている。これらの論文では、ロボットの位置と目標軌道の偏差(幾何偏差)やロボットの速度と指定速度の偏差(速度偏差)がある場合に、これらの偏差を減少させるための制御手法が議論されている。軌道追跡を複数ロボットに行なうことを考えると、目標軌道が交差している場所などでロボット同士が衝突する恐れがある。このような問題に対して筆者ら¹²⁾は、複数ロボットに対して与えられた目標軌道を衝突回避を考慮してオンライン修正する手法を提案した。この手法では、時間関数として与えられた目標軌道を各時刻である種の写像を用いて修正する。この手法は速度偏差を増加させない代わりに、ロボットは目標軌道から離れるため、幾何偏差を増加させる。しかし、軌道追跡では、ロボットを目標軌道上で減速させることで、幾何偏差をできるだけ増加させず衝突回避させることが望ましい。一方、Akella ら¹³⁾はロボットの起動時間をスケジューリングし、ロボットを目標軌道上で動かすことで衝突回避を実現している。この手法では、速度偏差を考慮しない代わりに幾何偏差を増加させずに衝突回避を実現している。ただし、全時間にわたっての目標軌道を用いてオフラインスケジューリングを行なうため、目標軌道がすべての区間にわたって事前に与えられていなければならない。

本論文では、複数ロボットに対して衝突回避を考慮した軌道追跡制御を考える。ここで問題となることは、軌道追跡の制御指標となる幾何偏差と速度偏差は衝突回避を考慮すると

* 電気通信大学大学院電子工学専攻 調布市調布ヶ丘 1-5-1

* Department of Electronic Engineering, The University of Electro-Communications, 1-5-1 Chofu-ga-oka, Chofu

(Received April 2, 2009)

(Revised July 15, 2009)

Fig. 1 Path following for multiple robots

必ずしも減少させることはできないことである。ここでは、軌道追跡を完全に実現することはあきらめ、これらの偏差を含んだ評価関数を抑制することを考える。そのために、まず目標軌道を軌道パラメータによって表現し、目標軌道の形状を修正しそれに追従させるコントローラによって衝突回避を実現する。つぎに、軌道パラメータの調整によって評価関数を抑制するために、評価関数の時間微分に関する min max 問題を解くことで、軌道パラメータの調整則を導出する。なお、幾何偏差を考慮することで、ロボットが前進する力と他ロボットを回避する力が釣り合う場合に、ロボットの停止、すなわちデッドロックが起こる。ここでは、このような場合に増大するペナルティ関数を評価関数に組み入れることで、デッドロック回避を実現する。最後に、3台の移動式ロボットによる数値例によって提案法の有効性を検証する。なお、必要な証明は付録に収録する。

2. ロボットの軌道追跡問題

本論文では Fig. 1 のような複数ロボットの軌道追跡問題を考える。ここでは、 m 次元空間上に存在する n 台のロボットを考える。適当に順番付けられたロボットのうち i 番目のロボットをロボット i と呼ぶことにする。各ロボットは適当なローカル制御によって、つぎの 2 階微分方程式に従って動作するとする。

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

ただし、 $\mathcal{N} = \{1, 2, \dots, n\}$ であり、 $q_i, u_i \in \mathbb{R}^m$ はロボット i の位置と入力を表わす。また、一般性を失うことなく初期時刻を 0 とする。なお、本論文では入力 u_i を設計する際に、各ロボットはグローバルな情報、すなわちすべてのロボットの情報を取得できることを想定する。

ここでは文献 8) をもとに、各ロボットに実現させたい軌道追跡の定式化を行なう。軌道追跡とはロボットを与えられた軌道を定められた方向に決められた速度でたどらせることである。まず、ロボットの所望の運動を定義する。ロボット i がたどるべき軌道の形状が、Fig. 2 のようにパラメータ表示された曲線 $\phi_i: \mathbb{R} \rightarrow \mathbb{R}^m$ で表わされているとする。 ϕ_i を目標軌道と呼び、そのパラメータ $s_i \in \mathbb{R}$ を軌道パラメータと呼ぶことにする。パラメータ s_i は軌道 ϕ_i 上の座標のようなものであり、軌道上の位置は $\phi_i(s_i) \in \mathbb{R}^m$ と表現される。したがって、軌道パラメータ s_i を ∞ まで動かすと、 $\phi_i(s_i)$ はロボットの移動すべき経路をたどることになる。なお、 ϕ_i

Fig. 2 Desired trajectory and assigned velocity

は (1) 式に従うロボットの位置 q_i がたどることができるために、 C^2 級であるとする。ここで、ロボットは軌道 ϕ_i 上を一定速度で進む、またはカーブで速度を落とし直線で速度を上げるなど、それぞれの地点 s_i で取るべき速度が決められているとする。このような指定速度が、Fig. 2 における点線の矢印の大きさのように、関数 $v_i: \mathbb{R} \rightarrow \mathbb{R}_+$ を用いて $v_i(s_i)$ で定められているとする。ここでは、軌道 ϕ_i に対して、

$$\|\phi'_i(s_i)\| = v_i(s_i), \quad \forall s_i \in \mathbb{R} \quad (2)$$

を満たすようにパラメータ s_i が導入されていることを仮定する。ただし、 $\|\cdot\|$ は $\mathbb{R}^m$ 上のユークリッドノルムであり、 $\phi'_i = d\phi_i/ds_i$ である。

(注意 1) パラメータ s_i が ϕ_i 上の座標として導入されているためには、任意の s_i に対して $\phi'_i(s_i) \neq 0$ であればよい。このようにしてパラメータが導入された軌道 $\phi_i(s_i)$ が (2) 式を満たしていない場合は、適当な定数 α_i に対して以下の関数 $\hat{s}_i(s_i)$ を新たな軌道パラメータとすればよい。

$$\hat{s}_i(s_i) = \int_{\alpha_i}^{s_i} \frac{\|\phi'_i(s)\|}{v_i(s)} ds$$

実際、その逆関数 $s_i(\hat{s}_i)$ に対して目標軌道を $\hat{\phi}_i(\hat{s}_i) = \phi_i(s_i(\hat{s}_i))$ とおけば、 $\hat{\phi}_i(\hat{s}_i)$ に対して (2) 式が成立する。

以上で与えた所望の運動に対して、ロボットの軌道追跡問題を定式化する。まず、ロボットの位置 q_i を目標軌道 ϕ_i 上に移動させるという幾何的な目標を考える。これは、軌道パラメータをある連続時間関数 $s_i(t)$ とすると、以下を意味する。

$$\lim_{t \rightarrow \infty} \|q_i(t) - \phi_i(s_i(t))\| = 0 \quad (3)$$

以降、 $\|q_i - \phi_i(s_i)\|$ を幾何偏差と呼ぶことにし、(3) 式を達成することを幾何追従と呼ぶ。つぎに、ロボットを目標軌道 ϕ_i 上において、速度 $v_i(s_i(t))$ で s_i が正の方向に移動させるという運動学的な目標を考える。(3) 式が成り立つとき、Barbalat の補題¹⁴⁾ と (2) 式より、ロボットの速度は

$$\begin{aligned} \|\dot{q}_i(t)\| &\rightarrow \|\dot{\phi}_i(s_i(t))\| = \|\phi'_i(s_i(t))\| |\dot{s}_i(t)| \\ &= v_i(s_i(t)) |\dot{s}_i(t)| \quad \text{as } t \rightarrow \infty \end{aligned} \quad (4)$$

を満たす^(注 1)。これより、 $|\dot{s}_i(t)|$ が 1 に収束すればロボットの速度 $\|\dot{q}_i(t)\|$ は指定速度 $v_i(s_i(t))$ に到達する。さらにロ

(注 1) 厳密には、Barbalat の補題を用いる際の仮定として、関数 $\phi'_i, \dot{q}_i, \dot{s}_i$ が一様連続でなければならない。この条件を満たすためには、そのような目標軌道 ϕ_i が与えられており、(1), (16) 式における u_i と σ_i が有界であるように設計されていけばよい。

ボットの動くべき方向は s_i が正の方向であるため、 $\dot{s}_i(t)$ は 1 に収束すればよい。以上より、ロボットの運動学的な目標はつぎを達成することに帰着する。

$$\lim_{t \rightarrow \infty} |\dot{s}_i(t) - 1| = 0 \quad (5)$$

$|\dot{s}_i - 1|$ を速度偏差と呼び、(5) 式を達成することを速度追従と呼ぶ。以降、(3), (5) 式を実現することを軌道追跡と呼ぶ。なお、軌道追跡を達成するために、入力 u_i のほかに軌道パラメータ s_i を設計することができることに注意されたい。(注意 2) このような軌道追従の重要な応用としてライントレースがある。ライントレースでは、ロボットは走行しているライン (目標軌道) 上でそのラインを検知して走行しており、その先のラインの形状は未知である。また、メモリの関係からそれまで通過したラインの情報を蓄えておくことは難しい。本論文ではこのような状況を考えて、時刻 t において目標軌道 ϕ_i のうち現在地 $\phi_i(s_i(t))$ の情報のみが利用可能であり、その前後の情報は利用できないものとする。

3. 衝突回避が軌道追跡へ及ぼす影響

本章では複数ロボットの衝突回避について考え、これを考慮して軌道追跡を実現することが困難であることを示す。Fig. 1 のように複数のロボットが同一フィールドに存在していると、ロボットは互いに衝突しないように移動しなければならない。ここで、ロボット i は半径 r_i の円形であるとする。このとき、衝突しないことはロボットの全ペアに対して位置 q_i と q_j が距離 $r_i + r_j$ より大きい、すなわちつぎを意味する。

$$\|q_i(t) - q_j(t)\| > r_i + r_j, (i, j) \in \mathcal{L}, \forall t \geq 0 \quad (6)$$

ただし、 $\mathcal{L} = \{(i, j) : i < j, i, j \in \mathcal{N}\}$ はロボットのペア (i, j) の集合である。なお、 $\mathcal{F} \subset \mathbb{R}^{nm}$ を (6) 式を満たす $q = [q_1^T \ q_2^T \ \cdots \ q_n^T]^T \in \mathbb{R}^{nm}$ の集合とし、(6) 式を $q(t) \in \mathcal{F}, \forall t \geq 0$ と表現する。

軌道追跡 (3), (5) を行なっている際に衝突回避 (6) を実現しようとする、軌道追跡を完全に達成することが困難になる場合がある。このような状況として Fig. 3 の 2 つを考える。Fig. 3 (i) のような目標軌道が交差する状況では、後から来たロボット 2 が速度を下げ、ロボット 1 を先にいかせることで衝突回避が実現できる。このとき、ロボットの位置 q_i は目標軌道 ϕ_i 上にあるため、幾何偏差 $\|q_i - \phi_i(s_i)\|$ は増加しない。しかし、ロボット 2 の速度 $\|\dot{q}_2(t)\|$ を落とすことは、(4) 式より $\dot{s}_2(t)$ を 1 から減少させることになり、速度偏差 $|\dot{s}_2 - 1|$ が増大する。一方、Fig. 3 (ii) のような軌道同士が近接している状況では、ロボットの速度を落として衝突回避を行なおうとするとロボットは停止し、デッドロックが起こる。このような場合は、両ロボットは本来の実線の目標軌道 ϕ_i から離れて、破線のような安全な軌道 $\tilde{\phi}_i$ 上を移動させて衝突回避を行わなければならない。このとき、ロボットの動きは $q_i(t) = \tilde{\phi}_i(t) \neq \phi_i(s_i(t))$ となるため、幾何偏差 $\|q_i - \phi_i(s_i)\|$ は増大する。以上より、衝突回避 (6) を実現しようとする

Fig. 3 Passing robots along the reference trajectories

幾何偏差または速度偏差は増大するため、衝突の危険がある状況では軌道追跡 (3), (5) を厳密に満足することはできない。後の議論では、軌道追跡を完全に達成することはあきらめ、これら偏差の増大を認めつつもできるだけ抑制することを考える。そのために、偏差を評価する関数を定義し、この評価関数を抑制するために軌道パラメータ s_i を調整する。

なお、このような方針で速度追従 (5) の達成を諦めたために、 $\dot{s}_i$ が 0 になり (4) 式よりロボットが途中で停止しデッドロックが起る可能性が生じる。デッドロックによってロボットが立ち往生することは避けなければならないため、パラメータ $s_i(t)$ がつぎのデッドロック回避条件を満たすことを要求する。

$$\forall \gamma_i > s_i(0), \exists \tau_i > 0 \text{ s.t. } s_i(\tau_i) = \gamma_i \quad (7)$$

この条件は、パラメータ $s_i(t)$ がどれだけ大きいポイント γ_i にもある時刻 τ_i で到達できることを意味する。これによって、パラメータ $s_i(t)$ は停止・後退するような時刻が存在しても、必ずどこまでも進むことが保証される。

以降の流れはつぎのようである。4 章において Fig. 3 (ii) のような安全な軌道 $\tilde{\phi}_i$ とこの軌道への追従コントローラ u_i の設計を行なう。5 章において幾何・速度偏差およびデッドロックを考慮した評価関数を定義し、それを抑制するパラメータ s_i の調整則を提案する。6 章においてデッドロック回避を保証するための調整則の条件を与える。

4. 軌道修正法による衝突回避を考慮したコントローラ

本章では、衝突回避 (6) のための安全な軌道 $\tilde{\phi}_i$ とこの軌道への追従コントローラ u_i の設計を行なう。これは以下を達成することに帰着する。

$$q_i(t) = \tilde{\phi}_i(t), \tilde{\phi}(s(t)) \in \mathcal{F}, \forall t \geq 0 \quad (8)$$

ただし、修正軌道 $\tilde{\phi}_i(t) \in \mathbb{R}^m$ に対して $\tilde{\phi} = [\tilde{\phi}_1^T \ \tilde{\phi}_2^T \ \cdots \ \tilde{\phi}_n^T]^T$ であり、 $\tilde{\phi}(t) \in \mathcal{F}$ はこの軌道上でロボットが衝突しないことを意味する。このような軌道の設計法として、筆者ら¹²⁾の提案した軌道修正法を採用する。そこで提案されている $\mathcal{C}^2$ 級写像 $\Phi_i : \mathbb{R}^{nm} \rightarrow \mathbb{R}^m$ を用い目標軌道 $\phi_i(s_i(t))$ を元に新しい修正軌道 $\tilde{\phi}_i(t)$ をつぎで与えれば、必ず (8) 式が満たされる。

$$\tilde{\phi}_i(t) = \Phi_i(\phi(s(t))) \quad (9)$$

ただし、 $\phi(s) = [\phi_1^T(s_1) \ \phi_2^T(s_2) \ \cdots \ \phi_n^T(s_n)]^T, s = [s_1 \ s_2 \ \cdots \ s_n]^T$ である。したがって、この軌道上をロボット

が移動すれば衝突回避 (6) が達成される. なお, 写像 Φ_i は $\mathbb{R}^m$ 次元の座標を拡大する写像であり, これによって衝突の恐れのない修正軌道 $\tilde{\phi}_i(t)$ が一意に定まる. またその計算には現時刻での目標軌道の点 $\phi_i(s_i(t))$ のみを使用するため, 本手法はオンラインである. したがって, 注意 2 のように目標軌道の先の形状がわからない場合でも使用できる. ここで, 写像 Φ_i に対して以下を満たす集合 $\mathcal{F}_i$ を回避抑制集合と呼ぶ.

$$\Phi_i(q) = q_i \Leftrightarrow q \in \mathcal{F}_i (\subset \mathcal{F}) \quad (10)$$

この集合はロボット i の回避が発生しないような全ロボットの配置を表わすものであり, このロボットが他機と衝突する危険がなければ $q \in \mathcal{F}_i$ が満たされる.

初期時刻における仮定として, 任意のロボットは衝突の危険がなく, 目標軌道に沿って移動している, すなわち任意の $i \in \mathcal{N}$ に対してつぎが満たされているとする.

$$q_i(0) = \phi_i(\alpha_i), \exists \alpha_i \in \mathbb{R}, q(0) \in \mathcal{F}_i \quad (11)$$

$$\dot{q}_i(0) = \beta_i \phi'_i(\alpha_i), \exists \beta_i \in \mathbb{R} \quad (12)$$

軌道追跡の主要な応用であるライントレースでは, ロボットは他機と衝突しない状況で与えられたライン (目標軌道) 上, あるいはその近くにいるため, これらの初期条件はほとんどの場合満たされる. このとき修正軌道 $\tilde{\phi}_i$ に対して (8) 式を実現する入力 u_i はつぎの定理のように与えられる.

《定理 1》 $\mathcal{C}^2$ 級写像 $\Phi_i: \mathbb{R}^{nm} \rightarrow \mathbb{R}^m$ によって, (9) 式の修正軌道 $\tilde{\phi}_i$ が (8) 第 2 式を満たすとする. また, 初期時刻において q_i が (11), (12) 式を満たしているとする. このとき, パラメータ s_i の初期状態とシステム (1) に対する入力を以下で与えると (8) 式が達成され, 衝突回避 (6) が実現される.

$$s_i(0) = \alpha_i, \dot{s}_i(0) = \beta_i \quad (13)$$

$$u_i^k = \sum_{j \in \mathcal{N}} \left\{ \left(\phi'_j(s_j) \right)^\top \frac{\partial^2 \Phi_i^k}{\partial \phi_j^2}(\phi(s)) \phi'_j(s_j) + \frac{\partial \Phi_i^k}{\partial \phi_j}(\phi(s)) \times \phi''_j(s_j) \right\} \dot{s}_j + \frac{\partial \Phi_i^k}{\partial \phi_j}(\phi(s)) \phi'_j(s_j) \dot{s}_j \quad (14)$$

ただし, ここの上付 k はベクトルの k 番目の成分を表わす.

なお, このコントローラを実際に使用するためには, つぎのように, ロボットの実際の位置と修正軌道に対する追従誤差を収束させるフィードバック制御を加える必要がある.

$$u_i = u_i^* - K_{pi}(q_i - \Phi_i) - K_{di} \left(\dot{q}_i - \sum_{j \in \mathcal{N}} \frac{\partial \Phi_i}{\partial \phi_j} \phi'_j \dot{s}_j \right)$$

ただし, $K_{pi}, K_{di} \in \mathbb{R}^{m \times m}$ は正定行列, u_i^* は (14) 式を $k = 1$ から m まで並べた m 次元ベクトルである.

5. 評価関数を抑制するパラメータ調整則

本章では 3 章の考え方に基づいて, 幾何・速度偏差およびデッドロックを評価する関数を定義し, それを抑制するパラメータ調整則を提案する. まず, このような評価をすべてのロボットに対して行なうつぎの関数を考える.

$$V(s, \dot{s}, t) = \sum_{i \in \mathcal{N}} \left\{ \frac{w_{gi}}{2} \|\Phi_i(\phi(s)) - \phi_i(s_i)\|^2 + \frac{w_{vi}}{2} |\dot{s}_i - 1|^2 + p_i(s_i, t) \right\} \quad (15)$$

ここで, $w_{gi} \geq 0, w_{vi} > 0$ は幾何・速度偏差に対する重み係数, $p_i: \mathbb{R} \times \mathbb{R}_+ \rightarrow \mathbb{R}_+$ はデッドロックに対するペナルティ関数である. 右辺第 1 項は定理 1 によって (8) 式が達成されているときの幾何偏差 $\|q_i - \phi_i(s_i)\| = \|\Phi_i(\phi(s)) - \phi_i(s_i)\|$ を, 第 2 項は速度偏差 $|\dot{s}_i - 1|$ を評価する. また, 第 3 項の $p_i(s_i, t)$ はデッドロックが生じたときに増加するペナルティ関数である. 実際にデッドロック回避 (7) を実現するペナルティ関数の設計については, 6 章において行なう.

以上で定義した評価関数 $V(s, \dot{s}, t)$ を抑制することで, 幾何・速度偏差およびデッドロックをすべて抑制することができる. これらを抑制するバランスは重み係数 w_{gi}, w_{vi} および関数 p_i の選び方によって定まる. ここでは評価関数の抑制のためにパラメータ s_i の調整を行なう. コントローラ (14) の最終項にあるように, s_i の挙動はその 2 階微分係数によって定まる. これより, ここではその調整則をつぎの形で与える.

$$\ddot{s}_i = -k_i(\dot{s}_i - 1) + \sigma_i(s, t), k_i > 0 \quad (16)$$

ここで, 右辺第 1 項は速度偏差を減少させ, 関数 $\sigma_i(s, t) \in \mathbb{R}$ は幾何偏差とペナルティ関数を抑制する. このような抑制効果をもつ $\sigma_i(s, t)$ をどのように設計したらよいかが問題となる. ここでは評価関数 $V(s, \dot{s}, t)$ を抑制することを, その時間微分 $\dot{V}(s, \dot{s}, t)$ を最小化することと解釈し, 関数 $\sigma_i(s, t)$ を設計する. 具体的にはつぎのように, パラメータの速度 $\dot{s}_i$ に関して最大となる $\dot{V}(s, \dot{s}, t)$ を最小化する $\sigma_i(s, t)$ を設計する.

$$\min_{\sigma(s, t) \in \mathbb{R}^n} \max_{\dot{s} \in \mathbb{R}^n} \dot{V}(s, \dot{s}, t) |_{(16), i \in \mathcal{N}} \quad (17)$$

ただし, $\sigma = [\sigma_1, \dots, \sigma_n]^\top$ である. 以上の観点から関数 σ_i はつぎのように設計される.

《定理 2》 $\mathcal{C}^2$ 級写像 $\Phi_i: \mathbb{R}^{nm} \rightarrow \mathbb{R}^m$, 定数 $w_{gi} \geq 0, w_{vi} > 0$ および $\mathcal{C}^1$ 級関数 $p_i: \mathbb{R} \times \mathbb{R}_+ \rightarrow \mathbb{R}_+$ に対して評価関数 $V(s, \dot{s}, t)$ を (15) 式で与える. 軌道パラメータ s_i はある連続関数 $\sigma_i: \mathbb{R}^n \times \mathbb{R}_+ \rightarrow \mathbb{R}$ に対して, (16) 式に従うとする. このとき, $\min \max$ 問題 (17) の解は

$$\sigma_i(s, t) = -\frac{1}{w_{vi}} \left(\sum_{j \in \mathcal{N}} w_{vj} g_{ij}(s) + \frac{\partial p_i}{\partial s_i}(s_i, t) \right) \quad (18)$$

で与えられ, そのときの (17) 式の値は以下で与えられる.

$$\sum_{i \in \mathcal{N}} \left(\sum_{j \in \mathcal{N}} w_{vj} g_{ij}(s) + \frac{\partial p_i}{\partial s_i}(s_i, t) + \frac{\partial p_i}{\partial t}(s_i, t) \right) \quad (19)$$

ただし, スカラ関数 $g_{ij}(s)$ はクロネッカデルタ δ_{ij} と恒等行列 $I_m \in \mathbb{R}^{m \times m}$ に対して以下を表わす.

$$g_{ij}(s) = (\Phi_j(\phi(s)) - \phi_j(s_j))^\top \times \left(\frac{\partial \Phi_j}{\partial \phi_i}(\phi(s)) - \delta_{ij} I_m \right) \phi'_i(s_i) \quad (20)$$

パラメータ調整則 (16) において, 関数 σ_i は (18) 式の括弧内の第 1, 2 項によってそれぞれ幾何偏差, ペナルティ関数の抑制を行なう. 定理 2 から $\dot{V}$ は (18) 式の σ_i に対して最小値 (19) 式をとることがわかる. ここで, (19) 式は正になる可能性のある関数である. したがって, σ_i をどのように設計しても $\dot{V}$ の負定性を保証することはできないため, 評価関数 V は増加する可能性がある. これは 3 章で指摘したように, 衝突回避によって幾何・速度偏差が増加する場合があることに起因している.

(注意 3) 関数 $V(s, \dot{s}, t)$ の時間積分値を評価関数として最小にするという問題も考えることができるが, そのためには目標軌道全体の情報を元にオフラインの運動計画によって s_i を設計することになる. しかし, 注意 2 で述べたように目標軌道としては現在点 $\phi_i(s_i(t))$ の情報しか与えられていないので, このような運動計画はできない. 一方, 提案法は入力 (14) とパラメータ調整則 (16) による動的フィードバック制御則から構成されており, 目標軌道 ϕ_i の先の情報を必要としないオンライン手法である.

(注意 4) ロボットを目標軌道からできるだけ乖離しないように走行させるために, (15) 式において評価関数に幾何偏差 $\|\Phi_i(\phi(s)) - \phi_i(s_i)\|$ を導入した. 付録 A では, 回避抑制集合 $\mathcal{F}_i$ の定義 (10) をもとに, これよりも目標軌道からの乖離をより防止できる評価関数を考える.

6. デッドロック回避と衝突の危険がない場合の追従性

本章では, デッドロック回避 (7) を達成するペナルティ関数 $p_i(s_i, t)$ を設計する. 主なアイディアは, 正に発散するスカラー関数 $\eta_i(t)$ を導入し, 軌道パラメータ s_i がデッドロックを起こしこの関数よりも遅れた場合に増加するペナルティ関数を考えることである. そのために, 非負単調減少関数 $\psi_i: \mathbb{R} \rightarrow \mathbb{R}_+$ を用意してペナルティ関数を

$$p_i(s_i, t) = \psi_i(s_i - \eta_i(t)) \quad (21)$$

とする. デッドロックが生じて $s_i(t)$ が停止すると, 正に発散する関数 $\eta_i(t)$ に対して $s_i(t) - \eta_i(t)$ は負に発散する. このとき, 非負単調減少関数 ψ_i に対して (21) 式は増加する. なお, $\dot{\eta}_i(t)$ を 1 より大きく設定すると, $s_i(t)$ が $\eta_i(t)$ から遅れないことと (5) 式が矛盾する. 以上より, ある $\delta_i > 0$ に対して $\eta_i(t)$ が以下を満たすことを仮定する.

$$\lim_{t \rightarrow \infty} \eta_i(t) = \infty, \dot{\eta}_i(t) \leq 1 - \delta_i, \forall t \geq 0 \quad (22)$$

デッドロック回避 (7) を達成するためには, 軌道パラメータ s_i の遅れが大きくなったときに, (18) 式第 2 項 $\partial p_i / \partial s_i$ の大きさが増加し s_i を前進させればよい. そのような条件は ψ'_i の大きさに依存しており, 十分大きい正数 $\bar{\psi}_i$ に対してつぎを満たされていればよい.

$$\exists \bar{x}_i \in \mathbb{R}, \forall x < \bar{x}_i \text{ s.t. } \psi'_i(x) < -\bar{\psi}_i \quad (23)$$

実際, デッドロック回避を保証するつぎの定理を得る.

《定理 3》 定理 2 の条件がすべて満たされており, 関数 $g_{ij}(s)$ が有界であるとする. また, $\eta_i(t)$ は (22) 式を満たすスカラー関数, $\psi_i(x)$ は

$$\bar{\psi}_i > \sup_{s \in \mathbb{R}^n} \sum_{j \in \mathcal{N}} w_{vj} g_{ij}(s) - w_{vi} k_i \quad (24)$$

を満たす正数 $\bar{\psi}_i$ に対して (23) 式を成立させる非負単調減少関数とする. このとき, ペナルティ関数 $p_i(s_i, t)$ を (21) 式で選べば, デッドロック回避 (7) が達成される.

以上より設計された衝突回避を考慮した軌道追跡コントローラは, (13), (16), (18) 式による軌道パラメータ s_i の調整則と (14) 式による制御入力 u_i から成り立つ. なお, ロボット同士が目標軌道を通っても衝突しない場合には衝突回避の必要がないため, このコントローラによって軌道追跡 (3), (5) が達成されなければならない. このような場合, つぎのようにペナルティ関数のとり方を工夫すれば問題ない.

[系 1] 定理 1, 2, 3 の各条件が満たされており, 非負単調減少関数 $\psi_i(x)$ が次式を満たすとする.

$$\exists \bar{x}_i > \underline{x}_i, \forall x \geq \bar{x}_i \text{ s.t. } \psi'_i(x) = 0 \quad (25)$$

このとき, ある時刻 $t_0 \geq 0$ が存在し $q(t) \in \mathcal{F}_i, \forall i \in \mathcal{N}, \forall t \geq t_0$ が成り立つとき, 軌道追跡 (3), (5) が実現される.

なお, (23), (24), (25) 式を満たす関数 $\psi_i(x)$ として, 任意の正数 a_i と定数 b_i に対して以下が与えられる.

$$\psi_i(x) = \begin{cases} a_i(x - b_i)^2, & x < b_i \\ 0, & x \geq b_i \end{cases} \quad (26)$$

7. 数 値 例

本章では数値例によって提案法の有効性を検証する. フィールドの次元を $m = 2$, ロボットの台数を $n = 3$, 各ロボットの半径を $r_i = 0.05$, 目標軌道 $\phi_i(s_i)$ を以下の直線軌道とする.

$$\phi_i(s_i) = (0, -0.25 + v_1 s_1), (-0.18 + v_2 s_2, 0), \\ (-0.05, 0.2 - v_3 s_3), v_i = 0.03 \quad (27)$$

Fig. 4 はロボットの移動する x - y 平面を表わしており, ロボット大きさは一マス分の円である. この図において目標軌道 (27) は実線で表わされており, ロボットはその上を矢印の方向に動くように指示されている. 初期時刻でロボットは×印のある座標 $(0, -0.25)$, $(-0.18, 0)$, $(-0.05, 0.2)$ において停止しているとする. このとき, 初期条件 (11), (12) は $\alpha_i = \beta_i = 0$ に対して満たされる. これに対して入力 u_i を (14) 式で, 軌道パラメータ s_i の調整則を $k_i = 2$ に対して (13), (16), (18) 式で与える. ここでは, (15) 式の評価関数における幾何・速度偏差に対する重み w_{gi} , w_{vi} とペナルティ関数 $p_i(s_i, t)$ の選び方を変えて, 3 つのシミュレーションを行なう.

Case 1 まず速度偏差のみを考慮するためその重みを $w_{vi} = 1$ とし, 幾何偏差に対する重み w_{gi} とペナルティ関数 $p_i(s_i, t)$ を 0 とする. ロボットの軌道を Fig. 4 上図に点線で示す. この場合は幾何偏差を考慮していないため, 目

Fig. 4 Trajectories of the robots in Cases 1, 2 and 3

標軌道である直線からかなり乖離して衝突回避を実現している。このときのパラメータの速度 $\dot{s}_i$ の時間変位を Fig. 5 上図に示す。すべてのパラメータ速度は単調に1に収束しており、速度偏差 $|\dot{s}_i - 1|$ は0になっていることがわかる。

Case 2 つぎに幾何偏差も考慮するためにその重みを $w_{gi} = 1$ とし、ほかを Case 1 と同じとする。ただし、ロボットの目標軌道からの乖離を抑制するために、評価関数 (15) において幾何偏差 $\|\Phi_i(\phi(s)) - \phi_i(s_i)\|^2$ の代わりに、(A.2) 式の関数 $f_i(q)$ に対して $f_i(\phi(s))$ を用いた。ここで、関数 $\xi_{ij}(x)$ を

$$\xi_{ij}(x) = \begin{cases} -(x - 0.25)^3/3, & x < 0.25 \\ 0, & x \geq 0.25 \end{cases}$$

とした。このときのロボットの軌道を Fig. 4 中図に点線で示

Fig. 5 Parameter velocities in Cases 1, 2 and 3

す。これより、すべてのロボットは目標軌道からほとんど離れずに直進していることがわかる。ただし、 y 軸と平行に進行しているロボット 1, 3 は幾何偏差を増加させないために、目標軌道から離れずに座標 $(-0.05, 0.03)$, $(0, -0.06)$ 付近で静止している。このときのパラメータの速度 $\dot{s}_i$ を Fig. 5 中図に実線、点線、一点鎖線で示す。これより $\dot{s}_1, \dot{s}_3$ が0に収束し、デッドロックが起こっていることがわかる。

Case 3 最後にデッドロックを考慮するために、ペナルティ関数 $p_i(s_i, t)$ を (21) 式で与え、ほかを Case 2 と同じとする。ただし、定理 3, 系 1 の条件を満たすように、 $\eta_i(t) = 0.8v_it$ とし、 $\psi_i(x)$ を $a_i = 0.5, b_i = -0.2$ に対して (26) 式とする。このときのロボットの軌道を Fig. 4 下図に点線で示す。これより、Case 2 と同様にロボット 2 は直線の目標軌道からほとんど離れずに直進していることがわかる。ただし、Case 2 でデッドロックを起こした部分で、ロボット 1, 3 は目標軌道から少し離れ衝突回避を行ない進んでいる。このときのパラメータの速度 $\dot{s}_i$ を Fig. 5 下図に実線、点線、一点鎖線で示す。これより、 $\dot{s}_1, \dot{s}_3$ はいったん0になるが、その後、デッドロック回避機能によって1に収束していることがわかる。

以上の結果から、本手法を用いると、重み係数やペナルティ関数の調整によってさまざまなロボットの動きを実現できることがわかった。さらに、Case 3 のように目標軌道からできるだけ離れずに、しかもデッドロックの起こらないロボットの移動を実現することができた。

8. おわりに

本論文では、複数ロボットに対して軌道追跡を行なう際の

衝突回避問題について検討した。ここでは軌道追跡を完全に実現することはあきらめ、幾何・速度偏差を含んだ評価関数を抑制することを考えた。そのために、まず目標軌道を軌道パラメータによって表現し、目標軌道の形状を修正しそれに追従させるコントローラによって衝突回避を実現した。つぎに、軌道パラメータの調整によって評価関数を抑制するために、評価関数の時間微分に関する min max 問題を解くことで、軌道パラメータの調整則を導出した。また、デッドロックが起こったときに増大するペナルティ関数を評価関数に組み入れることで、デッドロック回避を実現した。最後にシミュレーションによって、設定パラメータの違いによってロボットの挙動が異なることを確認した。この際、パラメータを適切に選ぶことで、目標軌道からはできるだけ離れずに、しかもデッドロックの起こらないロボットの移動を実現した。

参 考 文 献

- 1) J.C. Latombe: Robot Motion Planning, Kluwer Academic Publishers (1991)
- 2) S.M. LaValle and S.A. Hutchinson: Optimal motion planning for multiple robots having independent goals, *IEEE Trans. Robotics and Automation*, **14**-6, 912/925 (1998)
- 3) C.G. Cassandras and W.Li: A receding horizon approach for solving some cooperative control problems: In *Proc. of 41th IEEE Conf. on Decision and Control*, 3760/3765 (2002)
- 4) A. Richards and J.P. How: Aircraft trajectory planning with collision avoidance using mixed interger linear programming: In *Proc. of American Control Conference*, 1936/1941 (2002)
- 5) Y. Hao, A. Davari and A. Manesh: Differential flatness-based trajectory planning for multiple unmanned aerial vehicles using mixed-integer linear programming: In *Proc. of American Control Conference*, 104/109 (2005)
- 6) 新井, 太田: 仮想的なインピーダンスを用いた複数移動ロボット系の動作計画, 日本ロボット学会誌, **11**-7, 1039/1046 (1993)
- 7) S.G. Loizou and K.J. Kyriakopoulos: Closed loop navigation for multiple holonomic vehicles: In *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2861/2866 (2002)
- 8) T.I. Fossen R. Skjetne and P. Kokotović: Output maneuvering for a class of nonlinear systems: In *Proc. of IFAC 15th Triennial World Congress* (2002)
- 9) T.I. Fossen R. Skjetne and P. Kokotović: Robust output maneuvering for a class of nonlinear systems, *Automatica*, **40**-3, 373/383 (2004)
- 10) J.P. Hespanha, A.P. Aguiar and P. Kokotović: Path-following for nonminimum phase systems removes performance limitations, *IEEE Trans. Automatic Control*, **50**-2, 234/239 (2005)
- 11) J.P. Hespanha, A.P. Aguiar and P. Kokotović: Limits of performance in reference-tracking and path-following for nonlinear systems: In *Proc. of the 16th IFAC World Congress* (2005)
- 12) 桜間, 中野: 衝突回避を考慮した複数ロボットのオンライン目標軌道修正—ロボット間の相対角度を用いた手法—, 計測自動制御学会論文集, **43**-11, 1037/1046 (2007)
- 13) S. Akella and S. Hutchinson: Coordinating the motions of multiple robots with specified trajectories: In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 624/631 (2002)
- 14) H.K. Khalil: Nonlinear Systems, 323, Prentice Hall (2002)

《付 録》

A. 幾何偏差の評価関数の再考

目標軌道からの乖離をより防止するために、(10) 式で定義した衝突回避集合 $\mathcal{F}_i$ をもとに、(15) 式の $V(s, \dot{s}, t)$ における幾何偏差の評価について再考する。(10) 式より

$$\|\Phi_i(\phi(s)) - \phi_i(s_i)\| = 0 \Leftrightarrow \phi(s) \in \mathcal{F}_i$$

が成り立つ。これより、 $\phi(s) \in \mathcal{F}_i$ の場合には、評価関数 V の幾何偏差に関する第 1 項は 0 となる。しかし、 $\phi(s)$ が $\mathcal{F}_i$ の縁にあり修正が行なわれる直前でも V は増加することはない。 $\phi(s)$ がこのような状況にあることは好ましくないが、(15) 式の V ではこの状況の評価することはできない。ここでは、このような評価を行なうために、幾何偏差 $\|\Phi_i(\phi(s)) - \phi_i(s_i)\|$ の代わりに $\phi(s)$ が $\mathcal{F}_i$ の縁にある場合に増加する関数を考える。まず、ベクトル $q \in \mathbb{R}^{nm}$ が $\mathcal{F}_i$ においてどれだけ内部に存在するかを評価する関数 $f_i(q)$ の定義を行なう。関数 $f_i(q)$ が満たすべき条件は、その値が小さいと q が $\mathcal{F}_i$ の内部にある、つまり以下を満たすことである。

$$\{q : f_i(q) \leq c_i\} \subset \mathcal{F}_i, \exists c_i \geq 0 \quad (\text{A.1})$$

この条件を満たす $f_i(q)$ に対する $f_i(\phi(s))$ を幾何偏差の代わりにすれば、所望の評価関数が得られる。すなわち、(15) 式において、第 1 項の $\|\Phi_i(\phi(s)) - \phi_i(s_i)\|^2$ を $f_i(\phi(s))$ で置き換えれば、 $\mathcal{F}_i$ の縁の部分 $\mathcal{F}_i \setminus \{q : f_i(q) = 0\}$ に $\phi(s)$ があると評価関数 V は増加する。この際のパラメータの調整則 (16) は、(18) 式で与えられる σ_i において $g_{ij}(s) = \partial f_i / \partial \phi_j(\phi(s)) \phi'_i(s_i) / 2$ に変更したもので与えられる。これを用いることで、より幾何偏差を抑制することができる。

文献 12) で提案した写像に対して、(A.1) 式を満たす $f_i(q)$ を与える。この際の回避抑制集合は任意の正数 e_{ij} に対して

$$\mathcal{F}_i \supset \{q : \|q_i - q_j\| \geq r_i + r_j + e_{ij}, \forall (i, j) \in \mathcal{L}\}$$

を満たす。このとき、以下の関数が (A.1) 式を満たす。

$$f_i(q) = \sum_{(i,j) \in \mathcal{L}} \xi_{ij}(\|q_i - q_j\|) \quad (\text{A.2})$$

ただし、 $\xi_{ij} : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ は $\xi_{ij}(x) = 0, x > r_i + r_j + e_{ij}$ を満たす非負単調減少関数である。

B. 定理 1 の証明

(11) 第 1 式、(13) 式から $q_i(0) = \phi_i(s_i(0))$ であり、これと (9)、(10)、(11) 第 2 式より $q_i(0) = \tilde{\phi}_i(0)$ となる。また、 $\phi(s) \in \mathcal{F}_i$ に対して (9) 式の時間微分を計算すると、

$$\dot{\tilde{\phi}}_i = \sum_{j \in \mathcal{N}} \frac{\partial \Phi_i}{\partial \phi_j} \phi'_j(s_j) \dot{s}_j = \sum_{j \in \mathcal{N}} \delta_{ij} \phi'_j(s_j) \dot{s}_j = \phi'_i(s_i) \dot{s}_i$$

を得る。ここで、第 2 等式において (10) 式を用いた。初期時刻において、(11) 第 2 式よりこの式が成り立つことと (12)、(13) 式より、以下が成り立つ。

$$\dot{\phi}_i(0) = \phi'_i(s_i(0))\dot{s}_i(0) = \phi'_i(\alpha_i)\beta_i = \dot{q}_i(0)$$

以上より, $\tilde{\phi}_i$ と q_i は初期状態が一致する. さらに, (9) 式右辺の2階時間微分を行なうと (14) 式に帰着されるため, (1) 式より $\ddot{q}_i = \ddot{\phi}_i$ が成り立つ. したがって, 初期時刻以降もこれらは一致し (8) 式が成り立つ.

C. 定理2の証明

(17) 式を達成する関数 σ_i とこの式の値を導出する. (15) 式の V を時間微分し (16) 式を代入する. さらに, それを $\dot{s}_i$ と σ_i についての2次式として順にまとめると以下を得る.

$$\begin{aligned} \dot{V} &= \sum_{i \in \mathcal{N}} \left\{ w_{gi}(\Phi_i - \phi_i)^\top \frac{\partial}{\partial s} (\Phi_i - \phi_i) \dot{s} \right. \\ &\quad \left. + w_{vi}(\dot{s}_i - 1)\dot{s}_i + \frac{\partial p_i}{\partial s_i} \dot{s}_i + \frac{\partial p_i}{\partial t} \right\} \\ &= \sum_{i \in \mathcal{N}} \left\{ -w_{vi}k_i \left(\dot{s}_i - 1 - \frac{\hat{\sigma}_i}{2k_i} \right)^2 + \frac{w_{vi}\hat{\sigma}_i^2}{4k_i} \right. \\ &\quad \left. + \sum_{j \in \mathcal{N}} w_{gj}g_{ij} + \frac{\partial p_i}{\partial s_i} + \frac{\partial p_i}{\partial t} \right\} \quad (\text{C.1}) \end{aligned}$$

ただし, $\hat{\sigma}_i$ は (18) 式の左右辺の差を表わす. 以上より (17) 式を計算すると, $\dot{V}$ を最大化する $\dot{s}_i$ は (C.1) 式第1項が0となる場合, それを最小化する σ_i は第2項が0の場合である. このときの σ_i は (18) 式で与えられ, (17) 式は (C.1) 式の第3項以降, つまり (19) 式で与えられる.

D. 定理3の証明

ここでは, パラメータ s_i がデッドロック回避 (7) を満たさないことを仮定し, 矛盾を導く. このとき, s_i は (16) 式に従う連続関数であることから以下が成り立つ.

$$\exists \gamma_i > s_i(0), \forall t \geq 0 \text{ s.t. } s_i(t) < \gamma_i. \quad (\text{D.1})$$

このような γ_i に対して, (22) 第1式より以下を得る.

$$\exists t_1 \geq 0, \forall t > t_1 \text{ s.t. } \eta_i(t) > \gamma_i - \underline{x}_i \quad (\text{D.2})$$

このような t_1 に対して, (D.1), (D.2) 式より以下を得る.

$$s_i(t) - \eta_i(t) < \underline{x}_i, \forall t > t_1$$

これより, (23) 式から以下を得る.

$$\psi'_i(s_i(t) - \eta_i(t)) < -\bar{\psi}_i, \forall t > t_1 \quad (\text{D.3})$$

(D.3) 式左辺が (18) 式の $\partial p_i / \partial s_i(s_i, t)$ であることに注意すると, (18), (24) 式から時刻 t_1 以降, 以下が成り立つ.

$$\begin{aligned} \sigma_i(s(t), t) &> -\frac{1}{w_{vi}} \left(\sup_{s \in \mathbb{R}^n} \sum_{j \in \mathcal{N}} w_{vj}g_{ij}(s) - \bar{\psi}_i \right) \\ &\geq -\frac{1}{w_{vi}} (w_{vi}k_i - \varepsilon_i) \quad (\text{D.4}) \end{aligned}$$

ただし, 正数 ε_i は (24) 式の左右辺の差を表わす. (16), (D.4) 式よりパラメータ s_i は以下を満たす.

$$\frac{d}{dt} \left(\dot{s}_i(t) - \frac{\varepsilon_i}{k_i w_{vi}} \right) > -k_i \left(\dot{s}_i(t) - \frac{\varepsilon_i}{k_i w_{vi}} \right) \quad (\text{D.5})$$

ここで微分方程式の比較定理¹⁴⁾を用いると, (D.5) 式は

$$\dot{s}_i(t) > \frac{\varepsilon_i}{k_i w_{vi}} + \left(\dot{s}_i(t_1) - \frac{\varepsilon_i}{k_i w_{vi}} \right) e^{-k_i(t-t_1)}$$

に帰着する. これらの両辺を積分区間 t_1 から t まで積分すると以下を得る.

$$\begin{aligned} s_i(t) &> s_i(t_1) + \frac{\varepsilon_i}{k_i w_{vi}}(t - t_1) - \frac{1}{k_i} \left(\dot{s}_i(t_1) - \frac{\varepsilon_i}{k_i w_{vi}} \right) \\ &\quad \times (e^{-k_i(t-t_1)} - 1) \quad (\text{D.6}) \end{aligned}$$

(D.6) 式右辺のうち, 第2項は正に発散, 第3項は0に収束するため, $s_i(t)$ は正に発散する. これは (D.1) 式に矛盾するため, デッドロック回避 (7) が成り立つ.

E. 系1の証明

定理1の証明と同様にして, $q(t) \in \mathcal{F}_i$ のとき $\phi_i(s_i(t)) = \tilde{\phi}_i(t)$ となる. これと (8) 式より幾何追従 (3) が達成される. また, このとき (20) 式より $g_{ij}(s(t)) = 0, \forall t \geq t_0$ となる. つぎに, 速度追従 (5) を示す. ψ_i が単調減少関数であることと $g_{ij}(s(t)) = 0$ より (18) 式の σ_i は非負である. 以上から, 比較定理¹⁴⁾を用いると (16) 式よりつぎを得る.

$$\dot{s}_i(t) \geq 1 + (\dot{s}_i(t_0) - 1)e^{-k_i(t-t_0)} \quad (\text{E.1})$$

(E.1) 式と (22) 第2式の左右辺を引き, 区間 t_0 から t まで積分するとつぎを得る.

$$\begin{aligned} s_i(t) - \eta_i(t) &\geq s_i(t_0) - \eta_i(t_0) \\ &\quad - \frac{1}{k_i} (\dot{s}_i(t_0) - 1)(e^{-k_i(t-t_0)} - 1) + \delta_i(t - t_0) \end{aligned}$$

右辺の第2項は0に収束, 第3項は正に発散するため, $s_i(t) - \eta_i(t)$ は正に発散する. したがって, ある時刻 $t_2 \geq t_0$ 以降, $s_i(t) - \eta_i(t) \geq \bar{x}_i$ が成り立つ. このとき, (25) 式より $\psi'_i(s_i(t) - \eta_i(t)) = 0$ となる. このとき $\sigma_i(s(t), t) = 0$ であるため, (16) 式より $\dot{s}_i(t)$ は1に収束する. 以上より, (5) 式が成り立つため, 題意は満たされた.

[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システム科学専攻博士後期課程修了. 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電子工学専攻助手, 2007年同助教, 現在に至る. 博士(情報学). マルチロボットシステム, 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, 電気学会の会員.

中野 和司 (正会員)

1980年九州大学大学院工学研究科電気工学専攻博士課程単位修得退学. 九州大学助手, 福岡工業大学助教授, 教授を経て, 99年より電気通信大学電気通信学部電子工学科教授. 工学博士. システム同定・制御に関する研究に従事. システム制御情報学会, 電気学会, 電子情報通信学会, IEEEなどの会員.