衝突回避問題に適合した分枝限定法に基づく複数移動体の編隊制御

根 和 幸*・福島 宏 明*・松 野 文 俊*

Multi-vehicle Formation Control Based on Branch-and-bound Method
Compatible with Collision Avoidance Problem

Kazuyuki KON*, Hiroaki FUKUSHIMA* and Fumitoshi MATSUNO*

This paper presents a new branch-and-bound (B&B) method to reduce the computational burden of online optimization in multi-vehicle formation control with collision avoidance that is formulated by model predictive control. One of the possible reasons for extremely heavy computation time of the standard B&B algorithm is that so many subproblems are generated since integer constraints are rarely satisfied in “relaxed” problems. The purpose of the proposed method is to decrease subproblems based on a new branching rule taking into account the inherent properties of collision avoidance problems. Numerical examples and experiments show that the proposed method drastically reduces computation time of online optimization for multi-vehicle formation control.

Key Words: formation control, collision avoidance, branch-and-bound method, model predictive control

1. はじめに

近年、複数の移動体が互いの相対位置を制御する“編隊制御”に関する研究が精力的に行なわれている^{1)~4)}。編隊制御における課題の1つに隊列変化に伴う移動体同士の衝突回避が挙げられる。このような衝突回避を考慮した編隊制御系の設計法として、モデル予測制御が注目を集めている^{5), 6)}。モデル予測制御とは、オンラインで最適制御問題を解くことにより、有限時間未来までの開ループ制御入力を決定する手法である。衝突回避は移動体の相対位置に関する0-1変数を含む不等式制約条件として記述できるため、制御入力を決定するための最適化問題は混合整数計画問題となる^{7)~9)}。したがって、モデル予測制御を用いることにより、衝突回避の理論的な保障を与えることが可能である反面、予測ホライズンや移動体の増加とともに計算量が急激に増加し、オンラインで解くことが困難となるという問題点がある。

モデル予測制御を衝突回避問題に適用する際、どの移動体に対して、いつ衝突回避が必要となるかは最適制御問題を解くまで未知であるため、予測を行なうすべての時刻に対して、すべての移動体との衝突回避制約が課せられる。したがって、移動体が極度に密集している場合を除き、衝突回避のための混合整数計画問題の中には、実際には不必要な0-1変数が数

多く含まれている。ところが、標準的な分枝限定法¹⁰⁾を用いると、このような衝突回避問題の特性は考慮されておらず、すべての0-1変数に対して同様の分枝操作を行なうため、衝突回避が必要ない状況でも多くの子問題が生成される。このことから、衝突回避問題に特化した新たな分枝限定法を考へることにより、不必要な子問題の生成を抑え、計算量を低減できるのではないかと考えられる。

本論文の目的は、衝突回避問題に適した分枝限定法を提案し、モデル予測制御に基づく複数移動体の編隊制御の計算量の低減化を行なうことである。標準的な分枝限定法では整数条件を緩和した“連続緩和問題”に基づいて分枝操作を行なうのに対して、提案する手法では衝突回避制約を除いた最適化問題を緩和問題として用いる。この緩和問題を解いて得られた軌道上で衝突が起きる場合には、衝突回避を行なう方向を0-1変数を用いて表現した制約条件が追加された新たな子問題が生成される。一方、得られた軌道上で衝突が起きない場合には、最適解が得られていることになるため、子問題は終了する。したがって、提案手法では衝突回避が必要な時刻、および移動体に対してのみ子問題が生成されるため、標準的な方法と比較して子問題の数が減少し、計算量の低減化が期待される。さらに、提案手法の有効性を数値例および実験により検証する。

2. 編隊制御における衝突回避問題

本論文では以下の n 台の二輪車両型の移動体を考える。

$$\dot{x}_i = v_i \cos \theta_i, \quad \dot{y}_i = v_i \sin \theta_i, \quad \dot{\theta}_i = \omega_i \quad (1)$$

ただし、 v_i , ω_i , (x_i, y_i, θ_i) はそれぞれ移動体 i ($i = 1, \dots, n$)

* 電気通信大学知能機械工学科 調布市調布ヶ丘 1-5-1

* Department of Mechanical Engineering and Intelligent Systems, University of Electro-Communications, 1-5-1 Chofugaoka, Chofu

(Received June 29, 2007)

(Revised October 29, 2007)

Fig. 1 Wheeled vehicle

Fig. 3 Collision avoidance region

Fig. 2 Leader and Follower

の並進速度、角速度、絶対座標系における位置・姿勢である (Fig. 1 参照)。また、編隊の基準となる以下のような二輪車両型の移動体を考える。

$$\dot{x}_r = v_r \cos \theta_r, \quad \dot{y}_r = v_r \sin \theta_r, \quad \dot{\theta}_r = \omega_r \quad (2)$$

ただし、 v_r , ω_r , (x_r, y_r, θ_r) はそれぞれ基準移動体の並進速度、角速度、絶対座標系における位置・姿勢である。以下では、(1), (2) 式の二輪車両をそれぞれ、“Follower i ”, “Leader” と呼ぶものとする。

Follower i の目標位置は Leader に固定した移動座標系 (r, l) において (r_i, l_i) と表現される。すなわち Follower i の目標軌道は絶対座標系において次式のように表わされる。

$$z_i^d := \begin{bmatrix} x_r + r_i \sin \theta_r + l_i \cos \theta_r \\ y_r - r_i \cos \theta_r + l_i \sin \theta_r \end{bmatrix} \quad (3)$$

編隊制御では、Follower 同士が衝突することなく、おのおのの Follower が

$$z_i := \begin{bmatrix} x_{vi} \\ y_{vi} \end{bmatrix} = \begin{bmatrix} x_i + d \cos \theta_i \\ y_i + d \sin \theta_i \end{bmatrix} \quad (4)$$

で表わされる制御点を目標軌道 (3) に追従させることが目的となる^{1)~4), 7)~9)} (Fig. 2 参照)。

ここでは、Follower 同士の衝突回避を考慮するため、Follower i と j が衝突しないための十分条件が二輪車両の大きさに基づき、つぎのように与えられているものとする (Fig. 3 参照)。

$$\|z_i - z_j\|_\infty \geq \psi, \quad \forall j \neq i \quad (5)$$

なお、(5) 式の衝突回避制約はつぎの線形制約で表わされることが知られている¹⁵⁾。

$$\begin{aligned} x_{vi} - x_{vj} &\leq \Psi \kappa_{ij1} - \psi, & -x_{vi} + x_{vj} &\leq \Psi \kappa_{ij2} - \psi \\ y_{vi} - y_{vj} &\leq \Psi \kappa_{ij3} - \psi, & -y_{vi} + y_{vj} &\leq \Psi \kappa_{ij4} - \psi \\ \sum_{p=1}^4 \kappa_{ijp} &\leq 3 \end{aligned} \quad (6)$$

ただし、 κ_{ijp} は 0, 1 のみをとる変数、 Ψ は z_i のとりうる値よりも十分大きな正数である。

衝突回避を考慮した編隊制御問題に対してモデル予測制御に基づく手法が数多く提案されているが、衝突回避を考慮する場合、共通して (6) 式のような制約条件を含む混合整数計画問題を解く^{7)~9)}。本論文の目的は、衝突回避問題に適した分枝限定法を提案し、上記のようなモデル予測制御に基づく複数移動体の編隊制御の計算量の低減化を行なうことである。以下では参考文献 9) の方法に基づいて議論を進めていくが、提案する分枝限定法は他の方法にも適用可能である。

3. 編隊制御則

参考文献 9) の手法は、フィードバック線形化とモデル予測制御に基づいた編隊制御手法である。3.1 節でフィードバック線形化について説明し、3.2 節でモデル予測制御に基づいた衝突回避の概略を述べる。

3.1 フィードバック線形化

まず、(2), (3) 式より

$$\begin{aligned} \dot{z}_i^d &= F_i u_r \\ F_i &:= \begin{bmatrix} \cos \theta_r & r_i \cos \theta_r - l_i \sin \theta_r \\ \sin \theta_r & r_i \sin \theta_r + l_i \cos \theta_r \end{bmatrix}, \quad u_r := \begin{bmatrix} v_r \\ \omega_r \end{bmatrix} \end{aligned} \quad (7)$$

が成り立つ。また、(1), (4) 式より

Fig. 4 Timing of optimization and input application. A mark • represents time when an optimization problem is solved, and the arrows represent the periods when optimal control sequences are applied

$$\dot{z}_i = G_i u_i \quad (8)$$

$$G_i := \begin{bmatrix} \cos \theta_i & -d \sin \theta_i \\ \sin \theta_i & d \cos \theta_i \end{bmatrix}, \quad u_i := \begin{bmatrix} v_i \\ \omega_i \end{bmatrix}$$

が成り立つ。(7), (8) 式より, 追従偏差 $e_i := z_i - z_i^d$ の時間微分は次式で表わされる.

$$\dot{e}_i = G_i u_i - F_i u_r \quad (9)$$

ここで, α_i をシステムの新たな入力として Follower i の制御入力 u_i を

$$u_i = G_i^{-1}(-\lambda e_i + F_i u_r + \alpha_i) \quad (10)$$

とすると, (9), (10) 式より閉ループ系は

$$\dot{e}_i = -\lambda e_i + \alpha_i \quad (11)$$

となる. ただし $\lambda > 0$ は設計パラメータである. 次節のモデル予測制御に基づいた手法は閉ループ系 (11) に対して適用され, α_i を操作することで衝突回避を行なう.

3.2 モデル予測制御による衝突回避

衝突回避アルゴリズムでは, おおのこの Follower が更新周期 δ ごとに順番に後述の衝突回避制約を含んだ最適化問題を解く. 最適化問題を解いて得られた予測軌道は他の Follower に送信され, 得られた入力はつぎに最適化問題を解くまでの区間適用される. いい換えると, 各 Follower はその他の Follower から送信された予測軌道情報を用い, それらの予測軌道と衝突が起きない軌道を最適化問題を解くことで決定する. Fig. 4 は各 Follower の最適化および入力適用のタイミングを示したものである. ●印は最適化を行なう時刻を, 矢印は最適化により得られた入力を適用する区間を示したものである. このように時系列に各 Follower が最適化を行なうことにより衝突回避や目標値への収束性の理論的保証が容易になる⁹⁾.

Follower i の最適制御問題 P_0 は以下のように表わされる.

$$\min_{\hat{\alpha}_i} \sum_{\tau=k+1}^{k+N} \hat{\alpha}_i(\tau-1|k)^T R \hat{\alpha}_i(\tau-1|k) \quad (12)$$

subject to

$$\hat{e}_i(\tau|k) = -\lambda \hat{e}_i(\tau-1|k) + \hat{\alpha}_i(k), \quad \hat{e}_i(k|k) = e_i(t) \quad (13)$$

$$\|\hat{e}_i(\tau|k) - \hat{e}_j(\tau|k) + \mu_i(\tau|k) - \mu_j(\tau|k)\|_\infty \geq \psi \quad (14)$$

$$\|-\lambda \hat{e}_i(\tau|k) + \hat{\alpha}_i(\tau|k)\|_\infty \leq \eta \quad (15)$$

$$\|\hat{e}_i(k+N|k)\|_\infty \leq \gamma_i \quad (16)$$

$$\forall j \neq i, \quad \tau = k+1, k+2, \dots, k+N$$

ただし, 最適制御問題を解く時刻を $t = k\delta$, 予測ホライズンを N とし, $\hat{e}_i(\tau|k)$ は時刻 $t = k\delta$ における $e_i(\tau\delta)$ の予測値を表わしている. なお, 予測ホライズン N は $N \geq n\delta$ を満たすように選ぶ必要がある. また

$$\mu_i(\tau|k) := \begin{bmatrix} \sin \theta_r(\tau|k) & \cos \theta_r(\tau|k) \\ -\cos \theta_r(\tau|k) & \sin \theta_r(\tau|k) \end{bmatrix} \begin{bmatrix} r_i \\ l_i \end{bmatrix} \quad (17)$$

であり, R は任意の正定行列である. (13) 式の等式制約は (11) 式のモデルを離散化したものであり, e_i を予測するために導入されている. (14) 式は Follower i とそれ以外のすべての Follower $j(j \neq i)$ との間の衝突回避制約 (5) で, (6) 式で表わされる形式に変換される. また, $\hat{e}_j$ は Follower $j(\forall j \neq i)$ から送信された予測データである. (15) 式は (10) 式の制御入力 u_i を制限するため導入した入力制約, (16) 式は閉ループ系の安定性を保証するために導入した終端制約である. なお, η, γ_i は設計者が定める正数である. アルゴリズムの詳細および可解性, 安定性に関する議論については参考文献 9) を参照されたい.

4. 従来の分枝限定法の問題点

前章の最適制御問題 P_0 は, (14) 式の衝突回避制約が 0-1 変数を含む (6) 式の形式に変換されるため以下のような混合整数二次計画問題となる.

$$\min_{\hat{\alpha}_i} \sum_{\tau=k+1}^{k+N} \hat{\alpha}_i(\tau-1|k)^T R \hat{\alpha}_i(\tau-1|k)$$

subject to eq. (13), (15), (16)

$$\begin{bmatrix} \hat{z}_i(\tau|k) - \hat{z}_j(\tau|k) \\ -\hat{z}_i(\tau|k) + \hat{z}_j(\tau|k) \end{bmatrix} \leq \Psi \begin{bmatrix} \kappa_{ij1}(\tau|k) \\ \kappa_{ij3}(\tau|k) \\ \kappa_{ij2}(\tau|k) \\ \kappa_{ij4}(\tau|k) \end{bmatrix} - \begin{bmatrix} \psi \\ \psi \\ \psi \\ \psi \end{bmatrix}$$

$$\sum_{p=1}^4 \kappa_{ijp}(\tau|k) \leq 3$$

$$\forall j \neq i, \quad \tau = k+1, k+2, \dots, k+N$$

ただし, $\hat{z}_i(\tau|k) = \hat{e}_i(\tau|k) + \mu_i(\tau|k)$ で, 連続値変数の次元は $2N$, 0-1 変数の次元は $4N(n-1)$ である. この問題は 0-1 変数を固定すると二次計画問題となり, 原理的には 0-1 変数のすべての組み合わせを列挙することにより最適解を得ることができる. 混合整数計画問題の標準的な解法である分枝限定法では, 以下のような限定操作と分枝操作を繰り返すことにより, 最適解が得られる可能性のない不必要な組み合わせをできるだけ省略することにより, 計算時間を短縮することを目的としている¹⁰⁾.

(分枝操作) 一部の 0-1 変数を固定した子問題を生成する.

具体的には, 整数条件 $\kappa = 0$ or $\kappa = 1$ を $0 \leq \kappa \leq 1$ に緩和した最適化問題 (連続緩和問題) を解き, 整数条件を満

たさなかった0-1変数を1つ選んで0と1に固定することにより、新たな2つの子問題を生成する。

(限定操作) ある子問題の連続緩和問題の最適値は、その子問題の最適値の下界を与える。したがって、連続緩和問題の最適値がそれまでに得られた最適値(暫定値)よりも大きければ、その子問題からは最適解を得られる可能性がないため、その子問題は終端する。その他、連続緩和問題の解がすべて整数条件を満たした場合や、解が得られなかった場合も終端する。最終的にすべての子問題が終端したときの暫定解が元問題 P_0 の最適解となる。

以上のように分枝限定法では子問題が早く終端すればするほど、計算量が短縮できることがわかる。ところが、上記の方法は一般的な混合整数二次計画問題に適用可能である反面、前節の最適化問題 P_0 に対して子問題が早く終端する根拠はなく、最悪の場合、すべての0-1変数の組み合わせを列挙するのと同様の計算時間がかかってしまう。

一方、移動体が極度に密集している場合を除いて、すべての移動体に対する衝突回避を常に行なわなければならないという状況は稀であるため、最適化問題 P_0 の衝突回避制約の中には実際には不必要な0-1変数が数多く含まれている。標準的な分枝限定法はすべての0-1変数に対して同様の分枝操作を行なうため、衝突回避が実際には必要ない状況で、不必要な子問題が数多く生成されていると予想される。この問題点が顕著に表れている数値例をつぎに示す。

[例] Follower を2台、予測ホライズンを6とし、Follower 1,2の初期状態をLeaderに固定した移動座標系においてそれぞれ $(-1.0, -3.0, \pi/2)$, $(1.0, -3.0, \pi/2)$ とし、目標位置をそれぞれ $(-1.0, -1.0)$, $(1.0, -1.0)$ とする (Fig. 5 参照)。なお、その他のパラメータは $v_r = 0.0$, $w_r = 0.0$, $\psi = 0.5$, $\Psi = 30$, $\delta = 0.25$, $\lambda = 0.3$, $\gamma_i = 0.1$, $\eta = 0.4$, $R = I_{2N}$ とする。この例では、Fig. 5 のように l 軸と並行に直進する軌道が最適解であり、衝突回避が必要ないことは明らかである。実際、最適化問題から衝突回避制約を除いてモデル予測制御を実装すると、衝突することなく目標の編隊形状が達成でき、各時刻の最適化問題にかかった計算時間は最長で0.0122[sec]であった。なお、計算に用いたPCはCPU: Intel Pentium IV 3.0GHz, RAM: 1GB であり、二次計画法はMATLAB Optimization Toolbox¹¹⁾を使って解いた。一方、衝突回避制約を含めた混合整数二次計画問題を標準的な分枝限定法で解く場合、子問題が82個生成され、計算時間は1.27[sec]となった。なお、混合整数二次計画問題はMATLAB Hybrid Toolboxの混合整数二次計画ソルバー¹²⁾を用いて解いた。

この例が示すように、標準的な分枝限定法は一般的な混合整数計画問題に適用可能である反面、衝突回避問題に対して必ずしも効率のよい方法とはいえない。次節では、衝突回避問題に対して不必要な子問題の生成をできるだけ抑えることを目的とした分枝限定法について述べる。

5. 衝突回避問題に適合した分枝限定法

前章でも述べたように、衝突回避のための最適制御問題においては、予測を行なうすべての時刻に対して、すべての移動体との衝突回避制約が課せられており、実際には不必要な制約条件も数多く含まれている。われわれが日常的に行なっている衝突回避においても、衝突を予測してからどの方向に回避するかを決定することが多く、常に衝突回避を意識して移動することは稀であると思われる。

提案手法は、整数条件を緩和した問題ではなく、衝突回避制約を除いた問題を緩和問題とする。緩和問題を解いて得られた軌道が衝突回避制約を満たさない場合には、Fig. 6 で表わされる4つの領域への回避を行なう制約条件を追加した子問題を生成する。具体的には、

$$\begin{aligned} x_{vi} - x_{vj} &\leq \Psi\kappa_{ij1} + \Psi\kappa_{ij2} - \psi \\ y_{vi} - y_{vj} &\leq -\Psi\kappa_{ij1} + \Psi\kappa_{ij2} + \Psi - \psi \\ -x_{vi} + x_{vj} &\leq \Psi\kappa_{ij1} - \Psi\kappa_{ij2} + \Psi - \psi \\ -y_{vi} + y_{vj} &\leq -\Psi\kappa_{ij1} - \Psi\kappa_{ij2} + 2\Psi - \psi \end{aligned} \quad (18)$$

において、 $(\kappa_{ij1}, \kappa_{ij2}) = (0, 0), (0, 1), (1, 0), (1, 1)$ と固定した制約条件を追加した4つの子問題を生成する。すなわち、衝突が予測されるまでは衝突回避を意識せずに移動し、衝突が予測されるに伴い、回避する方向を決定するという方法になっている。この方法により、衝突回避制約を新たに必要としない子問題は必ず終端するため、不必要な子問題の生成が抑えられると考えられる。

提案する分枝限定法のアルゴリズムを以下に示す。

Step 0: 初期化

リスト $L = \{P_0\}$ 、暫定値 $z^* = \infty$ 、暫定解 $X^* = \phi$ 、番号 $l = 0$ とおく。

Step 1: 最適性判定

リスト $L = \phi$ であれば終了。このとき暫定値 $z^* < \infty$ であれば、対応する暫定解 X^* が P_0 の最適解となる。一方、 $z^* = \infty$ であれば、 P_0 は実行可能解をもたず終了。また、 $L \neq \phi$ であればStep 2へいく。

Step 2: 子問題の選択

リスト L から1つの子問題 P_k を選び出して $L = L - \{P_k\}$ とする。

Step 3: 限定操作

P_k から分枝操作で固定した衝突回避制約以外を取り除いた緩和問題 P'_k を解く。このとき P'_k が実行可能解をもたなければ、Step 1に戻る。一方、最適値 z'_k をもち、暫定値 z^* に対して $z'_k \geq z^*$ であれば、Step 1に戻り、 $z'_k < z^*$ であればStep 4へいく。

Step 4: 更新操作

P'_k の最適解 X'_k が衝突回避制約を満たすか否か判定する。衝突回避制約を満たす場合、 $z^* = z'_k$, $X^* = X'_k$ とし、Step 1に戻る。一方、衝突回避制約を満たさない場合には

Fig. 5 Example of formation control

Fig. 6 Collision avoidance region represented by eq.(18)

Step 5 へいく。

Step 5: 分枝操作

P'_k に対して、衝突回避制約を満足しない最小の τ を m , そのときの Follower を j とする。このとき、衝突回避制約 (18) の κ を

$$[\kappa_{ij1}(m), \kappa_{ij2}(m)]^T = [0, 0]^T, [0, 1]^T, [1, 0]^T, [1, 1]^T$$

に固定した子問題 $P_{l+1}, P_{l+2}, P_{l+3}, P_{l+4}$ を生成する。また、リスト $L = L \cup \{P_{l+1}, P_{l+2}, P_{l+3}, P_{l+4}\}$, $l = l + 4$ として Step 1 へ戻る。

以上のように提案手法は従来手法と同様に列挙法となっているため、従来法と同様に最適解を得ることができる。その一方で提案手法では、分枝操作は衝突回避制約を満たさない時刻 τ , Follower j に関する 0-1 変数に対してのみ行なわれる (Step 5)。多くの衝突回避問題では移動体が密集していることは稀であり、軌道上で起きうる衝突も限られていることを考慮すると、提案手法では生成される子問題の数の低減化が期待できる。したがって従来手法と比較して計算量が低減化されると考えられる。次節では数値例により生成される子問題の数が低減されることを示す。

6. 数 値 例

本章では 3.2 節の衝突回避問題を 5 節で提案した方法と従来手法のそれぞれを用いて解いた場合のシミュレーション結果を示す。ここでは Follower が 3 台、4 台の場合につい

Table 1 Maximum computation time and number of subproblems ($n=3$)

Horizon	Computation time[s]		Number of subproblems	
	Standard	Proposed	Standard	Proposed
5	1.84	0.111	187	17
6	5.01	0.136	391	17
7	12.2	0.196	859	21
8	43.3	0.669	1840	73
9	151	1.26	5300	113
10	336	1.33	10000	113

て予測ホライズン N を 5~10 に変化させシミュレーションを行なった。このとき各時刻の最適化問題を解くのに要した最大の計算時間および子問題の数の比較を行なった。なお、Leader の制御入力 u_r は衝突回避問題の計算量には大きな影響を与えないため本論文では Leader が固定の場合の結果のみを示すこととする。

混合整数二次計画問題を解くソルバーには従来手法として MATLAB Hybrid Toolbox の混合整数二次計画ソルバー (miqp.m)¹²⁾ を用いた。また、提案手法では miqp.m と同じ二次計画ソルバーを用い、miqp.m の分枝操作を変更したソルバーを作成して計算を行なった。なお、計算に用いた PC は CPU: Intel Pentium IV 3.0GHz, RAM: 1GB である。

6.1 Follower が 3 台の場合

Follower 1, 2, 3 の初期状態は Leader に固定した移動座標系においてそれぞれ $(1.0, 0.0, \pi/2)$, $(0.0, -1.0, \pi/2)$, $(-1.0, 0.0, \pi/2)$ とし、目標位置は Leader に固定した移動座標系においてそれぞれ $(-1.0, 0.0)$, $(0.0, 1.0)$, $(1.0, 0.0)$ とした。その他のパラメータは $v_r = 0.0$, $w_r = 0.0$, $\psi = 0.5$, $\Psi = 30$, $\delta = 0.25$, $\lambda = 0.3$, $\gamma_i = 0.1$, $\eta = 0.4$, $R = I_{2N}$ である。最適化問題 (12) での離散化はサンプリング周期 1.0[sec] で行なった。なお、0-1 変数の総数の変化による影響を除くため従来手法・提案手法ともに (18) 式の形式の制約を衝突回避制約として用いた。

シミュレーションの結果を Table 1 に示す。Table 1 は各時刻の最適化問題にかかった最長の計算時間とその際に解いた緩和問題の数を示したものである。この表から従来手法では予測ホライズン N の増加に伴い計算時間が急激に増加してしまうことが確認できる。一方、提案手法では、計算時間の増加が抑えられていることが確認できる。また、従来手法では解いた緩和問題の数が予測ホライズンの増加とともに急激に増加していることが確認できる。一方、提案手法では解いた子問題の数が低減化されており、子問題の数が低減化されたことで計算時間が低減化されたことが確認できる。

6.2 Follower が 4 台の場合

Follower 1, 2, 3, 4 の初期状態は Leader に固定した移動座標系においてそれぞれ $(1.0, 0.0, \pi/2)$, $(0.0, -1.0, \pi/2)$, $(-1.0, 0.0, \pi/2)$, $(0.0, 1.0, \pi/2)$ とし、目標位置は Leader に固定した移動座標系においてそれぞれ $(-1.0, 0.0)$, $(0.0, 1.0)$, $(1.0, 0.0)$, $(0.0, -1.0)$ とした。その他のパラメータおよび条件は 6.1 節と同じである。

Table 2 Maximum computation time and number of subproblems ($n=4$)

Horizon	Computation time[s]		Number of subproblems	
	Standard	Proposed	Standard	Proposed
5	—	—	—	—
6	245	0.346	22500	45
7	1350	0.591	102000	69
8	*	1.21	*	113
9	*	1.81	*	141
10	*	5.48	*	401

Fig. 7 Vehicles used for experiments

シミュレーションの結果を **Table 2** に示す。Table 2 は各時刻の最適化問題にかかった最長の計算時間とその際に解いた緩和問題の数を示したものである。ただし、予測ホライズンが5の場合は実行不可能であった。この表から従来手法では予測ホライズン N の増加に伴い計算時間が急激に増加してしまうことが確認できる。特に $N \geq 8$ では計算が困難となってしまった。一方、提案手法では、計算時間の増加が抑えられ $N \geq 8$ でも計算を行なうことが可能であることが確認できる。また、提案手法では子問題の数が低減化されており、6.1 節の結果と同様に子問題の数が低減化されたことで計算時間が低減化されたことが確認できる。

7. 実験

3章で示した衝突回避手法に対し5章の提案手法を適用しその有効性の検証を実機で行なった。ここでは従来手法では困難であった Follower が3台の場合について、その初期位置および目標位置を変えて行なった3つの実験結果を示す。実験には市販のラジコン戦車を改造したものをを用い、その位置と姿勢情報は上方に固定したカメラ情報により取得した (**Fig. 7** 参照)。Follower の制御入力への計算はシステムの簡略化のため1台の PC (CPU: Intel Pentium IV 3.0GHz, RAM: 1GB) で行なった。また、混合整数二次計画問題のソルバーは6節で作成したものを MATLAB Compiler¹³⁾により変換し用いた。

7.1 Case1

Follower 1, 2, 3 の初期状態は絶対座標系においてそれぞれ $(0.5, 1.0, 0.0)$, $(1.5, 1.0, 0.0)$, $(2.5, 1.0, 0.0)$ 、目標位置は絶対座標系においてそれぞれ $(2.5, 1.0)$, $(0.5, 1.0)$, $(1.5, 1.0)$

Fig. 8 x-y plot of followers (Case1) **Fig. 9** Minimum distance among followers (Case1)

とした。また、Leader の初期位置は絶対座標系において $(3.5, 1.0, 0.0)$ とし、移動は行なわないものとした。実験で用いたその他のパラメータは、 $d = 0.15$, $\psi = 0.5$, $\Psi = 30$, $N = 5.0$, $\delta = 0.25$, $\lambda = 0.3$, $\gamma_i = 0.1$, $\eta = 0.4$, $R = I_{2N}$ である。最適化問題 (12) での離散化はサンプリング周期 $1.0[\text{sec}]$ で行なった。

実験結果を **Fig. 8**, **Fig. 9** に示す。Fig. 8 は移動体の軌跡を示したもので、○印、×印はそれぞれ各 Follower の $1.0[\text{sec}]$ ごとの位置と目標位置を示したものである。Fig. 9 は Follower 間の最小距離を示したものである。Fig. 8, Fig. 9 から、移動体同士の衝突が回避され目標位置に収束していることが確認できる。

7.2 Case2

Follower 1, 2, 3 の初期状態は絶対座標系においてそれぞれ $(1.75, 0.0, \pi/2)$, $(0.75, 1.0, 0.0)$, $(1.75, 2.0, -\pi/2)$ 、目標位置は絶対座標系においてそれぞれ $(1.75, 2.0)$, $(2.75, 1.0)$, $(1.75, 0.0)$ とした。なお、予測ホライズン N は $N = 6$ とし、その他のパラメータおよび実験条件は Case1 と同様にした。

実験結果を **Fig. 10**, **Fig. 11** に示す。Fig. 10 は移動体の軌跡を示したもので、○印、×印はそれぞれ各 Follower の $1.0[\text{sec}]$ ごとの位置と目標位置を示したものである。Fig. 11 は Follower 間の最小距離を示したものである。Fig. 10, Fig. 11 から移動体同士の衝突が回避され目標位置に収束しているこ

Fig. 10 x-y plot of followers (Case2)

Fig. 11 Minimum distance among followers (Case2)

Fig. 12 x-y plot of followers (Case 3)

Fig. 13 Minimum distance among followers (Case 3)

とが確認できる。

7.3 Case3

Follower 1, 2, 3 の初期状態は絶対座標系においてそれぞれ $(1.0, 0.0, 0.0)$, $(1.0, 1.0, 0.0)$, $(1.0, 2.0, 0.0)$, 目標位置は絶対座標系においてそれぞれ $(2.25, 2.0)$, $(2.25, 1.0)$, $(2.25, 0.0)$ とした。なお、予測ホライズン N は $N = 6$ とし、他のパラメータおよび実験条件は Case1 と同様にした。

実験結果を Fig. 12, Fig. 13 に示す。Fig. 12 は移動体の軌跡を示したもので、○印、×印はそれぞれ各 Follower の 1.0[sec] ごとの位置と目標位置を示したものである。Fig. 13 は Follower 間の最小距離を示したものである。Fig. 12, Fig. 13 から移動体同士の衝突が回避され目標位置に収束していることが確認できる。

8. おわりに

本論文では、隊列変化に伴う移動体の衝突回避問題の計算量を低減化するため、新たな分枝条件に基づいた分枝限定法を提案した。提案手法では従来手法のように整数条件を緩和するのではなく衝突回避制約を除いた問題を解く。もし緩和問題を解いて得られた軌道上で衝突が起きる場合には、衝突回避を行なう方向を表わした制約条件を新たに追加した子問題を生成する。従って提案手法では衝突が起きる時刻、移動

体に関する子問題のみ生成されるため計算量の低減化が期待できる。また、提案手法を衝突回避を考慮した文献 9) の編隊制御手法に適用し、数値例および実験によりその有効性の検証を行なった。

本論文では MATLAB Hybrid Toolbox の混合整数二次計画ソルバーに基づいて提案手法のソルバーを作成したが、現在広く用いられている CPLEX¹⁷⁾などのソフトウェアにはより高速な混合二次計画ソルバーが含まれている。したがって、これらのソフトウェアを利用するなど、提案手法のソルバーを改善していくことが今後の課題である。

なお、提案した分枝限定法は文献 9) だけでなく、その他の衝突回避手法^{7), 8), 14)}や障害物回避手法^{15), 16)}にも適用可能であることを付記しておく。

参考文献

- 1) A.K. Das, R. Fierro and V. Kumar: A Vision-Based Formation Control Framework, *IEEE Trans. on Robotics and Automation*, **18**-5, 813/825 (2002)
- 2) Z. Lin, B. Francis and M. Maggiore: Necessary and sufficient graphical conditions for formation control of unicycles, *IEEE Trans. Automatic Control*, **50**-1, 121/127 (2005)
- 3) A. Jadbabaie, J. Lin and A.S. Morse: Coordination of groups of mobile autonomous agents using nearest neighbor rules, *IEEE Trans. Automatic Control*, **48**-6, 988/1001

(2003)

- 4) R. Olfati-Saber, W.B. Bunbar and R.M. Murray: Cooperative Control of Multi-Vehicle Systems Using Cost Graph and Optimization, *Proc. of American Control Conference*, 2217/2222 (2003)
- 5) D.Q. Mayne, J.B. Rawlings, C.V. Rao and P.O. Scokaert: Constrained model predictive control: Stability and optimality, *Automatica*, **36**, 789/814 (2000)
- 6) J.M. Maciejowski: Predictive Control with Constraints, Prentice Hall (2002)
- 7) A. Richards and J. How: A Decentralized Algorithm for Robust Constrained Model Predictive Control, *Proc. of American Control Conference*, 4261/4266 (2004)
- 8) T. Kopfstadt, M. Mukai, M. Fujita and O. Sawodny: A networked formation control for group of mobile robots using mixed integer programming, *IEEE Int. Conf. on Control Applications*, 579/584 (2006)
- 9) H. Fukushima, K. Kon and F. Matsuno: Distributed Model Predictive Control for Multi-Vehicle Formation with Collision Avoidance Constraints, *Proc. IEEE Conf. on Decision and Control*, 5480/5485 (2005)
- 10) 西川, 三宮, 茨木: 最適化, 93/115, 岩波書店 (1985)
- 11) MATLAB Optimization Toolbox, The MathWorks, Inc.
- 12) A. Bemporad: MIQP.m, <http://www.dii.unisi.it/hybrid>
- 13) MATLAB Compiler User's Guide, The MathWorks, Inc.
- 14) 長井, 児島: MLD システム表現を用いた群集挙動のモデリング, 第5回計測自動制御学会制御部門大会, 405/409 (2005)
- 15) J. Bellingham, A. Richards and J. How: Receding Horizon Control of Autonomous Aerial Vehicles, *Proc. of American Control Conference*, 3741/3746 (2002)
- 16) A. Richards and J. How: Model Predictive Control of Vehicle Maneuvers with Guaranteed Completion Time and Robust Feasibility, *Proc. of American Control Conference*, 4034/4040 (2003)
- 17) CPLEX, <http://www.ilog.co.jp/products/cplex/>, ILOG

[著者紹介]

根 和 幸 (学生会員)

1983年2月17日生。2007年電気通信大学大学院電気通信学研究科博士前期課程卒業。同年同大学大学院同研究科博士後期課程進学。現在に至る。複数移動体の制御の研究に従事。

福 島 宏 明 (正会員)

2001年京都大学大学院情報学研究科博士後期課程修了。1999年から2004年まで日本学術振興会特別研究員。2004年から電気通信大学知能機械工学科助手。2007年から同助教となり、現在に至る。ロバスト制御、システム同定の研究に従事。2005年本学会論文賞受賞。システム制御情報学会、IEEEの会員。

松 野 文 俊 (正会員)

1957年7月26日生。86年大阪大学大学院基礎工学研究科博士課程修了。大阪大学助手、神戸大学助教授、東京工業大学大学院助教授を経て、2003年電気通信大学教授(知能機械工学科)となり、現在に至る。主に、ロボティクス・制御理論・レスキュー学に関する研究に従事。2002年NPO国際レスキューシステム研究機構理事・川崎ラボリーダ、2005年より副会長。競基弘賞委員会委員長。89年度日本ロボット学会研究奨励賞、93年度システム制御情報学会論文賞、97年度消防防災科学論文賞(消防庁長官表彰)、2001年度、2006年度計測自動制御学会論文賞、2001年度同学会武田賞などを受賞。IEEE、日本機械学会、GIS学会、情報処理学会などの会員。工学博士。