UAV の衝突回避経路生成に関する一手法^{*1} —2 次計画法と Force Field Algorithm の併用— A Method of Collision-Free Path Calculation for UAV —Combined Application of Quadratic Programming and Force Field Algorithm—

横 山 信 宏^{*2}
Nobuhiro YOKOYAMA

Key Words: Guidance and Control, Collision Avoidance, Quadratic Programming, Decentralized Algorithm

Abstract: This paper presents a decentralized collision-free path calculation method for multiple UAVs (Unmanned Aerial Vehicles). The developed method, which is composed of force field algorithm for conflict reduction and quadratic programming for path optimization, can execute parallel path calculation. While conventional path calculation methods tend to require prohibitive computational time as the number of concerned aircraft increases, the computational time in the developed method is not so sensitive to the number of concerned aircraft. Results of numerical simulations are presented to demonstrate the reasonableness of resulting paths, the computational efficiency, and robustness of the developed method.

1. は じ め に

UAV (Unmanned Aerial Vehicle) の自律飛行においては、飛行中に変化する外部環境に応じた実時間の経路生成が必要である。特に、複数機が狭い空域で飛行する場合においては、時々刻々位置が変化する他機との間隔を常に確保しながら所定のミッションを遂行しなければならず、実時間の経路生成がより一層重要となる。これまでに、他機との衝突回避が可能な経路生成手法については、UAV の自律飛行だけでなく有人航空機のフリーフライト¹⁾への応用もターゲットとして、様々なものが提案されてきた²⁻¹⁰⁾。これらの手法は、(a) ルールベース²⁻⁴⁾ か最適化ベース⁵⁻¹⁰⁾ か、(b) 中央集中的⁸⁻¹⁰⁾ か自律分散的²⁻⁷⁾ か、という観点から分類できる。(a) に関して、ルールベースの手法は演算量の少なさが魅力であるが、近年の計算機性能の向上に伴い、ミッション達成にとって有益な飛行経路を最適化ベースで求める手法も、実用的になりつつある。また、(b) に関して、中央集中的な衝突回避経路生成は UAV を地上から管制する場合に有効な手段であるが、システムトラブルの際のロバスト性や、環境変化への即応性という点では、各機で自律分散的に衝突回避経路生成を行う方が、UAV の飛行にとっては望ましいと考えられる。

以上の観点から、本稿では最適化ベースの自律分散的な

衝突回避経路生成手法について提案を行う。この種の手法はさらに、直列型⁵⁾、並列型^{6, 7)}の手法のいずれかに分類できる。直列型の手法は、各機がリレー形式で経路生成を行うものである。この手法では、ある機体が経路生成を行っている間、他の機体は一切計算を行わず、自分の順番が来るのを待つこととなるため、各機の経路生成が一巡するのに要する時間は、機数の影響を強く受けてしまう。一方、並列型の手法では、各機の経路生成が同時に実行され、順番待ちのために計算機を休止させる必要が無い。この点では並列型の手法の方が直列型の手法よりも効率的であると言えるが、従来の並列型手法は、依然として計算時間が機数の影響を強く受ける傾向を持っている。例えば、Bicchi らの手法⁶⁾は、各機において、認識可能な機体の経路をすべて最適化し、その結果から自機の経路を更新するというものである。すなわち、全体として冗長な最適化計算が実行され、各機で要求される演算量は扱う機数の増加に伴い急増してしまう(経路間の非干渉条件の個数が、機体の組み合わせの数に比例することによる)。Tomlin らのゲーム理論に基づいた手法⁷⁾ではこういった冗長な最適化計算を行わないが、アルゴリズム自体が機体の組み合わせの数に伴って急激に複雑になるため、3 機以上が飛行する環境への適用は実用的でないとの指摘がある⁵⁾。

本稿で提案する衝突回避経路生成手法も並列型の手法であるが、その新規性は、計算時間が機数の影響を受けにくいように設計している点にある。これは、ルールベース手法の一種である Force Field Algorithm と、高速な最適化手法である 2 次計画法 (Quadratic Programming, QP)

^{*1} © 2008 日本航空宇宙学会

平成 19 年 11 月 25 日、第 50 回自動制御連合講演会において発表。平成 19 年 10 月 1 日原稿受理

^{*2} 防衛大学校システム工学群航空宇宙工学科

を逐次的に適用することで実現される。また、経路の非干渉条件において複数機の最適化変数がカップリングしないように定式化することで、QP による最適化の並列化を可能にしている。

本稿は次のように構成される。まず第 2 節で衝突回避経路生成手法の提案を行い、第 3 節で提案する手法の有効性について、シミュレーションを通じた検証を行う。最後に、第 4 節で結論を述べる。

2. 衝突回避経路生成手法

2.1 運動モデル 状態方程式として、水平面内における機体のキネマティクスのみを考慮し、簡単のため風の影響も無視する。扱う機体の数を M とし、以下では自機の識別番号を表す記号を i ($= 1, \dots, M$) とする。機体のキネマティクスは以下で表される。

$$\begin{aligned} \dot{x}^i &= V^i \cos \psi^i, & \dot{y}^i &= V^i \sin \psi^i, \\ \dot{V}^i &= a_c^i, & \dot{\psi}^i &= g_0 \tan \phi^i / V^i \end{aligned} \quad (1)$$

ただし、各記号の定義は次の通りである— x, y : 慣性座標系における機体の位置, V : 機体固定座標系における速度, ψ : 方位角, a_c : 機軸方向加速度, ϕ : バンク角, g_0 : 重力加速度。経路生成においては、飛行性能を考慮して、速度、機軸方向加速度、バンク角に制約を与える必要がある。これらを式で表すと、

$$V_{\min}^i \leq V^i \leq V_{\max}^i \quad (2)$$

$$|a_c^i| \leq a_{\max}^i \quad (3)$$

$$|\phi^i| \leq \phi_{\max}^i \quad (4)$$

となる。ここで、 $V_{\min}^i, V_{\max}^i, a_{\max}^i, \phi_{\max}^i$ は機体の飛行性能に基づく定数である。

式 (1) は連続系の非線形微分方程式であり、そのままでは QP を適用することができないため、離散系の線形方程式に置き換える。慣性座標系における速度と加速度をそれぞれ $u \equiv \dot{x}, v \equiv \dot{y}, a_x \equiv \ddot{x}, a_y \equiv \ddot{y}$ と表し、 x, y, u, v を状態変数、 a_x, a_y を制御入力とすると、機体 i の離散系状態方程式は、以下の式で表すことができる。

$$\begin{aligned} \begin{bmatrix} x_{k+1}^i \\ y_{k+1}^i \\ u_{k+1}^i \\ v_{k+1}^i \end{bmatrix} &= \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k^i \\ y_k^i \\ u_k^i \\ v_k^i \end{bmatrix} \\ &+ \begin{bmatrix} \Delta t^2/2 & 0 \\ 0 & \Delta t^2/2 \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \begin{bmatrix} a_{xk}^i \\ a_{yk}^i \end{bmatrix} \end{aligned} \quad (5)$$

ここで、添え字 k は離散系の時間ステップを、 Δt は時間ステップの長さを表す。次に、状態方程式に合わせて式 (2)~(4) も離散系に置き換える。慣性座標系と機体固定座標系における速度の関係式 $u^i = V^i \cos \psi^i, v^i = V^i \sin \psi^i$ の両辺を微分し、 $[a_c^i \ g_0 \tan \phi^i]^T$ について解くと、以下を

得る。

$$\begin{bmatrix} a_c^i \\ g_0 \tan \phi^i \end{bmatrix} = \begin{bmatrix} \cos \psi^i & \sin \psi^i \\ -\sin \psi^i & \cos \psi^i \end{bmatrix} \begin{bmatrix} a_x^i \\ a_y^i \end{bmatrix} \quad (6)$$

この関係から、式 (3), (4) は離散系において以下に置き換えることができる。

$$|a_{xk}^i \cos \psi_k^i + a_{yk}^i \sin \psi_k^i| \leq a_{\max}^i \quad (7)$$

$$|-a_{xk}^i \sin \psi_k^i + a_{yk}^i \cos \psi_k^i| \leq g_0 \tan \phi_{\max}^i \quad (8)$$

さらに、式 (7), (8) が任意の ψ_k^i において満たされる場合、これらの式から ψ_k^i を消去した以下の式が成り立つ。

$$(a_{xk}^i)^2 + (a_{yk}^i)^2 \leq (b^i)^2 \quad (9)$$

ただし、

$$b^i = \min(a_{\max}^i, g_0 \tan \phi_{\max}^i) \quad (10)$$

である。また、速度の制約条件を表す式 (2) も、状態変数 u_k^i, v_k^i を用いて以下のように置き換えることができる。

$$(V_{\min}^i)^2 \leq (u_k^i)^2 + (v_k^i)^2 \leq (V_{\max}^i)^2 \quad (11)$$

提案手法では、以上に示した離散系の状態方程式 (5) および制約条件式 (9), (11) を機体の運動モデルとして用いる。なお、QP を適用する際には、制約条件式も線形である必要があるため式 (9), (11) をさらに線形の式に変換するが、その方法については後述する。

2.2 アルゴリズムの概要 提案手法では、時々刻々において、ホライズンと呼ばれる一定時間先までの経路が実時間で計画される。ホライズンに対応する時間ステップ番号を N とすると、飛行経路は、初期状態からホライズンまでの状態変数 $x_k^i, y_k^i, u_k^i, v_k^i$ ($k = 0, \dots, N$) と制御入力 a_{xk}^i, a_{yk}^i ($k = 0, \dots, N-1$) の列から構成される。また、参照経路 (何も障害物がない状態で個々の機体が本来追従すべき経路) は予め与えるものとし、これを $\tilde{x}_k^i, \tilde{y}_k^i, \tilde{u}_k^i, \tilde{v}_k^i$ ($k = 0, \dots, N$) と表す。なお、各機に搭載されたデータリンクシステムを介して、計画経路、参照経路、および飛行性能を機体間で共有できるものと仮定する。

提案手法の単位サイクルでは、時間 Δt の間に、干渉軽減 (CR, Conflict Reduction), 経路最適化 (TO, Trajectory Optimization), 計画経路の放送 (TB, Trajectory Broadcast) という 3 フェーズの処理が行われる。各サイクルでは、まず CR フェーズにおいて他機の計画経路との干渉が軽減されるように計画経路を更新する。次に、TO フェーズにおいて所定の目的関数を最小化するように計画経路を更新する。最後に、TB フェーズにおいて、データリンクを介して各機の更新後の計画経路を送受信する。CR フェーズ, TO フェーズの詳細は、2.3 節および 2.4 節において述べる。

サイクルの開始時刻を t_s と表す。各サイクルでは、時刻 $t_0 = t_s + 2\Delta t$ 以降の経路を更新対象とする。すなわち、時

第 1 図 Force Field Algorithm の概念図

刻 t_s から時刻 t_0 までの間の計画経路は機体が追従すべき経路として固定され、時刻 t_0 における変数に時間ステップ番号 $k = 0$ が割り当てられる。また、計画経路への追従性は本稿における検討の対象外であるため、追従誤差を零と仮定する。

2.3 干渉軽減 (CR) フェーズ CR フェーズでは、必要に応じて Force Field Algorithm を適用し、機体間の干渉の軽減を図る。以下では、CR フェーズにおいて更新された経路を $\hat{x}_k^i, \hat{y}_k^i, \hat{u}_k^i, \hat{v}_k^i$ ($k = 0, \dots, N$), $\hat{a}_{xk}^i, \hat{a}_{yk}^i$ ($k = 0, \dots, N-1$) と表す。機体 i ($i = 1, \dots, M$) において、CR フェーズでは以下のアルゴリズムが実行される。

Step 1: 干渉の探索 すべての機体 $j = 1, \dots, i, \dots, M$ について、他機 $l = 1, \dots, i, \dots, M$ ($l \neq j$) との経路の干渉が最初に発生する時刻を探索し、対応する時間ステップ番号を $k = k_c^j$ とおく。この探索には、機体間で共有している現在の計画経路 x_k^j, y_k^j ($k = 0, \dots, N$), x_k^l, y_k^l ($k = 0, \dots, N$) を用いる。すなわち、次式が成り立つとき、干渉していると判定する。

$$(x_k^j - x_k^l)^2 + (y_k^j - y_k^l)^2 < R^2 \quad (12)$$

ただし、 R は必要間隔で、定数として予め与えられる。次に、すべての機体 $j = 1, \dots, i, \dots, M$ について、経路の更新を開始する時間ステップ番号 $k_s^j = \max(0, k_c^j - n)$ を求める。ここで、 n は経験的に設定する正の整数である。

Step 2: 初期化 $k = 0$ とし、すべての機体 $j = 1, \dots, i, \dots, M$ について $[\hat{x}_0^j \ \hat{y}_0^j \ \hat{u}_0^j \ \hat{v}_0^j]^T = [x_0^j \ y_0^j \ u_0^j \ v_0^j]^T$ とおく。

Step 3: 制御入力 の計算 すべての機体 $j = 1, \dots, i, \dots, M$ について、以下の処理を行う。

- $k < k_s^j$ の場合は制御入力に対する更新を行わず、 $[\hat{a}_{xk}^j \ \hat{a}_{yk}^j]^T = [a_{xk}^j \ a_{yk}^j]^T$ とおく。

- $k_s^j \leq k \leq N-1$ の場合、Force Field Algorithm に基づき、機体を参照点に近づける引力相当のベクトルと、他機から遠ざける反発力相当のベクトルを合成して指示速度ベクトルを求め、制御入力に変換する (第 1 図参照)。必要な演算は以下の通りである。

ホライズン時刻 ($k = N$) における参照経路上の位置 $(\hat{x}_N^j, \hat{y}_N^j)$ を、参照点として与える。機体 j を参照点に近づける引力相当のベクトル $(V_{k+1}^j)_A$ は、次式から求める。

$$(V_{k+1}^j)_A = \max \left[V_{\min}^j, \max \left(\|e_{k+1}^j\|, V_{\max}^j \right) \right] \cdot \frac{e_{k+1}^j}{\|e_{k+1}^j\|} \quad (13)$$

$$e_{k+1}^j = \left[\frac{\hat{x}_N^j - \hat{x}_k^j}{(N-k)\Delta t}, \frac{\hat{y}_N^j - \hat{y}_k^j}{(N-k)\Delta t} \right]^T \quad (14)$$

また、時間ステップ k における機体 j と機体 l の位置 $(\hat{x}_k^j, \hat{y}_k^j)$ および $(\hat{x}_k^l, \hat{y}_k^l)$ を起点とし、各々が有する引力相当ベクトル $(V_{k+1}^j)_A$ および $(V_{k+1}^l)_A$ に基づいて直線状に経路を外挿する。これら外挿経路の最接近時刻を τ_k^{jl} 、対応する位置をそれぞれ $(\bar{x}^j(\tau_k^{jl}), \bar{y}^j(\tau_k^{jl}))$, $(\bar{x}^l(\tau_k^{jl}), \bar{y}^l(\tau_k^{jl}))$ と表す。機体 j に作用する反発力相当ベクトル $(V_{k+1}^j)_R$ は、最接近点の幾何学的関係から以下のようにして求める。

$$(V_{k+1}^j)_R = \frac{\max[0, R - \|h_k^{jl}\|]}{\tau_k^{jl} - k\Delta t} \cdot \frac{h_k^{jl}}{\|h_k^{jl}\|} \quad (15)$$

$$h_k^{jl} = \left[\bar{x}^j(\tau_k^{jl}) - \bar{x}^l(\tau_k^{jl}), \bar{y}^j(\tau_k^{jl}) - \bar{y}^l(\tau_k^{jl}) \right]^T \quad (16)$$

さらに、引力相当ベクトルと反発力相当ベクトルを合成して、機体 j の指示速度ベクトル $(V_{k+1}^j)_C = [(u_{k+1}^j)_C \ (v_{k+1}^j)_C]^T$ を以下から求める。

$$(V_{k+1}^j)_C = (V_{k+1}^j)_A + \sum_{l(\neq j)=1}^M (V_{k+1}^l)_R \quad (17)$$

なお、外挿経路の最接近点において機体 j と機体 l の距離が零 ($h_k^{jl} = 0$) となる場合、式 (15) は特異となってそのままでは適用できない。この場合は、最接近点を中心とした半径 R の円に対して $(\hat{x}_k^j, \hat{y}_k^j)$ から接線 (2 本存在するが、 $(\hat{x}_k^j, \hat{y}_k^j)$ から円の中心に向かって左となる接線) を引き、 $(\hat{x}_k^j, \hat{y}_k^j)$ から接点に向かう方向の単位ベクトルを式 (15) の $h_k^{jl} / \|h_k^{jl}\|$ と置き換えることで対応する。次に、制御入力 $\hat{a}_{xk}^j, \hat{a}_{yk}^j$ を次式から求める。

$$\begin{aligned} \hat{a}_{xk}^j &= [(u_{k+1}^j)_C - \hat{u}_k^j] / \Delta t, \\ \hat{a}_{yk}^j &= [(v_{k+1}^j)_C - \hat{v}_k^j] / \Delta t \end{aligned} \quad (18)$$

ただし、加速度 $\hat{a}_{xk}^j, \hat{a}_{yk}^j$ と、これらによって達成される次の時間ステップの速度 $\hat{u}_{k+1}^j, \hat{v}_{k+1}^j$ が、式 (9), (11) をそれぞれ満たすように $\hat{a}_{xk}^j, \hat{a}_{yk}^j$ を修正する。

Step 4: 次の時間ステップへの移行 すべての機体 $j = 1, \dots, i, \dots, M$ について、状態方程式 (5) より次ステップの状態変数 $\hat{x}_{k+1}^j, \hat{y}_{k+1}^j, \hat{u}_{k+1}^j, \hat{v}_{k+1}^j$ を求める。続いて、時間ステップ k を 1 つ増やす。ここで $k = N$ となった場合は CR フェーズを終了し TO フェーズに移行する。 $k < N$ の場合は、Step 3 に戻る。(アルゴリズム終了)

以上、CR フェーズにおいては、同一内容の計算がすべての機体で同時に実行され、更新される計画経路もすべて同一のものとなる。したがって、CR フェーズでは M 重

の冗長な計算を行うこととなり、1 機当たりの計算時間が機数の増加の影響を受けやすい。しかしながら、実用的な機数の範囲内では、CR フェーズの計算量は TO フェーズの計算量と比べて少なく、機数増加が CR フェーズの計算量に及ぼす影響は、トータルの計算時間を考える上で無視し得る。なお、式 (17) の反発力相当ベクトルが相殺して、右辺第 2 項がゼロになる場合には、機体 j が回避経路をとらない点に注意が必要である。ただし、この場合も、他機に機体 j の存在によって生じる反発力相当のベクトルが働けば、干渉は低減される。また、仮にすべての機体において反発力が相殺する場合があっても、TO フェーズにおいて干渉の軽減が可能である。

CR フェーズで得られる全機分の更新経路が各機で同一であることから、この経路情報を次の TO フェーズの基準として利用しても、その結果に不整合は起こらない。(逆に、各機で異なる経路情報を基準として利用した場合、最適化によって新たな干渉が起こり得る。)そこで、後述するように、TO フェーズでは CR フェーズで更新された他機の計画経路も非干渉化の際の基準として利用している。

2.4 経路最適化 (TO) フェーズ TO フェーズにおいては、QP を用いた最適化によって、CR フェーズで得られた経路を更新し、所定の目的関数の最小化を図る。

まず、式 (9)、(11) で表される非線形の制約条件を、QP が適用可能な形式とすべく、凸で線形な制約条件に置き換える。第 2 図の (a) に示すように、各時間ステップにおいて、式 (11) で表される非凸なドーナツ状の速度制限は、CR フェーズの解 $(\hat{u}_k^i, \hat{v}_k^i)$ を基準とした凸な内接 5 角形に置き換える。この条件は以下の式で表すことができる。

$$\begin{aligned} (\eta_{lk}^i - \eta_{(l+1)k}^i)(u_k^i - \xi_{lk}^i) + (\xi_{(l+1)k}^i - \xi_{lk}^i)(v_k^i - \eta_{lk}^i) &\geq 0 \\ (l = 1, \dots, 4, \quad k = 1, \dots, N) \\ (\eta_{5k}^i - \eta_{1k}^i)(u_k^i - \xi_{5k}^i) + (\xi_{1k}^i - \xi_{5k}^i)(v_k^i - \eta_{5k}^i) &\geq 0 \\ (k = 1, \dots, N) \end{aligned} \quad (19)$$

ただし、各記号の定義は以下の通りである。

$$\begin{aligned} \begin{bmatrix} \xi_{1k}^i \\ \eta_{1k}^i \end{bmatrix} &= \begin{bmatrix} V_{\min}^i \cos \hat{\psi}_k^i - \sqrt{V_{\max}^2 - V_{\min}^2} \sin \hat{\psi}_k^i \\ V_{\min}^i \sin \hat{\psi}_k^i + \sqrt{V_{\max}^2 - V_{\min}^2} \cos \hat{\psi}_k^i \end{bmatrix} \\ \begin{bmatrix} \xi_{2k}^i \\ \eta_{2k}^i \end{bmatrix} &= \begin{bmatrix} V_{\max}^i \cos(\hat{\psi}_k^i - \hat{\chi}_k^i) \\ V_{\max}^i \sin(\hat{\psi}_k^i - \hat{\chi}_k^i) \end{bmatrix} \\ \begin{bmatrix} \xi_{3k}^i \\ \eta_{3k}^i \end{bmatrix} &= \begin{bmatrix} V_{\max}^i \cos \hat{\psi}_k^i \\ V_{\max}^i \sin \hat{\psi}_k^i \end{bmatrix} \\ \begin{bmatrix} \xi_{4k}^i \\ \eta_{4k}^i \end{bmatrix} &= \begin{bmatrix} V_{\max}^i \cos(\hat{\psi}_k^i + \hat{\chi}_k^i) \\ V_{\max}^i \sin(\hat{\psi}_k^i + \hat{\chi}_k^i) \end{bmatrix} \\ \begin{bmatrix} \xi_{5k}^i \\ \eta_{5k}^i \end{bmatrix} &= \begin{bmatrix} V_{\min}^i \cos \hat{\psi}_k^i + \sqrt{V_{\max}^2 - V_{\min}^2} \sin \hat{\psi}_k^i \\ V_{\min}^i \sin \hat{\psi}_k^i - \sqrt{V_{\max}^2 - V_{\min}^2} \cos \hat{\psi}_k^i \end{bmatrix} \end{aligned} \quad (20)$$

$$\begin{aligned} \hat{\psi}_k^i &= \tan^{-1}(\hat{v}_k^i / \hat{u}_k^i) \\ \hat{\chi}_k^i &= [\tan^{-1}(\eta_{3k}^i / \xi_{3k}^i) - \tan^{-1}(\eta_{1k}^i / \xi_{1k}^i)] / 2 \end{aligned} \quad (21)$$

第 2 図 TO フェーズにおける更新可能領域

また、第 2 図の (b) に示すように、式 (9) で表される円状の加速度制約は、各時間ステップにおいて、CR フェーズの解 $(\hat{a}_{xk}^i, \hat{a}_{yk}^i)$ を基準とした凸な内接 8 角形に置き換える。この条件は以下の式で表すことができる。

$$\begin{aligned} (s_{lk}^i - s_{(l+1)k}^i)(a_{xk}^i - b^i c_{lk}^i) + (c_{(l+1)k}^i - c_{lk}^i)(a_{yk}^i - b^i s_{lk}^i) &\geq 0 \\ (l = 1, \dots, 7, \quad k = 1, \dots, N-1) \\ (s_{8k}^i - s_{1k}^i)(a_{xk}^i - b^i c_{8k}^i) + (c_{1k}^i - c_{8k}^i)(a_{yk}^i - b^i s_{8k}^i) &\geq 0 \\ (k = 1, \dots, N-1) \end{aligned} \quad (22)$$

ただし、 c_{lk}^i, s_{lk}^i は以下のように定義する。

$$\begin{aligned} c_{lk}^i &= \cos[\tan^{-1}(\hat{a}_{yk}^i / \hat{a}_{xk}^i) + (l-1)\pi/4] \\ s_{lk}^i &= \sin[\tan^{-1}(\hat{a}_{yk}^i / \hat{a}_{xk}^i) + (l-1)\pi/4] \end{aligned} \quad (l = 1, \dots, 8) \quad (23)$$

次に、機体間の非干渉条件について考える。通常、自機 i の経路と他機 j の経路の非干渉条件は、次式で与えられる。

$$(x_k^i - x_k^j)^2 + (y_k^i - y_k^j)^2 - R^2 \geq 0 \quad (k = 1, \dots, N) \quad (24)$$

上式は非凸であるとともに、異なる機体の最適化変数 $((x_k^i, y_k^i)$ および (x_k^j, y_k^j)) がカップリングしているため、最適化における制約条件としてそのまま適用すると、最適化計算の収束のロバスト性を損なうとともに、並列化を困難とする。提案手法では、この点を改良すべく、非干渉条件を凸な制約条件に置き換えるとともに、異なる機体に属する最適化変数のカップリングを無くす。ここでは、第 3 図に示すように、CR フェーズの解である $(\hat{x}_k^i, \hat{y}_k^i)$ と $(\hat{x}_k^j, \hat{y}_k^j)$ を基準として、自機 i のステップ k における位置 (x_k^i, y_k^i) に対して以下の線形制約条件を課す。

$$\begin{aligned} \alpha_k^{ij}(x_k^i - p_k^{ij}) + \beta_k^{ij}(y_k^i - q_k^{ij}) - \left[\frac{R}{2} + \left(\gamma_k^{ij} - \frac{R}{2} \right) \delta^i \right] &\geq 0 \\ (k = 2, \dots, N, \quad j = 1, \dots, M \quad (j \neq i)) \end{aligned} \quad (25)$$

$$0 \leq \delta^i \leq 1, \quad (26)$$

ただし、各記号の定義は以下の通りである。

$$p_k^{ij} = (\hat{x}_k^i + \hat{x}_k^j) / 2, \quad q_k^{ij} = (\hat{y}_k^i + \hat{y}_k^j) / 2 \quad (27)$$

$$d_k^{ij} = \sqrt{(\hat{x}_k^i - p_k^{ij})^2 + (\hat{y}_k^i - q_k^{ij})^2} \quad (28)$$

$$\gamma_k^{ij} = \min(d_k^{ij}, R/2) \quad (29)$$

第 3 図 TO フェーズにおける非干渉条件

$$\alpha_k^{ij} = (\hat{x}_k^i - p_k^{ij})/d_k^{ij}, \quad \beta_k^{ij} = (\hat{y}_k^i - q_k^{ij})/d_k^{ij} \quad (30)$$

式 (25) には他機の最適化変数である (x_k^j, y_k^j) が現れないため、独立して最適化計算を行うことが可能となる。(第 3 図から分かるように、式 (25) で定義される更新可能領域は各機について定義され、各機の点 (x_k^i, y_k^i) がこれらの更新可能領域に限定されるならば、並列的に更新が行われても新たな干渉の発生を防ぐことが可能である。) また、式 (25) の制約を緩和するための最適化変数として、 δ^i を導入している。式 (25) は、 $\delta^i = 0$ のとき (x_k^i, y_k^i) と (x_k^j, y_k^j) の間隔が R 以上 (非干渉状態) となるための制約条件となり、 $\delta^i = 1$ のとき (x_k^i, y_k^i) と (x_k^j, y_k^j) の間隔が CR フェーズの解より悪化しないための制約条件となる。すなわち、ここでは非干渉条件をソフト制約として扱い、以下に述べるように、 δ^i を零に近づけるための項を目的関数に付加することで、可能な限り非干渉状態を達成する。

最小化すべき目的関数は、次式で与える。

$$\begin{aligned} \phi^i = & \mu \sum_{k=1}^N [(x_k^i - \tilde{x}_k^i)^2 + (y_k^i - \tilde{y}_k^i)^2] \\ & + \lambda \sum_{k=0}^{N-1} [(a_{xk}^i)^2 + (a_{yk}^i)^2] + \rho(\delta^i)^2 \end{aligned} \quad (31)$$

上式において、 μ , λ , ρ は重みを表す正の定数である。式 (31) の第 1 項は参照経路への追従誤差、第 2 項は経路中における加減速の使用量、第 3 項は制約条件の緩和の度合いにそれぞれ対応する。

以上に基づき、機体 i において解くべき QP 問題は次のように定式化できる。

・ Optimization variables :

$$\begin{aligned} & x_k^i, y_k^i, u_k^i, v_k^i \quad (k = 0, \dots, N), \\ & a_{xk}^i, a_{yk}^i \quad (k = 0, \dots, N-1), \delta^i \end{aligned}$$

・ Minimize : Eq. (31)

・ Subject to : Eqs. (5), (19), (22), (25), (26), and

$$[x_0^i \ y_0^i \ u_0^i \ v_0^i]^T = [\hat{x}_0^i \ \hat{y}_0^i \ \hat{u}_0^i \ \hat{v}_0^i]^T \quad (32)$$

また、初期解は以下のように設定する。

$$\begin{aligned} [x_k^i \ y_k^i \ u_k^i \ v_k^i]^T &= [\hat{x}_k^i \ \hat{y}_k^i \ \hat{u}_k^i \ \hat{v}_k^i]^T \\ & \quad (k = 0, \dots, N), \\ [a_{yk}^i \ a_{xk}^i]^T &= [\hat{a}_{yk}^i \ \hat{a}_{xk}^i]^T \\ & \quad (k = 0, \dots, N-1), \quad \delta^i = 1 \end{aligned} \quad (33)$$

すなわち、初期解の経路は CR フェーズで得られた経路である。上記の QP 問題は、実行可能領域に CR フェーズの解が必ず含まれるように設定しているため、QP の最適解だけでなく、QP の各反復計算で得られる中間解もすべて実行可能解となる。このため、仮に QP が収束する前にサイクル時間 Δt が経過して時間切れになったとしても、QP で得られた最新の中間解を使用することが可能となる。特に小型 UAV においては、演算速度が十分でない機上計算機を搭載せざるを得ない場合が考えられるため、この性質は実用上有益であると考えられる。また、上記の QP 問題はスパースな行列から構成されるため、その性質を利用した QP ソルバ¹¹⁾ と、行列の因子分解手法¹²⁾ を適用することで、計算を高速化している。

3. シミュレーションによる有効性の検証

ここでは、複数 UAV の衝突回避経路生成のシミュレーションに提案手法を適用した結果から、その有効性を検証する。各機の飛行性能は同一とし、各種定数を以下のように設定した。

$$\begin{aligned} & \left. \begin{aligned} V_{\min}^i &= 20 \text{ [m/s]}, & V_{\max}^i &= 30 \text{ [m/s]}, \\ a_{\max}^i &= 3 \text{ [m/s}^2\text{]}, & \phi_{\max}^i &= \pi/9 \text{ [rad]} \end{aligned} \right\} \\ & (i = 1, \dots, M), \\ & R = 500 \text{ [m]}, \quad \Delta t = 2 \text{ [s]}, \quad N = 30, \quad n = 10, \\ & \mu = 10^{-8}, \quad \lambda = 10^{-4}, \quad \rho = 10^2 \end{aligned} \quad (34)$$

まず、第 1 表に示すような 5 通りの参照経路を考える。参照経路はすべて等速の直線飛行経路とし、各機は初期状態において参照経路の初期位置・速度に従っているものと仮定する。ここでは、1-1) 参照経路 A, B に従う 2 機、1-2) 参照経路 A, C に従う 2 機、1-3) 参照経路 A, E に従う 2 機、1-4) 参照経路 A, C, E に従う 3 機、1-5) 参照経路 A, B, C, D に従う 4 機、の 5 ケースの衝突回避経路生成問題について、それぞれシミュレーションを行った。いずれのケースも、各機が参照軌道に沿って飛行した場合、時刻 $t = 60 \text{ [s]}$ において原点 $(0, 0)$ で衝突するため、経路の非干渉化が必要となる。シミュレーションにおいては、Intel Core 2 Duo 2.40 GHz プロセッサ搭載の計算機を 1 台用いた。また、1 台の計算機内において、複数のスレッドとして衝突回避経路生成のプログラムを実行することで、分散環境を模擬している。

各問題において、最終的に機体が飛行した経路を第 4 図から第 8 図に示す。すべてのケースにおいて、最近点 (CPA, Closest Point of Approach) 間の距離の最小値は R と同一の値である 500 [m] となった (有効数字を 3 桁とした場合)。また、各図からすべての機体が CPA 通過後速やかに参照軌道に戻っている様子が確認できる。以上より、提案手法によって妥当な非干渉経路が求められていることが分かる。

次に、参照経路をランダムに変えた様々な衝突回避経路生成問題を通じて、提案手法の計算速度とロバスト性を検証する。ここでの参照経路は、

第1表 Case 1-1 から Case 1-5 における参照経路

Path	Initial position [m]	Velocity [m/s]	Azimuth [rad]
A	(0, 1500)	25	$3\pi/2$
B	(0, -1500)	25	$\pi/2$
C	(1500, 0)	25	π
D	(-1500, 0)	25	0
E	(1061, 1061)	25	$5\pi/4$

第4図 飛行軌跡 (Case 1-1)

第5図 飛行軌跡 (Case 1-2)

- ・速度: 20 [m/s], 25 [m/s], 30 [m/s] の3通り
- ・方位角: $i\pi/4$ ($i = 0, \dots, 7$) の8通り
- ・ $t = 60$ [s] での原点からの距離: 0 [m], 100 [m] の2通りの組み合わせの中からランダムに選択し, 方位角については1ケース内で重複が無いようにした. 3機の衝突回避経路生成問題, 4機の衝突回避経路生成問題のそれぞれについて, 20ケースずつのシミュレーションを実行した. いずれのケースにおいても, 原点近傍ですべての参照経路が干渉する設定となっている.

これらのケースのシミュレーション結果を第2表および第3表に示す. “Avg. time” と “Max time” は実際に要した計算時間の平均値と最大値を表す. 最後の行の “Avg. of 20 cases” は20ケースの問題の平均値を表している. シミュレーションを1台の計算機上で行っているため, 3機のケースよりも4機のケースの方が平均計算時間, 最悪計算時間ともに増加する傾向を持つが, これらの平均値を機

第6図 飛行軌跡 (Case 1-3)

第7図 飛行軌跡 (Case 1-4)

第8図 飛行軌跡 (Case 1-5)

数で除すとほとんど差が無い. 用いた CPU がマルチコア対応で2つ以上のスレッドを効率良く並列処理する能力を持つため, 実際の計算時間を機数で除した値が1機分の計算時間に対応するわけではないが, 大雑把な傾向を捉える上では, この値を1機分の計算量の程度を表す指標として扱っても差し支えないと思われる. 従来手法では機数が3機から4機に増えただけで計算時間が急激に増加する傾向を持つのに対し, 上記の結果から, 提案手法では機数の影響を受けにくいことが分かる.

また, 第2表, 第3表における “CPA distance” は CPA

第 2 表 3 機の衝突回避経路生成におけるランダムシミュレーションの結果

Case	Avg. time [ms]	Max time [ms]	CPA distance [m]
2-1	44	140	495
2-2	33	187	500
2-3	19	47	500
2-4	30	93	500
2-5	38	124	500
2-6	38	140	500
2-7	58	187	500
2-8	33	187	500
2-9	59	234	500
2-10	39	219	500
2-11	40	110	500
2-12	32	93	500
2-13	48	109	500
2-14	28	156	500
2-15	38	78	500
2-16	42	188	500
2-17	44	188	500
2-18	31	93	500
2-19	29	250	500
2-20	29	109	499
Avg. of 20 cases	38	147	—

第 3 表 4 機の衝突回避経路生成におけるランダムシミュレーションの結果

Case	Avg. time [ms]	Max time [ms]	CPA distance [m]
3-1	57	344	500
3-2	32	125	499
3-3	37	78	500
3-4	47	188	500
3-5	49	203	492
3-6	46	250	465
3-7	42	141	500
3-8	53	249	500
3-9	52	203	500
3-10	31	156	500
3-11	40	203	500
3-12	33	110	500
3-13	52	296	500
3-14	64	266	499
3-15	31	125	500
3-16	42	110	500
3-17	40	187	500
3-18	49	188	500
3-19	46	203	500
3-20	49	312	494
Avg. of 20 cases	45	197	—

間の距離の最小値を表す。CPA 間の距離の最小値の有効数字を 3 桁とした場合、3 機の衝突回避経路生成問題では 20 ケース中 2 ケースが、4 機の衝突回避経路生成問題では 20 ケース中 5 ケースが必要間隔 $R (= 500 \text{ [m]})$ を下回ってい

る。しかしながら、最悪のケースにおいても必要間隔 R の 90%以上を確保できているため、経路の非干渉化という面では、提案手法が実運用上許容し得るロバスト性を有することが確認できる。

4. お わ り に

本稿では、複数の UAV の自律分散的な衝突回避経路生成手法として、QP と Force Field Algorithm を併用した手法を提案した。シミュレーションの結果から、提案した手法において、計算時間が機数増加の影響を受けにくいこと、および妥当な非干渉経路がロバストに得られることを確認した。今後の課題として、風の影響や高度方向の運動も考慮したより一般的な衝突回避経路生成問題に本手法を拡張することが挙げられる。

参 考 文 献

- 1) RTCA: Final Report of RTCA Task Force 3: Free Flight Implementation, RTCA Inc., Washington, D.C., 1995.
- 2) Eby, M. S.: A Self-Organizational Approach for Resolving Air Traffic Conflicts, Lincoln Laboratory J., **7** (1994), pp. 239–254.
- 3) Hoekstra, J. M., van Grent, R. N. H. W. and Ruigrok, R. C. J.: Designing for Safety: The ‘Free Flight’ Air Traffic Management Concept, Reliability Engineering and System Safety, **75** (2002), pp. 215–232.
- 4) Frazzoli, E., Pallottino, L., Scordio, V. and Bicchi, A.: Decentralized Cooperative Conflict Resolution for Multiple Nonholonomic Vehicles, AIAA Paper 2005-6048, 2005.
- 5) Schouwenaars, T., How, J. and Feron, E.: Decentralized Cooperative Trajectory Planning of Multiple Aircraft with Hard Safety Guarantees, AIAA Paper 2004-5141, 2004.
- 6) Bicchi, A. and Pallottino, L.: On Optimal Cooperative Conflict Resolution for Air Traffic Management Systems, IEEE Trans. Intelligent Transportation Systems, **1** (2000), pp. 221–232.
- 7) Tomlin, C., Pappas, G. J. and Sastry, S.: Conflict Resolution for Air Traffic Management: A Study in Multi-Agent Hybrid Systems, IEEE Trans. Automat. Contr., **43** (1998), pp. 509–521.
- 8) Menon, P. K., Sweriduk, G. D. and Sridhar, B.: Optimal Strategies for Free-Flight Air Traffic Conflict Resolution, J. Guid. Control Dynam., **22** (1999), pp. 202–211.
- 9) Frazzoli, E., Mao, Z. H., Oh, J. H. and Feron, E.: Resolution of Conflicts Involving Many Aircraft via Semidefinite Programming, J. Guid. Control Dynam., **24** (2001), pp. 79–86.
- 10) Raghunathan, A. U., Gopal, V., Subramanian, D., Biegler, L. T. and Samad, T.: Dynamic Optimization Strategies for Three-Dimensional Conflict Resolution of Multiple Aircraft, J. Guid. Control Dynam., **27** (2004), pp. 586–594.
- 11) Gill, P. E., Murray, W., Saunders, M. A. and Wright, M. H.: A Schur-Complement Method for Sparse Quadratic Programming, Reliable Numerical Computation, Cox, M. G. and Hammerling, S., eds., Oxford University Press, Oxford, U.K., 1990, pp. 113–138.
- 12) Ashcraft, C., Grimes, R. G. and Lewis, J. G.: Accurate Symmetric Indefinite Linear Equation Solvers, SIAM J. Matrix Analysis Applications, **20** (1998), pp. 513–561.