PSO (Particle Swarm Optimization) を用いた 小型 UAV (Unmanned Aerial Vehicle) の障害物回避技術

平野 貴之^{*1}, 中山 万希志^{*2}

An Obstacle avoidance technique of small UAV (Unmanned Aerial Vehicle) using PSO (Particle Swarm Optimization)

Takayuki HIRANO^{*1} and Makishi NAKAYAMA^{*2}^{*1,*2} Osaka Sangyo University

3-1-1 Nakagaito, Daito-shi, Osaka 574-8530, Japan

Received: 25 April 2018; Revised: 25 July 2018; Accepted: 28 October 2018

Abstract

We discuss a new obstacle avoidance technique for a UAV (Unmanned Aerial Vehicle) using PSO (Particle Swarm Optimization) called optimization imitated animal foraging. In this method, the avoidance orbit is generated by the response by step change of the target position in the left and right direction and the response by the step change of the target speed in the forward direction. The magnitude of the change of the two target values is determined by the PSO based on the unique evaluation function. The evaluation function is a function consisting of collision risk based on the closest distance to the obstacle, energy based on magnitude of step change of the target value to be applied, and avoidance time based on speed change in the forward direction. Here, the calculation of the closest approach distance is obtained by performing a simulation based on the mathematical model expressing the flight trajectory of the obstacle and the own vehicle each time the evaluation function calculation of the repetitive calculation in the optimization calculation of the PSO is performed. We confirmed that this technique is effective by the numerical simulation.

Keywords : PSO, UAV, Drone, Obstacle avoidance, Optimization, Simulation

1. 緒 言

近年、小型 UAV (Unmanned Aerial Vehicle) の高性能化や多様化が進んでおり、様々な利用方法が検討されている。先進国を中心として少子高齢化が進んでおり人件費を削減するために、省力化・無人化の促進や低コスト化に向けて UAV の活用が進められている。空中輸送用途での期待より、医療品などの緊急輸送や一般的な物流への利用も進められており、また撮影データから三次元地形データの測量や警備、インフラ点検などに対してもその活用が広まっている ((佐々, 2014), (鈴木, 2016), (野波, 2017))。さらに、リモートセンシングとしてのニーズがあり、人が立ち入ることのできない場所での状況把握や生育調査などの精密農業(野波, 2017)など、UAV の利用には多くの可能性があると考えられる。UAV の制御手法に関しても様々な研究が行われ、高度な制御ロジックも適用されてきている ((Azard et al., 2010), (河合他, 2013), (中山他, 2018), (鈴木他, 2010), (白井他, 2017))。しかし、いろいろな活用が広まり、多くの UAV が利用されると機体同士が衝突する危険性がしだいに高まってくると考えられる。そのため衝突することなくうまく回避する軌道を自動的に生成するシステムの構築が望まれており、近年の UAV に関する研究の大きな課題の 1 つとして衝突回避が挙げられる。この課題に対し先行研究では、小型 UAV を用いて時々刻々と変化する複数の機体と一定の間隔を取りながら、衝突回避が

可能な軌道を生成する提案（横山，2008）や交通事故を自動車と障害物の衝突と考え，これを回避する軌道を生成する提案（向井他，2006）などがあげられる．

前者の衝突回避経路生成手法は，計算時間が機体の数に影響を受けにくい Force Field Algorithm と 2 次計画法を用いた手法であるが，2 次計画法に帰着させるために速度を円の内接五角形，加速度を内接八角形の煩雑な制約条件に置き換えるなどの近似が必要となっている．

後者では，道路環境下で，等速直線運動する移動障害物に対して，禁止領域を設けた自動車が移動障害物を回避する問題を混合整数計画問題として解くことで最適な経路を生成する事ができると述べている．しかし，この手法では問題を数理計画問題として扱い易くするために，複数の OR 条件を新たな 0-1 変数を用いて AND 条件に変換したり，絶対値の項からなる評価関数最適化問題を混合整数計画問題として扱うために，新たな変数を用いて不等式条件として定式化を行うことが必要となっている．このように 2 つの先行研究で述べられている障害物を回避する軌道の生成には，速度や加速度，道路の幅や走行している自動車の大きさなどからなる制約を含んだ数理モデルを基にして，2 次計画法や線形計画法，混合整数計画法等を適用するために，定式化がかなり煩雑になり，設計を複雑にしている．

そこで我々は，障害物と自機の軌道を規定する時間関数の式を直接使い，独自の評価関数を考案し，評価関数さえ求めれば最適化問題を求解できる PSO (Particle Swarm Optimization: 粒子群最適化) 手法（石亀，安田，2008）により障害物を自機が回避する軌道を生成する手法を提案する．即ち，図 1 のように一般に水平面を飛行する正面・左右方向を独立に飛行制御可能な小型 UAV の正面等速直線運動する飛行軌道上に障害物が移動し衝突する場合に，自機を正面方向に対して 1 回の左右方向の位置目標値変更により左右に避けることと，正面方向に 1 回の速度目標値変更で加速，減速することにより，機体同士の最接近距離で表現する衝突リスクを抑えながらも同時に用いるエネルギーも削減できる簡易で効果的な回避方法を求める事に着目した．本論において，提案手法の説明とシミュレーションによりその効果を検証した結果について報告する．

Fig.1 An orbit flight where small UAV avoids obstacles.

2. PSO(Particle Swarm Optimization)を用いた小型 UAV の障害物回避技術について

2・1 PSO(Particle Swarm Optimization)を用いた小型 UAV の障害物回避技術の概要

本論では正面・左右方向に独立に飛行制御可能な小型 UAV を飛行させる際に，小型 UAV の飛行経路に障害物が移動接近し，飛行中の小型 UAV と衝突してしまうことを防ぐために，回避軌道を生成することを考える．衝突回避する軌道は，前方正面に対して左右方向の回避では目標位置の 1 回のステップ状の変更に対する 2 次遅れ系の軌道として生成し，正面方向の加減速による回避では目標速度の 1 回のステップ状の変更に対する 1 次遅れ系の応答に基づく位置の軌道として生成し，回避軌道の定式化を合理的な形で単純化した．また 2 つのステップ応答の大きさに対する最適解を探索するために，独自に評価関数を考案し，障害物と自機の最接近距離の自乗で表現できる衝突リスクを評価し，かつなるべく最小限のエネルギーで，さらに最短時間で飛行する事を考慮した関数として構築した．この評価関数を最適化するための手法として，粒子群最適化手法である PSO (Particle Swarm Optimization) を用いて，正面方向，左右方向のそれぞれのステップ状の変更ゲインに相当するパラメータを最適化し，衝突回避軌道を生成させる．以下具体的な回避軌道について詳細を述べる．

2・2 障害物と小型 UAV の左右・正面方向の衝突回避軌道について

前方より自機に接近する障害物は、ここでは議論を簡単にするために等速直線運動をしていると仮定し、図 2 (a) の状況で回避軌道を生成し、図 2 (b) に示すように障害物と自機の最接近距離と使用するエネルギーや所要時間も考慮した障害物回避を行うことを考える。

Fig.2 (a) Obstacle orbit and UAV orbit (At start). Fig.2 (b) Obstacle orbit and UAV orbit (At closest approach).

原点を現時点の自機の小型 UAV の位置とし、正面方向右向きを X 軸、正面方向を Y 軸とする座標系を地上座標系とする。自機は正面方向に等速直線運動をしている。一方障害物は、等速直線運動をすると仮定すると、その軌道 (x_1, y_1) は、媒介変数を時間 t とすると式 (1) および式 (2) で示される。ここで $a_{x_1}, b_{x_1}, a_{y_1}, b_{y_1}$ はそれぞれ対応する係数である。

$$x_1(t) = a_{x_1}t + b_{x_1} \quad (1)$$

$$y_1(t) = a_{y_1}t + b_{y_1} \quad (2)$$

Fig.3 Avoidance trajectory time function.

上記で規定できる障害物を自機の X 軸における位置制御系に対するステップ状の目標位置変更情報と、Y 軸におけるステップ状の目標速度変更情報をそれぞれ 1 回ずつ与えることにより、回避軌道を生成することを考える。すなわち、自機からみて左右方向にあたる X 軸においては、自動車走行における道路のように一定の幅の飛行経路内での回避を考えるために目標位置変更情報を与えることとし、進むべき正面方向である Y 軸においては加速減速による回避軌道を生成できるように速度変更情報を与えることにした。図 3 にその様子を示す。自機は正面前方方向に飛行しているとする。図 3 (a), (b) の緑線と青線は時刻 T_0 で共に交差しているため、回避軌道をとらない場合は、Y 軸上で自機と障害物は衝突する。これに対して、回避軌道の X 軸成分を、例えば図 3 (c) の水色線のように時刻 0 で X 軸のプラス方向に位置の目標値をステップ状に変化させ、図 3 (c) の赤線のようにその目標値に自機が追従飛行するものとし時間軸に対する X 軸方向の位置として算出する。Y 軸成分は、例えば図 3 (e) の紫色線のように時刻 0 で Y 軸方向の速度の目標値をプラス方向にステップ状に変化させ、図 3 (e) の桃色線のように、その目標値に自機が追従して速度が変化し、その変化に応じて図 3 (d) の赤線のように飛行するものとし時間軸に対する Y 軸方向の位置として算出する。その結果、時間軸に対する X 軸成分および Y 軸成分の位置が求まることになり、XY 平面上の回避軌道が定まる。本論では、X 軸方向（左右方向）の位置制御系を 2 次遅れ系として考え、また Y 軸方向（正面前方方向）の速度制御系を 1 次遅れ系として考え、XY 平面上の回避軌道を算出する。以下それぞれの方向に対する軌道生成の詳細について述べる。

2・2・1 自機の左右方向（X 軸）の回避軌道の生成

自機の正面方向に対して左右の方向にあたる X 軸において、まず実績の速度をフィードバックし、目標速度との差に基づき力を発生させ、それにより生じる加速度を積分することにより実績の速度が発生するという速度制御系をまとめて一つの 1 次遅れ系のブロックとして近似する。次に、これを内蔵し、位置情報をフィードバックし、目標位置との差に基づき速度目標値を修正する位置制御系において、その目標値に様々な大きさのステップ

Fig.4 Block diagram for the secondary order lag system.

関数を入力する事で、X 軸の座標値が 2 次遅れ系のステップ応答波形で表現できる回避軌道を生成する。その時のブロック線図を図 4 に示す。図において T は速度制御系の時定数、 K は制御ゲイン、 k_1 は位置制御系への目標値ステップ関数のゲインである。図 4 のブロック線図の閉ループ伝達関数を求め、 $\omega_n = \sqrt{K/T}$ 、 $\xi = 1/(2\sqrt{KT})$ と定義し、入力 r を単位ステップ関数とすると、次式が得られる。

$$X_2(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \cdot \frac{k_1}{s} \quad (3)$$

今回の回避軌道は、オーバーシュートを行わないものと考え、議論を簡単にするために $\xi = 1$ の時の特性多項式が重根の場合とすると式 (3) は式 (4) の様に変形することができる。

$$X_2(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \cdot \frac{k_1}{s} = \frac{\omega_n^2}{(s + \omega_n)^2} \cdot \frac{k_1}{s} \quad (4)$$

(4) 式において入力ステップ関数の大きさ k_1 の値が変化する事で、様々な軌道が生成できると考える。
 自機の X 軸上の位置、即ち小型 UAV の左右方向の軌道を $x_2(t)$ とし、式 (4) を逆ラプラス変換すると、式 (5) となる。

$$x_2(t) = L^{-1}(X(s)) = k_1(1 - e^{-\omega_n t} - \omega_n t e^{-\omega_n t}) = k_1(1 - e^{-\omega_n t}(\omega_n t + 1)) \quad (5)$$

式 (5) に基づき、左右方向での時間応答波形を作成した例 ($\omega_n = 1, k_1 = 1$) を図 5 に示す。

Fig.5 The left and right avoidance orbit of small UAV.

2・2・2 小型 UAV の正面方向の加速と減速による回避軌道生成

自機である小型 UAV は正面方向である Y 軸方向において、上述したように回避前は Y 軸方向に一定速度 v_0 で前進する等速直線運動をしていると仮定している。この時、障害物を回避するための操作として、Y 軸方向では加速と減速を考える。これらの速度変化は、UAV が内蔵している速度制御ロジックにおいて、目標値ステップ関数を印加することにより行い、この変更により、Y 軸上の時間に対する軌道を修正し、障害物を回避することができると考える。Z(s)を基準速度からの変化分とした時、速度一定ロジックのブロック線図を図 6 に示す。

Fig.6 Block diagram for the first-order lag system.

図 6 において、 $k_2 = k_2' K_2 / (1 + K_2)$ 、 $T_2 = T / (1 + K_2)$ と定義すると $Z(s)$ は式 (6) で示され、対応するブロック線図は、図 7 となる。

$$Z(s) = \frac{1}{T_2 s + 1} k_2 r \quad (6)$$

Fig.7 Block diagram for the first-order lag system.

上述した速度変更量を等速直線運動をしていた時の一定速度 v_0 に加えた場合のブロック線図を図 8 に示す.

$Y_2(s)$ は Y 軸上の自機の位置である.

Fig.8 Block diagram when constant speed is applied in the forward direction.

図 8 における入力 r を単位ステップ関数とすると, Y 軸上の自機の位置の時間関数である y_2 は, 式 (7) で示される.

$$y_2(t) = \int_0^t \left(v_0 + k_2 \left(1 - e^{-\frac{\tau}{T_2}} \right) \right) d\tau = v_0 t + k_2 t + k_2 T_2 e^{-t/T_2} - k_2 T_2 \quad (7)$$

式 (7) に基づき, 正面方向での時間応答波形を作成した例 ($v_0 = 1, k_2 = 1, T_2 = 1$) を図 9 に示す. Y

Fig.9 Distance in the direction which the own UAV moves forward.

軸方向は速度を制御することにより回避することになるが, 回避軌道を求めるためには, Y 軸方向の時々刻々の位置情報が必要になるため, 図 3 (d) の赤線のような時間応答波形を図 9 に示している.

2・3 回避軌道演算タイミングと障害物軌道情報の検出タイミング

次に, 回避軌道を求める時間タイミングや, 障害物の飛行経路の検出タイミング等について記載する.

図 10 にその概念図を示す．この図は横軸を時間，縦軸を UAV の前方面面方向である Y 軸とし，自機は Y 軸方向に時速 1m/s で等速直線運動をしているものとする．ここでは説明を分かり易くするために，自機は進行方向に対して左右方向のみで回避することにし，正面の Y 軸方向では加速減速は行わないものとする．図のハッチング領域は回避軌道を求める演算対象領域を意味する．また T_c を最適回避軌道を求めるために必要な計算時間とする．従って，必要とされる時間領域の回避軌道を求めるためには，その最適化計算を行うために， T_c 時間前に計算を開始する必要がある．図 10 では回避軌道演算対象領域を時刻 0 秒から 10 秒とし， $-T_c$ 時点で計算を開始することを表している．そのためには， $-T_c$ 時点において，ハッチングされた回避軌道演算対象領域，即ち，時刻 0 秒から 10 秒までに Y 軸で 0m から 10m の範囲に侵入する障害物の飛行経路情報はすべて把握できていることが必要である．また把握している障害物が計算終了以前に自機と衝突しないこと，すなわち図 10 の等速直線運動をしている障害物 A においては，境界線 L_0 より軌道がグラフ上で上側に存在していることを前提とする．また回避軌道演算終了後，回避軌道を飛行している最中に，新たに障害物が回避軌道演算対象領域に侵入することはないとしており，1 回の軌道生成に基づく回避軌道が対象領域を脱出するまで継続することとした．本論では図 10 に示すように，障害物は 1 機のみとして，これが計測された後の回避プロセスについて報告している．

Fig.10 Relationship between own aircraft and obstacle.

3. 評価関数

前章までに記載した通り，本論での自機の衝突回避軌道は，X 軸方向（左右方向）はパラメータを k_1 とした 2 次遅れ系のステップ応答軌道で表現でき，Y 軸方向（正面方向）はパラメータを k_2 とした速度の 1 次遅れ系のステップ応答軌道で修正した距離関数の軌道として表現できる．従って，本論における衝突回避軌道はいかに合理的にパラメータ k_1, k_2 を設定するかの問題に帰着される．そこで本章においては，この 2 つのパラメータによる回避軌道の最適性を評価するための，評価関数を独自に導入する．衝突回避する基準として等速直線運動の障害物の軌道の式 (1)，式 (2) と自機である小型 UAV の軌道の式 (5)，式 (7) に基づき，図 3 のように 2 つの軌道が接近する時の衝突リスクと，自機が正面方向と左右方向のそれぞれの衝突回避を行うエネルギーと，自機が最終的に到達する時間を考慮した評価関数を考える．等速直線運動する障害物の軌道と自機の軌道の座標に基づき時間を 0 から適切なサンプリング時間ごとに軌道を演算（シミュレーション）することにより最接近距離の二乗を算出する事で，2 つの軌道が接近しすぎるリスクを表現する．次に消費エネルギー相当として，自機の X 軸（左右）方向では入力ステップ関数の大きさ k_1 の 2 乗の値，自機の Y 軸（正面）方向では減速する場合はエネルギーを蓄積できると考え評価関数の値が減少するようにマイナスの値となるよう，入力ステップ関数の大きさ k_2 で表現する．最後に，最終的に到達する時間は自機がもとの速度に対し時定数分時間がたった場合の速度を用いることとし，最終値の 0.63 倍の値で一定区間飛行する時間で表現することとした．ここでは一定区間を 10m とし，等速直線運動時の Y 軸方向の速度を 1m/s としており，到達する時間の項目を，式 (8) の第 4 項として表現した．以上の考えに基づき評価関数を式 (8) により導入する．式 (8) の中で， $\alpha \cdot 1/D^2$ の D^2 部分は，障害

物の軌道の座標 (x_1, y_1) と自機の回避軌道の座標 (x_2, y_2) を用いる事で, $D^2 = (x_1 - x_2)^2 + (y_1 - y_2)^2$ と表す事ができる.

$$J = \alpha \cdot 1/D^2 + \beta_1 k_1^2 + \beta_2 k_2 + \gamma \cdot 10/(1 + 0.63k_2) \quad (8)$$

ここで, α , β_1 , β_2 , γ は重みパラメータである.

重みパラメータは, 障害物回避に対して, 安全を重視するか, 到達時間を短くするかなど, システム利用者の価値感に基づいて, 試行錯誤的に決められるものとする.

4. PSO(Particle Swarm Optimization)について

4・1 PSOのメカニズムについて

PSO(Particle Swarm Optimization)は, 粒子群最適化という実用上十分な最適化を有する解を決定するための発見的近似手法のメタヒューリスティクスの一つである(石亀, 安田, 2008). 鳥や昆虫などの生き物の群れは, 群れの仲間の誰かが, 餌がある所や敵に襲われない安全な所などの良い場所を見つけると, 次第に仲間が集まって周辺を探索する. 図 11 のように生物の群れでの採餌行動より Particle(粒子)と呼ばれるランダムに配置されている探索点が群れを作り, 過去の行動履歴に従い動的に調整された速度に基づき解の空間を目的関数の値が良い方向へ移動させ探索するアルゴリズムである. 群れ全体で情報共有し粒子独自の情報と組み合わせ, 一定の規則に従い行動する.

Fig.11 Search point moves in the evaluation function of PSO is good direction.

4・2 PSOのアルゴリズムについて

前節で紹介したように, PSO では各粒子は最適値を求めて空間を探索することになるが, 本論では, x_{i1}^k を k_1 , x_{i2}^k を k_2 と考え, 2次元空間を探索する. 各粒子の速度情報ベクトルは式 (9) により修正を行い, 位置情報ベクトルは式 (10) により更新を行う.

$$\begin{aligned} \begin{pmatrix} v_{i1}^{k+1} \\ v_{i2}^{k+1} \end{pmatrix} &= \begin{pmatrix} w & 0 \\ 0 & w \end{pmatrix} \begin{pmatrix} v_{i1}^k \\ v_{i2}^k \end{pmatrix} + \begin{pmatrix} C_1 \cdot rand_{1i1} & 0 \\ 0 & C_1 \cdot rand_{1i2} \end{pmatrix} \begin{pmatrix} pbest_{i1}^k - x_{i1}^k \\ pbest_{i2}^k - x_{i2}^k \end{pmatrix} \\ &\quad + \begin{pmatrix} C_2 \cdot rand_{2i1} & 0 \\ 0 & C_2 \cdot rand_{2i2} \end{pmatrix} \begin{pmatrix} gbest_1^k - x_{i1}^k \\ gbest_2^k - x_{i2}^k \end{pmatrix} \end{aligned} \quad (9)$$

$$\begin{pmatrix} x_{i1}^{k+1} \\ x_{i2}^{k+1} \end{pmatrix} = \begin{pmatrix} x_{i1}^k \\ x_{i2}^k \end{pmatrix} + \begin{pmatrix} v_{i1}^{k+1} \\ v_{i2}^{k+1} \end{pmatrix} \quad (10)$$

Fig.12 Particle corrects velocity vector information, Update position vector information under the PSO.

ここで、 k は探索更新回数、 $rand_{*i*}$ は 0 から 1 の間に存在する一様乱数、 $v_{i1}^{k+1}, v_{i2}^{k+1}$ は更新した速度情報、 $x_{i1}^{k+1}, x_{i2}^{k+1}$ は更新した位置情報、 v_{i1}^k, v_{i2}^k は現在の速度、 x_{i1}^k, x_{i2}^k は現在の位置情報、 $x_{i1}^{k-1}, x_{i2}^{k-1}$ は 1 つ前の位置情報、 $pbest_{i1}^k, pbest_{i2}^k$ は各粒子が今まで探索してきた中での最良解、 $gbest_1^k, gbest_2^k$ はそれぞれ $pbest_{i1}^k, pbest_{i2}^k$ の最良解である。全粒子の中での i は粒子の番号、 w, C_1, C_2 は重みパラメータである。まず、各粒子の初期位置情報を x_{i1}^0, x_{i2}^0 、初期速度情報を v_{i1}^0, v_{i2}^0 としてランダムに実行可能領域に与える。次に粒子の中での評価関数の評価値が最小になる値を求める。速度ベクトル情報の修正を行い、位置情報の更新を行い更に評価関数が最小になる値を求める。最大反復回数に達するまで探索し群れでの最終的最適解と評価値を求める。求めた左右方向の回避軌道でのパラメータ k_1 と、正面方向での加減速による回避軌道でのパラメータ k_2 を用いて図 13 で表すような自機の回避軌道を生成させる。

Fig.13 Using the values of the avoidance orbit parameters k_1 and k_2 obtained by PSO, generation of avoidance orbit of small UAV.

5. 数値シミュレーション

(石亀, 安田, 2008)を基に PSO のプログラムを作成し Matlab (The MathWorks, Inc.の登録商標) を用いてシミュレータを構築した. 下記の図 14 は PSO のフローチャートを表す.

Fig.14 PSO's flowchart.

シミュレーションの条件として, 自機は速度 1m/s で正面方向に進んでいると仮定し, 回避しない場合は座標 (0,5m) で衝突するものとし, 回避時間は 10 秒とした. また左右方向の軌道を表現する式 (5) の ω_n を 1.0, 前方方向の軌道を表現する式 (7) の T_2 を 1.0 とした. PSO の条件としては, 解の探索を行う粒子の数を 10 個, 次元数を 2 次元, 重みパラメータ $w = 1$, $C_1 = 0.5$, $C_2 = 0.5$, 最大反復回数は, 200 回とする. 初期条件として, 各粒子の位置情報ベクトルと速度情報ベクトルをランダムに実行可能な領域に与える. またここでは最接近距離に対して一定の制約を設け最接近距離が必ず 0 にならないようにしている. 式 (1), 式 (2) の障害物の軌道と式 (5), 式 (7) の自機の軌道を表現する数式を用いて, 最接近する時の衝突リスクや自機の正面方向および左右方向それぞれの衝突回避するエネルギー, および到達時間を考慮した評価関数が最小となるように, 下記のように各粒子が式 (9) の速度情報ベクトルの修正を行い, 式 (10) の位置情報ベクトルを更新することで, $\mathbf{x}_{i1}^{k+1} = \mathbf{k}_1$ と $\mathbf{x}_{i2}^{k+1} = \mathbf{k}_2$ を求め, 左右方向の回避軌道による入力ステップ関数の値 $\mathbf{k}_1$ および正面方向での回避軌道の加減速による入力ステップ関数の値 $\mathbf{k}_2$ の最適解を求める. 最大反復回数に達するまで探索し群れでの最終の最適解を求める.

6. シミュレーション結果

図 15,16 にシミュレーション結果の一例を示す. この例では評価関数の重みパラメータはそれぞれ $\alpha = 4$, $\beta_1 = 5$, $\beta_2 = 5$, $\gamma = 5$ とした. 図 15 は等速直線運動している 10 本の障害物の軌道を示す. 図 16 は図 15 に対する 10 本の自機の回避軌道を示す.

Fig.15 Obstacle orbit graph.

Fig.16 Based on the formulation of the avoidance trajectory in the left and right direction and the front direction and the optimization of the avoidance trajectory using PSO based on the independently constructed evaluation function discussed in this thesis, we confirmed that the avoidable solution can be found without problem.

7. 考察

図 15,16 より本論で論じた左右、正面方向の回避軌道の定式化と独自に構築した評価関数に基づき、PSO を用いて最適化を行ったところ、回避できる解が問題なく求まった。図 15 における①の障害物に対して、図 16 における①の回避軌道により自機が回避した時の障害物と自機が最接近距離となる軌道上の位置を図 17 に示す。図より最接近している障害物の位置が Y 軸上の 6.0 上にあることから、時刻 4.0 秒において最も接近し(約 1.4m)、加速して障害物を回避しながら飛行している様子が確認できる。図 17 は PSO アルゴリズムにおいて、最適解を求める繰り返し計算を行うループ内で、毎回設定された k_1, k_2 に基づき、時刻 0 から 10 秒までの間を、0.1 秒ごとに式 (1), (2) に基づき障害物の軌道の計算を、また式 (5), (7) に基づき自機の軌道の計算を毎回行っており、その過程で、障害物と自機の軌道がもっとも近づいた時点を求めておき、評価関数が最適値となった場合の軌道と最接近距離(図 17 の灰色線)を表したものである。図よりその距離は、約 1.4m であり、シミュレーション期間中においては、これを下回ることはない。従って、障害物と自機が衝突することはない、回避できたと考える。また図 16 より、すべての回避軌道における Y 軸方向の移動距離が回避前の等速時の Y 軸方向の速度 (1m/s) による移動距離 10m を超えているため、すべての場合において加速飛行をしていることがわかる。このことから、この評価関数の今回設定した重みでは、全般的に加速することにより回避する方が、評価値がよくなることが分かった。また①~⑩の障害物の軌道を規則的に変更した場合でも、回避軌道はそれぞれ傾向なく求まっており、

最適化手法により解を探索することの意味付けもなされたと考える。今回のシミュレーションでは、プログラムの高速化や並列化による高速化は全く実施しておらず、CPU が Celeron 2957U のノート PC を用いて行ったが、すべて 3.5 秒以内で求解ができており、今後の速度向上も望めると考えられる。提案手法では、繰り返し計算を伴う最適化手法を用いており、時間的には従来手法に劣る点もあるが、障害物の軌道が時間関数として規定できる場合は、どのような軌道に対しても対処できるとともに、設計時に面倒な近似や定式化も必要ない。

さらに様々な制約を追加する場合でも、基本的には評価値が求まれば求解できるために応用可能性の点で利点があると考ええる。

Fig.17 Position where own-machine and the obstacle are the closest distance.

8. 結言

障害物と小型 UAV である自機が衝突するリスクと正面方向および左右方向での回避のエネルギーと自機が最終的に到達する時間に基づく独自の評価関数を構築し、PSO を用いて最適化することにより回避軌道を生成する手法を構築した。本論では、本手法の定式化と評価関数の考え方を述べ、シミュレーションにより評価した。結果として、本手法の有効性が確認できた。今後は、処理速度の短縮と、本論を拡張して回避した後に元の軌道に復帰する部分についても軌道が生成できるように改良していきたい。またドローンへの実装に向けて、動的障害物のセンシングについては、小型のレーザ距離計を想定しており、このレーザ距離計から得られる距離の時系列データに基づき、時間の多項式関数でフィッティングすることを考えている。さらに障害物が複数存在する場合についても同様の議論ができると考えられるが、今後シミュレーションにより明らかにしていきたい。

文献

- Azard, S., Kendoul, F. and Nonami, K., Visual servoing of quadrotor micro-air vehicle using color-based tracking algorithm, *Journal of System Design and Dynamics*, Vol.4, No.2(2010), pp.255–268.
- 石亀篤司, 安田恵一郎, 群れの知能:Particle Swarm Optimization, 知能と情報(日本知能情報ファジィ学会誌), Vol.20, No.6(2008), pp.829–839.
- 河合宏之, 村尾俊幸, 鶴尾有生, 藤田政之, 入力飽和を考慮した動的視覚オブザーバによる位置姿勢制御, 計測自動制御学会論文誌, Vol.49, No.6(2013), pp.639–645.
- 向井正和, 川邊武俊, 西羅光, 高木良貴, 出口欣高, モデル予測制御による自動車の障害物回避へのアプローチ, 自動制御連合講演会講演論文集, 第 49 回自動制御連合講演会(2006).
- 中山万希志, 奥井 博己, 木村 尚人, 栗屋 効典, 小型 UAV(Unmanned Aerial Vehicle)の動特性モデリングと最適サーボ系を用いた目標値追従制御, 日本機械学会論文集, Vol.84, No.857(2018), DOI:10.1299/transjsme.17-00362.

- 野波健蔵, 「空の産業革命」をもたらすドローンの課題と展望, 計測自動制御学会誌, Vol.56, No.1(2017), p.2.
- 佐々修一, 無人機の動向と課題, システム／制御／情報, Vol.58, No.2(2014), pp.51–56.
- 白井淳平, 山口崇司, 鷹羽浄嗣, むだ時間を考慮したドローンの遠隔ビジュアルサーボ制御, 第 61 回システム制御情報学会研究発表講演会(2017).
- 鈴木智, 中澤大輔, 野波健蔵, 田原誠, クォータニオンフィードバックによる小型電動ヘリコプタの姿勢制御, 日本機械学会論文集 C 編, Vol.76, No.761(2010), pp.51–60.
- 鈴木真二, 小型無人航空機<ドローン>による「空の産業革命」へ向けて, 日本機械学会誌, Vol.119, No.1174(2016), pp.501–505.
- 横山信宏, UAV の衝突回避経路生成に関する一手法——2 次計画法と Force Field Algorithm の併用——, 日本航空宇宙学会論文集, Vol.56, No.653(2008), pp.262–268.

References

- Azard, S., Kendoul, F. and Nonami, K., Visual servoing of quadrotor micro-air vehicle using color-based tracking algorithm, Journal of System Design and Dynamics, Vol.4, No.2(2010), pp.255–268.
- Ishigame, A. and Yasuda, K., Swarm Intelligence, Particle Swarm Optimization, Journal of Japan Society for Fuzzy Theory and intelligent informatics, Vol.20, No.6(2008), pp.829–839(in Japanese).
- Kawai, H., Murao, T., Turuo, U. and Fujita, M., Visual motion observer-based pose control with input saturation, Transactions of the Society of Instrument and Control Engineers, Vol.49, No.6(2013), pp.639–645(in Japanese).
- Mukai, M., Kawabe, T., Nishira, H., Takaki, Y. and Deguchi, Y., On model predictive control for pedestrian avoidance using mixed integer programming, Automatic control union conference lecture papers collection, the 56th Joint Automatic Control Conference(2006) (in Japanese).
- Nakayama, M., Okui, H., Kimura, N. and Kuriya, K., Dynamic modeling and optimal servo control system of a UAV which tracks the reference trajectory, Transactions of the JSME(in Japanese), Vol.84, No.857(2018), DOI:10.1299/transjsme.17-00362.
- Nonami, K., Drone's challenges and prospects leading to the industrial revolutions of the sky, Journal of the Society of Instrument and Control Engineers, Vol.56, No.1(2017), p.2. (in Japanese).
- Sasa, S., Trends and issues in unmanned aerial vehicle(UAV), SYSTEMS, CONTROL AND INFORMATION, Vol.58, No.2(2014), pp.51–56(in Japanese).
- Shirai, J., Yamaguchi, T. and Takaba, K., The visual servo control of drone in consideration of dead time, Proceedings of the 61th Annual Conference of the Institute of Systems, Control and Information Engineers(2017) (in Japanese).
- Suzuki, S., Nakazawa, D., Nonami, K. and Tahara, M., Attitude control of small electric helicopter by using quaternion-feedback, Transactions of the Japan Society of Mechanical Engineers, Series C, Vol.76, No.761(2010), pp.51–60(in Japanese).
- Suzuki, S., Toward the aerial industrial revolution using drone, Journal of the Japan Society of Mechanical Engineers, Vol.119, No.1174(2016), pp.501–505(in Japanese).
- Yokoyama, S., A method of collision-free path calculation for UAV : Combined application of quadratic programming and force field algorithm, Journal of the Japan Society for Aeronautical and Space Sciences, Vol.56, No.653(2008), pp.262–268 (in Japanese).