BVC を用いたクワッドロータの障害物回避手法の検討

本 仲 君 子* 三 好 誠 司*

Obstacle Avoidance using BVC for a Quadrotor

Kimiko Motonaka* and Seiji Miyoshi*

D. Zhou et al. proposed an algorithm based on the buffered Voronoi cells (BVC) for quadrotors to reach each given target position without mutual collision. However, in the method, it is assumed that all quadrotor positions are known and the control inputs for all quadrotors are calculated in a single computer. In this paper, we describe the method to compute the Voronoi diagram using only the obstacle information around a quadrotor in an unknown environment and to reach a given target position. The usability of this method is confirmed by simulation.

Key Words: Voronoi Cells, Quadrotors, Collision Avoidance

1. は じ め に

人の操縦なしに飛行し、与えられたタスクを実行する自律飛行型のクワッドロータに関する研究が盛んに行われている。特に近年は広範囲の効率的な探索や協調作業を目的とした複数のクワッドロータの同時動作に関する研究が多く行われており、複数台が同じ空間を飛行する状況が想定されている。複数のクワッドロータが個別に動作する環境においては、互いの衝突を回避するためのアルゴリズムが重要である [1]。D. Zhou らは、クワッドロータどうしの相互衝突を回避するためのボロノイ分割に基づくアルゴリズムを提案し、シミュレーションによりその有用性を検証した [2]。障害物回避のための手法としては RVO アルゴリズム [3]~[5] を用いた手法などすでに多くが提案されているが、速度や障害物の位置など複数の、かつ正確な情報を必要とするものが多い。一方で D. Zhou らの手法では機体の相対位置のみを用いて簡単に制御入力を決定することができ、速度や正確な位置情報といった多くの情報を含む通信が不要であることから、オンラインでの実装に適していると考えられる。

筆者らの先行研究において、D. Zhou らの手法を動力学シミュレータ V-REP 上で複数台のクワッドロータに適用し、その挙動と相互衝突回避性能について確認した [6]。しかし、D. Zhou らによる検証を含むこれらの検証においては、すべてのクワッドロータの位置が既知であることが仮定されていた。また、各機体の位置を母点としたボロノイ分割と、そこから得られるすべての機体の制御入力を管制塔において計算し、各機体に送信していた。これに対し、管制塔を配置せずに個々のクワッドロータがセンシングできる周囲の情報のみを用いて衝突回避を行うことができれば実際の運用において多くの利点がある。そこで本稿では、クワッドロータが機体周囲の障害物情報のみを用い

てボロノイ分割を行い、目標位置までたどり着くための手法を提案する。なお、クワッドロータどうしが上下にすれ違う場合には互いにローターによる風の影響を受けるため、左右に避けたほうが安全であると考え、本稿では水平方向の回避についてのみ検討する。さらに、従来手法を用いた場合、静止した障害物が複数散在する環境や、障害物が袋小路状に配置されている環境においてデッドロックが生じる場合がある。そのため D. Zhou らの論文においては、デッドロックを回避するために右手法などのアルゴリズムを組み合わせることが推奨されている。これに対し、本稿では従来手法の非常に単純な拡張によってデッドロックを回避する手法を提案する。これらの提案手法の性能は、シミュレーションにより確認する。

2. ボロノイ分割による衝突回避

2.1 先行研究の概要

D. Zhou らは、複数の機体が動作する環境を想定し、ボロノイ分割を用いた分散衝突回避アルゴリズムを提案した [2]。彼らの手法は、環境内の機体位置が既知である状況を想定し、すべての機体が相互衝突することなく、個々に与えられた最終目標位置へ到達することを目的としている。このとき、各機体が 1 ステップ移動することによってバッファ付きのボロノイ分割である buffered Voronoi cells (BVC) を計算する。その後、BVC の境界上に定めた一時目標点の方向に機体位置を更新する。この手法は、機体速度などの情報を必要とせず、大まかな相対位置情報のみで衝突回避が可能である。これは、位置計測において高い精度が得られない場合には生成されるボロノイ境界が正しい位置からずれる可能性があるが、BVC にバッファ幅を設けていることで多少の誤差を許容できるためである。よって高いセンサ精度を必要とせず、さらに計算が高速であるといった利点があるため、実機実装に適していると考えられる。

Fig. 1 に 5 台のクワッドロータに D. Zhou らの手法を適用した例を示す。ただし、図中の三角形が各クワッドロータの現在位置、星印が設定した最終目標位置、丸印が一時目標点、黒

原稿受付 2020 年 8 月 10 日

*関西大学システム理工学部

*Faculty of Engineering Science, Kansai University

■ 本論文は有用性(要素分野)で評価されました。

Fig. 1 Collision avoidance using BVC for multi-agent

の実線が機体位置から算出したボロノイ境界、色付きの実線が各機体の BVC の境界である。まず、各機体の位置を母点としてボロノイ分割し BVC を算出する。次に、自分の機体が含まれる BVC の境界上において、最終目標位置から最も近い点を一時目標点として設定する。機体は設定された一時目標点に向かって一定速度で移動し、時間ステップごとに BVC の再計算を行って新たな一時目標点を設定する。このとき、最終目標位置が自分の機体が含まれる BVC の内側にある場合には、機体は最終目標位置に向かって直進する。以上の処理を繰り返すことで、機体は他の機体との距離を保ったまま、最終目標位置へ移動することが可能となる。このとき、時刻 t における i 番目の機体の位置 $\mathbf{x}_i(t)$ からその一時目標点 $\mathbf{s}_i(t)$ へのベクトルを $\mathbf{P}_i(t)$ とし、以下の式で機体位置を更新する。

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + k_v \frac{\mathbf{P}_i(t)}{\|\mathbf{P}_i(t)\|} \quad (1)$$

なお、 k_v は速度を調整するためのゲインである。

2.2 機体の周囲情報を用いた障害物回避

先述した先行研究においては、ボロノイ分割の母点とすべきすべてのクワッドロータの位置が既知であり、相互衝突を回避するための制御入力を管制塔にて一括で計算する必要があった。これに対して実機実装を考える際には、1 台のクワッドロータがセンシングできる範囲の障害物情報のみを用いて回避を行えることも重要である。そこで本節では先行研究における BVC を用いた相互衝突回避手法を基に、機体の周囲情報を用いて障害物回避を行う手法について述べる。

まず、機体の自己位置を中心とした半径 r の円内を障害物情報の取得が可能なセンシング範囲と仮定する。このとき、機体の自己位置とセンシング範囲内の障害物の座標に加え、上下左右に四つの母点を追加し、ボロノイ分割を行う。これは無限遠となるボロノイ境界を避け、機体の位置座標と障害物を内包するボロノイ領域を常に閉じた状態にするためである。また、4 点を追加することで、周囲に障害物がなくボロノイ分割のための母点が一つしかない場合にもボロノイ分割が実行でき、機体を内包する BVC を必ず求めることができる。4 点の追加の方法は以下のとおりである。マージン距離 $m > 0$ を用いて、ボロノイ分割に用いる母点の (x, y) 座標の最小値 $x_{\min}, y_{\min}$ からマージン距離 m を引いたものと最大値 $x_{\max}, y_{\max}$ にマージン距離 m を足したものを

Fig. 2 Collision avoidance using BVC for single-agent

Fig. 3 Initial state of the simulation

$$x_a = x_{\min} - m, \quad x_b = x_{\max} + m$$

$$y_a = y_{\min} - m, \quad y_b = y_{\max} + m$$

から得られる

$$M = x_b - x_a + y_b - y_a \quad (2)$$

より、追加する 4 点の座標を $(x_a - M, y_a)$, $(x_b + M, y_b)$, $(c_x, y_a - M)$, $(c_x, y_b - M)$ と設定する。ただし、 (c_x, c_y) はボロノイ分割に用いる母点の中心座標であり、以下で表される。

$$c_x = \frac{x_{\min} + x_{\max}}{2}, \quad c_y = \frac{y_{\min} + y_{\max}}{2} \quad (3)$$

Fig. 2 に 1 台のクワッドロータに対して提案手法を適用した例を示す。図中の三角が機体の現在位置であり、機体を中心とした円がセンシング可能な範囲である。また、黒の実線はボロノイ境界、機体位置を含む着色された領域が機体の BVC である。BVC の境界上にある丸は一時目標点であり、星印は最終目標位置を示している。周囲の菱形は障害物を想定した点である。なお、センシング範囲の上下左右にある白抜きの円は前段落で述べた四つの追加された母点である。このとき、機体が障害物に衝突することなく最終目標位置に到達することを目的とする。ただし、本稿では動く障害物は存在しない静的な環境を想定する。

2.3 デッドロックの回避

本手法では、障害物が袋小路状に配置されている場合にデッドロックが生じる。その回避手法として、最終目標位置をランダムに変更することを提案する。なお、本稿では機体が 1 秒以上同じ位置にとどまった場合にデッドロックであると判定した。このとき、機体は最終目標位置 $\mathbf{g} = [g_x \ g_y]^T$ を以下の式を用い

Fig. 4 Simulation results

でランダムに動かした仮の目標位置 $\mathbf{g}_{\text{tmp}}$ に向かって移動する.

$$\mathbf{g}_{\text{tmp}} = \begin{bmatrix} g_x \text{sgn}(r_x) \\ g_y \text{sgn}(r_y) \end{bmatrix} \quad (4)$$

ただし, r_x と r_y は標準正規分布から得た乱数, sgn は符号関数を表している. 機体は生成された $\mathbf{g}_{\text{tmp}}$ に向かって 4 秒間移動した後, 再び式 (4) により生成した新たな $\mathbf{g}_{\text{tmp}}$ に向かってさらに 4 秒間移動する. その後, 元の最終目標位置 $\mathbf{g}$ に向かって移動を再開する.

3. シミュレーション

3.1 問題設定

本シミュレーションでは, 提案手法を用いることで, 二次元平面において 1 台のクワッドロータが初期位置 $\mathbf{x}(0) = [-10, -10]^T$ から最終目標位置 $\mathbf{g} = [10, 10]^T$ まで環境内にランダムに 100 個配置された障害物を回避しながら移動できることを確認する. ただし, クワッドロータの動力学モデルは考慮しておらず, 機体位置の更新は式 (1) を用いて行うものとする. シミュレーションには MATLAB を用いた. また, 速度調整ゲイン $k_v = 0.1$, タイムステップ $\Delta t = 0.1$, バッファ幅 $r_s = 0.5$, センシング半径 $r = 2$ とした. Fig. 3 に初期状態を示す.

3.2 結果と考察

Fig. 4 にシミュレーションの結果を示す. 4 枚の図はそれぞれ時刻 $t = 20, 40, 50, 62.8$ (最終目標点への到達時刻) における状態を表している. 図中の赤い実線はクワッドロータの軌跡である. また, 色のついた菱形は機体のセンシング範囲内に入った障害物であり, ボロノイ分割の母点として用いられて

Fig. 5 The situation that deadlock occurred

いる.

以上の結果より, MATLAB によるシミュレーションにおいては, 複数の障害物が点在する空間において周囲の障害物情報のみを用いて最終目標位置に到達できることが確認できた. 一方で, 障害物の配置によってはデッドロックが生じることも確認された. Fig. 5 にデッドロックが生じた例を示す. この例では, 障害物が密に配置され, 機体周囲が袋小路のような状態になっており抜け出すことができなかった.

最後に, 提案したデッドロック回避手法について検証する. 本シミュレーションでは, 障害物数を 200 とし, 速度調整ゲイン $k_v = 0.2$ とした. その他の設定は前述のシミュレーションと同様である. Fig. 6 に提案手法によるデッドロック回避の一例を示す. 4 枚の図はそれぞれ時刻 $t = 0, 9.4, 17.8, 55$ における状態を表している. 結果より, 時刻 $t = 9.4$ においてデッド

Fig. 6 Deadlock avoidance with proposed method

ロックが生じたが、目標点を一時的に変更することで袋小路を抜け出し、最終目標位置まで到達できることが確認された。ただし、提案手法では目標位置をランダムに動かすため、同じ場所を何度も往復する場合や、遠回りをする場合も見受けられた。しかし、元手法の単純な拡張でデッドロックを回避可能である点は有用であると考えられる。

4. お わ り に

本稿では、ドローンが障害物の存在する空間において障害物に衝突することなく目標位置まで到達するための手法について述べた。提案手法はD. Zhouらのボロノイ分割による相互衝突回避アルゴリズムに基づいた手法であり、MATLABによるシミュレーションにおいては、障害物を避けて最終目標位置に到達できることが確認された。一方で、デッドロックが生じることも確認されたため、元の手法を拡張する形でデッドロックに対処可能な手法を提案した。なお、本稿では水平方向の障害物回避についてのみ検討したが、三次元ボロノイ分割を用いることで、高さ方向も考慮した回避が可能であると考えられる。また、本稿では移動障害物のない静的な環境においてのみ検証を行ったが、複数のクワッドロータが同じ環境で個々に動作するような動的な環境における検証も必要である。これらの検証については今後の課題である。さらに、実機実装のため、センシングした障害物をボロノイ点に落とし込む方法や、機体の姿勢制御器への組み込み方法についても検討する必要がある。

謝 辞 本研究の一部は、JSPS 科研費(18K13782, 18K11483, 20K04494)の助成を受けたものである。また、本取組の一部は2019年度関西大学教育研究高度化促進費において課題「革新的イノベーション創生を目指した取組」として促進費を受け、その成果を公表するものである。

参 考 文 献

- [1] S. Ramasamy and R. Sabatini: "A Unified Approach to Cooperative and Non-Cooperative Sense-and-Avoid," Enabling Low-Altitude Airspace and UAS Operations, pp.765–773, 2014.
- [2] D. Zhou, Z. Wang, S. Bandyopadhyay and M. Schwager: "Fast, on-line collision avoidance for dynamic vehicles using buffered voronoi cells," IEEE Robotics and Automation Letters, vol.2, no.2, pp.1047–1054, 2017.
- [3] J. van den Berg, M. Lin and D. Manocha: "Reciprocal velocity obstacles for real-time multi-agent navigation," Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA), vol.2, no.2, pp.1928–1935, 2008.
- [4] J. van den Berg, S.J. Guy, M. Lin and D. Manocha: "Reciprocal n-body collision avoidance," Proc. of 14th Int. Symp. Robotics Research, pp.3–19, 2011.
- [5] A. Giese, D. Latypov and N.M. Amato: "Reciprocal velocity obstacles for real-time multi-agent navigation," Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA), vol.2, no.2, pp.3234–3241, 2014.
- [6] 中川健人, 権裕煥, 本仲君子, 三好誠司: "ボロノイ分割を用いた複数のクワッドロータの相互衝突回避に関する数値的検証", 計測自動制御学会論文集, vol.56, no.1, pp.1–6, 2020.