

風車ブレードのダウンコンダクタ断線点検のための ドローン半自律飛行システムの開発

Development of a semi-autonomous drone flight system
for inspection of down-conductor disconnection of wind turbine blades

電子・機械技術部 電子・情報科 吉田英一、機械・加工科 菅野雄大
南相馬技術支援センター 機械加工ロボット科 三浦勝吏
委託企業 株式会社福島三技協

風車の落雷対策用接地線の断線点検用ドローンの実用化に向け、レーザ測距センサを使用し、ドローンから風車タワーまでの距離を一定に保つ機能及びカメラ画像からブレードを AI で検出し、ブレード先端まで接近する機能を備えた半自律飛行システムを開発した。また、本システムによるドローンの実証実験を行い、半自律飛行できることを確認した。

Key words: ドローン、半自律飛行システム

1. 緒言

風車の大型化や風力発電システムの増加に伴い、落雷による被害が増加している。風車に落雷すると、雷電流はブレード先端付近の受雷部（以降、レセプタという）から接地線（以降、ダウンコンダクタという）を通じて地中に流れる仕組みになっているが、ダウンコンダクタが断線した状態で被雷すると電流を地中に流せなくなり、ブレードの破損や飛散事故につながる。そのため、定期的にダウンコンダクタの断線点検を行う必要があるが、現在の点検方法ではロープワークで人がレセプタまで近づいて導通チェックを行うなどの高所作業が必要となっており、多くの危険を伴うことが問題となっている。

委託企業はより安全な断線点検を実現するため、風車ブレードのダウンコンダクタ断線点検用ドローンの開発を行っており、熟練の操縦者が導通チェック用プローブを搭載した点検用ドローンを目視で手動操縦し、ブレード先端のレセプタにプローブを接触させて断線点検できることを実証した。しかし、熟練のドローン操縦者以外でも風車に衝突せずに安全に飛行でき、レセプタに確実にプローブを接触させることができる半自律飛行の実現が強く求められている。

そこで、本研究ではドローンの半自律飛行による断線点検の実用化を目指して、レーザ測距センサ（以降、LiDAR という）及びカメラ画像の AI 認識を活用した半自律飛行システムの開発と実証実験を行った。

本研究は令和4年度地域復興実用化開発等促進事業（ロボット・ドローン分野）の採択テーマ「ドローンを使った大型風力発電用ブレードに内装されている雷対策用接地線の断線点検実用化開発」の一部として、委託を受けて実施した。

2. 実験

2. 1. 半自律飛行の概要

ダウンコンダクタ断線点検用ドローンへの適用を目指して半自律飛行システムを開発した。また、システムの動作を確認するため、図1に示すペイロード2[kg]のドローン機体を作製した。

本研究が目指す半自律飛行は、ドローンの姿勢を維持しつつ、風車タワーまでの距離を一定に保つ半自律飛行と、カメラ画像からブレードを AI 認識し、ブレード先端のレセプタ付近まで接近する半自律飛行で、離着陸や上昇下降は操縦者が手動操縦で行うこととした。

2. 2. 半自律飛行のシステム構成

ドローンに搭載した LiDAR、カメラ及びフライトコントローラ内蔵 IMU からのセンサ情報に基づき、飛行制御用 PC（以降、コンパニオン PC という）で作成した目標速度や向きなどの飛行制御の指令値をフライトコントローラに送信して、ドローンの半自律飛行を行う。

ドローン機体にはPixhawk 系フライトコントローラ



図1 飛行実験用ドローン

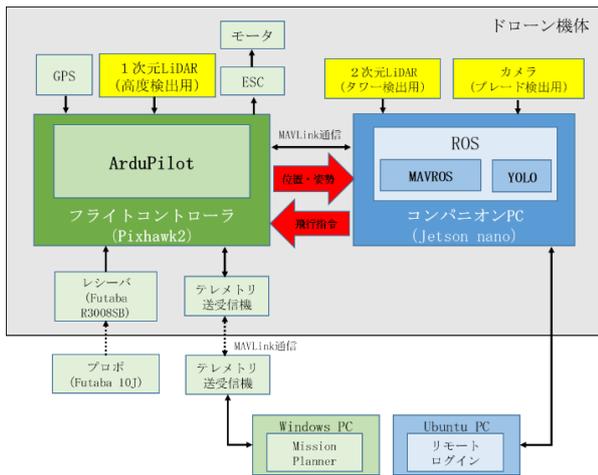


図2 半自律飛行システムの構成図

Cube オレンジとオープンソースの飛行制御ソフトウェア ArduPilot を搭載した。また、コンパニオン PC として NVIDIA Jetson Nano を搭載し、風車タワー検出用の LiDAR センサと風車ブレード検出用のカメラをコンパニオン PC の USB ポートに接続した。

ArduPilot は地上局とのテレメトリ通信に MAVLink というメッセージングプロトコルを使用しており、コンパニオン PC でロボットオペレーティングシステム¹⁾ (以降、ROS という) のパッケージである MAVROS²⁾ を動作させることで、フライトコントローラとコンパニオン PC がテレメトリ通信できるようにした。

これにより、コンパニオン PC で作成した飛行制御の指令値をフライトコントローラに送信できるようにした。

図2に半自律飛行のシステム構成図を示す。

2. 3. 半自律飛行の制御手法

2. 3. 1. LiDAR によるタワー検出と飛行制御

LiDAR により風車タワーを検出し、ドローンからタワーまでの距離を常に一定に保つことで、タワーに沿いながら安全にドローンを上昇させる半自律飛行が実現できる。

風車のタワーを検出するために、2次元 LiDAR である Slamtec 社製の RPLIDAR A3 を用いた。RPLIDAR A3 は低コストであり、測距半径が 25[m]、日光に対する信頼性の高い耐性を備えており屋外環境で使用可能である。

ドローンの姿勢を維持しつつ、風車タワーまでの距離を一定に保つ飛行制御の方法を図3に示す。ドローンの姿勢は、風車タワーに対し常に正対した姿勢を保つために、IMU により機体の向き (ヨー軸) を補正する。また、LiDAR で測距したタワーの点群データの座標値から重心 $G(x, y)$ を計算し、ドローンから重心 G までの距離 D が一定となるように、 x 軸方向の速度

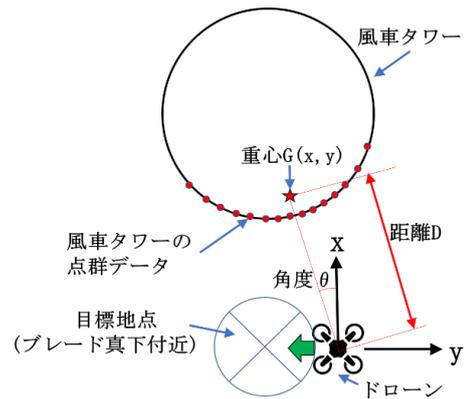


図3 タワー検出方法

V_x を制御する。その他、重心 G が常にドローンの前方 x 軸上に位置するようにするため、角度 θ を $0[^\circ]$ で一定となるように、 y 軸方向の速度 V_y を制御する。なお、地上付近では、タワー以外の物体を LiDAR で検出する可能性が高いため、ドローンが風車ブレード真下付近に位置する地上から離陸し、ホバリングした状態から上記の飛行制御を開始する。

飛行速度の制御方法には PID 制御を用いた。PID 制御はフィードバック制御の1つで、出力値と目標値の偏差 (Proportional)、積分値 (Integral) 及び微分値 (Differential) により出力値を制御する手法³⁾ であり、計算がシンプルで扱いやすい特徴がある。

2. 3. 2. カメラによるブレード検出と飛行制御

2. 3. 2. 1. 学習モデルの構築

風車ブレード先端への飛行制御をするために、ブレードをリアルタイムに検出する必要がある。そこで、リアルタイム物体検出 AI システムである YOLO^{4) 5)} を用いた。この項では、ブレードを検出するための学習モデルの構築について記す。

学習は当所で動作実績のある Darknet⁶⁾ を利用した。学習モデルは、風車のブレード (blade) とタワー (prop) の2クラスを検出できるように構築した。学習に用いた計算機の仕様を表1に示す。

学習に用いた画像は、国立研究開発法人産業技術総合研究所福島再生可能エネルギー研究所 (以降、FREA という) に設置されている風車の画像 500 枚⁷⁾ に、秋田県や飯館村で実施したドローンによるブレード点検実験で得た画像 700 枚⁸⁾ を追加した計 1200 枚の風車の画像を用いた。学習用の画像データの例を図4に示す。

2. 3. 2. 2. ROS による追従飛行制御

カメラ画像からブレードを AI 認識し、ブレード先端のレセプタ付近まで接近する半自律飛行を実現するため、以下のとおり追従飛行を実装した。

表 1 学習に用いた計算機の仕様

OS	Ubuntu18.04LTS
メモリ	32GB
CPU	Core i9 10900X
GPU	Geforce RTX2080Ti
GPGPUプラットフォーム	CUDA V10.1
学習モデル	YOLO-v3
フレームワーク	Darknet

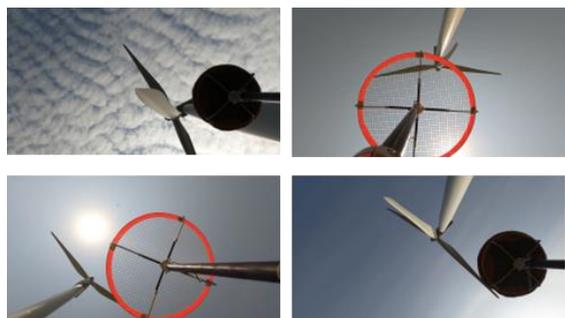


図 4 学習用画像データ例

YOLOにより物体を検出し、常に検出した物体がカメラ画像の中心に映るようにROSによってドローンを操作することで追従飛行を行うことができる。これは令和3年度に開発したROSによる追従飛行機能⁷⁾を改良することで今回開発したドローンによる追従飛行制御を可能とした。この機能は、YOLOにより物体検出する「物体検出プログラム」、物体検出プログラムにより得られた結果からドローンの制御信号を算出する「追従制御プログラム」、算出された制御指令通りドローンを操作する「操作プログラム」の3つのプログラムを構築し、「トピック」としてデータのやり取りをすることにより、物体追従飛行を行っている。

物体検出には、YOLOにより物体検出及びカメラ画像内の物体の座標値を検出するDarknet_ROS⁹⁾を用いた。検出したクラス名と検出した物体の位置情報をDarknet_ROSでは「/darknet_ros/found_object」というトピック名で送信している。

物体追従の制御は、自作した「blade_tracking」プログラムで行っている。ドローンのカメラ映像が640x480[px]であるため映像の中心座標(320, 240)を目標値として、検出した物体の中心が中心座標に移動するようにドローンのx軸、y軸方向の速度を算出する。ドローンの座標軸とカメラの向きを図5に示す。制御は比例制御のみで制御した。

算出結果をフライトコントローラと通信することができるMAVROSに与えることで指令通りの飛行が可能となる。「/mavros/setpoint_velocity/cmd_vel」というトピック名で速度データを送信する。制御のイメージを図6に示す。

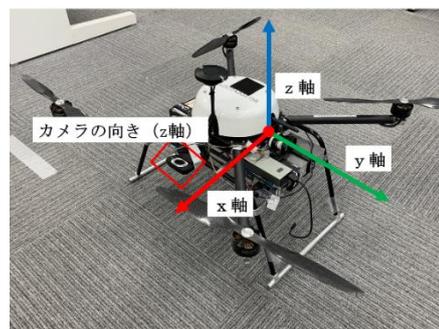


図 5 ドローンの座標軸

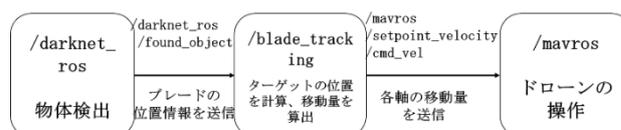


図 6 ドローンの制御イメージ

2. 4. 実証実験

ドローンに搭載したセンサを活用してドローンの半自律飛行が正常に動作することを確認するため、ロボットテストフィールド(以降、RTFという)浪江滑走路及びFREAで実験を行った。

2. 4. 1. スクエア飛行の実験

半自律飛行システムによりドローンの飛行制御が可能であることを検証するため、飛行実験を行った。図7に示すように図の①→②→③→④→①→②→③→④の各ポイントを順番でスクエア形状に飛行させた。ドローンへ指令する速度は0.8[m/s]とし、各ポイント間を5秒間飛行させた。

2. 4. 2. LiDARによるタワー検出と飛行制御の実験

LiDARの性能を確認するため、FREAの敷地内にある風車を対象に、タワーの測距実験を行った。

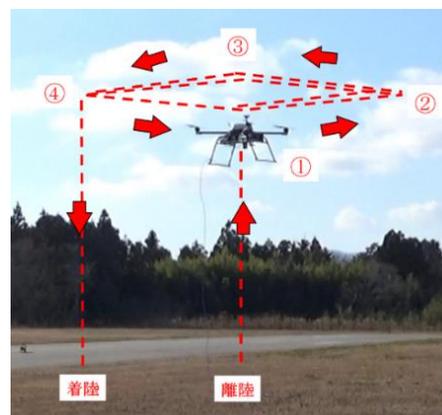


図 7 スクエア飛行のルート

また、ドローンの姿勢を維持しつつ、風車タワーまでの距離を一定に保つ飛行制御が正しく動作することを確認するため、ROSで動作する3DモデルのバーチャルシミュレータであるGAZEBOを用いた。ドローンの飛行を模擬した環境¹⁰⁾を使用し、図8のようにシミュレータ環境内にCADで作成した風車タワーを追加した。

2. 4. 3. カメラによるブレード検出と飛行制御の実験

2. 3. 2. 1項で構築した学習モデルを使用してブレードの検出実験を行った。FREAの風車を使用し、ドローンのカメラ映像からリアルタイムにブレードを検出できることを確認した。

実際に風車のブレードを点検する際は、点検するブレードを真下に向けて固定しドローンを風車の下から飛行させ、徐々に上昇しながらブレードに近づける必要がある。今回の実験では風車に搭載されているウィンチにドローンを吊るし、徐々に上昇することでドローンの飛行状態を模擬した。実験の様子を図9に示す。

また、RTF 浪江滑走路にて追従飛行実験を行った。滑走路に風車のブレードがないため、ドローンのカメラの向きを-z軸方向に変更し、ドローン用のランディングパッドを検出、追従するように設定し、物体追従飛行が可能であることを検証した。飛行時の高さは5[m]で飛行させた。

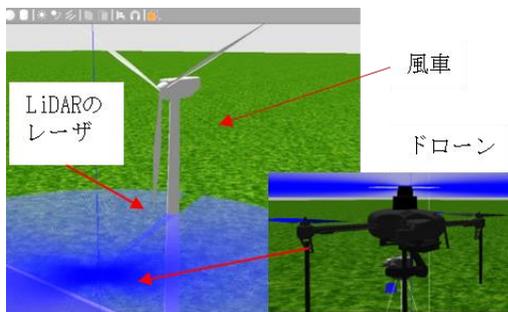


図8 ミュレーション環境



図9 実験の様子

3. 結果

3. 1. スクエア飛行の実験結果

スクエア飛行の軌跡を図10に示す。図のように、飛行軌跡がスクエア形状となり、各ポイント間の飛行距離も約4[m]となった。速度0.8[m/s]、飛行時間5[s]であることから、コンパニオンPCからの速度指令により正しくドローンを制御できることが確認できた。

3. 2. LiDARによるタワー検出と飛行制御の実験結果

FREA 風車タワーの検出結果を図11に示す。点群データの表示には、ROSで使用される様々なセンサデータを可視化するツールであるRVizを用いた。

図11のように、LiDARにより風車タワー側面の形状を点群データとして取得できており、風車タワーの検出にはLiDARを使用できることを確認した。

また、図12のようにシミュレーションを使用することで、安全にドローンの飛行制御の動作を確認でき、容易にPID制御の各ゲイン調整もできた。

今回、シミュレーション上で風車タワーまでの距離を一定に保つ飛行動作を確認できた。

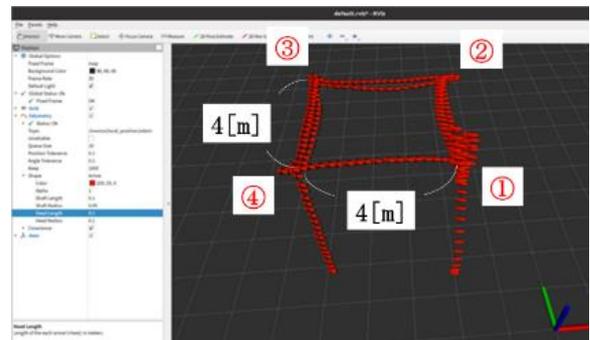


図10 飛行軌跡

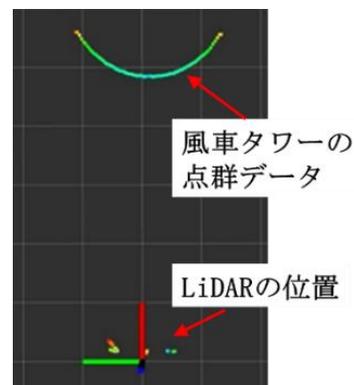


図11 風車タワーの点群データの可視化

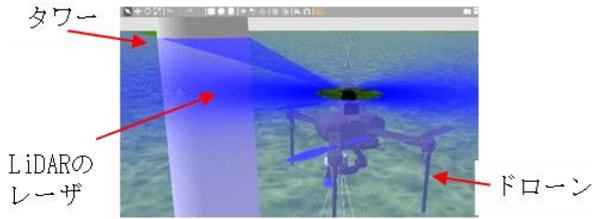


図 1 2 GAZEBO によるシミュレーションの様子

3. 3. カメラによるブレード検出と飛行制御の実験結果

FREA の風車を検出したところ、ブレード及びタワーをリアルタイムに検出していることが確認できた。検出結果の例を図 1 3 に示す。

また、RTF 浪江滑走路での追従飛行実験では図 1 4 に示すように x 軸、y 軸ともに目標値の前後を繰り返すような飛行をしたが、ランディングパッドを移動させると追従できていることが確認できた。

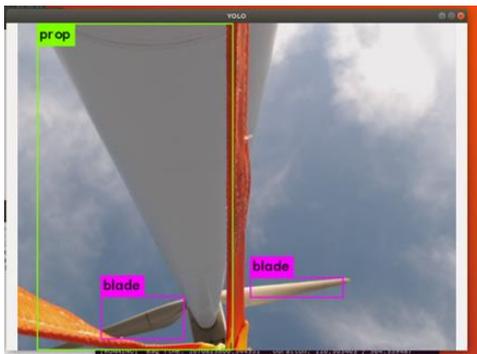


図 1 3 ブレード検出結果

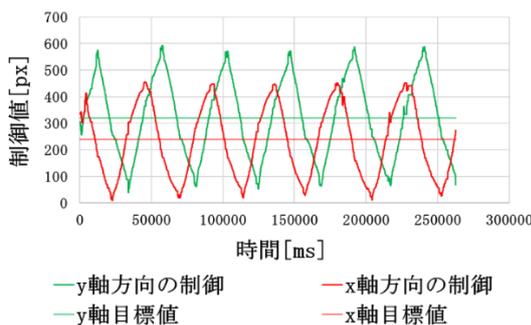


図 1 4 追従飛行結果

4. 考察

本システムによりドローンが半自律飛行できることを確認した。しかし、実験では飛行高度が上下に大きくぶれることがあった。フライトコントローラ内蔵の気圧センサで計測した高度の誤差が大きいことが原因

と考えられる。そのため、高度計測用 LiDAR を使用し、ドローンから地面までの距離を高精度に測ることで、機体のぶれを軽減するなどの対策が考えられる。また、ペイロード 2[kg] 以内という制約があるため、高性能で重量のあるコンパニオン PC 及び GPS を使った実験ができなかった。今後はペイロードの大きいドローンでの検証を実施していく必要がある。更に、シミュレーション環境を活用して実験回数を増やす工夫が必要だと思われる。

5. 結言

風車のダウンコンダクタ断線点検用ドローンの実用化を目指し、LiDAR と AI 画像認識を活用した半自律飛行システムを開発した。また、実験により半自律飛行システムを搭載したドローンで半自律飛行できることを実証した。

今後は、委託企業が作製したペイロードの大きいドローンに本システムを搭載し、風車を対象とした実験を行っていく予定である。

参考文献

- 1) “ROS.org | Powering the world's robots”. Open Source Robotics Foundation. <http://www.ros.org/>, (参照 2022-1-17). <https://github.com/pjreddie/darknet>. (参照 2022-1-17).
- 2) 吉田樹立, 他. “構造物の点検・補修を目指した吹付ドローンのための半自律飛行”. 日本ロボット学会誌, Vol.39 No.8, pp.759~762, 2021.
- 3) “自動制御の主力「PID 制御」とは”. 松定プレジジョン. https://www.matsusada.co.jp/column/pid_control.html, (参照 2023-2-1).
- 4) Joseph Redmon. “YOLO:Real-Time Object Detection”. <https://pjreddie.com/darknet/yolo/>, (参照 2022-1-17).
- 5) J. Redmon, S. Divvala, R. Girshick and A.F arhadi. “You only look once: Unified, real-time object detection”. arXiv preprint arXiv:1506.02640, 2015.
- 6) GitHub. “pjreddie/darknet”. Convolutional Neural Networks.
- 7) 三浦勝吏, 他. “AI を利用した風車点検用ドローンの自律飛行機能の開発”. 福島県ハイテクプラザ試験研究報告書, 2022.
- 8) 株式会社福島三技協. “風車ブレード実証試験データ”(2022.5.19 提供)

- 9) “darknet_ros –ROS Wiki”.
http://wiki.ros.org/darknet_ros, (参照 2022-1-17).
- 10) GitHub. “Intelligent-Quads/iq_sim”.
https://github.com/Intelligent-Quads/iq_sim,
(参照 2023-1-23).