システム制御特論
21P2033 山田竜輝

学術論文テーマ

シミュレーション環境での Lidar センサを用いた UAV の自動衝突回避システムの実装と評価

論文詳細

この論文では、シミュレーション環境（Gazebo ソフト）で、UAV model を設定し、外部 Lidar センサを UAV model に統合し、UAV の自動衝突回避システムの実装を行う。また、シミュレーション環境で、無事 UAV が障害物の衝突を回避できたかの検証を行う。

実験補足情報

実験補足情報として、今回の論文内容を説明するにあたり、説明するべき単語や補足情報を下記項目ごとにまとめた。

- UAV（搭載されている機器含め）

  今回シミュレーション環境（Gazebo）で使用する UAV は図 1 である。図 1 の UAV に関しては、Gazebo ソフトが提供している UAV であり、Iris クアッドコプターモデルと呼ばれている。

  Iris クアッドコプターモデル構成に関して、フライトコントローラは Pixhawk(図 2)を使用しており、加速度センサ、ジャイロセンサ、地磁気センサ、気圧センサが内蔵されている。また、外部センサとして、GPS が含まれているモデルとなっている。重量やプロペラ、モーターに関して、詳細不明である。(*論文で確認出来ず。)

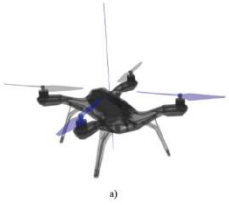

図 1 Iris クアッドコプターモデル

図 2 フライトコントローラ(Pixhawk)

- **Lidar センサ**

今回使用する Lidar センサは、Hokuyo URG-04LX である。Hokuyo URG-04LX の詳細に関して、箇条書きとして、以下にまとめた。

- レーザークラス 1　λ=785nm
- 範囲、20～10,000 mm
- 精度、60～1000mm；±30mm
- 1000mm から 10000mm まで　±3
- スキャンタイム、100ms
- 電源、DC5V + - 5
- 重量, 160 g

使用する理由は、信頼性の高さ、ペイロード（UAV 重量）のバランスが取れるからである。

上記の詳細パラメータを Gazebo ソフトで、パラメータ設定を行い、Iris クアッドコプターの外部センサ（Lidar）として、フライトコントローラ（Pixhawk）に統合する。

- **シミュレーション環境（Gazebo について）**

今回シミュレーション環境として、Gazebo を使用した。

使用した理由に関して、フライトコントローラ（Pixhawk）をサポートしているからである。将来現実環境で、UAV の衝突回避システムを実装するにあたり、フライトコントローラ（Pixhawk）を使用するためである。

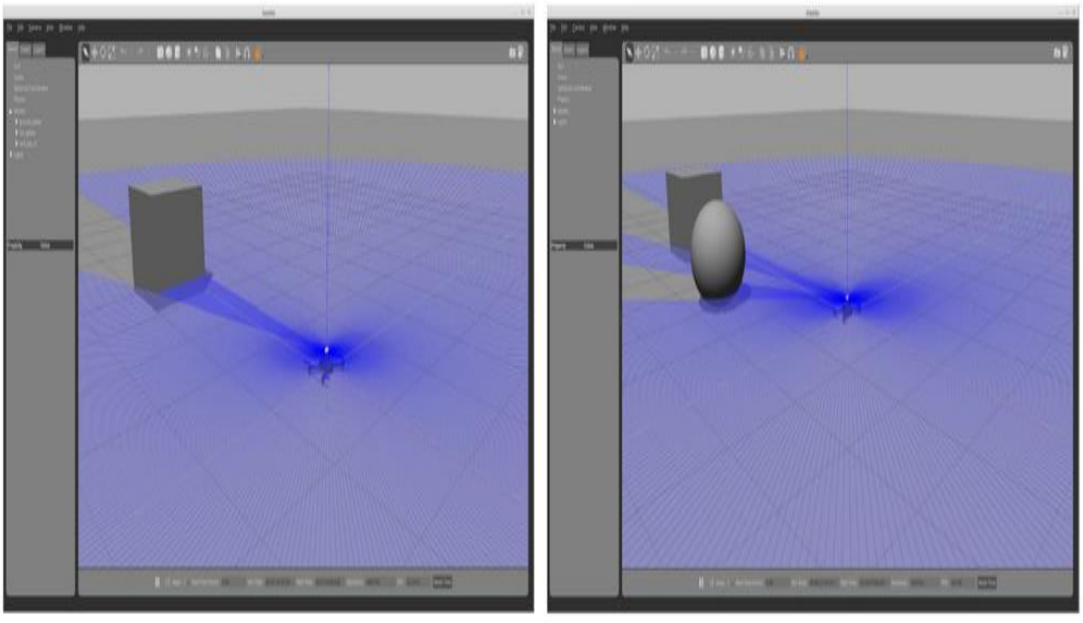

図 3 シミュレーション環境（Gazebo）

- **Q Ground Control**

  Q Ground Control に関して、サポートされているフライトコントローラを搭載している UAV の飛行経路を設定、自立飛行することができ、設定情報をフライトコントローラに書き込むことができるソフトである。また、UAV の飛行経路を MAP として可視化することが可能である。今回使用する Pixhawk はサポートされているフライトコントローラである。

図 4 Q ground Control

- **MATLAB/Simulink**

  MATLAB/Simulink に関して、センサ情報などをグラフとして表示し、センサ情報の分析、衝突回避システムの評価をするためのツールで使用する。

- **Pixhawk による衝突回避システムの処理内容について**

  Pixhawk による衝突回避システムの処理内容に関して、図 5 を見ていただきたい。衝突回避システムに関して、Iris クアッドコプターモデルの Pixhawk フライトコントローラ内部で処理される。

  今回の衝突回避システムのプログラム場所は、図 5 の Obstacle Avoidance Module である。シミュレーション環境のもと、Lidar センサで障害物との距離を Obstacle Avoidance Module で管理し、UAV と障害物で衝突しそうな距離になる際に、Obstacle detection＆tracking ブロックでトリガーを生みだし、回避させるための提案を Evasion logic にする。Evasion logic で、EKF の処理後の UAV の位置情報や姿勢情報を入力値として、回避処理を行う流れとなっている。

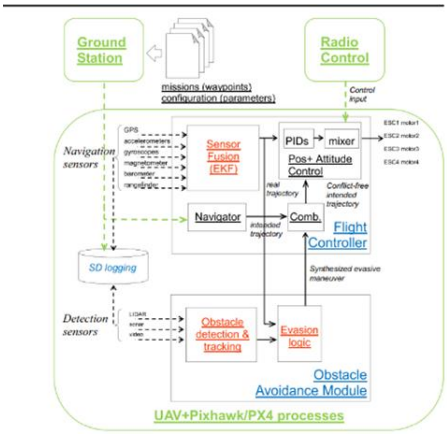

図 5 Pixhawk の内部処理に関して

実験手順

初めに、Gazebo ソフトで Iris クアッドコプターモデルを選択。選択後、衝突回避システム実装のために、今回使用する Lidar センサ情報を Gazebo ソフトで入力し、フライトコントローラ（Pixhawk）に統合する。そして Lidar センサによる衝突回避システムをプログラミングし、統合する。次に、統合がうまくいっているかどうかを Gazebo simulator で確認（図 5)。その後、衝突回避システムのための障害物（家）を Gazebo Simulator で用意する。(図 6)

次に Q ground Control ソフトを立ち上げ、Iris クアッドコプターの飛行経路を生成する。（図 7）今回一番左側を出発点として、直線経路を作成している。飛行経路を Pixhawk に書き込みを行う。

そして Gazebo シミュレーターで実行し、衝突回避の実装確認を行う。

最後に、Gazebo シミュレーターでの実行終了後、シミュレーター環境で得られたセンサデータを MATLAB/Simulink ソフトに反映させ、グラフ化し、衝突回避システムの性能評価を行う。

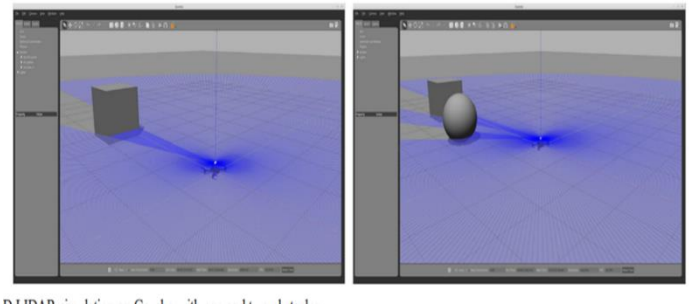

図 6 Gazebo simulator による Lidar センサの実装確認

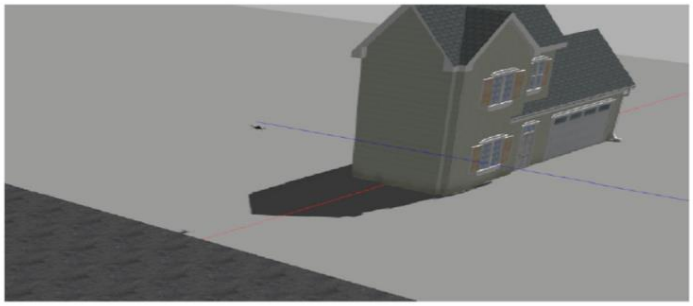

図 7 Gazebo Simulator による障害物生成

図 8 Iris クアッドコプターの飛行経路

実検結果

初めにシミュレーション終了後の Q groud control での飛行経路結果が図 9 である。赤丸印が障害物（家）である。図 9 からわかるように障害物を回避したことがわかる。

図 9 Q groud control での飛行経路結果（障害物回避）

次に直交座標系での衝突回避時の機体の速度状況を MATLAB/Simulink ソフトでグラフにした結果である。横軸は時間、縦軸が x,y,z 速度となっている。（図 10(a),(b)）比較対象として、障害物がない時のグラフも記載する。

図 10(a)に関して、障害物がない場合、離陸からミッション終了までの速度が一定である。速度の鉛直成分では、最初に UAV が上昇する際に変化しており、それ以降は一定である。図 10(b)に関して、UAV が Lidar センサで障害物を検知すると、衝突回避のためにブレーキとして、機体の速度を 0 もしくはマイナスになる。衝突回避後、機体の速度が 0 以上となり飛行目標経路に向かう。

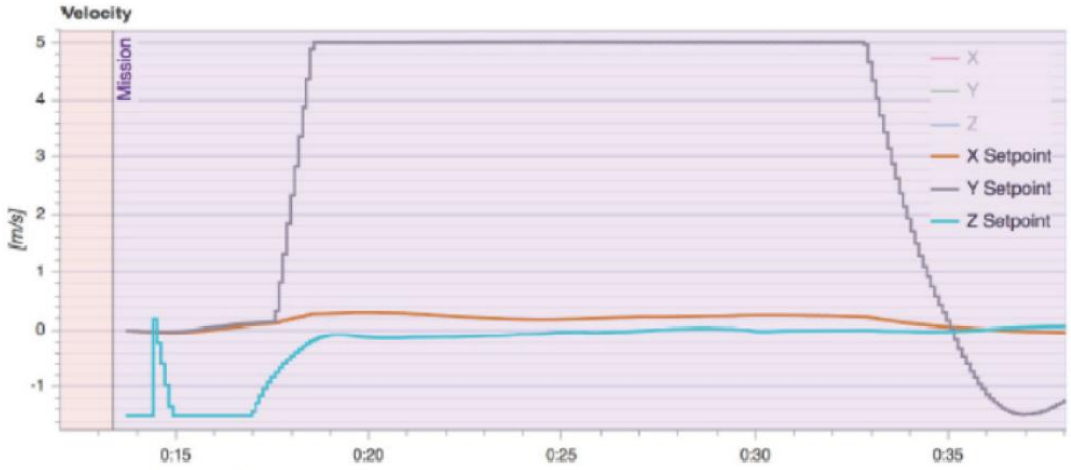

Fig. 18 Velocity without obstacles

図 10(a) 障害物がない時の機体の速度

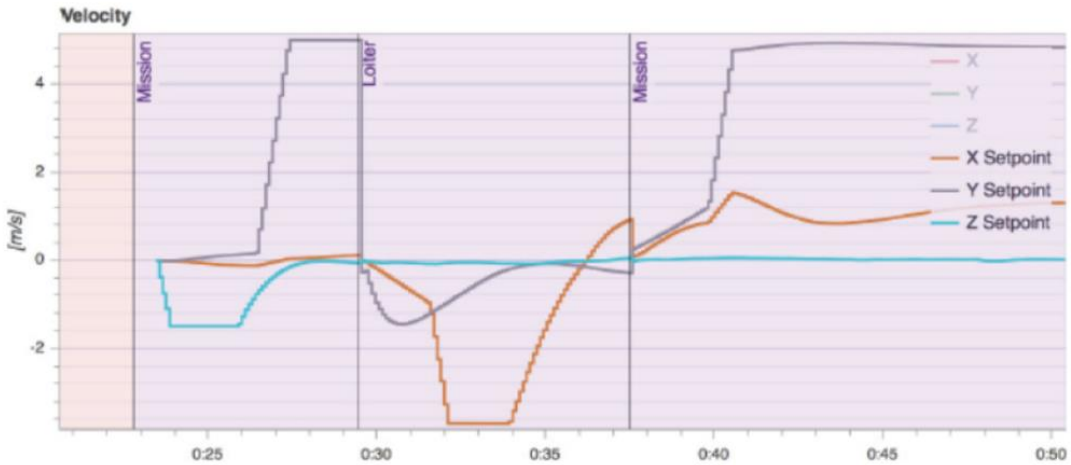

Fig. 19 Velocity with obstacles

図 10(b) 障害物（家）がある時の機体の速度

最後に機体のピッチに関してのグラフが図 11(a),(b)である。横軸は時間、縦軸が角度となっている。赤線は拡張カルマンフィルタで推定されたピッチである。図 11(a),(b)の初めの時間、UAV の鉛直離陸のためピッチが一定になる時間がある。図 11(b)では、衝突回避システムが起動すると、正の値までピッチが増加し、ドローンを停止させ、0 で安定し、再びピッチ上昇し、障害物を回避するようなグラフとなった。

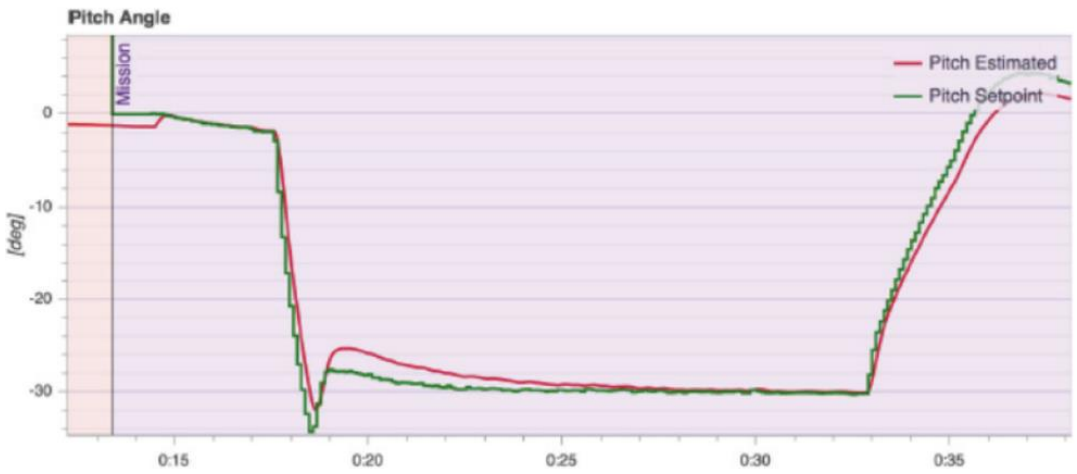

Fig. 20 Pitch without obstacle

図 11(a) 機体のピッチ（障害物なし）

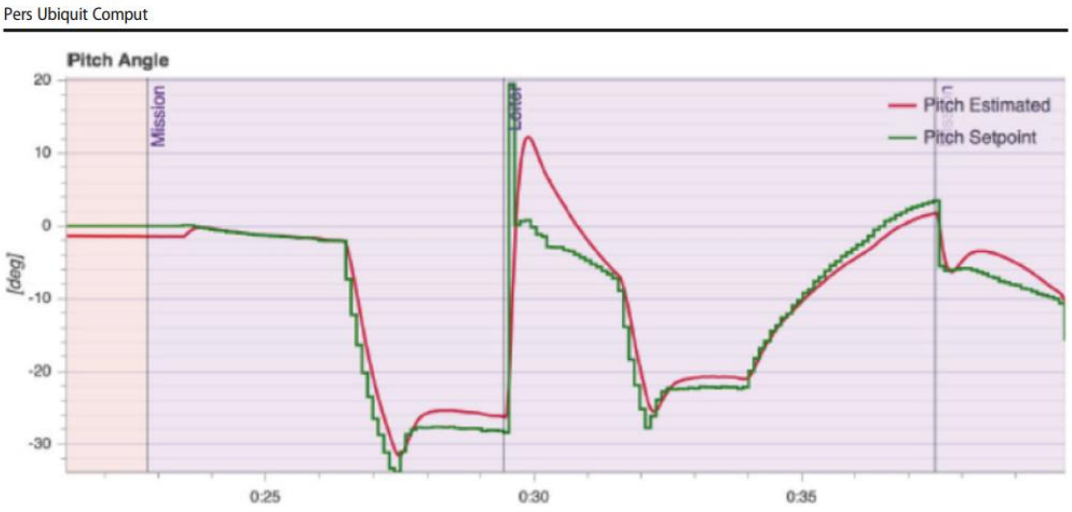

Fig. 21 Pitch with obstacle

図 11(b) 機体のピッチ（障害物あり）

自分の考察と思ったこと

考察として、衝突回避システムを実現するにあたり、UAV の高度情報がかなり重要になってくると考えられる。

理由として、図 12 の Lidar センサの検出範囲を拝見した際に、Lidar センサの縦軸の検出範囲がかなり短いのではないかと考えたからである。

今回の実験では Lidar センサを UAV の上に搭載されていた。(図 12) そして、UAV の高度に関して、今回障害物の高さ以下であり、障害物の衝突回避を行っていた。もし、UAV の高度が、図 12 の赤線を少し上回った高度の場合、Lidar センサは障害物を認識せず、UAV の下側部分が障害物にあたる問題が発生する可能性がある。

上記の解決方法として、Lidar センサの位置を変更するか、Lidar センサをもう一つ増やし、UAV の下側に取り付ける方法があるのではないかと思った。もう一つ取り付けるにあたり、上側の Lidar センサの光の干渉が起きないように気を付ける。

また、UAV にミラーを搭載し、Lidar センサの光の一部を屈折させ、Lidar センサの縦軸の検出範囲を増やすことが可能になるのではないかと考えられる。

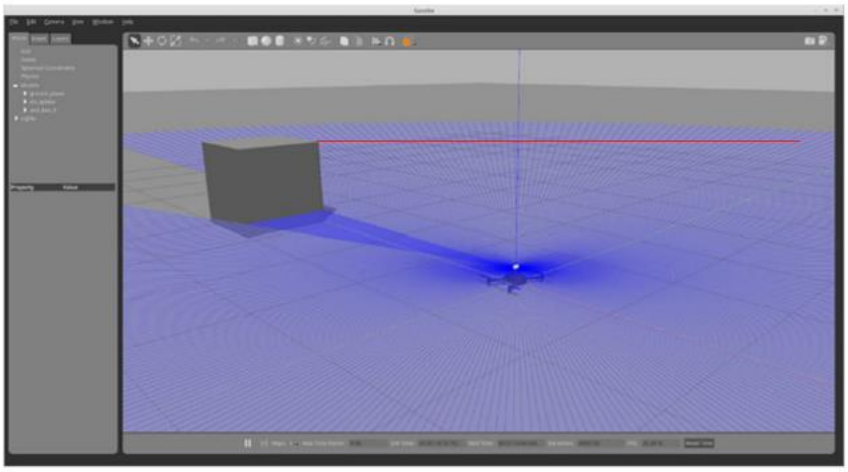

図 12　Lidar センサ取り付け位置・センサ範囲