

インドア飛行機の自動制御と 飛行中のデータの取得

著者 上杉廉 共同研究者 井口昇之 指導教員 高久有一

1. 研究目的

本研究では、前年度に研究されていた、インドア模型飛行機の自動旋回制御を改良し、より安定性を重要視したシステムを構築することを目的とする。またそのために、各種センサの値やサーボへの出力情報などを取得する仕組みを構築する。

2. インドア模型飛行機について

1. 模型飛行機とその制御

本研究では、制御対象として、自立安定性に優れたラダー機を利用する。図 1 に今回制御する模型飛行機を示す。



図 1 制御する模型飛行機^[1]

今回使用する模型飛行機は以下の 3 つの部位を操作することで制御する。

- ①スロットル(Throttle)
- ②エレベータ(Elevator)
- ③ラダー(Rudder)

スロットルは、機体の推進力を生じさせる機構である。エレベータは、機首を上や下に、ラダーは機首を左右に向けるために使用する。

この飛行機は、室内での飛行を想定し、壁や床への衝突に強く壊れにくい発泡素材を用いている。また修理もしやすいため、研究用の練習機として採用した。

3. 自動旋回制御

先行研究では飛行機の自動旋回制御として、ジャイロセンサを用いた定常飛行制御に加え、エレ

ベータオフセットを行うことで旋回中の高度維持を行った。しかし制御開始から約 30 秒経過すると、ジャイロセンサの誤差の蓄積により、高度が上昇あるいは下降するといった問題があった。そこで今回は、加速度センサから得られる機体の角度情報を用いて、自動旋回中の高度維持を行うプログラムを作成した。

1. 自動旋回

飛行機を旋回させるとき、機体を傾けるためにラダーを操作するが、それによって垂直方向の揚力が減少し、飛行機の高度が下がる。そのため急なラダー操作を行うと、飛行機が急激に墜落する危険がある。今回は、自動で滑らかなラダー操作を行うことで、急激な機体の降下を防ぐように制御した。その中で、舵を切る角度や速さによって旋回半径や高度減少の勢いを調節可能にするために、送信機のレバーを用いて旋回に必要な最大の舵角やそれに達する時間を変更できるようにした。さらに、旋回させるためには機体が傾いた状態で、エレベータ操作を行う必要がある。今回は加速度センサを用いて、機体の迎え角を検出し、機首が上向きになるように制御を行った。このとき目標とする迎え角も送信機のレバーで調節できるようにした。また、加速度センサの値は高周波ノイズを含んでいるため、過去の値を重みづけした指数移動平均によってノイズ除去を行った値を使用した。旋回に成功したときの迎え角は水平を基準として、上向き 3° であった。これにより、高度を維持したままの旋回を実現した。

4. 飛行データ取得プログラム

自動制御プログラムの開発において、センサの値を用いたり、手動での操作を模した記述を行ったりする。そのため、サーボに与えられる制御量や飛行機に加わる力などをデータとして確認できると、プログラムの開発の手助けになる。本研究では、飛行システムに影響を及ぼさずに、飛行中の様々なデータを取得するプログラムを作成した。

1. データ取得ヘッダ

データを取得するにあたって、汎用性を持たせるためにデータ取得に関するヘッダを作成した。ヘッダの内容を表 2 に示す。

表 2 ヘッダ部の内容

| data 順 | bit 数 | 名前 | 内容 |
|--------|---------|------------|-------------|
| 0 | 8 | hd_len | ヘッダ長 |
| 1~2 | 12 | dt_len | データ長 |
| | 3 | sampling | サンプリング周波数 |
| | 1 | none | 空き |
| 3~16 | 28×4bit | data_info | データ取得情報 |
| 17 | 5 | data_num | パラメータ数 |
| | 3 | auto_count | 自動モード回数 |
| 18~38 | 12 | auto_st_t | 自動モードに入った時間 |
| | 12 | auto_ed_t | 自動モードを抜けた時間 |

hd_len は、ヘッダ部の長さ (byte 数) である。dt_len は、ヘッダ部+データ部の長さ (byte 数) である。sampling は 1 秒間に取得するデータ数である。data_num は取得するパラメータ数である。auto_count はデータ取得中に自動モードに入った回数である。auto_st_t と auto_ed_t はセットで扱い、飛行機を自動モードにする度に最大で 7 セット分ヘッダ部に書き込まれる。

data_info には、4bit ずつパラメータの書き込み情報を記録する。詳細を表 3 に示す。

表 3 data_info の詳細

| bit 数 | 名前 | 内容 |
|-------|-----------|-------------------|
| 1 | available | 1…取得, 0…取得しない |
| 1 | type | 1…float, 0…int |
| 2 | cast | 00…x1, 01…x10 |
| | | 10…x100, 11…x1000 |

取得を想定した 28 個のパラメータ 1 つ毎にこの 4bit を設定するため、全体で 14byte 確保している。available は 1 のときそのパラメータを取得する。type はパラメータの型を設定する。cast は、float 型の変数を int 型にキャストするときの倍率を設定する bit である。キャストすることで、4byte から 2byte にデータ量を削減するために用意した。

2. 取得の方法

データは Arduino Mega mini に搭載されている EEPROM に書き込む。したがって最大で 4096bytes のデータが記録可能であり、そのうち 39bytes はヘッダ領域に使用する。現行の飛行システムでは、送信機の操作情報を 16ms 以下の周期で取得したい。しかし、ROM への書き込みは、Arduino の仕様上 1byte につき 3ms 要し^[2]、飛行中に大量のデータを書き込むと、飛行システム全体に影響が出るため、データを取得する度、ROM サイズと同じ大きさの配列に一時的にデータを格納し、格納が終わり次第、プログラムの 1 ループに付き配列の先頭から

1byte ずつ ROM へ書き込むようにした。現在のプログラムのループ周期は計測により 11ms であったため、これによりシステムに影響を与えることなく ROM への書き込みが行えた。

3. データの解析

図 2 に自動旋回時のジャイロセンサの値のグラフを示す。

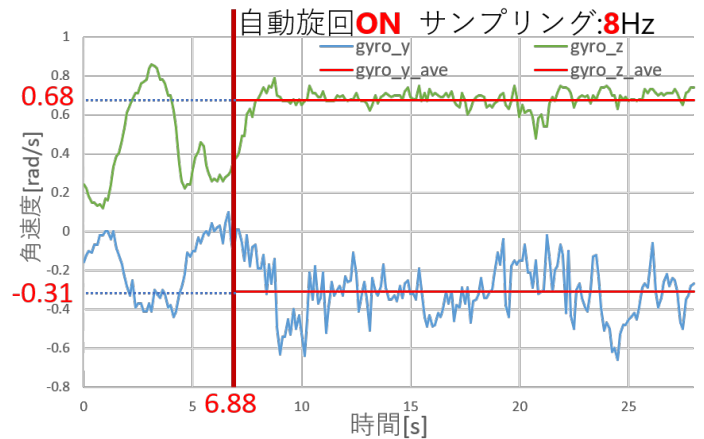


図 2 自動旋回中のデータ

グラフは、飛行中に自動旋回モードに入ったときのものである。データの取得を開始してから 6.88 秒後に自動モードに入った。旋回開始と同時に旋回方向の角速度である gyro_z が 0.68rad/s を軸として旋回していることが確認できる。また、機首の上下方向の角速度である gyro_y が減少し、以降値が激しく上下している。これは、加速度センサの値によるエレベータ制御で、機首が頻繁に上下しているからであると考えられ、正しくデータを取得できていることが確認できる。

5. まとめ

本研究で、加速度センサから得られる機体の迎え角を用いて、自動旋回を実現した。加速度センサを制御に用いるにあたって、ノイズが多い問題があったため、指数移動平均を用いてノイズを抑えた値を使用した。これにより、先行研究の問題点であった、自動旋回中に時間が経つにつれて機体が次第に上昇あるいは下降するという問題を解決し、常に同じ高度を保って旋回可能となった。また、飛行中の様々なデータをマイコンに搭載された EEPROM に記録し確認可能なシステムを開発した。

参考文献

- [1] トンボデザインクラフト
<http://www.tonbodesigncraft.server-shared.com/>
- [2] Arduino - EEPROM
<https://www.arduino.cc/en/Reference/EEPROM>