

平成 26 年度修士論文

マルチロータ型小型 UAV による
観測装置の移送および設置

指導教員

鬼頭 孝治 准教授

三重大学大学院

生物資源学研究科

共生環境学専攻

エネルギー利用工学教育研究分野

中嶋 雅佳

第1章	緒言	1
1.1	研究背景	1
1.2	研究目的	1
第2章	マルチロータ型 UAV	3
2.1	概要	3
2.2	搭載される主なセンサとその役割	4
2.2.1	ジャイロセンサ	4
2.2.2	加速度センサ	5
2.2.3	地磁気センサ	5
2.2.4	気圧センサ	5
2.2.5	GPS センサ	6
2.3	利点と課題	6
第3章	観測装置の移送・設置	7
3.1	基本概念	7
3.2	自動接近機能	7
3.2.1	基本アルゴリズム	7
第4章	実験装置	10
4.1	Quad 型マルチロータ型 UAV	10
4.1.1	本体	10
4.1.2	MultiWii フライトコントローラ(Crius All In One Pro v2.1)	11
4.2	センサユニット	12
4.3	超音波センサ(DFRobot SRF-02)	14
4.4	φ56 中空丸棒	16
4.5	扇風機(Kashiba EF-303R)	16
4.6	MultiWiiWinGUI ⁸⁾	16
第5章	実験方法, 結果および考察	18
5.1	予備実験	18
5.1.1	概要および目的	18
5.1.2	実験方法	18
5.1.3	実験結果	20
5.1.4	考察	23
5.2	維持機能	24
5.2.1	概要および目的	24
5.2.2	実験方法	24
5.2.3	実験結果	25
5.2.4	考察	30

5.3 接近機能	31
5.3.1 概要および目的.....	31
5.3.2 実験方法	31
5.3.3 実験結果	32
5.3.4 考察	34
5.4 探索機能	35
5.4.1 概要および目的.....	35
5.4.2 実験方法	35
(1)探索運動 1.....	35
(2)探索運動 2.....	36
5.4.3 実験結果	36
(1)探索運動 1.....	37
(2)探索運動 2.....	37
5.4.4 考察	39
第 6 章 結言	40
謝辞	41
参考文献	42
付録	43
ソースコード	43
A.維持機能.....	43
B.接近機能.....	44
C.探索機能.....	45

第1章 緒言

1.1 研究背景

現在、地球上では地球温暖化、ダイオキシン問題、酸性雨、オゾン層破壊など様々な環境問題が起きている。そして、その環境の変化が原因で、自然環境の破壊を起こし、中には種の存続が危ぶまれる生物も存在する。

今まで起こってきた環境問題に対して人間は、その現象が起きてしまったから、事態の重大さに気づき、それに対処をしていることが多い。例えば、地球温暖化問題では、地球の温暖化が 1900 年頃から始まり、その対策として国際的な取り組みが始まったのは 1980 年代である。このような、ある結果が出た後に対処し、さらに望ましい結果を得るという方法は、制御工学でいうならばフィードバック制御に相当する。制御工学では、より高度な制御を行うには、このフィードバック制御に、先を予測して行うフィードフォワード制御を併用することが要求される。

環境破壊問題を取り扱う場合、悪い結果が出てしまったから対処するという方法は、手遅れになる可能性が大いにあり、フィードフォワード制御のように、少しでも先を予測し、被害を最小限にしなければならない。少しでも先を予測するためには、地球上の二酸化炭素量の変遷や産業廃棄物の不当投棄による周囲環境への影響など、さまざまな情報を得ることが必要となる。それらの情報の一つとして、生物の情報収集も必要と考えられる。

また、自然環境の変化は生物へ影響を及ぼすため、逆にある生物を長期間観察することで、大気や水質、土壌などを観測することと同様に、自然環境の変化を捉える助けになり得ると考えられる。また、それと同様に、樹上の生物の生態調査や樹木から葉などのサンプルを採集することによって、周囲環境の情報を得ることも自然環境の変化を捉える助けになると推考する。

日本の森林では、恒久的森林保護の観点に立った野鳥の保護、拡大造林に付随して広がりつつある森林病虫害、大気汚染による影響などの問題が主に挙げられる。したがって、樹木上での野生動物の生育状況や樹木の病害の情報収集、資料のサンプリングは、そのような問題の解決につながるだけでなく、その生物の周囲の環境の変化も捉えることができ、環境破壊解決に有用であると考えられる。

樹木は上に行くほど枝が細くなり、木の葉が生い茂るなど、人間が容易に行くことが困難かつ高所であるため危険な場所である。そのため、高所にアクセスする手段として、人の代わりとなるロボットを使用することが考えられる。

1.2 研究目的

本研究の目的は、マルチロータ型小型 UAV を用いて、樹上環境観測装置を樹上に移送および設置し、得られた環境情報をもとに、樹自体だけでなく、樹の周囲の環境について知ることである。飛翔体であるマルチロータ型 UAV を用いることによって、人間が容易に行くことが困難、かつ危険を伴う高所へのアクセスを可能にしている。一例として、現在マルチロータ型 UAV が、人が容易に行くことのできない災害現場等において、情報収集を目的とした空撮を行うために利用されているという事例がある。これは UAV に搭載するセンサの制御技術の発達、機体の小型化が進んだことにより、狭小な場所への進入も可能となってきたことも挙げられる。本研究では、空撮のような UAV による直接的な情報収集に加

えて，長期観測を目的とした観測装置自体の移送・設置やサンプル採集を目的として，その制御技術の開発を目指す．

本論文では，高所の樹枝に観測装置を移送し，設置するための基本的な概念とその具体的方法を提案するとともに，試作装置および樹枝へ機体を自動で接近させるための基本アルゴリズムとその実証実験について述べる．

第2章 マルチロータ型 UAV

2.1 概要

マルチロータ型 UAV とは、マルチコプターとも呼ばれるヘリコプターの一種であり、2 つ以上のロータを搭載した回転翼航空機のことである。マルチロータ型 UAV は放射状に配置される複数のロータを同時にバランスよく回転させることによって飛行する。機体の飛行において、上昇・下降は全ロータの回転数の増減によって行い、前進・後進などの機体の移動は、機体前後、あるいは左右のロータ回転数に差を生じさせ、重心移動によって行っている。各ロータの回転方向と運動原理を図 2-1 に示す。太い下向きの矢印の大きさは回転数の大きさを意味している。また、ロータは固定ピッチの物がよく使われ、時計回り、反時計回りのものが交互に配置されることで、回転モーメントを打ち消し、機体自身がヨー軸に対して回転してしまうことを防いでいる。機体の姿勢制御にはマイコンを搭載したフライトコントローラが用いられ、搭載された各種センサから得られた機体情報をもとに、機体制御の演算が高速で行われることで機体の姿勢を制御している。

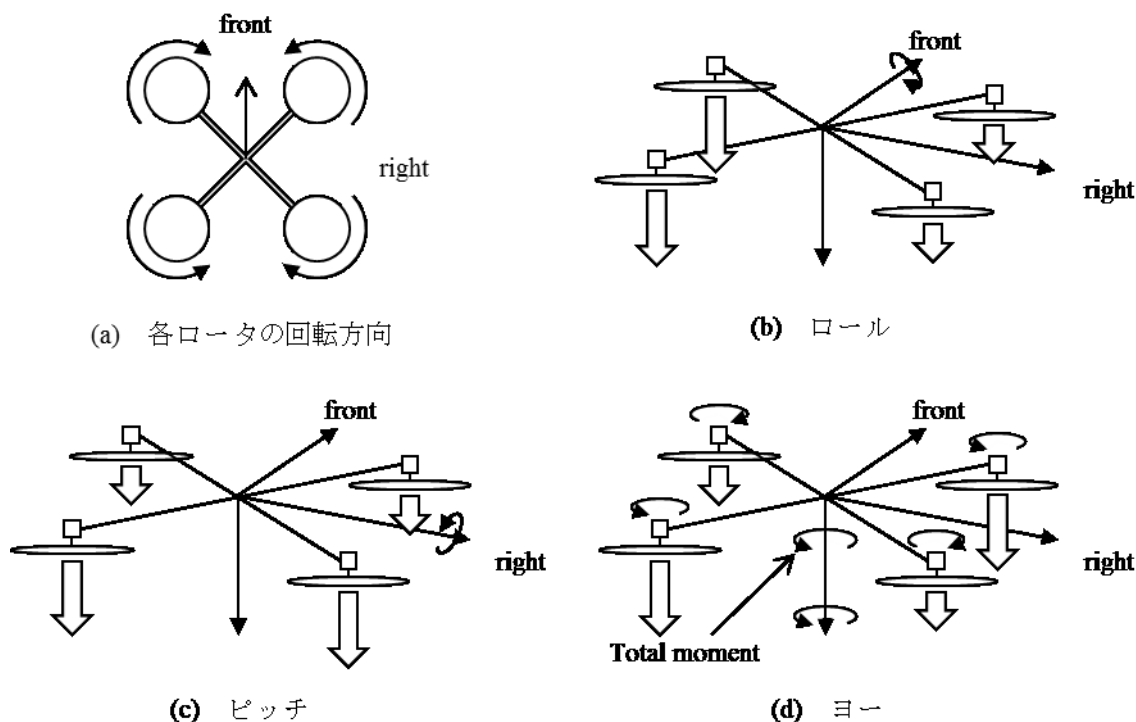


図 2-1 マルチロータ型 UAV の各ロータの回転方向と運動原理

機体の操縦もフライトコントローラを介して行う。まず、操縦者が送信機を用いて機体に指示を出す。その信号は受信機によって受け取られ、フライトコントローラに入力される。その入力をもとにフライトコントローラは、各モータに接続された ESC (Electric Speed Controller) にモータの出力の指示を出す。その結果として、モータの出力が変化し、機体の姿勢が変わり、操縦者の入力が機体に反映される。図 2-2 に操縦時の信号の流れを示す。

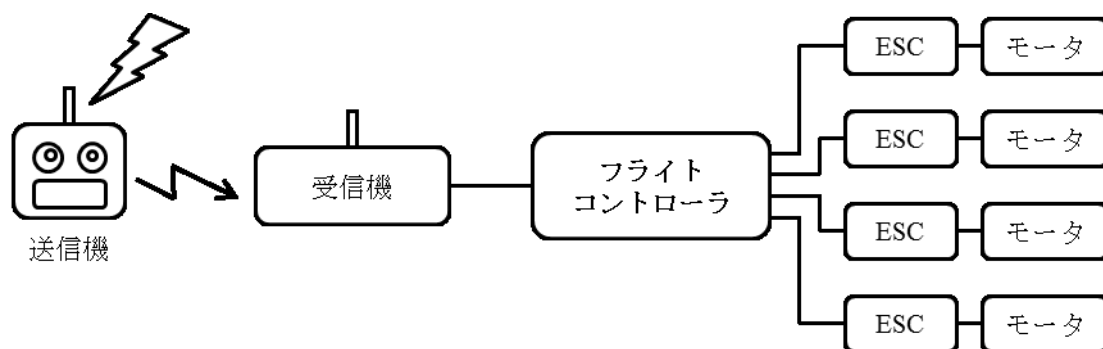


図 2-2 マルチロータ型 UAV の操縦時の信号の流れ

2.2 搭載される主なセンサとその役割

マルチロータ型 UAV に搭載される各センサは機体を安定させるために用いられる。機体を安定な状態で静止、つまりホバリングさせるためには機体の姿勢角と方位角の検出が必要となる。具体的には、進行方向を軸としたときの回転の度合いを示すロール、前後の傾きを示すピッチ、進行方向の向きを示すヨー角を得ることが大きな目的となる。

次に主なセンサについて述べる。

2.2.1 ジャイロセンサ

ジャイロセンサは機体の角速度を検出するセンサで、このセンサを用いることで、微小な時間での機体の姿勢の変化を検出し、元の姿勢に戻すために用いる。しかしながら、機体の姿勢角を得るためには得られたセンサ出力である角速度を積分する必要があるが、センサを機体に装着して読み取った値には、プロペラやモータの回転による振動、モータ駆動による電源ノイズなどによるノイズの混入が避けられない（図 2-3 参照）。その結果として、センサ出力を積分して得られた値と実際の値との間に乖離が生じてしまうため、機体の水平を保つために使うことはできない。図 2-4 に参考文献 2)より引用した、ノイズの混入による実際の値との乖離について示す。

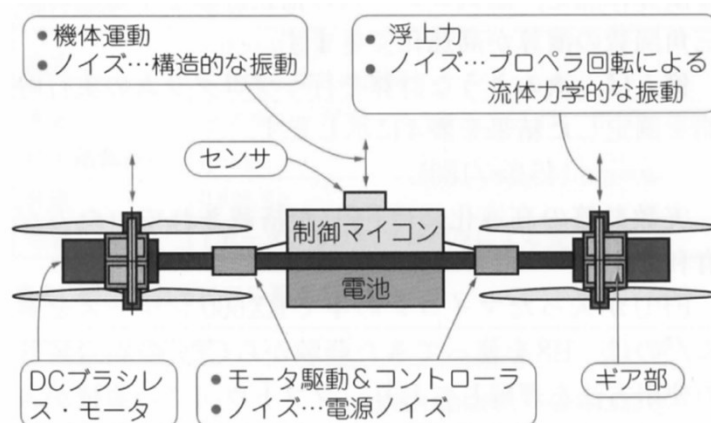


図 2-3 機体に装着したセンサへのノイズ²⁾

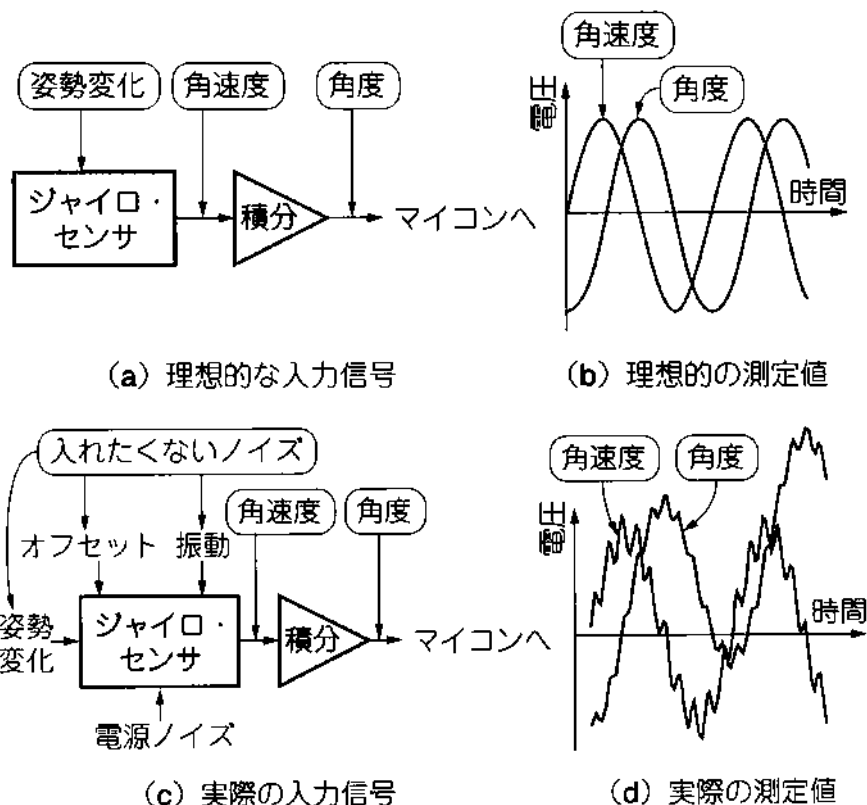


図 2-4 ノイズの混入による実際の値との乖離⁽²⁾

2.2.2 加速度センサ

機体の各軸についての加速度を検出するセンサで、加速していないとき、つまり静止しているか一定の速度で飛行しているときは重力を検出する。このセンサを用いることによって、機体のどの部位が上を向いているかが分かり、水平な姿勢を検出することができる。先述したように、ジャイロセンサだけではノイズによる誤差が積み重なるため、機体の姿勢を安定させることは困難だが、加速度センサを組み合わせることによってどのような姿勢からでも機体を水平に戻し、安定させることができる。

2.2.3 地磁気センサ

地磁気を検出するセンサで、加速度センサを用いることで機体の上下が分かるが、静止時に方向を検出することはできない。このセンサを用いることによって機体の向いている方向を検出することができる。

2.2.4 気圧センサ

気圧を測定するセンサで、このセンサを用いることで、高度を知ることができる。センサの能力によって増減するが、20～30cm 程度、あるいはそれ以下の高度差を検出できる。これを用いて、地上から一定の高さでホバリングさせることができる。

2.2.5 GPS センサ

先述した 4 つのセンサが正常に働いていれば、機体は一定の高度に静止し続けることができるが、風には流されてしまう。そこで、GPS センサを搭載することによって、一定の位置にとどまる、あるいは元の位置に戻ってくるといった機能を実現することができる。また、予めウェイポイントとして指定した位置を経由させて飛行させることもできる。しかし、GPS を用いるので屋内など受信感度の低い環境では使用することができない。

2.3 利点と課題

マルチロータ型 UAV はこれまでよく利用されてきたシングルロータの小型ヘリコプターと異なり機械的部分が少なく、構造が非常にシンプルなため制御が簡単で航空力学に精通していない人であっても比較的容易に扱うことが出来る。また、機体重量の 3 倍程度のペイロードがあることも、従来のシングルロータヘリでは実現できなかった仕様である。一方、欠点は飛行時間が 15～30 分と短いことであり、高性能なバッテリーの開発が待たれる。

第3章 観測装置の移送・設置

3.1 基本概念

マルチロータ型 UAV にテレメトリ式の観測装置を搭載し、目標とする樹木の枝にその装置を設置して、帰還することを基本概念としている。なお、装置の回収は命令信号による自由落下で行う。ここでの技術的課題は、UAV に搭載した観測装置の樹枝への装着技術、および目標とする樹枝への UAV の誘導技術である。

特に UAV の誘導について、ここでは機体を樹木の樹冠部にある程度接近させた後、機体が自動で樹冠部に進入、および目標とする樹枝へ接近するものとする。これは、樹冠部は離陸時に想定される周辺環境に比して狭小であり、樹枝との接触による墜落の危険性が高く、熟練した操縦技術をもってしても困難を伴うと判断したためである。要約すると、離陸および樹冠部への UAV の接近を手動で行った後、UAV が自動で樹冠部への進入と樹枝への接近を行う。その後、観測装置を樹枝に装着させ、樹冠部から離脱・着陸させることとなる。そのためには、樹枝へ機体を自動で接近させる機能を開発する必要がある。

次に、機体の樹枝への自動接近について述べる。

3.2 自動接近機能

先述したように、機体の樹冠部から樹枝への誘導は自動で行うものとする。この機能を自動接近機能とする。自動接近機能の仕様を表 3-1 のように設定した。

表 3-1 自動接近機能の仕様

- | |
|---|
| <ul style="list-style-type: none">● 機体がホバリングしている状態から機能を開始する。● 機体の直上に直線的な樹枝があることとする。● 樹枝は機体の前面から後面に伸びているものとする。● 機体が接近すべき目標距離は外部から与えられる。● 目標とする樹枝を見失った際は、樹枝を探索する。 |
|---|

3.2.1 基本アルゴリズム

自動接近機能の基本アルゴリズムは、機体が樹冠部付近でホバリングしており、目標となる樹枝が機体の直上にある状態から開始される。その状態で目標となる樹枝を検出し、距離を測定する。その後、機体は樹枝との距離が目標値に達するまで接近を行う。その最中、何らかの理由によって樹枝を見失った場合は樹枝を探索し、再発見した後で接近運動を再開する。このとき、探索は機体を左右に移動させることで行う。これは、機体が前後に移動したとしても、樹枝が機体の直上にあるという位置関係は崩れることはないが、機体が左右に移動した場合では、樹枝が機体の直上から移動してしまうためである。つまり、機体は樹枝に対して左右に移動することで樹枝を見失うが、同様にして樹枝を発見することもできるためである。なお、今回は樹枝の検出・距離の測定に超音波センサを用いた。

次に機体が樹枝を見失う場合について考える。機体が樹枝を見失うのは、機体が横方向に移動したときだということは先述したとおりである。この状況が発生するのは、不安定系である機体が左右に移動してしまうことで起こるが、この場合は移動によって生じた変位を補正することで樹枝を再発見することができる。それ以外では、樹枝へ接近運動を行

う際の機体の上昇に伴い、樹枝が後述する超音波センサの特性によって、超音波センサの検出範囲外になってしまうことが考えられる。このときの様子を、図 3-1 に示す。

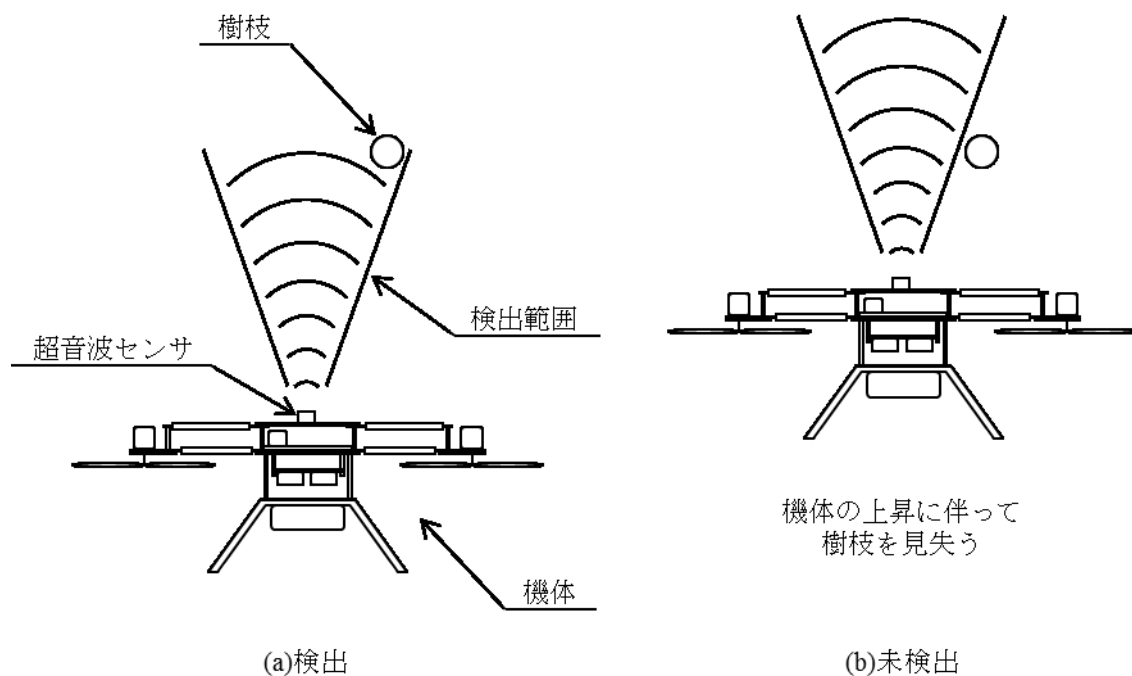


図 3-1 機体の上昇に伴って樹枝を見失う様子

このとき、検出に用いるセンサが 1 つしかない場合では、樹枝を左右どちらの方向に見失ったかが断定できない。よって、この場合では機体を、見失った点を基準にして左右に同程度動かすことによって樹枝を探索するものとする。

以上を踏まえ、自動接近機能のフローチャートを図 3-2 に示す。なお、自動接近機能の中でも実際に接近する部分を「接近機能」、見失った樹枝を探索する部分を「探索機能」と呼ぶことにする。

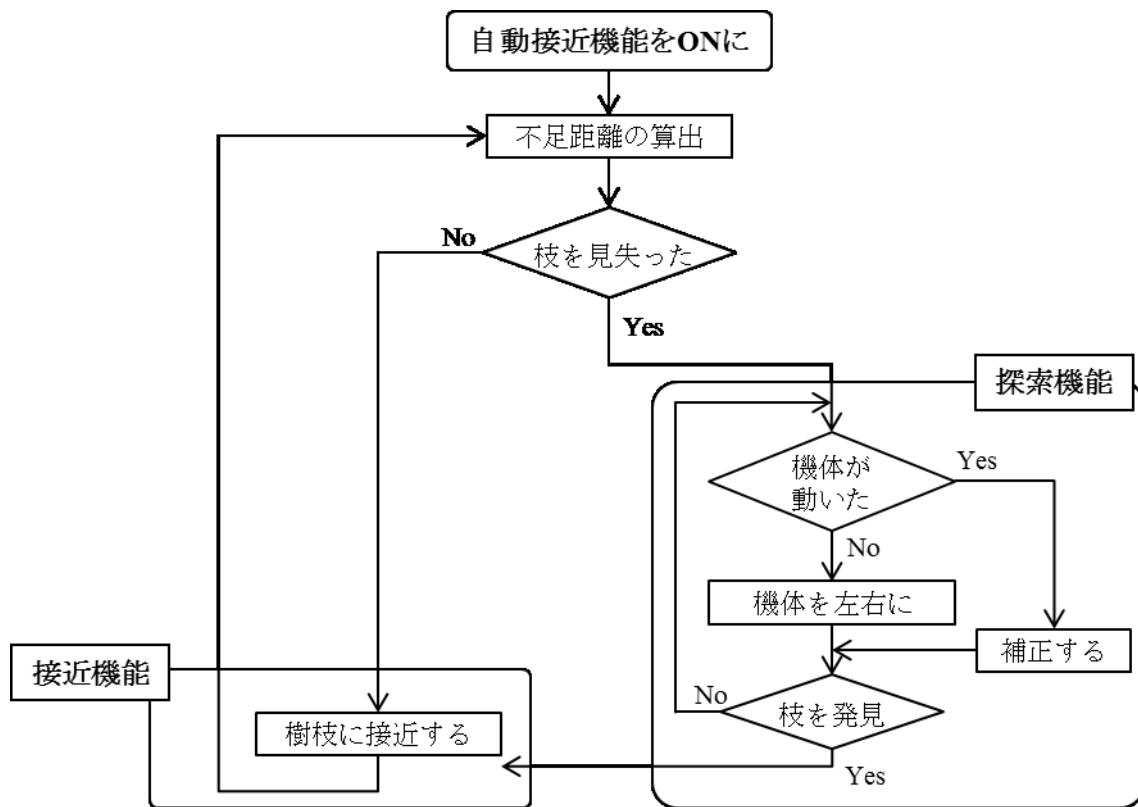


図 3-2 自動接近機能のフローチャート

次章で，本アルゴリズムを検証するための実験を行うための装置について述べる．

第4章 実験装置

4.1 Quad 型マルチロータ型 UAV

4.1.1 本体

プラットフォームとなる UAV は、小型化を考慮し、ブラシレスモータ 4 基を搭載した Quad 型とした。UAV の外観、および模式図を図 4-1、図 4-2 に示す。モータは樹枝への干渉を考慮し、下向きとしている。



図 4-1 Quad 型マルチロータ型 UAV の外観

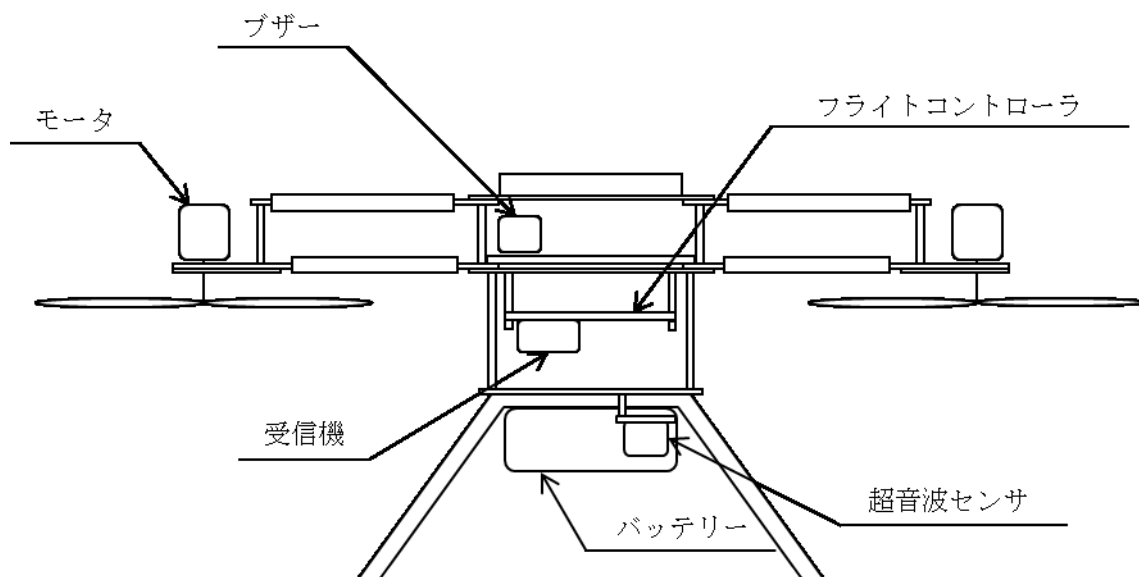


図 4-2 Quad 型マルチロータ型 UAV の模式図

表 4-1 に UAV の詳細を示す。なお、モータ間距離は対角線上、KV は 1V あたりのモータの無負荷回転数を表す。

表 4-1 供試マルチロータ型 UAV の詳細

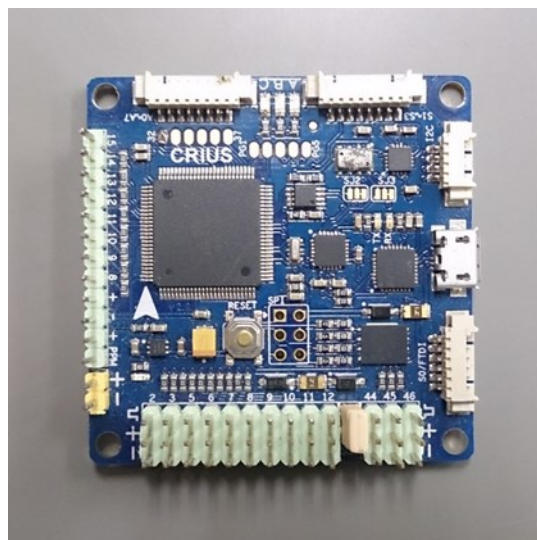
モータ間距離	230mm	
重量 (バッテリー除く)	415g	
推進用動力ユニット数	4 基 (Quad)	
プロペラ直径	127mm	
推進用モータ	$\Phi 22 \times 22\text{mm}$ ブラシレス DC モータ	
	KV	1800
バッテリー	11.1V 1300mAh リチウムポリマ	

4.1.2 MultiWii フライトコントローラ(Crius All In One Pro v2.1)

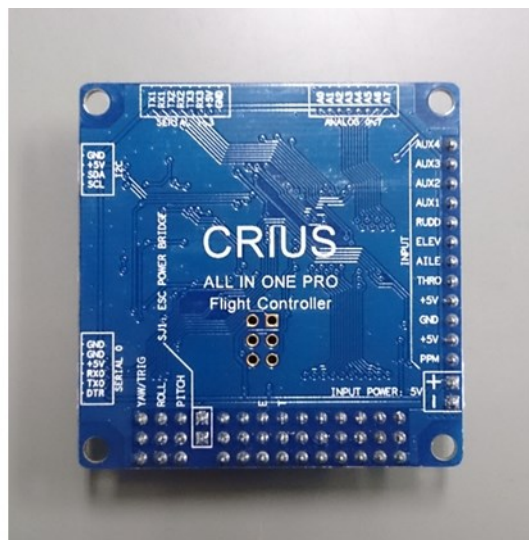
フライトコントローラは 3 軸ジャイロ, 3 軸加速度計, 地磁気センサ, 気圧センサを搭載した Open source で利用可能な MultiWii を使用した。

MultiWii は Open source での開発が進められているプロジェクトで, 主にマルチロータ型 UAV の RC モデルを制御するための汎用ソフトウェアであり, 開発言語に Arduino を用いる。

図 4-3 にフライトコントローラの外観を示す。ファームウェアのバージョンは MultiWii2.2⁴⁾である。フライトコントローラには I²C で他のセンサを接続することができる。また, 表 4-2 にフライトコントローラの主な仕様について示す。



(a)表面



(b)裏面

図 4-3 フライトコントローラの外観

表 4-2 フライトコントローラの主な仕様

寸法	50mm×50mm×11.6mm
重量	14.2g
CPU	ATmega2560-16AU
3 軸ジャイロ	MPU6050
3 軸加速度計	
地磁気センサ	HMC5883L
気圧センサ	MS5611-01BA03

4.2 センサユニット

センサユニットには，樹枝への装着機構が必要になるが，ここではマルチロータ型 UAV による樹枝への押圧力により，装着を行う機構とした．2 本の指を持つハンドの中央にある円盤が樹枝に当たり，中央のシャフトが押されることにより，2 本の指が閉じて樹枝を把持し，ロックがかかる構造となっている．なお，本ロックは RC 用サーボモータにより，無線で解除可能とした．また，実際のセンサからのデータ通信やロック解除は無線通信モジュール XBee による無線通信で行う．センサユニットの外観図，および模式図を図 4-4，図 4-5 に示す．また，センサユニットを UAV に搭載した様子を図 4-6 に示す．センサユニットの重量は 72g，装着後のハンドの内径は 53mm である．

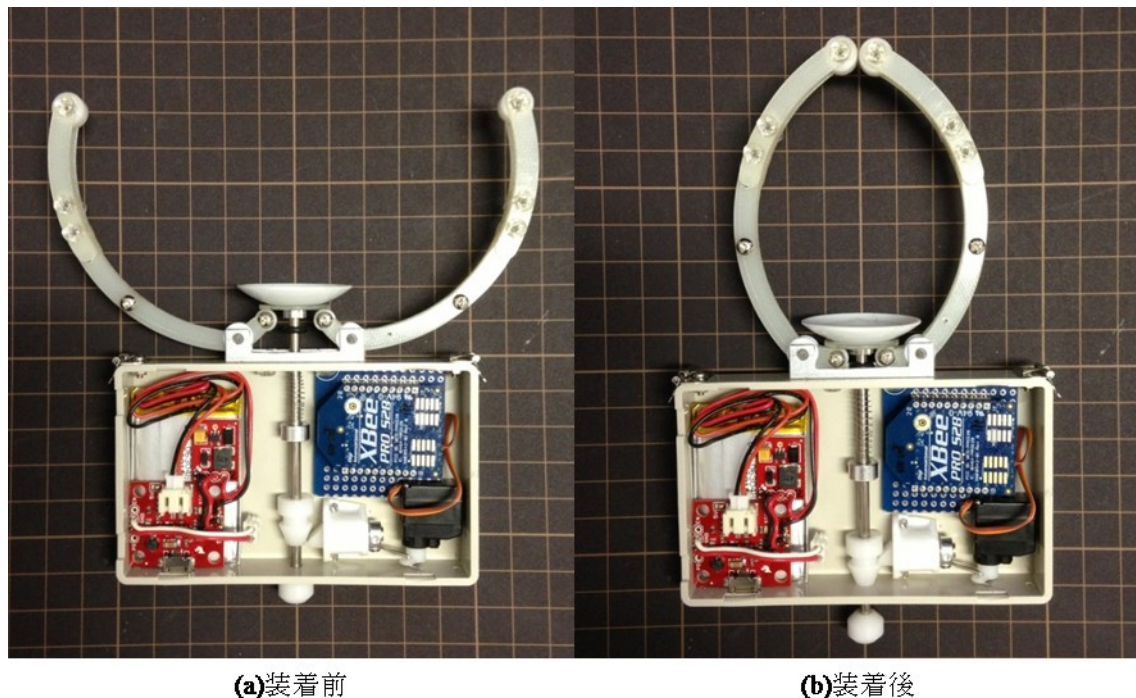


図 4-4 センサユニットの外観図

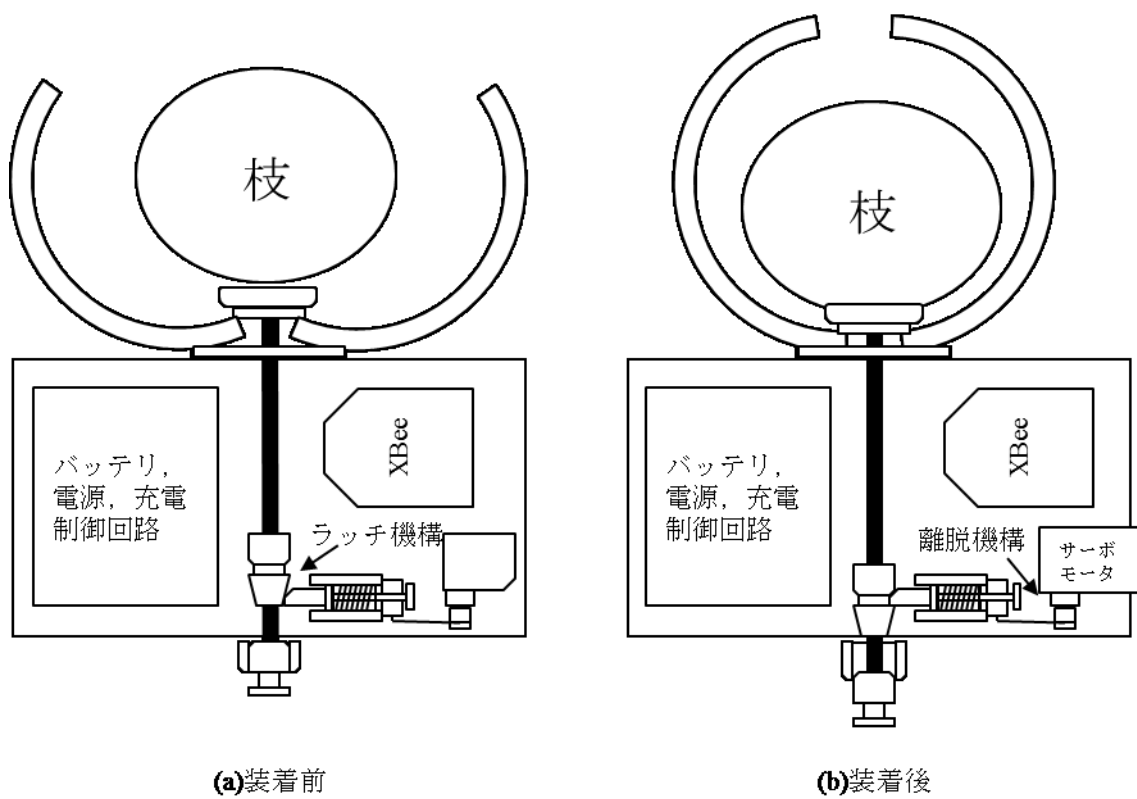


図 4-5 センサユニットの模式図

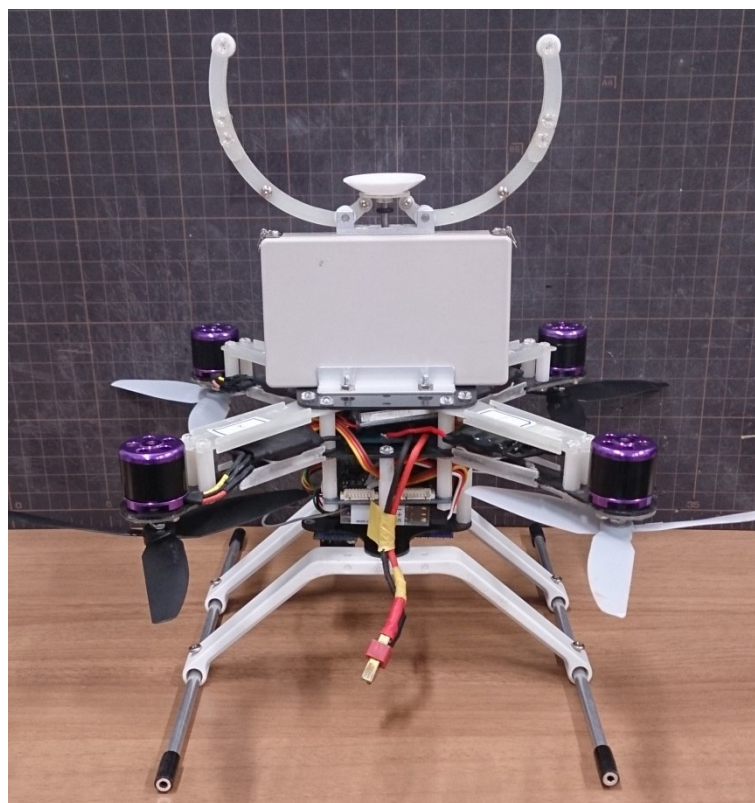


図 4-6 UAV にセンサユニットを搭載した様子

4.3 超音波センサ(DFRobot SRF-02)

樹枝の検出，および距離の測定に用いるためのセンサで，シングルセンサタイプの超音波測距ユニットでマイコンを搭載しており，オートチューン機能や自動ゲインコントロール機能を搭載している．I²C とシリアルインターフェースをもち，ひとつのバスに最大で 16 のセンサを接続することが可能となっている．外観を図 4-7 に示す．また，主な仕様について表 4-3 に示す．なお，フライトコントローラとは I²C で接続・通信を行う．

参考文献 5)より引用した，本センサのトランスデューサの感度分布，φ55 プラスチックパイプの最大検出範囲について，図 4-8 および図 4-9 に示す．



図 4-7 超音波センサ SRF-02 の外観

表 4-3 超音波センサ SRF02 の主な仕様

使用マイコン	16F687-I/ML
測定範囲	16cm～600cm
電源	5V（消費電流 4mA Typ.）
サイクルタイム	65ms
使用周波数	40kHz
アナログゲイン	64 段階の自動ゲインコントロール
接続モード	I ² C またはシリアルバス
全自動調整機能	電源投入後キャリブレーション不要
測距タイミング	エコー時間計測 ホストによるタスクコントロール
測定単位	μs, mm, cm
分解能	1cm
サイズ	24mm×20mm×17mm
入出力端子	5 ピン
重量	4.6g

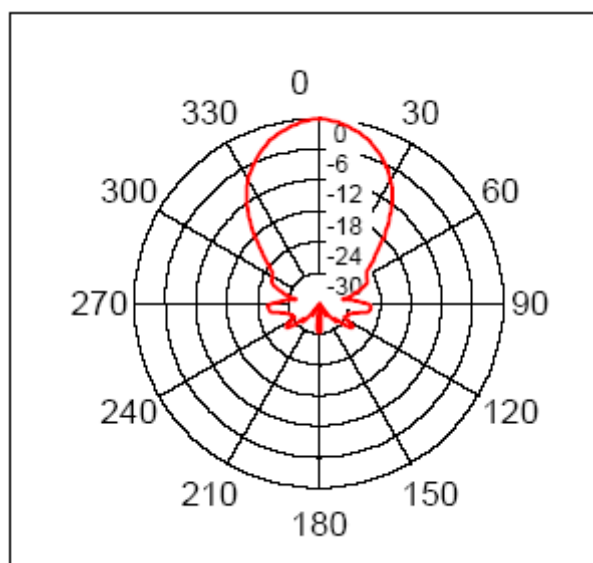


図 4-8 トランスデューサの感度分布（単位は db）⁽⁷⁾

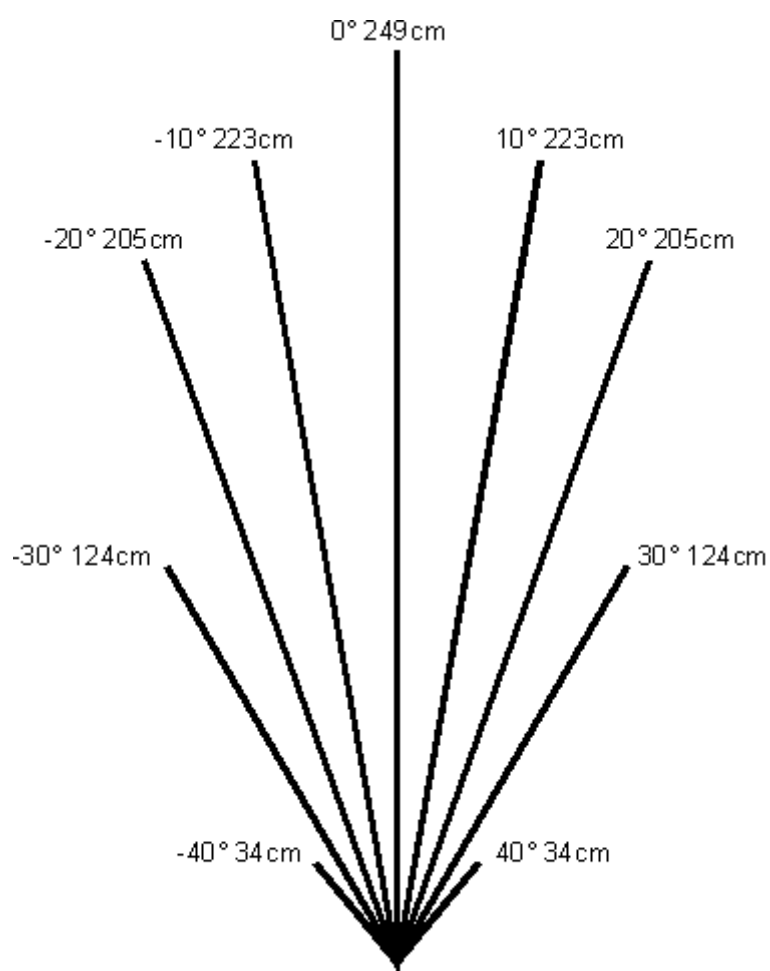


図 4-9 SRF-02 での φ55 プラスチックパイプの最大検出範囲⁽⁷⁾

4.4 φ56 中空丸棒

模擬樹枝として，超音波センサで検出するために用いる．紙製．

4.5 扇風機(Kashiba EF-303R)

外乱として風を発生させるために使用．表 4-4 に主な仕様を示す．

表 4-4 扇風機の主な仕様

電圧 (V)	100
周波数 (Hz)	60
消費電力 (W)	42
回転数 (rpm)	1150
風速 (m/min)	200
風量 (m ³ /min)	41

4.6 MultiWiiWinGUI⁸⁾

フライトコントローラに搭載された各センサから得られたデータの計測を行うために用いる．また，機体のパラメータ調整にも用いる．コンソール画面の例を図 4-10 に示す．

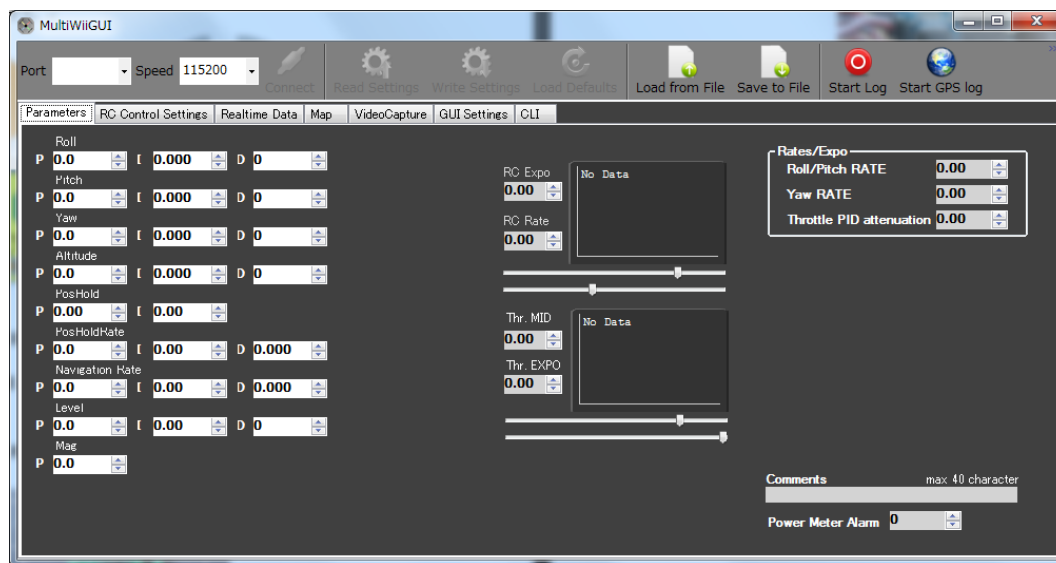


図 4-10(a) MultiWiiWinGUI のコンソール画面 (パラメータ調整画面)

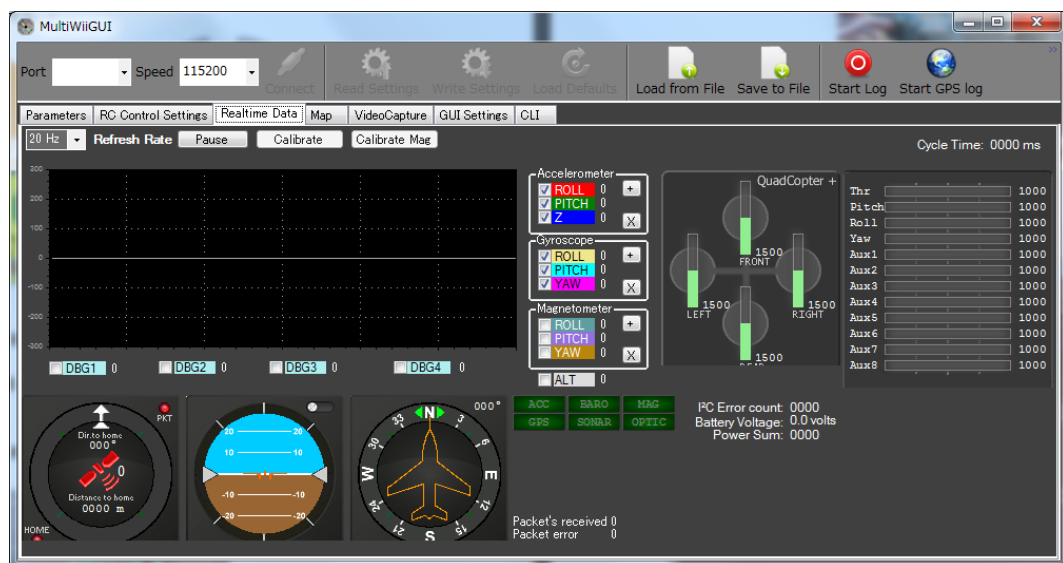


図 4-10(b) MultiWiiWinGUI のコンソール画面（出力確認画面）

第5章 実験方法、結果および考察

ここでは本研究で行った以下の3つの実験について述べる．それぞれ

- 維持機能
- 接近機能
- 探索機能

についての機能を実装し，その動作を検証・実験する．

また，以上の実験を行うための予備実験についてもここで述べる．すべての実験におけるデータの取得には MultiWiiWinGUI を用いた．

5.1 予備実験

5.1.1 概要および目的

実験を行うにあたって，樹枝の検出に使用する超音波センサの特性を知っておく必要がある．超音波センサは，ある程度の広さがある面をもつ物体との距離の測定に適したセンサである．しかしながら，物体の検出，特に物体の正確な位置を検出する事には適していない．それは，センサから出た音波はセンサから離れていくにつれて徐々に広がっていくため検出可能範囲が広くなり，検出対象がぼやけてしまう事による．このような超音波センサの特性から，使用時にはホーンを付けることが普通である．ホーンを付けることにより，超音波を一定方向に収束して発射，受波することができるため，指向角を狭くすることが期待できる．図 5-1 に今回用いたホーンの形状を示す．なお，ホーンはポリ塩化ビニル管を加工することで作製した．

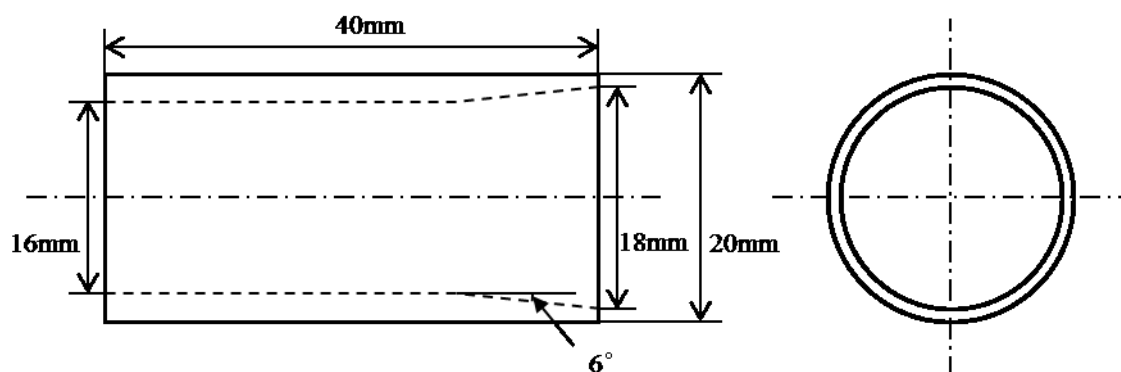


図 5-1 作製したホーンの形状

本研究では，樹枝の正確な位置を知ることが重要である．そのため，ここではまず予備実験として超音波センサの特性，特に指向角および測定可能距離について調べた結果を記す．また，ホーンの有無による，指向特性の差異についても比較し，ホーンの効果についても述べる．

5.1.2 実験方法

実験の様子を図 5-2 に示す．指向角については，超音波センサを固定し，検出したい物体

超音波の進行方向に対して垂直方向に動かし、検出の可否を確認することで行った。また、超音波センサから物体までの距離を 25cm, 30cm, 40cm, 50cm, 60cm, 70cm, 80cm, 90cm, 100cm と変更して同様の実験を行った。

測定可能距離については、固定された超音波センサの正面に物体を設置し、超音波センサからの距離を大きくしていくことで、測定可能距離の範囲を調べた。

また、外乱発生時の超音波センサの挙動を調べるため、設置した超音波センサの側面より風を発生させ、センサの出力値の変化を観察した。センサと検出物である中空丸棒との距離は 100cm, 風は扇風機により発生させ、風速は 3.3m/s である。

なお、それぞれ超音波センサは地上 45cm の高さに地面に対して水平方向に超音波を発射する向きに設置し、ホーンを装着したときについても調べた。すべての実験は風が起きない室内で行った。

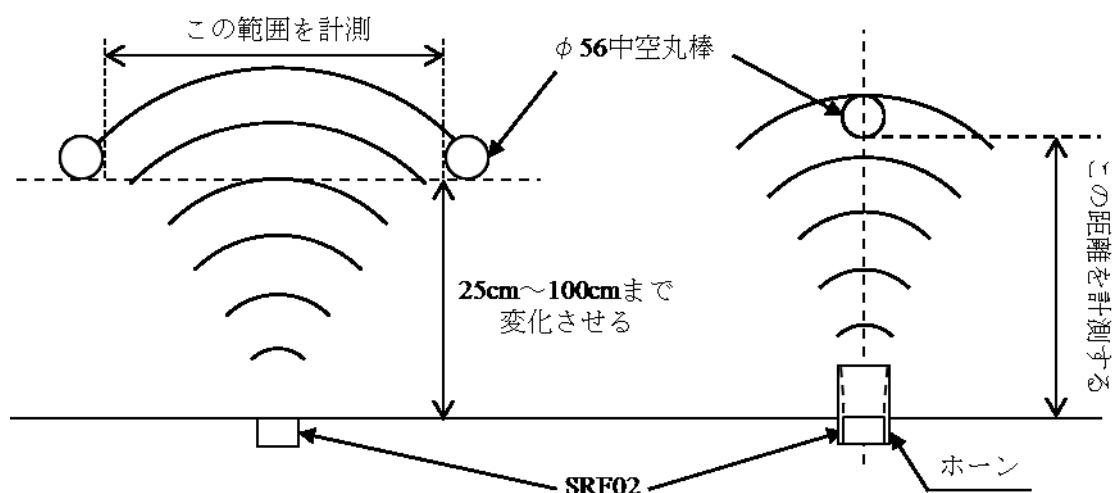


図 5-2(a) 超音波センサ SRF02 の指向角、測定可能距離を調べている様子

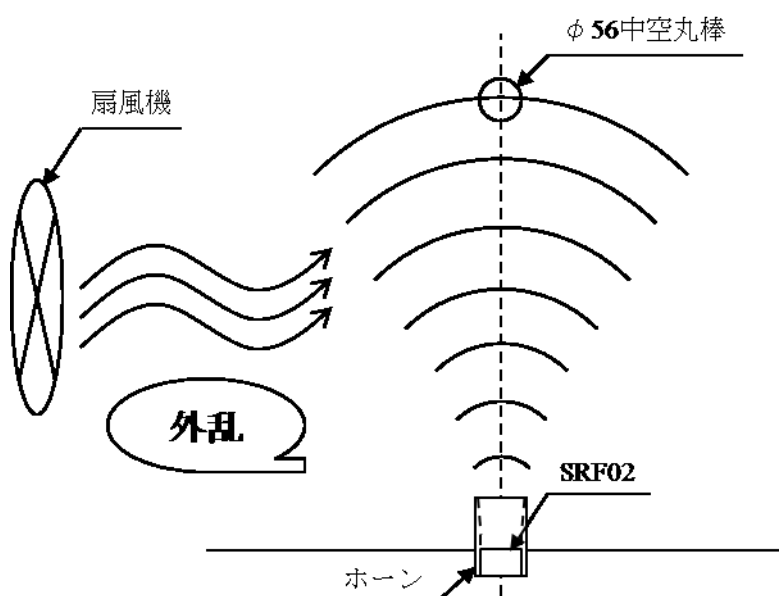


図 5-2(b) ホーン装着時の外乱に対する影響を調べている様子

5.1.3 実験結果

図 5-3 に指向角についての実験の結果を示す。図 5-3 の「安定」は物体を検出し、その後も定常的に検出し続ける状態を指し、「不安定」は物体を検出するも定常的ではなく、検出と未検出の間で揺れている状態を指す。なお、横軸の距離は超音波センサを中心として、右側を正の値、左側を負の値で表している。これを見ると、距離によって差はあるが、ホーンを装着していないときでは、物体との距離がセンサの正面から±10cm 程度ずれていても正面に存在するときと同様の結果を得ることがわかる。ホーンを装着したときでは、物体との距離がセンサの正面から最小で±10cm 程度、最大で±20cm 程度ずれていても正面に存在するときと同様の結果を得ることがわかる。

また、ホーンによる指向角を狭くする効果は近距離（物体との距離が 30cm 以下）において多少認められるが、それ以外の距離では認められず、むしろホーンを装着した方が指向角は広がっていることがわかる。

加えて、ホーンを装着していないときはホーンを装着している時と比べ、安定と不安定の差が大きいことがわかる。この傾向は特に比較的遠距離において認められる。

測定可能距離についての実験の結果を表 5-1 に示す。これを見ると、測定可能距離の範囲は 23cm～200cm であることがわかり、第 4 章で述べた仕様（16cm～600cm）との差異が認められた。

図 5-4 に外乱発生時の超音波センサの出力値について示す。これを見ると、ホーンが装着されていないときでは、出力値の誤差は最大 20cm と非常に乱れ、検出物との実際の距離を正確に取得し続けていない。一方、ホーンを装着した場合では、出力値に若干の乱れはあるが、最大でも 3cm 程度であり、外乱の影響をかなり軽減していることがわかる。

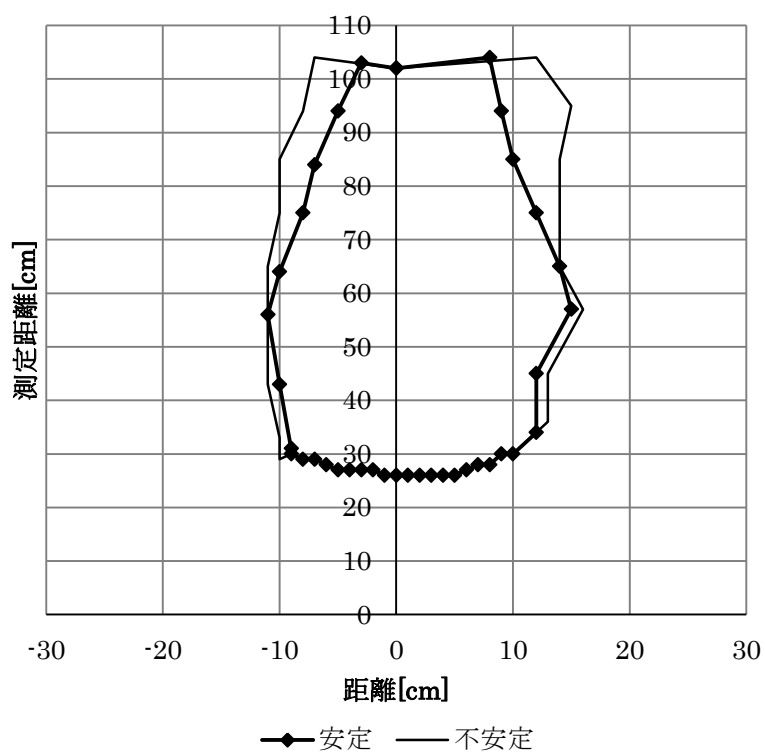


図 5-3(a) 超音波センサ SRF-02 の指向角 (ホーン無)

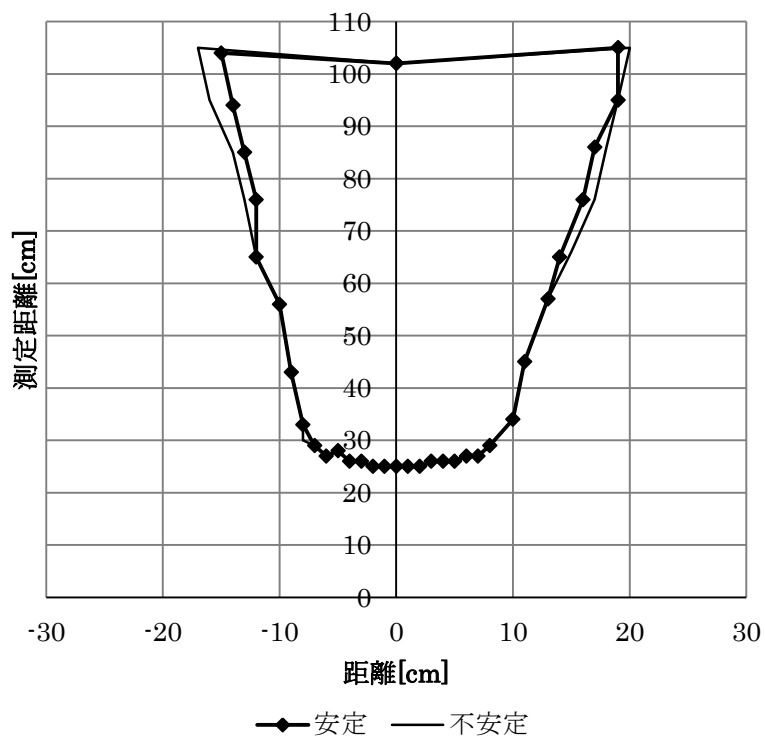


図 5-3(b) 超音波センサ SRF-02 の指向角 (ホーン有)

表 5-1 測定可能距離の範囲

範囲[cm]	挙動	備考
～200	実際とは異なる値を計測	200～210 程度の値で推移
200～23	計測に影響なし	
23～17	実際とは異なる値を計測	変化が緩やかになる
17～	実際とは異なる値を計測	200～210 程度の値で推移

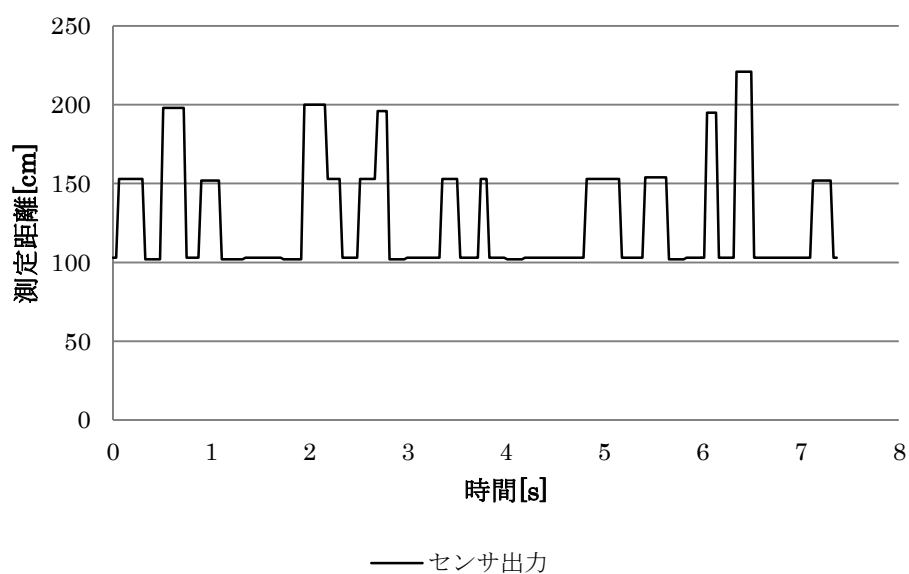


図 5-4(a) 外乱発生時の超音波センサの出力（ホーン無）

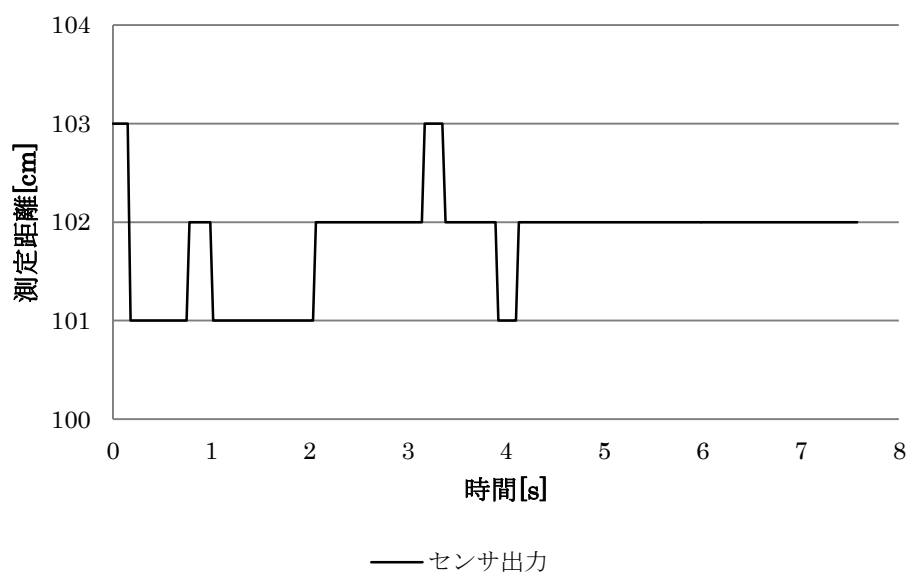


図 5-4(b) 外乱発生時の超音波センサの出力（ホーン有）

5.1.4 考察

以上の実験の結果から、本センサを用いて樹枝の検出を行う場合は、以下の条件を満たす必要があると考えられる。

- 超音波センサの検出誤差である 20cm に、プロペラを含んだ機体の幅 258mm の半分である 129mm を加えた $\pm 32.9\text{cm}$ の範囲に機体と干渉する物体が存在しない。
- 樹枝との距離が 23cm～200cm の間である。

特に、樹枝との距離に関しての制限から、本センサのみを用いて自動接近機能を達成することは困難であると考ええる。この課題を解決するためには、本超音波センサの最短測定可能距離よりも短い距離を検出可能な他のセンサを用いて、至近距離の樹枝の検出を行う必要があるだろう。つまり、操縦者が目視で樹枝と 200cm の距離まで機体を誘導し、本超音波センサを用いて 200cm から 23cm までの接近を行った後、23cm よりも短い距離の測定が可能な他のセンサを用いて、より樹枝へ接近するという流れを構築する必要があると考える。

ホーンを装着した結果、ホーンを装着していないときに比べて、安定と不安定の差が小さくなったことから、ホーンには超音波の受波感度を安定させる効果があると考えられる。また、外乱に対しての影響を軽減することから、ホーンを装着することは外乱の発生する屋外での使用時に有効だといえる。しかしながら、ホーンを装着することによって指向角を鋭くすることはできなかった。この点に関して、今後調べていく必要がある。

5.2 維持機能

5.2.1 概要および目的

自動接近機能とは、機体と目標とする樹枝との距離を徐々に減らしていき、目標とする距離に至ったとき、その距離を保つ機能である。そのためには、目標値と現状の値に差がある段階では、偏差を小さくするための操作量を与え、目標値に達した後ではその距離を維持し続けるための操作量を与える必要がある。これらはどちらも目標値との偏差に応じた操作量を与えるという操作で、偏差の大きさによって操作量に差は出るが、本質的に同じことである。そのため、自動接近機能を実現するための前段階として、まずは目標との距離を維持し続ける「維持機能」を実現させる。

維持機能とは、機能が ON になったときの機体と目標との距離を維持する機能である。目標との距離の検出は超音波センサを使って行い、機体の制御方法は P 制御を用いた。P 制御は偏差に比例した操作量を加える制御方法であり、ある時刻 t での出力値を $y(t)$ 、目標値を $r(t)$ としたとき、加えられる操作量 $u(t)$ は式(1)のようになる。

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau \quad (1)$$

ただし、 $e(t) = y(t) - r(t)$ 、 K_p : 比例ゲイン、 K_i : 積分ゲイン

このとき、 $y(t)$ は機体の現在の目標との距離、 $r(t)$ は機能を ON にしたときの維持すべき距離であり、それぞれ超音波センサを用いて取得した。

本実験では UAV に PI 制御を用いた維持機能を実装し、維持機能を達成するために適切な K_p 、 K_i の値を決定することを目的とした。

本来の目標である樹枝は機体よりも上部に存在するため、超音波センサを上に向けて搭載する必要があるが、本実験では目標とする物体を地面に設定し、下に向けて搭載することとした。

5.2.2 実験方法

実験の様子を図 5-5 に示す。維持機能を搭載したマルチロータ型 UAV を離陸させ、ホバリングさせたのち維持機能を ON にする。その後、機体の下に高さ 25cm のブロックを挿入し、機体がブロックで底上げされた 25cm 分上昇することを確認した。なお、この実験は機体の墜落の危険を考慮して、天井から機体を地面と激突しない高さに吊るして行った。機体を吊るす装置は自動で紐を巻き取るため、機体の上昇に伴う紐のたわみがプロペラ等の駆動部に干渉することはない。なお、制御則は維持機能と同様に PI 制御で行うが、まずは P 制御を適用し、比例ゲインを決めた上で積分動作を加えていくこととした。

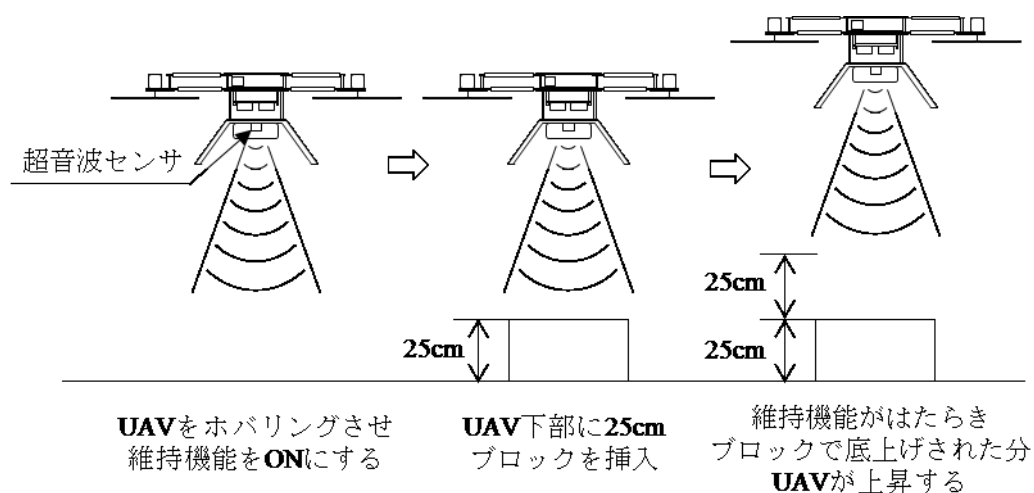


図 5-5 維持機能実験の様子

5.2.3 実験結果

実験の結果を表 5-2, 図 5-6 に示す. 表 5-2 は本実験で得られた各条件における最大偏差である. P 制御ではブロック挿入時, 除外時を, PI 制御では除外時のものを示している. PI 制御時の挿入時に関しては, ブロックを機体下部にとどめておくことが困難であったため, 記録できなかった. これをみると, 最大偏差はブロック挿入時の方が除外時よりも小さい結果となっている.

図 5-6 は, それぞれ異なる比例ゲインおよび積分ゲインによって, 超音波センサで測定された距離の推移である. 縦軸, 横軸はそれぞれ距離と時間を示している. これをみると, 測定距離は 20cm 程度から開始していることがわかるが, これは天井から吊り下げていることで機体自体が地面から浮いているためである. その後, 測定距離が上昇し, ある程度の高さで落ち着いているのは, 機体上昇し, ホバリングを行っているためである.

図 5-6(a), 図 5-6(b), 図 5-6(c-1)は P 制御による結果であり, それぞれ $K_p = -5, -4, -3$ のときの結果を示している. これをみると, $K_p = -5$ のときに最も偏差が小さくなり, 維持すべき距離を追従していることがわかる. しかし, 制御のかかり方が急で, 場合によっては機体が大きく暴れてしまうことがあった. この傾向は $K_p = -4$ のときにもみられ, 操作量が過大だったため発生した現象であるといえる. そのため, $K_p = -3$ に積分動作を追加した.

図 5-6(c-2), 図 5-6(c-3), 図 5-6(c-4)は $K_p = -3$ に積分動作を追加した結果である. これをみると, 時間に差はあるがどの場合も維持距離を追従していることがわかる.

表 5-2 維持機能実験の結果（各条件での最大偏差）

	最大偏差[cm]	
	挿入時	除外時
$K_p = -5$	3	6
$K_p = -4$	18	21
$K_p = -3$	14	28
$K_p = -3, K_i = -0.5$	—	3
$K_p = -3, K_i = -1$	—	4
$K_p = -3, K_i = -1.5$	—	3

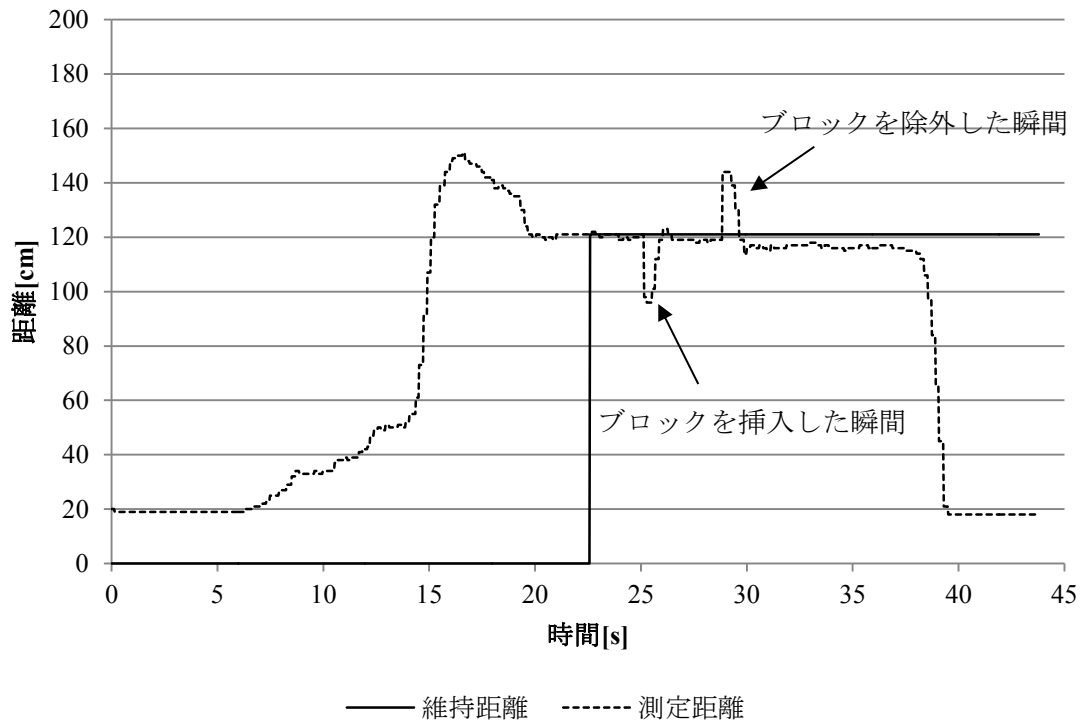


図 5-6(a) 維持機能の実験結果（ $K_p = -5$ のときの距離の推移）

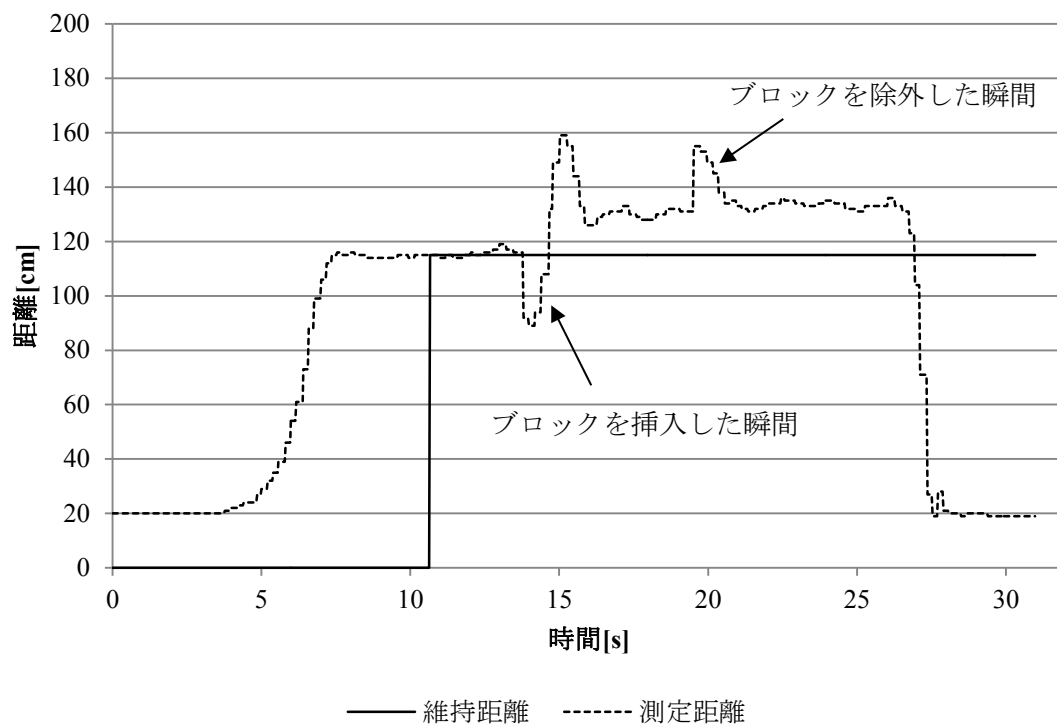


図 5-6(b) 維持機能実験の結果 ($K_p = -4$ のときの距離の推移)

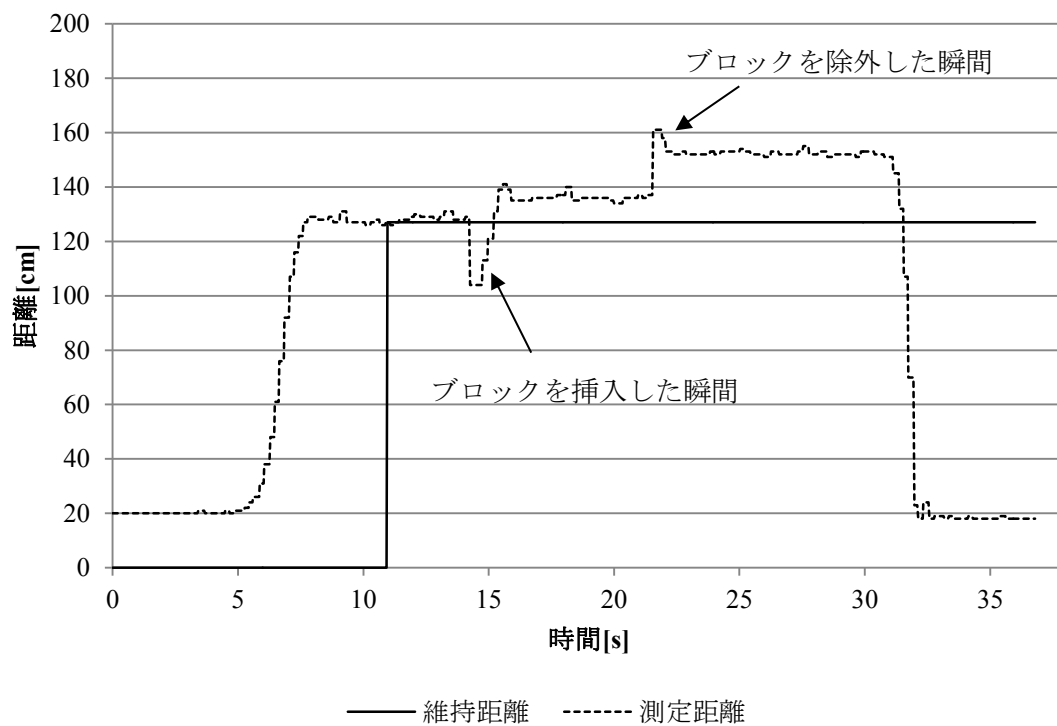


図 5-6(c-1) 維持機能実験の結果 ($K_p = -3$ のときの距離の推移)

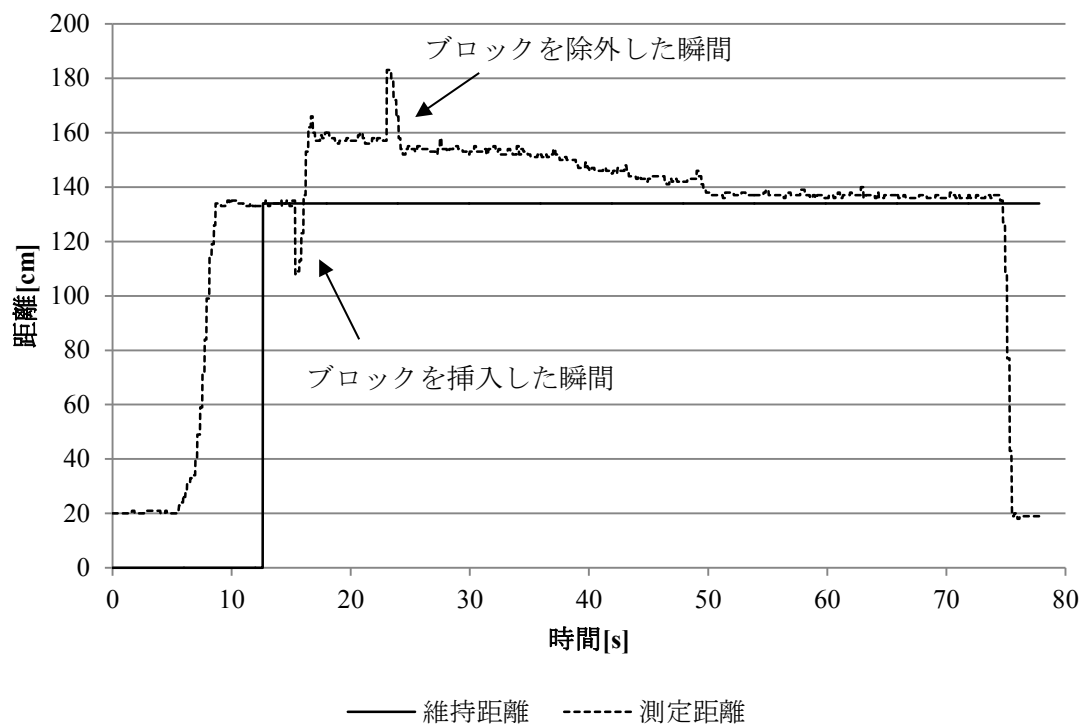


図 5-6(c-2) 維持機能実験の結果 ($K_p = -3, K_i = -0.5$ のときの距離の推移)

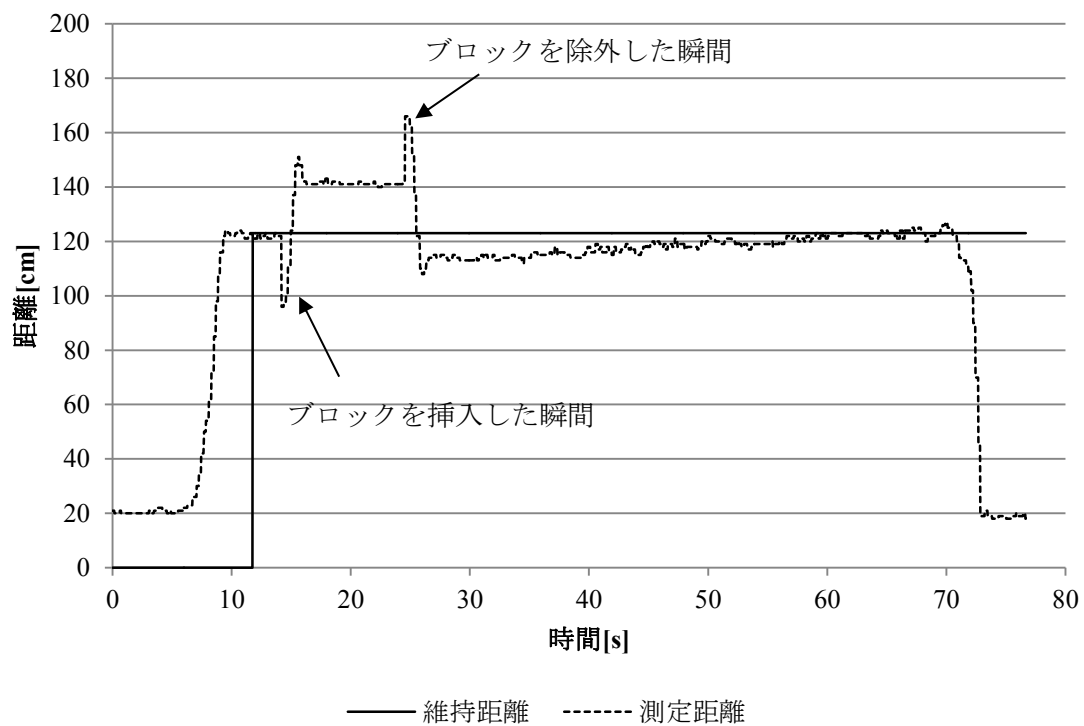


図 5-6(c-3) 維持機能実験の結果 ($K_p = -3, K_i = -1$ のときの距離の推移)

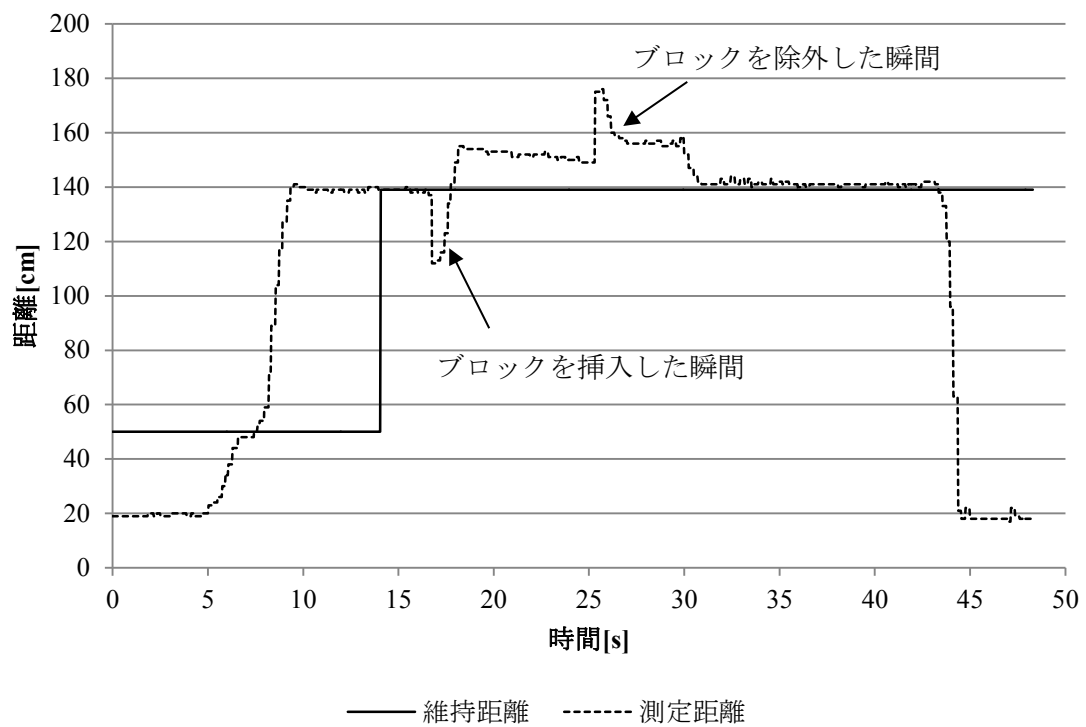


図 5-6(c-4) 維持機能実験の結果 ($K_p = -3, K_i = -1.5$ のときの距離の推移)

5.2.4 考察

実験の結果、PI 制御を用いることによって、UAV と目標との距離を維持する機能を実装することができた。P 制御のみでは残留偏差の関係から維持距離に一致することはなかったが、今回 P 制御において最も維持距離への収束が速かった $K_p = -5$ のときに積分動作および機体の暴れを防止、軽減する減速項を追加することで、より応答性の高い機能とすることができると考えられる。

ブロック挿入時よりも除外時の方の偏差が大きくなっていることから機体は上方向に引っ張られていることがわかる。このことから、実験環境の影響を多分に受けていることがわかるため、実験環境の改善が必要と考えられる。

この維持機能を基本にして、自動接近機能を構成する接近機能の実装を行うこととする。

5.3 接近機能

5.3.1 概要および目的

接近機能とは第 3 章で述べたように、自動接近機能のなかで実際に接近運動を行う部分であり、目標とする樹枝まで自動で接近を行う機能である。ここでは、先の維持機能を基本に機能の実現を目指す。

接近機能を実現するためには、維持機能と同様に、PI 制御により目標値との偏差に応じた操作量を与えることで行った。維持機能での目標値は、機能が ON になった時の機体と目標との距離であったが、接近機能では目標とする樹枝に接近していくことから、目標値は樹枝の位置そのものであり、樹枝との距離を 0 にすることである。しかしながら、本実験で樹枝に検出に用いる超音波センサの測定可能距離は、予備実験の結果 23cm～200cm であり、フィードバックをかけながら目標との距離を 0 にすることは不可能であることは 5.1.4 で先述したとおりである。そのため、本実験では目標値を、超音波センサの測定可能距離である目標と 50cm の距離に設定して行った。

5.3.2 実験方法

実験の様子を図 5-7 に示す。接近機能を搭載したマルチロータ型 UAV を離陸させ、ホバリングさせたのち接近機能を ON にする。その後、機体が目標とする地面と 50cm の高さまで接近し、目標値に達したあとはその距離を維持することを確認する。また、そのとき機体に加えられる操作量についても記録・確認した。

なお、制御則は維持機能と同様に PI 制御で行うが、まずは P 制御を適用し、比例ゲインを決めた上で積分動作を加えていくこととした。

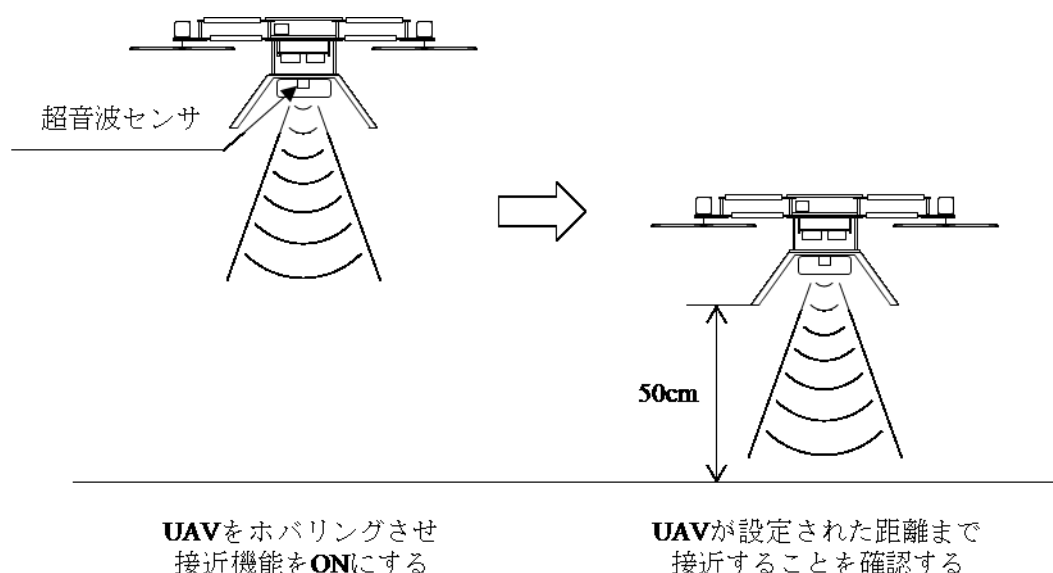


図 5-7 接近機能実験の様子

5.3.3 実験結果

実験を行った結果を図 5-8 に示す。図 5-8 はそれぞれ、P 制御、PI 制御を用いて接近実験を行ったときの測定距離の変化である。縦軸、横軸はそれぞれ測定距離と時間を表している。

測定距離が 20cm 程度の距離から開始しているのは、維持機能実験と同様に UAV を天井から吊り下げたことから、UAV が浮いているためである。その後、UAV は測定距離を上昇させ、約 150cm の高さに到達し、安定している。これは、UAV が上昇し、ホバリングしていることを示している。ホバリング後、測定距離が変動しているが、これは接近機能を開始したことを示している。

図 5-8(a)より、今回適用したどの比例ゲインにおいても、目標距離である 50cm に到達しなかった。特に、比例ゲイン $K_p = -3$ のとき、測定距離が激しく変動しているため、操作量が過大であることがわかった。 $K_p = -2, -1$ のときにはこの現象が起きていないため、 $K_p = -2, -1$ に積分動作を加えることとした。

$K_p = -2, -1$ に積分動作を加えた結果が図 5-8(b)である。これをみると、 $K_i = -0.4$ のとき、UAV は目標距離である 50cm に到達した。一方、 $K_i = -0.1$ のときは到達しなかった。また、 $K_p = -2$ のときでは接近機能を ON にしたとき急峻な変化が起こったが、 $K_p = -1$ のときにはそれがみられなかった。

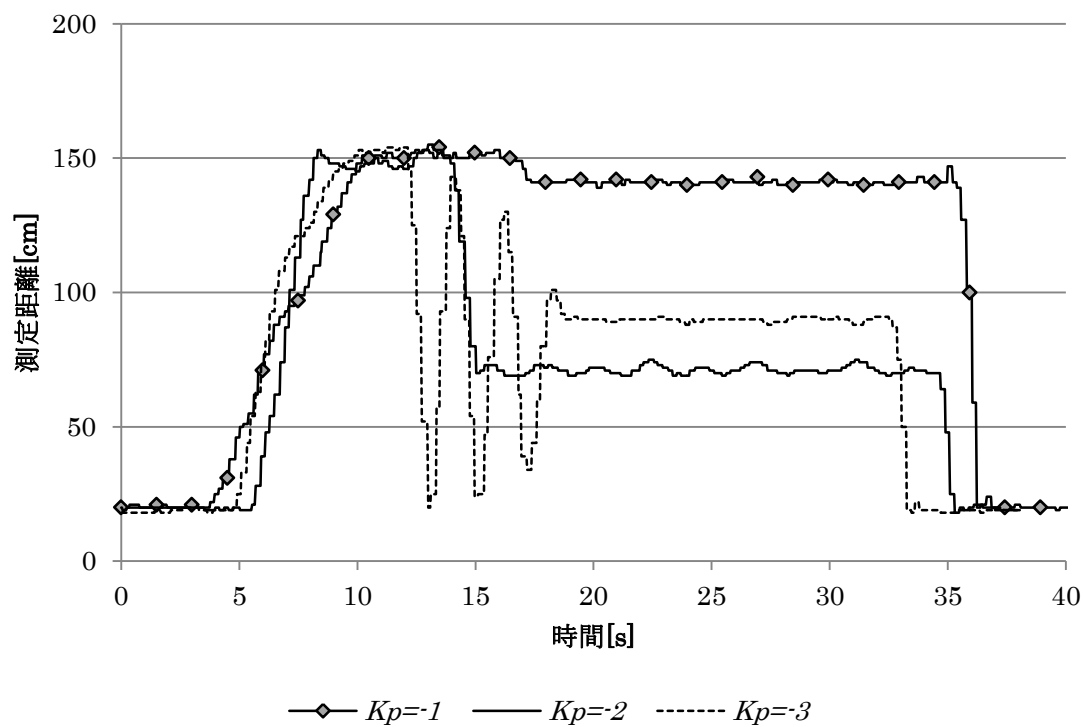


図 5-8(a) 接近機能実験の結果 (P 制御)

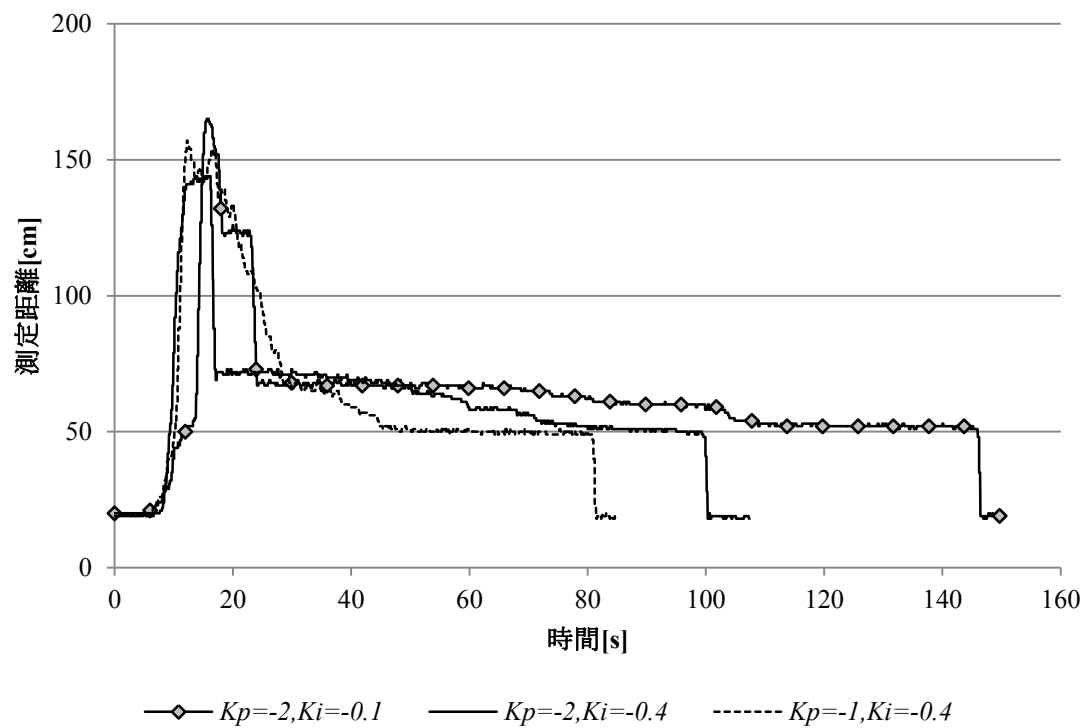


図 5-8(b) 接近機能実験の結果 (PI 制御)

5.3.4 考察

実験の結果、PI 制御を用いることによって、UAV を目標値に接近させる機能を実装することができた。 $K_i = -0.4$ のとき、目標値 50cm に到達することができたが、 $K_p = -2$ のとき急峻な変化が起こったことから、実際の運用状況である狭小な環境に鑑みると、機体と樹枝との接触により墜落してしまうことが予想される。そのため、急峻な変化を起こさない制御則を構築・適用する必要がある。

また、本実験は、維持機能実験と同様に、機体を吊り下げて行っているものである。そのため、常に上向きの外力を受け、UAV の実際の飛行状況とは異なっている。今回、目標値に到達できなかった条件が実際の飛行状況でも、同様の結果を示すとは考えにくい。

今後は、急峻な変化を起こさない安全性を考慮した制御則の考案および機体への実装、実際に機体を飛行させての接近実験を行い、本機能を改良していく必要がある。

5.4 探索機能

5.4.1 概要および目的

探索機能とは第 3 章で述べたように、自動接近機能のなかで接近運動を行わず、見失った樹枝の再発見を行う部分であり、そのための探索運動および再発見を判断する機能である。

本機能は機体が見失ったときから機能を開始し、樹枝を再発見するまで継続する。そのため、まずは樹枝を見失った状態を「樹枝との距離が、瞬間的に 20cm 以上変化したとき」と定義した。これは、超音波センサがそれまで検出していた目標を見失うと、目標よりも奥にある物体を検出することになり、得られる値が瞬間的に変化するから決定した。このように定義した関係上、目標とする物体の前後 20cm 以内に他の物体が存在しないという条件のもとで実験を行った。

樹枝を見失ったのち、機体は探索運動を開始するが、その運動は第 3 章で先述したように大きく 2 つに分けることができる。機体の横方向の運動によって生じた変位を補正する運動（以下、探索運動 1）と、機体の上昇に伴って見失った樹枝を発見する運動（以下、探索運動 2）である。これらの状況によって機体に与える命令は異なる。探索運動 1 では、樹枝を見失ったときの機体の運動方向を検出し、その運動と逆向きの運動をさせる必要がある。なお、運動の検出は機体に搭載された加速度センサを用いて行う。また、探索運動 2 では、見失った点を中心にして、左右に運動させる必要がある。

本実験では、探索機能を搭載したマルチロータ型 UAV を用いて、探索運動（探索運動 1、および探索運動 2）がそれぞれ正常に働いていることを確認した。また、本実験では探索運動によって機体に生じる加速度は考慮しなかった。

5.4.2 実験方法

(1)探索運動 1

実験の様子を図 5-9(a)に示す。探索機能を搭載したマルチロータ型 UAV を、地上 70cm の高さに設置し、機体と地面の間に模擬樹枝を設置することで、超音波センサに模擬樹枝を検出させた。その状態で探索機能を ON にする。その後、機体の横方向の運動によって樹枝を見失った状態を再現するため、機体を横方向に移動させ、横方向の加速度の変化を伴って模擬樹枝との距離を瞬間的に 20cm 以上変化させた。そのとき生じた加速度によって探索運動 1 が開始されることを確認した。また、樹枝を再発見したときに探索運動を終了することも確認した。

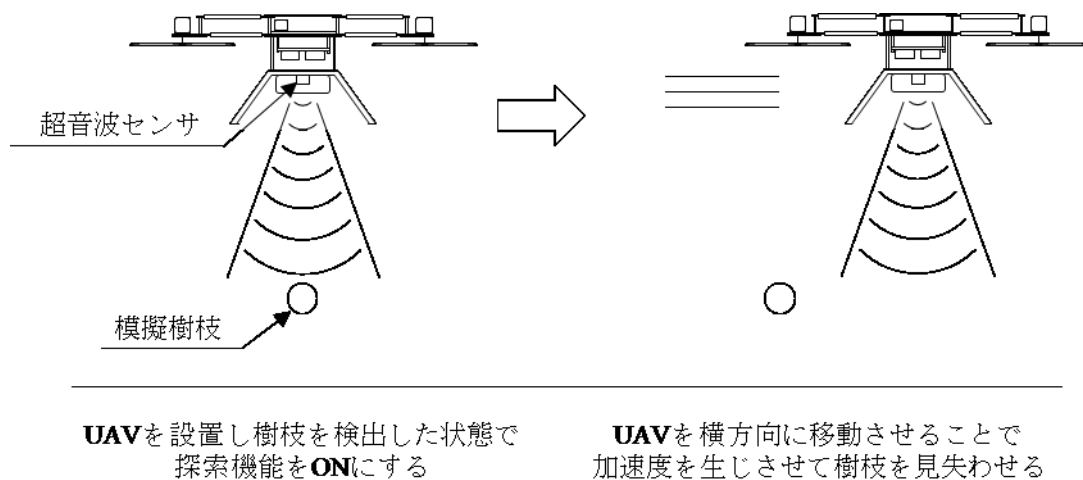


図 5-9(a) 探索機能実験の様子（探索運動 1）

(2)探索運動 2

実験の様子を図 5-9(b)に示す. 探索機能を搭載したマルチロータ型 UAV を, 地上 70cm の高さに設置し, 機体と地面の間に模擬樹枝を設置することで, 超音波センサに模擬樹枝を検出させた. その状態で探索機能を ON にする. その後, 機体の上昇に伴って樹枝を見失った状態を再現するため, 模擬樹枝を取り除き, 横方向の加速度の変化なしに模擬樹枝との距離を瞬間的に 20cm 以上変化させた. そのとき, 探索運動 2 が開始されることを確認した. また, 樹枝を再発見したときに探索運動を終了することも確認した.

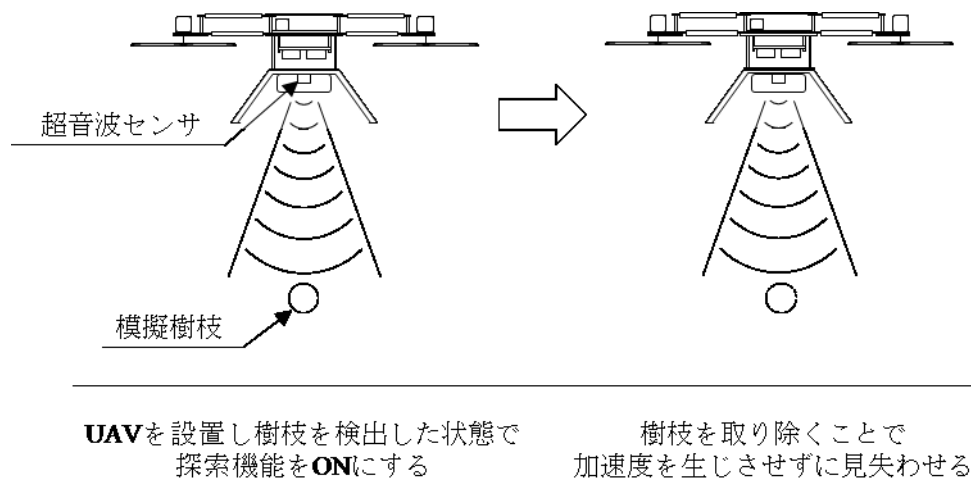


図 5-9(b) 探索機能実験の様子（探索運動 2）

5.4.3 実験結果

実験の結果を図 5-10 に示す. グラフ中の「Lost」「AccX」「rcCommand[ROLL]」はそれぞれ, 「樹枝の検出状態」「機体の左右方向に生じる加速度」「ロール軸方向の操作量」を表している. なお, Lost は第 1 軸, AccX, rcCommand[ROLL]は第 2 軸に対応しており, 横軸は

時間を表している。

また，Lost は式(2)のように定義している。

$$\text{Lost} = \begin{cases} 0, & \text{樹枝を検出} \\ 1, & \text{樹枝を見失う} \end{cases} \quad (2)$$

(1)探索運動 1

図 5-10(a)を見ると，はじめは Lost が 0 であることから，模擬樹枝を検出している状態であることがわかる．その後，AccX が大きく変動しはじめ，Lost が 1 となっている．つまり，機体の横移動によって模擬樹枝を見失った状態である．このとき，rcCommand[ROLL]は操作量を与えている．また，この操作量は AccX の変化がある程度の大きさでなければ与えられないこともわかる．加えて，AccX の正負によって与えられる操作量の正負も異なっている．これらのことから，機体の横移動によって樹枝を見失ったとき，横移動の方向に応じた操作量が与えられていることがわかる．その後，Lost は再び 0 になっているため，樹枝を再発見したことがわかる．Lost が 0 になった後，AccX の値の変化によって rcCommand[ROLL]は変動していないことから，樹枝を再発見したときに探索運動を終了していることがわかる．

(2)探索運動 2

図 5-10(b)を見ると，はじめは Lost が 0 であることから，模擬樹枝を検出している状態であることがわかる．その後，Lost が 1 となっているが，AccX に顕著な変化は見られない．つまり，機体の上昇に伴って樹枝を見失った状態である．このとき，rcCommand[ROLL]は操作量を与えている．また，この操作量は正負を変えて同じ大きさに交互に与えられている．このことから，機体を見失った点を中心として，左右に同じ距離だけ運動させるための操作量が与えられていることがわかる．その後，Lost は再び 0 になっているため，樹枝を再発見したことがわかる．そのとき，rcCommand[ROLL]は変動していないことから，樹枝を再発見したときに探索運動を終了していることがわかる．

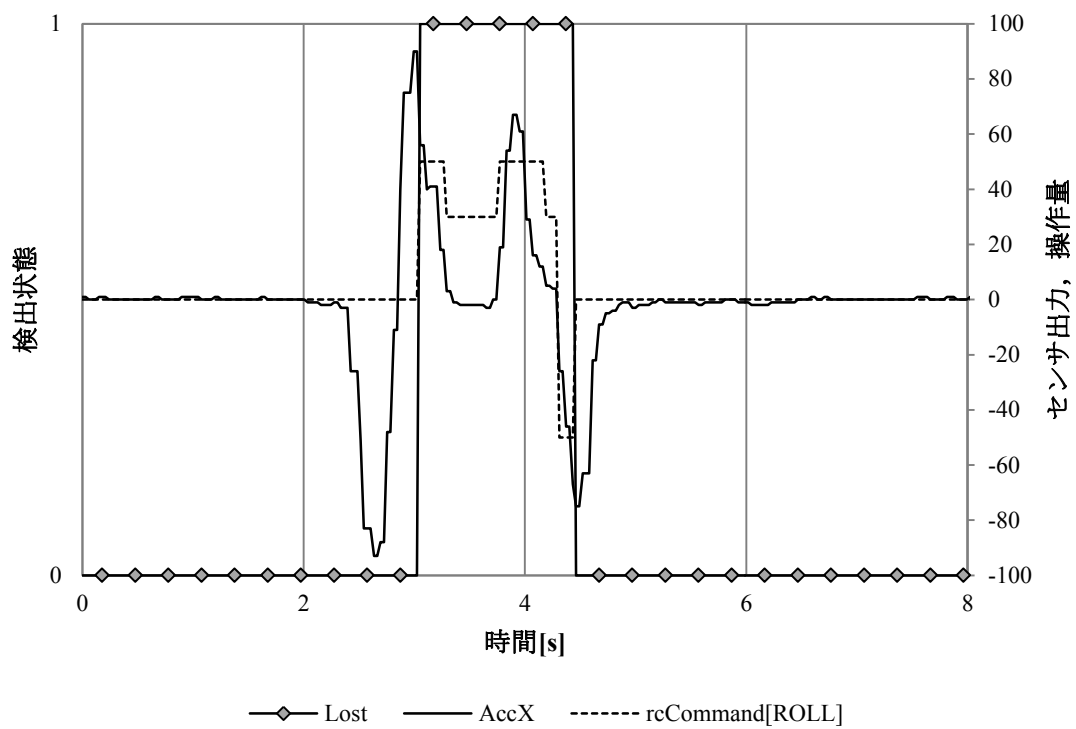


図 5-10(a) 探索機能の実験結果（探索運動 1）

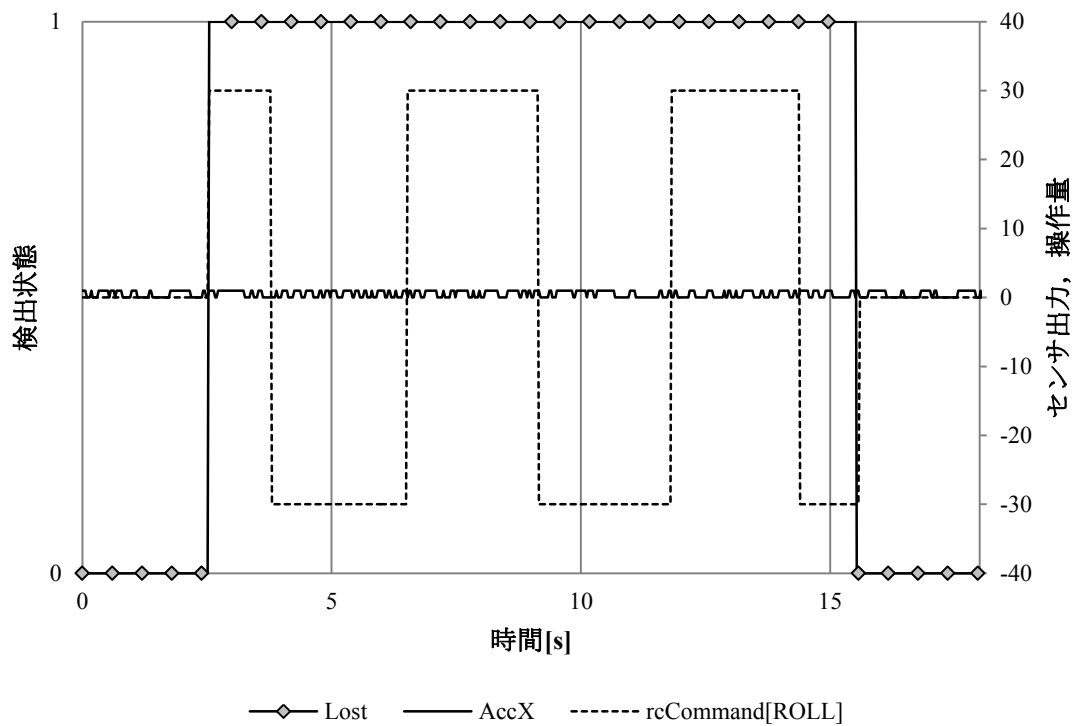


図 5-10(b) 探索機能の実験結果（探索運動 2）

5.4.4 考察

以上の実験の結果から、探索機能は正常に働いていることがわかる。しかしながら、本実験は実際の UAV の飛行状況とは異なり、探索機能が働くための理想的な状況を設定したものである。実際の飛行では、今回考慮しなかった探索運動によって生じる機体の加速度の影響はもちろん、外部からの影響のほか、UAV 自身が発生させる様々なノイズがセンサに影響を与える。そのため、今回開発した探索機能はあくまで基本的なものであることを踏まえ、今後は実際の飛行状況をより良く反映した制御則を考案する必要がある。また、超音波センサの数を増やすなどして樹枝を見失った方向を特定できれば、探索運動に無駄がなくなり、UAV の実際の運用状況である狭小な環境により適応したものになることができる。と考える。

今回探索運動を行うために機体に与えた操作量は、UAV を飛行させた際に得た経験から決定したものであるが、機体が実際の飛行状況で十分な探索運動を行うかどうかは UAV を飛行させたうえで実験を行い、操作量を決定していく必要がある。

第6章 結言

本論文では、マルチロータ型小型 UAV を用いた観測装置の移送および設置についての基本アルゴリズムとその技術的課題を示した。技術的課題の中でも特に UAV の誘導技術について、UAV に実装する「自動接近機能」およびその要素機能である「維持機能」、「接近機能」、「探索機能」について述べ、開発し、UAV に実装して実験および考察を行った。また、樹枝の検出に用いる超音波センサの特性も併せて調べた。

今後の課題として、超音波センサで測定不可能な至近距離で用いるセンサの考案、各比例定数の変更をはじめとした制御量の調整、実際の飛行状況に即した制御則の再考、より UAV の実際の運用状況に近い実験方法の考案および実験環境の構築などを行う必要があると考えられる。また、以上の課題を解決するとともに、各要素機能を組み合わせた自動接近機能を開発・実装し、観測装置の樹枝への移送・設置への有効性を明らかにする必要があると考える。

謝辞

本研究を進めるにあたり，終始助言とご指導を賜りました王秀崙教授ならびに鬼頭孝治准教授には，感謝の意を表し，御礼申し上げます。

エネルギー利用工学研究室の皆様には本研究を進めるにあたりご協力していただいたことを厚くお礼申し上げます。

最後に本論文作成にあたり，多大なるご協力をいただきました王秀崙教授，鬼頭孝治准教授に重ねて厚くお礼申し上げます。

平成 27 年 3 月

三重大学大学院
生物資源学研究科
共生環境学専攻
エネルギー利用工学教育研究分野
中嶋 雅佳

参考文献

- [1] 電波実験社 : 『ラジコン技術』 2012 年 4 月号
- [2] CQ 出版社 : 『Interface』 2014 年 4 月号
- [3] オーム社 : 『ロボコンマガジン』 2014 年 11 月号
- [4] MultiWii
<http://www.multiwii.com/>
- [5] MultiWii-software download
<http://code.google.com/p/multiwii/>
- [6] SRF02 Ultrasonic sensor (SKU:SEN0005)-Robot Wiki
[http://www.dfrobot.com/wiki/index.php?title=SRF02_Ultrasonic_sensor_\(SKU:SEN0005\)](http://www.dfrobot.com/wiki/index.php?title=SRF02_Ultrasonic_sensor_(SKU:SEN0005))
- [7] SRF02 Ultrasonic sensor user manual
<http://robot-electronics.co.uk/htm/srf02tech.htm>
- [8] mw-wingui download
<http://code.google.com/p/mw-wingui/downloads/list>

付録

ソースコード

以下に本研究で開発した自動接近機能のソースコードを示す．なお，ここに示すのは実際の機能部分であり，UAV の飛行およびその他の制御に関する記述は参考文献 4)から得られるため省略する．

A.維持機能

```
#if !defined(AUTO_APPROACHING)
    AltErr[0] = AltErr[1];
    AltErr[1] = sonarAlt - AltHold;
    ErrIntg += (AltErr[0]+AltErr[1])/2 * INT_TIME;
    if (abs(rcCommand[THROTTLE]-initialThrottleHold) >
    ALT_HOLD_THROTTLE_NEUTRAL_ZONE) { //スロットルが変化したら
        isAltHoldChanged = 1;
        rcCommand[THROTTLE] += (rcCommand[THROTTLE] > initialThrottleHold) ?
        -ALT_HOLD_THROTTLE_NEUTRAL_ZONE : ALT_HOLD_THROTTLE_NEUTRAL_ZONE;
        AltHold = sonarAlt;
        initialThrottleHold = rcCommand[THROTTLE];
    }

    // 維持機能 start
    if (abs(AltErr[1]) > ALT_HOLD_SONAR_NEUTRAL_ZONE) { //高度変化があったら
        isAltHoldChanged = 1;
        //フィルタリング (外れ値の除外)
        if(sonarAlt > 190) {
            sonarAlt = PresonarAlt;
            rcCommand[THROTTLE] += GAIN_P * AltErr[1] + GAIN_I * ErrIntg; //PI 制御
        } else {
            rcCommand[THROTTLE] += GAIN_P * AltErr[1] + GAIN_I * ErrIntg; //PI 制御
            PresonarAlt = sonarAlt;
        }
    }
    // 維持機能 end
    else {
        if (isAltHoldChanged) {
            AltHold = sonarAlt;
            isAltHoldChanged = 0;
        }
        rcCommand[THROTTLE] = initialThrottleHold;
    }
#endif
```

B.接近機能

```
AltErr[0] = AltErr[1];
AltErr[1] = sonarAlt - TARGET_DISTANCE;
ErrIntg += (AltErr[0]+AltErr[1])/2 * INT_TIME;

// 接近機能 start //
#if defined(APPROACH_FUNCTION)
    if(abs(rcCommand[THROTTLE]-initialThrottleHold) >
    ALT_HOLD_THROTTLE_NEUTRAL_ZONE){ //スロットルが変化したら
        rcCommand[THROTTLE] += (rcCommand[THROTTLE] > initialThrottleHold) ?
        -ALT_HOLD_THROTTLE_NEUTRAL_ZONE : ALT_HOLD_THROTTLE_NEUTRAL_ZONE;
        AltHold = sonarAlt;
        initialThrottleHold = rcCommand[THROTTLE];
    }
    //フィルタリング（外れ値の除外） start
    if(sonarAlt > 190){
        sonarAlt = PresonarAlt;
        rcCommand[THROTTLE] += GAIN_P * AltErr[1] + GAIN_I * ErrIntg; //PI 制御
    }else{
        rcCommand[THROTTLE] += GAIN_P * AltErr[1] + GAIN_I * ErrIntg; //PI 制御
        PresonarAlt = sonarAlt;
    }
    //フィルタリング（外れ値の除外） end
#endif
// 接近機能 end //
```

C.探索機能

```
AltErr[0] = AltErr[1];
AltErr[1] = sonarAlt - TARGET_DISTANCE;
ErrIntg += (AltErr[0]+AltErr[1])/2 * INT_TIME;

// 探索機能 start //
#if defined(SCAN_FUNCTION)
    if (abs(rcCommand[ROLL]) > SCAN_ROLL_NEUTRAL_ZONE) { //ロールが変化したら
        rcCommand[ROLL] += (rcCommand[ROLL] > 0) ? - SCAN_ROLL_NEUTRAL_ZONE :
SCAN_ROLL_NEUTRAL_ZONE;
        ScanCount = 0;
    } else {
        if (abs(AltErr[1]) > TARGET_LOST) { //樹枝を見失ったとき
            TargetLost = 1;
            //横移動による見失い start
            if (accSmooth[ROLL] > 10) {
                rcCommand[ROLL] = 5;
                ScanCount = 0;
            }
            else if (accSmooth[ROLL] < -10) {
                rcCommand[ROLL] = -5;
                ScanCount = 0;
            }
            //横移動による見失い end
        } else {
            //上昇による見失い start
            if ((ScanCount < 500) || (ScanCount > 1500)) {
                rcCommand[ROLL] = 3;
                ScanCount++;
                if (ScanCount == 2000) {
                    ScanCount = 0;
                }
            }
            if ((ScanCount >= 500) && (ScanCount <= 1500)) {
                rcCommand[ROLL] = -3;
                ScanCount++;
            }
            //上昇による見失い end
        }
    }
} else {
```



```
    if (TargetLost) {  
        TargetLost = 0;  
        rcCommand[ROLL] = 0;  
        ScanCount = 0;  
    }  
    AltHold = sonarAlt;  
}  
}  
#endif  
// 探索機能 end //
```