

## Abstract

In Japan, a lot of bridges built during high economic growth period from the 1950s to the 1970s get older now. In order to safely use old bridges in the future, periodical inspection is necessary. Many bridges are inspected currently by visual inspection and hammering test by humans to detect cracks and loose bolts. However, bridge inspections by humans are difficult due to the number of bridges, risks of accidents for working persons and traffic control during the inspection. Therefore, various UAVs (Unmanned Aerial Vehicles) have been developed to inspect bridges easily and safely. It is hard to introduce manual operation UAVs to inspection because manual operation is difficult for inspectors who do not have control skills. So, it is necessary to develop autonomous UAV in order to promote the introduction of UAVs. Most of the autonomous UAVs are dependent on GPS (Global Positioning System); however, these UAVs cannot be used for bridge inspection because the bridge inspection UAV flies under the bridge. In this study, a position estimation system using UWB (Ultra-Wide Band), IMU (Inertial Measurement Unit) and a distance sensor and flight control system for the inspection UAV aiming at autonomous flight in non-GPS environment were developed.

## 概要

現在日本ではインフラの老朽化が進んでおり、特に橋梁は2023年に半数以上が築50年を迎える。老朽化した橋梁は定期的な点検が必要になるが、点検に必要なコストや点検作業者の不足が深刻な問題になっている。そのため点検作業の簡略化が望まれており、点検用 UAV (Unmanned Aerial Vehicle) の開発・導入が進められている。しかし UAV の操縦経験が浅い点検作業者にとって、点検にマニュアル操縦の UAV の導入することは困難である。したがって自律飛行型の点検用 UAV の開発を進める必要がある。今ある自律飛行型 UAV の多くは GPS (Global Positioning System) に頼って飛行を行うが、橋梁の点検の場合、ロボットが橋梁の下に潜って点検を行うため、GPS を使うことができない。そこで本研究では点検用 UAV の非 GPS 環境での自律飛行を目的として、UWB (Ultra-Wide Band) や慣性センサ、距離センサを組み合わせた自己位置推定法及び Integral Backstepping 制御を使った自律飛行制御手法を提案する。また小型のクアッドローターを用いて実機での提案手法の評価を行った。

# 目次

|  |           |
|--|-----------|
| <b>1. 緒言</b>   | <b>1</b>  |
| 1.1 研究背景 . . . . .                                       | 1         |
| 1.2 インフラ点検用 UAV の開発 . . . . .                            | 2         |
| 1.3 研究目的 . . . . .                                       | 4         |
| <b>2. クアッドロータの位置推定アルゴリズム</b>                             | <b>6</b>  |
| 2.1 拡張カルマンフィルタの原理 . . . . .                              | 6         |
| 2.2 UWB 通信 . . . . .                                     | 8         |
| 2.2.1 UWB 通信の測距原理（発散の回避方法, 取付角の工夫） . . . . .             | 8         |
| 2.2.2 更新周波数の改善のアプローチ（マイコンスペックの比較, 周波数グラフの比較） . . . . .   | 16        |
| 2.3 オプティカルフローセンサ . . . . .                               | 16        |
| 2.3.1 オプティカルフローセンサの原理（センサ座標系のワールド座標系への変換） . . . . .      | 16        |
| 2.3.2 センサのキャリブレーション . . . . .                            | 17        |
| 2.4 UWB モジュール, IMU, 距離センサのみを用いた位置推定システム . . . . .        | 21        |
| 2.5 従来の構成にオプティカルフローセンサも加えた位置推定システム（複数センサの統合環境） . . . . . | 28        |
| <b>3. 位置制御用クアッドロータの開発</b>                                | <b>32</b> |
| <b>4. 位置推定アルゴリズムの精度評価</b>                                | <b>33</b> |
| 4.1 カルマンフィルタの分散が精度に与える影響の考察 . . . . .                    | 35        |
| <b>5. 位置制御アルゴリズムの評価</b>                                  | <b>35</b> |
| 5.1 位置制御手法 . . . . .                                     | 35        |
| 5.2 ステレオカメラを用いた位置測定システム . . . . .                        | 36        |
| 5.2.1 位置計測システムの原理及び構築 . . . . .                          | 36        |
| 5.2.2 奥行き誤差の検証 . . . . .                                 | 41        |
| 5.3 ロータの推力測定 . . . . .                                   | 42        |
| 5.4 従来の推定アルゴリズムによる制御性能の評価 . . . . .                      | 43        |
| 5.4.1 定点における位置保持実験 . . . . .                             | 43        |
| 5.4.2 複数点間を移動する位置制御実験 . . . . .                          | 44        |
| 5.5 提案する推定アルゴリズムによる制御性能の評価 . . . . .                     | 45        |
| <b>6. 結言</b>   | <b>46</b> |
| 謝辞   | 48        |
| 参考文献   | 49        |

# 1. 緒言

## 1.1 研究背景

近年の日本において、インフラ、特にその中においても橋梁の老朽化が問題視されている。国土交通省が発表するデータによると、長さ 15 m を超える橋梁の数は 2013 年時点で 40 万橋存在し、その中でも建設後 50 年が経過したものの割合は 18 % (7 万 1000 橋) である。こうした老朽化した橋は今後も増加するとされ、10 年後には 43 %、20 年後には 67 % と、過半数の橋梁が建設後 50 年を迎えることとなる。適切に管理、検査がなされていない老朽化した橋梁はやはり崩落事故などの危険性をはらんでいる。橋梁の崩落事故はアメリカのミネアポリスのものが有名である。しかし、日本国内においても崩落や崩落事故になりかけた事例がいくつか存在する。2007 年の 6 月三重県にある国道 23 号の木曽川大橋で、コンクリートの床版に覆われたトラスの斜材が腐食して破断するという事故が起こった。幸いにもこの事故では崩落や死傷者を出すことは免れたが、ずさんな点検のあり方が浮き彫りになった。事故の 1 年半ほど前には点検が行われたとされているが、腐食の進んだ鋼材の錆の上から塗装を行うなど、常識を逸脱する補修方法が行われているのにも関わらず、それが見過ごされていた。こういった不適切な橋梁の管理、運営には橋梁を所有する地方自治体の財源不足なども背景とされている。国土交通省によると、1 橋を近接目視で点検するだけでも 50 万円の費用がかかるとされている。しかし、市区町村が管理する橋の年間の維持、修繕費は 1 橋平均でわずか 8 万円であり、いかに橋梁の点検に掛ける予算が低いのかがわかる。また、橋梁の点検に用いられる一般的な方法は近接目視及び打音検査であり、傷やひび割れの正確な検出には多くの経験値を要する。これらの理由より、低コストかつ簡便に橋梁を点検可能にする新しい技術を開発することが現在の日本において喫緊の課題である。

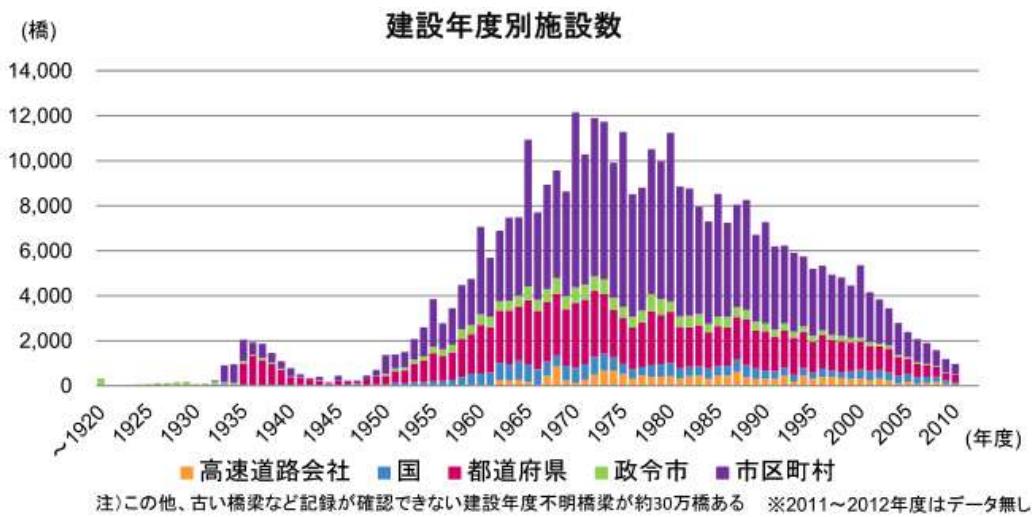


Fig. 1 Number of bridges in Japan by construction year

そこで、着目したのがドローンなどのロボットを用いた橋梁の点検手法である。ここで、ロボットを用いた点検手法は大別して 2 種類に分けることができる。1 つ目は機体に車輪を有し、橋梁の壁面や床面を伝って移動することで点検箇所へアプローチする地上型ロボットである。

もう一つはドローンなどの飛行型ロボットでありこれは前者と異なり、空中から点検箇所にアプローチすることが可能である。地上型ロボットとしては高田らが開発した BIREM が挙げられる。このロボットは車輪の先端に磁石を装備しており、吸着対象が磁性体ならば、床や壁を伝いながら点検箇所へ接近し、カメラを用いた目視検査などがすることが可能である。しかし、このロボットは点検箇所まで構造物が地続きである必要があり、橋梁の途中で大きなギャップがあった場合にその先に進めないといったことや、大きな段差があった場合にそれらを乗り越えられないといった問題もある。しかし、飛行型のロボットの場合、大きなギャップや障害物があってもそれらを乗り越えて点検箇所まで近づける。これらの理由より、本研究では飛行型ロボットを開発するに至った。



Fig. 2 BIREM

## 1.2 インフラ点検用 UAV の開発

先行研究において、赤堀らはインフラ点検用 UAV を開発した<sup>[1]</sup>。この UAV は機体の上部に EPM (Electro-Permanent Magnet) を搭載し、磁力を用いて構造物の鋼材に吸着することで、安定した点検を可能にした。この EPM と呼ばれる磁石の特徴は、磁力の吸着状態と開放状態を自在に制御することが可能なことである。また、この磁石は電磁石と異なり、吸着と離脱時のみ電力を消費し、吸着中は電力を消費しない。この UAV は EPM に加え、先端にカメラを有するアームも装備するため、EPM を用いて吸着した状態で安定した目視点検をすることが可能である。

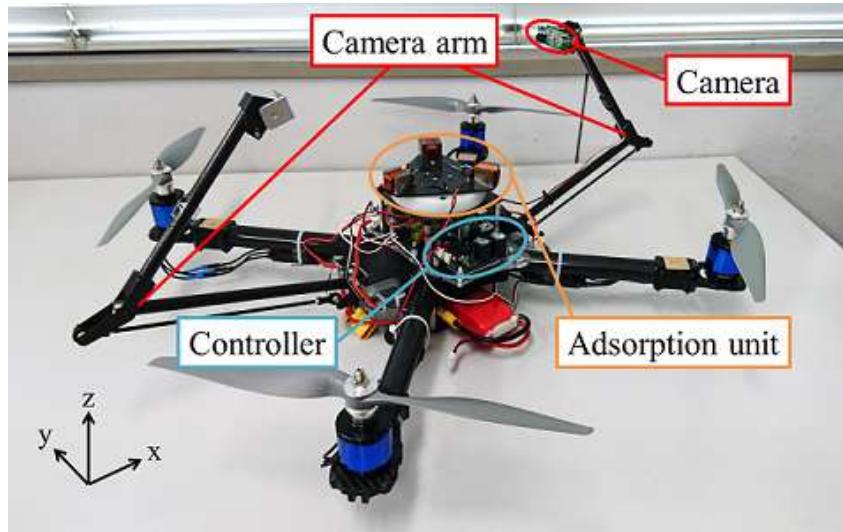


Fig. 3 Overall view of the developed inspection robot in the previous study

また、自身の先行研究ではこのEPMを吸着機構に採用した振動計測モジュールを開発した[]。このモジュールは加速度センサ及びデータロガーを搭載し、対象物に吸着した後、加速度を計測、ロギングすることが可能である。その振動を解析することで対象物の健全性を評価することが可能である。点検方法としては、点検箇所までUAVを用いて運搬し、EPMの磁力で貼り付け、必要なデータを計測した後、再びUAVで回収する流れである。また、重量が約230gであるため、赤堀らが開発したUAVなどであれば、複数個積載し、運搬することが可能である。

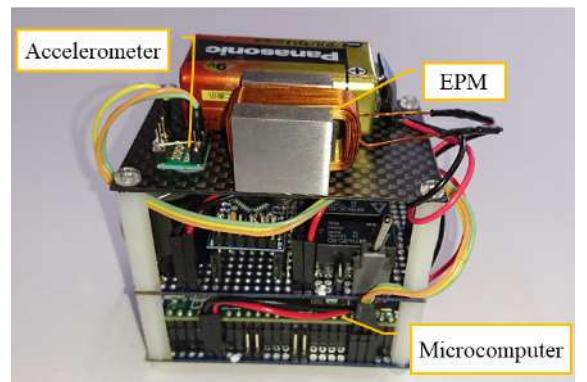
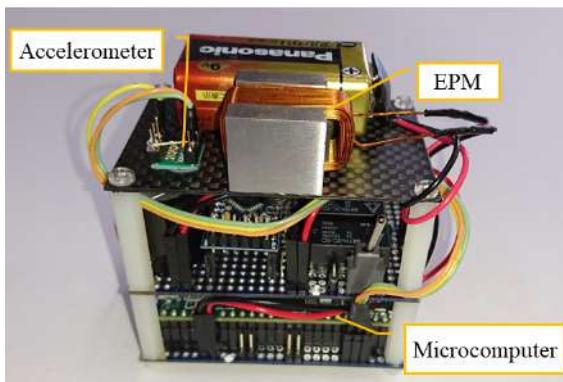


Fig. 4 Front view of vibration measurement unit Fig. 5 Front view of vibration measurement unit

これらのUAVを用いた点検では、構造物の地形に対して柔軟な対応性を有するが、機体の操作は全て自身の手で行う必要性がある。経験の豊富なパイロットであれば、狙い通りの点検箇所に接近することやセンサを取り付けることが可能であるが、経験の浅いパイロットであると、そういったことは非常に困難である。

### 1.3 研究目的

上記の理由より、飛行型ロボットの有用性は高いが、操作面においてまだまだ課題が多いのが現状である。そこで、現在市販される多くのドローンはGPSなどの位置計測システムを用いて全自動若しくは半自動飛行が可能である[1][2]。本研究でも用いたライトコントローラもGPSアンテナが搭載されており、屋外のGPS環境であれば、目的地を入力するだけでその地点まで全自動飛行することが可能である。しかし、研究対象とする橋梁などの構造物の周辺環境においては、遮蔽物によりGPSの電波が遮断されるため、GPSを用いた自動飛行は不可能である。そこで、非GPS環境においても自動飛行可能なUAVも開発が進んでいる。株式会社自律制御システム研究所はSLAM（Simultaneous Localization and Mapping）の機能を搭載したUAVを開発しており、非GPS環境においても自己位置推定と環境地図の作製を同時にいながら自律飛行をすることが可能である。



Fig. 6 ACSL-PF1

手法としては搭載された下向きのモノラルカメラや前方に向けて取り付けられたステレオカメラを用いて撮影した画像に処理を施すことで自己位置や方角を推定し自律飛行を行う。しかし、それらをするためには高価で精度の高いカメラを複数個積むことや画像処理用のGPUを積む必要があるため、機体自体の値段が高価になる。また、画像から得た情報を基にあらゆる値を計算するには計算負荷の高い演算をしなくてはならない。点検に掛けられる財源が限られている自治体にとって、人件費がかからなかったとしても、機体が高価であった場合、大きなコスト削減は見込めない。また、多くのセンサや負荷が高い計算をするに伴って性能の高いCPUやGPUを搭載する必要があるため、機体が大型になってしまい、狭い点検空間に入していくのが難しくなるといった課題がある。

そこで、それらの課題を克服するため、本研究ではローカル座標を推定する手法にUWBモジュールを用いる手法に着目した。UWB（超高帯域無線通信、Ultra Wide Band）とは無線通信方式の一種であり、Wi-FiやBluetoothなど他の無線通信方式に比べ広い帯域幅を持った通信方式である。これを利用し、電波を送信し返ってきた時間より距離を測定することができるため、UAVの位置を3次元空間内で推定することが可能である。模式図をFig. 7に示す。

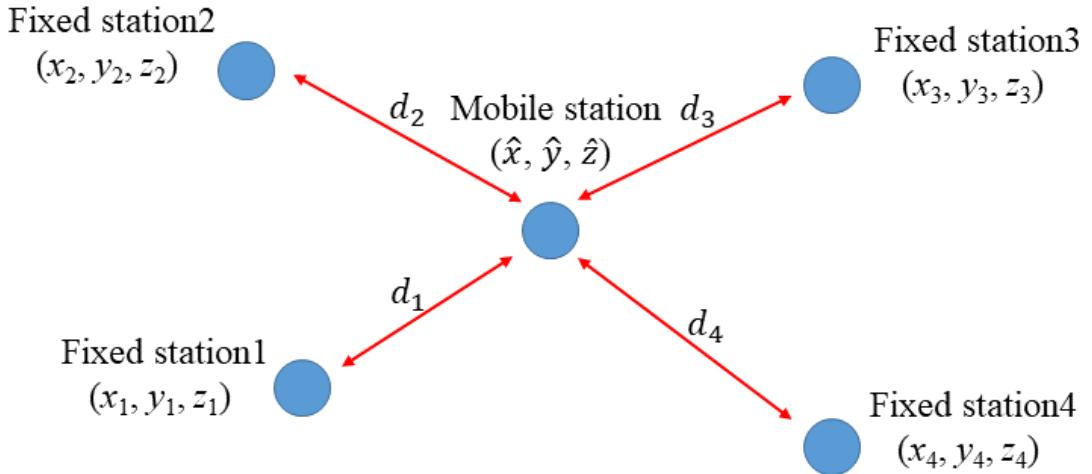


Fig. 7 Schematic diagram of the position estimation method

詳細は以降の章にて後述するが、座標が既知の各固定局と移動局との距離を計測し、マルチラテレーションの手法を用いることで、移動局の座標を得ることが可能である。この手法は GPS アンテナと GPS 衛星との距離を用いて移動局の座標を得るのと同手法である。また UWB モジュールは一個あたり数千円と非常に安価であり、機体のコスト削減に有効である。それに加えてマルチラテレーションの計算は画像解析を用いる計算などに比べて非常に負荷が軽く、高性能なコンピュータを乗せる必要がない。しかし、UWB モジュールの通信周期は 4 つの固定局を用いた場合に 10 Hz と低く、クアッドロータが高速度で移動した際には位置推定周期としては不十分である。そこで、本研究での目的は UWB モジュールの更新周期の遅さを補うために UWB モジュールを用いたローカルポジショニングシステムに IMU、距離センサ、及びオプティカルフローセンサの値も参照することでクアッドロータの位置を推定し、その位置を制御する手法を提案する。IMU (Inertial Measurement Unit) は加速度、角速度、地磁気を数 1000 Hz 程度の高い更新周波数で計測可能であり、加速度や角速度を積分することで速度や変位を得ることが可能である。しかし、値を積分した際に誤差が蓄積してしまう。また距離センサは計算することなく、直接地面との距離を計測することが可能である。最後にオプティカルフローセンサは地面の特徴点の移動量より機体の速度を算出することが可能である。先行研究において赤堀ら<sup>11</sup>は UWB モジュール、IMU、距離センサを用いたクアッドロータの位置推定システム及び位置制御システムを開発した。しかし、姿勢角制御の周期の遅さが原因でクアッドロータの位置を制御することはなし得ていなかった。そこで、本研究では姿勢角制御の遅さを補うためにライトコントローラを追加で導入し、改善を図った。次章からは UWB モジュール及び各種センサをカルマンフィルタに適用したクアッドロータの位置推定方法および位置制御方法を示していく。

## 2. クアッドロータの位置推定アルゴリズム

本研究で提案するクアッドロータの位置及び姿勢推定にはカルマンフィルタ, その中でも特に非線形なシステムに適応可能な拡張カルマンフィルタ (EKF : Extended Kalman Filter) を使用している。ここで, カルマンフィルタとは, 誤差を含むある観測値 (2種類以上も可) を使用し, 時々刻々と変化する状態量を推定するのに用いられる, 逐次処理が可能なフィルタリング手法である。代表例としてはカーナビゲーションシステムが挙げられる。カーナビゲーションシステムはGPSから得られる誤差の乗った位置情報と内部の加速度センサから得られる誤差の乗った加速度情報を統合することにより, 自車の位置座標や速度などを推定している。本研究ではIMU, UWBモジュール, 距離センサ, オプティカルフローセンサから得られるセンサ値を統合し, 機体の姿勢角, 速度, 座標を推定するのに用いられる。本章では初めに拡張カルマンフィルタの原理を示した後, UWBモジュールやオプティカルフローセンサの原理やキャリブレーション方法について述べる。そして最後にそれらセンサを統合したシステムを示し, 位置の推定方法を説明する。

### 2.1 拡張カルマンフィルタの原理

拡張カルマンフィルタは非線形システムを書く自国において線形化し, それぞれの時刻において時変カルマンフィルタを適用するという考えに基づいている。まず離散時間非線形状態空間表現が次の二式で表される場合を考える。

$$\mathbf{x}(t+1) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{B}\mathbf{v}(t) \quad (1)$$

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t)) + \mathbf{w}(t) \quad (2)$$

ここで  $\mathbf{f}(\cdot)$ , および  $\mathbf{h}(\cdot)$  はベクトル値をとる  $\mathbf{x}(t)$  の非線形関数である。また  $\mathbf{v}(t)$  は平均値ベクトル 0, 共分散行列  $\mathbf{Q}$  の  $r$  次元システム雑音ベクトル,  $\mathbf{w}(t)$  は平均値ベクトル 0, 共分散行列  $\mathbf{R}$  の  $p$  次元観測雑音ベクトルであり, 互いに独立な正規性白色雑音と仮定する。時刻  $t$ ,  $t+1$  において, それぞれ事前状態推定値  $\hat{\mathbf{x}}^-$  と事後推定値  $\hat{\mathbf{x}}$  が利用可能であるという仮定のもとで, 式1と式2の非線形関数を泰ラー級数展開を用いて線形近似すると,

$$\mathbf{f}(\mathbf{x}(t)) = \mathbf{f}(\hat{\mathbf{x}}(t)) + \mathbf{A}(t)(\mathbf{x}(t) - \hat{\mathbf{x}}(t)), \mathbf{A}(t) = \left. \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}(t)} \quad (3)$$

$$\mathbf{h}(\mathbf{x}(t)) = \mathbf{h}(\hat{\mathbf{x}}^-(t)) + \mathbf{C}^T(t)(\mathbf{x}(t) - \hat{\mathbf{x}}^-(t)), \mathbf{C}^T(t) = \left. \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}^-(t)} \quad (4)$$

が得られる。式(3)を式(1)に, 式(4)を式(2)に代入すると, それぞれ,

$$\mathbf{x}(t+1) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{b}\mathbf{v}(t) + \mathbf{f}(\hat{\mathbf{x}}(t)) - \mathbf{A}(t)\hat{\mathbf{x}}(t) \quad (5)$$

$$\mathbf{y}(t) = \mathbf{C}^T(t)\mathbf{x}(t) + \mathbf{w}(t) + \mathbf{h}(\hat{\mathbf{x}}^-(t)) - \mathbf{C}^T(t)\hat{\mathbf{x}}^-(t) \quad (6)$$

が得られる。さらに

$$\mathbf{u}(t) = \mathbf{f}(\hat{\mathbf{x}}(t)) - \mathbf{A}(t)\hat{\mathbf{x}}(t) \quad (7)$$

$$\mathbf{z}(t) = \mathbf{y}(t) - \mathbf{h}(\hat{\mathbf{x}}^-(t)) + \mathbf{C}^T(t)\hat{\mathbf{x}}^-(t) \quad (8)$$

とおくと、式(5)と式(6)はそれぞれ次のようになる。

$$\mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{v}(t) + \mathbf{u}(t) \quad (9)$$

$$\mathbf{z}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{w}(t) \quad (10)$$

このように非線形システムをテイラー級数展開を用いて線形近似すると制御入力がある場合のカルマンフィルタになる。以上より予測ステップにおける更新式は

$$\hat{\mathbf{x}}^-(t) = \mathbf{f}(\hat{\mathbf{x}}(t-1)) \quad (11)$$

$$\mathbf{P}^-(t) = \mathbf{A}(k-1)\mathbf{P}(t-1)\mathbf{A}^T(k-1) + \mathbf{B}\mathbf{Q}\mathbf{B}^T \quad (12)$$

となる。ここで  $\mathbf{A}(t-1)$  は

$$\mathbf{A}(t-1) = \left. \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}(t)} \quad (13)$$

となる。またフィルタリングステップの更新式は

$$\hat{\mathbf{x}}(t) = \hat{\mathbf{x}}^-(t) + \mathbf{g}(t)(\mathbf{y}(t) - \mathbf{h}(\hat{\mathbf{x}}^-(t))) \quad (14)$$

$$\mathbf{P}(t) = (\mathbf{I} - \mathbf{G}(k)\mathbf{C}^T(t))\mathbf{P}^-(k) \quad (15)$$

となる。ここでカルマンゲイン  $G(t)$  及び  $\mathbf{C}^T(t)$  は次式で計算することができる。

$$\mathbf{G}(t) = \mathbf{P}^-(k)\mathbf{C}^T(\mathbf{C}\mathbf{P}^-(k)\mathbf{C}^T + \mathbf{R})^{-1} \quad (16)$$

$$\mathbf{C}^T(t) = \left. \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}^-(t)} \quad (17)$$

拡張カルマンフィルタにおいては各時刻ごとに  $\mathbf{f}$  と  $\mathbf{h}$  のヤコビアンを求めるため偏微分の計算を行わなければならない。そのため、微分可能な滑らかな非線形性の場合には拡張カルマンフィルタを適用できるが、不連続な非線形性をもつ場合には適用できない。

## 2.2 UWB 通信

### 2.2.1 UWB 通信の測距原理（発散の回避方法、取付角の工夫）

まず、無線通信を利用した距離測定の技術は Bluetooth や Wi-Fi を用いたものなどが挙げられる。片方のデバイスが電波を発し、それをもう片方が受け取り、デバイス同士が通信を行う。測距の方法は、受信側の受け取った電波の強弱により発信側が近くにあるのか遠くにあるのかを判別し、距離を算出する。しかし、弱い電波を受け取った際、距離が離れているため電波が弱いのか、それともデバイス間に障害物があるため電波が弱くなっているのか、といったことを判別できない。したがって、Wi-Fi や Bluetooth を用いた測距は誤差が 5 m ほど乗るとされている。一部の企業は Wi-Fi 電波の飛行時間（Time of flight, ToF）や到着時間（Time of Arrival, ToA）を使用し、距離をより正確に測るための代替アルゴリズムを開発したが、既存の Wi-Fi を用いてこれを行うことは、ハードウェアの問題により難しいとされている。

UWB 通信（超広帯域無線通信、Ultra Wide Band）とは無線通信方式の一種であり、非常に広い帯域幅にわたって電波を拡散させることで高速通信を可能とする。Fig. 3 に UWB 通信と既存の無線通信方式とのスペクトラムの比較を示す。周波数の帯域は 3.1 GHz から 10.6 GHz であり、その帯域幅は 7.5 GHz と従来の無線の数百 kHz や Wi-Fi の 20 MHz に比べてかなり広い帯域を利用している。

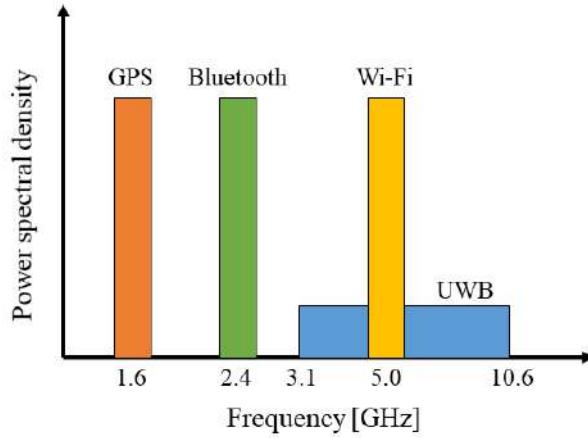


Fig. 8 Comparison of power spectrum between UWB and other wireless communication system

送信出力は-41.3 dBm/MHz 以下に制限されており、これは家庭用テレビやパソコン等の一般的な電子機器等が発生する雑音レベルの約 500 分の 1 以下という非常に小さな出力である。また、UWB 通信は Fig. 9 に示すようにナノ秒オーダーのパルスを通信に使用し、狭帯域信号のような立ち上がりが遅い信号に比べて、デバイス間の距離を高精度で測定可能である。

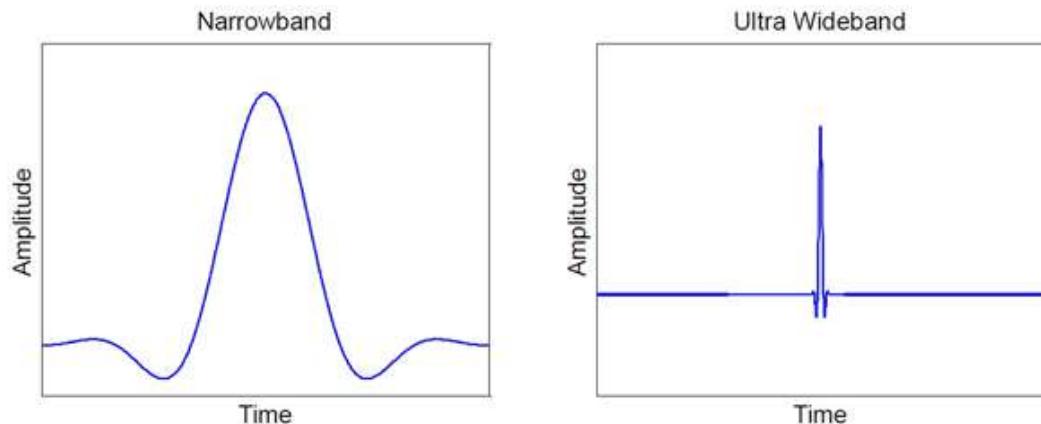


Fig. 9 Comparison of pulses between narrow band and ultra wide band signals

また、Fig. 10, Fig. 11 に示すように狭帯域信号に比べ、UWB 信号は SN 比が良いため、ノイズの影響を受けにくいくことなどが特長として挙げられる。

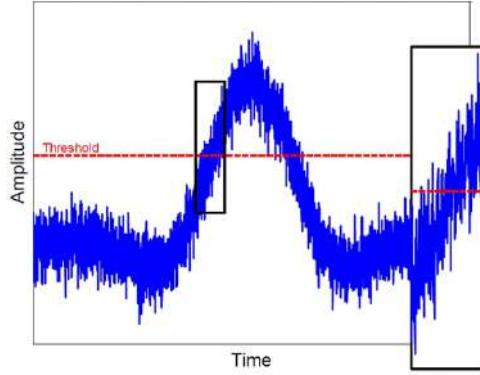


Fig. 10 Narrow band signal with noise

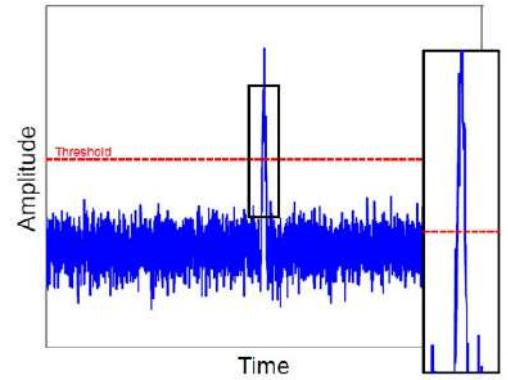


Fig. 11 Ultra wide band signal with noise

無線通信を利用する際に大きな問題とされるのがマルチパスの発生である。マルチパスとは電波を受信した際に、直接波の他に壁などの障害物に反射した反射波を受信してしまう場合があり、この際、正確な信号受け取れないといったことや、直接波の位相を反射波が遅らせてしまうといったことが発生する。テレビやラジオにてゴースト障害が起きるもの、このマルチパスが原因とされている。無線通信を利用した測距の場合には、正確な到達時間を計測できず、測距誤差が発生するということが起きる。次の Fig. 12, Fig. 13 に狭帯域信号と UWB 信号のマルチパスの影響の比較を示す。

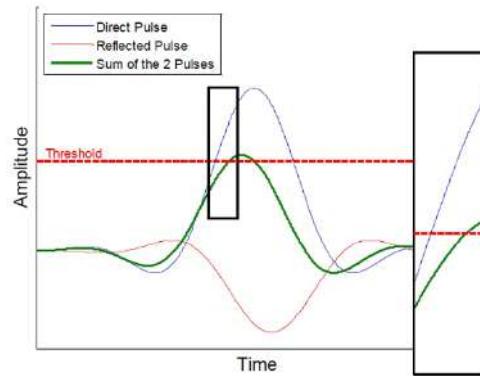


Fig. 12 Narrow band signal with reflections

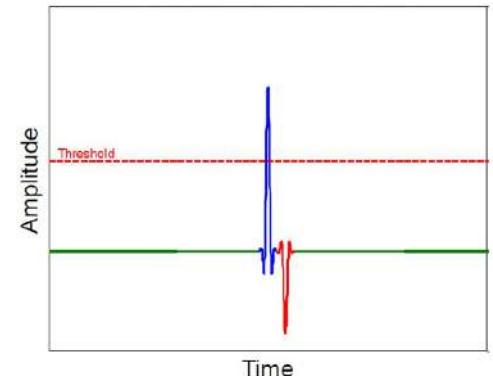


Fig. 13 Ultra wide band signal with reflections

狭帯域信号では立ち上がりと立ち下がりが鈍く、障害物で位相が反転した反射波の影響により直接波の位相や振幅が変化させられているのが分かる。しかし、UWB 信号は立ち上がり時間と立ち下がり時間が短いため、直接波の後に反射波が来ていても直接波に悪影響を与えない。このようなマルチパス耐性が高いといったことも UWB の特長である。

次に UWB 通信を用いた測距方法について見ていく。UWB 通信による測距は主に電波の飛行時間 (ToF) を用いている。送信側と受信側が互いに通信し、電波の到達するのに掛かった時間 (Tof) に電波の速度 ( $c$ ) を掛けることにより、距離 ( $d$ ) を測定することが可能である。次に計算式を示す。

$$d = \text{ToF} \times c \quad (18)$$

また Fig. 14 に通信プロトコルを示す.

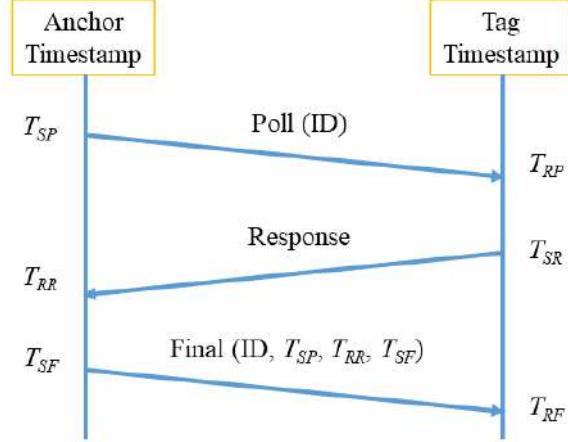


Fig. 14 Ranging protocol of UWB module

これ以降、クアッドロータに搭載された移動局をタグ、床に固定された固定局をアンカと記載する。クアッドロータの座標を3次元で推定するためには1つのタグと3つ以上のアンカが必要である。それに伴い、複数のデバイスを識別するためにまずアンカ側から16進数からなるIDを送信する。その後タグ側から応答があり、最後にアンカ側のタイムスタンプ( $T_{SP}, T_{RR}, T_{SF}$ )を送信する。この送られてきたタイムスタンプ及びタグ側のタイムスタンプ( $T_{RP}, T_{SR}, T_{RF}$ )を用い、次式のように電波の飛行時間(ToF)を計算可能である。

$$\text{ToF} = \frac{1}{4}((T_{SP} - T_{RR}) - (T_{SR} - T_{RP}) + (T_{RF} - T_{SR}) - (T_{SF} - T_{RR})) \quad (19)$$

次のFig. 15 に製作したUWBモジュールのアンカを示す。搭載されたUWBチップはDecaWave社製のDWM1000である。マイコンはTeensy3.6を用いて制御を行っている。電源はFig. 16に示すUltraLife社製、1 cell, 1800 mAh のリチウムイオンバッテリを用いている。図の基板構成で満充電から約3時間容量が保つ。Fig. 15 からわかるように、小型であるためアンカの設置や回収は容易に行うことが可能である。

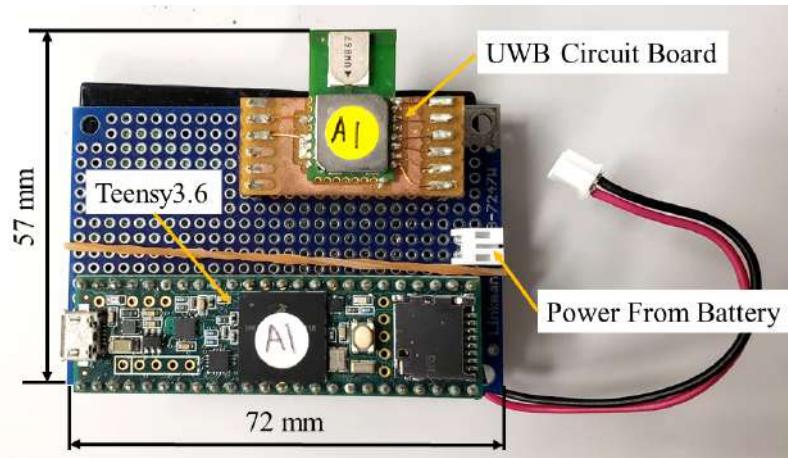


Fig. 15 UWB anchor



Fig. 16 Battery for UWB Anchor

前述したように UWB 通信はマルチパスに耐性があり、また [] でも示されているように、壁のように強い反射体の近くにおいても測距性能は正確であるが、森林のようなデバイス間に障害物が存在する環境：NLOS(Non Line Of Sight, 非見通し環境)においてはマルチパスが発生し、真値より大きく外れた値や負の値を取ることがある。本研究においても、このマルチパスの影響を確認した。Fig. 17 に測距した際にマルチパスが発生し、値が発散した結果を示す。またそれの拡大図も Fig. 18 に合わせて示す。

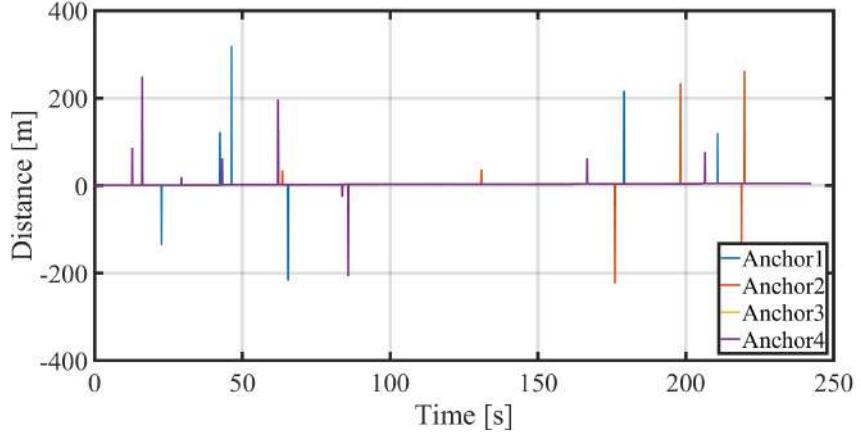


Fig. 17 Ranging result when outliers occur

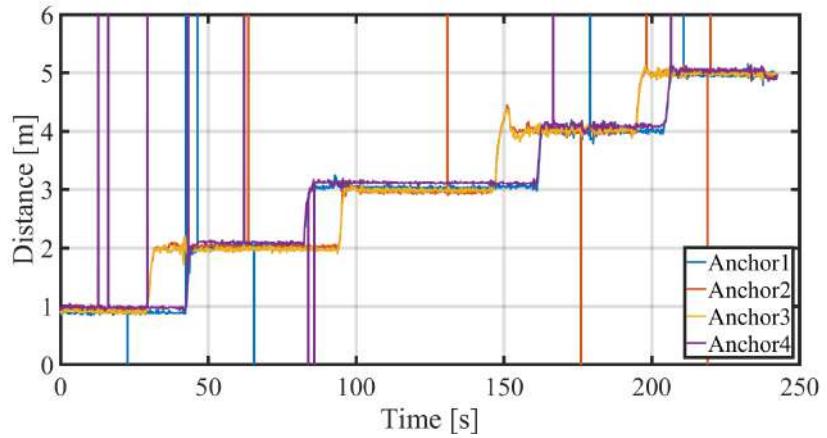


Fig. 18 Ranging result when outliers occur (enlarged image above)

測距手順はまず、実験室の床に距離を測り、真値の点を 1 m から 5 m まで 1 m 間隔で取った。そして、その点に合わせる形で 4 つのアンカを 1 m ずつ動かし、それぞれのアンカと固定したタグとの間の測距を UWB モジュールにより計測した。2 つの図からわかるように、測距した最長の距離は 5 m であるのにも関わらず、値が発散し最長 300 m ほどの距離や負の距離を計測している。UWB モジュール間に障害物が無かったのにも関わらず、こういったマルチパスが発生した原因は、壁や天井により電波が反射したためであると考えられる。タグと各アンカとの正確な距離が得られていない場合、3 次元空間内におけるクアッドロータの位置推定値も不正確になるため、こういった発散した値は取り除く必要がある。そこで、まず初めに行ったのが電波の指向性を考慮したモジュールの改良である。UWB モジュールには指向性があり、電波強度の強い向きと弱い向きが  $x$ ,  $y$ ,  $z$  軸の回転方向に存在する。本研究にて採用した UWB チップ (DWM1000) の軸を次の Fig. 19 のように定義する。

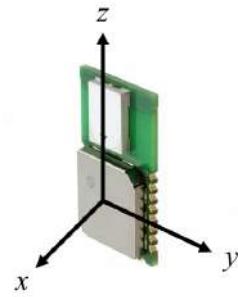


Fig. 19 UWB chip axis

DWM1000 を製造する DecaWave 社がチップから発せられる電波の指向性をデータシートにおいて公開している。 $x, y, z$  軸周りにおける電波の指向性を Fig. 20 から Fig. 22 に示す。

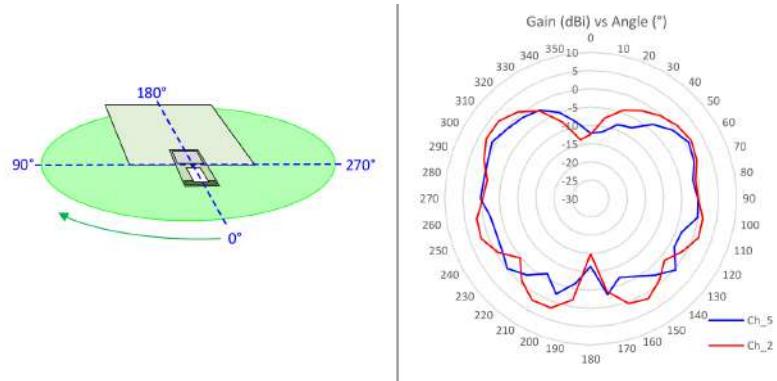


Fig. 20 Directivity of radio wave around  $x$  axis

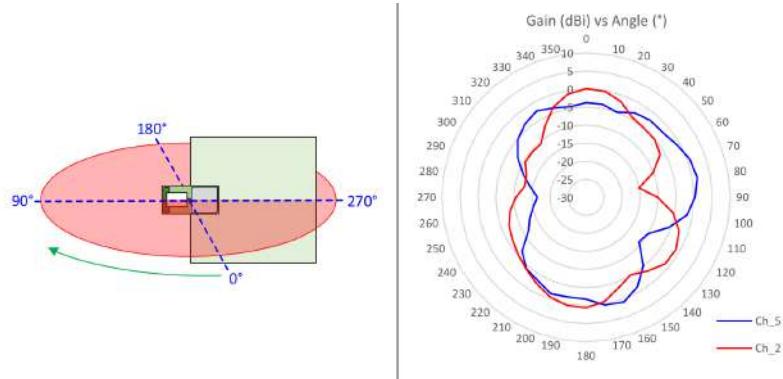


Fig. 21 Directivity of radio wave around  $y$  axis

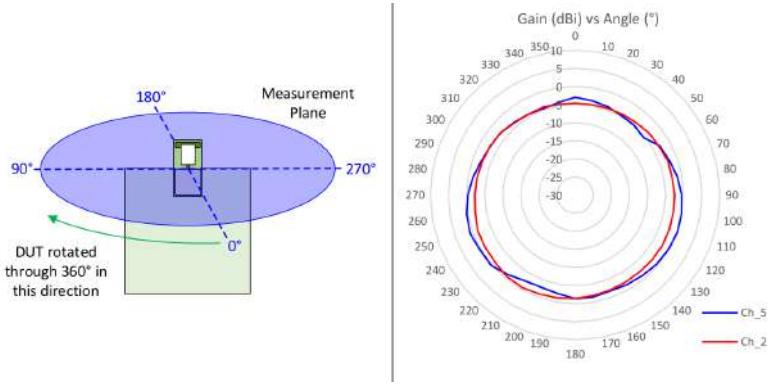


Fig. 22 Directivity of radio wave around  $z$  axis

3つの図から考慮すると、UWB チップの平面方向から垂直な方向、つまり Fig. 19 の  $x$  軸の方向が通信対象を向ければ、電波強度を確保できることがわかる。そこで、Fig. 23 に示すアンカ用のホルダを 3D プリンタにより製作した。

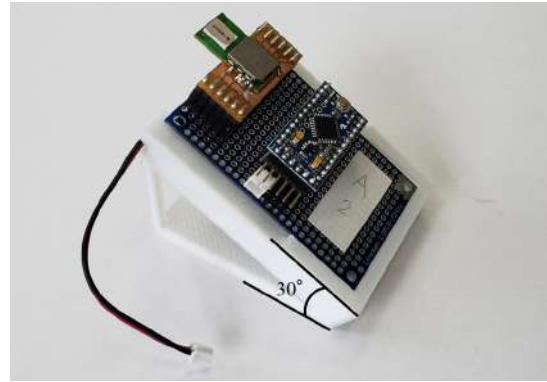


Fig. 23 UWB module holder considering radio wave directivity

これを用いる前は、実験室の壁面に対して垂直にアンカを貼り付けていたが、このホルダを用いることでチップ垂直方向をクアッドロータの方向付近に向けることが可能である。これを垂直な壁面に貼り付けることで以前より  $30^\circ$  ほどオフセット角を稼ぐことができる。そしてこのホルダをアンカの 4 つ分使用し、実験室の 4 隅に取り付け、アンカで取り囲まれる範囲内でタグを動かし、測距を行った。アンカの電波の強い方向がタグの方を向くため、マルチパスは発生しなくなると考えられたが、結果は Fig. 17 と同様に発散した値を計測した。したがってアンカの取り付け方法によってマルチパスの発生を抑えることは難しいと考えた。

次に行ったのがプログラムによる発散値の処理を行い、発散値をロギングしないという方法である。処理の内容は、アンカとタグとで通信を行い、計測した距離データを一度メモリ内にバッファする。そして、次の距離データが送られてきた際にバッファした距離と比較し、閾値を上回る変化があった場合にはその直近のデータを破棄し、前のデータを採用するというものである。距離データの発散は、条件にもよるが数秒に 1 サンプル程度の頻度で発生し、連続したサンプル同士が両方発散することはない。したがって、この処理の方法により真値からかけ離れた値のみを適切に無視することが可能である。閾値としては、クアッドロータが  $1m/s$  の

速度で水平面内を飛行可能であると考慮し、測距の1ループ内で移動不可能な距離を閾値として設定した。このプログラムによる処理を行い、発散値を回避した測距結果をFig. 24に示す。

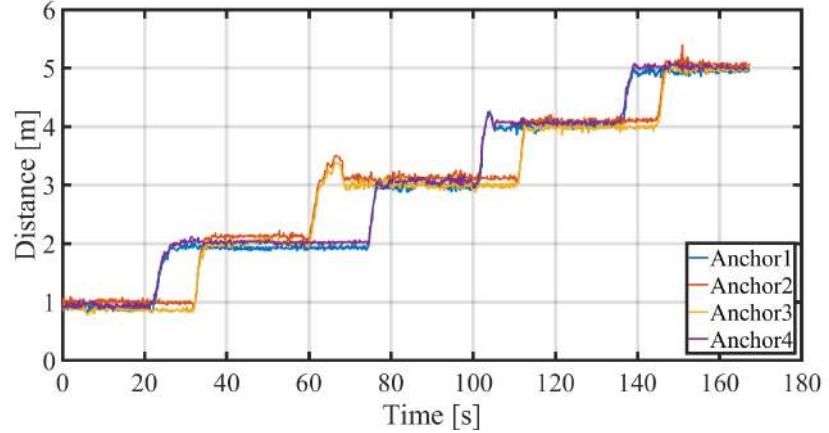


Fig. 24 Ranging result avoiding outlier

実験方法はFig. 17の実験を行った際と同様である。アンカを動かした際の振動による値のずれは生じているが、処理する前の結果と比べて適切に発散値を回避している。

### 2.2.2 更新周波数の改善のアプローチ（マイコンスペックの比較、周波数グラフの比較）

## 2.3 オプティカルフローセンサ

### 2.3.1 オプティカルフローセンサの原理（センサ座標系のワールド座標系への変換）

先行研究では、クアッドロータの位置や速度などの状態量をUWBモジュール及びIMUを用いて推定していた。しかし、UWBモジュールのアンカを4つ用いた際に得られる距離データの更新周期は10 Hzと遅く、クアッドロータが高速度で動いた際や運動の不安定さが原因の細かな振動が生じた際に更新周期が不十分であった。そこで、更新周期の遅さを補うために本研究からは新たにオプティカルフローセンサを搭載した位置推定および位置制御システムを構築した。オプティカルフローセンサとは、カメラで撮影した画像の特徴点の移動量から、移動する物体の速度や移動量を算出可能なセンサである。オプティカルフローはPython環境にてOpenCVなどのモジュールを用い、パソコン用コンピュータを使用し計算することも多い。例としては、あらかじめ撮影した車の写真から車速を割り出すことなどに使用される。そして、オプティカルフローセンサは画像の撮影から移動量の算出までを、搭載したチップで行うことが可能である。そして、それらのセンサ値をマイコンにより適切に処理することにより、搭載したロボットなどの速度や移動量を得ることができる。また、別の例としては光学式マウスはこのセンサにより移動量を算出している。[]や[]においてもクアッドロータにこのセンサを搭載し、クアッドロータの位置保持制御や目標点まで移動させる制御などを行っている。本研究で採用したオプティカルフローセンサモジュールの外観をFig. 25に示す。

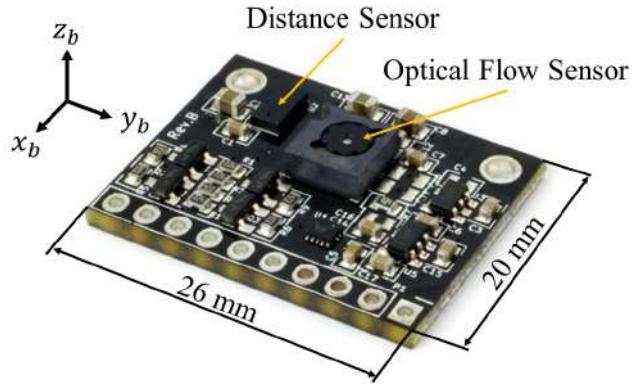


Fig. 25 Optical Flow Sensor Module

このモジュールはオプティカルフローセンサとして PMW3901 のチップを搭載しており、それに加え距離センサとして VL53L0x ToF センサを搭載している。二つのセンサが同一の基板上に搭載されているため汎用性が高いモジュールであり、オプティカルフローセンサは SPI 通信、距離センサは I<sup>2</sup>C 通信をそれぞれ用いるため、配線が少なくてすむ。また、図より大きさもとても小型であることがわかり、積載量の限られたクアッドロータなどにおいても容易に搭載可能である。後述するが、距離センサはオプティカルフローセンサから得た値を補正するのに用いられる。

### 2.3.2 センサのキャリブレーション

オプティカルフローセンサから得られる速度のキャリブレーション方法について述べる。まず、オプティカルフローセンサから得られる値は撮影した写真の変化から得られる特徴点の移動量、つまり単位は距離ではなくピクセルである。例えば、ある物体の上でクアッドロータが 100 mm 移動したとする。その際、クアッドロータの高さが低ければピクセル数の変化量は大きいが、高さが二倍になった場合、ピクセル数の変化量は二分の一になる。それを表した図が Fig. 26 である。

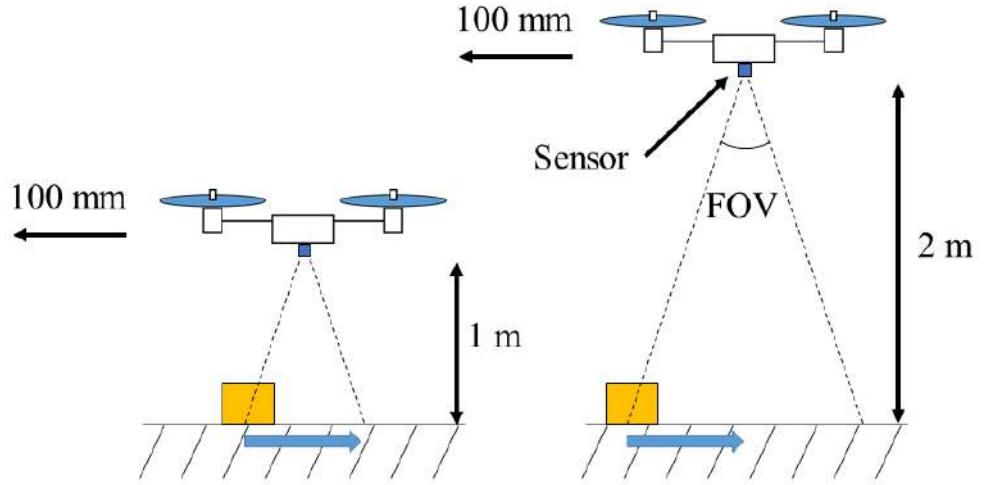


Fig. 26 Difference in Sensor Value Due to Difference in Quad-Rotor Height

したがって、オプティカルフローセンサから得られるピクセル数の変化量と距離センサから得られる距離を乗算し、適切な定数を掛けることで真の移動距離を算出することができる。なお、飛行体の高度とオプティカルフローセンサから得られる値の関係は<sup>10</sup>に詳しい記述がある。また<sup>10</sup>を参考にすると、センサが値を取得し、次の値を取得するまでをセンサの1ループと考えると、その1ループ中に移動した距離は次式で表せる。

$$\text{Moving distance} = \left( \frac{\text{Sensor value} \times \text{Altitude}}{\text{Sensor's resolution in pixels} \times \text{Scalar}} \right) \times 2.0 \times \tan\left(\frac{\text{Field of view}}{2.0}\right) \quad (20)$$

Sensor value はオプティカルフローセンサから得た値であり、 Altitude は距離センサから得た地面までの距離である。Field of view はオプティカルフローセンサが撮影可能な画角であり、 Sensor's resolution in pixels は撮影画像の画素数つまり解像度である。この二つの値はセンサ固有のものであり、それらを次の Table. 1 に示す。

Table 1 Sensor characteristic value

|                    |         |
|--------------------|---------|
| FOV [°]            | 42      |
| Resolution [pixel] | 30 × 30 |

Scalar は定数であり、この Scalar の値を次のキャリブレーションにより求めた。式 (21) から得られるのは、あくまでも 1 ループ中に動いた距離、つまり速度である。物体が移動した真の速度を用いてこの Scalar の値を求ることは可能であるが、速度を外部のセンサなどを用いて求めるのはシステムが複雑になる。そこで、速度ではなく真の移動距離を用いてこの Scalar の値を求ることにした。過去のある時点から現在までに移動した距離を算出するには、センサから得られた速度に対してマイコンが 1 ループにかかる処理時間を掛け、それを足し合わせるこ

とで可能である。つまり、センサ値を積分することである。そこで次のような実験を行った。次の Fig. 27, Fig. 28 のようにある高さに対し、センサが下を向くように固定し、一方向のみレールにより自由度を与える。レールに既知の距離を複数点マークし、その距離分だけセンサを動かした。

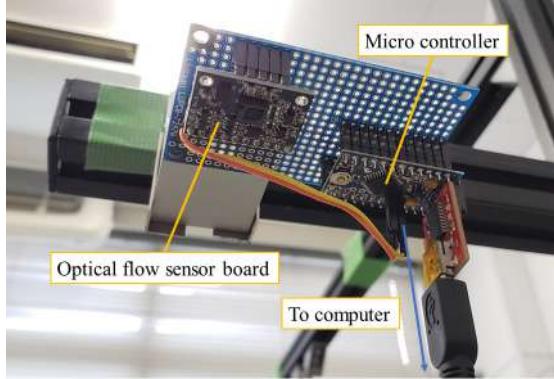


Fig. 27 Narrow band signal with reflections

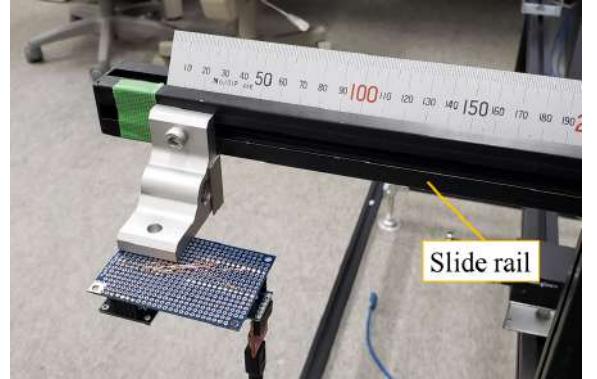


Fig. 28 Ultra wide band signal with reflections

センサ値を積分して得られた距離データと真の距離を最小二乗法により近似することで Scalar を求めた。センサを動かした距離は Table 2 に示すように 0 mm から 800 mm までの 200 mm 間隔である。固定したセンサの床からの高さは 775 mm である。また、センサ値の値は三回試行を行った上でのそれらの平均値である。

Table 2 Sensor values and moving distance

| Moving distance [mm] | Average of sensor value [-] |
|----------------------|-----------------------------|
| 0                    | 0                           |
| 200                  | 3061                        |
| 400                  | 6165                        |
| 600                  | 9198                        |
| 800                  | 12296                       |

そして、結果を Fig. 29 に示す。

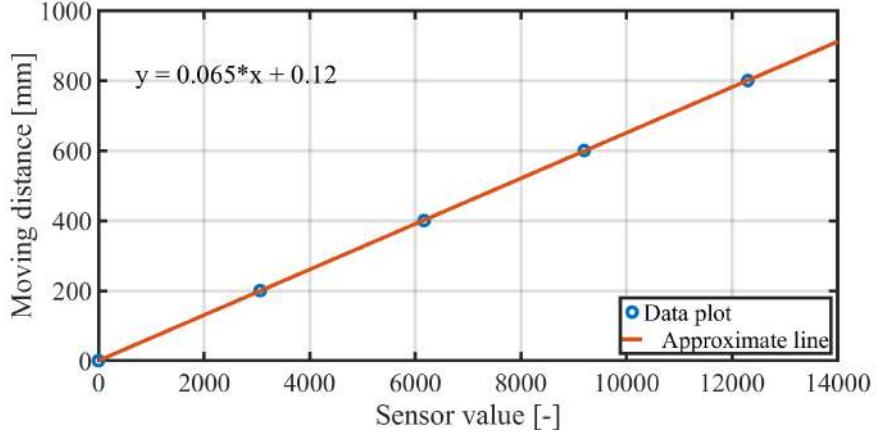


Fig. 29 Relationship between sensor value and moving distance

近似直線からわかるように、センサ値と移動距離の関係は線形的に近似可能である。センサの床からの距離、つまり式(21)中の Altitude は距離センサ得られる値を用いたが、Altitude の値を真値 (775 mm) に固定して計算しても結果は同様のものであった。また、この結果は Fig. 25 に示す  $y$  軸方向の移動の実験におけるものであったが、同様に  $x$  軸方向における実験も行ったが結果は同様のものであった。したがって、各軸における Scalar の値は同一の値を採用した。以上の検証より Scalar が求まったため、オプティカルフローセンサよりセンサ座標系における速度が得られるようになった。

次にセンサ座標系のワールド座標系への変換方法について述べる。前述した方法で得られるのはセンサ座標系における速度であり、クアッドロータの位置や速度をワールド座標系内で推定し、位置を制御するにはセンサ座標系をワールド座標系へ変換する必要がある。まず、Fig. 30 にワールド座標系と機体座標系の関係を示す。

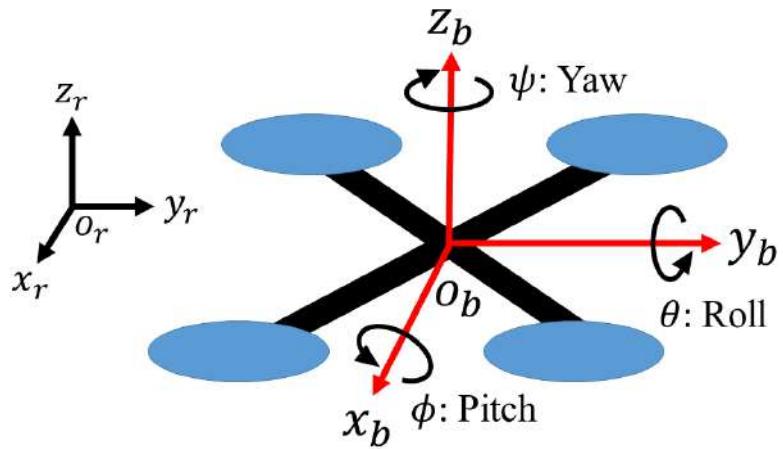


Fig. 30 Relationship between the world coordinate system and the quad-rotor coordinate system

ワールド座標系は地上の任意の点を原点とする地上に固定されている座標系であり、 $r$ -frame と表す。そして、機体座標系は機体重心を原点とし、機体前方を  $y_b$ 、右方向を  $x_b$ 、上方向を  $z_b$ とした座標系であり、 $b$ -frame と表す。また、3 次元の姿勢表現としてオイラー角を用いた表現

を行う。r-frame と b-frame の回転角の関係は  $\phi, \theta, \psi$  の 3 つのオイラー角を用いて表現できる。 $\phi, \theta, \psi$  はそれぞれ  $x, y, z$  軸周りにおける回転であり、ピッチ角、ロール角、ヨー角と呼ばれる。もし、機体がワールド座標系内にて  $z$  軸周りに  $\psi$  だけ回転した場合、オプティカルフローセンサから得られる速度をワールド座標系へ変換しなければ、ワールド座標系内での速度が得られない。オプティカルフローセンサから得られる  $x, y$  軸それぞれの速度を  $\mathbf{v}_o = [v_{ox}, v_{oy}]^T$  とし、変換した後のワールド座標系内での速度を  $\mathbf{v}_r = [v_{rx}, v_{ry}]^T$  とすると、それらの関係は  $z$  軸周りにおける回転行列を用いて次式により表せる。

$$\begin{bmatrix} v_{rx} \\ v_{ry} \end{bmatrix} = \begin{bmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{bmatrix} \begin{bmatrix} v_{ox} \\ v_{oy} \end{bmatrix} \\ = \begin{bmatrix} v_{ox} \cos \psi - v_{oy} \sin \psi \\ v_{ox} \sin \psi + v_{oy} \cos \psi \end{bmatrix} \quad (21)$$

$z$  軸周りの回転角度は地磁気及び角速度センサを使用して推定したもの用いる。推定方法は次章にて説明する。式(21)を用いることでクアッドロータのワールド座標系内での速度をオプティカルフローセンサにより得られるようになった。

## 2.4 UWB モジュール, IMU, 距離センサのみを用いた位置推定システム

まず、先行研究において開発され、本研究において改良したクアッドロータの外観を Fig. 31 に示す。また詳細な仕様を Table 4 に示す。

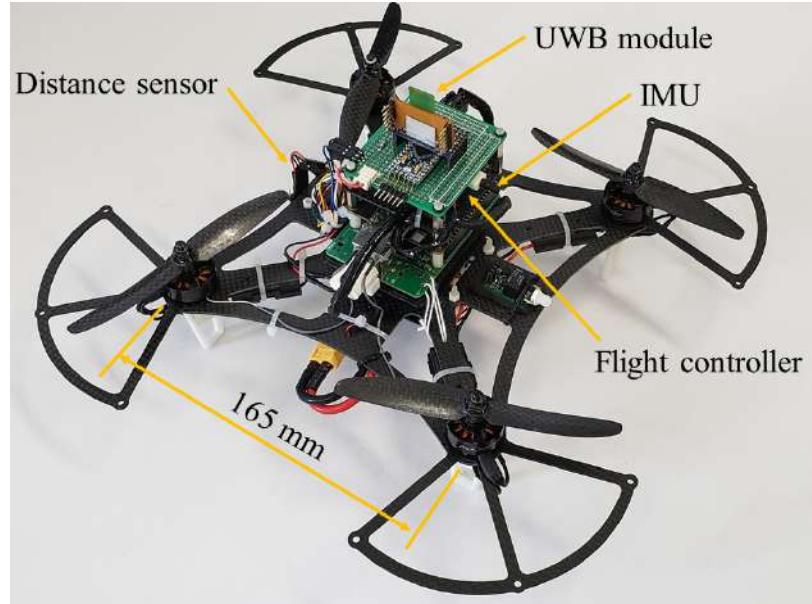


Fig. 31 Quad-rotor equipped with UWB module, IMU and distance sensor

Table 3 Quad-rotor specifications.

|                           |      |
|---------------------------|------|
| Total weight [kg]         | 0.65 |
| Height [mm]               | 110  |
| Width [mm]                | 250  |
| Depth [mm]                | 250  |
| Propeller size [inch]     | 5×3  |
| Maximum flight time [min] | 10   |
| KV value [rpm/V]          | 2633 |
| Buttery capacity[mAh]     | 1800 |

先行研究からのクアッドロータの変更点は新たにライトコントローラを追加した点である。次の Fig. 32 に先行研究にて開発された位置推定アルゴリズムを用いた位置制御用のシステムを示す。

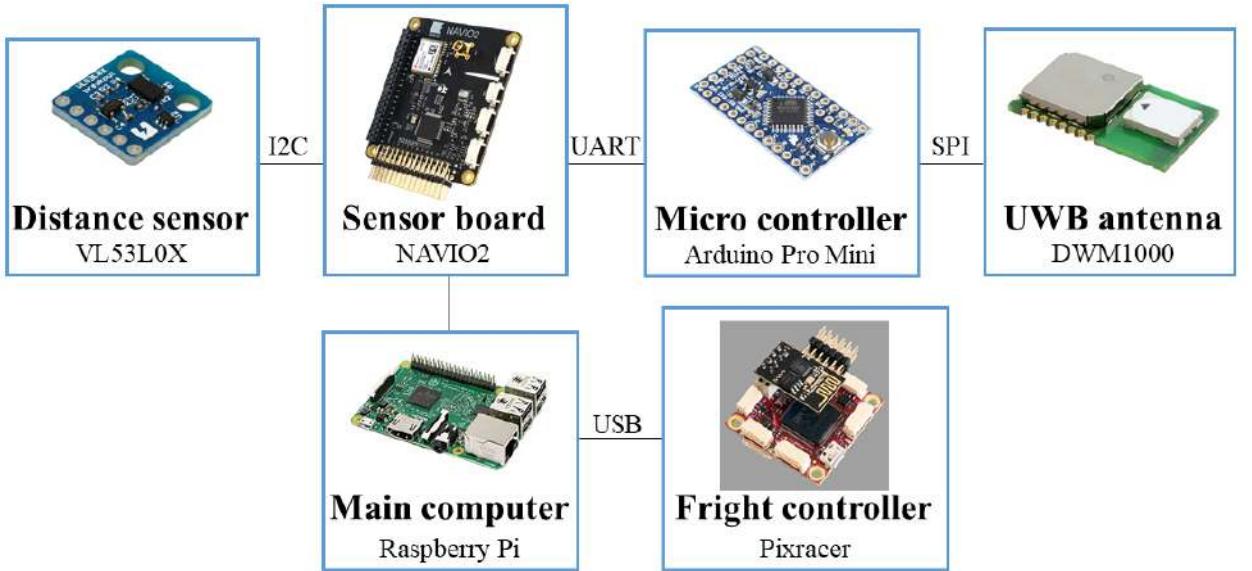


Fig. 32 System configuration of sensors

本クアッドロータでは姿勢角や位置などの推定値の計算や位置制御のための姿勢角の計算などは全て Linux OS 搭載のコンピュータ, Raspberry Pi を用いて行う。センサボードは NAVIO2 を搭載しており、加速度センサ及び角速度センサの機能を有している。UWB モジュールは SPI 通信で Arduino Pro Mini と通信しており、4つのアンカから集めた距離データを集約し UART 通信で Raspberry Pi に送っている。距離センサの値は I<sup>2</sup>C 通信を用いて Raspberry Pi に送っている。先行研究においては、推定した自己位置や姿勢を基に姿勢角を制御する際、ライトコントローラとして Raspberry Pi を用い、Integral Backstepping 制御を適用することでクアッドロータの位置制御を行っていた。しかし、その制御方法では姿勢角制御のループ周波数が遅く、位置制御が振動的になり目標の位置にクアッドロータを収束させることができなかった。そ

こで、本研究ではその姿勢角制御の遅さを補うため、推定値の演算を行うコンピュータとクアッドロータの姿勢角制御を行うコンピュータを分離させた。推定値の演算や位置の偏差から目標姿勢角を計算することなどは従来通り Raspberry Pi を用いて行うが、姿勢角制御は mRobotics 社が製作している Pixracer と呼ばれるライトコントローラを用いて行うように仕様を変更した。Fig. 33 に Pixracer の外観を示す。

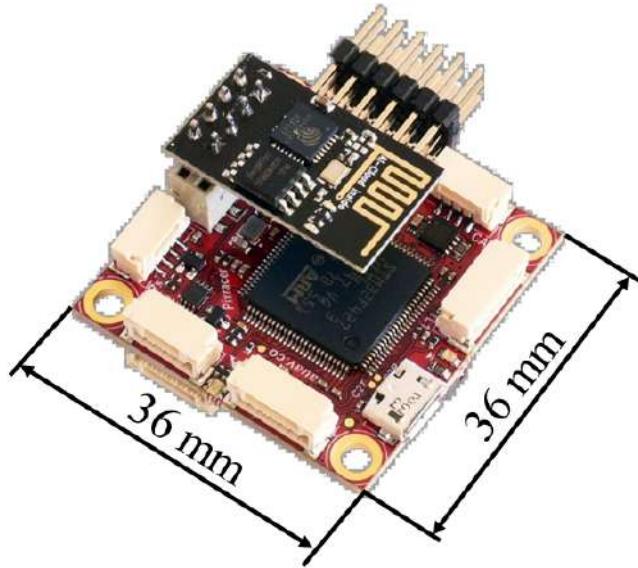


Fig. 33 Appearance of Pixracer

このボードは 6 個の PWM 出力ピンを備えており、重量も約 11 g と軽量のため、小型の航空機や地上機などのコントローラとして採用されることが多い。加速度センサ、角速度センサ、地磁気センサ、気圧センサといった各種センサや Wi-Fi モジュールも搭載されるため、汎用性が高いコントローラでもある。また、GPS も搭載されており、GPS 環境下であればオートフライトも可能である。本研究にてこのコントローラを採用した理由は非常に小型・軽量であるため、従来のクアッドロータに追加で搭載した場合にも、大きな改良を施さなくて良い点やファームウェアやプログラムがオープンソースなため、自分で内部の仕様を変更可能であることなどが理由である。また、何よりも大きな利点は高速な CPU を搭載し、完成度の高い姿勢角制御のアルゴリズムが使用可能なため、自分で構築した姿勢角制御アルゴリズムよりも精度良くクアッドロータの姿勢角を制御可能であることである。Fig. 32 に示すようにライトコントローラは USB を介して Raspberry Pi と接続されており、Raspberry Pi が計算した目標姿勢角をライトコントローラに送ることで、クアッドロータを目標姿勢に追従させることが可能である。Fig. 34 に位置制御のブロック線図を示す。

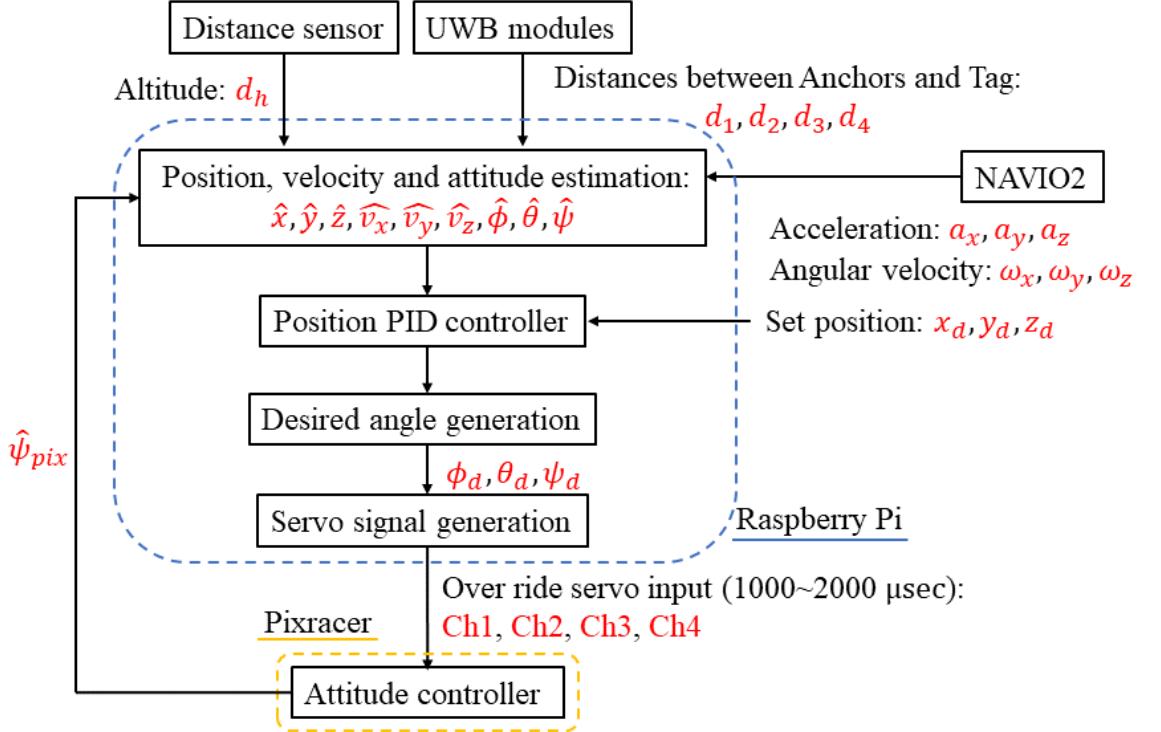


Fig. 34 Block diagram of positon control system

手順としてはまず、UWB モジュールのタグと各アンカとの距離 ( $d_1$  から  $d_4$ ), 距離センサから得た地面との距離 ( $d_h$ ), センサボードから得た加速度 ( $a_x, a_y, a_z$ ), 角速度値 ( $\omega_x, \omega_y, \omega_z$ ) をミキシングし, 拡張カルマンフィルタを適用することで位置 ( $x, y, z$ ), 速度 ( $v_x, v_y, v_z$ ), 姿勢角 ( $\phi, \theta, \psi$ ) を推定する. そして, 目標座標と現在座標の偏差を PID 制御器に通すことで目標姿勢角 ( $\phi_d, \theta_d, \psi_d$ ) を求める. そしてそれをフライトコントローラへ送信可能なサーボ信号 (Ch1 から Ch4) へと変換し, フライトコントローラへ送ることでクアッドロータの姿勢角及びスラストが制御され, 目標位置へと機体が移動するという流れである.

次に拡張カルマンフィルタを用いてクアッドロータの位置を推定する方法について述べる. まず, 姿勢角や位置を推定するのに拡張カルマンフィルタを用いる理由について述べる. 初めに姿勢角についてであるが, 姿勢角はカルマンフィルタの状態方程式側にて角速度センサの値を積分して角度に変換する. しかし, 積分した際, 角速度センサに乗ったノイズまで積分するため, 推定した角度が時間とともにドリフトしてしまう. また, 観測方程式側では加速度センサから得られる値よりヨー角以外の姿勢角を直接計算できる. その際, 積分を行わないためドリフトをすることはないが, クアッドロータが並進移動した際に余計な加速度がかかるため, 正確な角度を計算できなくなってしまう. それら二種類のセンサの不利な点を補うのに今回のようないかんカルマンフィルタを用いた姿勢角の推定は有効である. 次に位置推定についてであるが, 位置推定の計算は状態方程式側において加速度センサの値を二回積分し, 移動距離を求め, 観測方程式側では UWB モジュールを用いて求めた距離を用い, マルチラテレーションの手法にて位置を推定している. 加速度センサからは数千 Hz にて加速度を得られるため, 位置推定値も変化に対する応答性が高いが, 姿勢角を求めた際と同様に, 積分した際にドリフトが生じてしまう. また, UWB を用いた推定では正確な絶対座標が得られるが, UWB アンカを 4 つ使用し

た際の更新周波数が 10 Hz と遅いため、クアッドロータが高速で運動した際には応答性の悪い推定になってしまふ。このように位置推定においても、お互いのセンサ不利な点を補えるためカルマンフィルタは有効である。UWB モジュールを用いてクアッドロータの位置を推定する方法にマルチラテレーションの手法を用いていると述べたが、これについて説明する。マルチラテレーションとは座標が既知の 3 つ以上の固定局と移動局間の距離を求め、アルゴリズムに基づいて計算することで、移動局の座標を求める手法のことである。身近な例では、複数個の GPS 衛星と GPS 移動局との距離よりモジュールの位置を割り出すのに使用されている。位置を計算する手法は数多く存在し、[1] や [2] に詳しい解法が載っている。Fig. 35 にマルチラテレーションにおける模式図を示す。

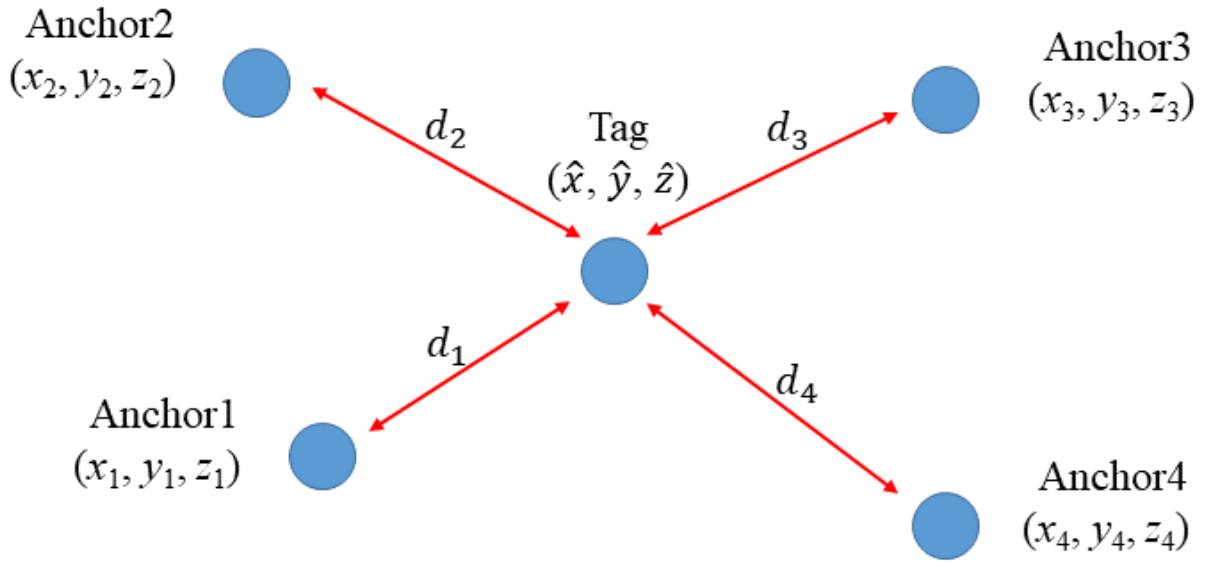


Fig. 35 Position relationship between Anchors and Tag

アンカが固定局のことであり、その絶対座標は既知である。そして、タグは座標未知の移動局であり、クアッドロータに搭載されている。このタグの位置を求めることがマルチラテレーションのことである。タグ及びアンカの座標とモジュール間の距離の関係は次式により表せる。

$$\begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix} = \begin{bmatrix} \sqrt{(\hat{x} - x_1)^2 + (\hat{y} - y_1)^2 + (\hat{z} - z_1)^2} \\ \sqrt{(\hat{x} - x_2)^2 + (\hat{y} - y_2)^2 + (\hat{z} - z_2)^2} \\ \sqrt{(\hat{x} - x_3)^2 + (\hat{y} - y_3)^2 + (\hat{z} - z_3)^2} \\ \sqrt{(\hat{x} - x_4)^2 + (\hat{y} - y_4)^2 + (\hat{z} - z_4)^2} \end{bmatrix} \quad (22)$$

方程式 (22) を  $\hat{x}, \hat{y}, \hat{z}$  について解くことによりタグの三次元位置座標を求めることが可能である。なお、この式は後に出てくる拡張カルマンフィルタの観測方程式の一部である。次に拡張カルマンフィルタにおける状態方程式を見ていく。機体の速度ベクトル  $\mathbf{v}_r = [v_x, v_y, v_z]^T$  と機体の位置ベクトル  $\mathbf{p}_r = [x_r, y_r, z_r]^T$  の関係は次式で表せる。

$$\frac{d}{dt} \mathbf{p}_r = \mathbf{v}_r \quad (23)$$

速度ベクトルの時間微分  $\frac{d}{dt}\mathbf{v}_r$  は加速度センサより得られるセンサ値  $\mathbf{a}_b = [a_x, a_y, a_z]^T$  と回転行列  $\mathbf{R}$  を用いて次式で表せる。

$$\frac{d}{dt}\mathbf{v}_r = \mathbf{R}\mathbf{a}_b - \mathbf{g}_r \quad (24)$$

ここで、ワールド座標系において重力は  $z$  軸負方向に重力加速度:  $g$  分だけ作用するため、 $\mathbf{g}_r$  は  $\mathbf{g}_r = [0, 0, -g]^T$  と定義できる。また、Fig. 30 における r-frame と b-frame 間の回転行列  $\mathbf{R}_r^b$  は次式で定義される。

$$\begin{aligned} \mathbf{R}_r^b(\phi, \theta, \psi) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \sin \phi \sin \theta \cos \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \sin \phi \cos \theta \\ \cos \phi \sin \theta \cos \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi & \cos \phi \cos \theta \end{bmatrix} \end{aligned} \quad (25)$$

そしてオイラー角  $\Theta = [\phi, \theta, \psi]^T$  の時間微分と角速度センサより得られる機体角速度  $\omega = [p, q, r]^T$  の関係は次式で表せる。

$$\frac{d}{dt}\Theta = \mathbf{W}\omega \quad (26)$$

なお、 $\mathbf{W}$  は [] より

$$\mathbf{W} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi / \cos \theta & \cos \phi / \cos \theta \end{bmatrix} \quad (27)$$

である。状態ベクトルを  $\mathbf{x} = [\Theta, \mathbf{p}_r, \mathbf{v}_r]^T$  と定義し、式 (23)(24)(26) をまとめると次のようになる。

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) \quad (28)$$

ただし  $\mathbf{f}(\cdot)$  は

$$\mathbf{f}(\mathbf{x}(t)) = \begin{bmatrix} \mathbf{W}\omega \\ \mathbf{v}_r \\ \mathbf{R}\mathbf{a}_b - \mathbf{g}_r \end{bmatrix} \quad (29)$$

角速度センサ及び加速度センサから得られる出力値に含まれる観測ノイズ  $\delta\omega = [\delta p, \delta q, \delta r]^T$  及び  $\delta\mathbf{a}_b = [\delta a_x, \delta a_y, \delta a_z]^T$  を考慮すると式 (28) は次式のようになる.

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{B}\mathbf{v} \quad (30)$$

ただし,  $\mathbf{B}, \mathbf{v}$  はそれぞれ

$$\mathbf{B} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \\ \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix}^T, \quad \mathbf{v} = [\delta p, \delta q, \delta r, \delta a_x, \delta a_y, \delta a_z]^T \quad (31)$$

である. さらにオイラー法を用いて式 (30) を離散化すると

$$\mathbf{x}(t+1) = \mathbf{f}_t(\mathbf{x}(t)) + \mathbf{B}_t\mathbf{v} \quad (32)$$

ただし

$$\mathbf{f}_t(\mathbf{x}(t)) = \mathbf{x}(t) + \mathbf{f}(\mathbf{x}(t))\Delta t, \quad \mathbf{B}_t = \mathbf{B}\Delta t \quad (33)$$

である. ここで  $\Delta t$  はサンプリング周期である.

次に観測方程式について見ていく. 観測値は UWB モジュールから得られる距離データ, 加速度センサデータ, 角速度センサデータ, 距離センサデータ及びライトコントローラである Pixracer が推定したヨー角 ( $\psi$ ) をそれぞれ採用した. まず, 加速度センサ値を用いた角度の算出である. 加速度センサから得られるセンサ値  $\mathbf{a}_b$  と重力ベクトル  $\mathbf{g}_r$  の関係は回転行列  $\mathbf{R}$  を用いて次のようになる.

$$\begin{aligned} \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} &= \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \sin \phi \sin \theta \cos \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \sin \phi \cos \theta \\ \cos \phi \sin \theta \cos \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \\ &= \begin{bmatrix} g \sin \theta \\ -g \cos \theta \sin \phi \\ -g \cos \theta \cos \phi \end{bmatrix} \end{aligned} \quad (34)$$

式 (34) の最終項を見ると,  $\psi$  つまりヨー角成分が現れていないことがわかる. つまり加速度センサのみではヨー角を推定することはできず, そのため今回はヨー角の推定に角速度センサの値と Pixracer から得られるヨー角をミキシングしている. ここで観測行列を  $\mathbf{y}(t) = [a_x, a_y, a_z, d_1, d_2, d_3, d_4, d_h, \psi_{pix}]^T$  とおき, 式 (22), (34) と合わせると

$$\mathbf{y}(t) = \mathbf{h}_t(\mathbf{x}(t)) \quad (35)$$

ただし  $\mathbf{h}(\cdot)$  は

$$\mathbf{h}_t(\mathbf{x}(t)) = \begin{bmatrix} g \sin \theta \\ -g \cos \theta \sin \phi \\ -g \cos \theta \cos \phi \\ \sqrt{(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2} \\ \sqrt{(x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2} \\ \sqrt{(x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2} \\ \sqrt{(x - x_4)^2 + (y - y_4)^2 + (z - z_4)^2} \\ z \\ \psi \end{bmatrix} \quad (36)$$

である。なお  $d_h$  は距離センサから得られた地面までの距離,  $\psi_{pix}$  は Pixracer が推定したヨー角の値である。式 (36) に加速度センサ, UWB モジュール, 距離センサ, Pixracer からの出力に含まれるノイズ  $[\delta a_x, \delta a_y, \delta a_z]^T$ ,  $[\delta d_1, \delta d_2, \delta d_3, \delta d_4]^T$ ,  $\delta d_h$ ,  $\delta \psi_{pix}$  を考慮すると観測方程式は

$$\mathbf{y}(t) = \mathbf{h}_t(\mathbf{x}(t)) + \mathbf{w} \quad (37)$$

となる。ただし  $\mathbf{w}$  は

$$\mathbf{w} = [\delta a_x, \delta a_y, \delta a_z, \delta d_1, \delta d_2, \delta d_3, \delta d_4, \delta d_h, \delta \psi_{pix}]^T \quad (38)$$

以上より状態方程式 (32) 及び観測方程式 (37) が導出できた。この離散プロセスモデルを 2 章第 1 節にて示した拡張カルマンフィルタに対して適用することで位置推定アルゴリズムを得ることができる。

## 2.5 従来の構成にオプティカルフローセンサも加えた位置推定システム（複数センサの統合環境）

ここではオプティカルフローセンサの値も推定アルゴリズムに組み込んだ手法について説明する。次の Fig. 36 に前のセンサ構成 (Fig. 32) にオプティカルフローセンサボードも加えた構成及び Fig. 37 にオプティカルフローセンサも組み込んだブロック線図を示す。

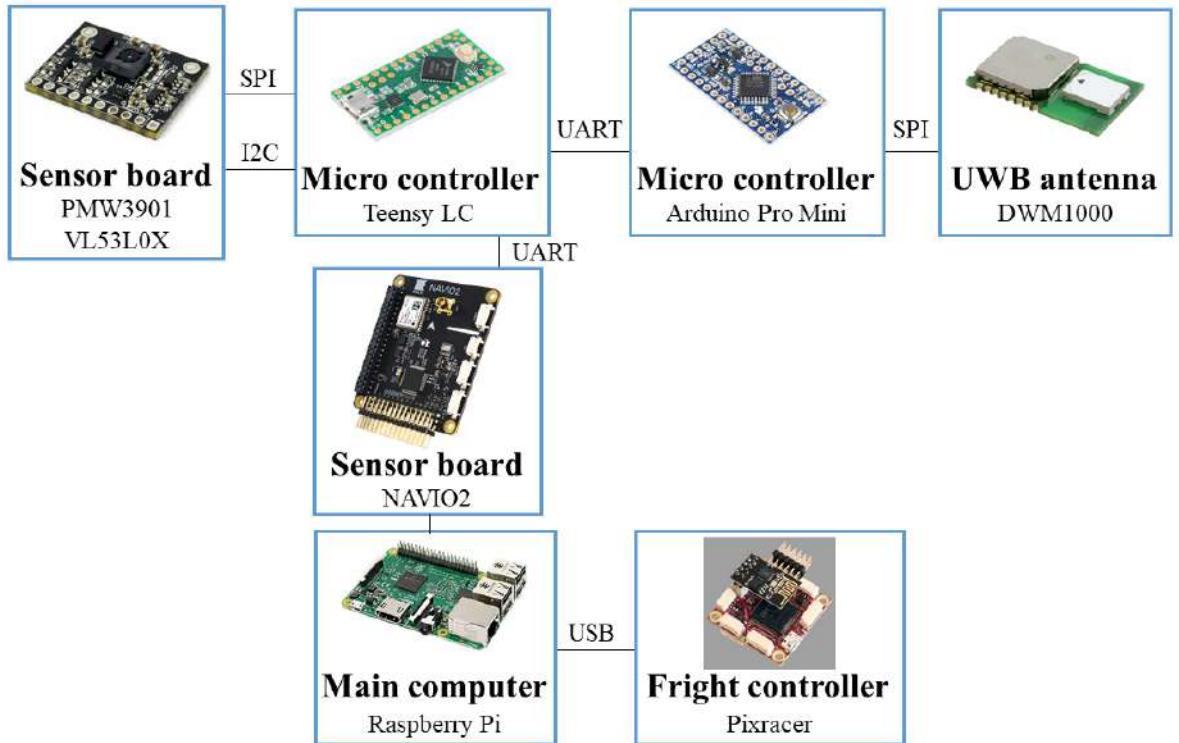


Fig. 36 Configuration of sensors with optical flow sensor board

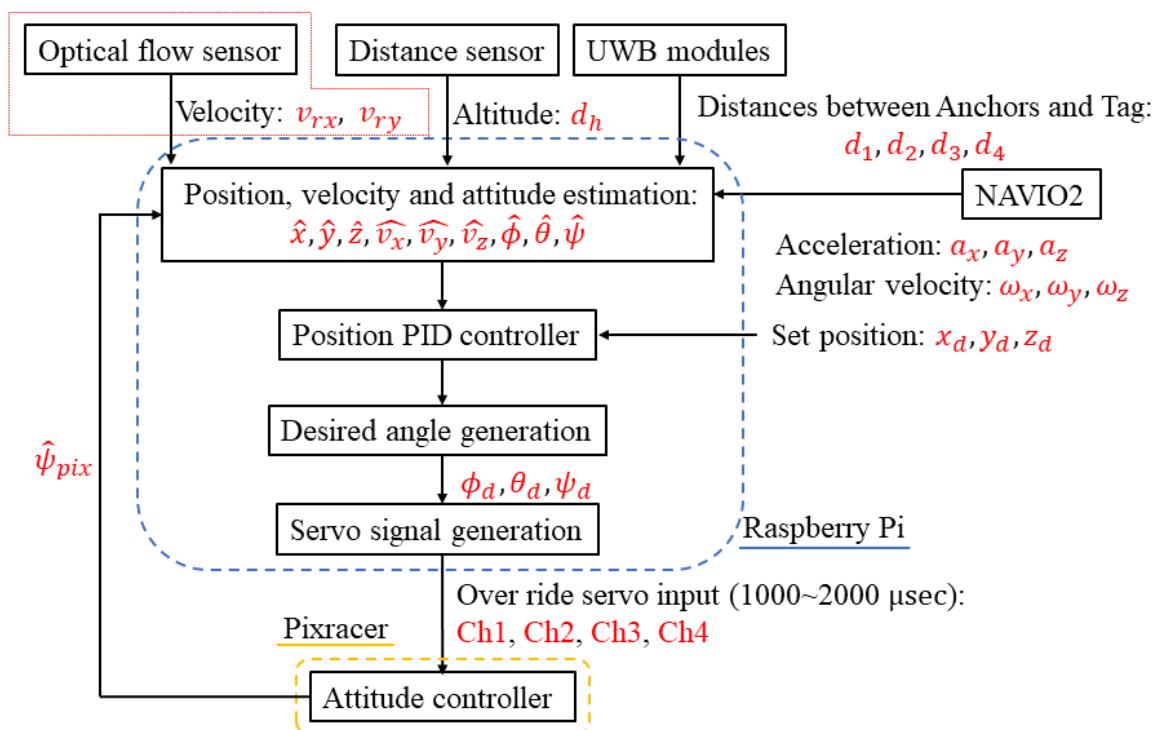


Fig. 37 Block diagram of positon control system

図からわかるようにオプティカルフローセンサボードとマイコン（Teensy LC）が新たに加わった

た。ここで、なぜマイコンを二つ（Arduino Pro Mini, Teensy LC）用いているかの理由を説明する。UWB モジュール（DWM1000）及びオプティカルフローセンサ（PMW3901）は SPI 通信を用いてマイコンと通信する。SPI 通信はマスター側とスレーブ側が 4 線で通信する通信規格であり、スレーブの数が複数あっても 1 つのマイコンでそれらを処理することが可能である。したがって、Fig. 36 に示す Arduino Pro Mini 1 つで 3 つのセンサ（PMW3901, VL53L0X, DWM1000）の値を処理することを考えた。しかし、実際は PMW3901 と DWM1000 の 2 つのセンサ間の相性が悪く、同時に 2 つのセンサの値を取得することが不可能であった。したがって、オプティカルフローセンサボードの値を処理するマイコン（Teensy LC）と UWB モジュールの値を処理するマイコン（Arduino Pro Mini）を分離し、別々に値を取得してから 1 つのマイコン（Teensy LC）に値をまとめ、UART 通信で Raspberry Pi に送る構成を組んだ。また、Teensy LC であるが、このマイコンは Arduino Pro Mini と異なり、1 つのマイコンで複数の UART 通信を行うことが可能であり、図のように Arduino Pro Mini 及び NAVIO2 の 2 種類のボードと通信を行う必要があったためこのマイコンを採用するに至った。

次にオプティカルフローセンサの値も拡張カルマンフィルタを組み込んだ式を示していく。このアルゴリズムにおいても状態方程式は式(32)と同じものを用いる。つまり、状態方程式側にて、加速度センサの積分値を用いてクアッドロータの変位を推定し、角速度センサの値を用いて姿勢角を推定する。観測方程式であるが加速度センサ値、UWB モジュールから得た値、距離センサ値、Pixracer から得た値を用いることは変わらず、そこにオプティカルフローセンサから得た速度値が加わった。つまり観測行列は  $\mathbf{y}(t) = [a_x, a_y, a_z, d_1, d_2, d_3, d_4, d_h, \psi_{pix}, v_{rx}, v_{ry}]^T$  と表される。ここで  $v_{rx}$  と  $v_{ry}$  は式(21)で表されるヨー角回転を補正した値である。そして、観測方程式は

$$\mathbf{y}(t) = \mathbf{h}_t(\mathbf{x}(t)) \quad (39)$$

となり、 $\mathbf{h}(\cdot)$  は

$$\mathbf{h}_t(\mathbf{x}(t)) = \begin{bmatrix} g \sin \theta \\ -g \cos \theta \sin \phi \\ -g \cos \theta \cos \phi \\ \sqrt{(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2} \\ \sqrt{(x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2} \\ \sqrt{(x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2} \\ \sqrt{(x - x_4)^2 + (y - y_4)^2 + (z - z_4)^2} \\ z \\ \psi \\ v_x \\ v_y \end{bmatrix} \quad (40)$$

である。前と同様に各種センサから得られるノイズを考慮すると観測方程式は

$$\mathbf{y}(t) = \mathbf{h}_t(\mathbf{x}(t)) + \mathbf{w} \quad (41)$$

となり、 $\mathbf{w}$  は

$$\mathbf{w} = [\delta a_x, \delta a_y, \delta a_z, \delta d_1, \delta d_2, \delta d_3, \delta d_4, \delta d_h, \delta \psi_{pix}, \delta v_{rx}, \delta v_{ry}]^T \quad (42)$$

である。以上より新たな構成においても観測方程式が導出できた。これら状態方程式、観測方程式を前節と同様に拡張カルマンフィルタに適用することでオプティカルフローセンサも組み込んだ位置推定アルゴリズムを得ることができる。

### 3. 位置制御用クアッドロータの開発

2.4章で示したクアッドロータは先行研究から利用されてきたものであり、本研究から新たにライトコントローラ（Pixracer）やオプティカルフローセンサを搭載するに当たって積載物の重量がかさみ、推力不足が懸念された。そして、クアッドロータの推力に余裕が生まれると、機体の応答性が向上し、位置制御の精度が高くなると考えた。そこで、推力の高い新たなクアッドロータのシステムを構築した。その新たに構築したクアッドロータを Fig. 38, Fig. 39 に示す。

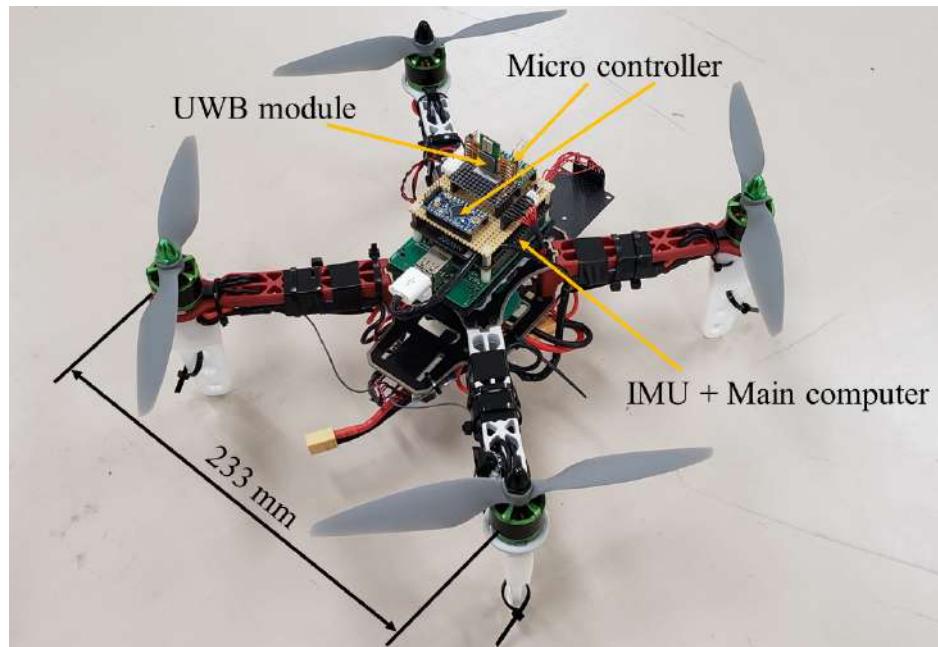


Fig. 38 Front side of quad-rotor

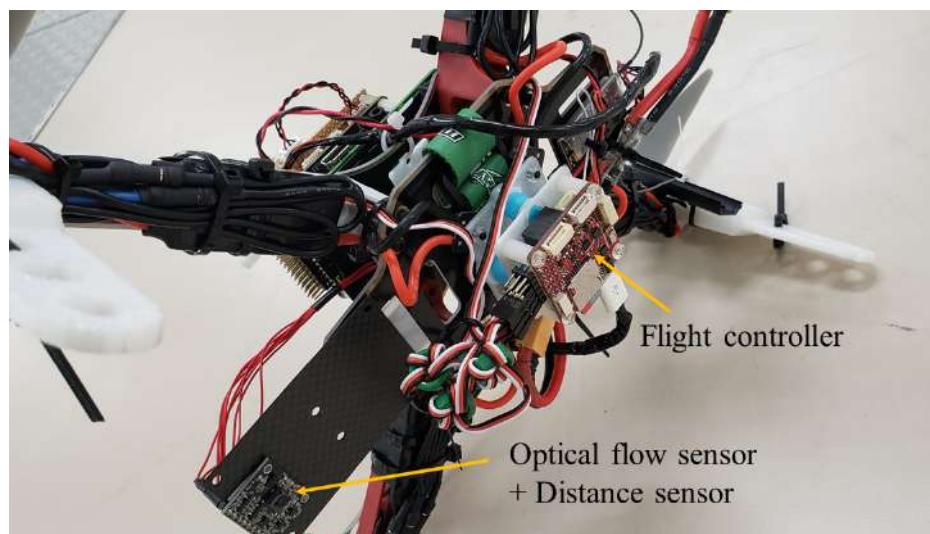


Fig. 39 Back side Of quad-rotor

図からわかるようにクアッドロータ上面にマイコンやメインコンピュータである Raspberry Pi を配置し、下面にライトコントローラやオプティカルフローセンサボードを配置した。先行研究から使用してきたクアッドロータの場合にはライトコントローラを上面に配置していたために重心が上がってしまい、不安定であった。したがって新たに構築したこのクアッドロータは下側にコントローラを配置し、重心が上がらないように配慮した。そして、次の表に詳細な仕様を示す。

Table 4 Quad-rotor specifications.

|                           |       |
|---------------------------|-------|
| Total weight [kg]         | 1.05  |
| Height [mm]               | 200   |
| Width [mm]                | 443   |
| Depth [mm]                | 443   |
| Propeller size [inch]     | 8×4.5 |
| Maximum flight time [min] | 10    |
| KV value [rpm/V]          | 920   |
| Buttery capacity[mAh]     | 2650  |

モータやバッテリなど飛行に必要な部品以外に追加で数百 g ほど搭載可能なスラストが出せるように部品選定を行った。モータの電圧やプロペラの種類にもよるが、最高で約 200 g の追加スラストを生み出すことが可能な設計である。また、総飛行時間は約 10 分であるが、高いパフォーマンス状態で飛行可能な時間は約 5 分である。

## 4. 位置推定アルゴリズムの精度評価

この章では前章にて説明した位置推定アルゴリズムを用いクアッドロータの位置を推定し、その精度の評価を行う。まず位置推定実験に使用した環境を次の図に示す。

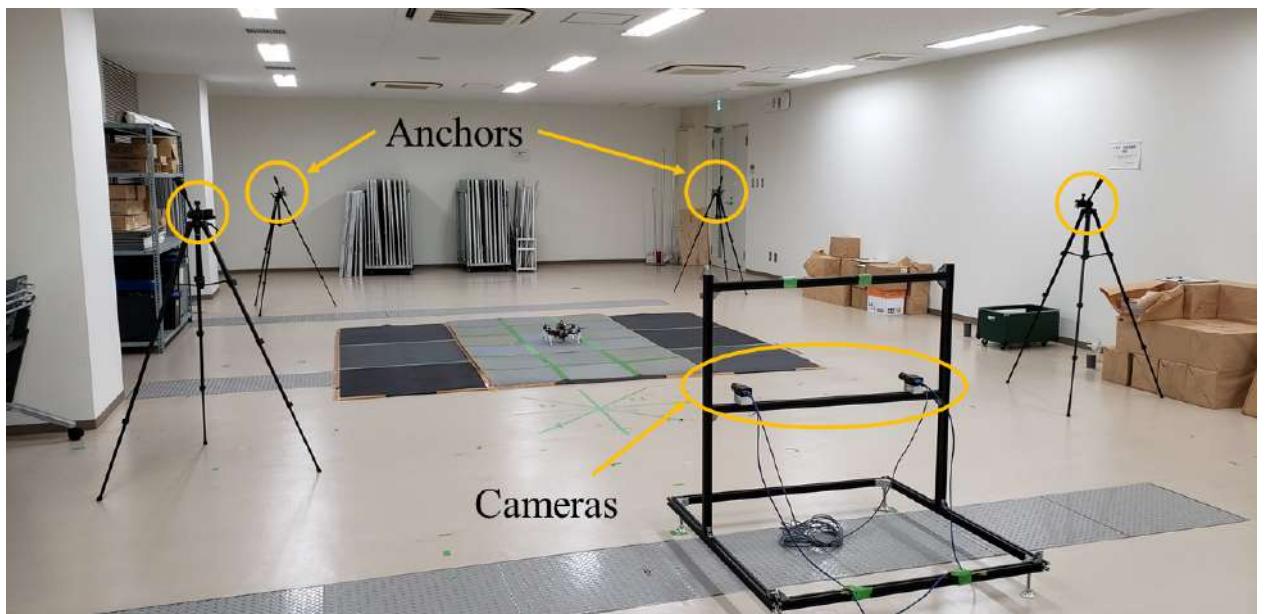


Fig. 40 Laboratory used for position estimation and position control experiments

クアッドロータの位置を推定する領域を取り囲むように4つのUWBアンカを設置した。アンカは次の図に示すようにカメラ用の三脚の頂点に固定した。



Fig. 41 Laboratory used for position estimation and position control experiments

また、アンカを設置した絶対座標を次のTable 5に示す。

Table 5 Absolute position of target UWB anchors

|         | $x$ position [m] | $y$ position [m] | $z$ position [m] |
|---------|------------------|------------------|------------------|
| Anchor1 | -5.460           | 2.698            | 1.763            |
| Anchor2 | 5.460            | 2.698            | 1.763            |
| Anchor3 | 5.460            | -2.698           | 1.763            |
| Anchor4 | -5.460           | -2.698           | 1.763            |

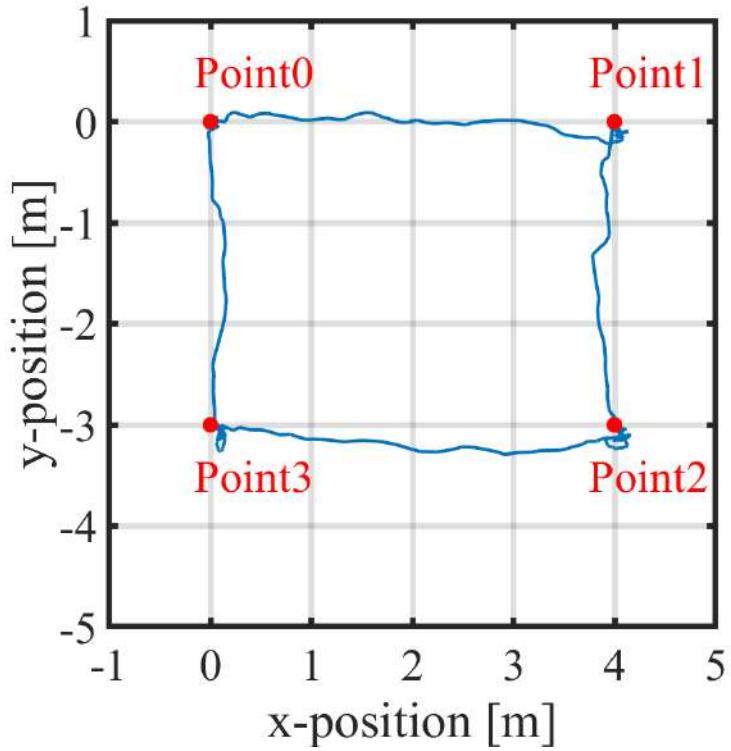


Fig. 42 Laboratory used for position estimation and position control experiments

Table 6 Absolute position of target points

|                 | $x$ position [m] | $y$ position [m] |
|-----------------|------------------|------------------|
| Point0 (Origin) | 0.00             | 0.00             |
| Point1          | 4.00             | 0.00             |
| Point2          | 4.00             | -3.00            |
| Point3          | 0.00             | -3.00            |

#### 4.1 カルマンフィルタの分散が精度に与える影響の考察

### 5. 位置制御アルゴリズムの評価

前章までは各種センサを用いたクアッドロータの位置推定方法について述べてきた。この説では実際にクアッドロータを飛行させ、提案する位置制御手法の精度を見ていく。

#### 5.1 位置制御手法

まず、この節では Fig. 34 及び Fig. 37 にある目標座標と推定座標の偏差からクアッドロータの位置を制御する手法について述べる。これから述べる手法は [1][2] などの手法を参考にした。まず、推定した座標と目標座標との偏差より仮想入力の値を次のように設定する。

$$U_x = k_{px}(x_d - \hat{x}) + k_{dx}(\dot{x}_d - \dot{\hat{x}}) + k_{ix} \int_0^t (x_d - \hat{x}) d\tau \quad (43)$$

$$U_y = k_{py}(y_d - \hat{y}) + k_{dy}(\dot{y}_d - \dot{\hat{y}}) + k_{iy} \int_0^t (y_d - \hat{y}) d\tau \quad (44)$$

$x$  軸,  $y$  軸はそれぞれ機体座標系ではなく, ワールド座標系における軸を表している. 式 (43) からわかるように仮想入力の値は偏差を用いた PID コントローラより計算できる. これらの入力値を用いると, ピッチ角 ( $\phi$ ) 及びロール角 ( $\theta$ ) の目標値は以下の式により計算できる.

$$\phi^{des} = \frac{1}{g}(U_x \sin \psi - U_y \cos \psi) \quad (45)$$

$$\theta^{des} = \frac{1}{g}(U_x \cos \psi + U_y \sin \psi). \quad (46)$$

これらの得た値をライトコントローラである Pixracer が受け取れる PWM 信号に変換した後, コントローラに送信することで, クアッドロータの各ロータの回転数が制御され, 姿勢角が追従するという流れである. またヨー角 ( $\psi$ ) に関しては, 飛行開始時に置いたクアッドロータの機首が向いている方向を目標角度とし, 推定した現在角度との偏差において単純な PID 制御器を組むことにより機首の向きが回転しないように制御を行う.

## 5.2 ステレオカメラを用いた位置測定システム

### 5.2.1 位置計測システムの原理及び構築

クアッドロータの位置を制御した場合, 位置が精度良く制御できているかを確認するため, その指標として位置の真値が必要である. 飛行体や地上移動機に関わらず, 移動型のロボットの真値を計測するには Fig. 43 に示す VICON[] などといったモーションキャプチャのシステムを用いるのが多い. しかし, こういったシステムは機材だけで数百万円の価格であり, 設置費を含むと数千万円に上ることもある. また, Fig. 44 のように一度室内に環境構築を行うと, システムを他の部屋に移動させることが困難である. したがって, 価格が VICON よりも安価であり, 必要に応じてシステムを移動させられることなどの利点があることから本論文ではステレオカメラを用いた位置の計測システムを構築した. Fig. 45 に環境構築したステレオカメラシステムを示す.



Fig. 43 VICON[]



Fig. 44 The room where VICON is installed[]

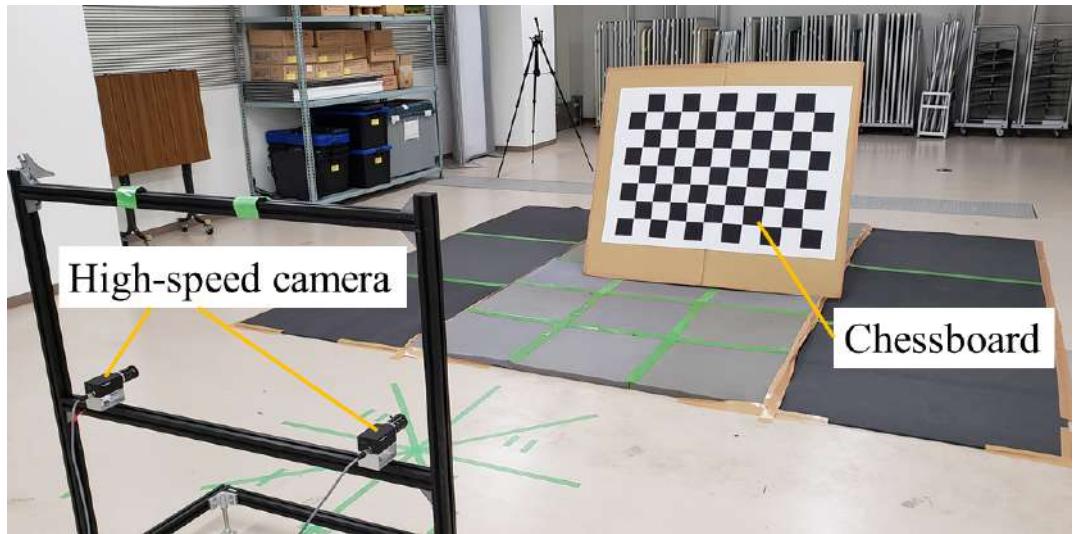


Fig. 45 Installed stereo camera and chessboard

システムの大まかな構成は二台の高速度カメラからなるステレオカメラとデスクトップPCである。使用したカメラはPoint Grey社（現FLIR社）製のGrasshopper3(GS3-U3-32S4C-C)である。Fig. 46に外観を示す。

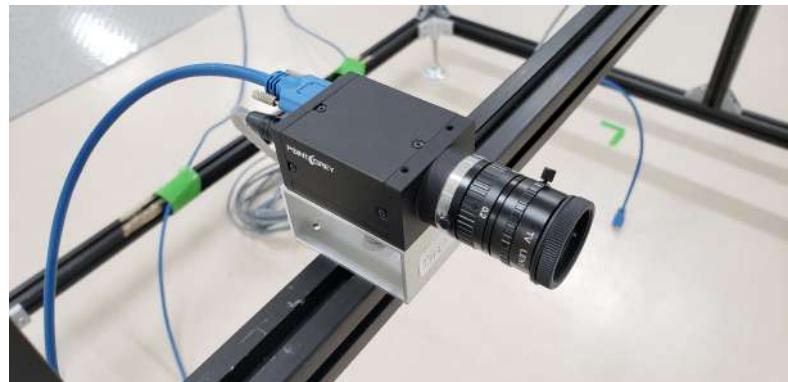


Fig. 46 High-speed camera(GS3-U3-32S4C-C)

そして詳細な仕様をTable 7に示す。

Table 7 High-speed camera specifications.

|                                 |                |
|---------------------------------|----------------|
| Maximum resolution (H×V)[pixel] | 2048×1536      |
| Maximum frame rate [fps]        | 121            |
| Image sensor                    | Sony IMX252    |
| Sensor type                     | CMOS           |
| Shutter type                    | Global shutter |
| Interface                       | USB3.1 Gen 1   |
| Weight [g]                      | 90             |

そして画像保存に使用したデスクトップPCの仕様をTable 8に示す。

Table 8 High-speed camera specifications.

|              |                        |
|--------------|------------------------|
| CPU          | intel Core i7          |
| Motherboard  | ASUS Z270F GAMING      |
| Graphic      | GTX1050Ti              |
| RAM [GB]     | 64 (DDR4)              |
| OS           | Windows 10 ×64         |
| Storage [GB] | 256 (SSD) + 2000 (HDD) |
| Software     | FlyCapture2            |

Table 7からわかるように、カメラの性能を最大限引き出した際の仕様は、解像度が $2048 \times 1536$ とフルHDを超える解像度であり、フレームレートも100 fpsを上回る121 fpsにて画像または動画を撮影可能である。しかし、Table 8に示すPCを用いてもカメラ2台を用いた状態で、 $1024 \times 768$ の解像度、100 fpsが限界であった。したがって本システムではこの環境で出せる最高解像度、最高fpsのこの設定で画像を撮影した。なお、撮影した画像のフォーマットは圧縮率75%のJPEGである。画像計測を用いた位置や角度の検出では撮影された画像中のマーカーを追跡することによりそれを実現する。VICONでは移動物体に取り付けられた反射マーカーに赤外線を照射することで反射した光の重心を求めることで位置や角度を検出する。本システムでは赤外線を照射しないため、それ自体が赤外光を放出する赤外線LEDをクアッドロータの位置検出用のマーカーとして採用した。Fig. 47, Fig. 48に二種類のクアッドロータに取り付けられた赤外線LEDを示す。



Fig. 47 Old quad rotor with infrared LED

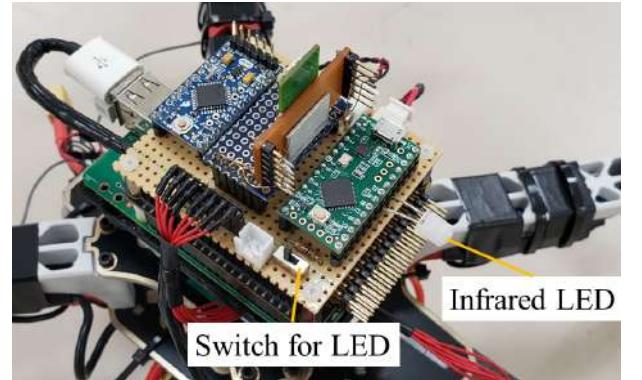


Fig. 48 New quad rotor with infrared LED

先行研究から使われてきたクアッドロータ（Fig. 47）には2台のカメラ間で同期が取れるようリレーが取り付けられている。メインコンピュータであるRaspberry Piから信号を送ることでLEDの点灯を制御可能なため、光ったタイミングを検出することで2台のカメラのマーカー検出タイミングを揃えることができる。先行研究においてはこの方法でカメラ間同期を取っていた。そして、新しく本論文にて新しく開発したクアッドロータは同期用のリレーが付いてい

ない。この理由について説明する。今回用いた高速度カメラにはPCへのデータ送信用のコネクタ以外に外部へ信号を送ることや受け取ることが可能なコネクタが備わっている。これを用い、片方のプライマリカメラからもう片方のセカンダリカメラへと同期用の信号を送り、その信号をトリガにしてセカンダリカメラは撮影を開始できるため、2台間の同期が取れた状態で画像を撮影可能である。本論文では新たにこの同期方法を採用したため、Fig. ??のクアッドロータにはリレーが備わっていない。クアッドロータに搭載したLEDマーカーの光はカメラで撮影可能であるが、画像中でLEDの光のみを浮き上がらせて検出が容易になるようカメラにはFig. 3に示す赤外線透過フィルタ（IR-80）を装着してある。



Fig. 49 Infrared longpass filter

赤外線LEDを光らせた状態でこのレンズにより撮影した画像を次に示す。

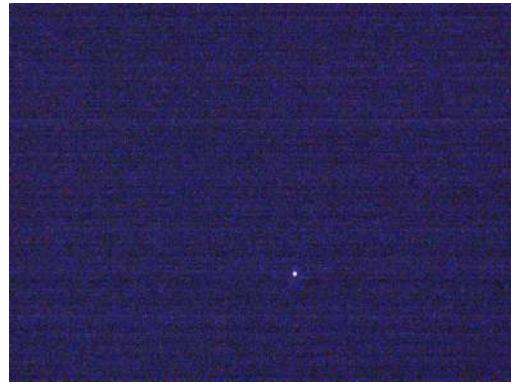


Fig. 50 Image taken of glowing infrared LED through filter

フィルタを通すことにより、LEDの光のみを浮き上がらせられているのがわかる。次にステレオカメラを用いた位置計測のアルゴリズムについて簡単に説明する。ここで説明する手法は[]や[]に詳しい方法が載っている。Fig. 51に示すように左右に並んだ2台のカメラで同じ対象物を撮影したとする。

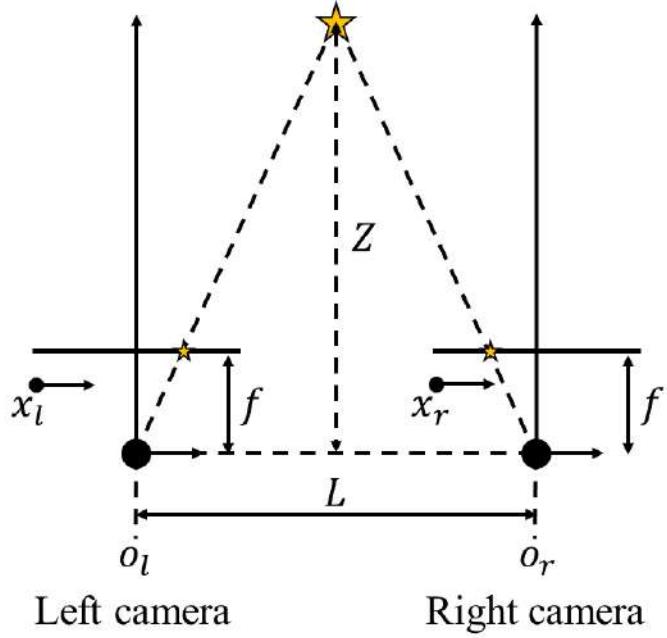


Fig. 51 Definition of stereo camera coordinates

ここでは2台のカメラの画像平面が同一平面上にあり、光軸が平行であり、同じ焦点距離  $f$  を持つとする。また、カメラの中心間距離を  $L$  とする。その位置にカメラを設置した状態で同じ対象物を撮影すると、左側のカメラでは  $x_l$ 、右側のカメラでは  $x_r$  の座標の画像平面にそれぞれ画像が投影される。この座標の差  $d$  が視差と呼ばれ、

$$d = x_l - x_r \quad (47)$$

と表せる。そして、図において三角形の相似の関係を用いると

$$\frac{L-d}{Z-f} = \frac{L-(x_l-x_r)}{Z-f} = \frac{L}{Z} \quad (48)$$

と表わせ、この式を変形することで

$$Z = \frac{fL}{x_l - x_r} \quad (49)$$

となり、対象物までの距離  $Z$  を求めることができる。同様に  $x$  軸方向  $y$  軸方向についても

$$X = \frac{x_l L}{x_l - x_r}, \quad Y = \frac{y_l L}{x_l - x_r} = \frac{y_l L}{x_l - x_r} \quad (50)$$

となり、対象物の位置を求められる。以上よりステレオカメラを用いて対象物の位置  $(X, Y, Z)$  を求める方法が導出できた。しかし、実際には2台のカメラの画像平面が同一平面になるよう

置くことや光軸が平行になるように置くことは物理的に不可能である。そこで、行うのがステレオカメラのキャリブレーションである。Fig. 45に示したチェスボードを2台のカメラで撮影し、アルゴリズムにしたがって計算することで、カメラ間の位置関係を表す並進ベクトル $T$ 及び回転行列 $R$ を算出可能である。それらに加え、カメラレンズの歪みやレンズの取付誤差が起因する画像の歪みを補正するためのカメラパラメータも同様のチェスボードを撮影した画像から得ることができる。これらのパラメータを用い、先に示した方法で計算を行うことでクアッドロータの位置を測定した。なお、これらの演算は全てPython2のモジュールとして提供されるモジュール、OpenCVの機能を用いて行った。

### 5.2.2 奥行き誤差の検証

ここで、ステレオカメラを用いた位置計測における奥行き誤差について議論する。Fig. 52に設定する状況を考える。ステレオ画像での計測精度は画像中の1画素が実際にどのくらいの長さに対応するかで決まる。つまり、カメラの解像度、対象物までの距離、カメラ間距離によって変わる。また、画像平面中の画像が投影される画素子は、それ自体が大きさを持っているため、実際の空間中では計測対象点はある範囲内ということしか特定できない。その特定可能な最小の範囲がステレオカメラにおける計測誤差となる。

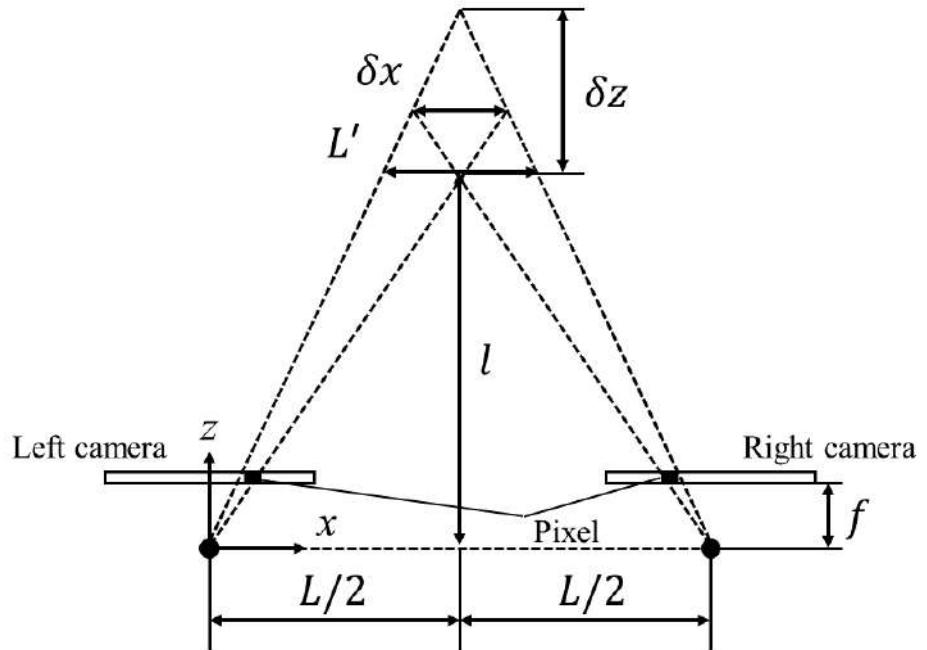


Fig. 52 Error in z direction of stereo camera

図における $\delta x$ が横方向誤差であり、 $\delta z$ が奥行き誤差である。また、図に示す $L'$ を計算のため導入する。まず、三角形の相似より

$$\text{Pixel} : \frac{L'}{2} = f : l \quad (51)$$

が言える。これを変形し

$$L' = \frac{2\text{Pixel}}{f} \quad (52)$$

となる。同様に相似より

$$\delta z = \frac{L'l}{L - L'} \quad (53)$$

が言える。式(52)を式(53)に代入し、整理すると

$$\delta z = \frac{2l^2\text{Pixel}}{fL - 2l\text{Pixel}} \quad (54)$$

となり、奥行き誤差が導出できた。同様に  $x$  方向、 $y$  方向誤差も相似より計算できるが、 $\delta z$  に比べて極めて小さく、無視できるため割愛する。奥行き誤差  $\delta z$  が 100 mm 以下になるように次の表のパラメータを用いて  $l$  を計算すると、 $l = 8.959$  m という結果を得たため、カメラから飛行するクアッドロータまでの距離が 8 m 以下になるように設定して実験を行うこととする。

### 5.3 ロータの推力測定

各ロータの推力に差がないかを確認するために実験を行った。実験方法は飛行実験に用いた4つのロータをそれぞれ 0 rpm から 5000 rpm まで 1000 rpm ずつ変化させてプロペラが発生させる推力を計測した。結果を次に示す。

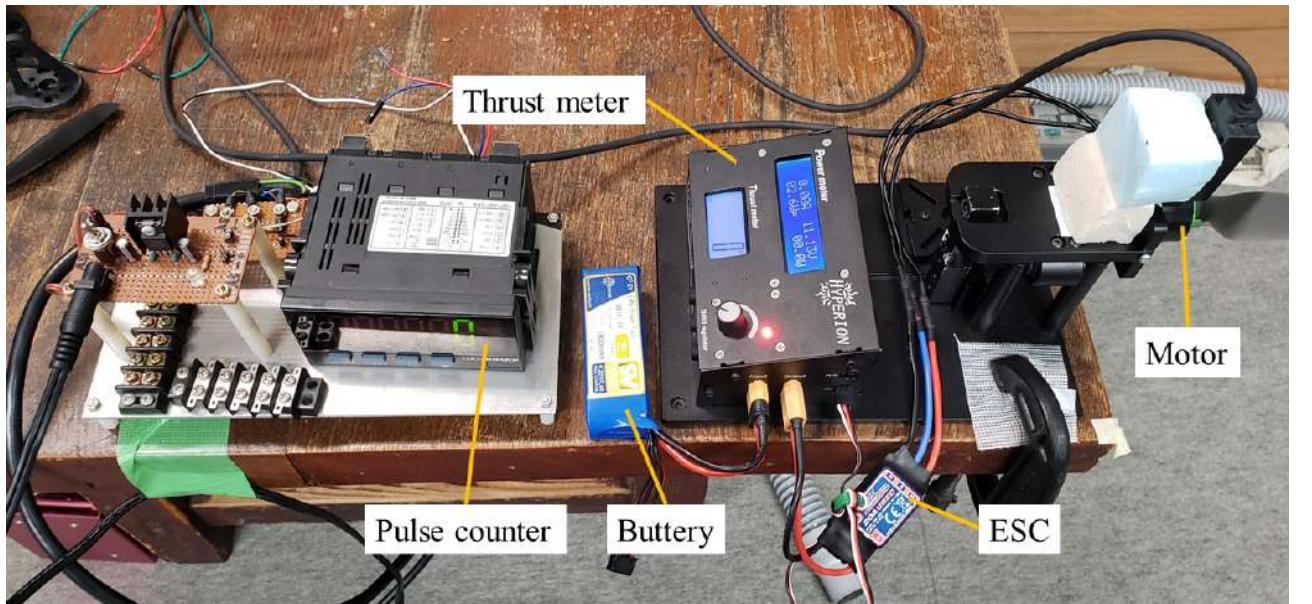


Fig. 53 Configuration for measuring propeller thrust

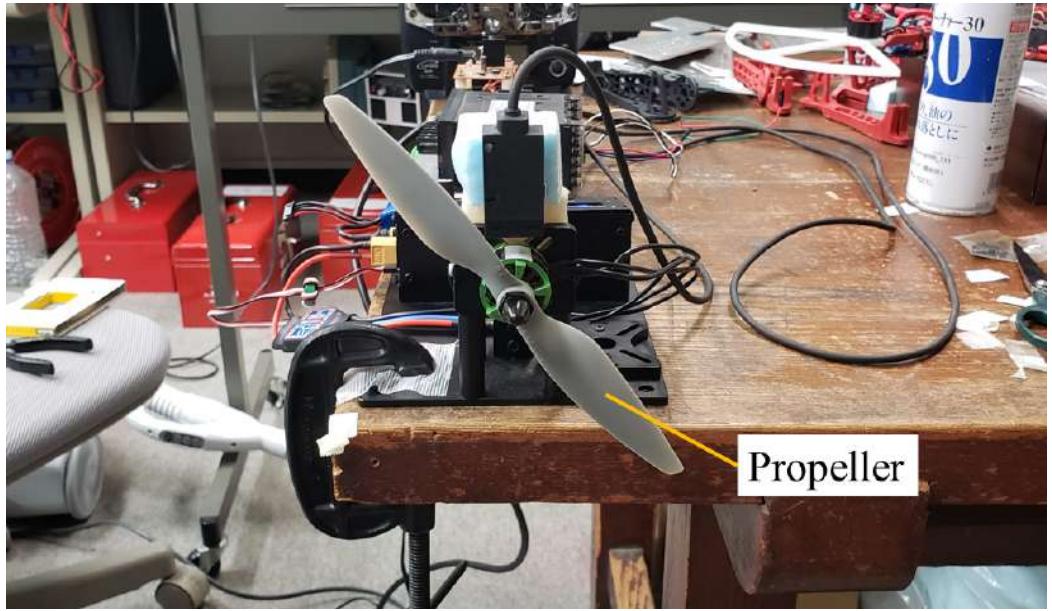


Fig. 54 Propeller fixed to thrust meter

次に飛行実験に実際に用いたプロペラと予備用に用意したプロペラの同回転数時に発生させる推力の差を確認した。結果を次の図に示す。

結果からわかるように、いずれの回転数においてもプロペラ 1 の推力の方がプロペラ 2 に比べて % 推力が低いという結果を得た。

## 5.4 従来の推定アルゴリズムによる制御性能の評価

2.4 章にて示した従来の位置推定方法及び 5.1 章にて示した制御手法を用いてクアッドロータの位置制御を行った。次の 2 つの実験で用いたクアッドロータは Fig. 31 に示すものである。

### 5.4.1 定点における位置保持実験

ある目標点を目標値とし、その点から逸脱することなくクアッドロータの位置を制御可能かどうかを確認するのが目的である。実験方法を以下に示す。まず、クアッドロータをアンカで取り囲まれた空間内の床面に置く。そして、その位置が原点になるようにプログラム内で座標変換を行い、キャリブレーションを行う。キャリブレーション終了後、クアッドロータを手動制御にて離陸させる。飛行が安定した時点でモードを手動制御モードから自動制御モードへと切り替え、その際の推定された位置をロギングする。なお、この実験時にはクアッドロータのスロットルの制御は自身の手により行った。本実験における目標値は  $(x, y) = (0, 0)$  つまり、実験開始時にクアッドロータを置いた位置である。結果を Fig. 55 に示す。

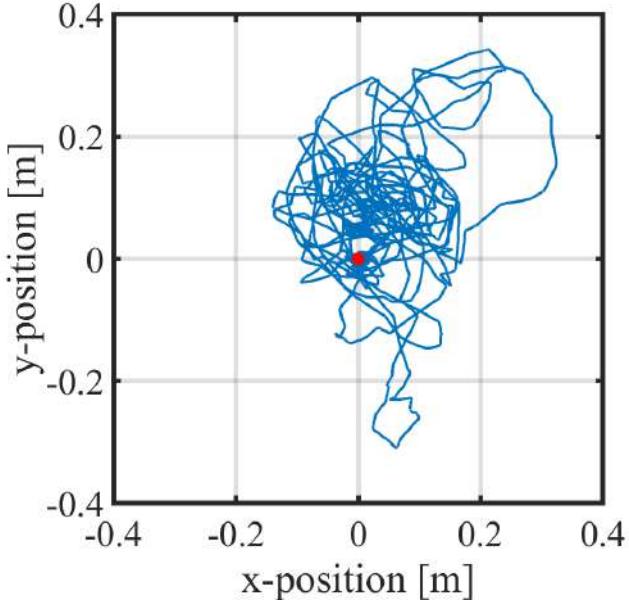


Fig. 55 Result of static position hold experiment

図中の赤い点が目標値を示す。なお、このグラフは自動制御モードに切り替えてから2分間の飛行をロギングしたものである。目標値からの標準偏差は0.083mであり、目標値からの最大偏差は0.406 mという結果を得た。先行研究においては、姿勢角制御の周波数が低いことが原因で位置制御が発散していた。しかし、フライトコントローラを新たに搭載し、位置制御と姿勢角制御を分けたことにより処理が軽くなったり、位置の発散が起きなくなり、こういった結果を得ることができたと考えられる。

#### 5.4.2 複数点間を移動する位置制御実験

次に行ったのが複数点間を移動する位置制御実験を行った。使用したクアッドロータは前述したように前の定点保持実験と同一の機体であり、位置推定方法も同様の方法である。実験目的は、目標値が複数存在し、飛行中にその値が変更されても目標値から逸脱することなく位置を制御可能か確認することである。実験方法を以下に示す。まず、クアッドロータを床面に置き原点としてキャリブレーションを行う。そして手動制御で離陸させた後、安定した時点で自動制御モードに切り替える。ここまで的方法は前の実験と同様である。その後、目標値を手元の送信機より切り替え、複数点を移動させる。そして、その際の推定された位置をロギングする。目標値の座標を次のTable 9に示す。

Table 9 Absolute position of way points

|                 | <i>x</i> position [m] | <i>y</i> position [m] |
|-----------------|-----------------------|-----------------------|
| Point0 (Origin) | 0.00                  | 0.00                  |
| Point1          | 1.50                  | -1.00                 |
| Point2          | -1.50                 | 1.00                  |

そして、実験結果を Fig. 56 に示す。

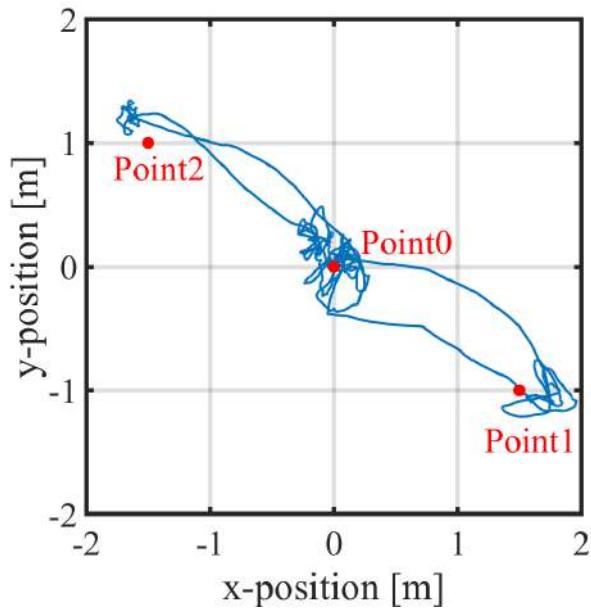


Fig. 56 Result of point to point experiment

目標値が原点の場合 (Point0) , 細かな振動はあるが、偏差の中心は目標値と一致していることがわかる。しかし、移動後の点 Point1, Point2 においては定常偏差が発生し、偏差の中心が目標値からずれていることがわかる。この原因は UWB アンカの設置誤差が起因する位置推定誤差や UWB モジュール間の距離が近くなると測距精度が悪くなることなどが考えられる。

## 5.5 提案する推定アルゴリズムによる制御性能の評価

## 6. 結言

本研究では非 GPS 環境における鋼構造物点検を目的とした自律飛行可能なクアッドロータの開発を行った。クアッドロータの自己位置推定には UWB モジュール, IMU, 距離センサの値を参照し, それらを拡張カルマンフィルタ (EKF) に適用することで位置を推定した。また, 先行研究では用いられなかったオプティカルフローセンサも新たに搭載し, 位置推定及び位置制御の精度が向上するかを確認した。

まず, 初めに行ったのが, 先行研究において未達成であった位置制御を達成するためのアプローチである。先行研究において, 上記の UWB, IMU, 距離センサを用いた構成にてクアッドロータの位置を推定することは達成されていた。しかし, メインコンピュータを用いてクアッドロータの姿勢角を制御する際, その周波数が不十分であり, クアッドロータの位置が発散してしまっていた。そこで, 新たに別のライトコントローラを搭載し, 位置推定用のコンピュータと姿勢角制御用のコンピュータを分けた。それにより演算負荷が軽くなったため, クアッドロータの位置を制御することが可能になった。先行研究から用いられてきたクアッドロータを用いた位置制御では定点ホバリング及び 3 点間を移動する制御を行った。まず, 定点ホバリングでは, 目標位置 (原点) からの標準偏差が 0.083 m, 最大偏差が 0.406 m という位置制御結果を得た。また, 3 点間を移動する制御では, 移動後の点において定常偏差が発生したが, 目標点を大きく逸脱することなく, 安定した位置制御を行うことができた。

次に, 上で行った位置制御において発生した,  $x - y$  平面上でのクアッドロータの細かな振動を取り除くために新たにオプティカルフローセンサを搭載することを考えた。それに伴い, 以前のクアッドロータでは搭載するスペースや推力が不足すると考えられため, 発生可能な推力が大きく, センサなどが搭載可能な余剰スペースがある新たなクアッドロータを構築した。そのクアッドロータに対しオプティカルフローを搭載し, 実験したデータから位置推定の精度を確認するシミュレーションを行ったところ, 位置推定の精度が向上し, オプティカルフローセンサの有効性を確認することができた。しかし, オプティカルフローセンサも位置推定に組み込んだ状態で飛行させ, 位置制御を行ったところ, オプティカルフローセンサの値を参照していない場合よりも位置制御の精度は向上したが, 以前のクアッドロータよりも格段に位置制御の精度が向上するという結果は得られなかった。理由としては, クアッドロータの変更に伴い, 機体特性が変化したにも関わらず, 位置制御用のゲインの調整を行わなかったことや, ライトコントローラにおける姿勢角制御用のゲインの調整が不十分であったことなどが理由として考えられる。また, 実際に実験に用いられたプロペラと, 予備用に確保した別のプロペラの同回転数時における発生推力を確認したところ, 発生する推力に大きな違いがあることがわかった。実験に使用したプロペラは比較したプロペラよりもいずれの回転数域においても平均 % ほど発生推力が低かった。したがって, 推力が不十分であったため機体の姿勢角制御における応答性が悪く, 位置制御結果に影響を及ぼしたことも理由の一つとして考えられる。

今後の課題としては, 部品選定を適切に行なった上で推力が確実に発生させられる機体構成に改善し, 位置制御ゲインや姿勢角制御ゲインをさらに調整する必要がある。また, 位置の真値を計測するのに用いたステレオカメラは, 奥行方向の計測誤差が大きく, ローパスフィルタを通さなければ, 適切な値を得られなかつたため, カメラを複数台用いるなどして測定精度の向上を図る必要がある。さらに, 本研究では UWB モジュールのアンカの数が 4 つと計測可能

な範囲や計測精度が限られていたため、計測周波数が落ち込まない工夫をした上でアンカの数をさらに増やす必要があると考えられる。

## 謝辞

この研究テーマを与えて下さり、またご指導ご鞭撻を下さった澤田祐一教授、東善之助教授に感謝致します。また、加えて報告会において貴重な意見を下さったロボティクス研究室の飛行ロボットグループメンバやアドバイスを下さった研究室の皆様に感謝致します。加えて、本研究の学会発表の際に共同研究者としてお名前を貸して下さり、またミーティングの際には貴重なアドバイスを下さった知的構造システム学研究室の増田先生や防振システム工学研究室の三浦先生に心より感謝を申し上げたいと思います。最後にクアッドロータの実験を行うに至って貴重な部屋を貸して下さった産学公連携推進センターの皆様にも謝意を表します。

## 参考文献

- (1) 国土交通省, 橋梁の現状,  
[http://www.mlit.go.jp/sogoseisaku/maintenance/\\_pdf/research01\\_pdf01.pdf](http://www.mlit.go.jp/sogoseisaku/maintenance/_pdf/research01_pdf01.pdf),  
2018/02/02 閲覧
- (2) 国土交通省, 道路構造物の現状（橋梁）,  
[http://www.mlit.go.jp/road/sisaku/yobohozen/yobo1\\_1.pdf](http://www.mlit.go.jp/road/sisaku/yobohozen/yobo1_1.pdf), 2018/02/02 閲覧
- (3) 大阪市立大学, 橋の鋼鉄部材を検査できる橋梁検査ロボット バイリム (BIREM) の開発,  
<https://www.osaka-cu.ac.jp/ja/news/2013/e4oxuo>, 2018/02/02 閲覧
- (4) 次世代インフラ用ロボット技術・ロボットシステム～現場実証ポータルサイト～,  
<http://www.c-robotech.info/> 平成 26 年度現場検証技術 db-1/ 平成 26 年度橋梁維持管理部会 db/ 橋梁-富士フィルム, 2018/02/02 閲覧
- (5) 水谷将馬ら, 錯雑した構造体中で飛行が可能な回転球殻を持つクアッドロータ, ロボティクス・メカトロニクス講演会 2014, (2014), 1-4.
- (6) 東北大学, Human-Robot Informatics Laboratory, ぶつかっても落ちないドローン (球殻ヘリ) の研究,  
[http://www.mlit.go.jp/sogoseisaku/maintenance/\\_pdf/research01\\_pdf01.pdf](http://www.mlit.go.jp/sogoseisaku/maintenance/_pdf/research01_pdf01.pdf),  
2018/02/02 閲覧
- (7) 構造物メンテナンス研究センター CAESAR,  
<http://www.pwri.go.jp/caesar/manual/>, 2018/02/02 閲覧
- (8) S. Akahori, Y. Higashi, A. Masuda, K. Takeuchi, Development of a Vibration Probe Foot Using an EPM for Aerial Inspection Robots, Proc. Of the 10th International Symposium on Advanced Science and Technology in Experimental Mechanics, (2015), 55.
- (9) S. Akahori, Y. Higashi, A. Masuda, Development of an Aerial Inspection Robot with EPM and Camera Arm for Steel Structures, TENCON 2016 - 2016 IEEE Region 10 Conference, (2016), 1264.
- (10) Autonomous Control Systems Laboratory Ltd.,  
<http://www.acsl.co.jp/products/>, 2018/02/02 閲覧
- (11) decaWave,  
<https://www.decawave.com/products/dwm1000-module>, 2018/02/02 閲覧
- (12) G. Bradski, A. Kaehler, 松田晃一, 詳解 OpenCV コンピュータビジョンライブラリを使った画像処理・認識, (2012), 377-387, 423-432, オーム社.
- (13) M. Kok, T. B. Schon, Magnetometer Calibration Using Inertial Sensors, IEEE Sensors Journal Volume: 16, (2016), 5679-5689.

- (14) Q. Li, J. G. Griffiths, Least Squares Ellipsoid Specific Fitting, Geometric Modeling and Processing 2014, (2014), 2-5.
- (15) 足立修一, 丸田一郎, カルマンフィルタの基礎, (2013), 99-110, 東京電機大学出版局.
- (16) T. Madani, A. Benallegue, Backstepping Control for a Quadrotor Helicopter, Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and System, 9419317, (2006), 1-6.
- (17) Z. Fang, W. Gao, Adaptive Integral Backstepping Control of a Micro-Quadrotor, Intelligent Control and Information Processing (ICICIP), 12217869, (2011), 1-6.
- (18) Z. Zuo, Quadrotor Trajectory Tracking Control: A PD Control Algorithm, 2010 3rd International Conference on Computer and Electrical Engineering (ICCEE 2010), (2010), 3-4.