

Abstract

In recent years, the aging of infrastructure has been regarded as a problem in Japan, but the local governments that manage them have a shortage of financial resources and inspectors. Therefore, we focused on the inspection method using UAV (Unmanned Aerial Vehicle). Replacing the inspection performed by the special work vehicle with the inspection by the UAV can reduce costs and ensure worker safety. However, because UAVs require technique to operate, automatic flight by GPS is generally used, but radio waves from satellites are blocked on bridges and cannot be used. Therefore, the position of the UAV was estimated using the UWB (Ultra-Wide Band) module which is a ranging device and the optical flow sensor which can measure the speed, and the method for autonomous flight was shown. Next, the experiment showed that the accuracy of the estimated position was sufficiently high. In addition, we performed position holding control with only one point as the target value and autonomous flight control with multiple points as the target values. As a result, it was confirmed that position control was possible without deviating from the target point.

概要

近年の日本ではインフラの老朽化が問題視されているが、それらを管理する地方自治体は財源や点検者が不足しているため、低コストで簡便に行えるインフラ点検の技術が求められる。そこで UAV（Unmanned Aerial Vehicle）による点検手法に着目した。特殊作業車にて行っていた点検を UAV による点検に置き換えることでコストを下げ、作業者の安全も確保できる。しかし UAV の操縦には技術が必要とされるため、GPS による自動飛行が一般的であるが、橋梁においては衛星からの電波が遮られ使用不可能である。そこで、測距機器である UWB（Ultra-Wide Band）モジュールや速度が計測可能なオプティカルフローセンサを用いて UAV の位置を推定し、自律飛行のための手法を示した。次に、実験により推定された位置の精度を確認すると十分に高い結果を得た。また、定点を目標値とする位置保持制御や複数点を目標値とする自律飛行制御を行ったところ、目標点から逸脱することなく位置制御が可能なことを確認した。

目次

1. 緒言	1
1.1 研究背景	1
1.2 インフラ点検用 UAV 及び点検機器の開発	2
1.3 一般的な自律飛行型 UAV	4
1.4 研究目的	5
2. クアッドロータの位置推定アルゴリズム	7
2.1 拡張カルマンフィルタの原理 ⁽¹¹⁾	7
2.2 UWB 通信	9
2.2.1 UWB モジュールの測距原理	9
2.3 オプティカルフローセンサ	18
2.3.1 オプティカルフローセンサの原理	18
2.3.2 センサのキャリブレーション	19
2.4 UWB モジュール, IMU, 距離センサのみを用いた従来の位置推定システム	23
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	30
3. 位置制御用クアッドロータの開発	34
4. 位置推定アルゴリズムの精度評価	36
4.1 従来推定アルゴリズムを用いた位置推定の精度評価	36
4.2 センサの種類及びカルマンフィルタの分散値が位置推定精度に与える影響の考察	38
5. 位置制御アルゴリズムの評価	44
5.1 位置制御手法	44
5.2 ステレオカメラを用いた位置測定システム	44
5.2.1 位置計測システムの原理及び構築	44
5.2.2 奥行き誤差の検証	50
5.3 従来の推定アルゴリズムによる制御性能の評価	51
5.3.1 定点における位置保持実験	51
5.3.2 複数点間を移動する位置制御実験	52
5.4 提案する推定アルゴリズムによる制御性能の評価	53
5.5 ロータの推力測定	58
6. 結言	62
謝辞	64
参考文献	65

1. 緒言

1.1 研究背景

近年の日本において、インフラ、特にその中で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⁽¹⁾。こうした老朽化した橋は今後も増加すると予想され、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、海外だけでなく、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の形跡が見過ごされていた⁽²⁾。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 つの橋梁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 つの橋梁平均でわずか 8 万円であり⁽³⁾、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。

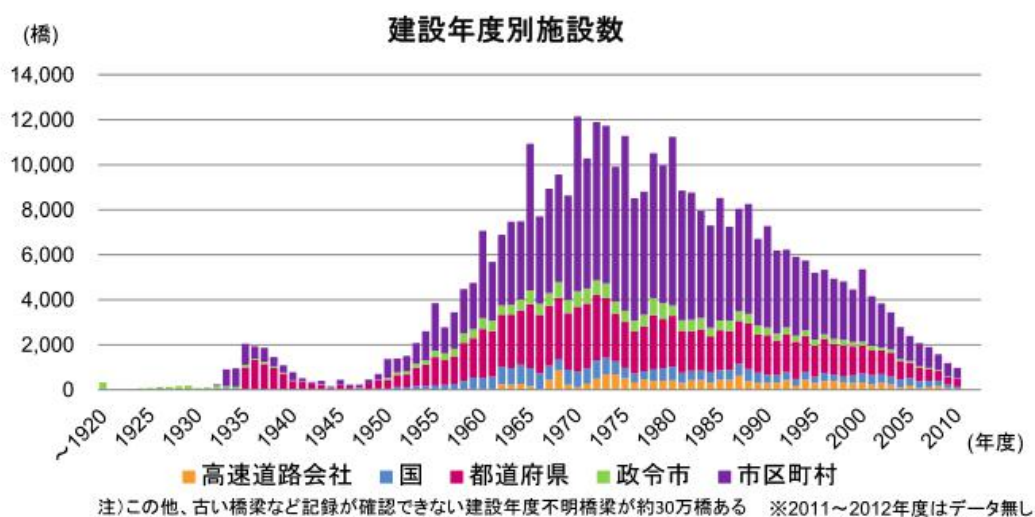


Fig. 1 Number of bridges in Japan by construction year⁽¹⁾.

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可能な地上型ロボットである。

もう1つはUAVなどの飛行型ロボットであり、これらは前者と異なり、空中から点検箇所アプローチ可能である。地上型ロボットとしては高田らが開発したBIREM⁽⁴⁾ (Fig. 2) 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。

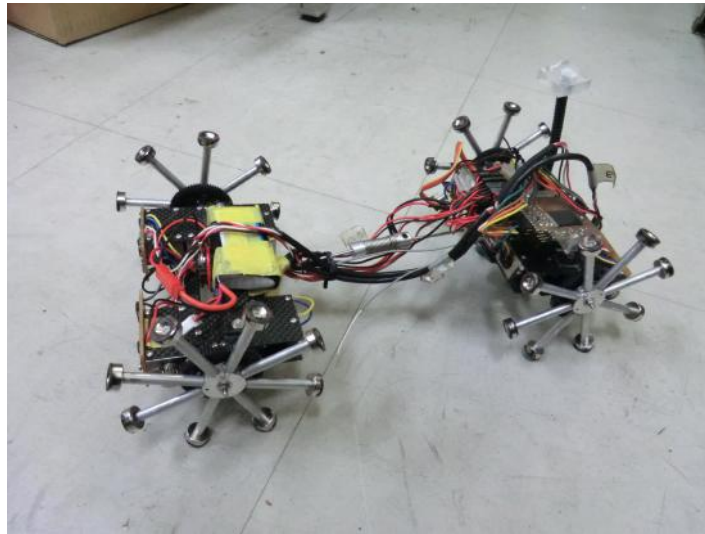


Fig. 2 BIREM⁽⁴⁾.

1.2 インフラ点検用 UAV 及び点検機器の開発

先行研究において、赤堀らは Fig. 3 に示すインフラ点検用 UAV を開発した⁽⁵⁾。この UAV は機体の上部に EPM (Electro-Parmanent Magnet) と呼ばれる磁石を搭載し、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行うことが可能である。

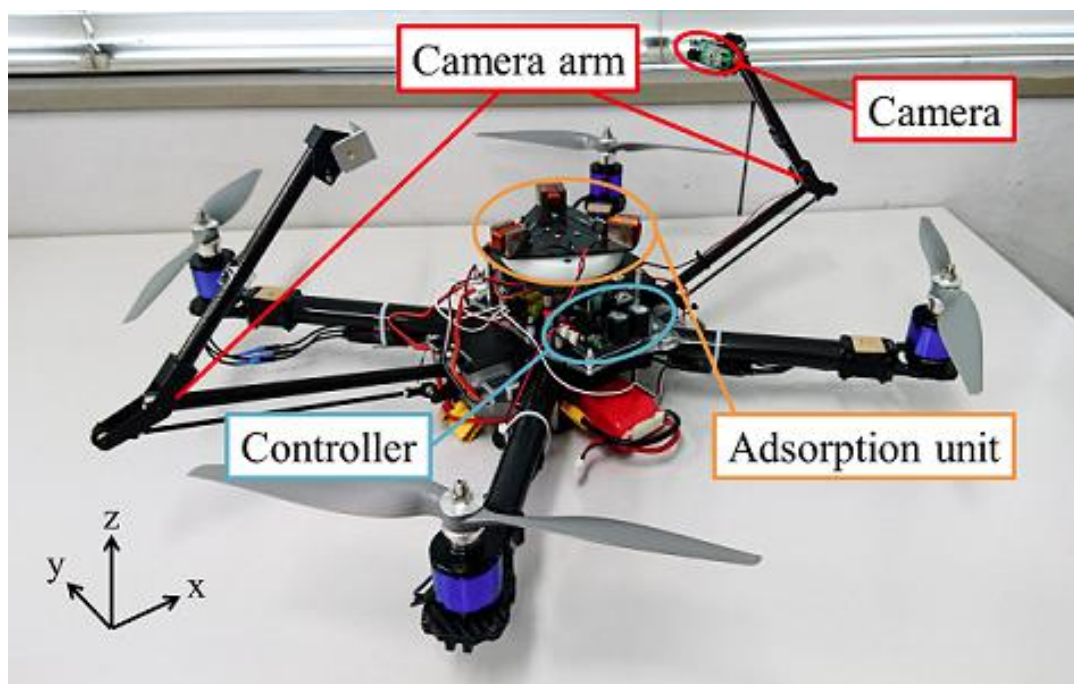


Fig. 3 Overall view of the developed inspection robot in the previous study⁽⁵⁾.

また、著者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（Fig. 4）．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ており，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．ロギングした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．このモジュールを用いた点検方法としては，まず点検箇所まで UAV を用いてこれを運搬し，EPM の磁力で貼り付かせる．必要なデータを計測した後，最終的に UAV で回収する流れである．また，重量が約 230 g であるため，赤堀らが開発した UAV であれば，複数個積載し，運搬することが可能である．

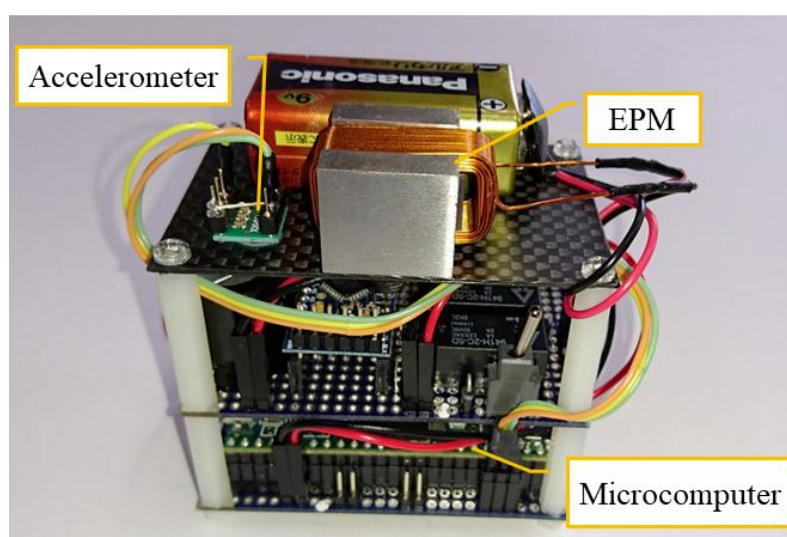


Fig. 4 vibration measurement unit.

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

は全て自身の操縦で行う必要がある。経験の豊富なパイロットであれば狙い通りの箇所に接近し、点検することやセンサを取り付けることが可能である。しかし、経験の浅いパイロットが操縦する場合、非常に困難である。

1.3 一般的な自律飛行型 UAV

飛行型ロボットを点検に用いることの有用性は高いが、操作面においては課題が多いのが現状である。手動操縦は操作技術を要するため、現在市販される多くのドローンは GPS やレーザーレンジファインダなどの位置推定システムを用いて全自動、もしくは半自動飛行が可能である⁽⁶⁾⁽⁷⁾⁽⁸⁾。本研究でも用いたフ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ことにより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、構造物自体が遮蔽物となり GPS の電波が遮断されるため、GPS を用いた位置推定は不可能である。そのような非 GPS 環境においても自動飛行可能な UAV も多くの企業や研究者によって開発が行われている。株式会社自律制御システム研究所は Fig. 5 に示す SLAM (Simultaneous Localization and Mapping) の機能を有した UAV⁽⁹⁾ を開発した。この UAV は非 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行いながら自律飛行をすることが可能である。



Fig. 5 ACSL-PF1⁽⁹⁾.

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行うことで自己位置や方角を推定し自律飛行を行う。しかし、それらのためには高価で精度の高いカメラや画像処理用の GPU を搭載する必要があるため、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様々な値を計算するには計算負荷の高い演算をこなさなくてはならない。点検に掛けることが可能な財源が限られている地方自治体にとって、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、多くのセンサや高負荷な計算が可能である高性能な CPU や GPU を搭載する必要があるため、機体が大型化し、狭い点検空間に入っていくのが困難になる問題がある。

1.4 研究目的

前節で述べたこ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⁽¹⁰⁾。このモジュールは片方のデバイスからもう片方のデバイスに電波を送信することで通信を行う。その際の通信に掛かった時間に電波の速度を掛けることにより、デバイス間の距離を算出することが可能である。このモジュールを用いて距離を求め、後に示す手法により演算することで UAV の位置を 3 次元空間内で推定することが可能である。自己位置を推定する際の UWB 移動局と固定局との関係を Fig. 6 に示す。

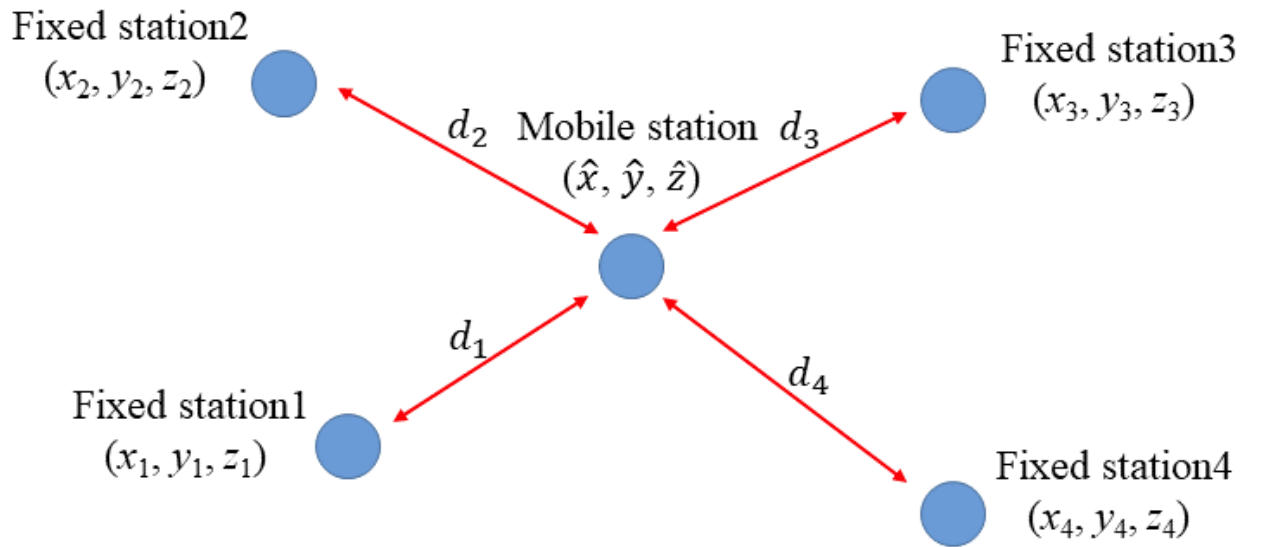


Fig. 6 Schematic diagram of the position estimation method.

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、マルチラレーションの手法を用いることで、移動局の座標を推定す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 1 個あたり数千円と非常に安価であり、位置制御システムのコスト削減に有効である。それに加えてマルチラ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搭載する必要がないため、機体が大型化することもない。しかし、UWB モジュールの更新周期は 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ことである。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により、速度や変位を得ることが可能であるが、値を積分した際にノイズ成分も積分するため、推定値が時間と共にドリフトする。距離センサは積

分することなく，地面との距離を計測することが可能である．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⁽⁵⁾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，姿勢角制御の周期が遅いことが原因で，クアッドロータの位置が発散し，位置を目標地点において制御するには至らなかった．そこで，本研究では姿勢角制御の遅さを補うためにフライトコントローラを追加で新たに導入し，応答性の改善を図った．また，既存の位置推定システムに対し，新たにオプティカルフローセンサも搭載した．それにより，従来までのシステムでは取り除けなかった，目標点上で発生する $x-y$ 平面内におけるクアッドロータの細かい振動を取り除くための改善も行った．次章からはUWBモジュール及び各種センサ値にカルマンフィルタを適用したクアッドロータの位置推定手法および位置制御手法を示していく．

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）を使用した．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．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⁽¹¹⁾

拡張カルマンフィルタは非線形システムを各時刻において線形化し，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2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 $\mathbf{0}$ ，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 $\mathbf{0}$ ，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) の非線形関数をテイ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

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

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

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

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

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

となる．ここでカルマンゲイン $\mathbf{G}(t)$ 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 モジュールの測距原理

UWB の技術が発展するまでは、無線通信を利用した距離測定において Bluetooth や Wi-Fi が多く用いられてきた．片方のデバイスが電波を発信し、それをもう片方が受け取ることでデバイス同士が通信を行う．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、デバイス間の距離を算出する．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いのか、という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⁽¹²⁾．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 1 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. 7 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⁽¹³⁾．

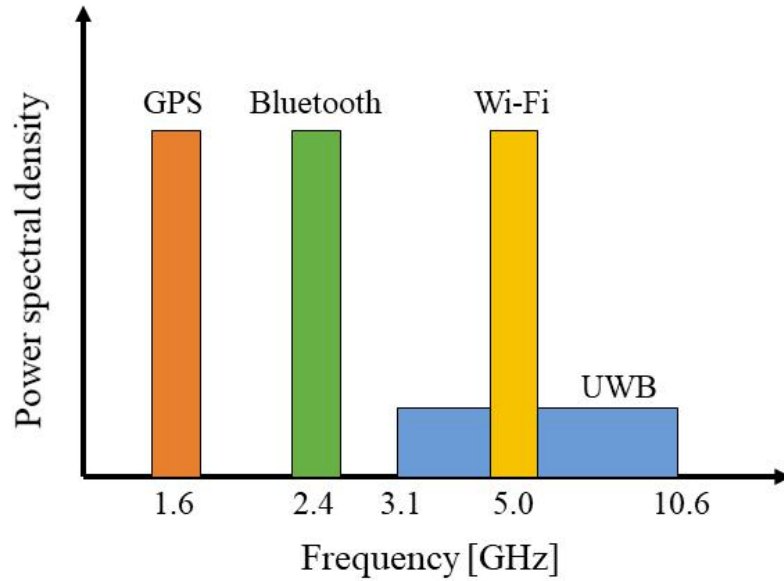


Fig. 7 Comparison of power spectrum between UWB and other wireless communication systems.

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の電子機器等が発生する雑音レベルの約 500 分の 1 以下の非常に小さな出力であり，人体に悪影響を及ぼさない．また，UWB 通信は Fig. 8 に示すようにナノ秒オーダーのパルスを用いて通信に使用し，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⁽¹²⁾．

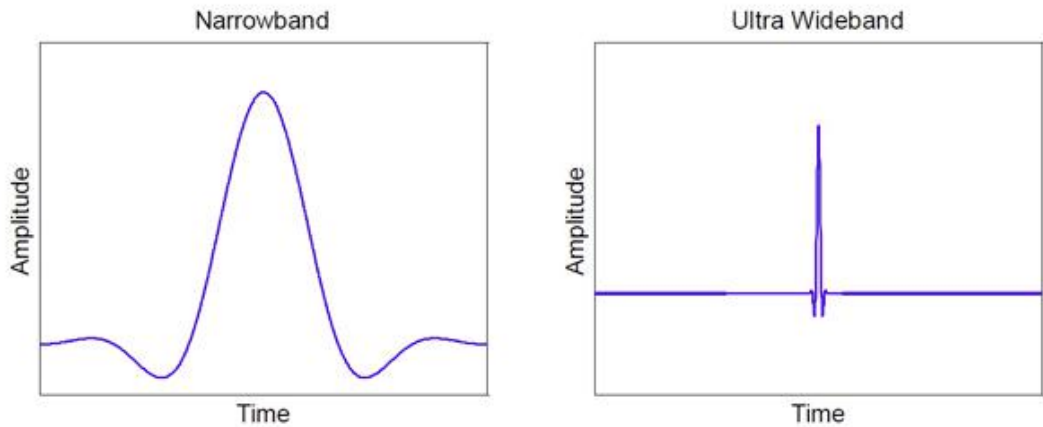


Fig. 8 Comparison of pulses between narrow band and ultra wide band signals⁽¹²⁾.

また，Fig. 9, Fig. 10 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ことなどが特長として挙げられる⁽¹²⁾．

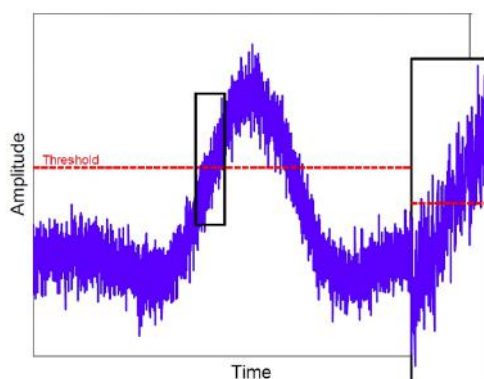


Fig. 9 Narrow band signal with noise⁽¹²⁾.

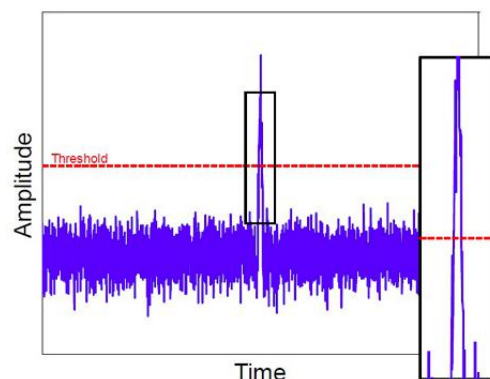


Fig. 10 Ultra wide band signal with noise⁽¹²⁾.

無線通信を測距に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する場合があります。この際、正確な信号受け取れないことや、直接波の位相を反射波が遅らせることが発生する現象である。テレビやラジオにてゴースト障害が起きるのも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。次の Fig. 11, Fig. 12 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。

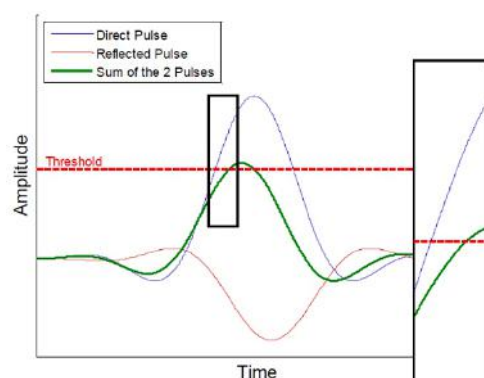


Fig. 11 Narrow band signal with reflections⁽¹²⁾.

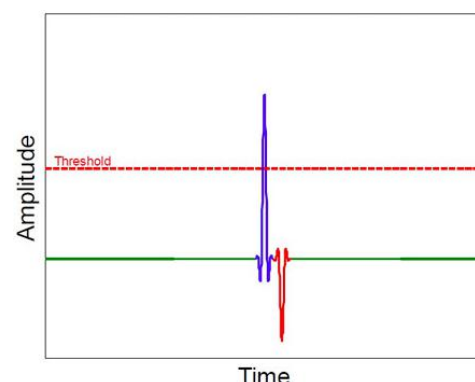


Fig. 12 Ultra wide band signal with reflections⁽¹²⁾.

狭帯域信号では立ち上がり立ち下がりが鈍く、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ことも UWB の特長である⁽¹²⁾。

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. 13 に通信プロトコルを示す.

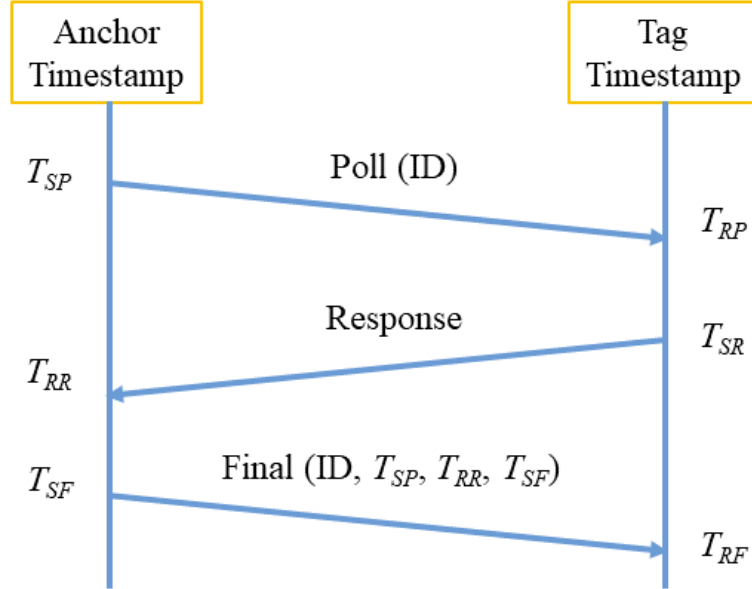


Fig. 13 Ranging protocol of UWB module.

これ以降，クア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. 14 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⁽¹⁴⁾ 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. 15 に示す UltraLife 社製, 1 cell, 1800 mAh のリチウムイオンバッテリーを用いている．図の基板構成で満充電から約3時間容量が保つ．Fig. 14 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．

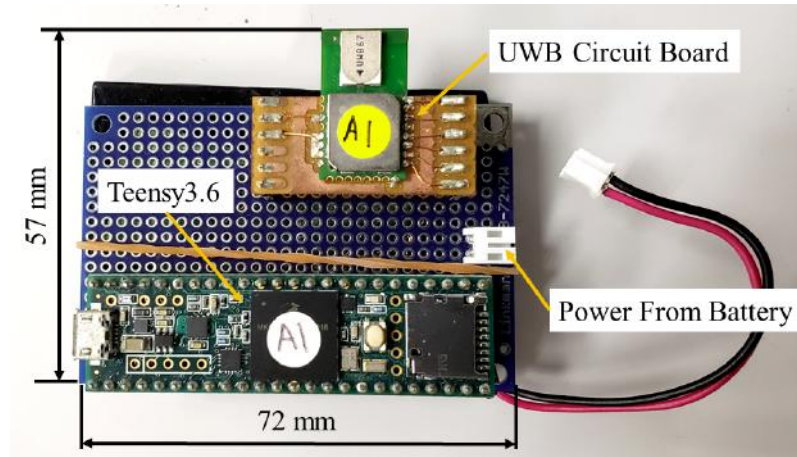


Fig. 14 UWB anchor.



Fig. 15 Battery for UWB Anchor⁽¹⁵⁾.

前述したように UWB 通信はマルチパスに耐性があり，本研究と同様に UWB モジュールを用いたクアドロータの位置制御を行っている研究⁽¹⁶⁾でも言及されているように，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による値の発散を確認した．Fig. 16 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．またその拡大図も Fig. 17 に合わせて示す．

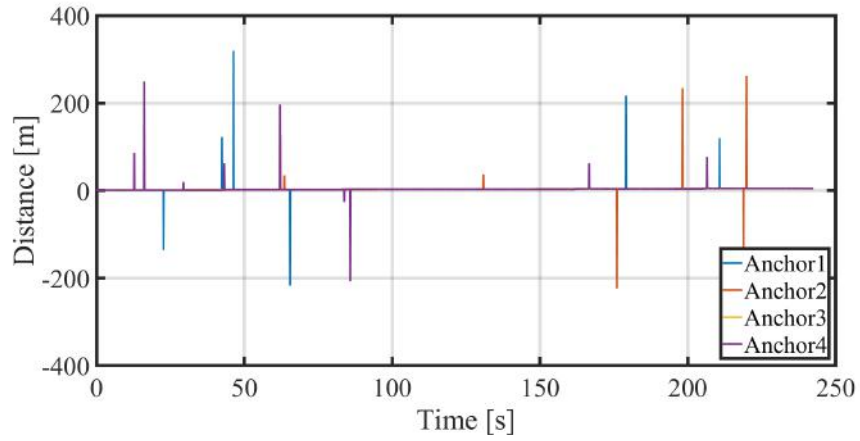


Fig. 16 Ranging result when outliers occur.

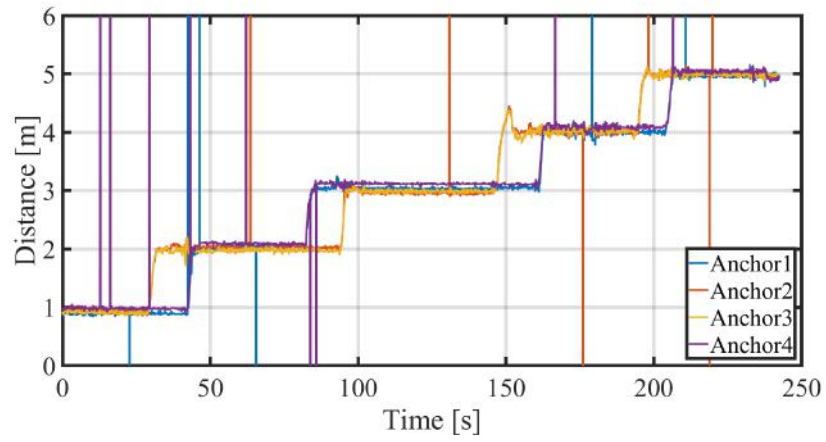


Fig. 17 Ranging result when outliers occur (enlarged image above).

測距手順を以下に説明する．まず，実験時の模式図を次の Fig. 18 に示す．実験室の床に基準となる線を 1 m から 5 m まで 1 m 間隔でマーキングした．図中の赤い線にタグを固定し，4 つのアンカを黒い基準線に添わせて並べる．数秒間測距を行った後，1 m 先の次の基準線へ 4 つのアンカをを移動させる．この手順を他の線においても繰り返した．実験時の模式図を次の Fig. 18 に示す．

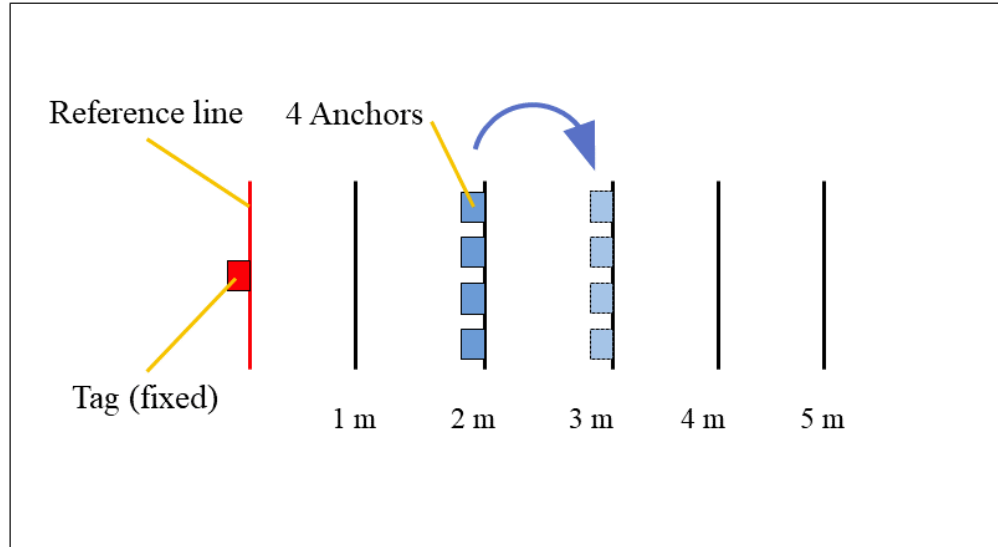


Fig. 18 Schematic diagram of ranging experiment.

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 ほどの値や負の値を計測している．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，電波の指向性を考慮し，モジュールケースの製作を行った．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 軸周りに存在する．ここで，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．

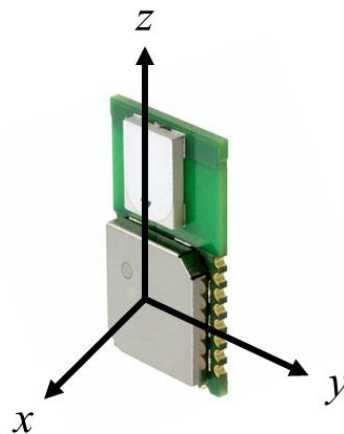


Fig. 19 UWB chip axis.

チップから発せられる電波の指向性を DWM1000 を製造する DecaWave 社がデータシート⁽¹⁴⁾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．

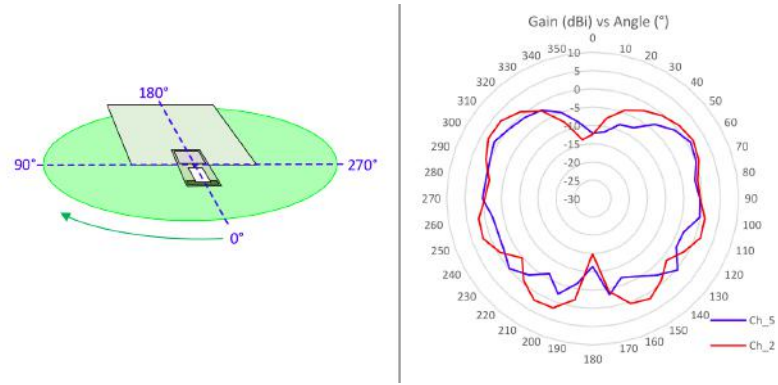


Fig. 20 Directivity of radio wave around x axis⁽¹⁴⁾.

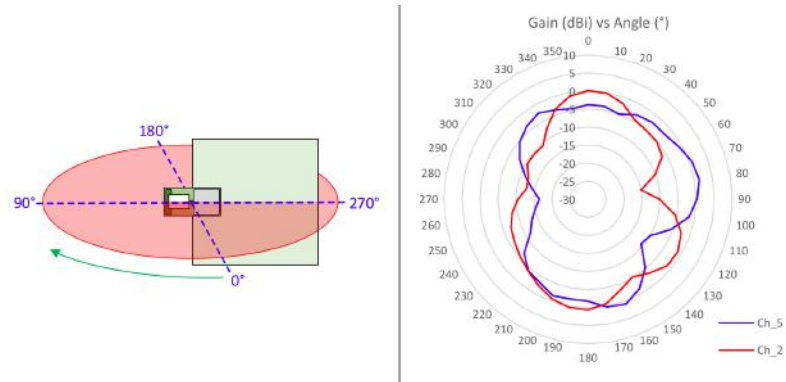


Fig. 21 Directivity of radio wave around y axis⁽¹⁴⁾.

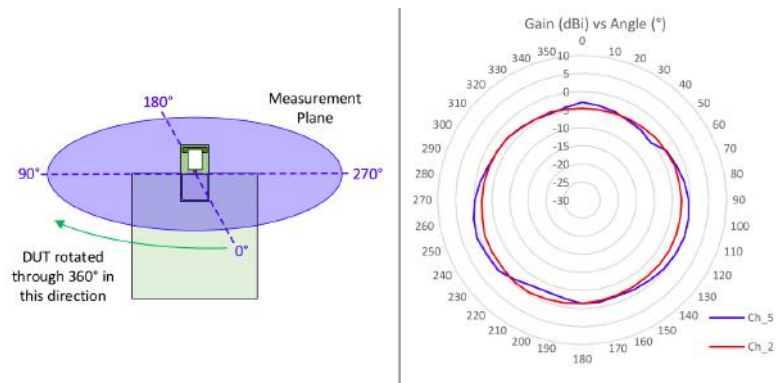


Fig. 22 Directivity of radio wave around z axis⁽¹⁴⁾.

3つの図を考慮すると，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．

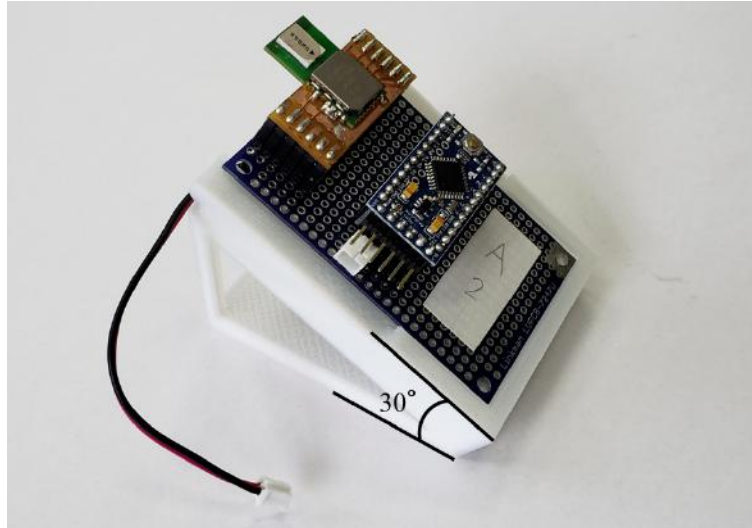


Fig. 23 UWB module holder considering radio wave directivity.

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 30° のオフセット角を与えることができる．次に，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空間内でタグを動かし，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. 16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角度によってマルチパスの発生を抑えることは困難であるという結論を得た．

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1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，1つ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破棄することが可能である．閾値としては，クアッドロータが 1 m/s の速度で水平面内を飛行可能であると仮定し，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．

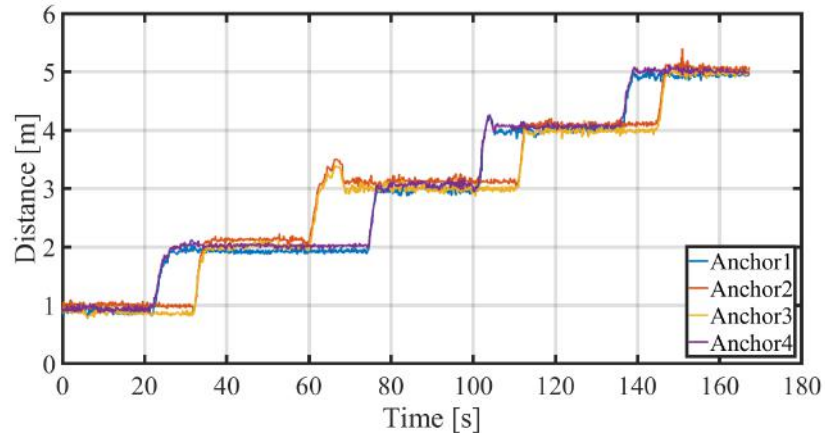


Fig. 24 Ranging result avoiding outlier.

実験方法は Fig. 16 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することができた．

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

先行研究では，クア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，クアドロータが高速度で動いた際や x, y 平面内において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で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，パーソナル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．また，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センサボードに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，オプティカルフローセンサの例としては光学式マウスなどが挙げられる．光学式マウスはオプティカルフローセンサにより，マウスの移動量を算出している．オプティカルフローセンサを用いたクアドロータの位置制御に関する研究⁽¹⁷⁾⁽¹⁸⁾においてもクアドロータにこのセンサを搭載し，クアドロータの位置保持制御や目標点まで移動させる位置制御を行っている．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．

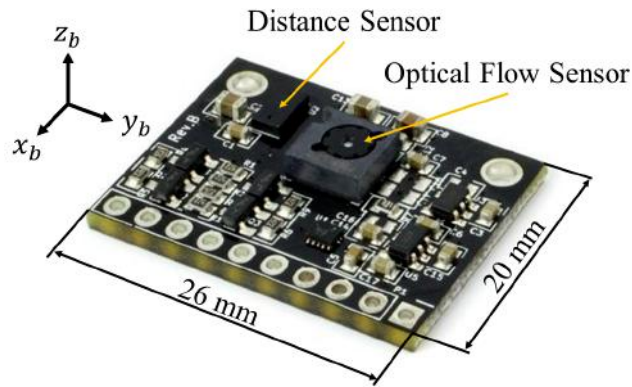


Fig. 25 Optical Flow Sensor Module⁽¹⁹⁾.

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．2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 I²C 通信をそれぞれ用いるため，配線が少なくすむ．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，同一基板上に搭載された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画像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，移動距離が同じであっても，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，高さが 2 倍になった場合，ピクセル数の変化量は半分になる．それを表した図が Fig. 26 である．

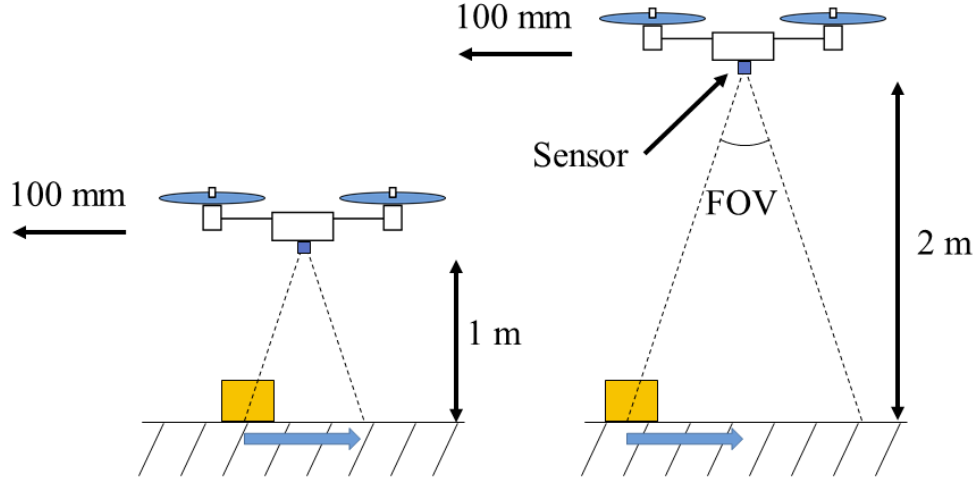


Fig. 26 Difference in Sensor Value Due to Difference in Quad-Rotor Height.

そこで，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可能である．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オプティカルフローセンサを利用した位置推定に関する研究⁽¹⁹⁾に詳しい記述がある．またその論文を参考にすると，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2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 (20) 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 の値を求め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 の値を求めることにした．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、1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。

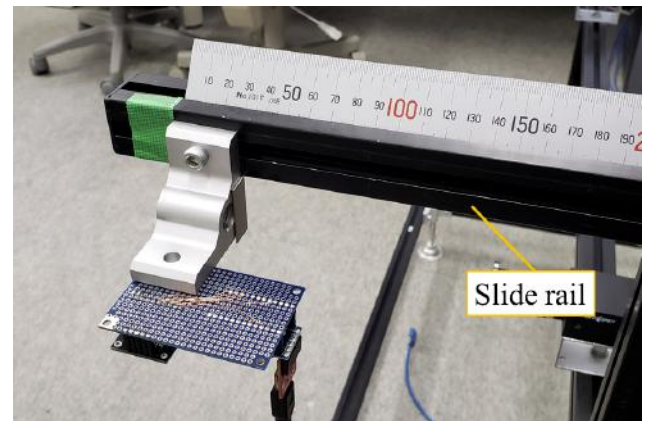
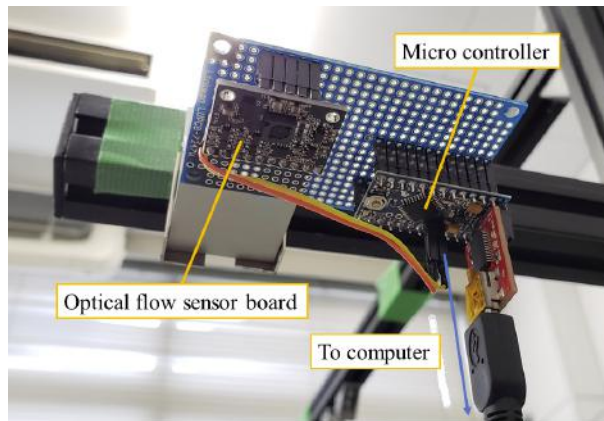


Fig. 27 Narrow band signal with reflections. Fig. 28 Ultra wide band signal with reflections.

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。また、固定したセンサの床からの高さは 775 mm である。センサ値の値は 3 回試行した結果の平均値である。移動距離と計測値の関係を次の Table 2 に、これらの関係をグラフにしたものを Fig. 29 に示す。

Table 2 Sensor values and moving distance.

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

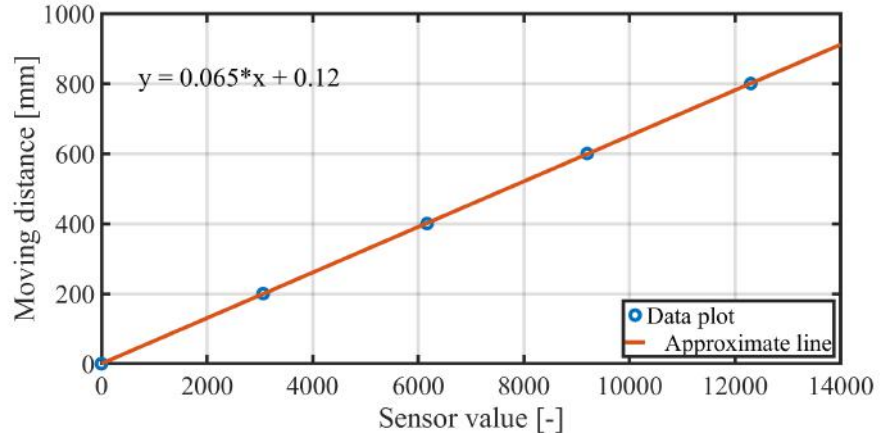


Fig. 29 Relationship between sensor value and moving distance.

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 (20) 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はあった。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はあった。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値はセンサ座標系における速度であり、クア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。

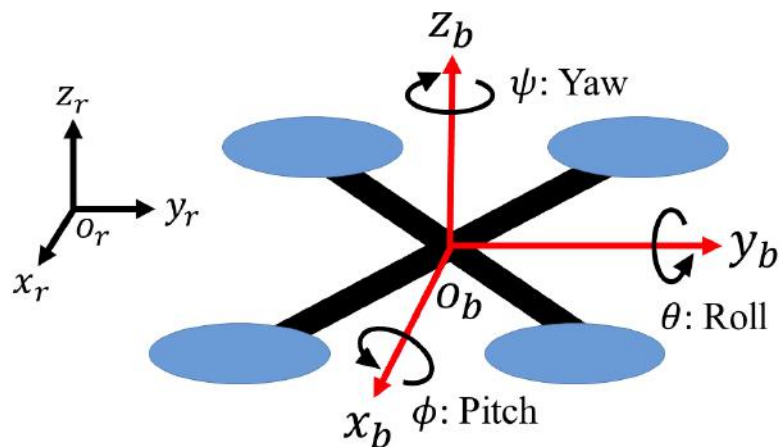


Fig. 30 Relationship between the world coordinate system and the quad-rotor coordinate system.

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。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 ϕ, θ, ψ の3つのオイラー角を用いて表現できる． ϕ, θ, ψ 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 ψ 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 ψ を用いてワールド座標系での速度へ変換する必要がある．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．

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

z 軸周りの回転角度は地磁気センサ及び角速度センサを使用して推定したものを用いる．推定方法は次章にて説明する．式 (20) を用いることでクア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従来の位置推定システム

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ドロータの外観を Fig. 31 に示す．また詳細な仕様を Table 3 に示す．

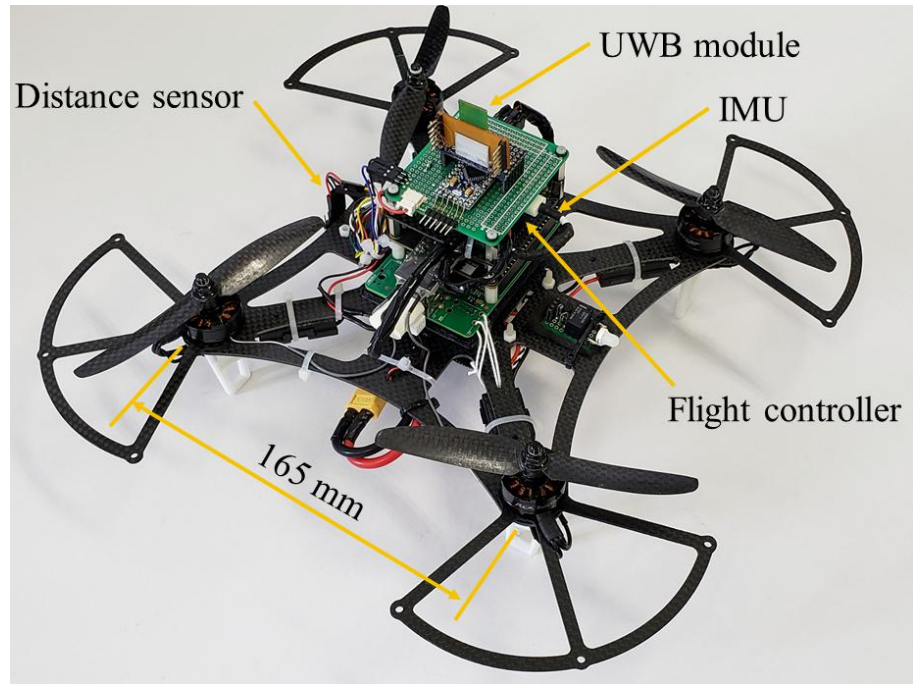


Fig. 31 Quad-rotor equipped with UWB module, IMU and distance sensor.

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

先行研究からの改良点は新たにフ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．

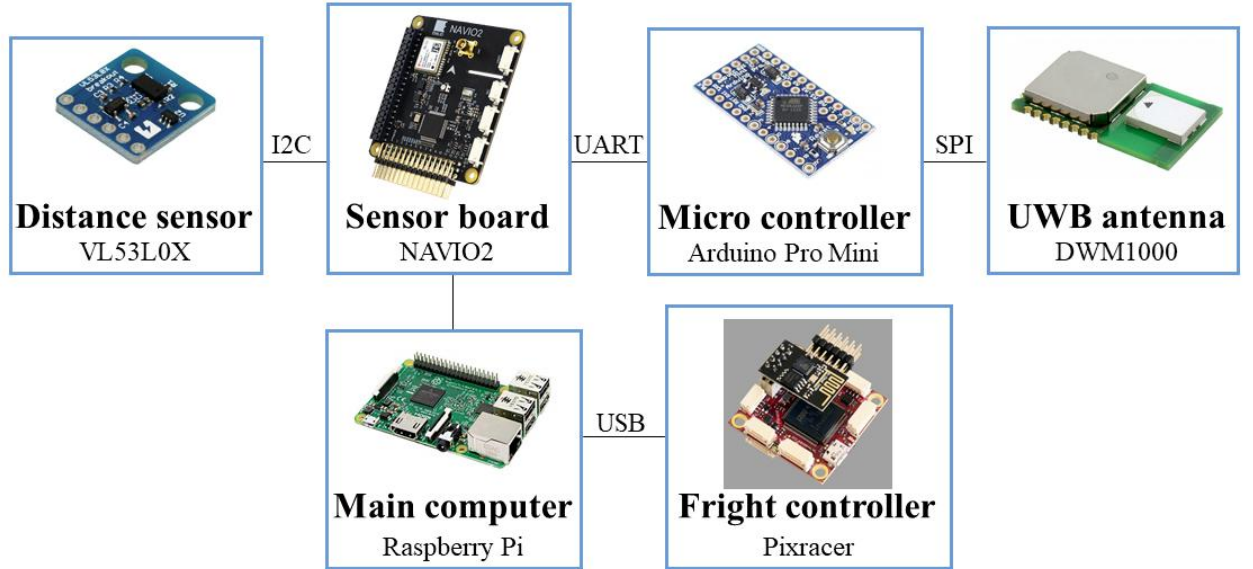


Fig. 32 System configuration of sensors.

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 I²C 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, フ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ドロータの位置制御を行っていた．しかし, その制御システム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ドロータを収束させることが不可能であった．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⁽²⁰⁾ と呼ばれるフ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．

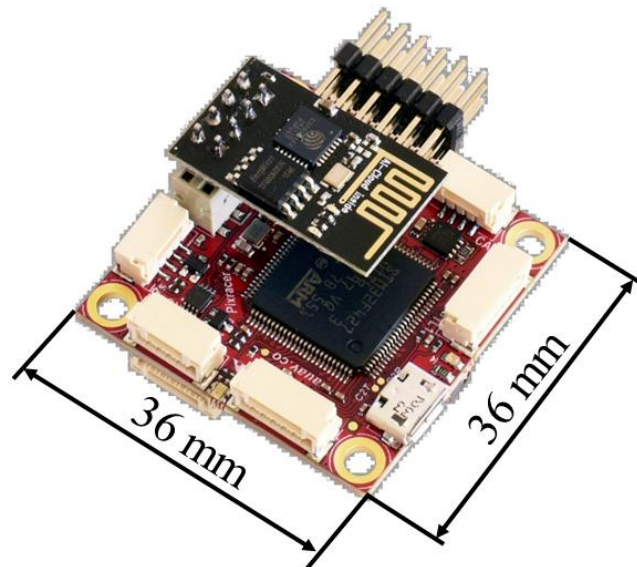


Fig. 33 Appearance of Pixracer⁽²⁰⁾.

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，気圧センサなどの各種センサや 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，自身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，自身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ドロータの姿勢角を制御可能なことである．Fig. 32 に示すようにフ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フライトコントローラに送ることで，クア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．

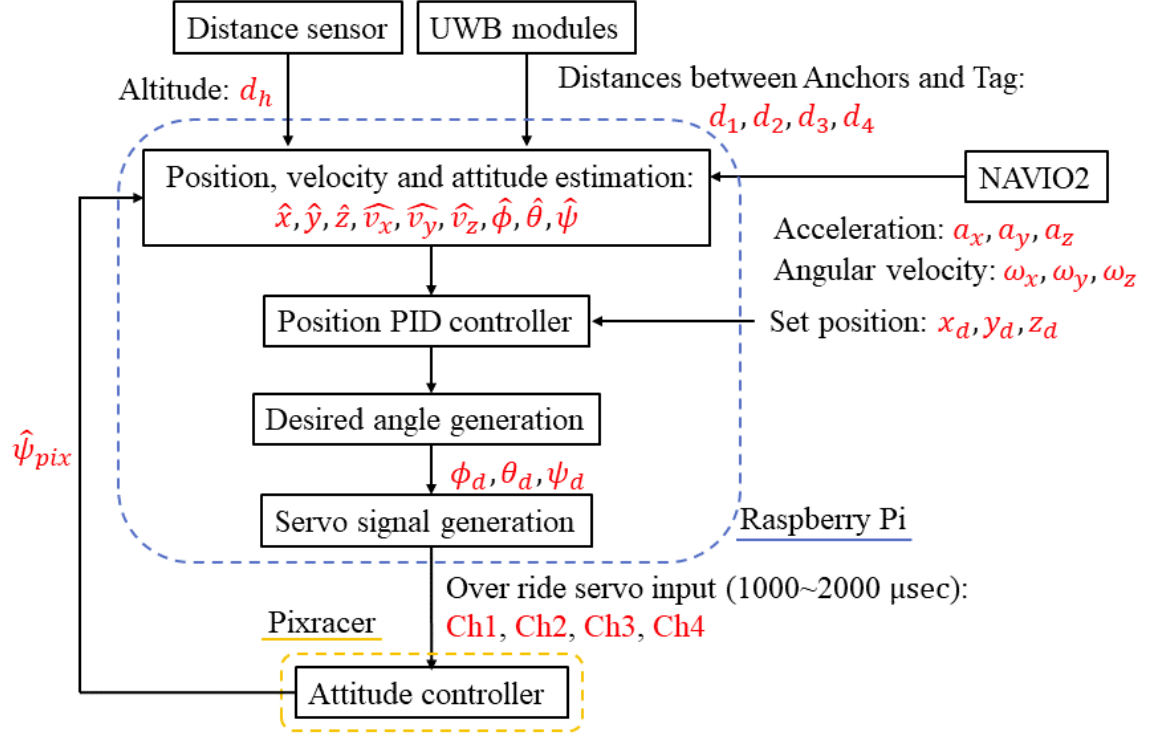


Fig. 34 Block diagram of position control system.

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 (ϕ, θ, ψ) を推定する。次に、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 (ϕ_d, θ_d, ψ_d) を求める。最後に、それら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おいて角速度センサの値を積分して角度に変換することで算出する。しかし、積分した際に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する。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算出値が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る。それら 2 種類のセンサの不利な点を補うのに今回のような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 2 回積分して求めた移動距離と観測方程式側において 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計算した座標をミキシングすることで位置を推定している。加速度センサからは数 1000 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するため値がドリフトする。また、UWB を用いた推定ではド

リフトしない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、クアドロータが高速で運動した際には応答性の悪い推定値になる．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ドロータの位置を推定する方法にマルチラ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、マルチラレーションに関する研究⁽²¹⁾⁽²²⁾に詳しい解法が載っている．Fig. 35 にマルチラレーションにおける模式図を再び示す．

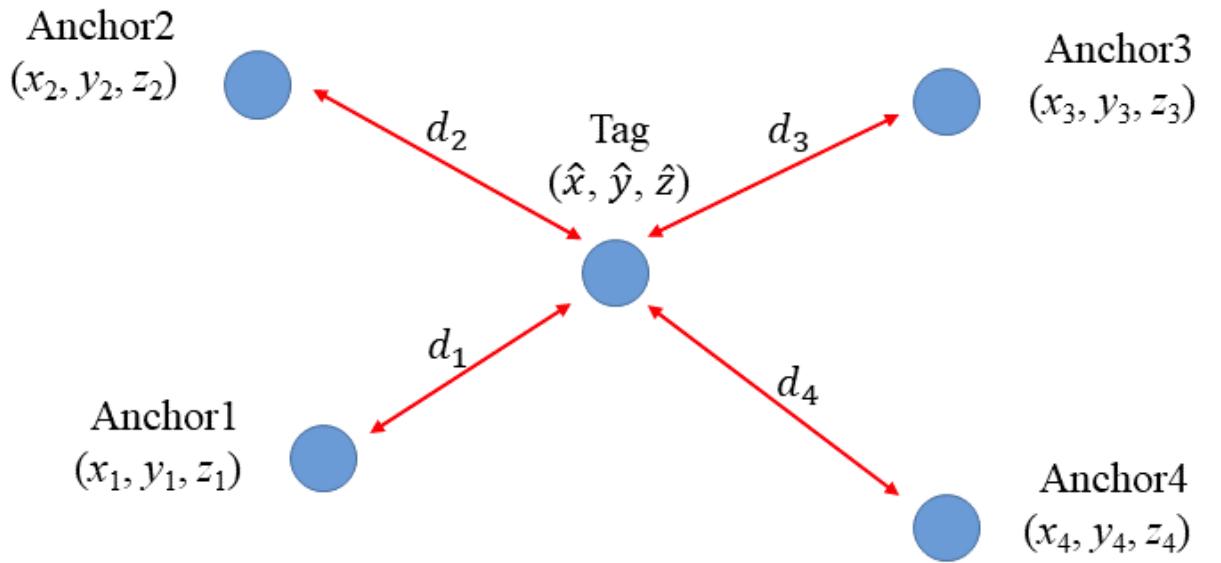


Fig. 35 Position relationship between Anchors and Tag.

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．また、タグは座標未知の移動局であり、クアドロータに搭載されている．このタグの位置を求める．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.

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

また, 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} は先行研究⁽⁵⁾ 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

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

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 $\delta\boldsymbol{\omega} = [\delta p, \delta q, \delta r]^T$ 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 Δt 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フライトコントローラであるPixracerが推定したヨー角 (ψ) 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, ψ 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, ψ_{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 $[\delta a_x, \delta a_y, \delta a_z]^T$, $[\delta d_1, \delta d_2, \delta d_3, \delta d_4]^T$, δd_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に対して 2.1 節にて示した拡張カルマンフィルタを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

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. 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を Fig. 36 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 Fig. 37 に示す.

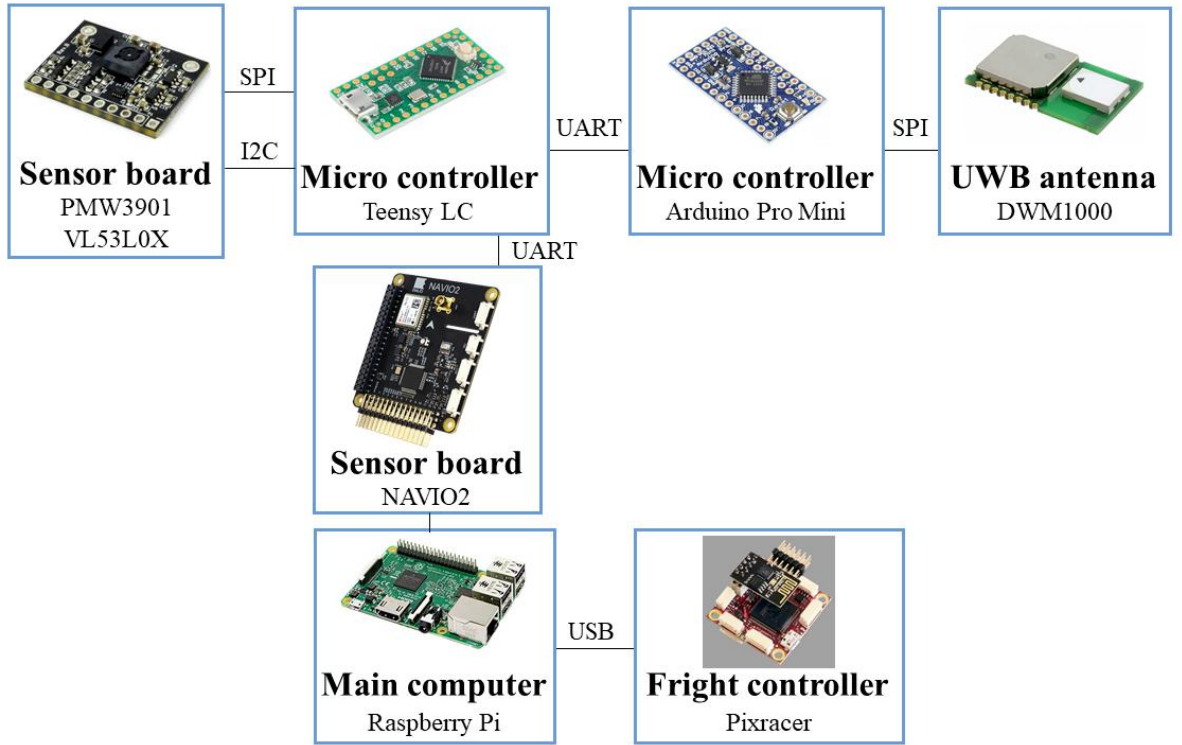


Fig. 36 Configuration of sensors with optical flow sensor board.

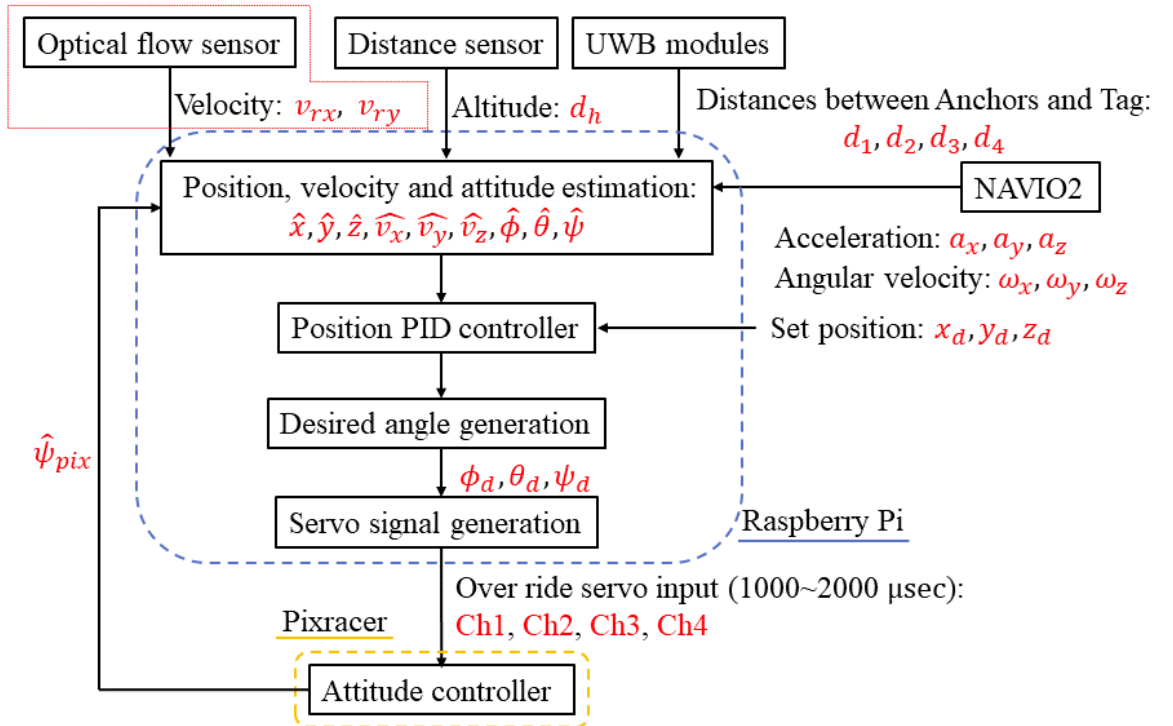


Fig. 37 Block diagram of position control system.

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．ここで、なぜマイコンを2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通信はマスタ側とスレイブ側が4線で通信する通信規格であり、スレイブの数が複数あっても1つのマイコンでそれら进行处理することが可能である．したがって、Fig. 36に示すArduino Pro Mini1つで3つのセンサ（PMW3901, VL53L0X, DWM1000）の値进行处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进行处理するマイコン（Teensy LC）とUWBモジュールの値进行处理するマイコン（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に送る本構成を組んだ．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⁽²³⁾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(20)で表されるヨー角回転を補正した値である．また、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. 位置制御用クアドロータの開発

2.4 節で示したクアドロータは先行研究から利用されてきたものであり、本研究から新たにフ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増加し、推力不足が懸念された。加えて、クア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。

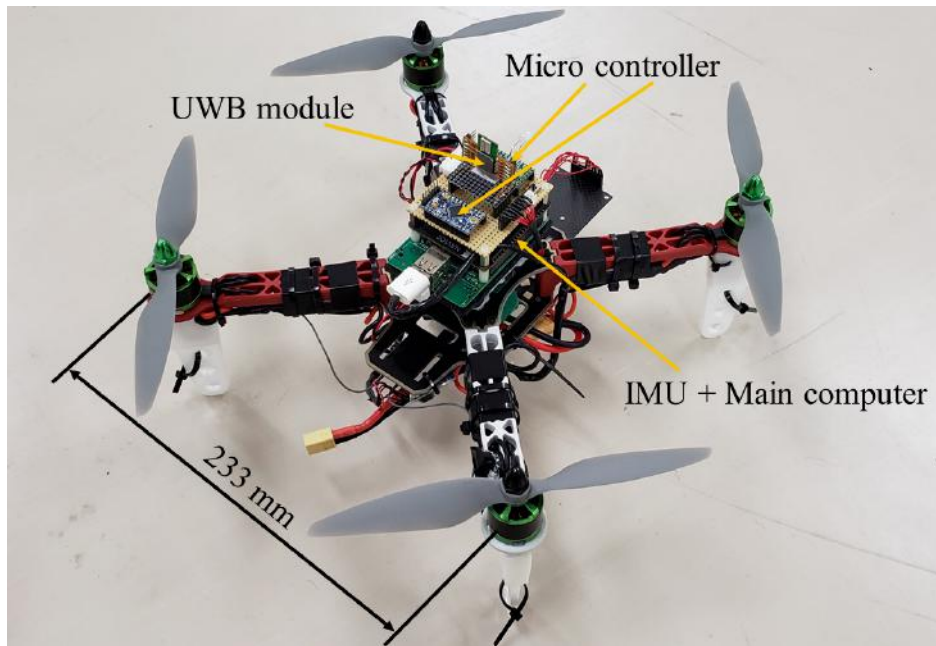


Fig. 38 Front side of quad-rotor.

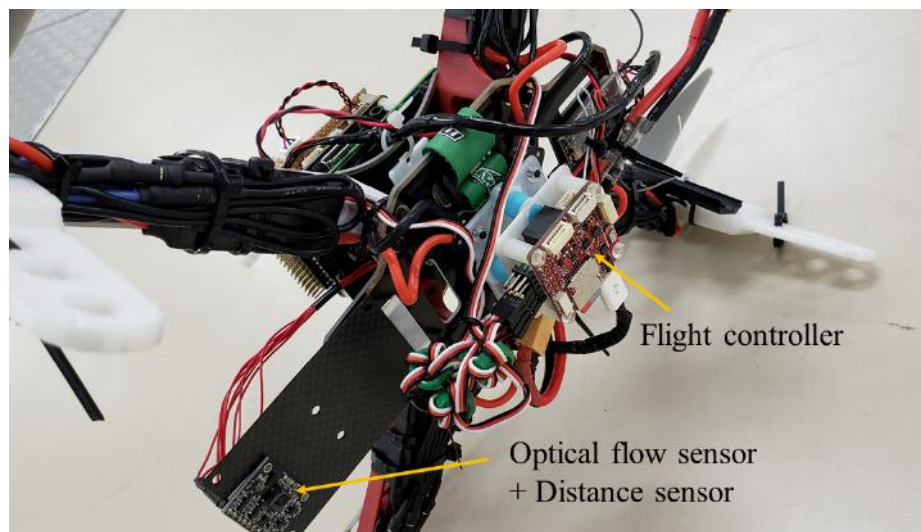


Fig. 39 Back side of quad-rotor.

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，下面にフ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フ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．次の Table 4 に詳細な仕様を示す．

Table 4 New 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

モータやバッテリー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構成である．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

4.1 従来推定アルゴリズムを用いた位置推定の精度評価

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、その推定精度の評価を行う。まず位置推定実験に使用した実験環境を次の図に示す。なお、本実験ではクアッドロータは飛行させず、床に置いた静止状態で目標点における位置を推定した。

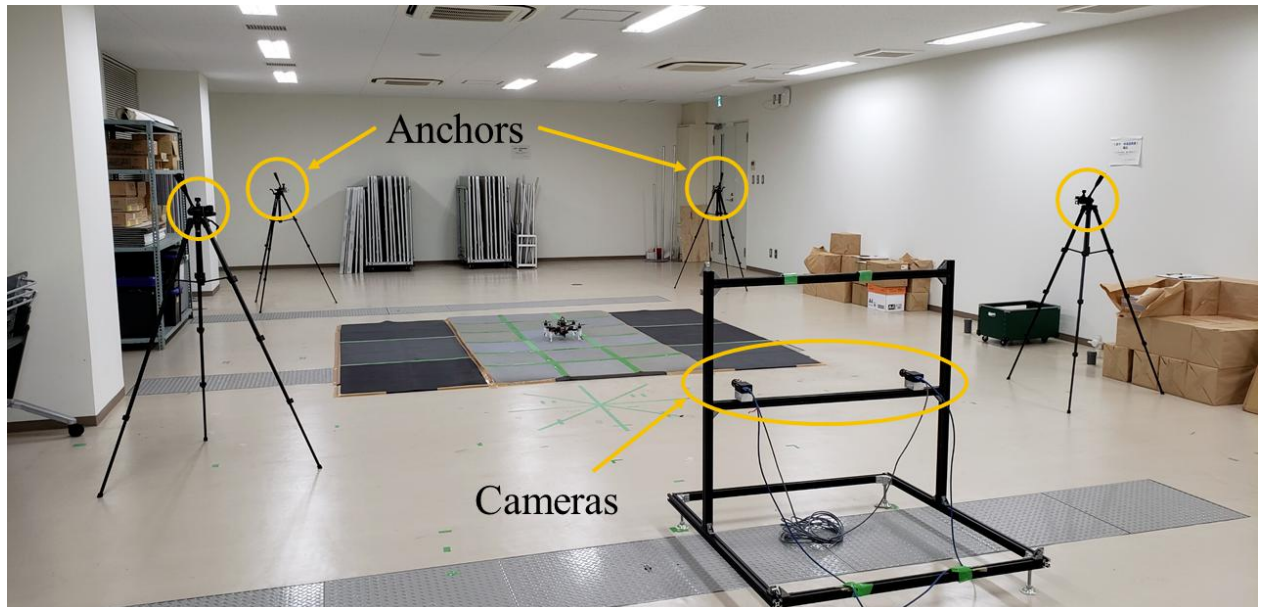


Fig. 40 Laboratory used for position estimation and position control experiments.

まず、クアッドロータの位置を推定する領域を取り囲む形で4つのUWBアンカをFig. 40に示すように設置した。アンカはFig. 41に示すようにカメラ用の三脚の頂点に固定した。また、アンカを設置した位置の絶対座標をTable 5に示す。



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

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

本実験での実験手順を以下に示す．まず，実験室の床に対し，目標点として与える点の真値を可能な限り精度良くマークした．次の Table 6 にその 4 点の目標値の相対座標を示す．なお，Fig. 40 において，実験領域に衝撃吸収用のウレタンマットが敷かれているが，この位置推定実験の際には敷かれていないため，マットは真値に対し影響を与えない．

Table 6 Relative coordinates 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

次にクアッドロータの左手前の足が床にマークした Point0 と重なるようにクアッドロータを床に置く．その際，プログラム内にて座標の並行移動を行い，その実験開始時にクアッドロータを接地した点を相対座標系における原点（Point0）としてキャリブレーションを行う．その点において，数十秒間クアッドロータを放置し，次の目標点（Point1）へと手で持って移動させる．Point1 においても同様に数十秒間放置した．この動作を他の残り 2 点でも同様に行い，各点の静止状態における相対座標を，前に示したオプティカルフローセンサを用いない従来の推定手法により推定した．その推定結果を次の Fig. 42 に示す．

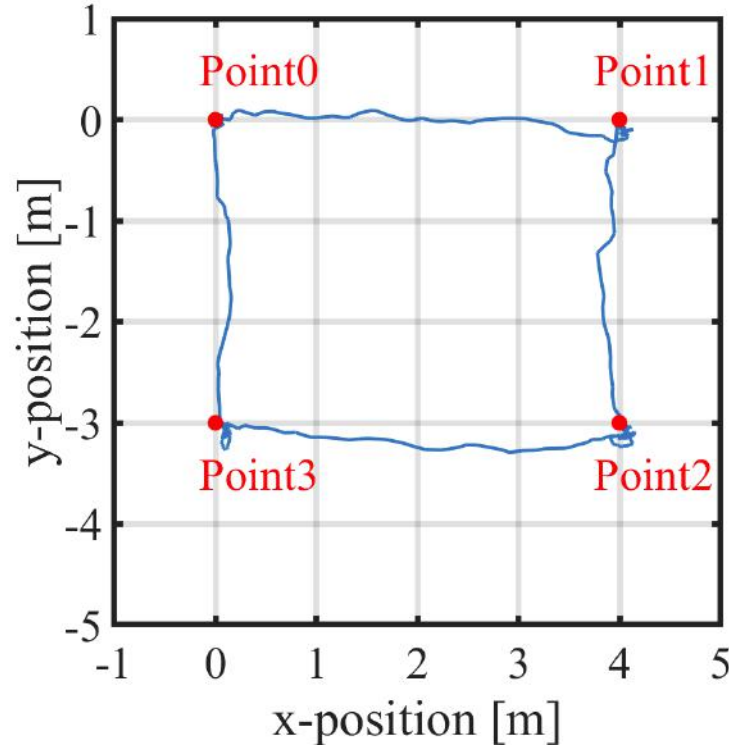


Fig. 42 Position estimation results when the quad rotor is stationary using the conventional estimation method.

図中における赤い点がそれぞれの目標点であり、青い線が推定結果を表している．初めにキャリブレーションを行った Point0 以外の 3 点において数 cm から数 10 cm ほどの定常偏差が生じているが、いずれの点においても誤差 1 m 以内の精度で位置を推定できていることがわかる．また、点間の移動の過程において、推定された座標の線が直線的でないのは、移動の際に目標となる基準線を与えず、目標点のみ移動の際の指標として与えたためである．目標点同士を線で結び、その上をトレースできれば、結果はより直線に近づくと考えられる．

4.2 センサの種類及びカルマンフィルタの分散値が位置推定精度に与える影響の考察

この章では先行研究において提案された手法を用いて位置推定を行った結果と、本研究において提案するオプティカルフローセンサも用いた手法を用いて位置推定を行った結果を数値シミュレーションを用いて比較する．また、後者におけるシミュレーションでは、カルマンフィルタにおけるセンサの分散値を変化させ、それが推定結果にどのような影響を与えるかも考察する．まず、シミュレーションに用いるためのセンサ値を実験の実験を通してロギングした．実験方法は床に描いた二等辺三角形の頂点及び線上を、台に固定したクアッドロータがトレースする形で動かし、その際のそれぞれのセンサ値をロギングした．ロギングしたセンサ値は IMU, UWB モジュール, 距離センサ, オプティカルフローセンサの値である．また、床面上の線をトレースするのは、クアッドロータに Fig. 43 のように取り付けたレーザポインタから光を下向きに照射し、それが線をなぞるように台を動かすことで実現した．

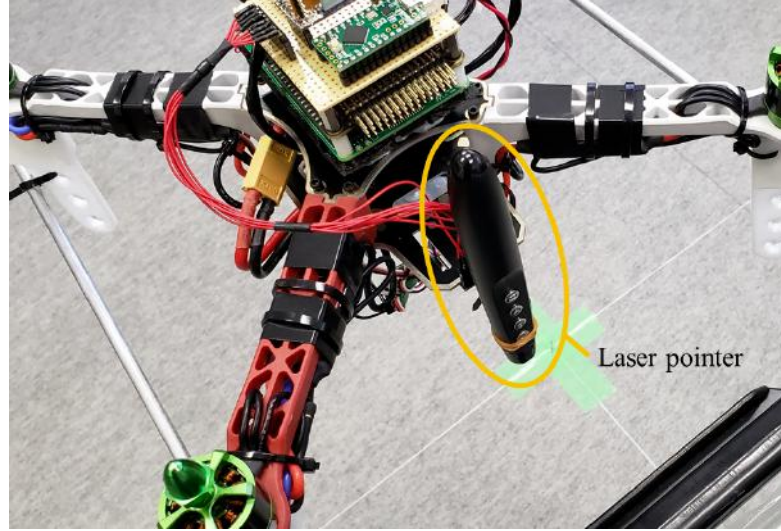


Fig. 43 Laser pointer attached for the purpose of tracing a line drawn on the floor.

また，設置した UWB アンカの絶対座標を Table 7 に，床面に描いた二等辺三角形の頂点の相対座標を Table 8 に示す．

Table 7 Absolute position of UWB anchors.

	x position [m]	y position [m]	z position [m]
Anchor1	-1.00	1.50	0.73
Anchor2	1.00	1.50	0.73
Anchor3	1.00	-1.50	0.73
Anchor4	-1.00	-1.50	0.73

Table 8 Relative coordinates of target points.

	x position [m]	y position [m]
Point0 (Origin)	0.00	0.00
Point1	1.00	0.00
Point2	0.00	1.00
Point3	-1.00	0.00

二等辺三角形の底辺の中点を相対座標系の原点とし，クアドロータを最初に設置した地点としてキャリブレーションを行う．クアドロータを Point0 から Point3 まで順に動かし，最後に Point0 に戻した．また，各点においてクアドロータを 10 秒間ずつ静止させた．以上の実験を通して集めたデータを数値計算ソフトである Matlab を用いて作製した位置推定プログラムに対して適用し，位置推定の計算をオフラインの環境で行った．ここで，拡張カルマンフィル

タを含むカルマンフィルタにおいて調整可能なパラメータは主に共分散行列の初期値及び観測方程式における各センサの分散値である．数値シミュレーションにおいて初期値は変えず，シミュレーションに使用するセンサの種類及び，それぞれのセンサの分散値を変化させ，その結果を見ていく．次の表に各シミュレーションにおいて使用したセンサの種類及び，分散値を示す．

Table 9 Type of used sensors and variance values.

	Accelerometer	UWB module	Optical flow sensor
Simulation1	0.4	0.02	-
	0.004	0.01	0.0006
Simulation2	0.4	0.02	-
	0.004	0.000001	0.06
Simulation3	0.4	0.02	-
	0.004	0.1	0.0006

シミュレーション1から3において，加速度センサ及びUWBモジュールのみの場合はそれらの分散値を変更せず，3種類のセンサを用いる場合のみ分散値を変更した．各シミュレーションにおける結果を次に示す．それぞれのシミュレーションにおいて，青色の線がUWBモジュールと加速度センサの観測値のみフュージョンしたもの，橙色はUWBモジュール，加速度センサ，オプティカルフローセンサの3種類をフュージョンしたもの，黄色はオプティカルフローセンサから得られた速度値を単純に積分して足し合わせたものである．なお，シミュレーション1においてはPoint2付近における拡大図も合わせて示す．

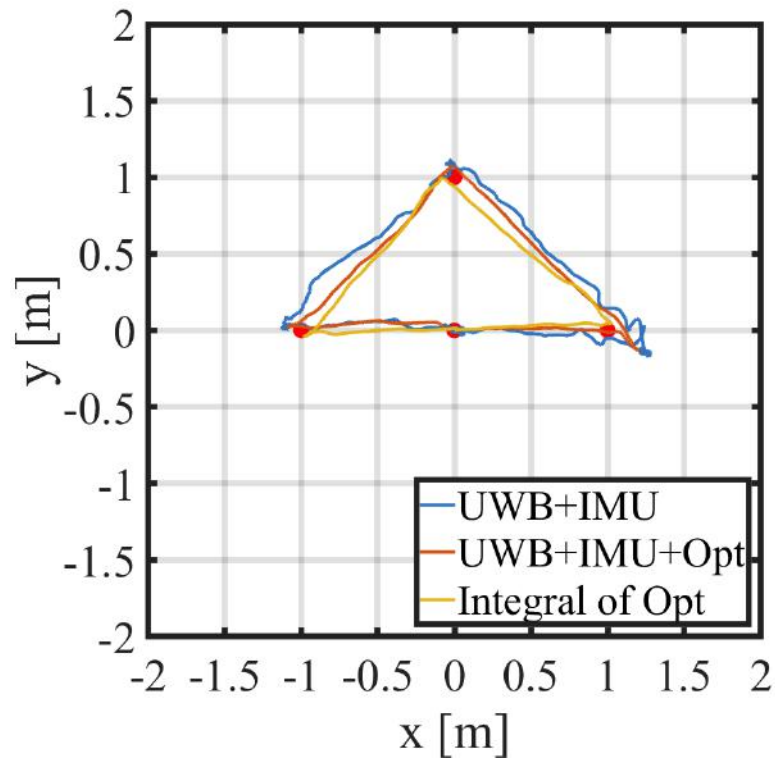


Fig. 44 Result of simulation1.

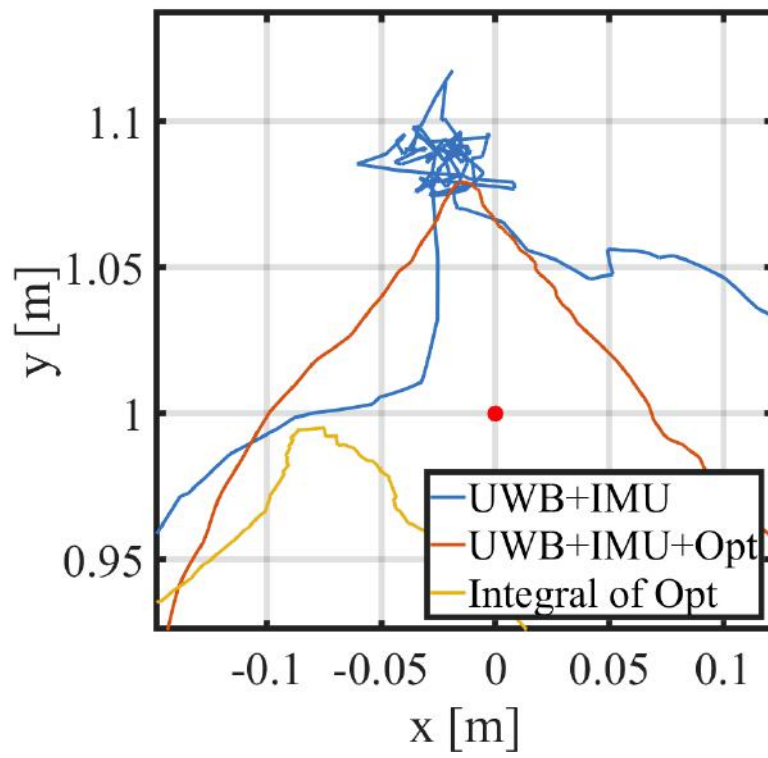


Fig. 45 Enlarged view of Fig. 44.

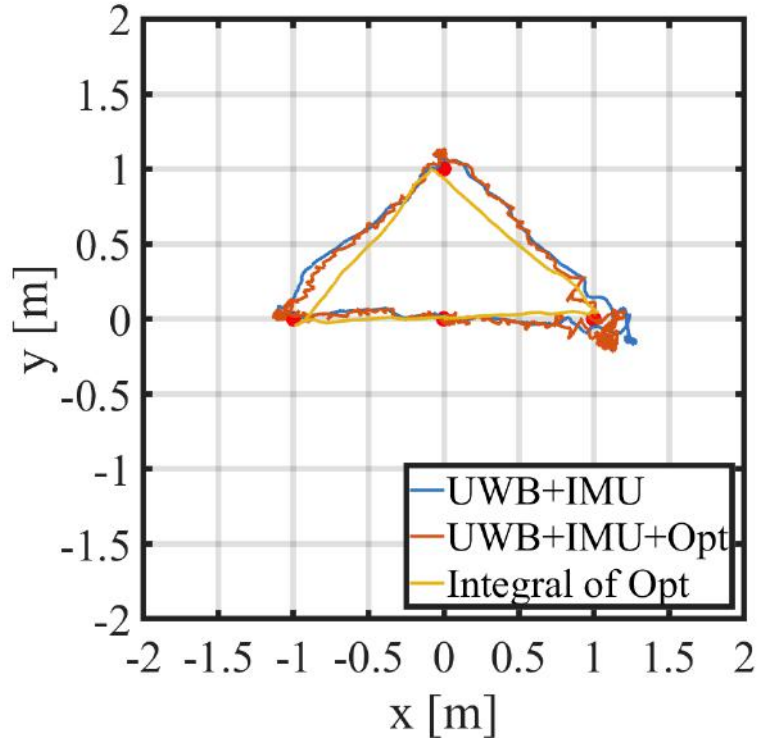


Fig. 46 Result of simulation2.

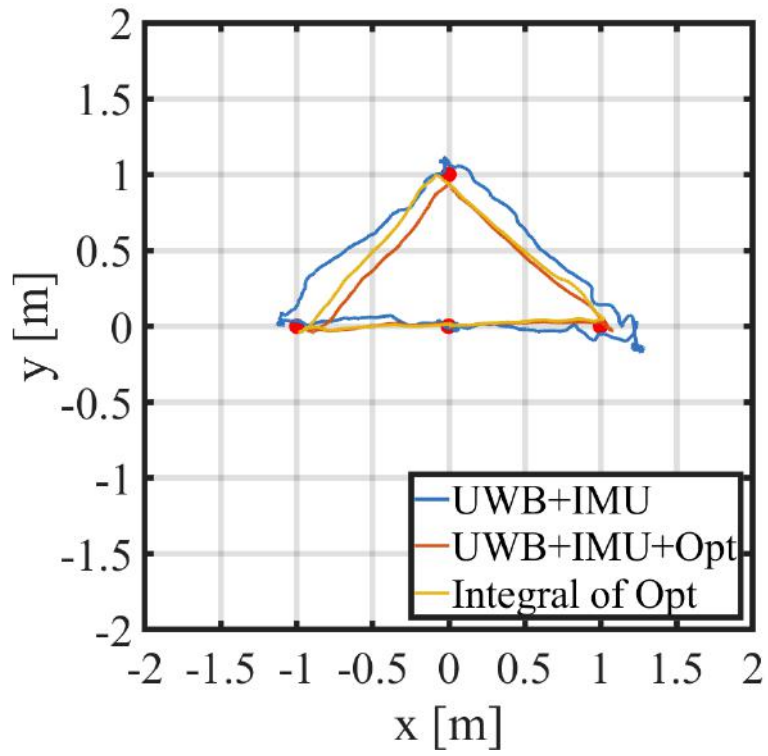


Fig. 47 Result of simulation3.

シミュレーション1は3種類のセンサの分散値をトライアンドエラーで調整し、オプティカルフ

ローセンサの値を積分した際に生じるドリフトやUWBモジュールの測距誤差が原因の推定値の分散が少なくなるようにして求めた結果である。また、シミュレーション2はUWBモジュールの分散値を小さくし、UWBモジュールから得られる観測値の影響を大きくした結果である。シミュレーション3はそれとは逆にオプティカルフローセンサの分散値を小さくし、オプティカルフローセンサから得られる観測値の影響を大きくした。次にそれぞれの結果について考察していく。まず、分散値の調整を行ったシミュレーション1であるが、青色の線は、目標点間の移動する過程において波打っていることがわかる。また、拡大図においても、目標点付近において青色の線は値の分散が生じている。しかし、橙色の線は移動中の過程も線が揺らぐことなく直線的であり、目標点付近においても分散が生じることなく点での曲がり角が明確に出ることがわかる。黄色の線は直線的であり分散も見られないが、目標点付近において他の2線とは離れた位置に定常値を取っている。青色のUWBモジュールと加速度センサのみを用いた場合には、UWBモジュールの測距値に分散が含まれることが原因で推定値にも分散が生じたが、橙色の線ではオプティカルフローセンサの効果によりその分散を打ち消しているため、こういった直線的で分散の少ない結果を得ることができた。また、黄色の単純積分値ではUWBモジュールを用いた座標の補正が効かないため、このように積分値がドリフトし、全体的に傾いた図形になった。次にシミュレーション2であるが、橙色の線はUWBモジュールの影響を大きくしたため、分散が大きくなりこのような直線的ではない結果となった。しかし、座標の補正はUWBモジュールの効果により働いているため、目標値の座標は青色の線と橙色の線で一致していることがわかる。最後にシミュレーション3においてオプティカルフローセンサの影響を強くした橙色の結果は黄色の線同様、滑らかではあるが、UWBモジュールの座標補正が弱くなり、ドリフトした結果となっている。これらのシミュレーション結果より、分散値の値を適切に調整した上で、既存の位置推定システムにオプティカルフローセンサの観測値も組み込むことにより、座標をUWBモジュールの観測値より推定しつつ、オプティカルフローセンサの効果で分散の少ない推定値を得ることが可能であるという結論を得た。

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．この章では実際にクアッドロータを飛行させ，提案した位置制御手法の精度を確認する．

5.1 位置制御手法

本節では Fig. 34 及び Fig. 37 中に示され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先行研究⁽⁵⁾やクアッドロータの位置制御に関する研究⁽²⁴⁾⁽²⁵⁾などの手法を参考にした．推定した座標と目標座標との偏差より仮想入力値を次のように設定する．

$$\begin{aligned} U_x &= k_{px}(x_d - \hat{x}) + k_{dx}(\dot{x}_d - \dot{\hat{x}}) + k_{ix} \int_0^t (x_d - \hat{x}) d\tau \\ U_y &= k_{py}(y_d - \hat{y}) + k_{dy}(\dot{y}_d - \dot{\hat{y}}) + k_{iy} \int_0^t (y_d - \hat{y}) d\tau \end{aligned} \quad (43)$$

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 (ϕ) 及びロール角 (θ) の目標値は以下の式により計算できる．

$$\begin{aligned} \phi^{des} &= \frac{1}{g}(U_x \sin \psi - U_y \cos \psi) \\ \theta^{des} &= \frac{1}{g}(U_x \cos \psi + U_y \sin \psi) \end{aligned} \quad (44)$$

これらの得た値をフ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，姿勢角が追従する流れである．またヨー角 (ψ) 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. 48 に示す VICON⁽²⁶⁾ などのモーションキャプチャシステムを用いることが多い．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. 49 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

カメラを用いた位置の計測システムを構築した．Fig. 50 に環境構築したステレオカメラシステムを示す．



Fig. 48 VICON⁽²⁶⁾.

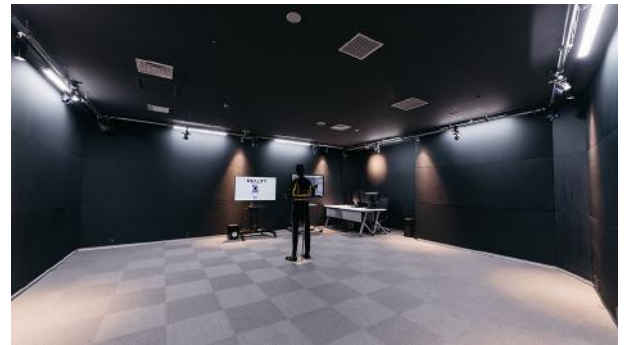


Fig. 49 The room where VICON is installed⁽²⁷⁾.

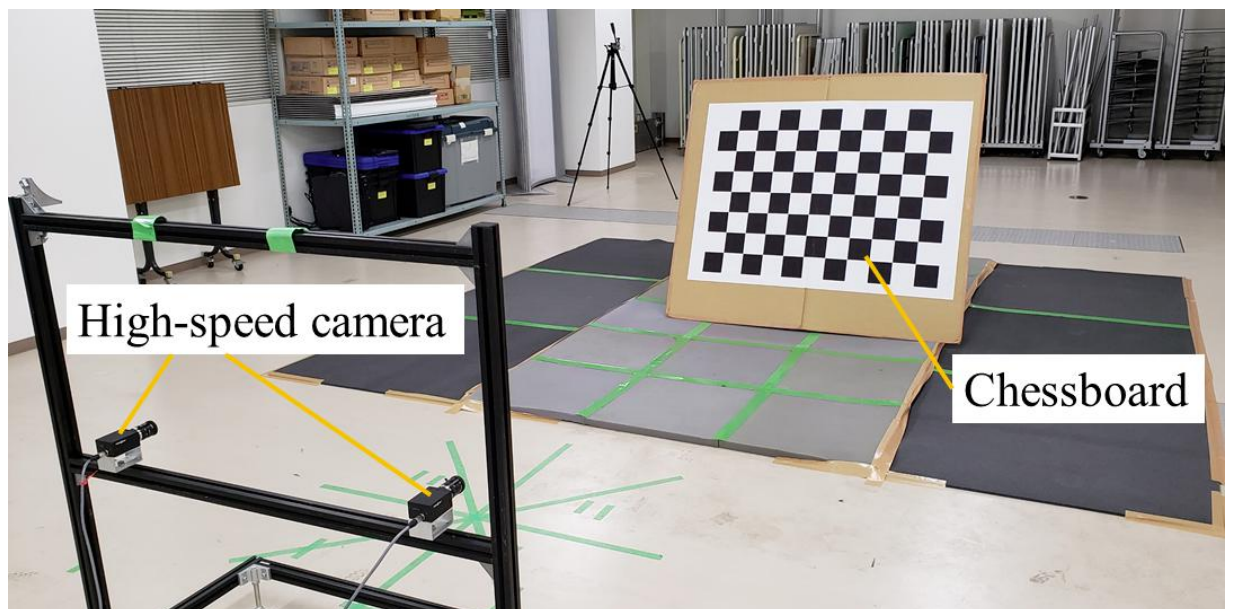


Fig. 50 Installed stereo camera and chessboard.

システムの大まかな構成は2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)⁽²⁸⁾である．Fig. 51 に外観を，仕様をTable 10 に示す．また，画像保存に使用したデスクトップPCの仕様をTable 11 に示す．



Fig. 51 High-speed camera(GS3-U3-32S4C-C).

Table 10 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

Table 11 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 10 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2048×1536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 11 に示すPCを用いてもカメラ2台を用いた状態で、1024×768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

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. 52, Fig. 53 に 2 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. 52 Old quad rotor with infrared LED.

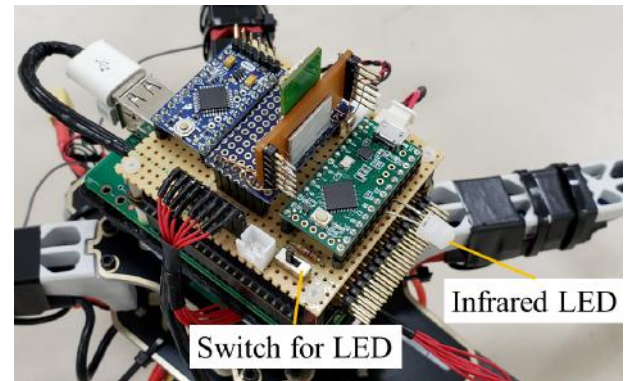


Fig. 53 New quad rotor with infrared LED.

先行研究から使われてきたクアッドロータ (Fig. 52) には 2 台のカメラ間で同期が取れるように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によって 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．しかし，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. 54 に示す赤外線透過フィルタ (IR-80) を装着した．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 Fig. 55 に示す．



Fig. 54 Infrared longpass filter.



Fig. 55 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先行研究⁽⁵⁾や詳解OpenCV⁽²⁹⁾に詳しい方法が載っている．Fig. 56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．

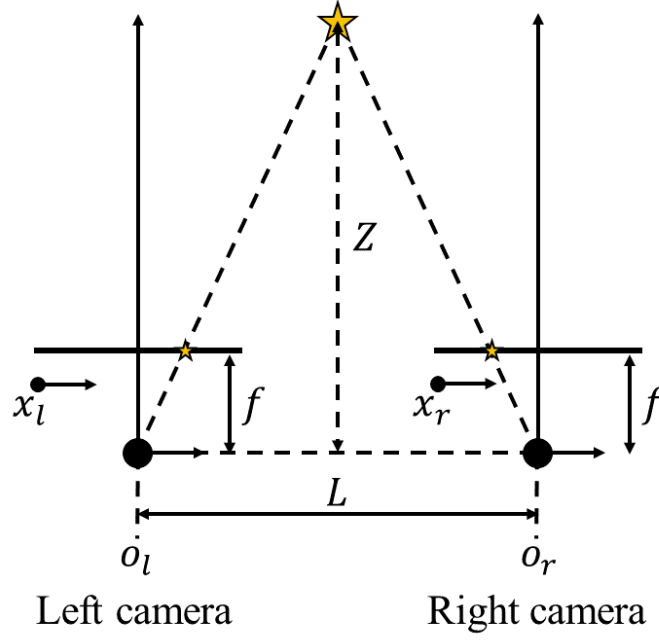


Fig. 56 Definition of stereo camera coordinates.

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，

$$d = x_l - x_r \quad (45)$$

と表せる．次に，図において三角形の相似の関係を用いると

$$\frac{L - d}{Z - f} = \frac{L - (x_l - x_r)}{Z - f} = \frac{L}{Z} \quad (46)$$

と表わせ，この式を変形することで

$$Z = \frac{fL}{x_l - x_r} \quad (47)$$

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

$$X = \frac{x_l L}{x_l - x_r}, Y = \frac{y_l L}{x_l - x_r} = \frac{y_l L}{x_l - x_r} \quad (48)$$

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. 50 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. 57 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像素子は、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である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．

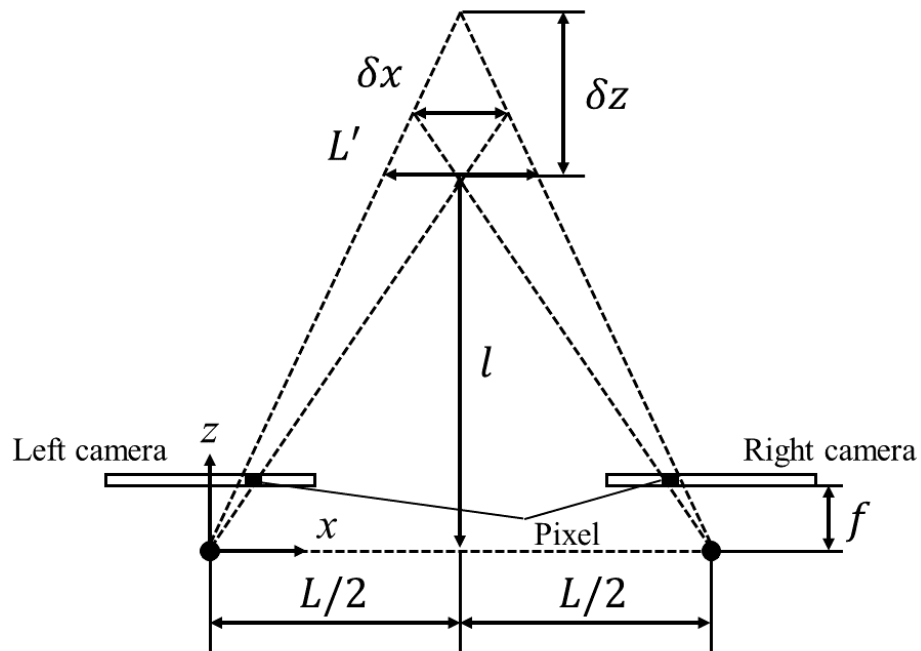


Fig. 57 Error in z direction of stereo camera.

図における δx が横方向誤差であり、 δz が奥行き誤差である．また、図に示す L' を計算のために導入する．まず、三角形の相似より

$$\text{Pixel} : \frac{L'}{2} = f : l \quad (49)$$

が言える．これを変形し

$$L' = \frac{2\text{Pixel}}{f} \quad (50)$$

となる．同様に相似より

$$\delta z = \frac{L'l}{L - L'} \quad (51)$$

が言える．式 (50) を式 (51) に代入し，整理すると

$$\delta z = \frac{2l^2\text{Pixel}}{fL - 2l\text{Pixel}} \quad (52)$$

となり，奥行き誤差が導出できる．同様に x 方向， y 方向誤差も相似より計算できるが， δz 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 δz 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 従来の推定アルゴリズムによる制御性能の評価

2.4 節にて示した従来の位置推定方法及び 5.1 節にて示した位置制御手法を用いてクアッドロータの位置を制御する実験を行った．次の 2 つの実験で用いたクアッドロータは Fig. 31 に示す従来機である．

5.3.1 定点における位置保持実験

ある目標点を目標値とし，クアッドロータが自律制御にてホバリングした際の精度を確認することを目的として，以下の実験を行った．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き，その位置が原点になるようにプログラム内で座標を平行移動させ，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マニュアル操作にて行った．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. 58 に示す．

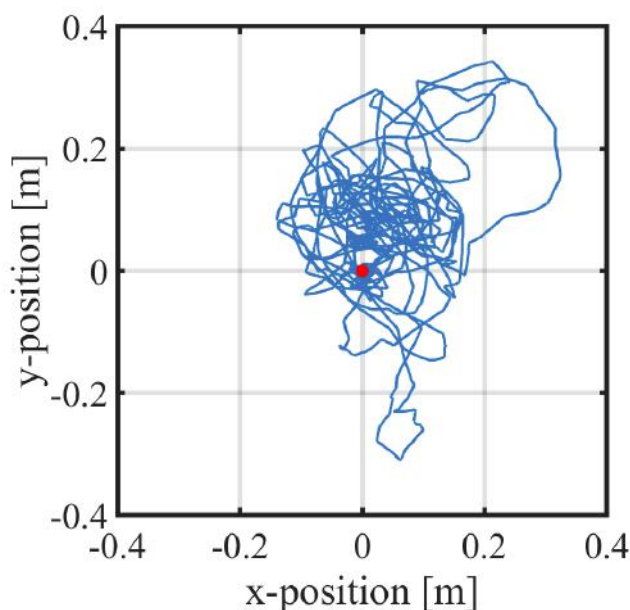


Fig. 58 Result of static position hold experiment.

図中の赤い点が目標値を示す．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である結果を得た．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ため，位置の発散が抑制され，このような結果を得ることができたと考えられる．

5.3.2 複数点間を移動する位置制御実験

次に複数の目標点を移動する位置制御実験を行った．本実験では複数の目標点を目標値とし，クアドロータが自律制御にてホバリング及び目標点間の移動をした際の精度を確認することを目的とした．使用したクアドロータは前述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．実験方法を以下に示す．まず，クア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．次に，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．ここまでの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 12 に，実験結果を Fig. 59 に示す．

Table 12 Absolute position of way points.

	x position [m]	y position [m]
Point0 (Origin)	0.00	0.00
Point1	1.50	-1.00
Point2	-1.50	1.00

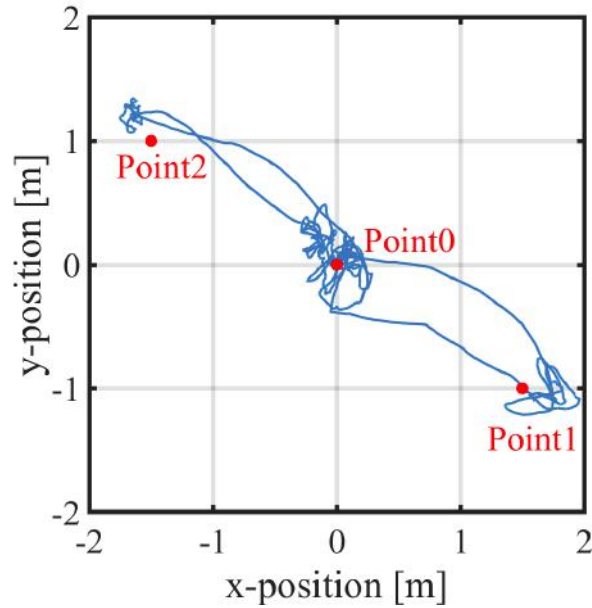


Fig. 59 Result of point to point experiment.

目標値が原点（Point0）の時，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．UWB アンカの設置誤差によって発生する位置推定誤差や UWB モジュールのモジュール間距離が近い場合に発生する測距精度の悪化などがこの偏差の原因であると考えられる．

5.4 提案する推定アルゴリズムによる制御性能の評価

次に，従来の推定方法及び，2.5 節にて示した新たな推定手法の 2 種類の推定方法により位置制御の実験を行うことで，2 つの推定方法による制御性能の違いを比較した．新たな推定手法では，オプティカルフローセンサの値も必要なため，用いた機体は Fig. 38 に示す新たに構築したクアッドロータである．実験内容は 5.4.1 節にて示したものと同様であり，1 点を目標地点とする位置保持実験である．実験方法も同様であり，2 種類の推定プログラムを実験試行毎に切り替えて位置制御を行った．まず，従来の推定手法による実験の結果 2 つを示す．

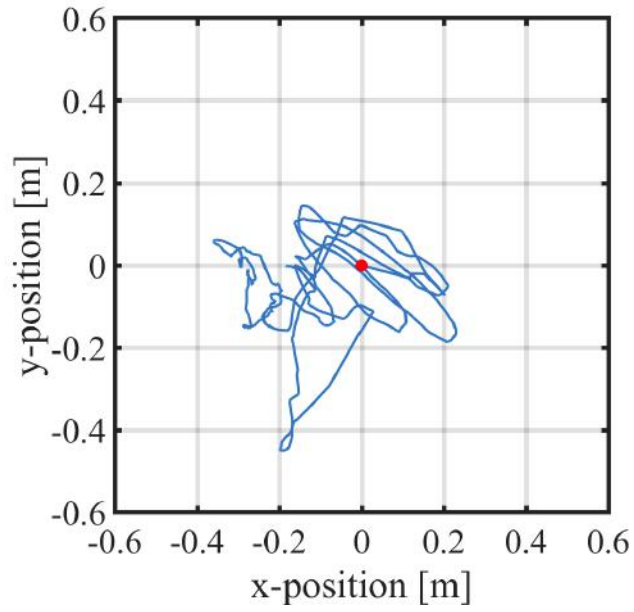


Fig. 60 Position control result using conventional estimation method (trial 1).

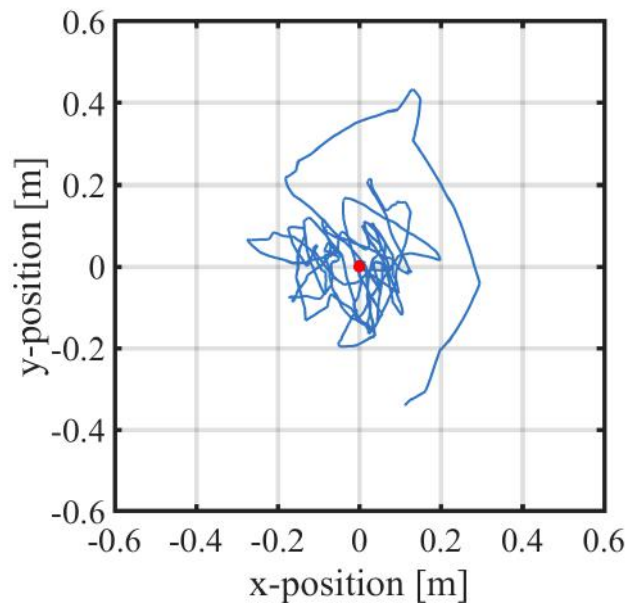


Fig. 61 Position control result using conventional estimation method (trial 2).

試行1では約18秒間、位置制御状態にて飛行していた。飛行制御の最後の方では大きく目標点から逸脱し、制御不能に陥ったため、制御を終了し、着地させた。試行2も同様に、位置制御を開始してから約28秒後には制御不能のため制御を終了した。次にオプティカルフローセンサも推定に組み込んだ結果を次に示す。

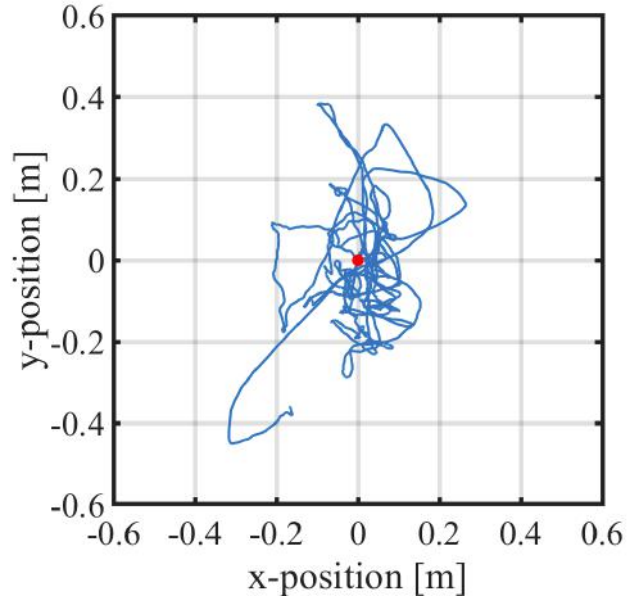


Fig. 62 Position control results using an estimation method incorporating an optical flow sensor (trial3).

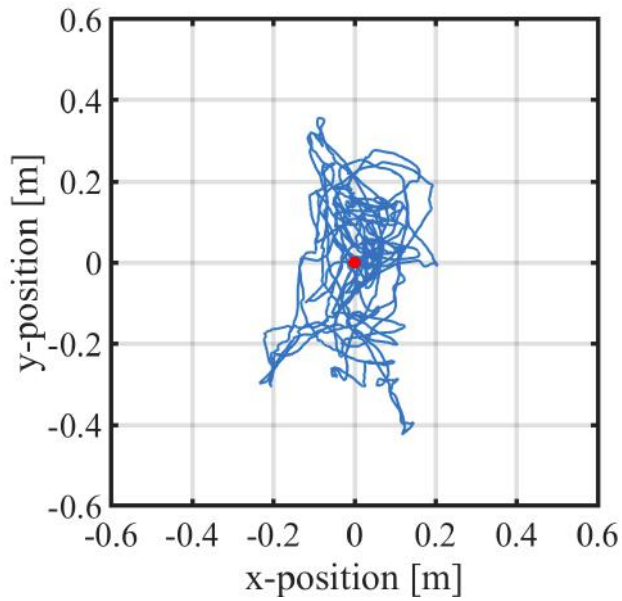


Fig. 63 Position control results using an estimation method incorporating an optical flow sensor (trial4).

試行3は約60秒間、制御状態で飛行していた。しかし、本試行においても飛行制御中に不安定状態に陥り、制御を終了した。試行4は約90秒間制御状態であり、意図的に制御を終了するまでの間、安定してホバリングを続け、最後まで制御不能状態にならなかった。試行4の結果から、5.3.1節と同様に偏差を計算したところ、目標値からの標準偏差が0.100 m、最大偏差が0.440 mとなった。

次にこれらの結果を考察していく。まず、従来の推定方法を用いた試行 1, 2 では安定した飛行制御が不可能であった。以前の機体の場合、5.4.1 節で示した実験のように飛行制御の開始時から終了時点まで目標地点から大きく逸脱することなく制御可能であったが、新たに構築した機体の場合、同じ推定方法を用いたにも関わらず、そのような結果は得られなかった。この原因は各ロータの推力不足によるものと、位置制御における PID ゲインの調整不足によるものと考えられる。次節にて示すが、実験に使用したプロペラとロータの組み合わせをスラストメータにて推力を計測したところ、その推力が十分なものではなかった。したがって、各ロータ及びプロペラから発生する推力が不十分であり、姿勢角制御の応答性が悪くなり、位置制御の精度も低下したと考えられる。また、位置制御における PID ゲインは本機体においては調整を行わず、以前の機体と同じ値を採用した。しかし、新たな機体は重量や発生推力が以前の機体と異なるため、そのゲインが適切ではなく、このような位置制御精度の低さにつながったと言える。次に、オプティカルフローも位置推定に組み込んだ結果である試行 3 と 4 であるが、この試行では試行 1, 2 と同じ機体を使用していたが、前半 2 回に比べて安定した飛行制御が可能であった。この理由は姿勢角応答の悪さをオプティカルフローセンサの推定周波数の速さが補正していたためであることや、UWB モジュールを及び IMU のみで行う推定精度の悪さをオプティカルフローセンサの値により補正できていたためであると考えられる。機体の姿勢角応答は以前の機体に比べ悪くなったが、オプティカルフローセンサも組み込んだことにより、クアドロータの運動に対する推定の追従性が向上し、後半 2 回の位置制御性能が向上したと考えられる。また、4.2 節にて示したようにオプティカルフローセンサが推定に組み込まれない場合は、推定値が UWB モジュールの測距誤差の影響で振動的になっていたが、オプティカルフローセンサが加わることでその振動を抑えられていた。したがって、本実験においてもそういった推定値の振動成分が抑制されたため、機体が推定値に追従して振動することがなくなり、前半 2 回に比べて飛行が安定していたと考えられる。ここで、5.3.1 節で示した偏差と試行 4 の偏差を比較すると、標準偏差、最大偏差ともに数 cm 誤差が増加しており、意図していた位置制御性能が向上した結果は得られなかった。しかし、前に述べたようなプロペラの選定やゲインの調節を適切に行えば、オプティカルフローセンサの効果により 5.3.1 節において得た結果よりも精度が向上した結果を得られると考えられる。

次に 5.2 節にて説明したステレオカメラを用い位置計測を行ったものと推定値の比較結果を示す。次に示す結果は上実験の試行 2 のものである。

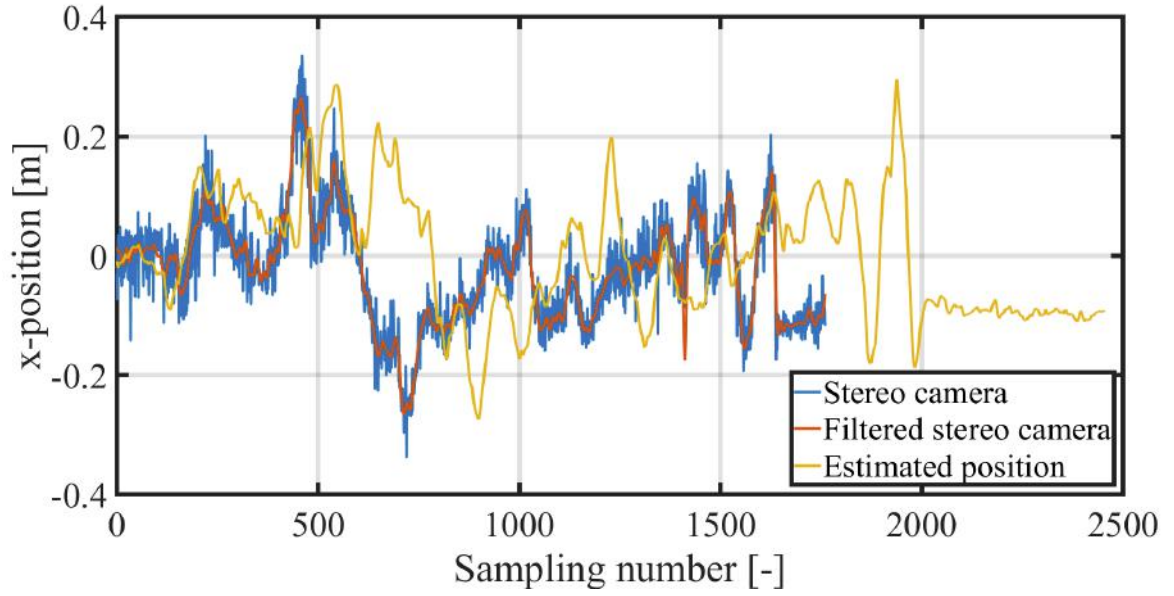


Fig. 64 Comparison of the position of the quad rotor measured by a system using a stereo camera with the estimated value.(x -axis).

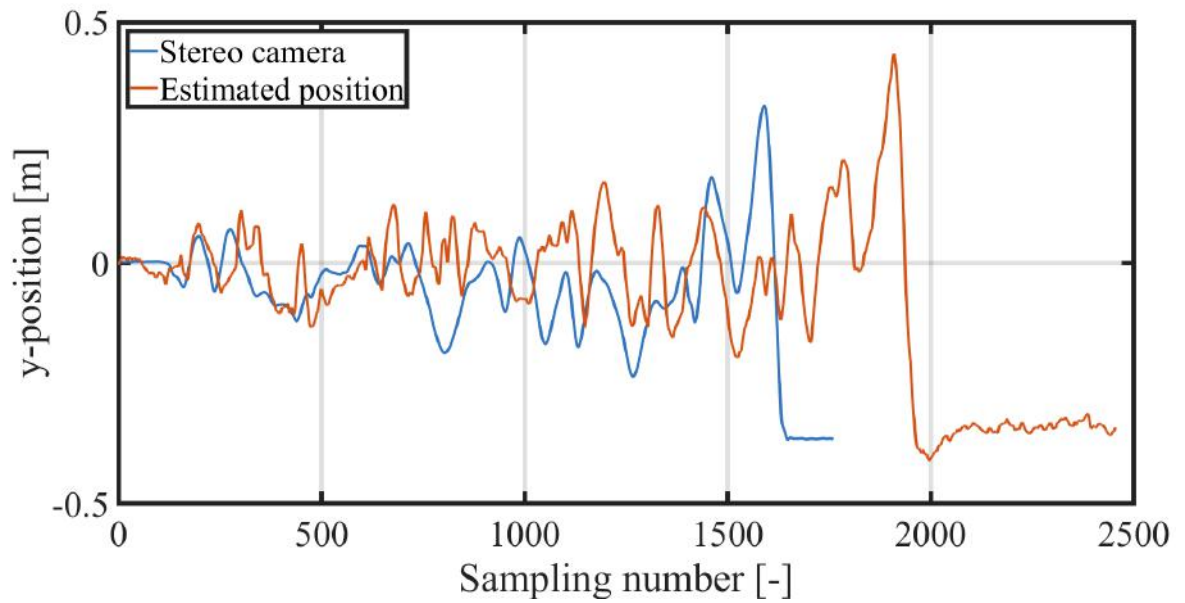


Fig. 65 Comparison of the position of the quad rotor measured by a system using a stereo camera with the estimated value.(y -axis).

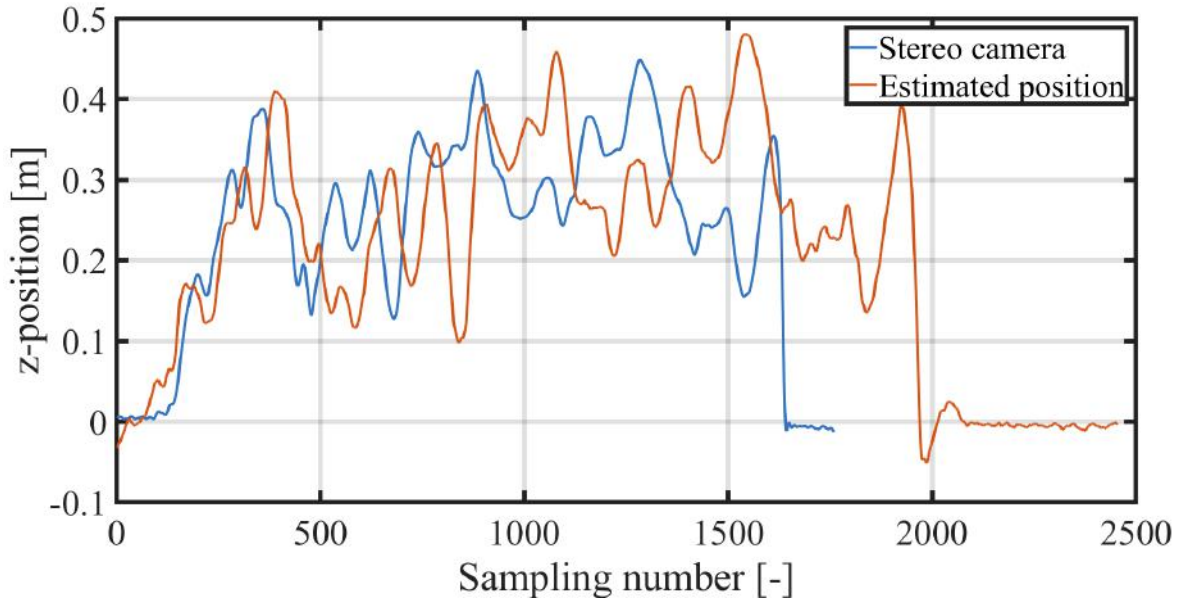


Fig. 66 Comparison of the position of the quad rotor measured by a system using a stereo camera with the estimated value.(z -axis).

計測値と推定値は時刻同期されておらず、サンプリングレートも異なるため、時刻をずらすことによりタイミングの同期を図った。しかし、推定値のサンプリング間隔が不均一なため、時系列データの初めの方ではタイミングが合っているが、最後の方でずれた結果となっている。また、結果からわかるように奥行方向である x 方向の計測値に対しノイズが乗る結果を得た。したがって x 方向の結果のみカットオフ周波数：3 Hz でローパスフィルタを掛けたものも合わせて示した。5.2.2 節にて示したとおり、ステレオカメラは特性上カメラから見て奥行方向に誤差が生じやすいため、このように x 方向にのみノイズが乗る結果となった。しかし、3 方向とも 2 つのデータに同期は取れていないが、近い値を取っているため、クアドロータに搭載した推定システムを用いて得られる推定座標は妥当なものであることが示された。

5.5 ロータの推力測定

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。理由は飛行に使用したプロペラは十分な推力が発生しておらず、姿勢角制御に悪影響を与えていると考えたためである。飛行実験に使用したプロペラ 1 及び予備用のプロペラ 2 を次に示す。



Fig. 67 Propeller1.



Fig. 68 Propeller2.

どちらも直径 8 inch, ピッチ 4.5 inch の設計である。しかし、図より翼面積に大きな差があることがわかる。次の図に推力測定実験時の様子を示す。

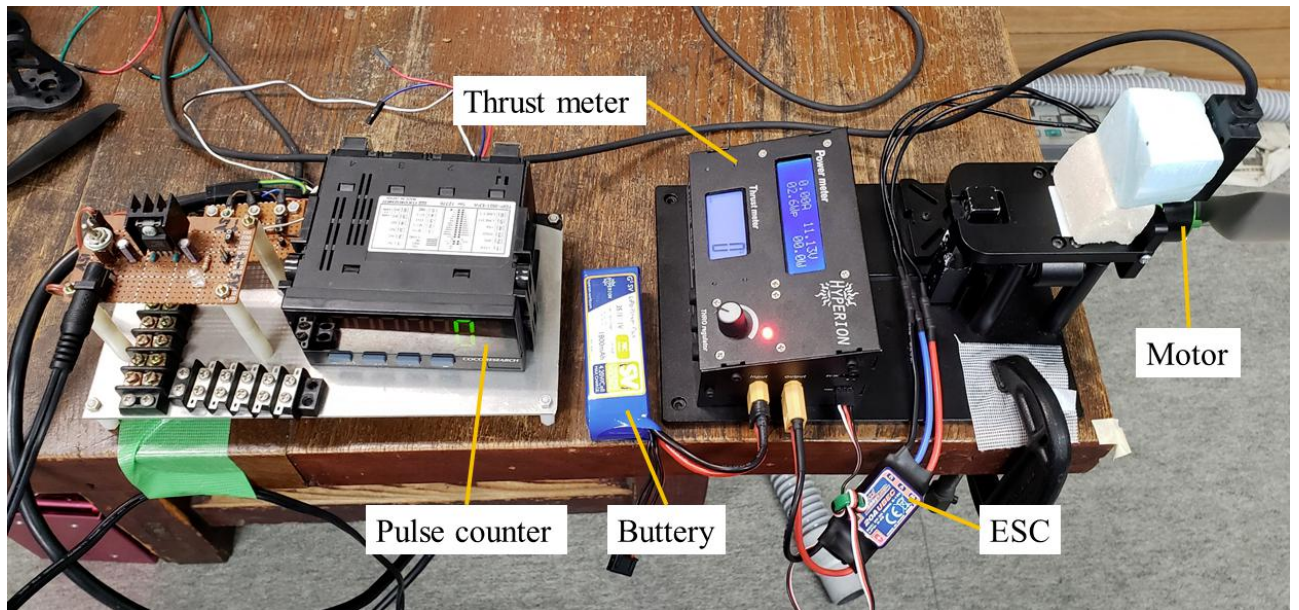


Fig. 69 Configuration for measuring propeller thrust.

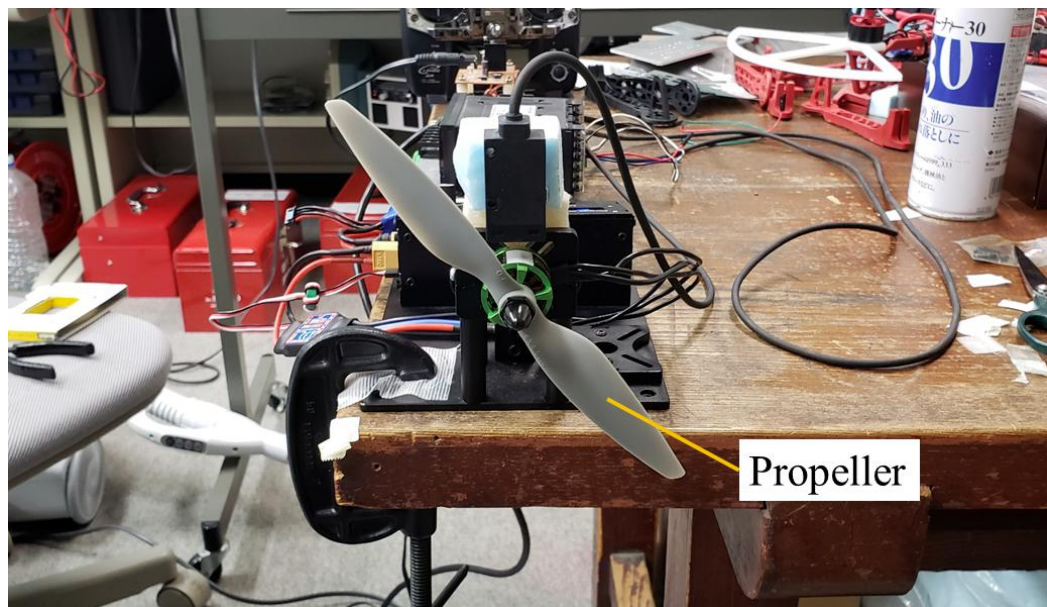


Fig. 70 Propeller fixed to thrust meter.

まず，スラストメータに対しプロペラが装備されたロータを設置する．次に各回転数になるようスラストメータからPWM信号をESCに送り，ロータを回転させる．ロータの回転数（rpm）は図中のパルスカウンタを用いて確認した．その際に各回転数における発生する推力をスラストメータから読み取った．推力は各回転数において3回ずつ計測し，それらの平均を取った．次の図に2種類のプロペラの各回転数における，計測した推力を示す．また，最小二乗法により近似した二次曲線も合わせて示す．

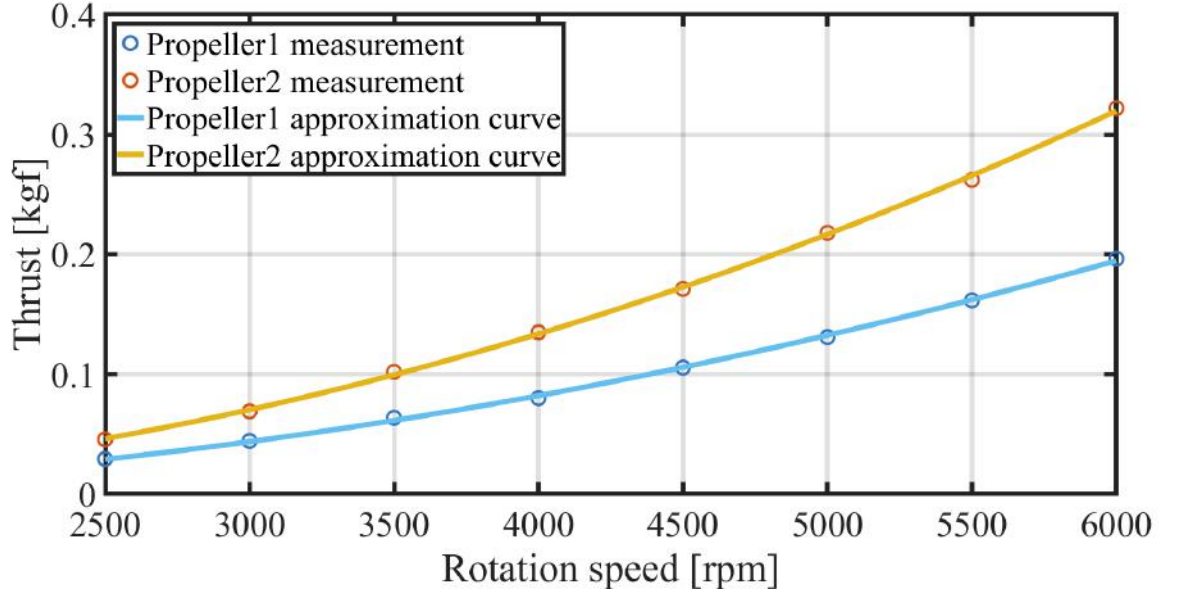


Fig. 71 Thrust measurement results of two types of propellers.

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 に比べて発生させる推力が低い．推力を f とし，回転数を ω [rpm] とすると，近似曲線の式はプロペラ 1 で

$$f_1 = 5.98 \times 10^{-9} \omega^2 \quad (53)$$

となり，プロペラ 2 で

$$f_2 = 9.94 \times 10^{-9} \omega^2 \quad (54)$$

となり，推力係数はプロペラ 2 がプロペラ 1 の 1.67 倍である結果を得た．直径，ピッチ共に同じ値に関わらず，このような結果となったのは翼面積の影響が大きいと考えられる．新たに構築したクアッドロータを用いて行った飛行制御実験において，同じ従来の推定手法を用いた場合であっても位置制御性能が低下したのはこの推力不足によるものが大きいと考えられる．各ロータでの推力が十分でなかったため，姿勢角制御の応答性が悪くなり，前の実験のような結果になったと言える．

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, それらに拡張カルマンフィルタを適用することで位置を推定した。また, オプティカルフローセンサを新たに搭載し, 位置推定及び位置制御の精度向上を図った。

まず, 先行研究において未達成であった位置制御に取り組んだ。先行研究において, 上記の UWB, IMU, 距離センサを用いた構成を用い, 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, 制御周波数が不十分であり, 位置制御飛行中においてクアッドロータの位置が発散していた。そこで, 新たに別のフ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, 演算負荷を軽くすることで, 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使用した位置制御では定点を目標地点とする位置保持制御及び3点間を移動する制御を行った。まず, 位置保持制御では, 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 である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, 安定した位置制御に成功した。

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センサを搭載し, 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, オプティカルフローセンサの値を参照した場合の方が位置推定精度が向上し, 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。次に, 新たな機体を用い, オプティカルフローセンサの値を参照した場合としない場合の2通りの条件において位置制御実験を行った。その結果, 同機体においてオプティカルフローセンサの値を参照しない場合よりも参照した方が位置制御性能が向上する結果を得た。しかし, 以前のクアッドロータにおいてオプティカルフローセンサの値を参照しない推定手法を用いた場合よりも位置制御の精度が向上することはなかった。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が不十分であったことや, プロペラから発生する推力が不十分なため姿勢角制御の応答性が悪かったことなどがこれの理由として考えられる。実際の飛行制御実験に用いたプロペラと予備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差があった。実験に使用したプロペラの推力係数は比較したプロペラの 0.6 倍であり, 実験に使用したプロペラの選定が不十分であったことがわかった。

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った。したがって, カメラの数を増やすことやチェスボードの改良等によるキャリブレーション精度向上などを行うことで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 つであり，計測可能な範囲や計測精度が限られていた．したがって，将来的には UWB モジュールの計測周波数が低下しない工夫をした上でアンカを増やし，計測範囲の拡大を図る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は感謝致します．加えて，学系横断型プロジェクトとして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新先生や防振システム工学研究室の三浦奈々子先生に心より感謝を申し上げたいと思います．最後に，クアッドロータの飛行実験を行うに当たって貴重な部屋を貸して下さいまし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, 道路構造物の現状 (橋梁),
http://www.mlit.go.jp/road/sisaku/yobohozen/yobo1_1.pdf, 2020/01/27 閲覧.
- (2) 山田健太郎, 木曽川大橋: トラス斜材の腐食による破断,
http://www.kougiken.jp/04_seika/pdfsystem/66_talkin/66_06.pdf, 2020/01/27 閲覧.
- (3) 日経コンストラクション, 日本の木曽川大橋も崩落に至る最悪のシナリオがあった,
<https://tech.nikkeibp.co.jp/kn/article/knp/20071127/513741/>, 2020/01/27 閲覧.
- (4) 大阪市立大学 ロボット工学研究室, 鋼橋検査ロボット・BIREM (研究紹介),
<http://www.robotics.mech.eng.osaka-cu.ac.jp/study/BIREM/BIREM.html>,
2020/01/27 閲覧.
- (5) 赤堀俊輔, 澤田祐一, 東善之, 鋼構造物点検用 UAV の飛行制御に関する研究, 京都工芸
繊維大学工芸科学研究科機械設計学専攻修士論文, (2018).
- (6) Microdrones,
<https://www.microdrones.com/en/>, 2020/01/27 閲覧.
- (7) ArduCopter,
<http://ardupilot.org/copter/index.html>, 2020/01/27 閲覧.
- (8) PHANTOM SERIES,
<https://www.dji.com/jp/products/phantom>, 2020/01/27 閲覧.
- (9) 株式会社自律制御システム研究所,
<https://www.acsl.co.jp/products/>, 2020/01/27 閲覧.
- (10) decaWave,
<https://www.decawave.com/products/dwm1000-module>, 2020/01/27 閲覧.
- (11) 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.
- (12) ElectronicDesign, What's The Difference Between Measuring Location By UWB, Wi-Fi,
and Bluetooth?
[https://www.electronicdesign.com/technologies/communications/article/
21800581/whats-the-difference-between-measuring-location-by-uw-b-wifi-and-
bluetooth](https://www.electronicdesign.com/technologies/communications/article/21800581/whats-the-difference-between-measuring-location-by-uw-b-wifi-and-bluetooth), 2020/01/27 閲覧.
- (13) Global Interface Technologies, UWB とは,
<http://www.git-inc.com/uwb.html>, 2020/01/27 閲覧.

- (14) DWM1000 Datasheet,
<https://www.decawave.com/wp-content/uploads/2018/09/dwm1000-datasheet-1.pdf>, 2020/01/27 閲覧.
- (15) スイッチサイエンス リチウムイオン電池 1800mAh,
<https://www.switch-science.com/catalog/2507/>, 2020/01/27 閲覧.
- (16) K. Guo, Z. Qiu, C. Miao, A. Zaini, C. Chen, W Meng, L. Xie, Ultra-Wideband-Based Localization for Quadcopter Navigation, Unmanned Systems, Vol. 4, No. 1, (2016), 23-34.
- (17) S. Lange, Niko Sunderhauf, P. Protzel A Vision Based Onboard Approach for Landing and Position Control of an Autonomous Multirotor UAV in GPS-Denied Environments, International Conference on Advanced Robotics, (2009), 1-6.
- (18) N. Gageik, M. Strohmeier, S. Montenegro, An Autonomous UAV with an Optical Flow Sensor for Positioning and Navigation, Int. j. adv. robot. syst., Vol. 10, (2013), 341.
- (19) Kim, J & Brambley, Dual Optic-flow Integrated Navigation for Small-scale Flying Robots, Australasian Conference on Robotics and Automation, (2007), 7.
- (20) mRobotics, Pixracer,
https://docs.px4.io/v1.9.0/en/flight_controller/pixracer.html, 2020/01/28 閲覧.
- (21) Federico Thomas and Llus Ros, Revisiting Trilateration for Robot Localization, IEEE TRANSACTIONS ON ROBOTICS, VOL. 21-1, (2005), 93-99.
- (22) D.E. Manolakis, Efficient Solution and Performance Analysis of 3-D Position Estimation by Trilateration, IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS, 32-4, (1996), 1239 - 1248.
- (23) PJRC Teensy LC,
<https://www.pjrc.com/teensy/teensyLC.html>,, 2020/01/28 閲覧.
- (24) Zongyu Zuo, Quadrotor Trajectory Tracking Control: A PD Control Algorithm, 2010 3rd International Conference on Computer and Electrical Engineering, 53-2, (2013).
- (25) Murray L. Ireland, Aldo Vargas and David Anderson, A Comparison of Closed-Loop Performance of Multirotor Configurations Using Non-Linear Dynamic Inversion Control, Aerospace 2015, (2015), 325-352.
- (26) VICON CAMERAS,
<https://www.vicon.com/hardware/cameras/>, 2020/01/28 閲覧.
- (27) REALITY Studio Roppongi
<https://le.wrightflyer.net/solution/studio.html>, 2020/01/28 閲覧.

- (28) エドモンド・オプティクス・ジャパン株式会社 USB3.0 カメラ FLIR GS3-U3-32S4 カラー
<https://www.edmundoptics.jp/p/gs3-u3-32s4c-c-1-18-inch-grasshopper-usb-30-color-camera/33122/>, 2020/01/28 閲覧.
- (29) 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.