

# クアッドコプタの制御のためのモーションキャプチャの利用

M2016SC013 奥山大介

指導教員：大石泰章

## 1 はじめに

近年、有人の航空機と比べて低リスク低コストで運用することができる無人航空機が注目され、空撮やインフラ点検、物資輸送などといった様々な用途での利用が考えられている。中でも特にクアッドコプタ型無人航空機（以下クアッドコプタ）は、構造が単純で比較的制御が容易であること、センサやバッテリー等の技術の発展により高性能なものが安価で入手できることなどの要因から急速に普及が進んでいる。しかし市販されている多くの安価なクアッドコプタは、既に制御器によって安定した飛行制御が行われており、その中身はブラックボックスであることが多い。一般的なクアッドコプタは、ジャイロセンサや超音波センサ、GPS等を用いて機体の姿勢や位置を測定し、フライトコンピュータ（FC）によって飛行制御を行っているが、制御の信頼性は使用しているセンサやFCの性能に大きく依存している。また、市販品は電圧や回転角速度といった制御入力ではなく上昇下降、前進後退などの方向指示しか与えられないこと、加速度センサや距離センサ等のセンサ出力が不明であることなどの問題点がある。そのため、以上のように様々な市販品に対して、クアッドコプタの性能に依存せずに適用できる制御システムを実現することは有用であると考えられる。

本研究では、既に飛行制御が行われている市販品のクアッドコプタを、機体に搭載されているセンサ類ではなく、モーションキャプチャを利用して制御することを考える。モーションキャプチャとは、観測する対象にマークを取り付け、複数台のカメラによって位置や姿勢を高精度に測定することのできるシステムであり、コンピュータグラフィックスやスポーツなどの分野で利用されることが多いが、近年では制御にも用いられ、成果が上がっている [1]。クアッドコプタに付属するセンサ類の性能に依存せず、機体外から機体の状態を取得するため、webカメラを用いた手法が提案されているが [2]、モーションキャプチャを利用することで、機体の状態をより高精度に測定して高度な制御を行うことができると考えられる。本研究では、モーションキャプチャを使用してクアッドコプタ飛行時のモーションデータの取得を行い、得られたデータからクアッドコプタのシステム同定を行う。また、得られたシステムを用いてシミュレーションを行い、実機実験と比較検討を行う。

## 2 使用する実験機

### 2.1 クアッドコプタ

本研究で使用するクアッドコプタは、ROBOLINK社のCoDroneである（図1左）[3]。CoDroneの仕様を表1に示す。このクアッドコプタはBluetooth4.0によってプロポ（無線送受信機）と無線通信することができる。プロ

ポにはArduino互換ボードであるSmart Inventor Boardが搭載されており、Arduino言語によるプログラミングが可能である。プロポからは上昇下降、前進後退、左右移動、旋回のコマンドを送信することができ、それぞれの値は-100~100で変化させることができる。離陸と着陸は離陸着陸コマンドの送信によって行い、コマンドによる離陸の後、入力がなければ一定高度を保ったまま自動でホバリングを行う。

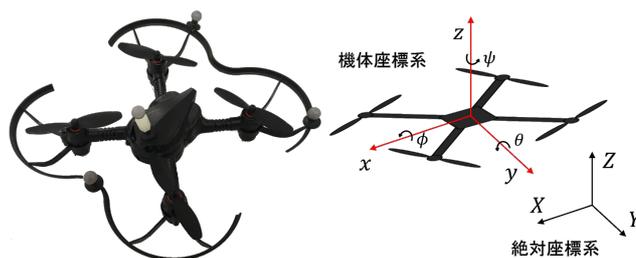


図1 左：マーカを取り付けたCoDrone；右：座標系

またCoDroneでは、プロポに搭載されているBluetoothボードをPCとUSB接続することで、PC上のソフトウェアから直接コマンドを送信することができる。具体的にはROBOLINK社のソフトウェアRokit Smart Brickを使用する。このソフトウェアは、フローチャートのようにブロックを配置することでプログラムを作成することができる。プログラムはPC上で動作し、接続されたBluetoothボードを介してクアッドコプタにコマンドを送信する。Arduino言語と違い、コンパイルやボードへの書き込みの作業が必要ないため、簡易なプログラムを作成する際は有用である。

表1 CoDroneの仕様

機体質量	37 g
大きさ	133 × 133 mm
ロータ軸から機体中心までの距離	47.5 mm
連続飛行時間	8分
充電時間	40分
通信規格	Bluetooth 4.0
内蔵センサ類	3軸ジャイロセンサ 気圧センサ オプティカルフローセンサ
バッテリー	3.7V 300mAh LiPo バッテリ

クアッドコプタの重心を原点として、機体の正面方向を正とする $x$ 軸、正面方向から反時計回りに90度の向きを正とする $y$ 軸、機体に対して上向きを正とする $z$ 軸によ

て機体座標系を定める。また、絶対座標系 ( $X, Y, Z$ ) を右手系の直交座標系で  $Z$  軸が鉛直上向きになるように定める (図 1 右)。絶対座標系の原点および  $X$  軸,  $Y$  軸はモーションキャプチャが定めるものを用いる。機体座標系における  $x$  軸まわりの回転角 (ロール角) を  $\phi$ ,  $y$  軸まわりの回転角 (ピッチ角) を  $\theta$ ,  $z$  軸まわりの回転角 (ヨー角) を  $\psi$  として機体の姿勢角を示す。

クアッドコプタに与えることができる入力として、機体座標系における  $x$  軸方向の入力を  $U_x$  と書き,  $y$  軸方向,  $z$  軸方向の入力をそれぞれ  $U_y, U_z$  と書く。また,  $z$  軸まわりの回転の入力を  $U_\psi$  と書く。いずれも値は  $-100 \sim 100$  の範囲である。

## 2.2 モーションキャプチャシステム

本研究で使用するモーションキャプチャシステムは NaturalPoint 社の OptiTrack であり, 使用するカメラは同社の Flex3 (図 2 左) である [4]。対象を囲むように 4 台のカメラを



図 2 左: Flex3 ; 右: システム概要

カメラを配置し, システムを構築する (図 2 右)。カメラと PC を接続し, NaturalPoint 社のソフトウェア Motive によって対象に取り付けられたマーカの  $X, Y, Z$  座標とソフトウェア上で定義した剛体の姿勢角を時系列で測定する。取得されたデータは  $0.01s$  ごとにサンプルされており, Excel 上で読み込みができる csv (Comma-Separated Values) 形式で出力される。

## 3 実機実験

クアッドコプタの横方向 ( $y$  軸方向) の動特性をシステム同定するために実機実験を行う。

### 3.1 実験環境

クアッドコプタを囲むように地上約  $3.5m$  の位置に 4 台のカメラを配置し, システムを構築する。次にソフトウェア上で剛体として定義を行うために, クアッドコプタの 4 箇所反射マーカを取り付ける。このとき, マーカの重量によってクアッドコプタのバランスが変化するため, クアッドコプタのずれの補正 (トリム調整) を行う。トリム調整は PC 上のプログラムによって行うことができ, 上下, 前後, 左右, 旋回の 4 つの値をそれぞれ  $-800 \sim 800$  の範囲で調整する。クアッドコプタはプログラムから Bluetooth でコマンドを送ることによって飛行を行い, PC 上のソフトウェア Motive によって飛行時のモーションデータの取得を行う。

## 3.2 実験手順

PC 上のプログラムによってクアッドコプタの飛行を行い, クアッドコプタの横方向の運動について測定を行う。具体的には, 初めにコマンドによる離陸を行い, モーションキャプチャのキャプチャエリア内に収まる地点でホバリング状態にする。ホバリング状態のとき, PC からクアッドコプタに与えられている入力は零である。次に,  $y$  軸の方向に入力  $U_y = 100$  を 1 秒間与えた後, 再び入力を零としてホバリング状態にする。すなわち, 以下の入力を与える:

$$U_y(t) = \begin{cases} 100 & (0 \leq t < 1), \\ 0 & (t \geq 1). \end{cases} \quad (1)$$

このときの一連の飛行をモーションキャプチャによって撮影し, モーションデータの取得を行う。得られたモーションデータでは, 4 つのマーカの重心位置と機体の姿勢角の時間変化に着目する。

## 3.3 実験結果

モーションキャプチャによって得られたクアッドコプタの姿勢角の時間変化を図 3 に示す。今回は横方向の運動について考えるため, ロール角  $\phi$  に着目する。グラフより, 与えられた入力によって  $1 \sim 2[s]$  付近でロール角が  $19[deg]$  になった後, 入力が零になり機体を静止させるために  $3[s]$  付近で  $-10[deg]$  に変化し, その後振動しながら  $0[deg]$  に収束していることがわかる。これにより, プロポや PC からの入力が零のとき, FC 上で姿勢角を安定させてホバリング状態にする制御が行われていると考えられる。

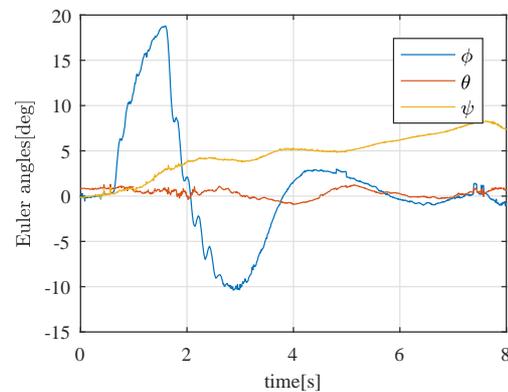


図 3 クアッドコプタの姿勢角の変化

## 4 システム同定

### 4.1 入力からロール角までの伝達関数の導出

3 章で得られたクアッドコプタの姿勢角のデータより, 横方向の入力  $U_y$  からロール角  $\phi$  までの伝達関数を導出する。プログラムによって送信した  $U_y$  を入力, クアッドコプタのロール角  $\phi$  を出力として Matlab のシステム同定ツールである System Identification Toolbox[5] を用いてシステム同定を行う。このツールは, 測定された入出

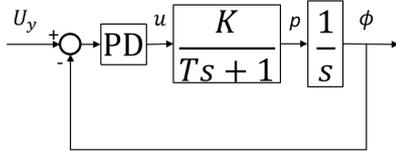


図4 ブロック線図

力データからシステムの数学モデルを最尤推定によって同定することができる。分母多項式を2次，分子多項式を0次として伝達関数モデルを導出したが，良好な結果が得られなかった。そのため，分母多項式を2次，分子多項式を1次として次の伝達関数モデルを導出した：

$$\phi(s) = \frac{0.3617s + 0.2412}{s^2 + 1.31s + 3.37} U_y(s). \quad (2)$$

この伝達関数の意味を考えるため，クアッドコプタの横方向の動特性が図4のブロック線図のようになっていると仮定する。すなわち，クアッドコプタの左右のプロペラの推力の差  $u$  が1次遅れ系を介してロール角速度  $p$  に伝達され，これを積分したものがロール角  $\phi$  になる。外部から加える横方向の入力  $U_y$  は  $\phi$  と比較され，その差からPD制御によって推力差  $u$  が決定される。このように考えたとき， $U_y$  から  $\phi$  までの伝達関数は，

$$\phi(s) = \frac{KK_d s + KK_p}{Ts^2 + (KK_d + 1)s + KK_p} U_y(s) \quad (3)$$

となる。ただし  $K_p$  はPD制御のPゲイン， $K_d$  はDゲインである。これは式(2)の伝達関数と同じ形である。以上より，CoDroneで使用されている制御はPD制御であると考えられる。

得られたモデルに対して，式(1)の入力  $U_y(t)$  を与えたシミュレーションを行う。シミュレーション結果と実機実験の結果をロール角  $\phi$  について比較したものを図5に示す。グラフ上の赤線はシミュレーションから得られたロール角を示し，青線は実機実験から得られたロール角を示す。グラフより，シミュレーションと実機実験が近

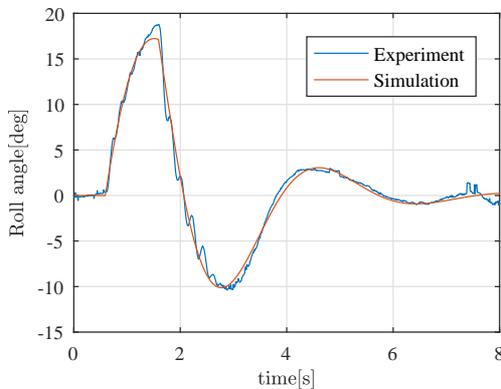


図5 ロール角  $\phi$  に関するシミュレーションと実機実験の比較

い応答を示していることがわかる。

## 4.2 ロール角から速度までの伝達関数の導出

4.1節と同じ実験で得られたロール角  $\phi$  を入力，機体の  $y$  軸方向の速度  $v_y$  を出力としてシステム同定を行い，ロール角  $\phi$  から機体速度  $v_y$  までの1次遅れ系の伝達関数モデルを導出する。得られた伝達関数は以下のようになった：

$$v_y(s) = \frac{0.1726}{s + 0.5754} \phi(s). \quad (4)$$

また，得られたモデルに対して，入力をロール角  $\phi$  の実験データとしたときのシミュレーション結果と実機実験の結果を機体の  $y$  軸方向の速度  $v_y$  について比較したものを図6に示す。シミュレーションの結果と実機実験の結果は十分近く，同定されたモデルが信頼できるといえる。

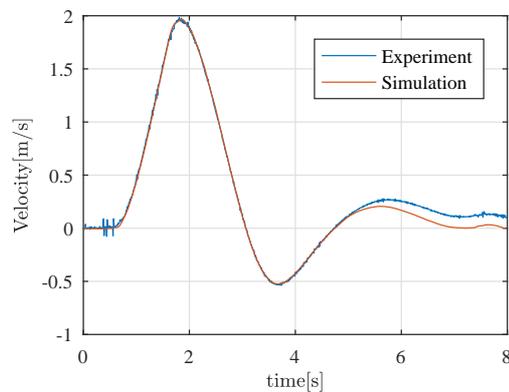


図6 機体速度  $v_y$  に関するシミュレーションと実機実験の比較

以上の結果から，クアッドコプタに対する横方向の入力  $U_y$  から機体の  $y$  軸方向の速度  $v_y$  までの伝達関数モデルが得られた。

## 5 シミュレーションと実機実験

現在，クアッドコプタの制御について，PID制御やLQ制御，モデル予測制御など様々な方法が議論されている[6, 7]。本研究では，4章で得たモデルに基づき，クアッドコプタの横方向の運動をフィードフォワードで制御することを考える。ここでの知見は今後フィードバック制御に発展させるうえで有用であると考えられる。本章では，4章で得られたモデルを用いてシミュレーションを行った後，実機実験を行い実際の挙動との比較検討を行う。

### 5.1 シミュレーション

3章と同様に，式(1)に従ってクアッドコプタの横方向に1秒間の入力  $U_y(t)$  を与え，飛行させることを考える。この入力をクアッドコプタに加えたとき，慣性によって入力を零とした位置で静止しないことがわかっている。そこで入力を工夫することで入力を零とする時刻  $t = 1$  の位置に収束させることを目標とする。機体を目標位置に素早く収束させるには，進行方向の逆向きに入力を入れる（当て舵）ことが有効であると考えられる。当て舵は航

空機や船において、操舵時の回転慣性を抑えるために一般的に用いられる手法である。逆向きの入力を考慮して設計したフィードフォワード制御の入力  $U_y$  を次に示す：

$$U_y(t) = \begin{cases} 100 & (0 \leq t < 1), \\ -69 & (1 \leq t < 2), \\ 0 & (t \geq 2). \end{cases} \quad (5)$$

式 (1), (5) で表される入力を加えたときの機体の  $y$  軸方向の位置に関するシミュレーションのグラフを図 7 に示す。

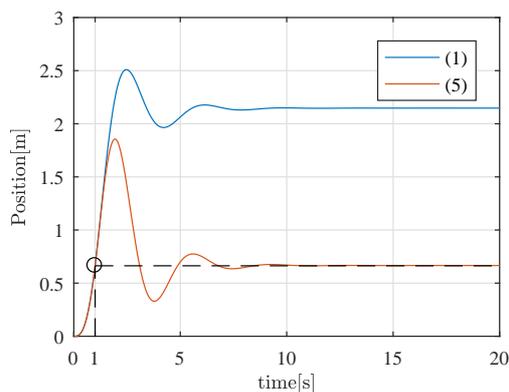


図 7 機体の  $y$  軸方向の位置に関するシミュレーションの比較

グラフの青線は式 (1) の入力を与えたときの応答を示し、赤線は式 (5) の入力を与えたときの応答を示す。また、グラフ中の○で示された場所は  $t = 1$  のときの機体位置を表す。グラフより、式 (5) の入力によって  $t = 1$  のときの機体位置に収束していることがわかる。また、この時の機体の  $y$  軸方向の速度  $v_y$  に関するシミュレーションの比較のグラフを図 8 に示す。グラフより、逆向きの入力を追加しても速度  $v_y$  が零に収束するまでの時間が変化していないことがわかる。

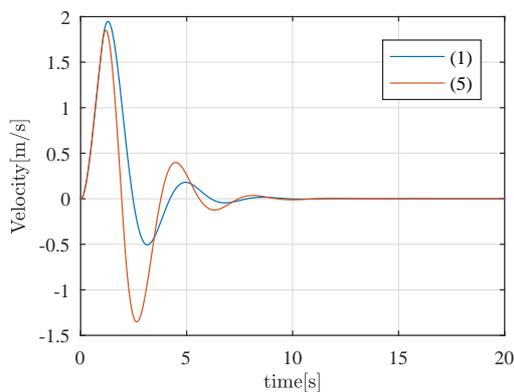


図 8 機体速度に関するシミュレーションの比較

## 5.2 実機実験

5.1 節のシミュレーションによって有効性が確認された式 (5) の入力を用いて実機実験を行った。機体の  $y$  軸方

向の位置の時間変化を図 9 に示す。グラフの青線、赤線はそれぞれ実機実験、シミュレーションを示す。シミュレーションでは  $t = 1$  のときの機体位置である約 0.7m の場所に収束しているが、実機実験ではそれよりも先に進んで約 1.8m の場所に収束した。しかし式 (1) を与えた場合の収束位置である約 2.2m よりも手前で静止した。このことから、シミュレーションと実機の差異はあるものの、入力を工夫することによって良好な結果が得られたと言える。

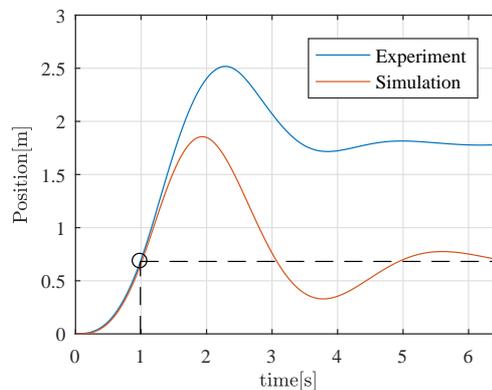


図 9  $y$  軸方向の位置に関する実機実験とシミュレーションの比較

## 6 おわりに

モーションキャプチャを用いて、クアッドコプタの運動に対するシステム同定を行い、伝達関数モデルを導出した。また、得られたモデルの信頼性の検証や、モデルを用いたフィードフォワード制御を行った。今後の課題として、フィードフォワード入力の最適化やフィードバック制御への応用が考えられる。

## 参考文献

- [1] 皆川佳孝・平田光男, 計測自動制御学会論文集, Vol. 49, No. 2 (2013), pp. 275–283.
- [2] 笠原暢・佐藤和也, 第 58 回自動制御連合講演会予稿集, 神戸, 2015 年 11 月.
- [3] ROBOLINK 社 web ページ, <http://www.robolink.com/>
- [4] NaturalPoint, OptiTrack, <http://optitrack.com/>
- [5] Mathworks, System Identification Toolbox, <https://jp.mathworks.com/products/sysid.html>
- [6] L. M. Argentim, W. C. Rezende, P. E. Santos and R. A. Aguiar, *International Conference on Informatics, Electronics and Vision*, Dhaka, Bangladesh, 17–18 May, 2013.
- [7] M. W. Mueller and R. D’Andrea, *European Control Conference*, Zurich, Switzerland, pp. 1383–1389, 17–19 July, 2013.