

VR ドローンシミュレータ Virtual Reality Drone Simulator

○出雲崎周平¹, 山口健², 吉川浩²*Shuhei Izumozaki¹, Takeshi Yamaguchi², Hiroshi Yoshikawa²

Abstract: In this study, we create a drone simulator for practice because drone control is difficult. The created simulator reproduces physical behavior of actual drones. Also, we use Head Mount Display to improve immersion into a virtual space.

1. まえがき

近年ドローンを活用した産業や娯楽が増加している。しかし、ドローンの操作は難しく安定飛行が出来るまでに多くの時間がかかるという問題がある。本研究では、Unity^[1]を用いて仮想空間の作成と飛行ドローンの物理挙動をシミュレートし、1人称視点で操縦のトレーニングを出来るようにする。また、Head Mount Display(HMD)と呼ばれる、映像の奥行や立体感を再現できるディスプレイを活用して没入感を向上させる。

2. システム概要

本システムの概要を Fig. 1 に示す。HMD の Oculus Rift^[2]にドローン視点の映像を表示することで、没入感や視認性を向上させる。操作端末は実際のドローンのコントローラと同様に無線通信、アナログスティックによる制御が可能な Oculus Touch^[2]を使用する。これはアナログスティックの前後左右への傾きをそれぞれ 0 から 1.0 の範囲で検出することが可能なため、段階的な操作が可能である。操作は Fig. 2 の様に行う。本研究で作成するドローンは、機体の四隅にプロペラを持ち、上昇下降、前後左右、左右回転が可能で物である。

3. ドローンの物理挙動の再現

3.1. 上下移動

現実のドローンが高度を変更するためには、プロペラの回転速度を変化させることで行う。しかし、本研究で使用する Unity では気体の概念が無いいため、プロペラの回転による風で、揚力を発生させることができない。よって、Fig. 3 の様にプロペラの直上に力を加え、この力を調整することで上昇下降を再現する。

F_{fr} , F_{fl} , F_{br} , F_{bl} をそれぞれ右前, 左前, 右後, 左後のプロペラに加わる力とする。上昇および下降操作が行われていない時、ドローンが同じ高度を保って飛行できるように重力と同様の力を加える。このとき1つのプロペラに必要な力 f_h の大きさは m を質量、 g を重力加速度として以下の式によって実装する。

$$f_h = \frac{1}{4}mg \quad (1)$$

コントローラの上昇および下降操作によってこの力を上下させることでドローンの高度の変更を行う。1

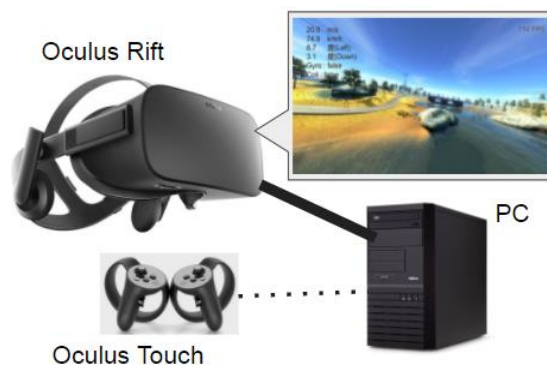


Figure 1. System overview

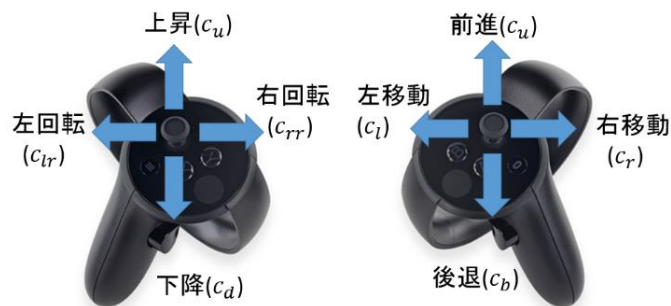


Figure 2. Drone Control

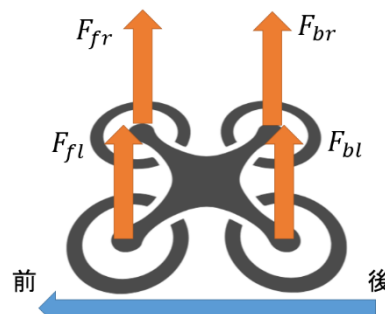


Figure 3. Force of drone

つのプロペラに加わる力 F は f を力の定数、 c_u をコントローラの上昇操作量、 c_d を下降操作量として以下の式によって実装する。

$$F = f_h + f(c_u - c_d) \quad (2)$$

3.2. 水平移動

現実のドローンが水平移動する場合は、機体を傾けて推進力を得る。Fig. 4 の様に前に進む場合、前2つのプロペラの回転速度を下げ、後ろ2つのプロペラの回転速度を上げることで、合力位置の変化から前に機体が傾き、ドローンの上昇に用いている力の分力で推進

力を得る．本研究では 4 つのプロペラに加わる力を個別に制御することで再現する． c_f, c_b, c_l, c_r をそれぞれ前後左右へのコントローラ操作量として，以下の式を用いて実装する

$$F_{fr} = F + f(-c_f + c_b + c_l - c_r) \quad (3)$$

$$F_{fl} = F + f(-c_f + c_b - c_l + c_r) \quad (4)$$

$$F_{br} = F + f(c_f - c_b + c_l - c_r) \quad (5)$$

$$F_{bl} = F + f(c_f - c_b - c_l + c_r) \quad (6)$$

また，Unity 上に実装されている空気抵抗を用いてドローンの速度に終端速度を設ける．終端速度 V_{max} は空気抵抗を k ，傾いた角度を θ ，4 つのプロペラの合力を F_{sum} とすると以下の式で算出できる．

$$V_{max} = \frac{F_{sum} \cos \theta}{k} \quad (7)$$

本研究では，ドローンレースや撮影などに用いられている小型軽量の一般的なドローン^[3]を参考に，終端速度が 90 km/h となるように力を調整した．

3.3. 回転運動

現実のドローンが回転する時，Fig. 5 の様に回転したい方向と逆向きに回っているプロペラの回転速度を速くして，もう一方を遅くする．その結果，プロペラの回転部分に反力が発生して機体が回転する．本研究では，Fig. 5 の様に反力に該当する力をプロペラ的位置に発生させることで機体の回転を行う．このとき発生するトルク T は反力の大きさを f_i ，機体の中心からプロペラまでの長さを L ，左右への回転操作量を c_{rr}, c_{lr} とすると以下の式で算出できる．

$$T = L f_i (c_{rr} + c_{lr}) \quad (8)$$

本研究では，一般的なドローンを参考に最大回転速度を 20 rpm 程度になるように反力を調整する．水平移動中は回転軸が傾いているため，円滑に曲がる事が出来ない．そこで，ユーザの操作によって機体を水平にする必要がある．

4. チェックポイントシステム

仮想空間に設置されたリング状のチェックポイントを順番に通過してゴールを目指し，掛かった時間を表示するシステムを作成する．ユーザが指定されたルートを通って飛行できるかを評価することができ，ドローンの練習に活用することが出来る．

5. 結果

5.1. ドローンシミュレータの作成

Unity を用いてドローンシミュレータを作成し，基本動作の実装とチェックポイントシステムを用いたコースの作成を行った．実行画面を Fig. 6 に示す．没入感

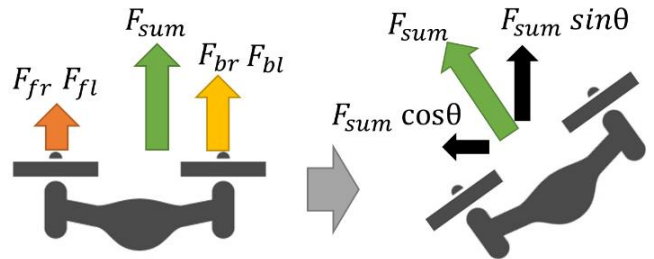


Figure 4. Horizontal movement of drone

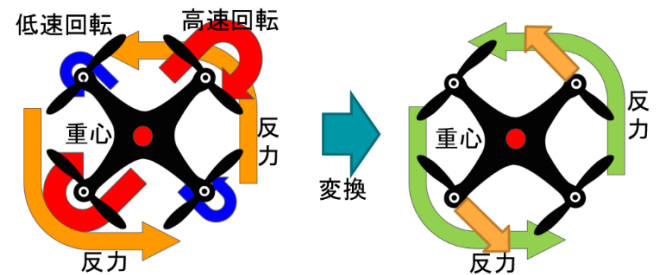


Figure 5. Rotating drone



Figure 6. Execution screen

向上のため HMD を使い，操作は Oculus Touch から行えるようにした．また，現在速度や機体の傾き等を UI として表示することでドローンの物理挙動を数値的に確認できるようにした．

5.2. 物理挙動の作成

現実のドローンと同じ操作方法で，ドローンの高度の変更，前後左右への水平移動，左右への回転をできるようにした．上下移動を行っていない場合は，重力と同様の力を加えることで高度を維持するようにした．水平移動と回転運動では，指定した終端速度および回転速度を再現することが出来た．

6. むすび

本研究では Unity を用いて，ドローンの物理挙動を実装した．今後の課題として，ドローンの操縦の難しさを再現するために，風や正確な空気抵抗の実装や，新しいデバイスを活用した没入感の向上などが考えられる．

7. 参考文献

- [1] “Unity” <http://japan.unity3d.com/>
- [2] “Oculus Rift” <https://www.oculus.com/rift/>
- [3] “Vortex 285” <https://www.immersionrc.com/fpv-products/vortex-racing-quad/>