マルチホップ通信を用いた群ドローンの位置制御の研究

情報工学科 5年19番
情報通信研究室
坂本惇一郎
指導教員 (主)田中晶、小嶋徹也、松崎頼人
発表 1月31日 (金)

目次

はじめに

先行研究ではマルチホップ通信を用いたドローンの制御を行うことでより広範囲のドローンに対して制御命令を送ることができていた.

今研究では更に群ドローン制御を実用に耐えうるものにするためにデッドレコニングを用いて求めたそれぞれのドローンの位置情報をもとにより細分化された制御ができるようにする.

今回は先頭ドローンの初期位置からの移動量を後続のドローンに共有することで先頭ドローンの動きを後続ドローンで再現することを目指す.

研究概要

ドローン (1)

今回の研究で使用するParrot社製「AR.Drone 2.0」

研究概要

ドローン(2)

本研究ではプログラムでの制御のしやすさや，実際に制御時の安定感から Parrot社製の「AR.Drone 2.0」を使用する.

Wi-Fiを利用した制御では，プログラムからUDPで接続することでAPIにアクセスできる．SDKはnode.jsとpythonで書かれているものがメジャーであり，実際に動作することを確認し.

バッテリーは一台当たり15分ほど持続して使用することができる.

研究概要

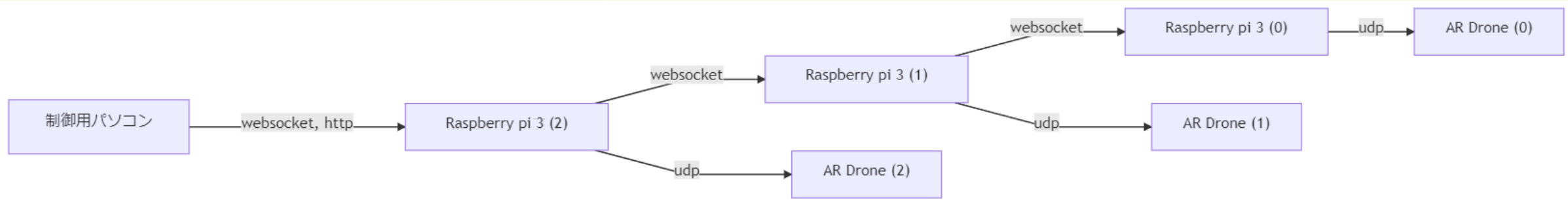
マルチホップ通信 (1)

マルチホップ通信とは，複数の端末がリレー方式でデータを中継ぎしながら通信する方式であり，これによって，山間部など直接では電波が届きにくい距離の端末と通信をすることができるようになる
またマルチホップ通信を用いて複数台のドローンを強調して制御することで産業運屋での利用が期待される

研究概要

マルチホップ通信 (2)

今回の実験での全体の構成図
Raspberry Piをアクセスポイントとして設定し，後続のRaspberry Piを接続することでマルチホップ通信を実現する. AR.Drone2.0とRaspberry pi3間の通信はUDP通信、Raspberry pi3同士間での通信は双方向通信が可能なwebsocketを用いた.



研究概要

マルチホップ通信 (3)

今回はRaspberry piのAP化にhostapd, dhcpにdnsmasqを使用した

研究概要

センサデータ (1)

今回使用するドローンはUDP通信を通してAPIにアクセスすることで，バッテリーの残量や，加速度センサ，ジャイロセンサなどの値にアクセスすることができる．AR.Drone 2.0との通信クライアントはサードパーティー制のOSSであるnode-ar-droneを使用した．

研究概要

センサデータ (2)

以下に実際にAR.Drone 2.0からセンサデータを取得するコードを記述する. このようにセンサデータの取得がコールバック関数によって実行できるため今回はpythonではなくnodejsを採用した.

```
var arDrone = require('ar-drone');

var client = arDrone.createClient();

client.on('navdata', (navdata)=>{
    console.log(navdata);
    // ここでnavdataを用いてドローンの初期位置からの変位を計算する.
}
```

研究概要

デッドレコニング (1)

センサデータから取得できる速度の値はローカル座標であるため姿勢角を用いてグローバル座標に変換する必要がある。
ドローンからはpitch, yaw, roll形式で姿勢角が得られるため回転行列を用いて行列の変換を行う

例) x軸を中心にθ角座標を回転させる行列

$$Rx(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\theta) & -sin(\theta) \\ 0 & sin(\theta) & cos(\theta) \end{bmatrix}$$

研究概要

デッドレコニング (2)

以下の式で与えられた姿勢角を用いてグローバル座標をローカル座標に変換

$$p' = R_z(\mathrm{yaw}) \cdot R_y(pitch) \cdot R_x(roll) \cdot p$$

姿勢角： $\mathrm{roll}, \mathrm{pitch}, \mathrm{yaw}$
回転行列： Rx, Ry, Rz
ローカル座標 ： p
グローバル座標 ： $p\prime$

研究概要

デッドレコニング (3)

以下の式にドローンから与えられた値を入れることでグローバル座標が算出できる
グローバル座標での移動量: p
回転行列 : R_X, R_y, R_z
姿勢角： $\mathrm{roll}, \mathrm{pitch}, \mathrm{yaw}$
微小時間: dt, 速度: vx, vy, vz

$$p = \left(R_z(\mathrm{yaw}) \cdot \mathrm{R_y}(\mathrm{pitch}) \cdot \mathrm{R_x}(\mathrm{roll})\right)^{-1} \cdot \begin{bmatrix} vx \cdot dt \\ vy \cdot dt \\ vz \cdot dt \end{bmatrix}$$

研究概要

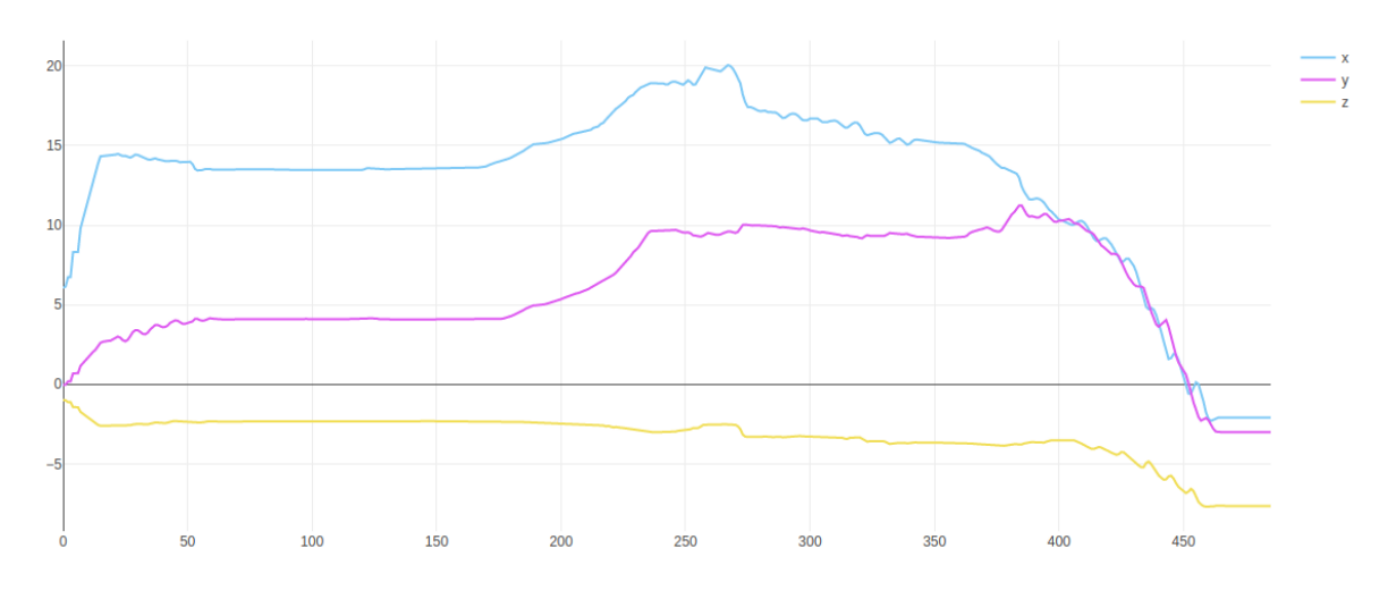
位置制御

今回は先頭ドローンの移動量の変化を後続のドローンが再現できることを目指す.このとき,まずそれぞれのRaspberry piは対応するドローンからセンサデータを取得する度の位置を計算する.さらに一定時間ごとに,マルチホップ通信を行うことで先頭ドローンの位置情報を取得し現在のドローンの位置情報との比較を行い差分を調整するための制御命令をドローンに対して送信する.

実験結果

デッドレコニング

デッドレコニングによって実際に取得した初期位置からの変化量を図4に示す.縦軸がメートル, 横軸がセンサデータの取得回数,青,赤,黄の軸がそれぞれx, y, z軸を示している.



まとめ,考察

センサデータには一定のノイズが乗っていたためにドローンが動かしていないときに変位が増加することや姿勢角が変化するなどの問題があった。

しかし、実験結果で示すように実際にセンサデータを用いて初期位置からの変化量を計算することができた。

今後の課題と解決策

センサデータの取得、グローバル座標の算出までは実装したが実際に協調動作するところの評価実験がまだできていないので引き続き実験をする予定

今回は特に姿勢角のノイズの影響が大きかったためより信頼性を向上させるためには地軸センサなどを用いたセンサフュージョンといった手法を用いることがよいと考えられる.

今後このデッドレコニングを応用することでより複雑なドローンの制御、農業など各種産業分野でのドローンの利用が期待される

参考文献

[1] 澁田叡知,マルチホップ通信による群ドローン制御の研究,H28年度東京工業高等専門学校情報工学科卒業論文,2017.2
[2]AR.DRONE 2.0 POWER EDITION, https://www.parrot.com/jp/doron/parrot-ardrone-20-power-edition (参照 2019/6/28)
[3] node-ar-drone, https://github.com/felixge/node-ar-drone (参照 2019/7/5)
[4] hostapd https://w1.fi/hostapd/ (参照 2020/1/29)
[5] dnsmasq https://web.chaperone.jp/w/index.php?dnsmasq (参照 2020/1/29)
[6] M5Stackで学ぶ「9軸センサ制御」, http://watako-lab.com/2019/01/01/m5stack_sens_9axis/ (参照 2020/1/30)