

Kinect を用いた AR.Drone2.0 の操作方法の提案

小林大輝[†] 下川和辰[†] 清水哲也[†]

サレジオ工業高等専門学校 情報工学科[†]

1. はじめに

現在、遠隔操作型ロボットを操作することによって、人間が環境的・地形的に立ち入ることの出来ない場所での作業が可能となっている。その際、地上を移動するロボットでは地形による障害を受けやすいため、使用するロボットは飛行型のものが好ましいと考えられる。また、ロボットを操作する際にボタン等により入力を行うコントローラを用いる場合、直感的に操作出来るようになるまである程度の訓練が必要だと考えられる。吉田らは[1]、ジェスチャにより小型飛行ロボットを直感的に操作する手法を提案している。しかし、吉田らの提案したジェスチャによる操作システムでは飛行ロボットの飛行速度は変化せず、飛行が上下左右前後の6方向に限られる。そこで本研究では、吉田らの提案したシステムをより改良したシステムを提案する。

2. AR.Drone の基本操作システム

吉田らの研究では身体性を拡張し、Parrot 社製の AR.Drone1.0 を疑似的に身体の一部とすることで、自分の手を伸ばしているような感覚や、幽体離脱して遠くへ飛んでいるような感覚を提示するシステムを提案している。手の拡張としては、Kinect を使用し手の位置によるジェスチャを認識し、PC を通してそのジェスチャに対応する操作指示を AR.Drone1.0 に送信する。また、視点の拡張として、HMD を使用して AR.Drone1.0 から送信された映像を見ながら操作できるようにしている。

2-1. AR.Drone2.0

本研究では Parrot 社製の小型飛行ロボット AR.Drone2.0(以下 AR.Drone) [2]を使用する。AR.Drone は Wi-Fi 経由で操作できる4ローターヘリコプターである。加速度計、ジャイロスコープ等を搭載しており、AR.Drone1.0 よりも安定して飛行し、カメラもより高画質のものになっている。この AR.Drone の X 軸、Y 軸、Z 軸の値を制御して飛行制御を行う。

2-2. Kinect

人間の動作等を認識するデバイスとして、Microsoft 社が販売する XBOX360 用 Kinect [3]を使用する。Kinect に搭載されている赤外線カメラによって人間の骨格とその座標を認識し追跡することで、操作者がどのような動作をしているのかを判断する。

3. システムの問題点と改良提案

吉田らが提案した AR.Drone1.0 の操作システムは、飛行方向が上下左右前後の6方向に限られ、飛行速度も一定である。そのため斜めに飛行する際は一度旋回を行ってから前後左右への飛行をするか、前後左右を交互に繰り返して飛行する必要がある。そのため、正面の物体を撮影したままでの飛行に大きな制限と操作の熟練が必要となる。さらに、円や弧のように飛行する際にも前後左右への動きを細かく繰り返す必要がある。また、飛行速度が一定であると、あらかじめ決められた速度に適した空間以外での飛行が大変困難になると考えられる。

改良システムは、前後左右・上昇下降・左右旋回の動作に加え、右斜め前後・左斜め前後への飛行を可能にした。また、飛行速度についても様々な空間に対応できるようにそれぞれの動作で2段階に速度が変化するように改良した。

3-1. 実験内容

縦 9.8m×横 12.6m の空間に図1のように障害物、PC と Kinect を配置する。AR.Drone の操作者は Kinect の正面に立ち操作を行う。AR.Drone は開始地点で離陸し終了地点で着陸をする。離陸後は右の障害物の上を通り一周八の字飛行し、これを①コントローラ、②吉田らのシステム、③改良システムの三種類の操作方法によって行う。

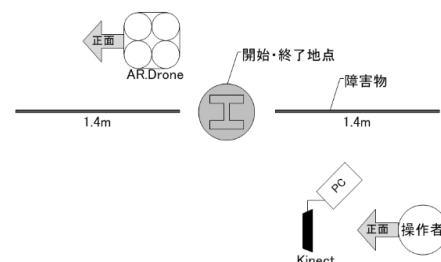


図1. 実験環境図

Proposal of operation method for AR.Drone2.0 using Kinect
[†]Daiki Kobayashi, Kazutoki SHIMOKAWA, Tetsuya SHIMIZU
 Computer Science & Technology, Salesian Polytechnic

3-2. 実験結果

図2に①, ②, ③の三種類による AR.Drone の操作の飛行経路を示す. 各操作手法による飛行経路の違いを明確にするために以下のようにして比較を行う. AR.Drone の向きを正面(図では左上)に固定し, 八の字飛行をする操作を行った. また, 飛行経路の違いを分かりやすくするために, 高度は考慮せずに上から見た平面図とする. 一番色の濃い線が①による操作で, 斜めへの飛行が可能で飛行速度は一定である. その次に濃い線が②による操作で, 斜めには飛行せず飛行速度は一定である. 一番薄い線が③による操作で, 斜めへの飛行が可能で飛行速度は2段階に変化する.

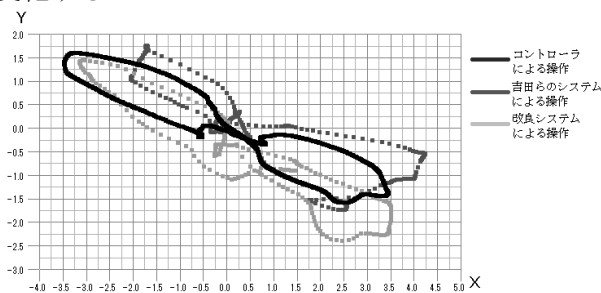


図2. AR.Drone の飛行経路

②による飛行経路と①による飛行経路を比較すると, 前後左右への飛行に限られる②では, 円のように切り返しをする部分で図のような角ついた飛行になってしまうことがわかる. また, 楕円のような軌道での飛行が出来ず直線飛行になっていることがわかる.

次に, ③による飛行経路と①による飛行経路を比較する. 図左上の飛行経路を見ると, 斜めへの飛行が可能になったことにより安定した切り返しが可能になったことがわかる. しかし, 飛行速度が変化することによる誤操作が発生した場合, 大きな修正が必要になる.

最後に, それぞれ3つの操作による飛行経路を比較すると, 表1より飛行経路の①との誤差の和を求めると, ②の誤差は2.42m, ③の誤差は1.22mとなる. よって②よりも③の方が前後左右へのずれが少なく, より①の飛行経路に近いことがわかる.

表1. コントローラによる飛行経路との誤差

	左極端 [m]	右極端 [m]	上極端 [m]	下極端 [m]
①	-3.5	3.5	1.68	-1.62
②	-2.1	4.25	1.80	-1.77
③	-3.2	3.51	1.5	-2.35
①-②	1.4	0.75	0.12	0.15
①-③	0.3	0.01	0.18	0.73

4. 結論

斜めへの飛行を可能にした改良システムは, 前後左右のみの飛行に限られる吉田らのシステムよりもコントローラでの飛行経路に近い飛行経路を得られたことから, 斜め飛行の導入による改良は適切であったと考えられる. これにより, 一方向を向いたままの飛行方向に制限がなくなり, 止まっているものに限らず動いているものを追跡する際にも適切な飛行が期待できる. また, 斜め方向を含め全方向への飛行が可能なので, 狭い空間や入り組んだ空間での飛行が容易になったといえる.

意図した速度での飛行に関していえば斜め飛行を導入したことによる誤操作は比較的少なく, 空間の広さに適した飛行が容易になったといえる. しかし, 誤操作によって速度が変化してしまい飛行経路が乱れてしまうこともあり, 狭い空間で誤操作した場合には素早い修正が必要となると考えられる. そのため, 速度変化をさせるには何かしらの誤操作しないアクションを起こし, それから速度の増減をする方法が望ましいと考えられる.

5. おわりに

本研究では, 吉田らのシステムを改良し, 離着陸, 上昇下降, 左右旋回, 前後左右, 左斜め前後, 右斜め前後へ速度変化のある飛行を実現し, 身体を使用した AR.Drone の操作システムを提案した. また, コントローラによる操作と, 吉田らのシステム, 改良したシステムの飛行経路を比較し, 行った改良が適切であったかを示した. また, 飛行速度を変化させる別の手法としては, あらかじめ決められた速度をジェスチャ等によって切り替えるほかに, 一方向へ飛行し続けた場合に加速度を増加させて速度変化を実装する方法も考えられる.

参考文献

- [1] 吉田成朗, 鳴海拓志, 橋本直, 谷川智洋, 稲見昌彦, 五十嵐健夫, 廣瀬通孝: ジェスチャ操作型飛行ロボットによる身体性の拡張, インタラクション2012
- [2] Parrot AR.Drone2.0, <http://ardrone2.parrot.com/> (2014年12月現在)
- [3] Kinect, <http://www.xbox.com/ja-JP/kinect> (2014年12月現在)