

論文

AR. Drone を用いた 自律型観測システムの最適制御

藤山 雷太 謙訪 敬祐

近年、産業用の無人航空機（UAV）が注目されているが、GPSを用いた自動制御にはGPS観測値の誤差による制御誤差が存在する。本研究では、GPS観測値のフィルタリングを行うことで観測値の精度改善を行い、UAVの自律制御システムを構築し、フィルタリングの影響の検証・評価を行った。その結果、観測値に対し最適化を行うことで、制御や飛行ルートに対してスムージングが行われることが確認できた。

キーワード：自律制御、カルマンフィルタ、AR. Drone、無人航空機、GPS

1 はじめに

1. 1 研究の背景

(1) 産業用 UAV の現状

近年、産業用の無人航空機（Unmanned Aerial Vehicle : UAV）が注目されている。UAVは日本でも産業用 UAV として農薬散布用などにも使用されており、日本産業用無人航空機協会（JUAV）によると 2300 機を超える数が利用されている^[1]。米国でも、近年までは小型の UAV を飛行させるには、高度 400 フィート（約 120m 以下）かつ人口密集地から離れた場所で、商業目的外でのみ自由な飛行が許されていた^[2]。商業目的の場合は事前に運用許可申請の必要があり、UAV の商業サービスへの活用の障害となっていた。しかし、2012 年 9 月に米国オバマ大統領が UAV 利用の規制緩和法案に署名したこと、2015 年までに連邦航空局（FAA）が、55 ポンド（約 25 キログラム）未満の小型 UAV の飛行解禁を進めている^[3]。米 Amazon.com では小型 UAV による配送サービス「Amazon Prime Air」を発表している。顧客が注文した商品を 30 分以内に小型 UAV で配達させるというサービスである^[4]。「Amazon Prime Air」は GPS を用いた自動制御を用いている。他の UAV 制御手段としては、画像追跡を用いた自動制御が提案されている。

GPS を用いた制御では、目標と自身の位置情報を緯度・経度によって取得し、目標へ誘導を行う。画像追跡を用いた制御では、予め登録した画像データを元に合致する物体を検出し、対象を追跡する。

「Amazon Prime Air」のような 2 点間を移動する小型 UAV の制御には GPS を用いた目標と自身の位置情報をリアルタイムに取得する手法が適切であり、画像追跡のような対象を認識するまで移動方向が不確定な制御手法は長距離を飛行するのに適さない。

(2) 従来の GPS 制御

GPS を用いた自動制御システムの例を図 1 に示す。従来の GPS 制御では、現在地と目的地の 2 点の座標を取得し、現在地に関しては、リアルタイムに取得することを前提としている。2 点間の移動は基本的に直線移動をし、位置情報が更新される度に、目標への移動方向を決定する。しかし、UAV という特性上、風に流されるなど想定と違う方向へ移動することがある。コースから外れた場合、GPS の値から移動方向を再設定するが、取得する GPS の値も正確ではなく誤差を含んでいる。そのため、すべての GPS 観測値をそのまま用いてしまうと、最大で観測誤差と移動誤差の 2 つの誤差の影響を受けることになり、移動ルート上にないビルなどの障害物に衝突する危険性がある。

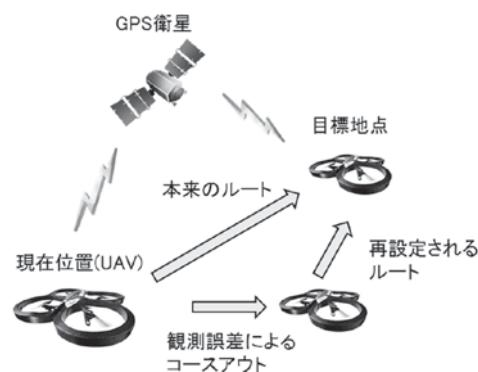


図 1 GPS を用いた自動制御

FUJIYAMA Raita

東京都市大学 環境情報学部 情報メディア学科 2013 年度卒業生

SUWA Keisuke

東京都市大学 メディア情報学部 情報システム学科 教授

1. 2 研究の目的

本研究では、精度の低い単独測位 GPS を使用した UAV の自律制御システムの最適制御手法について検討する。精度の低い単独測位 GPS を積載した小型 UAV でも安定した制御を実現できるシステムの構築と検証を行う。

2 研究内容

従来の研究では、取得した GPS 観測値をそのまま制御に用いる方法を取っているため、目標に到達してもオーバーランするといった問題が発生している。これは、移動体に誤差のある観測値が加わることで、タイムラグと誤差によって引き起こされる問題と言える。

本システムでは、実験対象の UAV として AR. Drone^[5] と呼ばれるクアッドコプターを使用し、Android OS のスマートフォンを GPS 受信端末として採用する。従来の研究に比べ、取得した観測値にカルマンフィルタを使用して制御を行う。研究の検証には観測値をそのまま制御に用いた場合とカルマンフィルタを適用した場合を比較して特性を明らかにする。

3 開発環境

本研究では、GPS 受信機のアプリケーション、UAV 制御のアプリケーションの開発をそれぞれ行う。開発環境及び言語を表 1 に示す。

GPS 受信機にはサムスン電子のスマートフォン Galaxy S を用いるため、アプリケーションは Android OS2.3 向けに開発した。使用言語は Java、開発環境は Eclipse4.3 Kepler となっている。UAV 制御端末には Windows のノート PC を用いるため、アプリケーションは Windows OS (Windows7) 向けに開発した。使用言語は C/C++、開発環境は Visual Studio C++ 2008 Express である。

AR. Drone には Parrot 社が提供している SDK として AR. Drone SDK^[6] があり、PC 向け、スマートフォン向けのプログラムを作ることができるが SDK のソースコードは難解かつバグが多いため、修正と理解に多大な時間を要する。そこで、フリーソフトウェアの AR. Drone 制御プログラム Cv Drone^[7] を利用することで、バグ修正の手間の軽減や、機能拡張のしやすさを向上させる。

Cv Drone は Open CV と呼ばれる画像処理プログラムを活用するために作成されたものであるが、手動で AR. Drone を制御する目的でも使用することが可能である。Parrot 社の公開している AR. Drone SDK と比べても、制御通信、動画像通信、センサ情報の取得など必要な機能が全て揃っているため、同じように使用することができる。ヘッダファイルが 1 つに纏められているという特長がある。なお、使用言語は C++ である。

表 1 開発環境

	開発環境	開発言語
Android アプリケーション	Eclipse4.3kepler	Java
PC アプリケーション	VisualStudioC++2008Express	C++

4 自律制御システム

4. 1 システム構成

本研究では、AR. Drone をスマートフォンと PC を用いて制御する。システム構成を図 2 に示す。AR. Drone は PC による Wi-Fi 通信により制御する。AR. Drone の制御には GPS 座標を用い、取得にはスマートフォンの GPS 情報取得機能を用いる。スマートフォンは AR. Drone に搭載され、取得した GPS 情報は PC に Wi-Fi 通信により送信する。PC は受信した GPS 情報をカルマンフィルタと呼ばれるフィルタリング手法により最適化し、制御に用いる。全体の制御フローチャートを図 3 に示す。

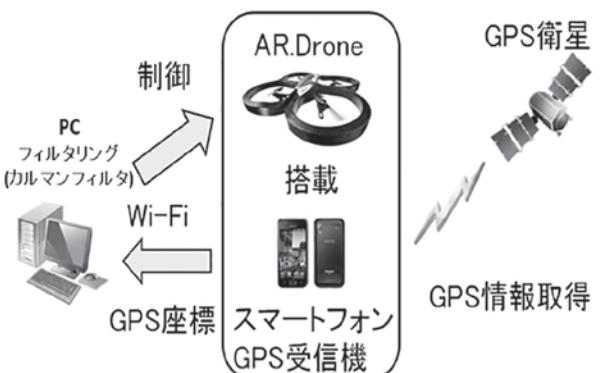


図 2 システム構成

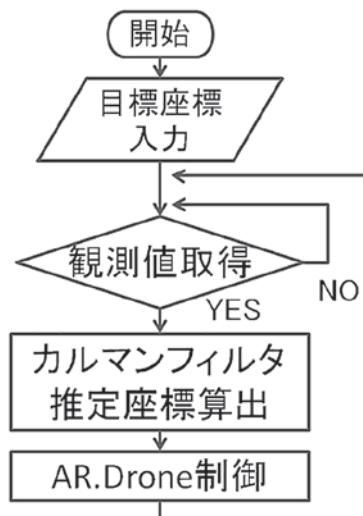


図 3 AR. Drone 制御フロー

4. 2 GPS 情報送信アプリケーション

スマートフォンで使用するアプリケーションとして、GPS 情報送信アプリケーションを作成した。アプリケーションの起動画面を図 4 に示す。動作のフローチャートを図 5 に示す。IP アドレスとポート番号は起動後に再設定でき、Send ボタンを押すことで IP アドレスとポート番号に入力された値をもとに GPS 情報の送信を開始する。送信中は Send ボタンが Stop ボタンに変わり、押すことで送信を停止することができる。GPS 情報の取得には、Android API に含まれる Android. Location [8] を使用し、GPS 情報は変動する度に自動で更新される。GPS 情報の送信には TCP のソケット通信を用いており、2 秒ごとに 1 つの文字列として纏め、GPS 情報を送信する。また、方位を取得する機能も追加したが、AR. Drone に搭載するという環境でスマートフォンを使用するため、端末の角度が不安定になり測定値に誤差が発生するため、本研究では、送信はしても制御アプリケーション側では使用しないこととした。



図 4 GPS 情報送信アプリケーション

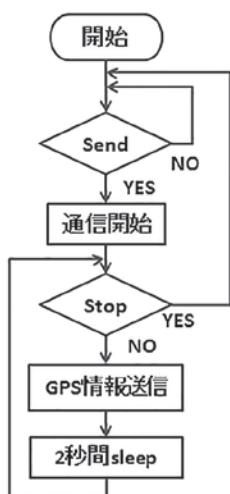


図 5 GPS 情報送信フロー

4. 3 PC アプリケーション

(1) GPS 情報受信アプリケーション

スマートフォンとのソケット通信を行うため Win OS 上で通信を行うのに Winsock1.1 [9] を使用し、TCP による受信を行う。無線通信は GPS 情報の受信だけでなく AR. Drone の制御にも使用するため、マルチスレッドによる通信処理を用いた。GPS 情報を受信された文字列データは緯度・経度・方位に分割し、数値に変換する。

(2) カルマンフィルタ

カルマンフィルタとは時系列解析を行うフィルタリング手法で、時間と共に変化する値であれば、航空・宇宙工学などさまざまな分野で利用されている [10]。時系列な観測値を用いて、ある時刻の状態に関する最適な推定値を求めるアルゴリズムである [11]。カルマンフィルタを用いることで単独測位の観測値のスムージングが期待できる [12]。

図 6 に 1 次元データ用のカルマンフィルタを簡略化した式を示す。カルマンフィルタは観測値から推定値を引いたものにカルマンゲインという係数をかけ、推定値に加算するものである。図 6 の④番の式が推定をする式で、③番がカルマンゲインと呼ばれる係数を決定するものである。カルマンゲインはプロセス誤差により決定される誤差共分散と観測誤差により 0.5 に収束するよう変動する。決定した推定値と誤差共分散は次回の推定ステップに持ち越される。

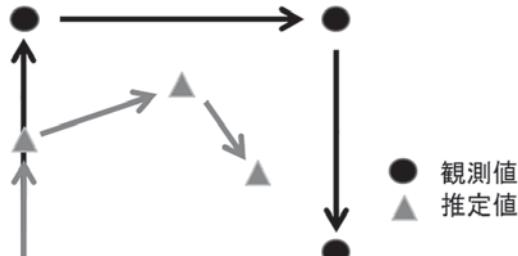
本研究では、推定値の初期値 $x[0]$ は最初に取得した観測値とし、誤差共分散の初期値である $p[0]$ は 0 とする。

図 7 は鋭角な観測値に対するカルマンフィルタの特性を示したものである。カルマンフィルタを図 7 の観測値のような鋭角な軌跡に適用した場合、緑点のように鈍角な軌跡を描くと考える。カルマンフィルタを制御に用いた場合、UAV が図 7 の緑点の軌道に沿った移動をすることで外乱によって引き起こされる移動量を減らすことができるのではないかと考える。

- | | |
|----------------------------|---------|
| ① $x = x[t-1]$ | 事前推定取得 |
| ② $p = p[t-1] + v$ | 誤差共分散 |
| ③ $k = p / (p + w)$ | カルマンゲイン |
| ④ $x[t] = x + k * (y - x)$ | 推定値決定 |
| ⑤ $p[t] = (1 - k) * p$ | 誤差共分散更新 |

x : 推定値 y : 観測値 k : カルマンゲイン p : 誤差共分散 t : ステップ数
および p は $[t-1]:1$ ステップ前の状態と $[t]$ 現在の結果を返す

図 6 カルマンフィルタ式



$$\text{rad} = \text{atan2}(y, x)$$

$$\text{角度} = \text{rad} * 180 / \pi$$

atan2:アークタンジェント2
y:縦座標 **x:**横座標 **rad:**ラジアン **π:**円周率

図 8 方位角決定式

AR. Drone を自律制御するのに必要な移動方向を決定にするシステムを開発した。図 8 の計算式を用いて、地球を完全な球体であると仮定し、2 点間の緯度経度の差からラジアンを算出する。この値は移動すべき方位に変換され、制御のピッチ・ロール角に反映される。本システムでは、AR. Drone が現在向いている方向を、スマートフォンから送られてくる方位センサの情報を元に取得し、移動方向に反映させる仕組みも開発したが、本研究では北に固定した。ラジアンの算出には、アークタンジェント 2 と呼ばれる位置座標を入力することで、 $-\pi$ から π までの範囲で値を返す。ラジアンは $180/\pi$ をかけることで角度に変換できるが、角度が右に 90 度ズレのある値を返すので修正してから制御入力値に変換している。

5 結果・考察

5.1 検証・評価方法

検証方法は、学内の敷地内（屋外）にてスマートフォンを搭載した AR. Drone で自動制御を行い、カルマンフィルタ有りの場合と無しの場合の 2 パターンで目標への誘導を行う。スマートフォンの方位センサによる制御は使用せず、スマートフォンにて北を確認し、AR. Drone を北に向けて制御する。2 パターンとも目標地点である緯度 35.5618850 度、経度 139.5745883 度に向けて離れた同じ位置から制御を行う。制御は 2 秒ごとに観測値が更新されるたびに目標へ直線的に向かうものとする。図 9 は検証用の飛行経路である。

図 10 はスマートフォンを搭載した AR. Drone である。カルマンフィルタの影響を確認するため、スマート



図 10 スマートフォン搭載 AR. Drone

フォン単体を手に持ち、意図的に外乱をかけた移動を行い、観測値の変動に対してカルマンフィルタがどのように変化するか検証する。

5.2 結果

(1) 飛行軌跡

AR. Drone の飛行経路に対して、カルマンフィルタがどのような影響を示すのかを確認する。AR. Drone の制御システムを屋外で使用し、カルマンフィルタの有無で運用した。観測値からカルマンフィルタを制御に用いない場合と用いた場合の結果を、図 11、図 12 に示す。方位は図 11、図 12 の上方が北で、右方向が東である。サンプリング間隔は 2 秒間隔で、距離は 0.00001 度で約 1m である。離陸地点と目標地点はカルマンフィルタ有りの場合と無い場合とも同じ位置で、離陸地点から目標地点までの距離は約 10m である。図 11、図 12 では離陸地点は図の左下で、目標地点が図の右上の三角形の印の位置である。検証は同日に連続して行った。当日は東に弱い風が吹いていた。

カルマンフィルタ無しの場合（図 11）では離陸直後に 5m ほど大きく北に移動した後、東に軌道修正し、ジ

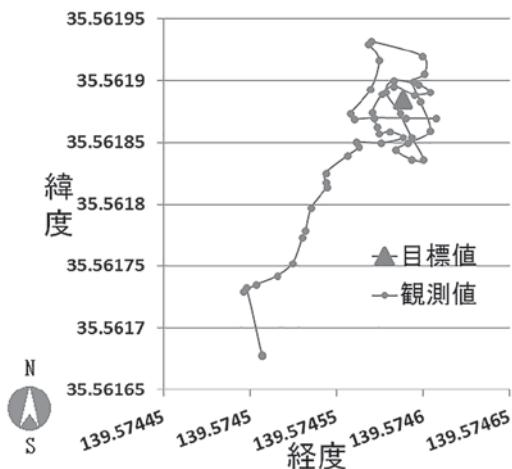


図 11 カルマンフィルタ無しの飛行結果

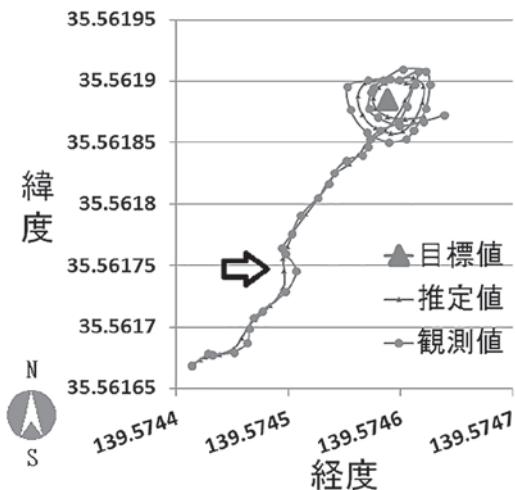


図 12 カルマンフィルタ有りの飛行結果

グザグに移動しながら目標へ移動した。移動中の観測値は、移動間隔（移動速度）が不規則に変動している。目標付近である約3m付近に近づいた後は、約3m東に移動し、その後西に軌道修正しつつ目標に移動した。目標付近に近づいた後は、急激な方向転換と移動を繰り返しながら離れたり近づいたりを繰り返した。

カルマンフィルタ有りの場合（図12）では離陸直後から少しずつ東にそれていたが、矢印付近で西に軌道修正した。目標付近までは、ほぼ一直線に移動を続け、移動間隔（移動速度）はほぼ等間隔に移動している。目標付近である3m付近に近づいたときは、目標の右側を通過した後、目標を中心に周囲を円形に周回しながら目標に収束するように飛行した。

目標付近での飛行経路について詳細に比較してみた。図13、図14は図11、図12の目標地点周辺を拡大したものである。

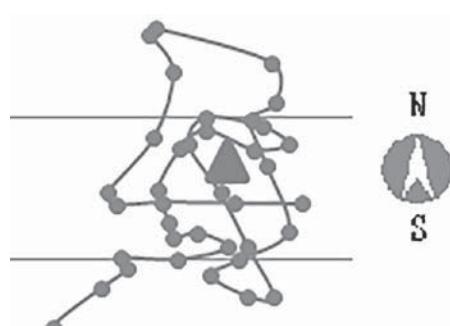


図 13 カルマンフィルタ無しの目標付近

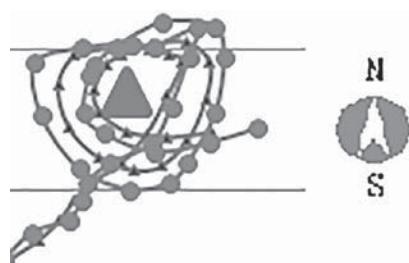


図 14 カルマンフィルタ有りの目標付近

目標付近では、カルマンフィルタ無し（図13）の場合は、目標に接近することは何度かあるものの、移動距離や移動方向が不規則で目標に収束する気配がない。移動の軌跡は、目標に近づくように移動しているが、移動間隔が大きく通り過ぎてしまい観測値が目標地点に収束しない。

これに対してカルマンフィルタ有り（図14）の場合では、目標に対して周回軌道を描くように収束しながら移動した。移動間隔はほぼ等間隔であった。

(2) 観測値の分布

観測値の分布について、カルマンフィルタがどのような影響を示すのかを確認する。目標地点から緯度経度それぞれに対して0.00003度（約3m）以内に侵入した後の観測値の95%がどの程度の範囲で分布しているのかを図15に示す。それぞれの観測値の平均を中心経度緯度の範囲を表したものである。破線の範囲がカルマンフィルタ無しの場合で、実線の範囲がカルマンフィルタ有りの場合である。三角形のマークは目標値である。

カルマンフィルタ無しの場合は、観測値の分布範囲が緯度の範囲は長く、経度の範囲が狭い長方形のような範囲となっている。目標値が経度の範囲の端に存在している。カルマンフィルタ有りの場合は、観測値の分布範囲が共に約1.5mの正方形に収まっており、目標値が分布の左上に存在している。

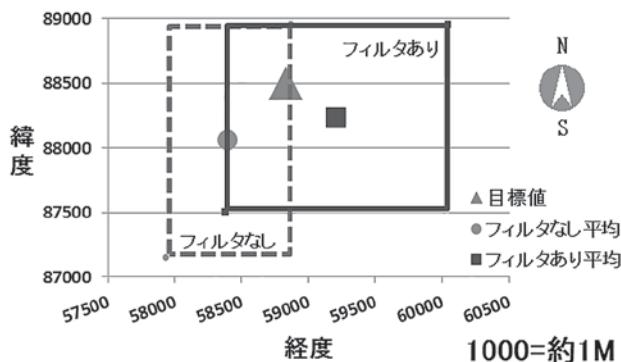


図 15 観測値の分布範囲

5. 3 考察

実際の飛行検証結果から、カルマンフィルタを適用することによりカルマンフィルタ無しの制御に比べ、目標への移動で観測値が鋭角になっていてもスムーズに移動できていることがわかる。目標地点周辺では、カルマンフィルタの有無に関わらず約4～5mの範囲で移動してしまっていることから、精度自体に影響は出でていない。カルマンフィルタを制御に用いた場合では、時間と共に目標地点との距離が縮まっているのでカルマンフィルタを用いることで時間をかければ精度が上がるようになると見える。飛行経路の違いについては、カルマンフィルタ無しの制御では、目標への移動が往復を繰り返して軌道が不安定であるのに対し、カルマンフィルタ有りの制御では、目標の周囲を反時計回りに旋回するという安定した軌道を取るため、カルマンフィルタによる移動経路のスマージングが行えていると言える。

6 おわりに

6. 1 まとめ

本システムの検証結果から、GPS情報の誤差最適化を行う自律制御システムが正常に動作していることが確認できた。カルマンフィルタを使用することで外乱のかかった観測値のスマージングが確認できた。最適化したGPS情報を制御に用いた場合、移動経路のスマージングが行われ、目標地点周辺では円形に周回軌道をとることが確認できた。

以上より、本研究のシステムでは、AR. Drone の軌跡に対してスマージングが行われ、安定した軌道を辿ることが分かった。

6. 2 今後の課題

本研究では、方位、移動速度といったデータを活用できていない点、積載量を超過している点、目標への到達精度が大きく向上していないなどの課題があげられる。

活用できていないデータに関しては、UAV上でも正

確な方位を取得し、加速度から移動速度を算出できるようにする必要がある。積載量の超過に関しては、積載量の多いUAVを使用することで、飛行時の安定性を獲得できる。目標への到達精度の向上に関しては、目標地点に近づいた場合に移動速度を抑える仕様の追加や方位や移動速度データの活用に加え、既存研究^[12]の速度推定や測位衛星の電波強度に閾値を設けることにより単独測位GPS制御による到達精度の向上が実現できると考える。

謝辞

本研究のシステム構築の際に御協力と助言をいただいた諫訪研究室OBの小野澤 清人 氏に感謝の意を表します。

参考文献

- [1] UVS International 2009/2010 UAS Yearbook - UAS: The Global Perspective - 7th Edition - June 2009
- [2] FAA-UAV Q&A
http://www.faa.gov/about/initiatives/uas/uas_faq/index.cfm?print=go
- [3] ウォールストリートジャーナル日本語版
http://jp.wsj.com/public/page/0_0_WJPP_7000-397465.html?mg=inert-wsj
- [4] Amazon Prime Air
http://www.amazon.com/b?ref_=tsm_l_tw_s_amzn_mx3eqp&node=8037720011
- [5] AR. Drone
<http://ardrone2.parrot.com/>
- [6] AR. Drone Open API Platform
<https://projects.ardrone.org/wiki/1/Japanese-top>
- [7] CV Drone
<http://pukulab.blog.fc2.com/blog-entry-11.html>
- [8] Android. Location
<http://developer.android.com/reference/android/location/package-summary.html>
- [9] Winsock
<http://www.c-worker.ch/dokuwsck/owinsck/winsock.htm>
- [10] 足立修一, 丸田一郎, カルマンフィルタの基礎, 東京電機大学出版局, 2012.10
- [11] 中野道雄, 西山清, パソコンで解くカルマンフィルタ, 丸善株式会社, 1993.2
- [12] 孫世偉, カルマンフィルタを用いたGPS単独測位精度向上に関する研究, 修士論文, 東京海洋大学, 2007