

修 士 論 文 の 和 文 要 旨

研究科・専攻	大学院情報システム学研究科 情報ネットワークシステム学専攻 博士前期課程		
氏 名	小野澤 清人	学籍番号	1252007
論 文 題 目	自律ナビゲーションの実装による 「浮遊ノート」システムの機能拡張とその応用		
要 旨	<p>ロボットの普及率は生活ロボットが大きく上昇していくとの見解があり、今後ロボットの利用が浸透するにつれ、人間と行動を共にするようになることが見込まれる。芝はロボットが各人のパートナーとして所有されることを目指し、「浮遊ノート」システムを提案している。このシステムは、飛行ロボットはユーザの近くを浮遊することでその人をサポートする存在であるが、ユーザの元を離れて動作するようになれば、誰もが任意の地点まで飛行ロボットをお使いに出すことができるようになる。お使い機能を達成する基礎技術として、誰でも手軽に利用できる遠隔地移動ロボットは重要である。それが完成すれば、ロボットに買い出しに行かせることやユーザの道案内をすることができる。お供ロボットをユーザが望む地点まで移動させるためには、「簡単に目的地設定を行えること」や、「人間が通行可能な経路を通してロボットが目的地まで向かう」といった課題を解決しなければならない。さらに、ユーザを目的地まで道案内するためには、ユーザの歩行に合わせてお供ロボットが移動する必要がある。</p> <p>本論文では、上記の課題を解決するための提案を行う。携帯情報端末、PC、小型飛行ロボットを用いて</p> <ul style="list-style-type: none"> ・ 誰もがどこにいても簡単に飛行指示を与えることができるユーザインタフェース ・ 飛行ロボットに GPS センサ・電子コンパスを搭載し、一般的なネットワーク地図サービスによって作成できる経路を飛行できるシステム ・ ロボットがユーザを先行しながら一緒に移動できるアプリケーション <p>を構築し評価検証を行った。</p> <p>それぞれの検証より、提案した課題解決を行えているかどうかを述べ、問題点や改善点がある場合は解決方法を提示する。結論として、</p> <ul style="list-style-type: none"> ・ どの年代の人も地図アプリケーションを利用することで目的地設定を簡単に行えること ・ GPS センサ・電子コンパスを搭載した飛行ロボットにより、作成した経路を飛行できること ・ 加速度センサにより歩行検知を行うことで、ユーザの歩行と連動させることができるようになった。 		

平成25年度修士論文

自律ナビゲーションの実装による
「浮遊ノート」システムの機能拡張とその応用

大学院情報システム学研究科
情報ネットワークシステム学専攻

学籍番号： 1252007

氏名： 小野澤 清人

主任指導教員： 入江 英嗣 准教授

指導教員： 吉永 努 教授

指導教員： 笠井 裕之 准教授

提出年月日： 平成26年1月27日

(表紙裏)

目次

第1章	序論	1
第2章	関連研究	3
2.1	自律動作に関する関連技術	3
2.1.1	マーカを用いた自律動作	3
2.1.2	GPSや電子コンパス, その他の外部センサを用いた自律動作	4
2.2	歩行検知	4
第3章	「浮遊ノート」システムのための提案	6
3.1	お使い機能の提案	6
3.2	お使い機能を実装する上での課題	6
3.3	簡単な目的地設定	7
3.4	作成された経路を移動するための自律ナビゲーションシステム	9
3.4.1	自律ナビゲーションシステム実現のためのアイデア	9
3.4.2	風の影響をキャンセルするための補正提案	9
3.4.3	GPSの誤取得を防ぐための補正提案	10
3.5	歩行検知システム	12
3.6	提案したシステム同士の連携	12
第4章	実装	14
4.1	本システムの概要	14
4.2	地図アプリケーションの実装	16
4.2.1	タッチ指示による目的地設定	16
4.2.2	手動での経路設計	16
4.3	自律ナビゲーションシステムの実装	18
4.4	歩行検知システムの実装	21
4.5	システムの処理の流れ	21
第5章	システムの評価	23
5.1	地図アプリケーションの評価	23
5.2	自律ナビゲーションシステムの評価	26
5.2.1	補正アルゴリズム導入前の飛行実験の評価	26
5.2.2	風による影響やGPSの誤取得補正システム導入後の評価	30
5.3	歩行検知システムの評価	32
5.3.1	検知システムの精度	32
5.3.2	検知から飛行ロボット制御までにかかる時間	36

第 6 章 結論	38
6.1 本研究の成果	38
6.2 今後の課題	39
謝辞	41
参考文献	43

目次

3.3.1 実現したいアイデア	8
3.4.1 風の揺らぎによる補正指示	11
3.4.2 移動速度による自己位置推定	11
3.6.1 歩行検知による飛行ロボットの動き	13
3.6.2 収束場に送信されるデータ	13
4.1.1 システム構成図	15
4.1.2 スマートフォンを装着した AR.Drone	15
4.2.1 目的地設定の手順	17
4.2.2 飛行ロボットの通過不可能な経路と可能な経路	17
4.3.1 制御信号決定までのフロー	19
4.3.2 前進・旋回の指示が出される条件	20
4.3.3 目的地までの移動イメージ	20
4.5.1 歩行と止まる時の検知の様子	22
4.5.2 システム全体のフロー	22
5.1.1 設定しやすいかどうか	25
5.1.2 設定方法がわかりやすいかどうか	25
5.1.3 GPS 情報を意識しないかどうか	25
5.1.4 総合的な難易度	25
5.2.1 飛行させた道路のイメージ図	28
5.2.2 緯度・経度と経路・目的地までの残りの距離	29
5.2.3 コンパスデータの比較と制御指示	29
5.2.4 実測値と推定地の軌跡	31
5.3.1 被験者 A のグラフ	33
5.3.2 被験者 B のグラフ	33
5.3.3 被験者 C のグラフ	34
5.3.4 被験者 D のグラフ	34
5.3.5 被験者 E のグラフ	35
5.3.6 被験者 F のグラフ	35
5.3.7 歩き始めと終わりの検知からロボット制御までの時間	37

表目次

4.1	iPhone4 のスペック	14
4.2	PC のスペック	14
4.3	サーバアプリケーションのスペック	14
5.1	離陸, 着陸, 経由を行う各地点の GPS の感度	28
5.2	目的地までの距離や方角の許容範囲	28
5.3	自律飛行の実験結果	30
5.4	風の影響や GPS の誤取得補正システム導入後の実験結果	30

第1章 序論

ロボットは医療や福祉で活躍するようになり人間が身近に感じることができるようになった [1]. 一般家庭内での人間の生活をサポートするロボットとして、ルンバ [2] のような自動掃除ロボットが普及している。また、専門家が人々をサポートする際に利用するロボットとして、Amazon Air Prime のようにナビゲーション機能による自動配送などを提唱するようになった [3]. さらに、ロボットの普及率は生活ロボットが大きく上昇していくとの見解がある [4][5]. これらより今後ロボットの利用が浸透するにつれ、人間のパートナーとして存在するようになり、人間と行動を共にするようになることが見込まれる。芝は「浮遊ノート」システムを提案し、ロボットが各人のパートナーとして所有されることを目指している [6]. 飛行ロボットが携帯情報端末を有した状態で所有者の傍を飛行することで、両手が塞がった状態においても携帯情報端末を利用することができるようになる。

このようにロボットが人間の補助なしに独立して何らかの作業を行うことは、人間の生活をサポートする上で重要なメリットの1つである。「浮遊ノート」システムでは、飛行ロボットはユーザの近くを浮遊することでその人をサポートする存在であるが、ユーザの元を離れて動作するようになれば、誰もが任意の地点まで飛行ロボットをお使いに出すことができるようになる。飛行ロボットのお使い機能を達成する基礎技術として、誰でも手軽に利用できる遠隔地移動ロボットは重要である。それが完成すれば、ロボットに買い出しに行かせることができるようになり、ユーザの望む地点まで行き、写真撮影を行うことでユーザの視野を拡大するといった動作へと応用できる。また、ロボットが遠隔地まで移動するにあたり、人間が移動する経路と同じ道を通ることができれば、ユーザの道案内を行うことが可能となる。

お供ロボットをユーザが望む地点まで移動させるためには、

- 簡単に目的地設定を行える
- 人間が通行可能な経路を通してロボットが目的地まで向かう

といった課題を解決しなければならない。さらに、ユーザを目的地まで道案内するためには、ユーザの歩行に合わせてお供ロボットが移動する必要がある。

そこで本論文では、上記の課題を解決するための提案を行う。地図上の任意の点を指で指し示すだけで目的地を設定できれば、ユーザは簡単に目的地を設定でき、地図サービスを利用すれば人間が通行可能な経路を作成することができる。さらに GPS センサ・電子コンパスを搭載したロボットを利用することで、簡単なアルゴリズムでその経路を移動できるようになる。また、ユーザの歩行状態を加速度センサにより検知し、そのタイミングでお供ロボットの移動を切り替えれば、ユーザの歩行に合わせた移動を行わせることができる。これらの提案手法により、ユーザは携帯情報端末により地図サービスを利用しなくても、お供ロボットがユーザの歩行に合わせて目的地まで案内できるようになる。

本論文では,

- 誰もがどこにいても簡単に飛行ロボットに指示を与えることができるユーザインタフェース
- 飛行ロボットに GPS センサ・電子コンパスを搭載し, 一般的なネットワーク地図サービスによって作成できる経路を飛行できるシステム
- ロボットがユーザを先行しながら一緒に移動できるアプリケーション

を構築した. 結果, GPS 取得精度が安定しない地点においても補正アルゴリズムにより, ナビゲーション可能なシステムを構築できた. このシステムに障害物を回避する技術を組み込み, バッテリーが持続するようになれば, どこへでも移動できるシステムとなる. 道案内機能に関して, ユーザの歩行アクションを検知しユーザと一緒に移動しながら目的地へ向かうことができるようになった.

本論文は以下のように構成される. 2 章で関連研究を紹介し, 3 章でお使い機能のための提案を行う. 4 章では実装について述べ, 5 章でシステムの評価を行う. 6 章で結論を述べる.

第2章 関連研究

2.1 自律動作に関する関連技術

2.1.1 マーカを用いた自律動作

芝は、ロボットが人間のパートナーとなり、ユーザを様々な場面から手助けする将来を目指し、「浮遊ノート」システムを提案している [6]. このシステムは、携帯情報端末を把持した飛行ロボットがユーザの要望に沿った位置に滞空することで、把持や画面注視の必要なく容易に情報端末を利用できることを目指したお供ロボットである。現在、「浮遊ノート」システムには

- ユーザ追跡モード
- 端末情報の浮遊表示
- 手書きメモを書き込む浮遊入力や自撮り支援システム
- ユーザのメモを特定の他人に伝える移動伝言板モード

が備わっている。このシステムのうち、自撮り支援システム、移動伝言板モードの実装にあたり、AR マーカを利用し飛行ロボットのユーザの識別を行っている。自撮り支援システムは、ユーザが自身を写真に収める際、飛行ロボットがユーザの顔が写真の中心になるように撮影するシステムである。この機能の利用にあたり、ユーザは AR マーカを体の前後に装着する。飛行ロボットには AR.Drone が利用されており、この端末に搭載されたフロントカメラを利用する事で AR ユーザの識別を行う。ユーザの背面から機体が離陸し、背面マーカを見つけた後ユーザの前面まで回り込む。その後機体は半回転し、前面マーカを見つけた後上昇を始め OpenCV により顔を識別し、ユーザの写真撮影を行う。浮遊伝言板モードでは、離れたユーザ同士で手書きメモを渡し合う際、飛行ロボットがユーザの元までそのメモを届けるシステムである。所有者及び伝言のターゲットとなるユーザの体の前面にそれぞれ AR マーカを装着する。所有者とターゲットは向かい合った状態で離れた場所で待機する。機体は所有者と向き合った状態で離陸し、所有者の AR マーカの位置まで上昇する。所有者が機体の把持する情報端末上に手書きメモを記すと、機体は半回転・前進することでターゲットの方へ向かう。ターゲットの AR マーカを見つけると機体はその場で浮遊し、その間にターゲットは所有者からの伝言を読み取り、所有者への伝言を記す。その後機体は再び半回転・前進する事で所有者の方へ向かい、所有者の AR マーカを見つけた後、その場で浮遊する。これにより所有者はターゲットからの伝言を読むことができる。

Wenhui らは、IR マーカを利用した自律した離着陸を行っている [7]. 飛行ロボットの底に赤外線 (IR) カメラを搭載し、IR マーカ上に機体を設置する。ロボットは離着陸時、IR マーカのパターン認識を行い、認識したパターンの中心点及び視野の角度から機体の傾きを計算し、機体を地面と水平に保つように自立的に調整する。これにより、飛行ロボットの安定した離着陸の自律動作を可能としている。

2.1.2 GPS や電子コンパス, その他の外部センサを用いた自律動作

Jean-Christophe らは, 飛行ロボットが静的な障害を避けながら自律飛行することを可能としている [8]. GPS センサ及び7つのオプティカルフローセンサを飛行ロボットに搭載することで, あらかじめ指定された GPS の経路を飛行する. この時, オプティカルフローにより静的な障害を検出した場合, 7つのセンサのうち障害が検出されていない方向へと移動しながら経路を飛行することで, 障害を避けながら飛行できる.

Tianmiao らは, 飛行ロボットの速度制御を行いながら目的地までの自律飛行を実現している [9]. GPS センサと電子コンパス及び加速度センサを飛行ロボットに搭載し, それらを合わせて利用する. これにより目的地までの距離と機体の向きを計算しながら, 機体が傾き過ぎて速度が早くなりすぎないように制御しつつ目的地まで到達することを可能としている.

Jasmin らは, GPS とネットワークを用いたロボットのナビゲーションシステムを開発している [10]. ユーザは PC 上のソフトウェアで目的地の緯度経度の数値を設定し, ネットワーク経由で GPS センサを搭載した車輪移動ロボットを移動させる. ロボットは4秒前と現在の GPS データから目的地までの GPS データよりそれぞれの距離を算出・比較を行い, 目的地に近づいていればそのまま走行し, 遠のいている場合は左に60度回転し, 再び移動する. この一連の動作を繰り返すことで, 目的地に到着する.

Cooper らは, 飛行ロボットの前方に搭載されたカメラを利用することで室内での自律的移動を可能としている [11]. 廊下を飛行する際, ロボットのカメラより取得した画像を利用する. その画像にキャニー法を用いることでエッジ検出の処理を行い, その処理した画像に対し確率的ハフ変換を行う. これにより, 廊下の端部へと続く多数のラインを取得する. それらのラインは, 廊下のパニングポイントにむかって収束するため, そのポイントを目指すように飛行させることで, 機体は自律的に移動する.

MIT Senseable City Lab では, GPS センサを利用することで飛行ロボットに大学内を案内させることを提試みている [12]. ユーザは自身の持つスマートフォンより飛行ロボットを呼び出すことで道案内を行わせることができる. このシステムでは道案内を行っている時, ユーザと飛行ロボットが一定距離離れてしまうと飛行ロボットが道案内を一時停止する. これはユーザが持つスマートフォンの GPS 及び飛行ロボットに搭載された GPS センサの値を利用し, 2地点間の距離を計算することで実装されている.

2.2 歩行検知

樫原らは, スマートフォンの3軸加速度センサーを利用することで, ユーザの悪い歩行の検知を可能としている [13]. ユーザの腰にスマートフォンを装着して歩行することで, 加速度センサから歩行の波形を取得することができる. 3軸のうち y 軸の加速度センサーの値を利用することで, 取得した波形を加工する. この時, 負のパルスが閾値内に入っていれば, 悪い歩行を検知したと見なし, スマートフォンから警告音が鳴る.

北陸電気株式会社は社内開発された3軸加速度センサを用いた歩数を認識するアルゴリズムを公開している [14]. 人間がこの会社の加速度センサを装着し歩行すると, 取得したセンサの合成値の波形は, 一定の重力閾値より低くなった後, その閾値よりも高くなることを繰り返す. この値の大きさが適切である場合は歩行とみなし, このサイクルをカウントすることで歩数の計測を行っている.

Michael は加速度センサによる歩行検知によりナビゲーション，セキュリティのアプリケーションへ利用できると提唱している [15]。ナビゲーションは通常情報端末内蔵の GPS センサを利用することができるが，屋内施設などの GPS の感度が悪い環境では正しく動作しない。そこで，GPS が利用可能な間に，同じく情報端末に内蔵された加速度センサにより歩行のステップを検出することで，ユーザの歩行ステップあたりの移動距離を推定する。GPS が利用不可能となった際に，電子コンパスおよび加速度センサによるステップ検知を行うことで，GPS の代わりに位置推定を行うことができ，ナビゲーションを継続できる。情報端末が盗まれた際，機密情報を流出しないようにするため，セキュリティを向上させる必要がある。セキュリティ向上のため人間の歩行に着目する。ユーザの歩行や歩行周期を検出しそのデータをあらかじめ端末に記憶しておく。その後，機密情報を開く際歩行することでユーザをマッチングする。現在この認識技術によるユーザの誤検知は 10 % となっており，今後の研究の余地があると述べている。

第3章 「浮遊ノート」システムのための提案

3.1 お使い機能の提案

「浮遊ノート」システムは、お供ロボットとして存在することで、ユーザのパートナーとなりその人の生活を支えることを目的としている。そのため、「浮遊ノート」システムはユーザの後を追うことが基本動作であった。しかし、「浮遊ノート」がユーザの元を離れ、自律して遠隔地まで移動することができれば、ロボットを買い出しに行かせることや、ユーザの道案内、ユーザの望む地点の写真撮影といったお使いをさせることが可能となる。そこで、ユーザによる簡単な指示によりお供ロボットがユーザの元を離れて、自律して遠隔地まで移動できる「お使い機能」を提案する。また、ユーザの道案内のようにユーザを先導する場合、ロボットはユーザと離れ過ぎてはいけない。そこで、ユーザの歩行アクションに合わせてロボットと一緒に移動できる機能を提案する。

3.2 お使い機能を実装する上での課題

関連研究より、マーカを利用することで機体の離着陸や人物特定を行うことができる。しかしマーカを利用する場合、マーカを適切な位置に設置しなければシステムを正常に動作させることができなくなる。また、暗い場所ではマーカを認識することができなくなってしまうため、利用できる状況は限定されてしまう。GPS情報をロボットに搭載することで、経路通りの移動や目的地への到達を行うことができるが、Jasminらの実装例では設定には緯度経度の数値入力が必要となる。緯度経度の数値はユーザにとって手軽に扱えるものではない。例えば、ユーザを目的地まで案内するときや、ユーザから離れた地点へ様子を見に行くと言った要求に即座に応えることができない。また、機体に搭載されたカメラを利用することで、廊下のような真っすぐな道を飛行させることはできる。しかし、提案するお使い機能は屋外での利用を想定しており、屋外では廊下のようにバニシングポイントを見つけることは難しい。MIT Senseable City Labのように、人間とロボットの距離が離れた場合にロボットが一時停止するシステムにより人間の歩行に合わせたナビゲーションを行うことができるが、人間が立ち止まってからのレスポンスに大幅な遅延が発生してしまう。

人々の生活を支えるためには、ロボットへの指示は誰もが行えるよう簡単にする必要がある。そのため、ユーザがロボットの目的地を設定及びロボットへの飛行指示も簡単なユーザインタフェースとなるように設計する。また目的地を設定した際、人間の歩行可能なルートも設計する。ロボットがそのルートを通って無事目的地まで到達できれば、ユーザを目的地まで道案内することができる。また道案内機能において、ユーザの歩行アクションを検知できれば、歩行中はお供ロボットのナビゲーションを稼働、歩くのをやめているときはナビゲーションを一時停止させることが可能となる。

以上より、「浮遊ノート」システムのお使い機能実現の課題として、

1. 目的地設定を簡単に行えるユーザインタフェース (UI) の設計
2. ユーザが歩行可能なルート設計
3. 指定地点へのナビゲーションアルゴリズム
4. ロボットの自律的な危険回避
5. ユーザの歩行アクションの検知

が挙げられる。本論文では、4 以外の解決手法を提案する。以下、これらの課題を解決するための手法を記す。

3.3 簡単な目的地設定

目的地の設定を行うにあたり、視覚的に行えるような UI であれば数値入力により設定するよりも遥かに簡単である。そこで、場所・道路が視覚的かつ適切に表現されている地図に着目した。

図 3.3.1 のように地図を開いてユーザが求める任意の地点を指差すだけで目的地を設定でき、その地点までロボットを移動させることができれば、簡単な UI となる。上記の UI を実現するにあたり、必要なことは、

- 地図に目的地として印をつけることができること
- ユーザが示した地点の情報を家庭用ロボットに送信できること
- 目的地までのルート設計

である。これらを実現するためには、タッチパネルを有した地図及びインターネットに接続可能な端末が必要である。

さらに、ルート設計を自動で行うため、ネットワークサービスである Google Map を利用する [16]。Google Map は、地点情報と位置情報が対応付けされているため、GoogleMap 上でユーザが任意の地点を指定すれば、その地点の位置情報を取得することができる。また、任意の 2 地点を指定することで、その 2 地点間の経路設計を行うこともできる。ここで設計される経路は、全てユーザが歩行できる経路となる。

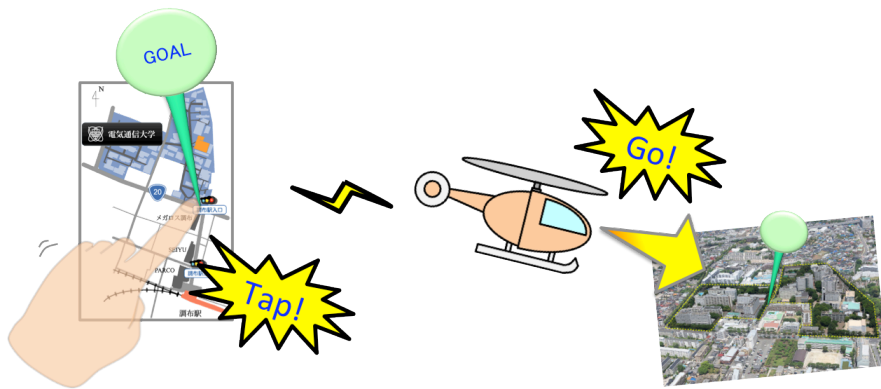


図 3.3.1: 実現したいアイデア

3.4 作成された経路を移動するための自律ナビゲーションシステム

3.4.1 自律ナビゲーションシステム実現のためのアイデア

お供ロボットが自律ナビゲーションを行うにあたり

- 機体の情報を経路設計・ナビゲーションに利用できるよう送信
- ナビゲーションに適した指示を受信
- 複雑な経路も飛行

が必要な機能となる。経路を辿るためには機体の現在地や向いてる方角がわかる必要がある。そのため、これらを実現するためには、下記を満たす飛行ロボット端末が必要である。

1. 機体自身の現在地情報
2. 機体の向いている方角情報
3. インターネットに接続可能

1,2の取得には、Tianmiaoらと同様GPSセンサ・電子コンパスを飛行ロボットに搭載することで実現する。これらのセンサを利用し、ロボットが現在地から行き先となる地点まで一直線に向かうことを基本動作とする。

現在地点からユーザの望む地点まで作成された経路を飛行させるためには、ルートの変更する地点に経由地点を設定する。ロボットは行き先までを一直線に向かうため、その地点を経由地点とすることでロボットに経路通りの道を飛行させることができる。ロボットが経由地点に到達後、ロボットの行き先を次の経由地点へと移行する。この動作を、ロボットの行き先がユーザの望む地点になるまで繰り返す。これにより、ロボットは作成された経路を飛行しながらユーザの望む地点へと移動することができる。

3.4.2 風の影響をキャンセルするための補正提案

飛行ロボットは風の影響を受けると、流されてしまい経路通りの飛行ができなくなってしまう。また、GPSセンサは狭い通路やトンネル内、大きな建造物の近くでは精度が落ちてしまいデータを誤取得する恐れがある。自律ナビゲーションシステムを実現するためには、このような風に流されてしまう問題やGPSの誤取得を解決しなければならない。そこで、風に流された時の補正及びGPSデータ誤取得の対策を提案する。

風に流された時の補正動作を図3.4.1に示す。飛行ロボットに搭載された加速度センサを利用することで、飛行ロボットの速度を算出する。ナビゲーションにより前進や旋回の指示を出すときの速度データと、飛行ロボットが指示を受けた後の速度データを比較することで、指示通り飛行ロボットが移動しているかどうかを確認できる。受けた指示とは異なる方向に進んだことを検出した場合は、風に流されたと判断し、誤って進んだ方向とは逆向きの移動指示を出すことで補正を行う。

3.4.3 GPS の誤取得を防ぐための補正提案

GPS データの誤取得対策には飛行ロボットの速度データ，と過去に取得した GPS データを利用する．飛行ロボットはナビゲーションアルゴリズムにより絶えず GPS 及びコンパスデータから目的地までの距離・方角を計算し，前進や旋回の指示を出しながら目的地到達を目指している．このナビゲーション実行中に GPS の測定に外乱があることを想定し，それまで取得した飛行ロボットの速度データ及び過去の GPS データを元に現在地を推定する．図 3.4.2 に推定を行いながら，移動する例を示す．

case1 のように，測定値と推定地が近似する状態の時は，今までと同じように測定値を利用することでナビゲーションを継続する．

case2 のように，測定値と推定地の差が大きい状態の際は，GPS センサが誤取得したと判断し，推定地をロボットの現在地と想定することでナビゲーションを行う．次の推定には今回の推定地を元としたものと，測定値を元としたものの二通りを用意する．

case3 では，全ての推定地と測定値が異なる状態を示す．このケースでは，GPS の測定ミスと判断し，2つの推定地のうち測定値に最も近い推定地を現在地として想定し，ナビゲーションを行う．次の地点の推定地には測定値及び採択された推定地を元に位置推定を行う．

case4 では，いずれかの推定地と測定値が近似する状態を示す．この状態の時は，近似した推定地が正しいと判断し，測定値を利用してナビゲーションを行う．次の地点の推定には case1 と同様に測定値を利用する．

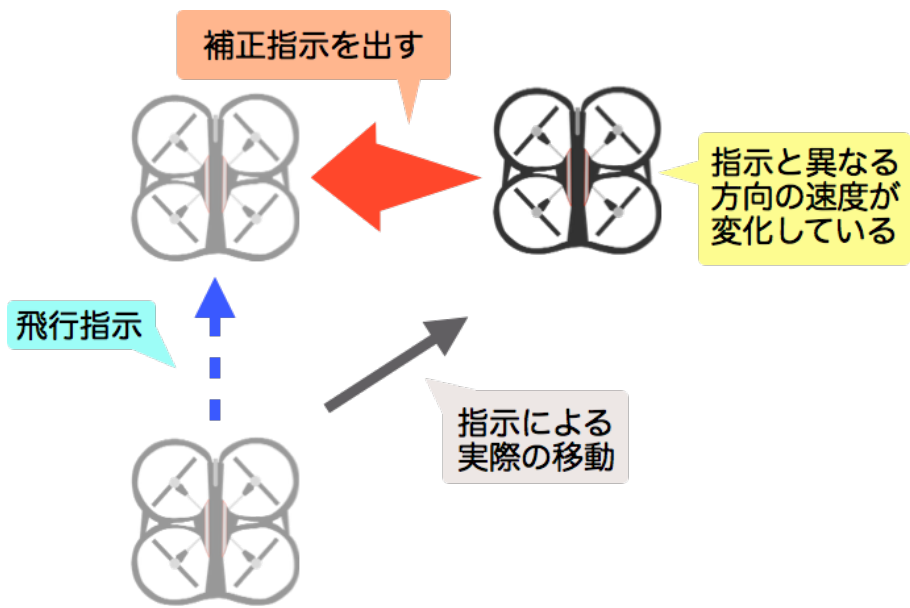


図 3.4.1: 風の揺らぎによる補正指示

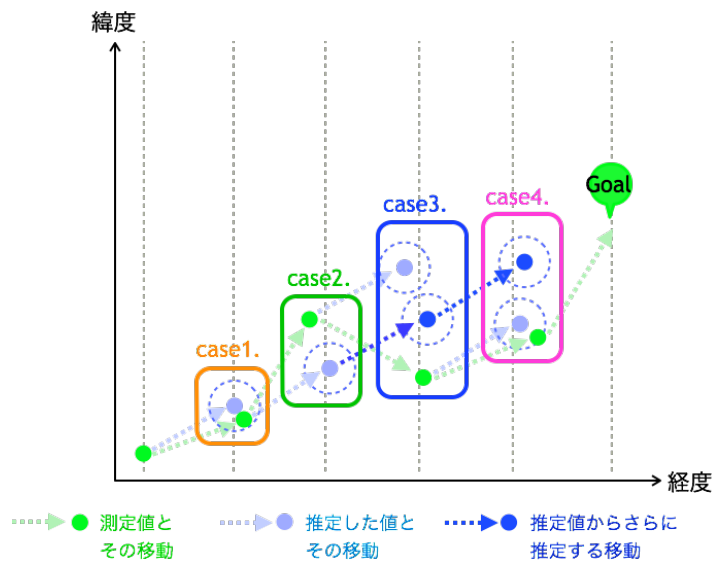


図 3.4.2: 移動速度による自己位置推定

3.5 歩行検知システム

ロボットがユーザの歩行に合わせて一緒に移動できるようにするため、歩行を検知できるようにする。ユーザの止まっている状態・歩行状態を検知する度、ロボットに知らせることができれば、図 3.6.1 のように、ユーザが歩いているときはロボットの自律ナビゲーションシステムを動作させ、ユーザが止まったときはシステムを一時的に止めることができるようになる。

ユーザの歩行を検知するために 3 軸加速度センサを利用する。関連研究より加速度センサを利用することで悪い歩行を検知したり歩数計として機能しているため、このセンサを利用すれば人間の歩行状態を検知することができる。

3.6 提案したシステム同士の連携

地図による目的地設定、飛行ロボットのナビゲーション、歩行検知による飛行ロボットへの指示を連携して動作するようにするため、情報の収束場を作成する。図 3.6.2 のように収束場にユーザが発信する目的地の指示や飛行ロボットへの制御情報、GPS・コンパス情報及びその他ロボットの情報を集積することでシステム同士を連携することができる。

ユーザにより決定した目的地は収束場にて保持し、飛行ロボットの現在地情報を収束場へ送ることで、保持した目的地との 2 地点間の経路作成を行うことができる。ナビゲーション中、飛行ロボットの GPS・コンパス情報、速度情報を随時収束場へ送ることで、収束場での行き先までの残り距離や方角を計算することができるようになり、飛行ロボットへ制御情報のみ返すことができる。また、収束場で飛行ロボットから送られてくる情報を全て蓄積することで、GPS 補正システムを導入した状態で飛行ロボットへ制御情報を送り返すことができる。また、ユーザの歩行を検知する度、収束場へ歩行状態を送信することで、飛行ロボットへの適切な指示を送信できるようになる。

このように収束場を作成することで、ユーザからの指示や飛行ロボットの制御における相互のやりとりが可能となる。

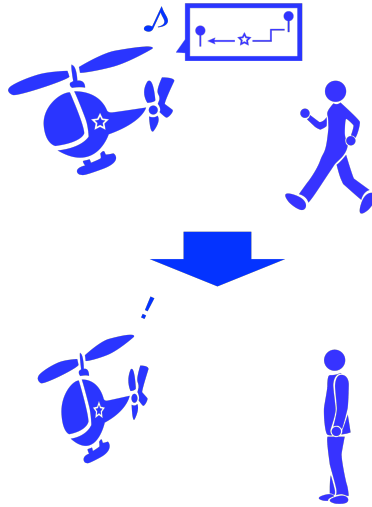


図 3.6.1: 歩行検知による飛行ロボットの動き

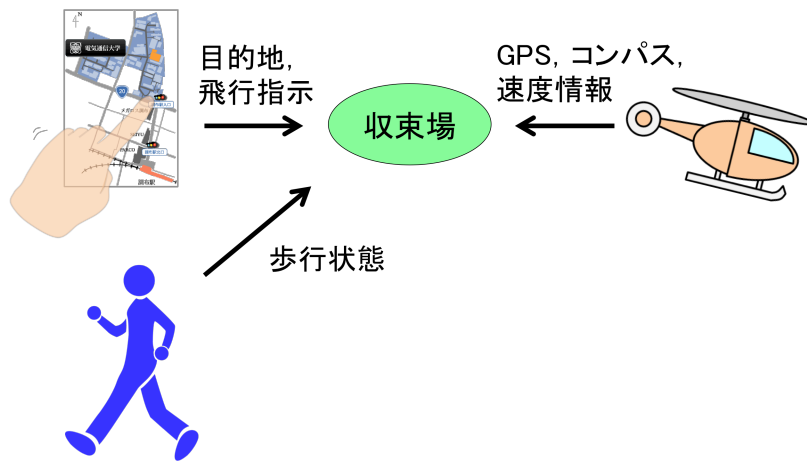


図 3.6.2: 収束場に送信されるデータ

第4章 実装

4.1 本システムの概要

簡単な目的地設定を実現するため、スマートフォンに着目した。スマートフォンはタッチパネルを搭載し、インターネットに接続できるだけでなく、GPS センサや電子コンパス、加速度センサ、ジャイロセンサなどといった様々なセンサを有する携帯情報端末である。スマートフォン用の地図アプリケーションを作成すれば、ユーザがタッチして目的地の情報や制御指示を家庭用ロボットに送ることが可能となる。また、飛行ロボットは AR.Drone[17] を利用する。AR.Drone は自身の前方と腹部にカメラを搭載し、スマートフォンや PC と Wi-Fi 接続することで操縦可能となる。また、4 翼で飛行するクアッドリコプターであるため、ヘリコプターよりも安定した飛行を行うことができる。さらに、AR.Drone は安価な価格で手に入れることができるため、ソフトウェアやアプリケーションのプロトタイプ開発や導入が行いやすい。

提案したシステムを実現するため、スマートフォンには iPhone[18] を利用し、AR.Drone へ制御指示を促す端末としてノート PC を利用して実装した。また、収束地点は PC 上にサーバアプリケーションとして XAMPP[19] を稼働させることで実装した。iPhone、ノート PC、サーバアプリケーションのスペックを表 4.1-表 4.3 に、全体のシステム構成図を図 4.1.1 に示す。

ユーザが利用する iPhone より HTTP 通信を行い、目的地のデータをサーバに送信・格納する。この時、同時に AR.Drone の現在地情報及び Google Map の API を利用する。これにより、現在地から目的地までの 2 地点間のデータのルート設計を自動的に行う。また、iPhone から AR.Drone への制御開始の飛行指示は、HTTP 通信で一度サーバへ送られる。その後、ソケット通信により PC に制御信号が送信され AR.Drone へ指示が届く。iPhone がユーザの歩行状態を検知した際も同様に、HTTP 通信によりその情報がサーバへ送られ、ソケット通信により PC に歩行状態のデータが送信される。歩行状態により適切な指示を AR.Drone へ送る。

AR.Drone には、GPS センサ及び電子コンパスが搭載されていないため、図 4.1.2 のように iPhone を AR.Drone に装着することで実装した。iPhone によって取得された現在地情報、コンパス情報は HTTP 通信でサーバに送信・格納される。その際、サーバではこれらの送信された情報と目的地の位置情報を利用することで、目的地までの距離・方角を算出し、AR.Drone への指示を定める。そしてソケット通信にて PC に制御信号を送信し、AR.Drone を制御する。

表 4.1: iPhone4 のスペック

OS	iOS4.3.5
ROM	32GB
RAM	4GB
CPU	Apple A4 1GHz
重量	137g

表 4.2: PC のスペック

OS	Windows7 Home Premium (64bit)
ROM	128GB (SSD)
RAM	4GB
CPU	1.80GHz Intel Core i7-2677

表 4.3: サーバアプリケーションのスペック

アプリケーション	XAMPP 1.8.2
apache バージョン	2.4.4
MySQL バージョン	5.5.32
PHP バージョン	5.4.19
Perl バージョン	5.16.3

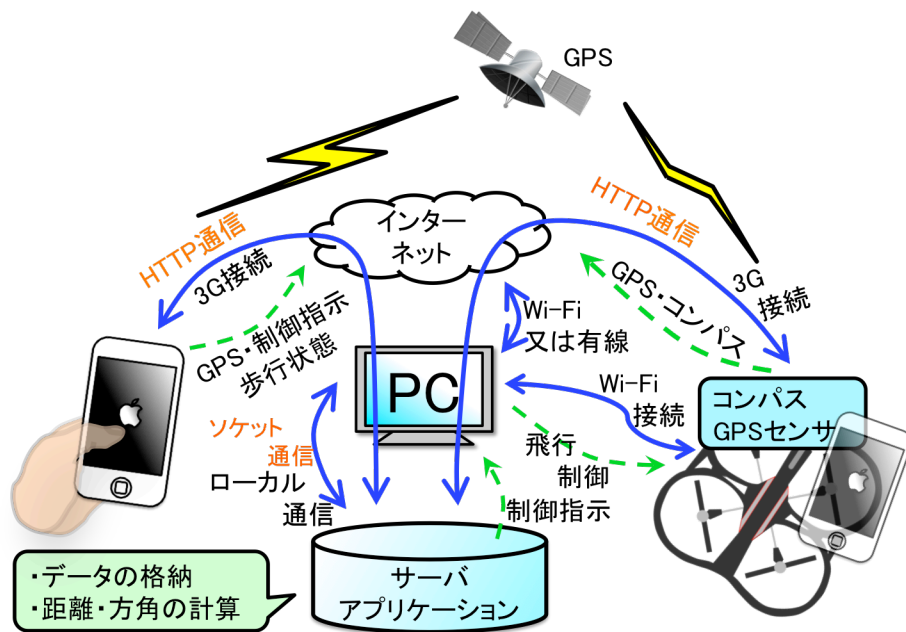


図 4.1.1: システム構成図



図 4.1.2: スマートフォンを装着した AR.Drone

4.2 地図アプリケーションの実装

4.2.1 タッチ指示による目的地設定

飛行ロボットの目的地設定を行うため、手持ちの iPhone 用の地図入力アプリケーションを開発した。

ユーザが目的地を設定するために必要となる動作は

- 目的地となる目印の表示
- 目印の微調整
- 目的地の決定

の3つだけである。

図 4.2.1 に自動ルート設計を行った際の目的地設定の手順を示す。地図入力アプリケーションを開き、おおまかな目的地を2回タップすると目的地の目印としてピンが表示される。その後、表示されたピンを押しっぱなしにすると、ピンが移動可能状態となる。その状態で指を動かしてピンを移動しながら微調整を行う。指を離すとゴールの位置を決定するための選択肢が表示されるため、OK ボタンを押す。OK ボタンは自動ルート設計を行うか否かの2種類存在する。「自動ルート設計を行う」を選んだ場合、目的地情報がサーバに送信され、その情報を元に Google Map の API が稼働し AR.Drone から目的地までのルート設計が行われる。その後、設計されたルートの進行方向が変わる箇所に経由地点としてピンが表示される。AR.Drone はこのピンを順番に経由することで目的地を目指す。

4.2.2 手動での経路設計

飛行ロボットは自身の現在地から経由地点及び目的地までを直線的な移動を行う。そのため図 4.2.2 の左のような、現在地と目的地の2地点間上に建物等の静的障害が存在する場合、飛行ロボットは経路を通過することができない。そこで、このように経路作成に失敗するような場合は、スマートフォン上でユーザ自身が経路を指定できる。目的地設定と同じ要領でだぶるタップによりピンを表示し、位置決定の選択肢において「経由地点」を選択することで、図 4.2.2 の右のように通過可能な経路を設定できる。これにより、経路設計に失敗するようなルートや、敷地内などを通る細かいルート設計も可能となる。



図 4.2.1: 目的地設定の手順



図 4.2.2: 飛行ロボットの通過不可能な経路と可能な経路

4.3 自律ナビゲーションシステムの実装

飛行ロボットは現在地から経由地点及び目的地まで直線的に進むことを繰り返すことで、目的地の到着を目指す。飛行ロボットからはGPS情報が更新される、または電子コンパス情報が複数回更新される度、飛行ロボットの現在地及び電子コンパスの最新情報がサーバに送信される。これらのデータのサーバへの送信頻度は0.5 1.5秒に1度である。現在地やコンパス情報がサーバに送信されると、それらの情報と設定された目的地情報を用いて図4.3.1のフローによって、AR.Droneに送信するための制御信号が定められる。図4.3.1中で登場する距離と方角の計算には、ヒュベニの距離計算公式や球面三角法を利用した。図4.3.1で実装した関数の式を以下に示す。距離計算を(1)、方角計算を(2)に示す式によって実装を行った。ここで、 $gpsXlat$ は $gpsX$ の緯度、 $gpsXlng$ は $gpsX$ の経度の数値を示す。また、 M は子午線曲率半径、 N は卯酉線曲率半径である。

$$\begin{aligned} calcDistance(gpsA, gpsB) \\ = \sqrt{(M \times \Delta P)^2 + (N \times \cos(P) \times \Delta R) \\ \times (N \times \cos(P) \times \Delta R)} \end{aligned} \quad (4.3.1)$$

$$\begin{aligned} P &= \frac{(gpsBlat - gpsAlat)}{2} \\ \Delta P &= gpsBlat - gpsAlat \\ \Delta R &= gpsBlng - gpsAlng \\ M &= \frac{6334834}{\sqrt{(1 - 0.00674 \times \sin(P) \times \sin(P))^3}} \\ N &= \frac{6377397}{\sqrt{1 - 0.00674 \times \sin(P) \times \sin(P)}} \end{aligned}$$

$$calcHeading(gpsA, gpsB) = 90 - atan2(\Delta x, y) \quad (4.3.2)$$

$$\begin{aligned} \Delta x &= gpsBlng - gpsAlng \\ y &= \cos(gpsAlat)\tan(gpsBlat) \\ &\quad - \sin(gpsAlat)\cos(\Delta x) \end{aligned}$$

飛行ロボットの現在地と目的地の2地点間の距離を算出し、算出した値が目的地到着の許容範囲内となった場合、目的地に到着したものとみなし、到着した地点が目的地の場合は「着陸」、経由地点の場合は「一時停止」の指示がPCへ送信される。到着の許容範囲の値は変更できる。また、目的地に到着していない場合は、2地点間の方角が算出され、ロボットの電子コンパス情報との比較を行う。算出した方角に任意の値を加算・減算し、許容範囲を定める。図4.3.2のように、算出された許容範囲内にロボットの電子コンパスが含まれていれば前進、加算した角度よりもコンパスの値が大きければ左旋回、減算した角度よりもコンパスの値が小さければ右旋回の指示が出される。この範囲を決める任意の値も変更可能である。

このように、「飛行ロボットの情報をサーバに送信→距離・方角の計算が算出→PCを通じて飛行ロボットへ指示」の流れを繰り返すことで、図4.3.3のような移動を行い目的地へ到達する。

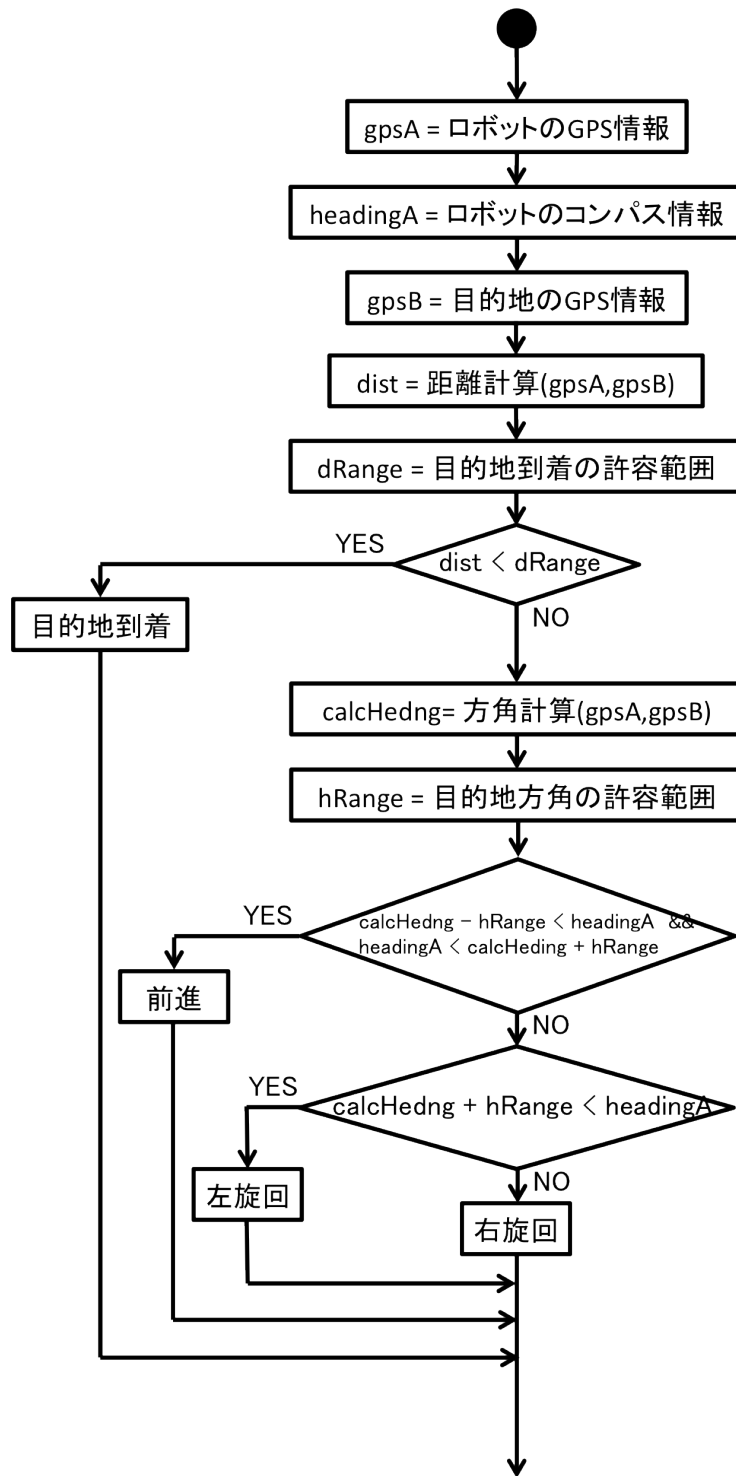


図 4.3.1: 制御信号決定までのフロー

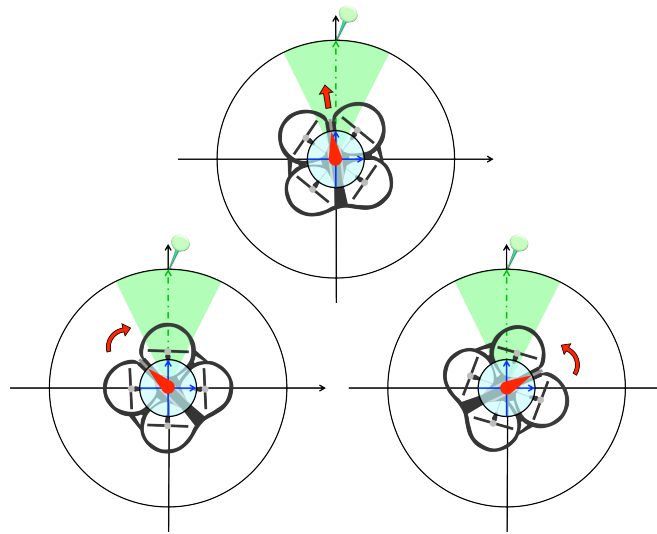


図 4.3.2: 前進・旋回の指示が出される条件

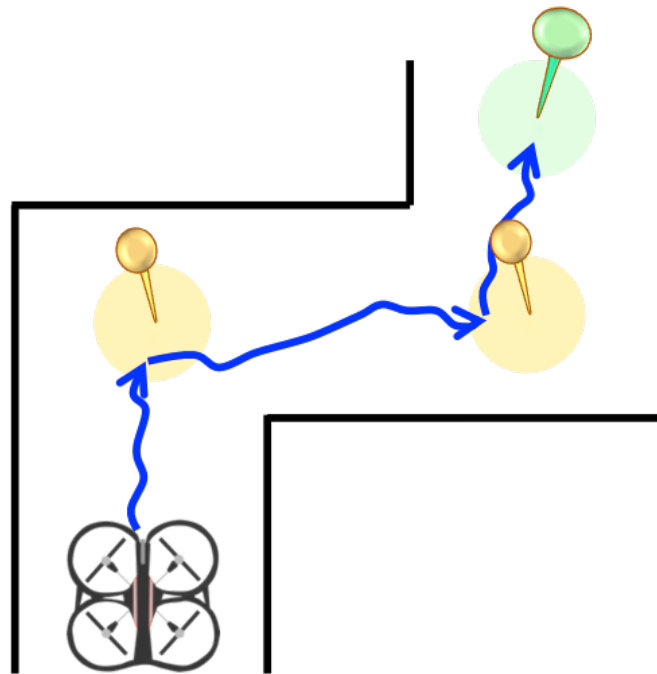


図 4.3.3: 目的地までの移動イメージ

4.4 歩行検知システムの実装

iPhone に内蔵された加速度センサを利用することで、ユーザの歩行状態と止まっている状態を検知するシステムを実装した。このシステムは iPhone をユーザのズボンやパーカーのポケットに入れることで利用することを前提としている。加速度センサは 0.1 秒ごとに更新される。また、x, y, z 軸の合成値を利用することで、iPhone をどの向きにポケットに入れても歩行状態の検知が可能となった。合成値 a の計算式を以下に示す。

$$a = \sqrt{a_x^2 + a_y^2 + a_z^2} \quad (4.4.1)$$

図 4.5.1 は実際にユーザが 5 歩歩いて止まった時に取得した加速度センサの様子である。グラフにおいて、上に凸の値を取得し、1 秒以内に下に凸の値を取得した時、それらの差が case2 のように大きければ「歩行」、止まっている状態から検出した場合は「歩き始め」とみなす。case1 では上下の凸の値の差が小さいため、「歩行」とはみなされていない。

また、加速度の最新のデータ 3 つ分を保持しておき、それらのデータがどの程度増減しているかを計算する。増減が case3 のように小さかった場合、「止まっている」、歩行中に検出した際は「止まり始めた」とみなす。

このような「歩き始め」、「歩行」、「止まっている」及び「止まり始め」の検知を行った際、その情報をサーバ経由で、PC に送信する。ユーザが歩いているときは、サーバから送信の指示をそのまま飛行ロボットに出すが、ユーザが止まっている状態でサーバから「前進」の指示が出た時は、「前進」の代わりに「一時停止」の指示を飛行ロボットに出す。

4.5 システムの処理の流れ

以上より、目的地設定から飛行ロボットの自律制御までの流れ及び歩行検知した際の流れを図 4.5.2 に示す。ユーザがスマートフォンで目的地設定を行った際、そのデータをサーバで格納し、Google Map の API を利用し経路データをスマートフォンで表示する。その後、飛行指示をサーバ経由で飛行ロボットに送信することで、ロボットの自律制御が始まる。制御開始後は、GPS・電子コンパスのデータがサーバに送信される度、それらのデータを格納すると同時に目的地までの方角と距離計算を行い、計算結果から導出された制御信号を飛行ロボットに送信することを繰り返す。また、ユーザの歩行状態を検知する度、それらのデータを PC へ送ることで、PC 上で常に最新の歩行状態を保持する。これらの情報を総合して利用することで、ユーザと共に目的地到着を目指す。

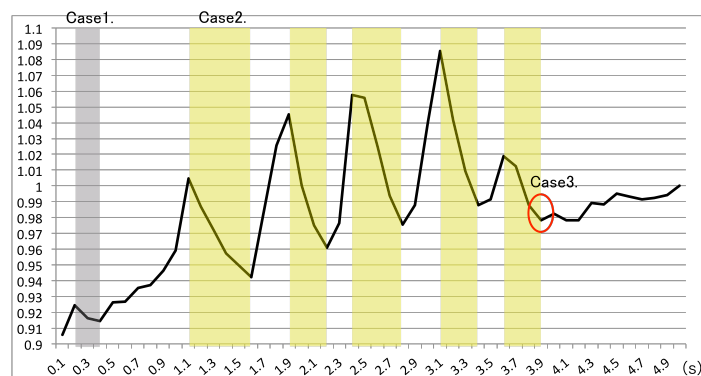


図 4.5.1: 歩行と止まる時の検知の様子

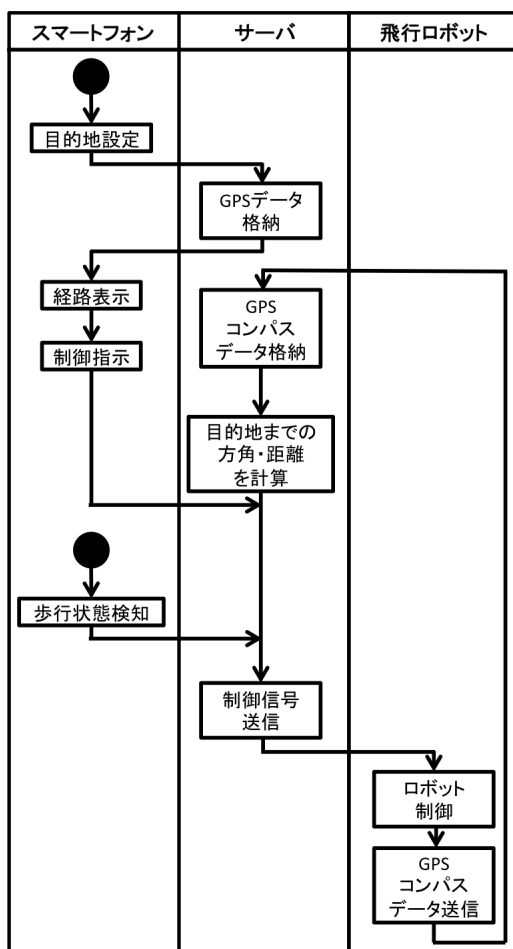


図 4.5.2: システム全体のフロー

第5章 システムの評価

5.1 地図アプリケーションの評価

構築した地図アプリケーションを iPhone 端末に導入し、本システムにおける目的地の設定方法がユーザにとって簡単な UI であるかを評価を行った。20-70 代の被験者 14 名（男性 11 人、女性 3 人）に、初めにどのような操作を行ってほしいかを説明し、iPhone 端末を用いて自由に目的地の設定を行ってもらった。その後、アンケートに回答してもらったことで調査を行った。アンケート内容は、

- 目的地を設定しやすいかどうか
- 設定方法はわかりやすいかどうか
- 設定時に緯度・経度の数値データを意識したかどうか
- 総合的な操作の難易度はどの程度か

であり、それぞれ 5 段階で評価を行い、現在利用している携帯電話や気づいたことや要望・感想も記載してもらった。

また、目的地の設定・アンケート調査を 1 セットとし、各被験者に 2 セット行ってもらったことで、1 回目と 2 回目で使いやすさ及び操作性の評価項目に変化が得られるかどうかを調査した。なお、この調査では地図アプリケーションの UI の評価に焦点を当てているため、飛行ロボットの飛行は行わない。自動ルート設計において、飛行ロボットの代わりにユーザの現在地と目的地の 2 地点間の情報を利用したルート設計を行うことで、飛行ロボットの位置情報を利用した時のフィードバックと同等な結果を得ることができるようにした。

図 5.1.1-図 5.1.4 にアンケート結果を示す。目的地を設定しやすいかどうかの評価結果は、図 5.1.1 のように 1 回目で 80 % 以上の人が設定しやすいと回答し、2 回目では半数を超える被験者が最高数値であると評価した。また、設定方法はわかりやすいかどうかの評価では、図 5.1.2 のように 1 回目で 70 % 以上の人がわかりやすいと回答し、2 回目では 1 名を除くすべての人がわかりやすいと評価した。このことから、一度目的地設定を行い手順を覚えてしまえば、2 回目以降はスムーズに設定することができると言える。

目的地設定を行う際、緯度・経度の数値データを意識したかどうかの調査では、図 5.1.3 のように 1 回目でおよそ 8 割、2 回目ではおよそ 9 割の被験者がこれらの数値情報を意識しないと回答した。このことから、目的地を設定する際、ユーザは緯度・経度の情報を意識することなく、目的地の情報を設定することができると言える。

本アプリケーションの総合的な操作難易度を調査した結果、図 5.1.4 のように、1 回目で 90 %、2 回目では全ての被験者が本アプリケーションでの目的地設定は簡単であると評価した。また、所有する携帯電話としてフィーチャフォンを選択した 20 代、50 代、70 代の 3 名の被験者からは 1 回目の実験ではやや簡単と評価した者が 2 名、普通と評価した者が 1 名であった。しかし 2 回目の

実験ではやや簡単と答えた者が1名，簡単と答えた者が2名となった．このことから，日常的にスマートフォンを利用していない者も，数回操作を行うだけで簡単に操作方法を理解でき，目的地設定を行うことができると言える．さらに，50代や70代といったご年配の方からも「操作が簡単である」との評価を得たことから，今回開発した地図アプリケーションはどのユーザにとっても簡単に利用することができるUIであることが示せた．

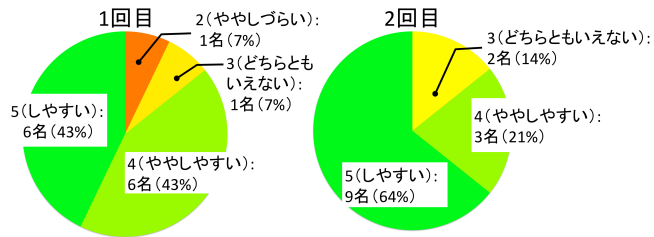


図 5.1.1: 設定しやすいかどうか

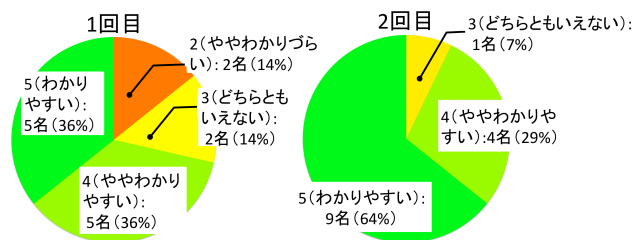


図 5.1.2: 設定方法がわかりやすいかどうか

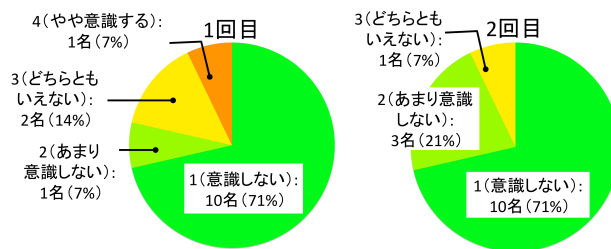


図 5.1.3: GPS情報を意識しないかどうか

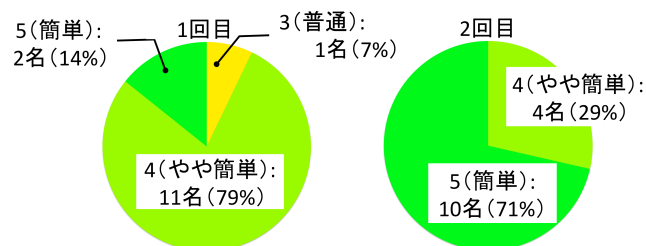


図 5.1.4: 総合的な難易度

5.2 自律ナビゲーションシステムの評価

5.2.1 補正アルゴリズム導入前の飛行実験の評価

まず補正アルゴリズムを適用せずに、構築したシステムにより飛行ロボットが目的地までたどり着けるかどうか図 5.2.1 のような道路を飛行させることで評価を行った。それぞれの地点の GPS の感度は表 5.1 の通りであり、目的地設定及び方角の許容範囲は表 5.2 とした。初めに、図 5.2.2 と図 5.2.3 に成功データを示す。この時、地図アプリケーションにて A 地点に飛行ロボットが存在する状態で B 地点を経由地点、A 地点をゴール地点と設定し、実験を行った。飛行ロボットは A 地点をスタートした後、B 地点まで向かい、B 地点到達後は 180 度方向転換を行い、A 地点へと戻ってくるように飛行した。

図 5.2.2 では移動中の緯度・経度の推移及び経由地点や目的地までの残り距離の関係を表す。この図より、飛行ロボットが目的地に向かって進むことで、目的地までの距離が縮まっていることがわかる。自律飛行開始直後や経由地点通過直後のように緯度経度の値に変化が無く、目的地までの距離が縮まらない場合は、飛行ロボットに前進の指示が出ておらず、旋回の指示が出されている時である。図 5.2.2 下の目的地までの距離に注目すると、経由地点通過直後、距離が大幅に上昇している。これは飛行ロボットが経由地点である B 地点に到達したことで、目的地が B 地点から A 地点へと更新されたためである。この直後、目的地までの距離が縮まらない状態が続いた後、再び距離が縮み始めている。このことから、目的地変更に伴って飛行ロボットに左右の旋回の指示がしばらく送られた後、前進の指示が出されていることがわかる。

また図 5.2.3 では、実際のコンパスの方角及びロボットの位置から目的地までの 2 地点間の算出方角とそれに基づく飛行ロボットへの制御指示を示す。飛行ロボットの取得データが算出データに近似していれば前進の指示が、近似していなければ算出データに近づくよう旋回の指示が出される。

飛行開始直後、前進ではなく右旋回の指示が 20 秒程出続けているが、これは開始時に飛行ロボットが経由地点から 90 度西方向を向いた状態から実験を開始したためである。またこの時、およそ 250 度の方角から 360 度を経由して 20 度の方角へと旋回している。これは 250 度から時計回りに 20 度するよりも、250 度から反時計回りに 20 度を目指す方が旋回する角度が小さいためである。経由地点到着後、算出される方角の情報が大幅に変わるが、これは図 5.2.2 の時と同様に目的地が B 地点から A 地点へと更新されたからである。目的地が更新された直後、右旋回の指令が出続け、それに伴いロボットの方角が算出された方角へと近づいている。ロボットの向く方角が算出方角の許容範囲内まで近づくと、再び前進の指示が出ている。このことから目的地と異なる方向を向いていても修正を行うことができ、経由地点到達後の大幅な方角の変化にも対応できることが示せる。

上記の成功例を含め計 14 回の実験を行った。本実験に関して、飛行ロボットの危機回避の実装を行っていないため、すれ違う人や障害物にぶつかりそうになった場合などのやむを得ない際は、手動で回避している。しかし、ごくわずかな左右の移動と一時停止の指示のみであり、自律ナビゲーションにより出る「前進・旋回」の指示は行っていない。また、危険回避後は手動制御を終了し、再び自律したナビゲーションを開始している。このように地点間移動に関わる連続移動指示は一切行っていないため、GPS 情報から観測できるレベルでは出力される結果のデータへの影響はない。

実験を行った経路、成功回数、失敗要因は表 5.3 の通りである。経由地点がなく A 地点から B 地点を目指す実験は 8 回中 5 回成功している。また、このうち B 地点を向いた状態での実験は 3 回中

2回成功し、B地点とは異なる方向を向いた状態からの実験は5回中3回成功した。実験成功時、飛行ロボットは目的地に向かうように旋回・前進を行った。風や、飛行ロボットに搭載しているiPhoneの重みなどが影響して次第に目的地と異なる方向を向きながら移動する様子も見られたが、その際は目的地の方向への旋回指示が出され、方角が補正された。また、失敗の要因としては、コンパスデータが正確に取得できなかったことやGPSの感度が悪い地点でのGPS誤取得があげられる。コンパスデータが正確に取得できなくなった時は、目標の方角を見失ってしまい左右の旋回を繰り返し続け、前進しなくなってしまった。またGPSを誤取得してしまうと、飛行ロボットの現在地と目的地点までの位置関係がわからなくなってしまいうため、導出してほしい指示とは全く異なる指示が出されてしまった。B地点からA地点を目指す実験を2回行ったが、B地点のGPSの感度が悪かったため、初めから正しい飛行指示を得ることができず1度も成功できなかった。

経路地点を1つ設定し、目的地までの飛行を行えるかどうかの実験は4回中2回成功した。実験した経路はA地点からC地点を経由しD地点を目指す経路と、A地点からB地点を経由しA地点へ戻ってくる経路である。2回ずつ実験を行ったが、成功回数はそれぞれ1回であった。成功時、ロボットは前進・旋回を繰り返しながら経路地点を目指した。経路地点到達後、飛行ロボットの目指す地点は経路地点から目的地点へと更新され、それに伴いロボットは目的地への方角へ旋回し、前進することで目的地に到着した。A→C→D経路の失敗時、ロボットは経路地点を通過し、目的地点を目指していた。しかし、D地点でのGPS感度があまり良くなかったため、目的地を見つけることができず、目的地を通り過ぎてしまい旋回し、再び戻ってくるといった動作を繰り返している最中に、飛行ロボットの電池が切れて自律動作が終了した。A→B→A経路の失敗時、B地点を経由後、順調にA地点へと向かっていたが、途中飛行ロボットに搭載したスマートフォンが不具合を起こしてしまった。これにより、飛行ロボットの現在地、電子コンパス情報を送信することができなくなってしまい目的地に向かう途中で飛行状態のまま停止した。

以上のことから、地図アプリケーションで設計した経路を、GPSセンサと電子コンパスを搭載した飛行ロボットが自律して飛行できることがわかった。しかし、成功率は50%であり、失敗要因の中でもGPSセンサの誤取得が50%を占めることがわかった。また、風による影響も手動で補正を行ってしまっている。このことから提案した風及びGPSの補正システムが必要である。

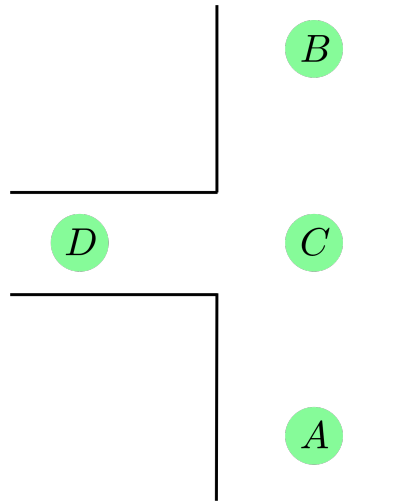


図 5.2.1: 飛行させた道路のイメージ図

表 5.1: 離陸, 着陸, 経路を行う各地点の GPS の感度

地点	GPS 感度
A	良好
B	悪
C	良好
D	やや悪

表 5.2: 目的地までの距離や方角の許容範囲

許容範囲	
目的地到達とみなす範囲	地点から半径 3m 以内
機体が目的地を向いているとみなす範囲	機体と目的地の 2 地点から算出できる方角 ± 10 度

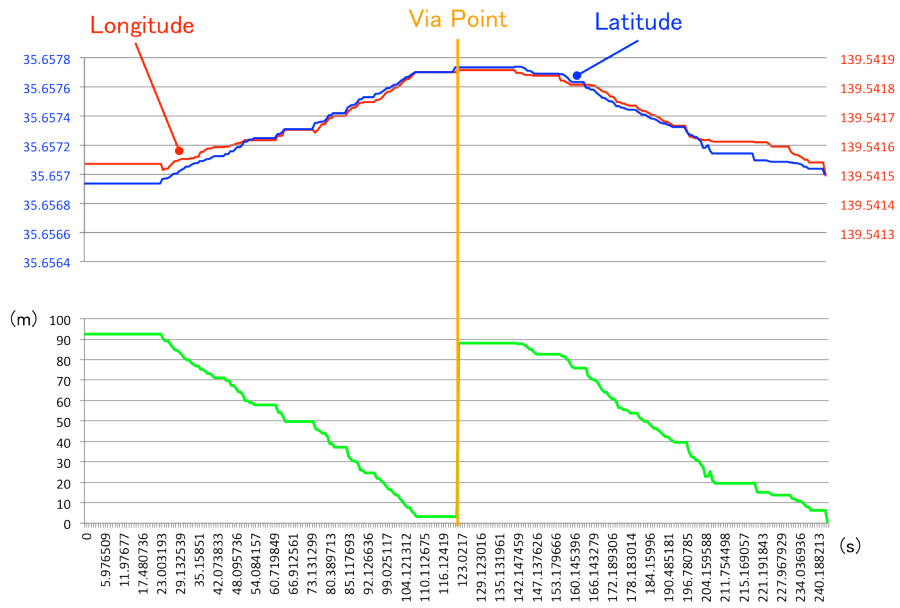


図 5.2.2: 緯度・経度と経路・目的地までの残りの距離

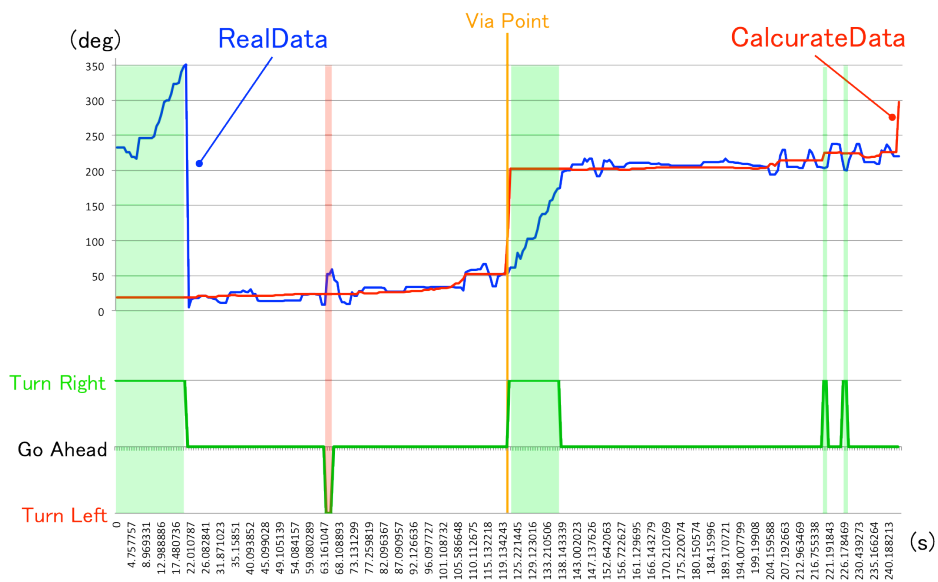


図 5.2.3: コンパスデータの比較と制御指示

表 5.3: 自律飛行の実験結果

経路	成功回数	失敗の要因
A → B	5/8	GPS の誤取得
B → A	0/2	GPS の誤取得, 電子コンパスの不具合
A → C → D	1/2	GPS の誤取得, ロボットの電池切れ
A → B → A	1/2	ロボット上のスマートフォンアプリの不具合
合計	7/14	

5.2.2 風による影響や GPS の誤取得補正システム導入後の評価

提案した補正システムによって、飛行ロボットの目的地到達の成功率が上昇するかどうかの評価を行った。実験は多摩川河川敷と大学院敷地内の2箇所で行った。この時の経路は、多摩川河川敷及び大学院敷地内いずれの場所においても、経由地点を1つ定め、離陸地点と目的地点をそれぞれ同一の場所とした。結果、表 5.4 のように1度も失敗することなく、飛行ロボットは目的地へと到着した。また多摩川河川敷では、風速 3-4m と観測されていたが、手動での制御を加えることなく目的地へと到達した。大学院敷地内に関しては、飛行ロボットの移動中、工事中の仕切りにぶつかりそうになったため、やむなく手動での補正を行った。

図 5.2.4 に、多摩川河川敷での実験成功時の GPS データの実測値のみの軌跡及び推定地を含む軌跡を示す。実測値が推定値と近似しているとみなす範囲は、推定値の半径 1m 以内とした。離陸地点と目的地付近で橙色の丸内のように、実測値のみの軌跡は複雑に入り組んでしまっているが、推定値を含んで飛行した軌跡は滑らかになっている。このことから、この付近では GPS の誤取得が多数見受けられ、推定値を利用することでナビゲーションを行っていたことがわかる。また、飛行中前進して次の指示を待機する間に、ロボットがやや後退してしまうような現象を多く観察した。これは、一時停止しなければならない時に前進した余力で前方向に進んでしまったことが、風による影響で前に進んだとシステムが誤検知してしまい、後退の指示を出したためである。

以上より、提案した補正システムによって、目的地到達の成功率が上昇したことから、これらの提案は有用であることが示せた。

表 5.4: 風の影響や GPS の誤取得補正システム導入後の実験結果

実験場所	成功回数
多摩川河川敷	2/2
電気通信大学敷地内	2/2
合計	4/4

— : 実測値のみの軌跡
— : 推定値を含む軌跡

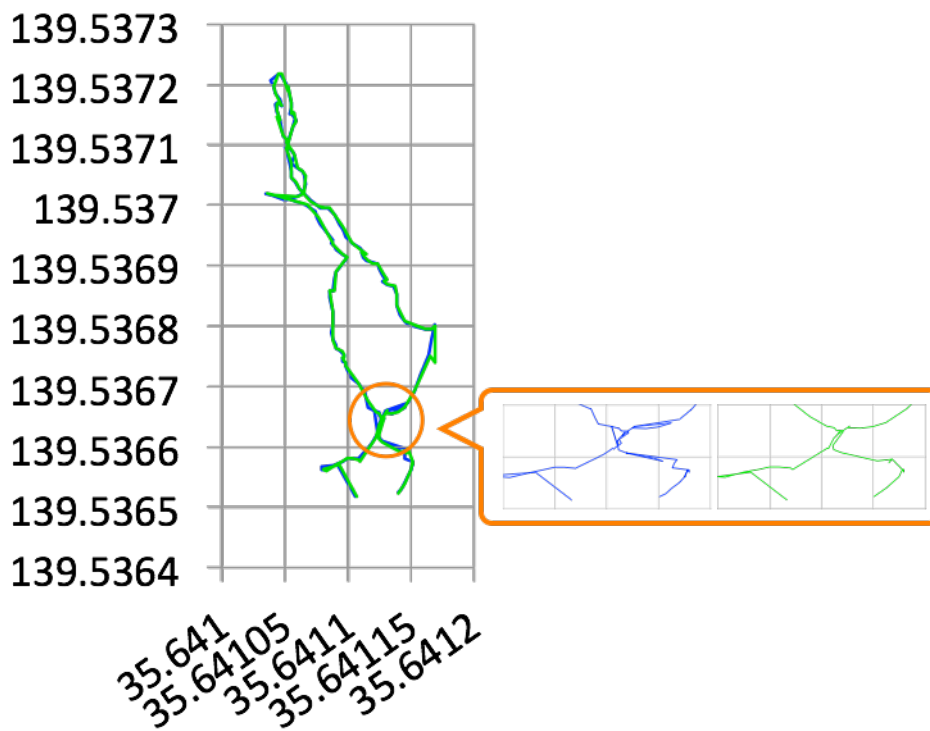


図 5.2.4: 実測値と推定地の軌跡

5.3 歩行検知システムの評価

5.3.1 検知システムの精度

構築した歩行検知システムにより、ユーザの止まっている状態と歩行状態を検知が行えるかの評価を行った。20代の被験者6人（男性5人，女性1人）に、歩行検知アプリケーションを導入したiPhone 端末を着衣しているズボンや衣類のポケットに入れてもらう。その状態で、5歩歩きしばらく立ち止まり再び5歩歩いてもらうことで検知システムの精度の調査を行った。

結果、6人中4人の歩行検知に成功した。図5.3.1-図5.3.4に歩行検知に成功した被験者のグラフを、図5.3.5, 5.3.6には歩行検知に失敗してしまった被験者のグラフを示す。また、図中の黄色い帯は歩行を、赤丸は止まった状態を検知したことを示す。

被験者AはパーカーのポケットにiPhoneを入れて実験を行ったが、歩行及び止まった状態いずれの検知も行えた。前半の5歩目の検知後、止まったことを検知するまでに0.4秒ほどかかっている。後半の5歩の歩行では7歩の検知を行ってしまっているが、これはパーカーのポケットに入れたため、歩行以外の揺れも検知してしまったためである。しかし、本実験では、歩行中であるか止まった状態であるかを判断できれば良いので、歩行中に実際の歩行とは異なる歩数を検知しても問題ではない。また、後半の歩行後、立ち止まった状態を検知できていないが、これは被験者が止まってすぐiPhoneを取り出してしまい、その動作がノイズとなったためである。

被験者B, CはズボンのポケットにiPhoneを入れて実験を行った。両者とも前半の歩行では4歩しか検知しておらず、特に1つ目の帯と2つ目の帯の間隔が大きいことから、2歩目の検知に失敗してしまったことがわかる。しかし、この間止まった状態の検知は行われていないため、問題ではない。また前半後半の歩行終了時から止まった状態の検知までにかかった時間は、被験者Bは0.6秒、0.4秒、被験者Cは1.4秒、0.8秒ずつかかった。被験者Cの止まった状態の検知が遅れた理由として、歩行終了時、取得した加速度の値が上下に凸になった状態から立ち止まったためである。

被験者DはズボンのポケットにiPhoneを入れて実験を行った。図5.3.4を見ると、他の成功した被験者よりも取得した加速度データの揺れが大きい。しかし、このような波形のデータからも歩行及び止まっている状態の検知に成功している。歩行終了時から止まった状態の検知までにかかった時間は、前半後半でそれぞれ0.7, 1.1秒ずつかかった。

被験者E, Fは歩行検知に失敗している。被験者Eは歩行中に最初に出現する赤丸のような波形が出てきたことによって、歩いているにも関わらず止まった状態であると誤検知してしまった。被験者Eはすり足気味に歩くことが癖になっており、その波形が歩行中ノイズとして表れ、誤検知を引き起こしてしまった。

被験者Fは小柄な女性であったため、他の被験者よりも加速度データの揺れが小さくなった。結果、上下の凸の差が歩行とみなされる閾値を超えず歩行の検知がほとんどされなかった。後半はある程度歩行を検知しているが、波形の揺れが小さいため歩行途中にも止まった状態であると誤検知してしまっていることがわかる。

以上のことから、本システムでは歩行・止まった状態を検知できるが、すり足の歩行や波形の揺れが小さくなってしまいうような歩行に関しては、止まった状態を誤検知してしまうことがわかった。また、歩行直後から止まったことを検知するまでに平均0.77秒かかることがわかった。

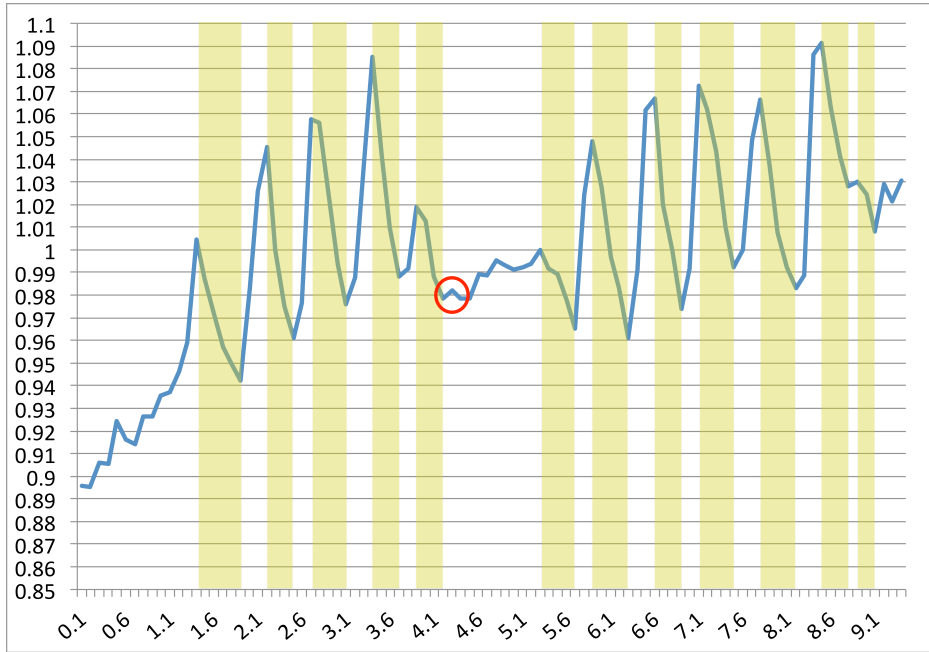


図 5.3.1: 被験者 A のグラフ

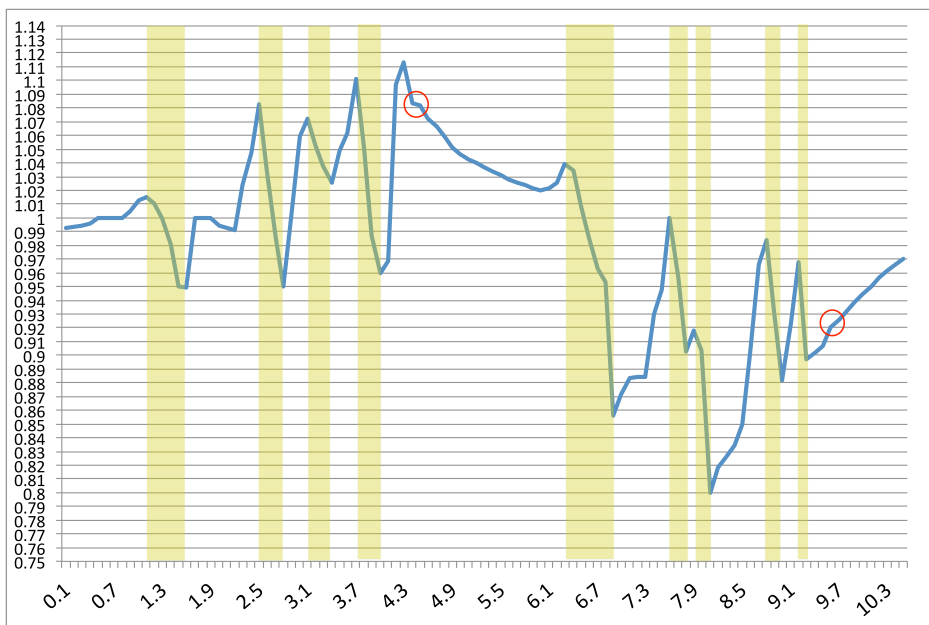


図 5.3.2: 被験者 B のグラフ

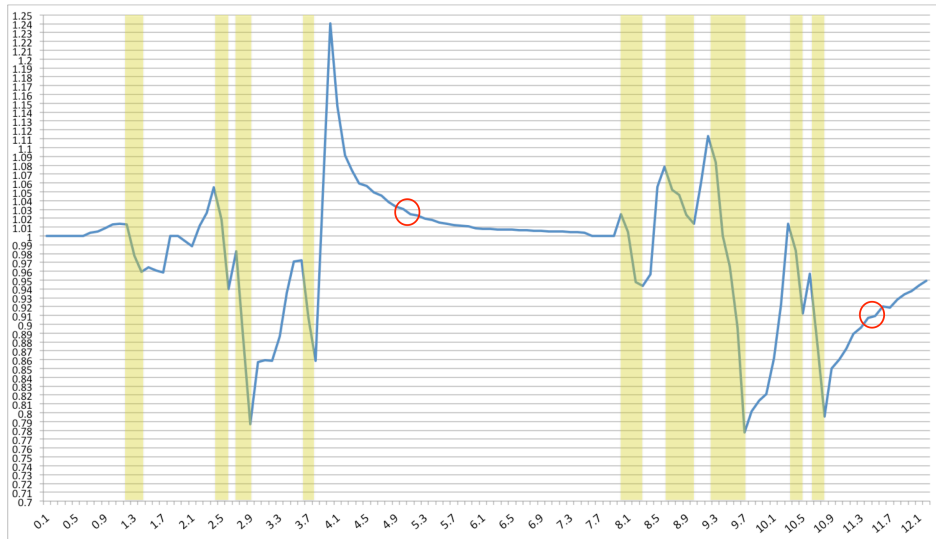


図 5.3.3: 被験者 C のグラフ

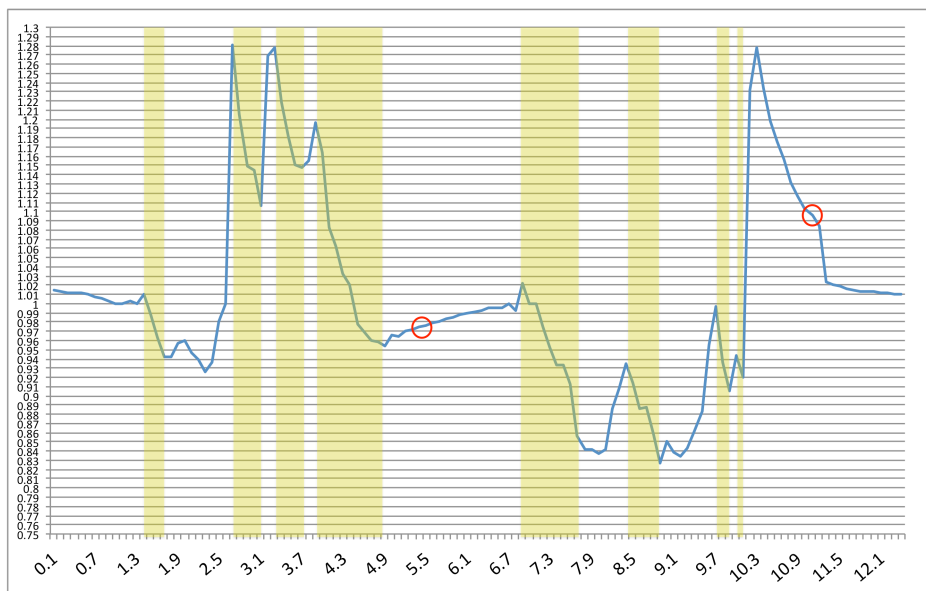


図 5.3.4: 被験者 D のグラフ

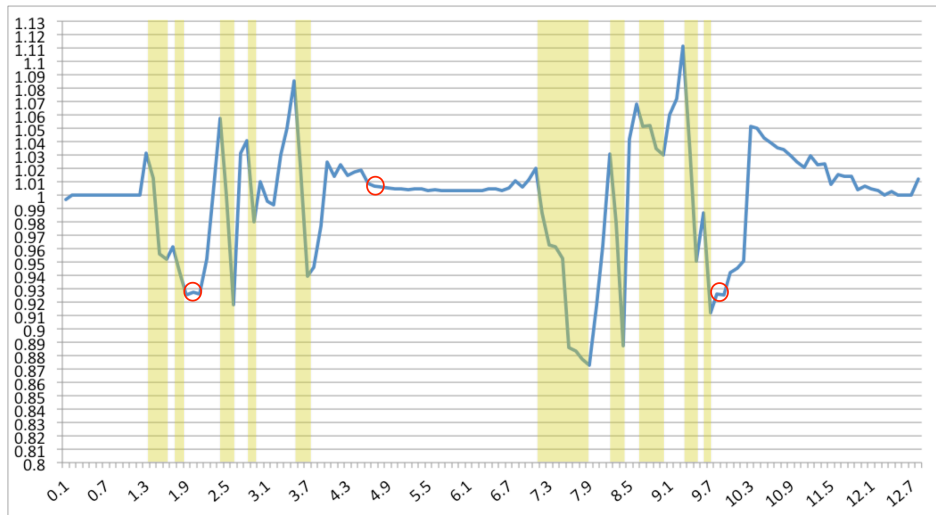


図 5.3.5: 被験者 E のグラフ

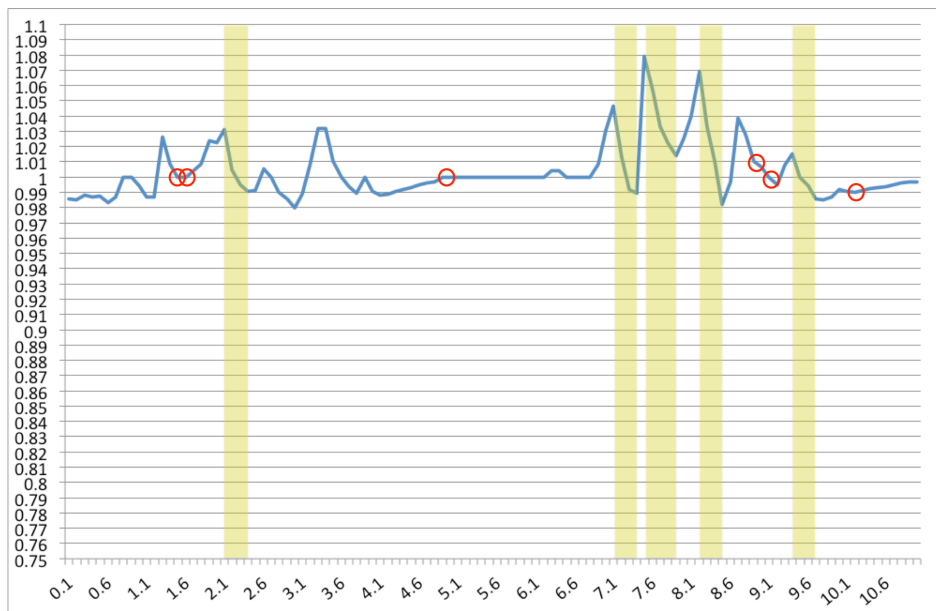


図 5.3.6: 被験者 F のグラフ

5.3.2 検知から飛行ロボット制御までにかかる時間

上記の歩行検知システムを利用して、歩き始めてから飛行ロボットが移動するまでにかかる時間、及び立ち止まってから飛行ロボットが移動をやめるまでにかかる時間の調査を行った。歩行検知の精度を調査した被験者とは異なる 20 代の被験者 5 人（男性 5 人）に、歩行検知アプリケーションを導入した iPhone をズボンのポケットに入れてもらう。この時、歩き出す足と同じ方のポケットに入れてもらった。飛行ロボットは被験者が歩くと前進し、止まると一時停止するようにして、被験者の斜め前を浮遊させた。この状態で被験者にはストップウォッチを持ってもらい、歩き始めると同時にタイム計測を行い、飛行ロボットが動き出したところで計測を終える。同様に、止まり始めると同時に計測を行い、飛行ロボットが停止し始めたところで計測を終えてもらった。

この実験結果のグラフを図 5.3.7 に示す。歩き始めた時の計測時間は平均 1.58 秒であり、かかった時間の誤差が最大で 0.5 秒と小さい。かかった時間の誤差が小さい理由として、立ち止まった状態から歩行を開始するので、波形が安定した状態から動き始めるので検知することが容易であるためである。

これに対し、止まった時の計測時間は平均 1.56 秒と歩き始めた時の計測時間に勝るが、かかった時間の誤差は最大で 1.3 秒離れてしまう。これは先ほどの精度評価で挙げたように、歩行終了時、取得した加速度の値が上下に凸になった状態から立ち止まったためである。

以上より、歩行や止まった状態を検知してから飛行ロボットを制御するまでにおよそ 1.5 秒程度であることがわかった。

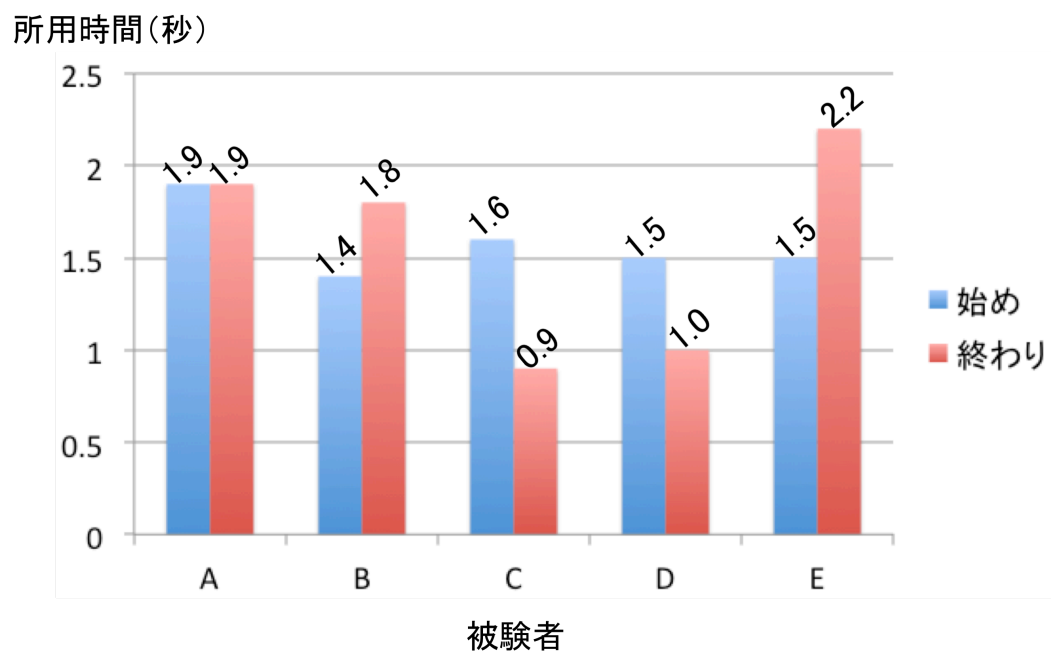


図 5.3.7: 歩き始めと終わりの検知からロボット制御までの時間

第6章 結論

6.1 本研究の成果

本論文では、「浮遊ノート」システムにお使い機能を持たせることを提案した。従来の「浮遊ノート」システムはユーザのお供ロボットとしてユーザの傍を飛行しながら、ユーザをサポートする存在であった。これにお使い機能システムを追加することで、ユーザから離れていてもユーザの手助けを行うことができるようになる。さらに、道案内機能では、ユーザが止まっているときは飛行ロボットも一時停止し、ユーザが歩き始めたら飛行ロボットもナビゲーションを再開できる。これによりユーザが転倒などにより動けなくなっても、飛行ロボットがナビゲーションを停止することができる。お使い機能を実装するにあたり、本論文では以下を実装した。

- 目的地設定を簡単に行うための地図アプリケーション
- 飛行ロボットの指定地点へのナビゲーションアルゴリズム
- 風の影響や GPS の誤取得に対する補正
- ユーザの歩行アクションと飛行ロボットの連動

実装には、ユーザが飛行ロボットに指示を与えるために利用する携帯情報端末として iPhone を、飛行ロボットは AR.Drone を用いた。また、GPS センサや電子コンパスを飛行ロボットに搭載するため、AR.Drone 上に iPhone を乗せて実装を行った。また、歩行検知には iPhone の加速度センサを利用した。

ユーザは手持ちの iPhone より本システムの専用アプリケーションに表示される地図の任意の点を指差すことで目的地を設定する。この時、AR.Drone の現在地と目的地の緯度経度情報及び Google Map API を利用することで、目的地までの経路を自動設計する。AR.Drone は離陸後、iPhone 上の GPS センサ・電子コンパスを利用することで設計された経路上を飛行する。また飛行ロボットには風に流されてしまう問題、GPS センサにはトンネルや狭い道などで誤取得してしまう問題が存在する。これらを解決するための補正案を提案・実装した。風に流された時の対処として、飛行ロボットに内蔵された加速度センサを利用した。加速度センサによりナビゲーションシステムから出される指示とは異なる方向への速度が検知された場合、風に流されたと判断し、流された方向と逆向きの移動指示を出すことで補正を行う。GPS を誤取得した際に対処として、飛行ロボットの加速度センサが取得できる速度データ及び過去の GPS の値から現在の位置情報を推定する。過去にナビゲーションに利用した GPS の値に関して、実測値・推定値のどちらを利用したかで次の推定値を決定する。その後、現在の位置情報の実測値と推定値にどれだけ誤差があるかで、実測値と推定値どちらを利用してナビゲーションするかを決定する。この GPS の実測値、推定値のどちらを採用するかを4つのパターンに分類し、実装を行った。

ユーザの歩行状態の検知には iPhone の加速度センサを利用する。iPhone をズボンやパーカーのポケットに入れて歩行することで、加速度センサより歩行の波形を取得する。この時、3軸加速度

の合成値を利用することで、iPhone がどの向きでポケットに入っているとしても同じ結果の波形を利用できるようにした。この波形のデータを利用することで上下に凸になる値を検出し、その差が一定数値以上かつ検出した波形の間隔が1秒以内であれば、歩行とみなし、最新の波形データを観察し、データの変動が小さいことを検知した場合は止まっている状態であるとみなすように実装を行った。

本システム専用アプリケーションにより目的地設定を簡単に行えるかを調査した結果、初めて利用したときでは90%の人が、2回以降の利用では全ての人が簡単に行えると答えた。また、50-70代の高齢者も被験者として含まれていたことから、どの年代の人も地図アプリケーションを利用することで目的地設定を簡単に行えることがわかった。

飛行ロボットはGPSセンサ・電子コンパスを利用することで、作成された経路通り飛行できることを確認した。何も補正しない場合は、成功率は50%であり、かつ風による影響は手動で対応しなければならなかったが、風による補正及びGPSの誤取得に対する補正を加えることで、全ての実験に成功することができ、手動による制御を一切加えずに目的地到達することも可能となった。

ユーザの歩行検知が正しく行えるか調査した結果、67%の歩行を正しく検知した。すり足のよう歩行したり、歩行の踏み出しが弱い場合は歩行中にも関わらず止まり始めとして誤検知することを確認した。また、歩行や止まった状態を検知してから飛行ロボットの制御までにかかる時間を計測した結果、およそ1.5秒であることがわかった。歩き始めはロボットの移動を速め、止まったときは1.5秒で移動した距離分だけロボットを後退させることで、ユーザの歩行と連動させることができるため、道案内機能において本論文の歩行検知システムは有用である。さらに、ユーザと飛行ロボットの距離が離れ過ぎてから一時停止するよりも歩行検知システムの方がロボットのレスポンスが速いため、よりユーザの歩行に連動した移動を行うことができる。

6.2 今後の課題

本論文では、地図をタッチすることで目的地及び目的地までの経路を設計でき、飛行ロボットにそこまで移動させるナビゲーションシステムを実装した。風による機体の流れに対する補正システムを導入したが、現状では前進した余力で前方向に進んだ動作を風に流されたと誤検知してしまっている。この問題は「前進→一時停止」を行う際に作動する風の補正システムの、後方への補正を一時的に解除すれば解決することができる。次に、道案内機能への発展を考えユーザの歩行を検知することで、ユーザが立ち止まった時はロボットのナビゲーションを一時停止させ、歩行時には再開させる機能を実装した。

今回構築したシステムにはロボットが自律的に静的な障害物を回避する機能を持たず、また長距離のトンネルを通る場合はGPSが常に取得しづらい環境であるためナビゲーションすることができない。これらは、2章の関連研究で取り上げた技術を活用することで解決可能である。オプティカルフローセンサを飛行ロボットの前方に複数搭載することで障害物を検知することができ、また飛行ロボットの前方に搭載されているカメラから取得できる画像を解析することで障害物を検知することができるため、回避することができる。人のように動的な障害物が迫ってきた場合は、一時的にその障害物の高さ以上の高度まで上昇してしまえば、避けることができる。長距離のトンネルを通るような際は、飛行ロボットのフロントカメラから取得できる映像にキャニー法・確率的ハフ変換を行うことでトンネルのパニングポイントを見つけ、そのポイントを目指して飛ぶようにモードを切り替える。これらの技術を本システムに導入することで、より安定した自律移動を行うことができるようになる。歩行検知システムによるロボットの動作の遅延を減らすこ

とができれば、飛行ロボットに搭載されたセンサを利用しなくても、信号待ちの時に一時停止することができるようになる。また、ロボット技術の発展により飛行ロボットが重たい荷物を運ぶことができるようになれば、本システムと組み合わせることで、無数の飛行ロボットが街中を飛び交い、買い物をする時代が到来すると考えられる。

謝辞

本論文は電気通信大学大学院情報システム学研究科情報ネットワークシステム学専攻博士前期課程に在籍中の研究成果をまとめたものである。筆者は在籍中研究の他に、kinectを活用し一人でも「ダルマさんが転んだ」が遊べるようなゲームを開発したり、競馬の予想システムをiPhoneアプリとして開発したりと自由奔放な学生生活を送っていた。そんな筆者を、温かく見守って頂きながら、日々研究成果だけでなく、筆者の様々な開発に関して多大なご指導とご助言を下された、情報システム学研究科情報ネットワークシステム学専攻入江英嗣准教授に深く感謝の意を表す。また、同講座の吉永努教授、吉見真聡助教にも、合宿や合同ゼミを通じてご助言とご指導を頂いた。ここに、感謝の意を表す。

最後に、外での実験やダルマアプリの開発に協力してくれた佐久間大輝氏、多摩川河川敷まで自転車で一緒に付き添ってくれた千竈航平氏、さらに様々な実験に全被験者並びに研究室の全メンバーに感謝する。皆様がいてくださったからこそ、最後まで充実した院生生活を送ることができた。

参考文献

- [1] 吉見 卓松日樂 信人. 人と共存する生活支援ロボット. 東芝レビュー Vol.60 No.7, 2005.
- [2] アイロボット社. ルンバについて. <http://www.irobot-jp.com/roomba/>.
- [3] 佐藤由紀子. Amazon、ドローンでの配送サービス「Prime Air」構想を発表. ITMedia. <http://www.itmedia.co.jp/news/articles/1312/02/news056.html>.
- [4] 独立行政法人産業技術総合研究所. 次世代ロボットに関する市場調査・市場予測の比較と分析. <https://staff.aist.go.jp/h.arai/robotics/rsj08.html>.
- [5] シード・プランニング. サービスロボットビジネスの市場動向. <http://www.seedplanning.co.jp/press/2011/2011102501.html>.
- [6] 芝星帆. 飛行型ロボットと携帯情報端末を利用した「浮遊ノート」システムの提案. 電気通信大学大学院 情報システム学研究科情報ネットワークシステム学専攻, 2013.
- [7] Wenhui Xiang, Yang Cao, and Zengfu Wang. *Automatic Take-Off and Landing of a Quad-rotor Flying Robot*, 2012.
- [8] Jean-Christophe Zufferey, Antoine Beyeler, and Dario Floreano. Autonomous flight at low altitude with vision-based collision avoidance and gps-based path following. In *IEEE International Conference on Robotics and Automation*, 2010.
- [9] Tianmiao Wang, Liewu Wang, Jianhong Liang, and Yang Chen. *Autonomous Control And Trajectory Tracking Of Quadrotor Helicopter*, 2012.
- [10] Jasmin Velagic, Nedim Osmic, Faris Hodzic, and Harun Siljak. Outdoor navigation of a mobile robot using gps and gprs communication system. In *IEEE International Symposium ELMAR*, 2011.
- [11] Cooper Bills, Joyce Chen, and Ashutosh Saxena. Autonomous mav flight in indoor environments using single image perspective cues. In *IEEE International Conference on Robotics and Automation Shanghai International Conference Center*, 2011.
- [12] MIT Senseable City Lab. *SKYCALL*. <http://senseable.mit.edu/skycall/>.
- [13] 樫原裕大, 吉永努, 清水裕基, 入江英嗣. スマートフォンによる歩行動作分析の評価. DI-COMO2012 シンポジウム, 2012.
- [14] 北陸電気工業株式会社. 3 軸加速度センサーアプリケーションノート. <http://www.hdk.co.jp/pdf/AP.Note/anhaam04.v1.01.pdf>.

- [15] Michael L. McGuire. An overview of gait analysis and step detection in mobile computing devices. In *2012 Fourth International Conference on Intelligent Networking and Collaborative Systems*, 2012.
- [16] Google Inc. *Google Maps Javascript API V3 Reference*. <https://developers.google.com/maps/documentation/javascript/reference?hl=ja>.
- [17] Parrot 社. *AR.Drone*. <http://ardrone2.parrot.com/>.
- [18] Apple 社. *iPhone*. <http://www.apple.com/jp/iphone/>.
- [19] Apache Friends. XAMPP とは ? <http://www.apachefriends.org/jp/index.html>.

発表論文

- [1] 小野澤清人, 芝 星帆, 吉永 努, 入江英嗣. タッチ指示によるお供ロボットナビゲーション. pp. 1897 - 1904, DICOMO 2013, July, 2013
- [2] 小野澤清人, 吉永 努, 入江英嗣. 浮遊ノートシステムにおける道案内機能の実装. pp. 205 - 206, WISS 2013, December, 2013