

公立はこだて未来大学 2022 年度 システム情報科学実習 グループ報告書

Future University Hakodate 2022 Systems Information Science Practice
Group Report

プロジェクト名

めざせ宇宙開発 - 自律移動ロボット飛行プロジェクト

Project Name

Flying Autonomous Robot Project

グループ名

グループ A

Group Name

Group A

プロジェクト番号/Project No.

16

プロジェクトリーダー/Project Leader

山口直人 Naoto Yamaguchi

グループリーダー/Group Leader

山口直人 Naoto Yamaguchi

グループメンバ/Group Member

今井瑞貴 Mizuki Imai
古賀海聖 Kaisei Koga
佐藤俊介 Shunsuke Sato
和山慧音 Keito Wayama
中澤一輝 Kazuki Nakazawa
寒河江改 Arata Sagae
長嶋康太 Kota Nagashima

指導教員

大澤英一 和田雅昭

Advisor

Eiichi Osawa Masaaki Wada

提出日

2023 年 1 月 18 日

Date of Submission

January 18, 2023

概要

CanSat とは缶サイズの模擬小型人工衛星のことを指す。本プロジェクトでは、CanSat の大会であるスペースプローブコンテストでの最優秀賞の獲得を目標とし、大会のレギュレーションに沿った CanSat の設計を検討してきた。「カムバックコンペティション」と呼ばれる方式を取る本コンテストだが、今回はその中でも「フライバック方式」と呼ばれる方式での出場を計画している。ここでは空中での機体の制御が開発に際する問題点として挙げられる。この問題に対する解決策を検討したうえでコンテストに出場し、その結果をもとに機体のアップグレードと屋外での実験を行い新たな課題を発見することを予定している。

キーワード CanSat, 模擬小型人工衛星, カムバックコンペティション

(※文責: 寒河江改)

Abstract

CanSat refers to a can-sized simulated small artificial satellite. In this project, we have been studying the design of CanSat in line with the regulations of the competition, with the goal of winning the highest award in the space probe contest, which is a CanSat competition. This contest takes a method called "comeback competition", but this time we are planning to participate in a method called "flyback method". Here, the control of the airframe in the air is mentioned as a problem in development. After considering the solution to this problem, we will participate in the contest, and based on the result, we plan to upgrade the aircraft and conduct outdoor experiments to discover new problems.

Keyword CanSat, small artificial satellite, comeback competition

(※文責: 寒河江改)

目次

第 1 章	はじめに	1
1.1	背景	1
1.2	CanSat について	1
1.3	従来 の 例	2
1.4	従来 の 問題 点	3
第 2 章	活動 内容	4
2.1	本 プロジェクト にお ける 目的	4
2.1.1	プロジェクト 学 習 で 行 う こ と の 利 点	4
2.2	プロジェクト の 目標	4
2.2.1	スペース プロブ コンテスト につ いて	5
2.2.2	サクセス クライテリ ア	7
2.3	課題 の 設定	8
2.4	課題 の 割 り 当 て	8
2.4.1	機 体 班	9
2.4.2	コ ー ド 班	9
2.5	スケジュー ル	9
2.5.1	前 期	9
2.5.2	後 期 の 予 定	10
第 3 章	課題 解決 の プロセス	13
3.1	機 体 班	13
3.1.1	パラ フォイル	13
3.1.2	スタビライ ザー	18
3.1.3	本 体	20
3.2	コ ー ド 班	24
3.2.1	アルゴリズム	24
3.2.2	電 装	25
3.2.3	メ イン プログラム	27
3.3	体 育 館 実 験 (1 回 目)	30
3.4	体 育 館 実 験 (2 回 目)	31
3.5	大 会 前 ダム 実 験	32
3.6	大 会 後 体 育 館 実 験	33
3.7	大 会 後 ダム 実 験	33
第 4 章	インターワーキング	35
4.1	プロジェクト 全 体 の インターワーキング	35
4.1.1	ナレッジ マネジメント	35

4.1.2	コード管理	36
第 5 章	成果	37
5.1	プロジェクトの成果	37
5.1.1	スペースプローブコンテスト前の成果	37
5.1.2	スペースプローブコンテストの結果	38
5.1.3	スペースプローブコンテスト後の成果	40
5.2	発表会	42
5.2.1	中間発表	42
5.2.2	成果発表	43
第 6 章	おわりに	47
6.1	プロジェクトにおける個人の役割	47
6.1.1	今井瑞貴	47
6.1.2	古賀海聖	47
6.1.3	佐藤俊介	48
6.1.4	和山慧音	48
6.1.5	山口直人	48
6.1.6	中澤一輝	49
6.1.7	寒河江改	49
6.1.8	長嶋康太	49
6.2	今後の課題と展望	50
6.2.1	機体	50
6.2.2	コード	50
6.2.3	大会	52
付録 A	プログラムリスト	53
参考文献		61

第 1 章 はじめに

この章では、本プロジェクトの背景、従来の問題点、課題について述べる。

(※文責: 和山慧音)

1.1 背景

現在、宇宙利用の形態は多岐に渡っており、宇宙そのものについて調べる宇宙探査だけでなく人工衛星による地球観測や測位事業など地上に対しての利用もされており、我々が普段利用している GPS や気象予報など、ごく身近なモノも宇宙開発の影響を受けている。

しかし、宇宙開発そのものは身近には感じられることは少ない。これは宇宙開発が巨大化、複雑化、高コスト化、高信頼化し、簡単に試作や実験を行うことができない為である。その為、現状で学生が本格的な宇宙工学を学ぼうと考えても困難である場合が多い。学生の宇宙工学において重要な人工衛星についての学びを深める目的として始まったのが CanSat プロジェクトである。

過去にカムバックコンペティションを行い、毎年 9 月上旬に米国ネバダ州ブラックロック砂漠で行われる、大学生を主とした CanSat の打ち上げ競技会である ARLISS にて Best Mission Award 2nd Place を受賞した CanSat の例を ARLISS2019 東京工業大学のホームページ [2] から紹介する。ARLISS2019 では各チームが規定のサイズに収めた CanSat をアマチュアロケットに搭載し、高度 4,000 メートルに打ち上げる。ARLISS には 350ml クラスとオープンクラスの 2 つのサイズ規定がある。350ml クラスは重量 350g 以下、高さ 240mm 以下、直径 66mm 以下であり、オープンクラスは重量 1050g 以下、高さ 240mm 以下、直径 146mm 以下である。東京工業大学のチーム Rosetta はオープンクラスにて大会に参加した。機構としては、アマチュアロケットから CanSat が放出後、パラシュートでの降下時にカメラで方向を観測し、パドルを動かすことによって進行方向を決定する機構であった。

(※文責: 和山慧音)

1.2 CanSat について

CanSat[1] とは一般的な飲料水の缶サイズの模擬人工衛星を指し、実際の衛星と類似の開発プロセスで制作される。CanSat プロジェクトはこの CanSat を用いて、ミッションの創出、衛星の設計・製作・試験・打ち上げ・運用等の過程を通して、宇宙工学における学生の教育を行おうと開始された。起源は、1998 年ハワイにて JUSTSAP 会議と連携して開催された、University Space Systems Symposium (USSS) においてスタンフォード大学の Robert Twiggs 教授によって提唱されたことに由来する [1]。以降アメリカや日本をはじめとして、各国で学生が競技会に参加するなど盛んにプロジェクトが行われている。

CanSat プロジェクトにおける競技会は CanSat を上空から投下し、降下中または着地後にミッションを行う形式である。競技会には、大きく分類してカムバックコンペティションとミッションコンペティションの 2 種目が存在する。

カムバックコンペティションとは、地上制御と飛行制御のどちらか、またはその両方を用いて地上に設けられた目標地点を目指す種目である。

ミッションコンペティションとは、自らミッションを設定しそのミッションの達成度とそのアイデア性を競う種目である。

本プロジェクトではカムバックコンペティションを採用した競技会に参加することを目的として CanSat の製作を行う。

(※文責: 佐藤俊介)

1.3 従来例

過去にカムバックコンペティションを行い、毎年9月上旬に米国ネバダ州ブラックロック砂漠で行われる、大学生を主とした CanSat の打ち上げ競技会である ARLISS にて Best Mission Award 2nd Place を受賞した CanSat の例を ARLISS2019 東京工業大学のホームページ [2] から紹介する。ARLISS2019 では各チームが規定のサイズに収めた CanSat をアマチュアロケットに搭載し、高度 4,000 メートルに打ち上げる。ARLISS には 350ml クラスとオープンクラスの 2 つのサイズ規定がある。

350ml クラスは、

- 重量 350kg 以下
- 高さ 240mm 以下
- 直径 66mm 以下

であり、オープンクラスは、

- 重量 1050g 以下
- 高さ 240mm 以下
- 直径 146mm 以下

である。東京工業大学のチーム Rosetta はオープンクラスにて大会に参加した。機構としては、アマチュアロケットから CanSat が放出後、パラシュートでの降下時にカメラで方向を観測し、パドルを動かすことによって進行方向を決定する機構であった。

さらに、本プロジェクトで昨年度作製した機体を紹介する。

昨年度の A グループでは、飛行を主な目的とするパラフォイル、パラフォイルの展開補助を目的とするパラシュートの二種類を採用した。降下時、加速度センサを用いて落下検知を行い、検知に成功したのちに GPS を用いて自身の現在位置を取得し、パラフォイルの紐を引くことによって進行方向を決定し目標地点を目指して滑空するというものであった。

昨年度の B グループでは、壊れやすいものを壊れないように運ぶことをグループのミッションに設定した。また、上述した ARLISS の大会規定のオープンクラスに沿って機体を作成した。一般的な CanSat では、降下の際の減速機構としてパラフォイルを採用する。しかし、B グループでは、降下時の縦方向の衝撃を緩和できることや降下中の芸術性から、グライダーを用いた。螺旋状に落下したのち着地を判定し走行を開始する、GPS から位置情報を取得し、目標地点を認識しそ

こへ向かうように制御を行うというものであった。

(※文責: 佐藤俊介)

1.4 従来の問題点

従来、カムバックコンペティションを行う CanSat における問題点の一部として以下があげられる。

姿勢制御

カムバックコンペティションでは、降下中の制御にパラシュートを用いる機体が多い。このような機体ではパラシュートを展開する際に機体に紐が絡まってしまい、制御が難航という問題点がある。また、風などの影響を受けて機体の姿勢が変わってしまい、制御に支障が出るという事例も報告されている [3]。

滞空性

機体の重量や減速機構などの設計ミスにより、パラシュートやパラフォイルによって落下中の速度を減速しきれずに十分な滞空を行えない場合がある。このような事態が発生すると滞空時間が短くなってしまい制御不能となる場合がある。

センサの誤作動

降下時または地上走行時において、センサの誤作動によって想定外の動作をしてしまう場合がある。このように想定外の動作をしてしまうと、目標地点への到達が困難になる。

機体の耐久性

パラシュートや制御翼などの減速機構で減速を行っても、着地時の衝撃で機体及び内部モジュールが破損してしまうことがある。この場合ミッション中のログが取得できなくなる。

サイズ規定

CanSat 競技会は CanSat の重量、高さ、直径などレギュレーションが規定されている。よって、搭載モジュールや電気回路、機体、パラシュートの製作に必要な最低限な材料に加えて、軽材料を選別し、設計する必要がある。

このように、カムバックコンペティション参加における CanSat の作成には多くの問題点があり、段階実験を通して検証、改善していく必要がある。

(※文責: 今井瑞貴)

第 2 章 活動内容

2.1 本プロジェクトにおける目的

本プロジェクトの目的は、CanSat の設計・構築・運用を通して宇宙工学に関連した回路設計制作技術、飛行制御技術、無線通信技術、プロジェクト運用法を学習することである。また、飛行制御で姿勢や向きを調整し、目的地まで破損せずに到達する機体の作成することである。

(※文責: 今井瑞貴)

2.1.1 プロジェクト学習で行うことの利点

本学の科目「システム情報科学実習」のシラバスを参照し、到達目標から以下の 3 点において本プロジェクトの利点を示す。

- プロジェクトを自主的に管理・運営する方法を学習する。
本プロジェクトでは 1 つのグループでコード班、機体班に分かれて活動しているため、進捗管理や情報共有を怠ると致命的なものになりかねない。そのため、2 つの班のコミュニケーションが重要なものになる。また、9 月に大会に出場するため、大会から逆算し、不測の事態にも備えたタスク管理能力を身に着けることができる。
- 通常の講義とは異なる多様な教育機会を、履修者に提供する。
CanSat では、大会のレギュレーションに沿っていれば機体、プログラムを自由に作ることができる。これによってミッション達成のために必要な技術、知識を自ら取捨選択し学ぶことが可能となる。
- 成果を内外に公表し、大学および地域社会に貢献する。
本グループでは大会に参加することで、外部へ公立はこだて未来大学の名前を広げることができる。また、学生が CanSat の大会に参加するということ自体が宇宙開発分野の盛り上げにつながる。

(※文責: 今井瑞貴)

2.2 プロジェクトの目標

本グループでは、CanSat を用いた大会への出場および、大会において最優秀賞の獲得を目標とした。

(※文責: 長嶋康太)

2.2.1 スペースプローブコンテストについて

スペースプローブコンテストとは、植松電機株式会社が主催している CanSat 競技会であり、2022 年 9 月 17 日に株式会社植松電機構内の現地開催とリモートのハイブリッドで開催される [4]。参加者はチームを組んで開発を行い、スペースプローブコンテストで制定されるミッションに対して取り組むこととなる。

本グループはチーム名「FUNSat」として、リモート参加で出場した。そして、フライバックコンペティションに取り組んだ。これは、上空から CanSat を投下し目標地点を目指し空中制御を行い、着地点と目標地点の距離の近さや芸術性で競う競技である。また、今年はコンテスト史上最多応募の総勢 9 チームが出場した。そして、本グループ以外のチームは、CanSat をロケットへ搭載し打ち上げて投下を行っていた。以下、本グループを加えたチームの所属、チーム名と出身地になる。

- 室蘭工業大学「SARD (サード)」北海道
- 社会人チーム「Cosmo Craft (コスモクラフト)」東京都
- 立命館慶祥高等学校「Fantome(ファントム)」北海道
- 「チーム岩井家」北海道
- はこだて未来大学「FUNSat(ファンサット)」北海道
- ものつくり大学「月雫 (つきしずく)」埼玉県
- 社会人チーム「リーマンサット」東京都
- 北海道情報大学「HIU 宇宙研」北海道
- 北見工業大学「Nociws(ノチウス)」北海道

また、これらのチームにはカムバックコンペティションに取り組むチームも確認できた。

スペースプローブコンテストの流れ

スペースプローブコンテスト当日、作製した CanSat について、事前に作成したプレゼン動画を用いて発表を行う。その後、下図 2.1 のミッション図の流れに沿って、ロケットまたはドローンを用いて高度 100m からの投下審査を行う。最後に、スペースプローブコンテスト後に作成するプレゼン動画と投下審査の際の動画やセンサのデータ解析などをもとに結果について発表を行う。

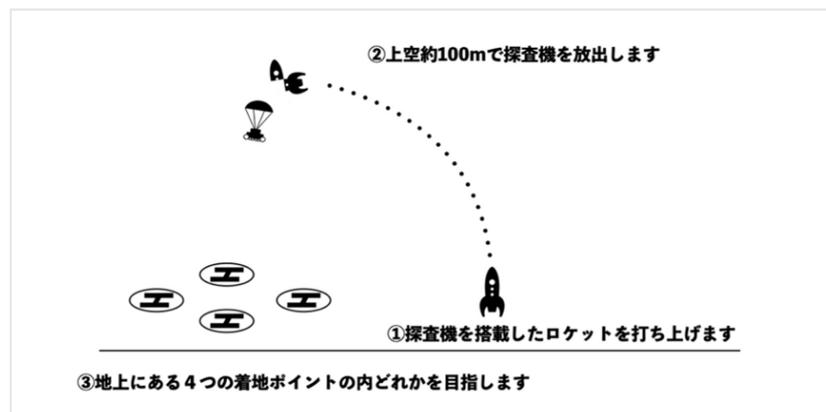


図 2.1 ミッション図

機体レギュレーション

以下スペースプローブコンテストの機体レギュレーションから引用する

プローブのサイズ

プローブはキャリア、本体（バッテリー等含む）、パラシュートなど回収機構を含めて下記の規定サイズに収まるように製作すること。

- 外径 ϕ 150 以内（突起物含む）
- 全長 340mm 以内（突起物含む）
- 重量 1000g 以内（全備重量）

プローブの落下速度

投下実験機からの放出後、競技会場外への流出防止およびプローブの安全な回収の為、下記項目を満足していること。

- プローブ本体にパラシュート等の回収機構（減速機構）を取り付けること
- 放出後の降下速度は 6 m/s 以上とすること

実験条件

以下スペースプローブコンテストの機体レギュレーションから引用する。

- 投下実験機にて高度 100 m まで上昇後、フェアリング部を開放しプローブを放出する（自由落下）放出時のプローブ姿勢は、地面に対し平行（水平）となる。
- 従来の Gas-CAMUI を使用した打上げより機体にかかる負荷は少ないが放出高度に到達するまで時間がかかる（離陸から放出までおよそ 60 秒を予定）
- 放出時の風速は 7 m/s 以下とし、風速などの状況により上空で待機時間が発生する可能性もある
- 投下実験機フェアリング部に、フライトピン用の結束などをすることは可能（ただし事前の申請と打ち合わせが必要）
- ターゲットは投下実験機の真下から半径 10m の距離に 90° 位相（東西南北）に 4 つの固定ターゲットを設置する。（下図参照）移動ターゲットは設定しない。
- ターゲットへのビーコン等の設置は可能（ただし事前の申請と打ち合わせが必要）

また、下図 2.2 の実験条件図を以下に示す。

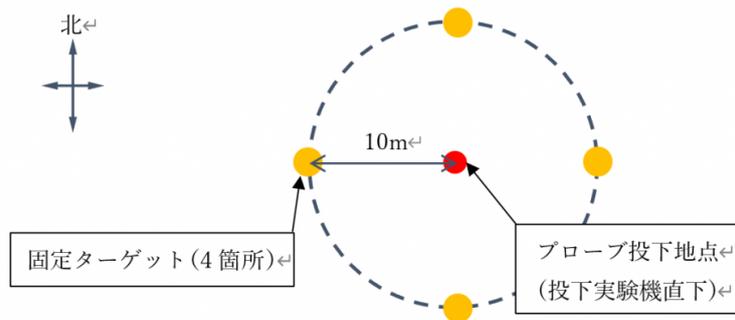


図 2.2 実験条件図

(※文責: 中澤一輝)

2.2.2 サクセスクライテリア

目標を達成するための到達レベルとしてサクセスクライテリアを設定した。レベルの低い順にミニマムサクセス、ミドルサクセス、フルサクセス、アドバンスサクセスとなっている。

ミニマムサクセス (60%)

前提目標を遂行・達成するうえで必ずクリアしなくてはならない基礎的な目標最低限 CanSat の制御がなされていることが確認できる。

ミドルサクセス (80%)

前提目標を達成できる状態での最低目標 CanSat を制御して目標地点の半径 10 m以内に着地することができる。

フルサクセス (100%)

前提目標を達成できる状態での本来の目標 CanSat を制御して目標地点の半径 5m 以内に着地できる。

アドバンスサクセス (120%)

前提目標が達成可能である状態でのより高度な達成目標スペースプローブコンテストの評価基準において着地精度以外でも高評価を獲得できる。

(※文責: 中澤一輝)

2.3 課題の設定

ここでは、前項で挙げられた目標を達成するために解決すべき課題を記述する。

1. 落下の検知

動作のタイミング CanSat が投下され、落下の検知後に制御を開始させるため、落下したかどうかの判定をする必要がある。

2. アルゴリズムの作成

風による影響風による機体の急激な軌道変化などの外的要因も加味したアルゴリズムの作成が必要である。

3. 機体の作成

昨年度の大会ではパラフォイルが開かずに落下したため、パラフォイルが開きやすい仕組みを取り入れた機体の作成が必要がある。

4. 機体の強度

着地時の衝撃当機体はマイコンやその他電子モジュールによって制御され、SD カードなどの繊細な部品も搭載するため、着地時の衝撃を最小限に押さえ、部品が破損しないようにする必要がある。

5. マイクロコンピュータによる制御

GPS や加速度センサによる値から進行方向を決め、マイクロコンピュータでサーボモータを制御し目標地点まで飛行させる。値の取得方法やサーボの制御方法などを間違えると目標地点へたどり着けないため、各モジュールの使用方法や実装方法を理解する必要がある。

6. データの保存

動作ログ実験やスペースプローブコンテストの結果を定量的に分析するため、各モジュールの数値的な履歴を残しておくことが必要である。その方法の1つとしてSD カードへの書き込みがあげられるが、設定や書き込み方法を理解する必要がある。

(※文責: 長嶋康太)

2.4 課題の割り当て

各人の得意分野および要望をもとに以下の2班に分かれ、前項であげられた課題を役割ごとに振り分けた。

(※文責: 山口直人)

2.4.1 機体班

機体班では CanSat 競技における機体構造の検討、製作を行う。

本グループでは昨年の A グループ同様減速及び移動する機構としてパラフォイルを採用し、サーボモータにより紐を引くことで制御することとした。昨年度の A グループの CanSat 競技の結果から、パラフォイルが空中で展開するための機構を大幅に改善する必要があることが分かった。

競技会の規定に沿ったスタビライザーの取り付け位置と機体全体の設計の研究。また、空中で目標地点まで近づくため、十分な機動性を確保する必要がある。課題としては、材料や翼長、断面の検討、等の研究があげられる。競技会の規定により 6m/s 以上の降下速度を維持する必要があるため、着地時の衝撃により基盤やマイコンが壊れないような機構の研究を行う必要がある。

(※文責: 佐藤俊介)

2.4.2 コード班

コード班では CanSat 競技における目標地点へ向かう機体の制御やその実装を担当する。主にアルゴリズム・プログラム・電装という 3 種類の役割をもち、相互に連携しながら開発を進める。

アルゴリズム

GPS や各種センサの値から目標地点へ向かう制御をおこなうためのアルゴリズムを検討する。また、風などの外的要因による機体への影響も考慮したアルゴリズムの構築も課題である。

電装

使用するモジュールの選定や基盤の設計、作製を行う。他のメンバーと連携を取り、重量などを考慮して動作に必要なモジュールを決め、正常に動作するような基盤の設計と作成が課題である。

プログラム

各種モジュールの制御やアルゴリズムをマイコンに組み込むためのプログラムを作成する。使用する部品の仕様を理解し、意図した動作をさせることが課題である。

(※文責: 中澤一輝)

2.5 スケジュール

2.5.1 前期

5月

グループ内の役職を決め、今年度出場する大会をどれにするかの検討を行った。また機体の形状や制御機構を決定した。

6月上旬

昨年度の機体を引き渡してもらいその構造を理解した。またプログラミングの学習を開始した。パラフォイルの設計図を生地に写し取り、実際に縫った。MDF でスタビライザーの試作品を作り、金具で固定した。

6月中旬

昨年度の機体制御で用いたソースコードの読み込みをした。ラズパイのインストールと初期設定をした。機体本体の設計図を制作し、レーザーカッターで制作をおこなった。試作パラフォイルを作った結果出てきた課題、改善点を探った。

6月下旬

昨年度のソースコードをラズパイ上で実際に動作させ、その挙動を確かめた。GPS 周りのコードの検討を行った。体育館実験を行ってデータを取り、パラフォイルの改善案を考えた。

7月上旬

ポスター作成を行い、中間発表に向けて準備した。

7月中旬

中間報告書の執筆項目を各メンバに割り当て、Notion 上で作成した。またスペースプローブコンテストに関わるメディア承諾書・機体三面図・設計仕様書などの書類を作成し、提出した。

7月下旬

期末提出物をすべて完成させ、提出した。

2.5.2 後期の予定

8月上旬

全体で機体の完成までの方針が決まったため、必要な材料の発注を行った。またコード班ではプログラムの作成のために、GPS センサーや 9 軸センサーの動作チェックも行った。

8月中旬

機体班はドローン投下での機体の実験を想定していたため、そのための機体の放出装置を作成していた。コード班では GPS センサーの動作が屋外での検証が必要であったため、屋外に出て動作実験をした。

8月下旬

機体班では実験を通じて得た改善点を解決した。またコードでは電装をつける場所の確認と配線を確認し、電装同士のはんだ付け作業を進めた。。

9月上旬

体育館からの落下実験では十分な高度が確保できなかったため、大会出場前に大学近くの笹流ダムの上から機体を降下させる実験を行った。

9月中旬

スペースプローブコンテストの主催者である植松電機へ機体の発送を行った。またオンラインでスペースプローブコンテストに出場した。

9月下旬

後期のプロジェクトの活動計画を立てた。また大会から帰ってきた機体を受け取り、その状況について確認した。

10月上旬

市立函館高校の学生に向けたプロジェクトの発表があったため、その発表資料を作成した。また実際に高校生に向け発表した。

10月中旬

後期のプロジェクトの活動計画について、細部の調整を行った。また大会の結果や大会から帰ってきた機体から改善点を導き出した。

10月下旬

導き出した改善点から、具体的にどういった改良を施していけばいいのかについて話し合いをし、決定した。

11月上旬

機体班では機体の紐の引っ掛かりなくす機構の作成を行った。またコード班ではプログラムの自動実行がうまくいかない原因について実験を行い、調査した。

11月中旬

改良を加えた機体の性能について検証するためにダムでの実験を計画した。またダムで実験を実際に行い、性能について確認をしたが、データの取得が不十分な個所があったため、再度のダムでの実験を計画した。

11月下旬

再度のダムでの実験を行い、必要であったデータの取得に成功した。また成果発表会に向けてスライドの作成を開始した。

12月上旬

成果発表会に向けてスライドとポスターの作成を進め、実際に発表を行った。

12月中旬

グループ報告書の執筆を開始し、担当個所を決めて行動した。またそれを tex ファイルにまとめた。

12月下旬

個々人の後期末提出物について作成を行った。

1月

期末提出物を提出した。またこれまで使ったプロジェクトの備品やプロジェクトで作成した成果物を次年度に引き継ぎやすいようにまとめた。

(※文責: 山口直人)

第 3 章 課題解決のプロセス

この章では、2.4 で割り振られた課題に対して行った作業について述べる。

(※文責: 寒河江改)

3.1 機体班

機体班では主に落下の際に使用する減速機構と各種モジュールを収める機体の制作を行っている。減速機構として、飛行を目的とするパラフォイルとパラフォイルの展開補助を目的とするスタビライザーの採用をした。それぞれの採用理由、制作方法は後述する。また、機体本体に関しても後述する。

(※文責: 和山慧音)

3.1.1 パラフォイル

本グループでは、機体制御を行う為の機構の一つにパラフォイルを採用した。

パラフォイルとはパラシュートに似たような飛行構造を持つ翼であり、おおよそ正方形または円のような形となるパラシュートと異なり、横幅が広い楕円形のような形になっている。構造的にはパラグライダーに近いものになっており、パラグライダーが両手に持って紐を引っ張ることにより進行方向を制御を行うように、制御紐の長さを調整することで機体の進行方向の調整が行えるが、紐を引いて意図的に機体バランスを崩す関係上、パラシュートよりも制御が困難である。

パラフォイルの採用理由は機体の進行方向の制御が行えるからである。今回参加したスペースブローブコンテストではただ着地するだけでなく、落下地点から目標地点までの移動が求められる。パラシュートでは安定した着地を行えるが機体の進行方向の制御が行えない為、目標地点への移動が困難であることが採用理由の一つである。また、昨年度の A グループが同様の大会にてパラフォイルによる機体の進行方向の制御を行っており、そのときのノウハウが残っており利用できる状態であることも採用理由の一つである。これら 2 点の理由から、本グループではパラシュートではなくパラフォイルを採用した。

パラフォイルは既存のパラフォイルを参考にそのパラフォイルの 1/2 縮小モデルを製作した。参考にしたパラフォイルはスタントパワーカイトインフレーターダブル耐久性サーフィンパラフォイルレインボーパラシュート 2m[5] である。実物を元に採寸を行い、以下のようにパーツの設計図を作り、試作品の製作を行った。

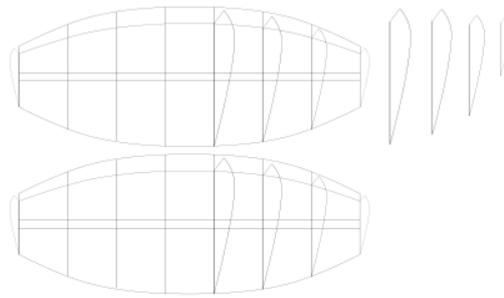


図 3.1 パラフォイルの設計図



図 3.2 パラフォイルの試作品

基本的な構造を説明する。パラフォイルは主に以下の2つに分かれる。

- 空気を受けて減速し移動する本体の翼部分
- 布部分を特定の位置から引くことによって制御を行う制御紐

翼部分の素材には昨年度と同様にリップストップポリエステル生地を使用し、耐久性の向上と軽量化を行った。製作方法としては生地をパーツの設計図に基づいて切り出し、縫うポイントをマジックペンで引いた後にその引いた部分に沿うように並縫いにより機体の上部分と内側部分の空気を取り入れる幅部分をおおよそ 0.5cm 幅で縫い合わせた。その後、並縫いで内側部分の幅部分と機体の下側部分をマジックで引いた部分に沿うように縫い合わせることで作成した。

翼に取り付けた制御紐は試作品では昨年度と同様に凧糸を使用し、それぞれ左右に前方に3本、中間に3本、後方に2本に設置した。この制御紐は前方3本と中間3本で機体の安定性をはかり、後方2本を左右でサーボモータで引くことによって機体の進行方向を制御することを想定している。制御紐は翼の方に1cmほどの中折りした布を縫い付け、その部分に制御紐を固く結ぶことで取り付けている。制御紐の長さに関しては次の小段落の制御ひもの長さにて記述する。

パラフォイルは試作品を制作した後、体育館での実験を行った。この実験は3.3にて記述する。

この実験後、機体の改良点を挙げ、その後、機体の本製作を行った。本製作した機体は既存のパラフォイルの約3/4の縮小サイズで製作を行った。試作品との違いとして、サイズの他に以下の3点が挙げられる。

- 縫い方
- 制御紐の変更
- 端部分での型の変更

Flying Autonomous Robot Project

縫い方に関しては下から風を受ける布部分での強度を上げる為、パラフォイルの下側の外側部分を端まつり縫いで縫う回数を増やすことで強度向上を行った。この際、上側部分と内側部分は縫いすぎると形が固まってしまい柔軟性がなくなる為、パラフォイルの空気を取り込む上側部分と空気を取り込んだ内側部分は試作品と同様の回数で並縫いを行った。



図 3.3 端まつり縫い



図 3.4 並縫い

制御紐では超高分子量ポリエチレンを使用した。超高分子量ポリエチレンはパラグライダーの制御紐などに使用されており、高い柔軟性と強度を持つ。この糸を採用することにより、紐の絡まりが麻糸を使用した場合よりも改善されており、安定性の向上も見られた。またパラfoil展開時の形の安定の為、制御紐の本数をそれぞれ左右に前方に4本、中間部分に4本、後方部分に3本に増やし、前方の2つ、後方の2つの計4つを以下のような器具をレーザーカッターで制作し、制御紐を纏めた。この留具により、パラfoilの制御紐を独立して調整を行いやすくなった。また、留具によってパラfoil展開時の形をより一定に保てるようになった。



図 3.5 留め具

前方2つの制御紐の束はそれぞれ左右の前方の紐の4本と中間の紐の3本を纏め、機体の姿勢を安定させる為に紐の長さも固めた。後方2つの制御紐の束は後方3本を纏め、サーボモーターで引くことで紐の長さを変化させるようにした。そうして纏めた制御紐の束はスタビライザーを経由して機体本体と接続し、前方2つの制御紐の束で機体の姿勢の安定を行い、後方2つの制御紐の束で機体の方向の制御を行った。

端部分での型の変更については、実験から判明した端部分からパラfoilが潰れるという事態を防ぐため、端部分を上下部分で空気を取り込む幅の布を作成して縫い、空気を取り込み口を大きくし、その代わりに三角形の布を端につなげた。この三角形の布は制御紐に引っ張られた際に空気を取り込み口が閉まるあるいは縮小し、パラfoil全体が潰れるという事態を防ぐ為に端に設けた部分である。これにより、空気を取り入れ部分を増えたことでパラfoilの安定性が上がった。その三角形の布部分は以下のようなものである。



図 3.6 端の布部分

そうして、完成したパラfoilが以下のものである。



図 3.7 パラフォイルの本制作品（表）

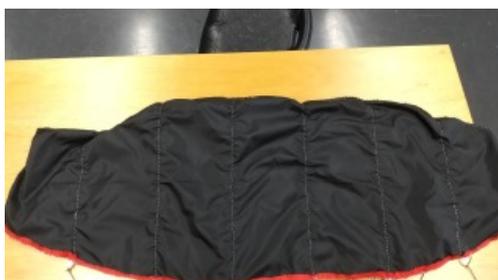


図 3.8 パラフォイルの本制作品（裏）

このパラフォイル製作後、再び体育館実験及びダム実験が行われた。この実験は 3.4,3.5,3.6 にて記述する。

また、大会後の体育館実験にてパラフォイル自体に想定していたパラフォイルと比べ微細なサイズの違いが生まれていることが分かった。この原因として、縫った人物による個人差の他に使用した生地の色が黒や茶色、赤色といった色である為、縫う線を表すチャコペンの線の確認が難しく、ズレが発生しやすかったことが挙げられる。改善点としてチャコペンが目視しやすい色の生地を採用することや機器を用いた個人差の出にくい縫い方などが挙げられる。

（※文責: 和山慧音）

制御紐の長さ

パラフォイルの制御紐の長さはパラフォイルの展開時の形を決定する部分であり、パラフォイルの形が少し崩れると機体制御が行えずに機体が墜落する為、制御紐の長さはパラフォイルが安定した飛行が行えるかが決まる部分である。その為、複数回の実験が行われており、その際の長さについて述べる。

試作品による体育館実験では、パラフォイルが減速機構として機能するのを確認する実験であった為、左右で前方、中間、後方の全ての紐をそれぞれで束ねて展開した状態での実験となり、またスタビライザーを使用せずに材木及び本体に直接繋げる形での実験となった。この際のパラフォイルの制御紐の長さは束ねた部分までは既存のパラフォイルの制御紐の長さの $1/2$ の長さで輪っかを作るように製作し、その輪っか部分を左右で束ねて新たな紐で纏めて固結びにし、その固めた部分から 10cm となる地点で材木及び本体と結べる長さにして実験が行われた。

実験結果に関しては 3.2 に記述するが、輪っかを作るようにする束ね方では左右の長さの違いが分かりづらく、輪っかのサイズも飛行時に変動することが考えられた。

本製作したパラフォイルでは、輪っかを作る束ね方ではなく、留具を用いた束ね方へと変更した為、ここでの細かい長さの調整がその場で行いやすくなり、輪っかのサイズの変動の問題も解消さ

れた。パラフォイルの翼部分から留具までの長さは、既存のパラフォイルの 3/4 の長さで製作し、その後スタビライザーの長さが確定したことに伴い、スタビライザーの端から左右の制御紐によってパラフォイルが展開した際にわずかに弧を描くように長さの調整を行った。留具からスタビライザーへの固定を行うまでの紐では、前方は 30cm ほどの長さで機体の安定を行う為、留具とスタビライザーの端部分までの長さを出来るだけ短くし、スタビライザーの空いている真ん中部分に巻きつけるようにした後に、スタビライザーの展開を阻害しない位置で固結びによって固定した。後方では、50cm ほどの長さを取り、留具とスタビライザーの端部分までの長さを前方に比べて 10cm ほど取り、スタビライザーの空いている真ん中部分に通した後に、機体本体の下側にあるサーボモーターへの接続を行った。

その後の大会前での体育館実験にて、翼部分から留具まで各制御紐の長さの調整及び後方の留具からスタビライザー間の紐の長さの調整が行われ、これにより進行方向へ正しく進むことの出来る制御紐の長さを確定した。

大会後、スタビライザー及び本体に変更が生まれた為、スタビライザーと接続する後方の留具からスタビライザー間の長さの調整を体育館実験で行った。その際、パラフォイル本体の微細なサイズの違いや機体の重量バランスの偏りなどから右を左から 2cm ほど長くした場合に正面方向へと飛ぶことが分かった。

大会後のダム実験にて、その調整によって外環境であっても本製作したパラフォイルが安定した飛行が行える調整であることが確認出来た。

制御紐の長さを決める際の問題点として、位置を固定する際は人の手による結びである為、細かな長さの変化が生まれやすいこと、機体バランスによって制御紐の長さの変更が発生しやすいことが挙げられる。そのため、結ぶ以外の制御紐の固定法を作成することや僅かな機体バランスの変化に影響されにくい制御紐の配置が求められる。

(※文責: 和山慧音)

3.1.2 スタビライザー

スタビライザーとはパラフォイルの展開を補助する為の機構であり、パラフォイルの紐が絡むことを防ぎ、かつ落下時にパラフォイルが開きやすくなる役割である。

昨年度の宇宙開発プロジェクトが参加した大会ではパラフォイルとパラシュートを採用したものの、大会ではパラフォイルの展開が出来ずに落下した。その原因としてセンサーが上手く反応しなかったこと、糸が絡んだことが挙げられた。その為、センサーに依ることなく糸が絡まずパラフォイルの展開を補助する機構が求められた。その為にスタビライザーを採用した。

スタビライザーは岐阜県立恵那高等学校の平成 22 年度の CanSat 開発を参考に設計を行った [6]。スタビライザーの素材には MDF 板を使用した。MDF 板を使用した理由として、市販しているため材料に困らないこと、レーザーカッターで加工がしやすいこと、MDF 板は木の細胞の大きさにまで解体、再び接着剤で固めたものであるため木の芯や木目を考慮する必要がないことが挙げられる。下記のようにパーツごとの図面を制作し、レーザーカッターを用いて切り出し、組み立てて制作した。なお、学生のレーザーカッターの使用には事前の講習と許可証が必要になるため事前にレーザーカッターの使用に関する講習を受けて制作した。

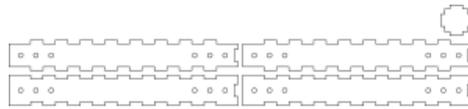


図 3.9 スタビライザーの設計図

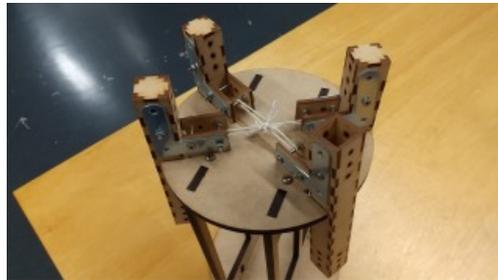


図 3.10 スタビライザーの試作品

上記のような試作後、機体重量やその幅、また制御紐を通す際の問題から大幅な改良を求められ、下図のような形へと変更した。下図のような形では、スタビライザー本体の重量の軽減を行うと共に、制御紐の取り付けが試作段階のものとは比べて容易となった。

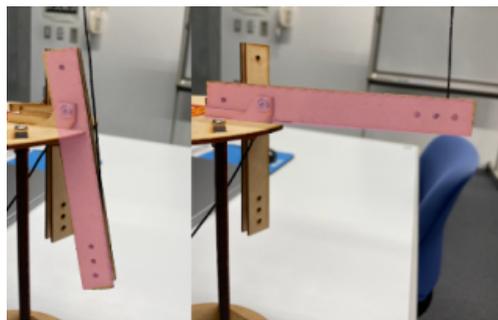


図 3.11 スタビライザー本制作品

また、試作品で用いたパラフォイルは体積が大きいのが、軽量化の為に中身としていた。これにより、スタビライザーが機体の範囲を大きくとってしまうのに加え、スタビライザーの柔軟性が無い為頑丈ではあっても衝撃で崩壊する可能性があった。その後変更したスタビライザーの方が体積を小さいが、より頑丈化する事に成功した。これは、左右方向に対する柔軟性が付いたからだと考えられる。このスタビライザーを用いた結果、大会ではパラフォイルの紐がスタビライザーの固定用ネジに引っかかってしまい、パラフォイルが十分な展開を成さなかった。紐の絡まりやすさという課題を発見した後、大会後、スタビライザーへの制御紐の取り付け部分の固定やスタビライザーの機体への付け根部分に紐が引っかかるなどの問題が判明した為、その改良を行った。

実際に改良した最終的なスタビライザーは下記の写真である。

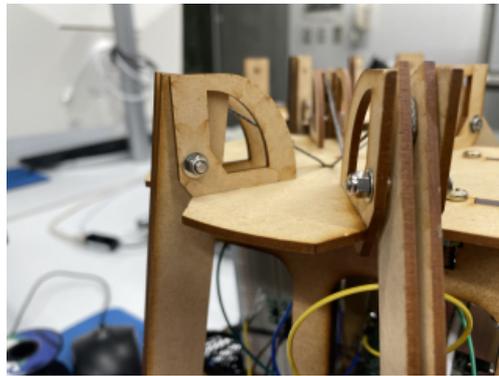


図 3.12 スタビライザー改良後

大会ではスタビライザー固定ねじが大きく出すぎてしまい、紐を巻き込んでしまった結果から、露出した邪魔なねじを削った。また、紐がスタビライザーに引っかからないように MDF 板を用いてカバーを取り付けた。完成した機体で実験を行った結果、紐が絡まずに安定した飛行が可能になった。

(※文責: 今井瑞貴)

3.1.3 本体

機体の本体は、機体の飛行制御を行うための電装の機器類やパラフォイルの制御紐を引くサーボモータ、バッテリー等を搭載する。これらを保護し固定する役割を担うため、可能な限り頑丈で軽量の設計が求められた。ここでは本体の製作過程について説明する。

機体の概形は大会に出ることを考慮し、円柱の筒のような形状に決定した。主な骨格は上下の円盤と柱となる二枚の板から成っている (図 3.1.3)。



図 3.13 機体本体の試作品

機体の素材はスタビライザーと同じく MDF 板を使用した。3D プリンタは骨格となる大きい部品が作れないこと、分割して作ると時間がかかることから使用を見送った。また、図面は illustrator で制作し、工場のレーザーカッターを用いて切り出し、組み立てた。詳細については小段落の工房にて述べる。

機体は試作を含め、最終的に 6 号機まで製作した。夏休み以前の 4 号機まではセンサー類の種類や配置、コードの配線が決定していなかったため、直径は大会規定の最大サイズで製作を行った。

本体の特徴と改良点を以下のようにまとめる。単位は mm(ミリメートル)、g(グラム)である。

1. 1号機 (直径 150、高さ 220)

- 機体の概形の決定
- 昨年の機体を参考に円盤と柱は差し込みとネジを使って垂直に固定
- 円盤にスタビライザーを格納する空間を用意
- スタビライザーを固定するための”コ型”の木組みを製作

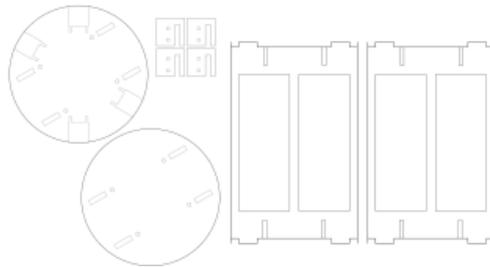


図 3.14 機体 1 号機の図面

2. 2号機 (直径 150、高さ 220)

- 強度を上げるため直角だった角を円く加工
- 固定用の木組みとhの接触部分に穴を開けた
- 固定用ねじ穴を追加

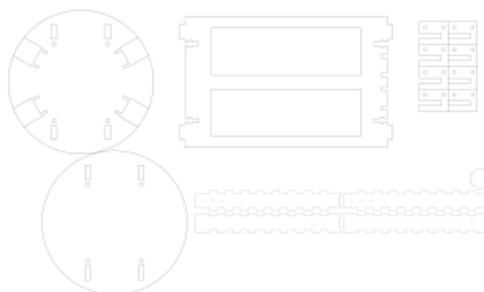


図 3.15 機体 2 号機の図面

3. 3号機 (直径 150、高さ 220)

軽量化するためスタビライザーと本体の金具での固定をやめ、MDF だけで固定できるように再設計を行った。

- スタビライザー固定用木組みを大きく変更。それに伴い円盤の穴を小さく修正。
- 強度を上げるため直角だった角を円く加工

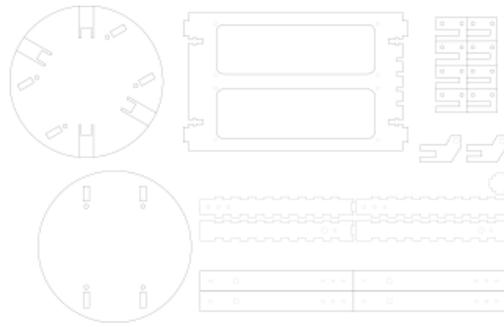


図 3.16 機体 3号機の図面

4. 4号機 (直径 150、高さ 220、重さ 357)

試作 1号機から 3号機までの間に軽微な改良を加えてはいたが、強度が足りず落下の衝撃で柱の根本から折れてしまうことが実験で多くみられた (図 2.5)。これは柱の向きが進行方向のみであることが原因であったため、以下の改良を行った。

- 軽量化のため上部円盤の厚みを変更 (5 → 2.5)
- 横からの衝撃に耐えられるように横板を追加
- 強度を上げるため固定用木組みの縦の厚みを増加
- スタビライザーに糸を通す隙間を開けるためにスペーサーを設計

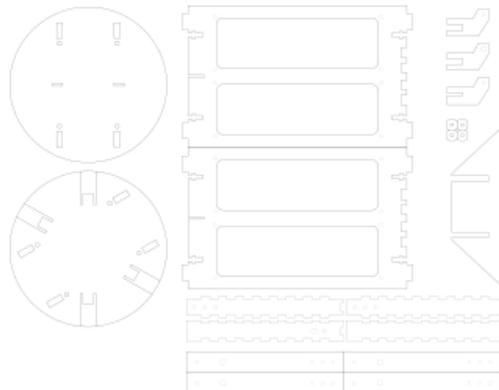


図 3.17 機体 4号機の図面

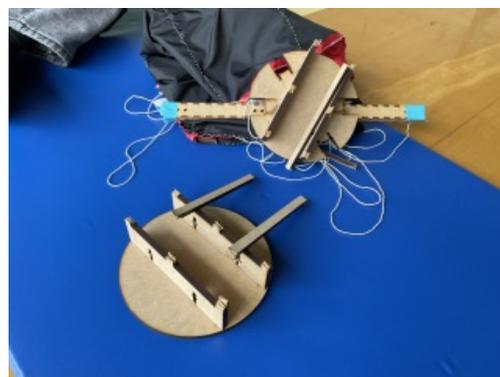


図 3.18 破損例 (2号機)

5. 5号機 (直径 150、高さ 220)

大会には5号機で出場した。結果的に大きな破損はなく、軽量で強度の高い機体を製作することができたと言える。

- 固定用木組みの溝を修正
- 軽量化のため横板の肉抜き
- 電装の取り付け用の板、ネジ穴を追加
- 下部に緩衝材となるスポンジ (直径 150、厚さ 20) を接着
- 紐の巻き取り、集束用の板を追加

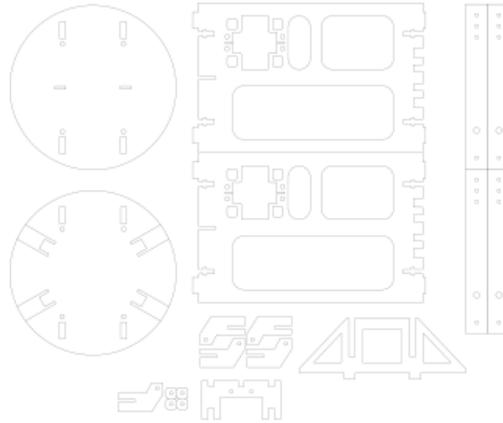


図 3.19 機体 5号機の図面

6. 6号機 (直径 150、高さ 220、重さ 324)

大会後は課題であった、パラフォイルの紐が絡まないように改良を行い、紐の巻き取りに関しても摩擦を減らす変更を行った。

- 上部円盤、固定用木組みの引っかかりを減らすよう改良
- 横板がズレないように突起を追加
- 紐集束用の板を削除、代わりに上部円盤に穴を追加
- 軽量化のためスタビライザーの根元の長さを短く変更

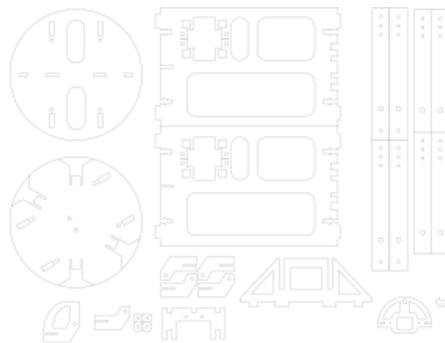


図 3.20 機体 6号機の図面

(※文責: 佐藤俊介)

工房

ここでは機体製作にあたって使用した図面の制作過程、工房のレーザーカッターについて説明する。

1. 図面

図面は adobe の illustrator を使用した。アカウントは PWG に申請し、受け取った学生用のキーを使って作業を行った。使用方法は工房の PC を使い、工房職員の方に教わって習得した。

レーザーカッターを使用する上で寸法を数値で打ち込み、線をパスとしてつなげる必要があり、最初はかなり苦戦したがプロジェクトを通して慣れることができた。

2. レーザーカッター

レーザーカッターは工房のものを利用した。レーザーカッター講習のビデオを視聴し、実際に工房職員立ち合いの下レーザーカッターを用い、使用方法を学んだ。前述した図面をレーザーカッター用のソフトを使い、線の選択や基準位置の設定、木くずを吸う吸引機の操作などを行う。

MDF 板は小さい部品には 2.5 mm、強度が必要な部品には 5.5 mm と 2 種類の厚さを用途に合わせて使用した。強度としては十分な信頼性があり、機体の様々な場所で使用した。

(※文責: 佐藤俊介)

3.2 コード班

コード班では落下検知と空中制御を行うためのアルゴリズムを作製し、それに基づいた電装とプログラムを作成した。

3.2.1 アルゴリズム

昨年度出場したスペースプローブコンテストにおいて空中制御を行っていたため、そのノウハウを生かせることから我々も空中制御を行って目標への到達を目指した。そのため我々は GPS と 9 軸センサーを用いたアルゴリズムを考案した。また、昨年度考案されていた GPS と 9 軸センサーを用いた接近方法は内容が複雑になりコーディング段階でのミスが予想されたため今年度は 9 軸センサーを落下検知にのみ使い、空中制御は GPS のみで行うこととした。

次に実際のアルゴリズムの内容について説明する。まず、機体がドローンから落下し始めると 9 軸センサーの内部の加速度センサが機体の落下を検知し、GPS が現在位置のデータ取得を開始する。この現在位置というのは経度と緯度の値で帰ってくる。そこから GPS のデータと目標地点との修正角を計算する。そして、GPS のデータより機体と目標地点との距離および機体の進行方向と修正角から、機体の旋回方向を割り出す。この流れを繰り返し、目標地点へ接近し最終的に着地する。

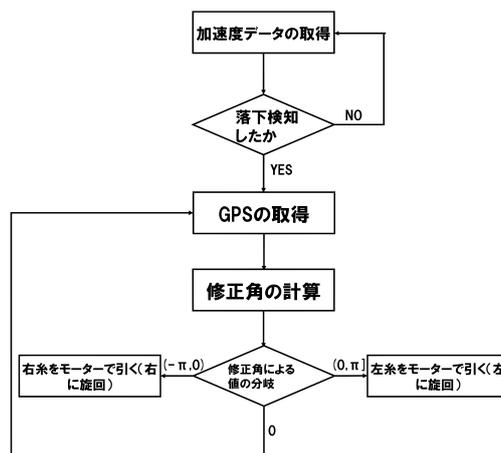


図 3.21 フローチャート

上記のアルゴリズムを用いて大会に出場した結果、プログラムを自動実行した際に外部からでは実行されているのかどうかという点がわからず、結果としてコードが正しく実行されているのかわからなかった。そのため大会が終了した後はそれを改善するため上記のアルゴリズムに新たな処理を追加した。上記の大会で使用したアルゴリズムにおいてはまず加速度センサが機体の落下を検知するとあるが、その前に GPS データを取得できたのであれば、モーターを動かすようにした。この処理を加えたことによって自動実行が成功しているのであればモーターの動きからそれがわかり、自動実行についての問題は解決された。

(※文責: 山口直人)

3.2.2 電装

本グループの参加するスペースプローブコンテストでは目標地点に機体を飛行誘導する必要がある。機体の制御に使用する主なセンサなどは以下のようにになっている。

- Raspberry pi 3 Model B
- モバイルバッテリー (FBT-PSEAC96PDNV)
- 9v 乾電池
- サーボモータ (MG996R)
- 3 端子レギュレータ (L7806CV)
- GPS (AE-GYSFDMAXB)
- 9 軸センサ (BMX055)

メインの制御用マイコンとして Raspberry pi 3 Model B を採用し、電源としては 5v のモバイルバッテリーを用意した。Raspberry pi 3 Model B が各センサ、モータなどの制御を行う。

今回は機体の飛行誘導を行うためにサーボモータでパラフォイルの制御紐を引っ張り、進行方向を制御することにした。昨年度の検証結果から、使用するサーボモータを MG996 に決定した。電源電圧がメインシステムとは異なるため、別途サーボ用の 9v 電池を搭載し、3 端子レギュレータを用いて電圧を調整する。

次に、進行方向の決定のため、GPS を用いて機体位置の特定をすることにした。スペースプローブコンテストでは目標地点の GPS 座標が配布されるため、GPS を用いて機体位置の特定をし、目

標地点との差分を求め制御の指針を決定する。

また、位置特定の補助や機体の状態の確認のため、機体の加速度検知や姿勢検知が可能な 9 軸センサを使用することにした。9 軸センサとは、加速度、ジャイロ、磁気を測定することができるセンサである。今回は加速度センサとして BMX055 を使用して、機体の加速度を検知し、制御する予定である。加速度センサで取得した値は落下速度の計算と風による機体への影響を計算することに使用する。

そして、これらのモジュールを搭載した機体の開発を円滑に進めるため、機体の行動履歴を調べることができるよう SD カードを組み込み、各種データを記録できるようにする。

今回の機体では、GPS と 9 軸センサは直接 RaspberryPi に接続し、サーボモータは 9v 電池、3 端子レギュレータやコンデンサなどの電子部品とともに別の基盤に接続することとした。機体の落下中に電子部品が外れ回路が切断されることを防止するため電子部品は基盤にはんだ付けした。また、GPS や 9 軸センサを完成させるため本体とピンをはんだ付けをした。

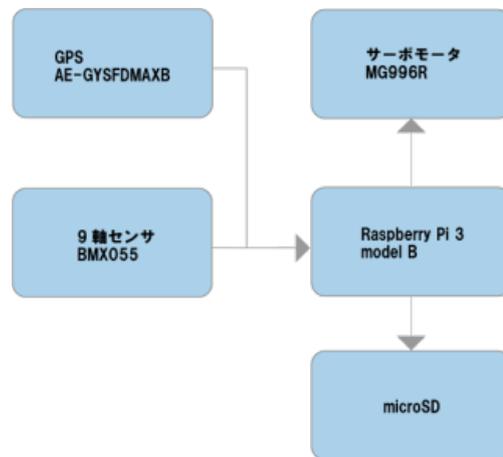


図 3.22 ブロックダイアグラム図

ここまでをスペースプローブコンテスト前に行った。スペースプローブコンテスト後は、回路の問題点の発見やその解決を重点的に行った。

回路の電圧が高すぎるのが原因でサーボモータの動作に問題が生じていることがわかった。電圧の調整のため 9v 電池や制御部品を取り外し、代用として以前よりも電圧の高いモバイルバッテリーを RaspberryPi とサーボモータの共用の電源として使用することでサーボモータの動作を改善した。

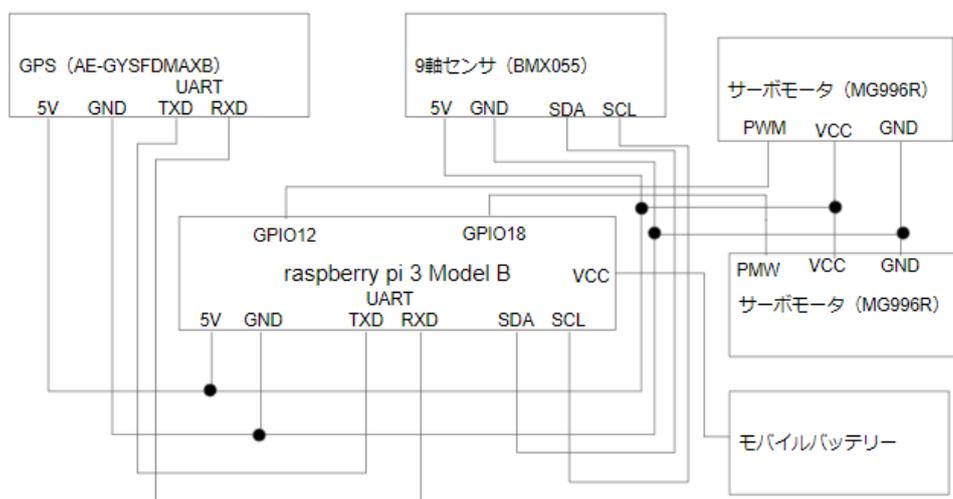


図 3.23 回路図

(※文責: 寒河江改)

3.2.3 メインプログラム

メインプログラムを作成するに当たって、採用するコンピュータの選定をした。候補には Arduino と Raspberry Pi が挙げられた。それぞれの利点を比較し、並行処理の性能と拡張性の高さという点から Raspberry Pi を採用した。開発に使用する言語は、Raspberry Pi に標準でインストールされていて運用が容易という点から Python を採用した。

メインプログラムのコーディングを開始する際、昨年度のソースコードを参考にして作成する方針と白紙の状態から開始する方針の2つが挙げられた。メンバーの Python の学習状況や昨年度からの引き継ぎを考慮した結果、昨年度のソースコードの解読と同時に Python の学習を進めるという方針に決定した。

また、GPS を操作する micropyGPS ライブラリの導入や各センサからのデータ取得を行う関数の作成を行った。

制御手順についてはアルゴリズムの項の通りである。

続いて、導入したライブラリやモジュール、作成したクラスやメソッドを説明する。

(※文責: 長嶋康太)

外部モジュール

micropyGPS

Python3.x と MicroPython で動作する GPS データの解析ライブラリで、入力されたデータを解析し GPS オブジェクトに追加・更新する。そして日付や緯度経度のデータを NMEA-0183 フォーマットの文字列として送化する。

serial

シリアル通信をするためのライブラリで、デバイス名、ポートレート、タイムアウト値を指

定して接続する。作成したプログラムの中では GPS のデータを取得するために使用した。

smbus

I2c 通信をするためのライブラリで、作成したプログラムの中では 9 軸センサから加速度データを取得するために使用した。

threading

Python で複数の処理を並行に実行するためのモジュールで、作成したプログラムの中では GPS データの取得と機体の制御を並列で処理させるために使用した。

RPi.GPIO

Python で GPIO ピンの入出力などを制御するためのモジュールで、作成したプログラムの中ではサーボモーターを制御するために使用した。

(※文責: 長嶋康太)

使用した標準ライブラリ

標準ライブラリとは、python に標準で付属されているためインストールしなくても、使うことができるライブラリである。ここでは、主に使用した標準ライブラリについて説明する。

csv

csv 形式のデータを読み書きするためのモジュールで、作成したプログラムの中では動作している機体の加速度、GPS データを csv ファイルに記録するために使用した。

sys

sys は Python の実行環境に関する情報を扱うためのライブラリで、作成したプログラムの中ではプログラムの終了、モジュール検索パスを追加のために使用した。

os

OS のさまざまな機能を利用するためのモジュール、作成したプログラムの中ではデータ記録用の csv ファイルを作成する際に、ファイルの存在を確認するために使用した。

math

数学計算用の関数を集めたモジュール、作成したプログラムの中では取得したデータから制御する値を計算するために使用した。

(※文責: 長嶋康太)

自作モジュール

main

落下検知を行い、制御用のモジュールを呼び出す。Raspberry Pi 起動時に実行される。

bmx055

Raspberry Pi から 9 軸センサ (bmx055) へアクセスするためのモジュールであり、i2c 通信によって 9 軸センサ (bmx055) から加速度データを取得する。

Controller

機体の制御のために計算を行うモジュール。GPS の値からサーボモータの制御するための値に変換するために用いる。

GPS

機体で使用する GPS (AE-GYSFDMAXB) からデータ取得を行うためのモジュール。

serve

機体の制御のために使用するサーボモータ (MG996R) を制御するためのモジュール。

(※文責: 長嶋康太)

各クラスとメソッドの説明

続いて、クラス図を以下に示す。

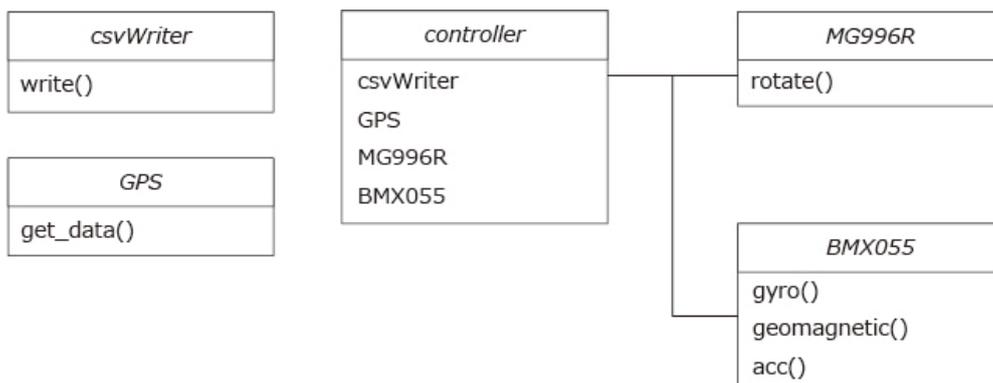


図 3.24 クラス図

各クラスとメソッドについて説明する。

class GPS(threading.Thread)

GPS との通信を行い、データ取得をするクラスである。threading.Thread クラスから継承することで並列処理できるようにした。

class servo(self,pin)

サーボモータを制御するためのクラスである。引数 pin に GPIO ピン番号を入れることでそのピン番号から出力するサーボモータを制御するインスタンスを作成できる。

def write(self, angle)

角度を指定しサーボモータを制御する関数である。引数 angle に 0 度から 180 度の角度を指定することで、その角度にサーボモータを回すことができる関数である。

class bmx055()

9 軸センサである BMX055 からデータ取得を行うクラスである。

def acc_value(self)

9 軸センサから x,y,z 軸の各加速度値を返す関数である。

class controller()

GPS のデータから様々な計算を行うクラスである。

def set_goal(self, gps_x, gps_y)

GPS のデータから 4 つある目標の中で最も機体と距離が近い目標を計算して、その目標の緯度経度を返す関数である。

def dis_gps(self, old_x, old_y, now_x, now_y, dir_x, dir_y)

GPS のデータから機体の進行方向を計算し、機体の進行方向と目標の方向が左右どちらにずれているかを値として返す関数である。

(※文責: 長嶋康太)

3.3 体育館実験 (1 回目)

実験概要

実際に試作したパラフォイルと MDF 板で作った本体の骨組みを接続し、体育館実験で簡易飛行実験を行った。実験内容としては体育館の上から試作品を落とし、パラフォイルが機能するかどうかを確かめた。

体育館の上から地上までは 9m あり、簡易的な飛行実験を行うには十分な環境であったまずはパラフォイルに軽い材木 (縦 5cm × 横 5cm × 長さ 10cm) を取り付け、パラフォイル単体の飛行実験を 2 回行い、続いて MDF 板で作った本体を取り付けて、実際に本体が装填された状態でパラフォイルが機能するかの飛行実験を 3 回の計 5 回飛行実験を行った。また、機体が床に着地した際

に床に傷がつかないように、実験の事前に衝撃吸収マットを4枚、推測される落下地点に敷いて実験を行った。実験は3名で行い、カメラで撮影する人、体育館上から投下する人、データをPCに入力する人に分かれて各自役割を遂行した。

実験結果

実験結果を以下に示す

パラフォイル単体実験 重り：材木（縦50mm×横50mm×長さ10mm）を取り付けた場合

一回目：406cm 進む 左に転回

二回目：70cm 進む 左に転回

MDF板試作品を取り付けた飛行実験 重り：MDF板試作品（直径150mm×高さ220mm）

一回目：390cm 進み、左に転回

二回目：320cm 進み、左に転回

三回目：235cm 進み、左に転回

この実験結果から得られた事は、どの試行も左に転回している事が分かった。のちに調べてみると、パラフォイルの左側の紐が右側の紐に比べて若干短くなっていたことが分かった。しかしこの事から、紐の長さがしっかりとパラフォイルの起動に影響を与えている事が判明した。また、パラフォイルは落下直後は安定した飛行がみられたが、後半パラフォイルが潰れてしまい、中に空気が入らずに落下する様子が見られた。この実験結果から、パラフォイルの耐久性が構造的に足りなかった事が分かり、本パラフォイルの作成では素材、縫い方、空気力学を考えた最適なパラフォイルの形状を再検討するスケジュールを組み立てた。また、体育館実験を行うにあたって、事前に体育館の床に衝撃吸収マットを敷いたが、機体がマット外に飛行し、手でキャッチする事が多かった為、次からの体育館実験ではより広範囲に衝撃吸収マットを敷く事とする。また、飛行終了した機体を手で持ってまた体育館の上まで登って投下、といった形を繰り返した為、実験以外の機体準備時間が大きくかかってしまった事から、次回からの実験では事前に機体に紐を取り付け、飛行が終了したら体育館上にいる人が紐を巻き上げて機体を回収する形式にして機体準備時間を短縮するようにする。

（※文責：今井瑞貴）

3.4 体育館実験 (2回目)

実験概要

機体班では、作製したパラフォイルの紐の長さの最終決定を行うため体育館実験を行った。実験は以下の手順で行った。

1. 軽いおもり (200 g) を取り付け、高さ 8m から広げた状態で投下し距離を記録。
2. 紐の調整を行い、同じ条件で投下し距離を記録。正面に進むまで繰り返す。
3. 正面に進むことを確認し、徐々におもりを増やし再度投下。

実験結果

実験の結果、以下の事象が見られたため修正を行った。

- ①その場で回転してしまい垂直に落下。
- ②パラfoilが中央で潰れてしまい落下。
- ③紐がスタビライザーに引っかかり展開せず落下。

①は左右の紐の長さを変え、正面方向に進むように調整を行った。②は中央部と端部での紐のバランスを見直した。③は根本的に修正することが難しく、スタビライザーを使ったこの機構の弱点ともいえる問題であった。大会までの期間も考え、全体的に紐の長さを短くすることで応急的な対処をとった。

今回の実験でパラfoilの紐の長さが決まり、進行方向に正しく進むように調整ができた。

(※文責: 佐藤俊介)

3.5 大会前ダム実験

実験概要

体育館からの落下実験ではパラfoilの正常な展開を確認が出来る十分な高度が確保できなかった為、大学近くの笹流ダムの上から機体を降下させる実験を行った。高度は約 25m で、本番の 4分の1の高さではあるが、パラfoilが正常に機能しているかを確かめるには十分な高度であった。当日の実験では

気温：26 度

湿度：66 %

風速：6m

風向：北西

という飛行には大きく影響を与えることのない環境であった。

実験結果

結果としては、4回の実験の中で3回がパラfoilが十分に開かず、比重が一方に集中することによって回転しながら自由落下する形になった。原因として、スタビライザー部分やスタビライザーを固定するネジの部分に紐が引っ掛かったこととパラfoilの強度のバランスが悪かったことが挙げられた。スタビライザー部分や固定するネジの部分に引っかかることについては実験日が機体を大会運営に送る1日前だったこともあり、ネジ部分の処理を行う時間がなく、パラfoilの畳み方を工夫する対策が取られ、パラfoilの強度のバランスについては縫い方を一部変更することで対策をした。

(※文責: 和山慧音)

3.6 大会後体育館実験

実験概要

大会ではコードでの進行方向の修正を行っていたため、機体班では紐の長さの違いによる角度の違いを把握しきれていなかった。従って、大会後見つかった課題に改良を加え、以下の目的を設定し、再度体育館実験を行った。

目的 1：改良した機体が紐が引っかかりづらくなったのかの確認

目的 2：パラフォイルが紐の引っ張った長さに対して正しい挙動をとっていたかの確認

大会ではコードでの進行方向の修正を行っていたため、機体班では紐の長さの違いによる角度の違いを把握しきれていなかった。また、大会では紐がスタビライザーの挙動部分に引っかかったために完全な展開に至らず、制御に移ることができなかった。今回の実験では、パラフォイルの紐の長さを左右 1cm ずつそれぞれ変更し、紐の長さで進行方向に影響を与えるかどうかを実験した。

実験は計 5 回、左右の紐を 1cm ずつ、8.5cm~10.5cm の範囲で変更しつつ進行方向を確認した。

実験結果

実験結果は、左右の紐が同じ長さの場合は投下する正面方向から約 60 度、2 時の方向に進行する事が分かった。これは、左右の長さが 8.5cm、9.5cm、10.5cm のいずれも変わらない結果となった。右に進行してしまうことから、パラフォイルのミクロな構造問題や機体の重さのバランス等で右側に重心があると考察し、次は右の紐を左の紐よりも 1cm 長くして飛行させた。結果は正面から 1 時方向、約 30 度のズレで進行した。結果から、我々は左右差 1cm の差で 30 度の角度調整が可能であると推測した。これをふまえ、左右差 2cm で実験を行ったところ、正面方向に飛行した。今回の実験から、正面方向に飛行する為の最適な紐の長さを計算する事ができた。これは当初想定していたよりも変化が大きく、パラフォイルを作る際にもミリ単位での調整が必要であることが分かった。

(※文責: 今井瑞貴)

3.7 大会後ダム実験

実験概要

スペースプローブコンテスト終了後に笹流ダムにて投下実験を行った。実験目的はパラフォイルの紐が絡まないように改良した機構が正常に機能しているかの確認と、落下検知及び制御システムの動作検証である。実験内容は機体をダムの頂上 (高度約 25m) からパラフォイルを開いた状態で投下し、飛行時の動作を確認するものであった。比較条件として、マイコンの電源を ON にした投下と、電源を OFF にした投下でシステムや機体の動作に差があるかを確認した。

実験結果

機体班の実験結果としては、パラフォイルを開いた状態である為、スタビライザーのパラフォイル展開の補助の役割の確認は行えなかったが、パラフォイル展開状態の維持と安定という役割は十分であることが確認された。

コード班の実験結果としては、モジュールの故障などにより今までの実験では GPS の動作を確認することができていなかったが、この実験で初めて動作を確認することができた。また、左右のサーボモータで紐を引っ張ることによって機体の左右の旋回することを確認できた。しかし、この実験において以下の 2 つの問題が判明した。

- サervoモータによる制御の精度

サーボモータの引っ張りが強かったため、旋回の色度が速く、この方式で機体を制御する場合にはどのくらい引っ張ることでより精度が良い制御をできるかをさらに細かく調べる必要があることが判明した。

- プログラムの自動起動

プログラムをマイコンの起動によって自動起動しようとしたところ、GPS データの取得によってエラーが起きた。それにより、自動起動が失敗し、自動起動によるプログラムの起動からデータの取得を確認することができなかつた。この問題の原因は起動時の GPS モジュールは動作が不安定であったため、GPS モジュールが起動後 GPS の取得が安定するまで待つように改良する必要がある。

(※文責: 長嶋康太)

第 4 章 インターワーキング

この章では、プロジェクト全体の進捗や班の活動を相互かつ円滑に進行するために行ったプロジェクト内における活動について述べる。

(※文責: 佐藤俊介)

4.1 プロジェクト全体のインターワーキング

本プロジェクトでは活動目的によってグループ分けを行い、機体班とコード班の 2 グループで活動を行った。前期のプロジェクトでは、プロジェクト全体で対面でのミーティングを週に 2 度行い、必要に合わせて他日にも班単位で大学に集まり活動を行った。また、全体ミーティングの開始時には、各班のリーダーが担当教員及びメンバーに先週の進捗を発表し、常に現状の把握と課題の見直しを図った。

プロジェクトの体制として、プロジェクト全体の管理を行うプロジェクトリーダー、各班の管理を行う班リーダー、班内では各個人に役割を設け活動を行った。

使用ツールに関しては、昨年度同様ミーティングに Zoom を一部使用し、メンバー間の意見交流や連絡には Slack を導入した。また、情報や資料は Notion で管理、共有を行い、タスク管理には Google ドライブを活用した。

(※文責: 佐藤俊介)

4.1.1 ナレッジマネジメント

ナレッジマネジメントとは「知識を共有して活用することで、新たな知識を想像しながら経営を実践すること。」である。

昨年度の活動では、「暗黙知」と呼ばれる個人の直感などに基づく言語化できない知識を、「形式知」と呼ばれる図や文章で表せる言語化可能な知識に変換し共有するという事に重点を当てていた。今年度の活動にもこのナレッジマネジメントを踏襲し、活動していくことになった。さらに、SECI (セキ) モデルと呼ばれるフレームワークを部分的に取り入れた。SECI モデルとは、個人が持っている暗黙知などを次の 4 つの段階に分けて繰り返すことで企業・組織の知的財産を築いていくフレームワークのことである。

- Socialization (共同化) : 個人の「暗黙知」を、同じ体験を通じてお互いに共感し理解する段階
- Externalization (表出化) : 共通に得た「暗黙知」を言葉や図などで明確にした「形式知」に変換する段階
- Combination (連結化) : 新しい「形式知」と既存の「形式知」を連結させて新しく知識体

系を作る段階

- Internalization（内面化）：新しく作られた知識体系を実際に体験し個人が身につけることで新たな「暗黙知」へと変換する段階

従来の SECI モデルに比べ不完全な物になるがこの 4 つの内、Socialization（共同化）と Externalization（表出化）の 2 つを取り入れることとした。これらの目標を達成するため、コミュニケーションツールである Slack を用いた。

（※文責: 寒河江改）

4.1.2 コード管理

本プロジェクトにおいてソースコードを管理する際には主に Github を用いて行った。また、その補助的なものとして slack と notion も効果的に活用した。

- Github
Github ではプログラムの変更やソースコード受け渡しについて使用した。またここでソースコードを見ながら改善点を探ったり、追加するクラスについて話し合うことも行った。また Raspberry pi 3 にソースコードをうつす時にも Github を介して移動させ、最終的に完成したソースコードもここに保存した。
- slack
slack では実際のコードの受け渡しなどは行わなかったが、その代わりにコードを作る際に参考になるサイトであったり利用した Github であったりの url や、エラーに対してどうしたら直ったのかの情報のやり取りを行った。
- - notion
notion では slack のように実際のコードの受け渡しなどはしなかった。利用方法としては notion も slack とほぼ同じ利用をしていた。異なる点として slack は過去の情報がどんどん流れてしまい、古い情報が再び必要になった時に利用できなくなるが、notion では古い情報も容易に発見できた。これを利用して長く必要になるであろう情報は slack だけでなく notion にも保持をしていた。

（※文責: 山口直人）

第 5 章 成果

この章では、中間発表会の概要と、開発やスペースプローブコンテスト前の成果について述べる。

(※文責: 今井瑞貴)

5.1 プロジェクトの成果

この項ではプロジェクトの成果について述べる。

(※文責: 佐藤俊介)

5.1.1 スペースプローブコンテスト前の成果

スペースプローブコンテスト前の成果として、まず活動を通して得られた知識について説明する。CanSat を設計及び制作する過程でハードウェアを利用するプログラム等についての知識やレーザーカッターの使用方法に関する知識を得ること出来た。また、チームで円滑に活動する為のコミュニケーション方法や協力することによる作業速度の向上を学んだ。機体班、コード班それぞれの成果物について説明する。

(※文責: 和山慧音)

機体班

機体班のスペースプローブコンテスト前までの成果物について説明する。機体班では、前期に機体本体及びパラフォイル、スタビライザーの試作品の設計、製作を行った。パラフォイルは後部の接続部につながる紐を引くことで進行方向を調整するという制御方式であった。スタビライザーはパラフォイルの展開を補助するため、機体本体の上部に接続する形での製作を行った。機体本体は前期期間ではサーボモータ及びセンサ、Raspberry Pi のサイズが不明瞭であった為、パラフォイル及びスタビライザーの展開に影響がでない固定を目的として製作を行った。

試作品完成後、パラフォイル及び機体本体、スタビライザーのみでのそれぞれの実験及びパラフォイルと機体本体、スタビライザーを組み合わせた実験を行った。その際にでたパラフォイルの安定性の問題やスタビライザーの展開が不十分となっている問題、機体本体が着地時に大きく破損してしまう問題についての解決策の模索を行った。

そこででた解決策をもとに夏休みの大会前の期間で機体本体及びパラフォイル、スタビライザーの本製作を行った。パラフォイルに関しては展開時の安定性の向上するように縫い方や空気取り込み口を増やすといった設計に変更し製作した。スタビライザーに関しては展開が十分となるようにスタビライザー自体の重量の軽減や幅の調整を行いながら製作した。機体本体は着地時に大きな破損がないように機体構造の変更や衝撃を吸収するスポンジと破損パーツの飛び散りを防ぐカバーの製作を行い、またスタビライザーや制御紐、サーボモータ及びセンサや Raspberry Pi を設置し、相互に悪影響がでない位置になるように製作した。

次に、スペースプローブコンテスト前の実験結果について説明する。試作品による実験ではパラフォイルの減速機構についての確認や制御紐の長さの違いによる方向の変化の確認を行えた。また、問題点としてスタビライザーの展開が試作品段階の構造では十分に行えないことが確認できた。その後の本製作したパラフォイルやスタビライザーを用いた実験では機体重量による機体制御の影響を確認した。しかし、スタビライザーの補助が正常に行えていないこと、パラフォイルの形が崩れ、確実な展開が行えていないということが問題点として上がった。スタビライザーに関してはパラフォイルの収納の際の畳み方を工夫することで展開の確実性を向上を目指し、パラフォイルの形が崩れることに関しては縫い方を一部変更することで展開の安定性の向上を目指した。

大会前の懸念点としてパラフォイルやスタビライザーを実装した機体の実験で確実な展開と制御が行えている結果がでなかったこと、またサーボモータ及びセンサ、Raspberry Pi を搭載した機体での実験が行えていないことが挙げられた。

(※文責: 和山慧音)

コード班

コード班のスペースプローブコンテスト前までの成果物について説明する。コード班では、アルゴリズムの検討、機体制御システムの作成と電装の検討及び作成を行った。検討したアルゴリズムはGPSと、9軸センサ内部の加速度センサの値によって機体投下後の落下検知を行い、目標地点へと逐一誘導させるというものである。機体制御システムはGPS、加速度センサの値から、パラフォイルの紐の長さを、サーボモーターを用いて調整し制御を行うというものである。電装に関して、機体のレギュレーション規定の重量を超えないようなモーター及びバッテリーと、必要となる機能のモジュールの決定と適切な回路を作成した。回路に関して、9v電池とモバイルバッテリーを採用したため、Raspberry pi 3 Model Bとサーボモーターそれぞれで過電圧が生じないような回路図を作成し、3端子レギュレータなどの必要となった電子部品のはんだ付けを行った。また、機体と電装を合わせる際に配線や各モジュールなどの絡まりやすいものに関しては、ボンドを固定やユニバーサル基盤上にはんだ付けをし、電装ができる限りコンパクトな状態かつ物理的な脱線が起きにくい電装を作成した。

大会前の懸念点として、機体班で挙げられたようにサーボモータ及びセンサ、Raspberry Pi を搭載した機体での実験が行えていないことが挙げられた。これらの原因として、コードの学習及び、文献調査に多くの時間を割いてしまったことや、GPSが大学構内であると受信できないことに気づくのが遅れてしまったこと、必要となるモジュールや電子部品が半導体などの影響により入手が遅れてしまったことなどが挙げられる。

(※文責: 中澤一輝)

5.1.2 スペースプローブコンテストの結果

ここではスペースプローブコンテストの大会側から受けた評価の結果と機体班・コード班それぞれの結果を述べる。

(※文責: 和山慧音)

大会から受けた評価

大会成績は以下の 6 つの評価基準で決定される。

- 着地の正確性
- 事前準備・実験環境
- 制作技術
- 大会当日の運用
- プレゼンテーション
- オリジナリティ

そしてこの 6 つの内、「当日の運用」を除き平均を下回っていた。機体の落下地点は 4 つのポイントのうち、西のポイントからさらに 29.04m 離れた地点であり、着地の正確性は全体で 4 番目に高い記録であった。

大会から受けた評価は以下のものであった。

機体の収納スペースとパラフォイル展開の 2 つの要求を満たす折りたたみ式のスタビライザーのアイデアは効果的であった。

落下時間 10 秒程度の間でパラフォイルにどのような動きを要求しているのか、10 秒間で姿勢制御がどの程度可能なのかを検討すると機体の改良の方針決定が容易になる。

飛行データは試験後の解析においても重要な要素なため、機体本体での保存と無線通信による地上局での保存など複数箇所での保存を推奨する。

以上の評価を受けて、機体班ではスタビライザーやパラフォイルの利点を活かすために展開の安定性や円滑さを向上させることができるような修正、コード班ではデータの保存方法を見直し、安定してデータ取得・保存が出来るような修正が必要だと考えられる。

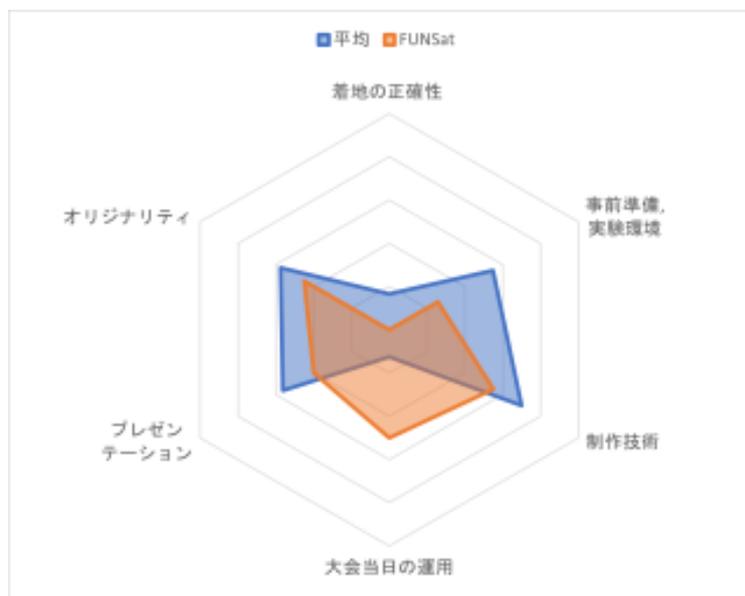


図 5.1 大会評価

機体班

スペースプロブコンテストの結果から機体班では、パラフォイルの紐が制御機構に引っかからないような機構を作る必要があることが分かった。そのために、パラフォイルの展開の実質的な失敗の要因であった飛び出していたスタビライザー固定用のネジを断ち切り、安全のためやすりで削った。

また、スタビライザー自体にパラフォイルの紐が巻き込まれる可能性、パラフォイルの紐の巻き取り機構が機体内部の電装に当たり巻き取りを阻害する可能性が見つかった。スタビライザー自体にパラフォイルの紐が巻き込まれる可能性については、機体上部スタビライザー部分にカバーを取り付けることで改善した。これにより、パラフォイルの紐が制御機構に引っかかることを防ぐことができる。さらに、パラフォイルの紐の巻き取り機構が機体内部の電装に当たり巻き取りが失敗する可能性については、よりスムーズに巻き取るため、半円とした。これにより、軽量化すると同時に、より正確に紐を巻き取れる機構とした。

（※文責: 今井瑞貴）

コード班

スペースプロブコンテストの結果からコード班では「データの取得」「GPS の取得」「サーボモータの動作」「自動起動」の4つの課題が明らかになった。そのため、スペースプロブコンテスト後の活動では、これらの課題を解決するために以下のような改善を行った。

まず「データの取得」の失敗に関して、原因はプログラムを動かすとファイル名を日付、時間を用いて自動で作成するようにしていたが、raspberry pi はインターネットに繋がっていないと正確に時刻でファイルを作成することができないと判明した。したがって、時系列が正確に設定できないことが原因で、時系列が同じファイル名でデータファイルを作成してしまっていることが確認できた。そのため、時系列を指定するのではなく、番号が付随したファイルを作成し、番号はファイルが作成されるたびに増えていくようなプログラムに変更した。

「GPS の取得」の失敗に関して、電源が入ってから GPS の起動には時間がかかることが原因であったため、電源が入ってから GPS が起動するまでプログラムの動作を待つプログラムを追加して問題を解消した。また、GPS の取得が可能な状態になったことが機体外部から直接目視できるように、サーボモーターを 180° 回転させるという予備動作を行うようにして、GPS の取得が可能かどうか判断できるプログラムも追加した。

「サーボモータの動作」が不安定である問題に関して、9v 電池の電圧が強いことが原因であったため、モバイルバッテリーの電力が小さいものを使用し、9v 電池と制御のための電子部品を使用しないことで問題を解消することができた。

また、「自動起動」が正常に行われたい問題に関して、コマンドプロンプトからプログラムを起動した場合にインポートエラーが発生することを確認したため、インポートするファイルを指定するプログラムを作成することでエラーを解消し、自動起動が正常に行うように改善した。また、自動起動の際にエラーが起きてしまったとき、そのエラーがどのような内容であったかを判断できるように、発生したエラー文をファイル内に書き込んで逐一確認できるようにプログラムを改善した。これにより、エラー文の内容から正しいプログラムの改善を行うことができた。

以上のような活動をコード班ではスペースプロブコンテスト後の活動として行った。また、スペースプロブコンテスト前には機体に電装を付けた状態で実験を行うことができなかったが、スペースプロブコンテスト後には機体に電装を付けた状態で実験を行えた。主に大学屋外で自動起動が行われるかの検証実験とプログラム全体が正常に動作できるかを検証するダム実験を行った。これらの検証実験から上記の様な改善を十分に確認できたと判断した。

(※文責: 中澤一輝)

5.2 発表会

この項では中間発表会の概要や結果とその評価を行う。

(※文責: 寒河江改)

5.2.1 中間発表

概要

2022年7月8日金曜日に行われ、形式としては事前に制作していたスライドやポスターを用いて発表を行い、その後に質疑応答が行われるという流れであった。

本プロジェクトではポスターを用いて10分程度のプレゼンテーションを行い、5分程度の質疑応答を行った。

(※文責: 寒河江改)

結果と評価

当日の質問内容としては、スタビライザーやモーターなどのプロジェクトで扱った道具などの詳しい説明や実際にどのように動作するのかなど具体的な質問を多く、プロジェクト自体の進め方や目的などの質問はあまり多く受けなかった。

また、有効な評価アンケートの結果は37件であった。詳細を以下に示す。

まず、発表技術についての評価グラフを図に示す。評価点の平均は6.7であった。また、好意的なコメントとして「動画と実際の機体を交えて発表していたので理解しやすかった」「とてもスムーズに発表ができていた」といった意見が多く見受けられた。一方で、批判的なコメントとして「声が小さかった」「スライドなどが欲しかった」などという意見も目立った。

最終発表会ではポスター以外にスライドを用いて視覚的に見やすい発表を構成していく必要がある。

次に、発表内容についての評価グラフを図に示す。評価点の平均は7.5であった。

また、好意的なコメントとして「機体班とコード班の成果と課題が簡潔にまとめられていて分かりやすかった」「これまでの活動と今後の予定を端的に正確に伝えられていた」といった意見が見受けられた。批判的なコメントとしては「作ったものを最終的にどう活用したいかが明確になるとい

発表技術についての評価 / Evaluation about Presentation Skill (基準: プロジェクトの内容を伝えるために、効果的な発表が行われて...ress the project and its plan?)
37 件の回答

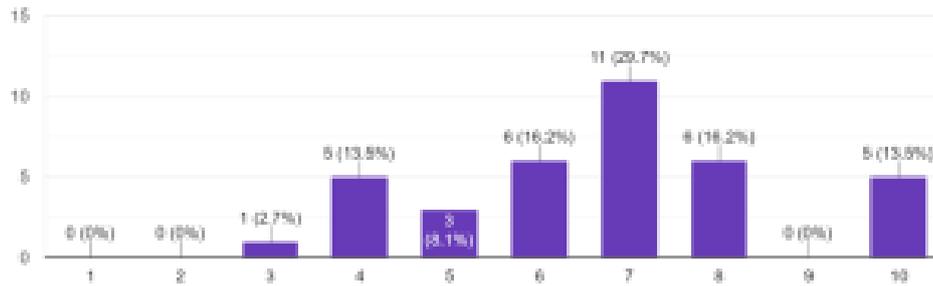


図 5.3 発表技術についての評価グラフ

い」「先輩から得たことを知りたかった」といった意見があった。

これらのコメントから、大会後の見通しや最終発表会では昨年度の具体的な引継ぎ内容の説明をする必要がある。

発表内容についての評価 / Evaluation about Presentation Plan (基準: プロジェクトの目標設定と計画十分なものであるか / Were the specified plans satisfied?)
37 件の回答

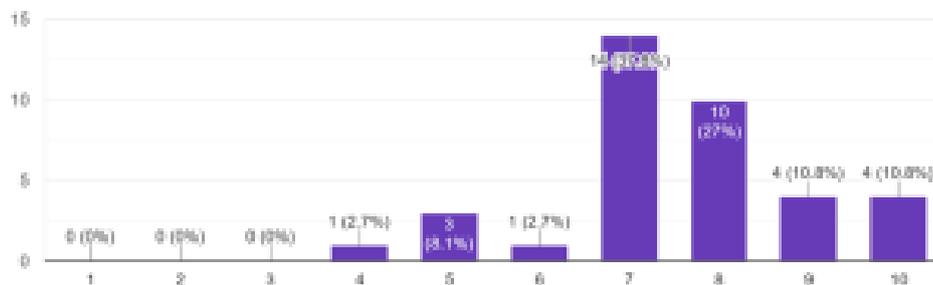


図 5.4 発表内容についての評価グラフ

(※文責: 中澤一輝)

5.2.2 成果発表

概要

2022年12月9日金曜日に行われ、形式としては事前に制作していたスライドやポスターを用いて発表を行い、その後に質疑応答が行われるという流れであった。

本プロジェクトではスライドを用いて10分程度のプレゼンテーションを行い、5分程度の質疑応答を行った。また、本プロジェクトの発表場所は食堂で行った。

結果と評価

まず中間発表で課題としていた、発表時の声の大きさや、スライドを用いた視覚的に見やすい発表に重きを置いて準備を行った。スライドに関しては図の活用と専門用語の説明をより短くかつわかりやすい文章にさせた。加えて、スライドでは不足している情報を補完させるためにポスターを計3枚用意した。ポスターに関しては、スペースプローブコンテストとその後の活動を詳しく記載した「メインポスター」を1枚と、機体班とコード班の詳細な活動を記載した「サブポスター」を2枚を用意した。そして、実際に大会で使用した機体と、大会後に改良して作成した機体を発表場所に展示して、評価者が理解しやすい発表になるよう工夫した。また、説明が10分以内に収まる内容の検討と、スムーズな発表になるように練習を入念に行った。以下が成果発表会の評価に関する情報である。

下の図 5.3 は成果発表会当日に評価を行っていただいた評価者数である。この図より、全評価者数 42 人のうち 88.1 % の学生と 11.9 % の教員から評価をいただいたと分かった。

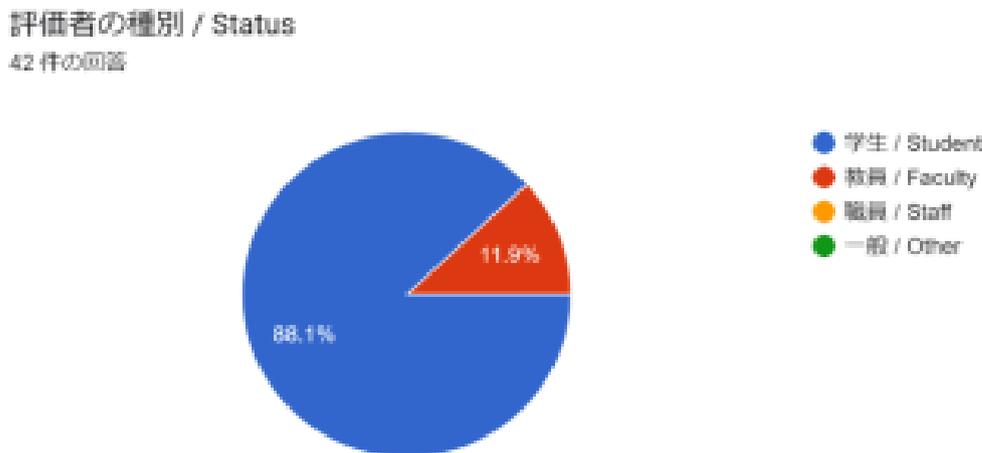


図 5.5 評価者の種別についてのグラフ

続いて当日受けた質問内容としては、昨年度の同プロジェクトとの変化や、中間発表前と後の活動の変化などの質問を受けた。「道具などの詳しい説明や実際にどのように動作するのか」などの機体に関する具体的な質問を受けた中間発表とは異なり、成果発表会では「全体的なプロジェクト内容に関する比較」の質問を多く受けた。

また、発表技術についてのコメント数は 39 件であった。詳細を以下に示す。

まず、発表技術についての評価グラフを下図 5.4 に示す。評価点の平均は 7.1 であった。これは中間発表と比べて値が大きかった。したがって、発表技術に関しては、中間発表と比べて高評価であったと考えられる。また、好意的なコメントとして「改良前と改良後どちらも実物を用意していたため、比較がわかりやすかった。」「何を目標にどのように取り組み、結果どのような成果が得られ、今後の展望も明らかにされており、とても身のある発表だったと思います。スムーズな発表で、発表準備に抜け目なさを感じました。」「少し緊張していたようですが、発表練習もしていたと思いますので、もっと自信をもってよいと思います。昨年度の発表よりも、ずっとわかりやすかったです。」といったコメントを受けた。そのため、実際の機体を展示したことや発表練習を入念に

行ったことは非常に効果的であったと考察できた。一方で、批判的なコメントとして「スライドや内容がとてもよかったです、音が広がりやすい食堂での発表だったので聞き取りにくかったです。」「発表者によって視線がこちらに向いていない人がいました。また、声量もちょうどいい人と少し小さい人がいました。」などというコメントも目立った。そのため、スライドを用いた発表は効果的であったと考えられたが、会場に合わせた声量の調整や、話者ごとの発表技術の向上を十分に練習する必要があると考察できた。

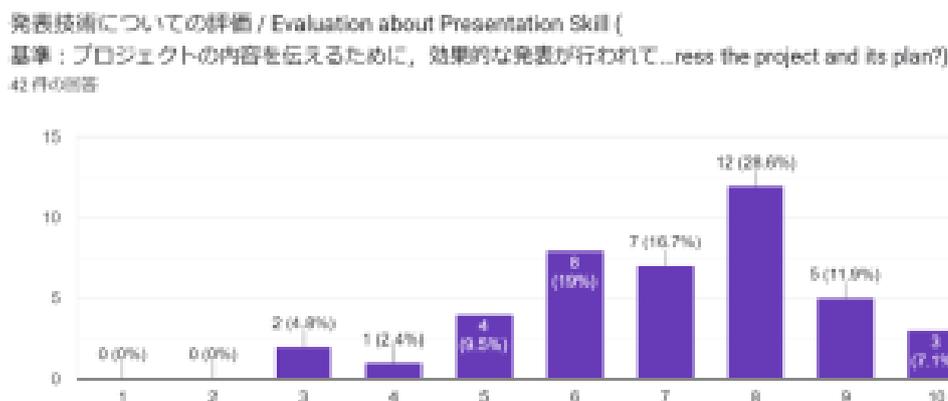


図 5.6 発表技術についての評価グラフ

続いて、発表内容についてのコメント数は 36 件であった。詳細を以下に示す。

まず、発表内容についての評価グラフを下図 5.5 に示す。評価点の平均は 8.2 であった。これに関しても中間発表と比べて値が大きかった。したがって、発表内容に関しても、中間発表と比べて高評価であったと考えられる。また、好意的なコメントとして「実際に実験ををこなった際の反省点をいかし、改良して再度実験を行なっているのがとても良いと感じました。」「映像を多く使用したり、実際のものを使用、用意していて良いと思った。」「どのようなものを作成したのか、とてもわかりやすく説明されていた。プロセスがはっきりしていた。班ごとに分かれて細かく設計していたと思った。アルゴリズムの表など載せたりしていた。大会に参加していたので充実していたことを感じられた。」といったコメントを受けた。そのため、本プロジェクトの具体的な活動などをスライドを通して視覚的に理解してもらうことができたと考察できた。一方で、批判的なコメントとして「専門用語をもう少し伝わりやすくしてほしい」「活動内容は分かりやすかった。質疑応答での回答を前提としたような発表内容の様に感じた。」といったコメントを受けた。そのため、10 分という短い時間でも初めて聞く人たちが十分に理解できる説明の発表や、ポスターにさらに詳しく記載する必要があると考察した。また、発表内容のコメントに関して全体的に好意的なコメントが見受けられた。したがって、中間発表の発表内容に関する課題を踏まえた改善ができ、理解しやすい発表が行えたと考えられる。

発表内容についての評価 / Evaluation about Presentation Plan (基準：プロジェクトの目標設定と計画十分なものであるか / Were the specified plans satisfied?)
42件の回答

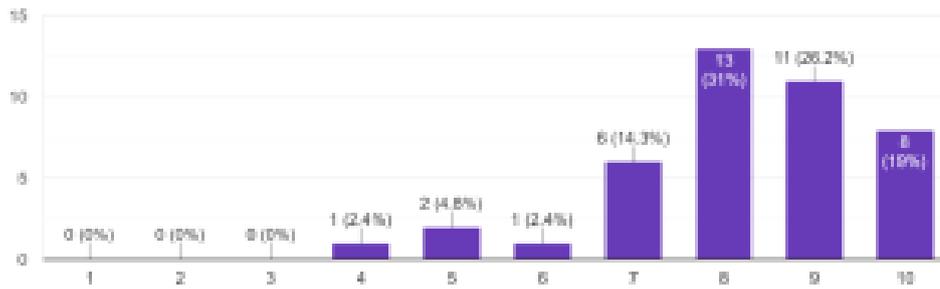


図 5.7 発表内容についての評価グラフ

以上より、これらの評価を踏まえて成果発表会では中間発表の課題を踏まえた十分な発表が行えたと評価者の視点からも判断することができた。そして、この成果を生かしてよりよい発表が行えるようにしていきたい。

(※文責: 中澤一輝)

第6章 おわりに

この章ではプロジェクトのまとめについて述べる。

(※文責: 寒河江改)

6.1 プロジェクトにおける個人の役割

この項では各メンバーが自分に割り当てられた課題についての結果とその評価を記述する。

(※文責: 寒河江改)

6.1.1 今井瑞貴

機体班のリーダーとして活動した。グループ全体への連絡、メンバーのタスクの進捗管理を行った。機体班としての活動としては、レーザーカッターを用いたMDF板の加工、illustratorを用いた設計図の作成、工房でのパラフォイルが機能するかの簡易実験、機体を構成する素材や部品の選定を行った。夏休み中は実際にスペースプローブコンテストに使うパラフォイルの紐の長さを選定、紐の取り付け作業を行う。また、メンバーの進捗管理やタスク割り振り、体育館実験やダム実験のデータ記録を行った。また、CanSatの概要やスペースプローブコンテストとの大会参加への連絡を請け負い、大会のレギュレーションについての質問、機体素材についての質問等、情報を得る役割を担った。一回目のダム実験の際の笹流ダムへのコンタクト、去年のダム実験での失敗を再現しないようにどのような対策のもと実験を行うのかを考えた。また、ダム実験や体育館実験でビデオ撮影による機体の撮影を行い、積極的に分析を行った。スペースプローブコンテスト終了後、機体が上手く飛行しなかった原因を把握し、どのような改善をするか、グループに目的と個人の役割、期日を明確化し、改善に向けて取り組んだ。プロジェクト期末発表では、グループまたはプロジェクト全体で行ってきた事を簡潔にスライド作成し、実際に機体を観客に見せて説明しながら、理解しやすい発表を行った。

(※文責: 今井瑞貴)

6.1.2 古賀海聖

本プロジェクトの機体班の一員として活動した。主な活動内容としては、機体、パラフォイルの機構の考案、作成を担当した。パラフォイルを正常に機能させるために既存のパラフォイルから採寸、作成をした。その後、体育館実験にてパラフォイルが正常に機能するかの確認を行った。結果としては、正常に機能せず多くの改善点が見つけられた。

(※文責: 古賀海聖)

6.1.3 佐藤俊介

本プロジェクトの機体班の一員として主に図面作成、設計を担当した。illustrator を使用してパラfoilと機体本体、スタビライザーの図面を作り、紙媒体への印刷やレーザーカッターでMDF 板等の素材の切り取りを行った。切り取った後は組み立てを行い、大きさに間違いがないか、強度不足なパーツがないかを確認し Notion を用いてメモをとり、管理を行った。また方向性に違いが出ないようにするため、機体班だけでなくコード班とも情報を共有しながらメンバーと機体の概要や改良案を話し合い、活動に取り組んだ。実験や実際に試作を多く製作したことで課題を見つけることができ、有意義な活動ができたと考えている。最終的に5回の改良を経て、当初の目標であった「頑丈で軽量の減速機構を備えた機体」を概ね達成することができた。

(※文責: 佐藤俊介)

6.1.4 和山慧音

機体班に所属し、主にパラfoilの設計、製作を主に取り組んだ。パラfoilに関する資料を集め、既存のパラfoilの採寸を行った。その後、印刷したパラfoilの設計図を元に素材の切り出し、縫い合わせて試作品のパラfoilの作成を行った。その後、体育館実験を行い、パラfoilの改善点としてパラfoil自体のサイズの問題や制御紐の絡まりが起りやすい問題を確認した。その後、パラfoilの本製作の設計図をもとに素材の切り出し、その素材の縫い合わせを行い、本製作のパラfoilの製作を行った。また、パラfoilの制御紐は絡まりが起りやすいという問題を解消する為、汎糸より適した超高分子量ポリエチレンに変更した。その後、制御紐を纏める留具や機体の着地時の衝撃を軽減する為の機体本体の下部分にあるスポンジ部分、機体の着地時の衝撃及び機体破損によって破損部分が飛び出ることを防ぐ機体を覆うカバーの製作を行った。

(※文責: 和山慧音)

6.1.5 山口直人

本プロジェクトのプロジェクトリーダーとして活動した。グループを代表して担当の先生に対して報告の取りまとめのほか、大会出場の手続きや購入する物品の取りまとめ、グループ全体への情報共有をおこなった。

前期の活動としては Python の勉強や前年度のソースコードの見直し、またラズパイの操作の習熟をおこなった。

夏休みの活動として、実際にアルゴリズムからプログラムを作成し、そのプログラムの動作実験を行った。また電装をまとめて機体に搭載できるようにした。ダムでの実験の手続きやスペースブローコンテストへの機体の発送といった雑務もこなした。

後期の活動としては大会の結果から機体の修正点を洗い出し、アルゴリズムの改良を進めた。またそのアルゴリズムをもとにプログラムの改良を行った。成果発表会で使ったスライドやポスターの作成にも携わり、また成果発表会で前半の発表を務めた。期末報告書の文章も作成し、来年度の後輩のために引き継ぎ資料の作成や成果物や資材の整頓も担った。

6.1.6 中澤一輝

コード班のグループリーダーとして活動した。プロジェクト、グループ全体への連絡、グループのスケジュール確認、タスクの進捗管理などを行った。「Slack」での担当教員との連絡や、リマインドや進捗状況の確認も重点的に行った。また、グループのスケジュール確認、タスクの進捗管理をするために「Notion」「Google カレンダー」の導入や管理などの情報システムの役割も担った。加えて、プロジェクトの予算管理や購入物の管理を行った。コード班では、Python の学習とコーディング、必要なモジュールの確認や制定、機体と電装の接続するための回路の学習・設計や、電子部品のはんだ付けなどを行った。これらのコード班の活動は夏季休暇中に重点的に行った。

また、大会スライド、中間発表や成果発表会などの発表資料に関する構成の作成及び管理も行った。内容に関しては、できる限り理解しやすく短い文章になるよう工夫した。発表に関しては前半に担当し、スペースプローブコンテスト後の活動に関して説明を行った。加えて、来年度の引継ぎがスムーズに行えるように、必要となるマイコンなどの資材の購入と、本プロジェクトが行った活動をまとめたページなどを「Notion」で作成した。

(※文責: 中澤一輝)

6.1.7 寒河江改

前期の活動ではコード班としてソースコードの解説や作成に携わった。主に Python の学習に重点を置き、より解説を進められるような状況を作った。また、中間発表のポスター製作なども担当した。後期の活動ではスペースプローブコンテストに向けたプログラミングや電子部品のはんだ付けなどの細かい作業をした。また、高大連携のプロジェクト学習紹介発表のスライドや期末発表のポスターやスライドの作成を行った。

(※文責: 寒河江改)

6.1.8 長嶋康太

コード班としてアルゴリズムの考案、プログラムの作成、電装を担当した。Raspberry pi マイコンの動作確認やモジュールの実験を行い、機体構成の検討に貢献した。また、システム全体の実装を行い、コンテストに向けて機体が動作するよう調整を行った。

前期は主に python、raspberry pi についての学習を行った。また、昨年度作成したコードを読み、どのようなアルゴリズムが作成されていたかを確認することに注力した。cansat がどのようなルールで行われるかの確認、そのルールに従ってどのようにアルゴリズムが作成されてきたかを他の大会の動画やサイトなどから下調べを行い、そこから自分たちの考えたコンセプトからどのようなアルゴリズムにするかを考えた。

夏季休暇中と後期は前期で考えたアイデアを実装していく作業を行った。電気回路の作成、コードの作成を行った。また、マイコンを用いた実験を行い、その結果から考察、コードの改良を行った。cansat の特性上、センサから取得した値から機体を制御するアルゴリズムを作成するため、コードを作成後、実際に屋外でコードが正しく動くかをテストする必要がある、その手間のかかる

コード作成とテストの作業に注力した。

(※文責: 長嶋康太)

6.2 今後の課題と展望

この項では本グループにおける今後の課題と展望について述べる。

(※文責: 今井瑞貴)

6.2.1 機体

機体班の課題について述べる。課題として以下の項目が挙げられる。

1. 減速機構の安定した展開

パラfoilが安定して展開するために以下の課題が挙げられる。

- 紐の絡まりを極限まで減らし、展開を阻害する要素の削減。
- 開口部の素材を変えるなど、空気がパラfoil内部により入りやすいような工夫。
- より厳密にパラfoilを縫い合わせ、パラfoil自体の歪みの削減。

2. スタビライザーの長さによる影響の研究

今回、スタビライザーの長短によって安定性がどれくらい影響するのかを詳しく調べることができなかった。長さが違うものを複数用意し、それぞれの飛行距離を調べるなど、適切な長さを研究することが課題として挙げられる。

3. 本体の改良と素材の選定

本体については現状大きく改良を施せる箇所はないが、3Dプリンタを使うなど素材を変えることで強度と軽量化の両立が望める。また、スポンジではなくバネやサスペンションを採用し、より衝撃を受け流すことができる本体を製作することが課題である。

今後の展望としては、課題の解決と風や投下状況に左右されない機体の検討、製作である。

(※文責: 佐藤俊介)

6.2.2 コード

次に、コード班の課題について述べる。課題として以下の項目が挙げられる。

1. 飛行制御の精度

今回作成したアルゴリズムでは、サーボモータを用いて左右の紐を引っ張ることでパラfoilを左右方向に飛行制御する必要がある。この方式で制御する場合、紐の引っ張り具合で旋回する速度が大きく変わるため、どのくらいの力で引っ張ることで安定して曲ることができるかを検証し、改良する必要がある。

パラフォイルの左右の紐を引っ張ることによる飛行制御の精度は、風などの外的要因や紐の長さが微妙に変わることによって大きく動作が変わってしまうことから、今回作成したアルゴリズムでは飛行制御の精度をそこまで高くすることはできないと考えられる。そのため、目的地に着地する精度を高くするためには、外的要因や紐の長さなどの微調整を必要としないような強い力で大きく動作させて目標地点に着地するような動力源、アルゴリズムが必要であると考えられる。

2. 安定した自動実行

起動時の GPS モジュールは動作が不安定であるため、プログラムをマイコンの起動によって自動起動しようとしたところ、GPS データの取得によってエラーが起きた。それにより、自動起動が失敗し、自動起動によるプログラムの起動からデータの取得を確認することができなかった。この問題の解決のため、GPS モジュールが起動後 GPS の取得が安定するまで待つように改良する必要がある。

3. 通信モジュール

マイコンでのプログラム検証の際、9 軸センサと GPS からデータを取得できているかを確認するために、毎回 csv ファイルに出力されているかを確認する必要があった。そのため、屋外での実験において、データを取得できているかをディスプレイにつなぎ、時間が限られた中で相当な時間と手間がかかってしまった。そのため、スムーズな実験を行うためにも PC とマイコンを遠隔で通信できるようなモジュールを取り入れる必要がある。

XBee モジュールのような通信モジュールを用いることで、マイコンから PC にリアルタイムでデータを送るようなプログラムを作成することが可能になる。マイコン側でファイルを作成してデータを保存するようなデータの取得方法では、機体の落下によってマイコンが壊れたり、sd カードが破損したりした場合に実験結果のデータを取得することができない可能性があるため、無線通信を行いリアルタイムでマイコンから PC にデータを送信し、PC 側にデータを保存することで落下後のマイコンの状況に関わらず実験のデータを保存し、確認することができる。また、cansat の大会によってはマイコンと通信することがルールとして必須条件となっていることもある。そのため、データの取得や他の cansat 大会への参加においては通信モジュールの導入が必須である。

4. 落下検知の精度

今回作成したアルゴリズムでは、9 軸センサの値から落下を検知して、飛行制御を開始する仕組みになっている。しかし、今回作成したプログラムでは、落下検知の精度が低く、少しの動作で落下を検知したと反応してしまうため、運搬中に落下を検知してしまうことが多くあった。そのため、9 軸センサの値の閾値を調整、検証、改良し、落下検知の精度を上げる必要がある。また、落下検知に関しては加速度センサーの中に落下検知モジュールが組み込まれているものが販売されているため、そのような既製品を使用して落下検知を行うことも検討するべきだと考える。

(※文責: 長嶋康太)

6.2.3 大会

大会についての結果や課題を挙げていく。初めに、大会結果についてだが本チームは落下地点の正確性が全体 4 位であった。内容としては、パラフォイルの紐の一部が制御機構に引っかかり、パラフォイルが完全に展開しない形で垂直落下してしまった。しかし、目標地点の近くに落ちたため着地時点でのポイントは高かった。また、パラフォイルの部分的な展開による減速と衝撃吸収機構がうまく作用したおかげで、機体が自由落下したにもかかわらず、機体内部の RaspberryPi や電源、センサ類は壊れることなく無事であり、機体本体の損傷も最小限にとどまっていた。大会後の実験の際に、動作するかどうかを確認したところ、大会前と同様の動きを示していた。

次に課題であるが、

- パラフォイルの展開機構に紐が引っかかりを防ぐための改良
- パラフォイルの展開をより安定させるための改良
- データ取得・保存を確実にを行うためのコードの改良
- 大会で確認できなかったサーボモータの動作の改良

等が挙げられる。そして、パラフォイルを確実に展開させる事が出来るという点が一回きりである本番では最も難しく必要不可欠であるということが分かった。今後はこれらの課題をクリアし、来年の同大会に出場するかチームメンバーと検討中である。

(※文責: 寒河江改)

付録 A プログラムリスト

ソースコード A.1 main.py

```
1 import sys
2 sys.path.append('/home/funsat/.local/lib/python3.9/site-packages')
3 import csv
4 import datetime
5 import time
6 import pytz
7 import os
8 import math
9 import GPS
10 import servo
11 import Controller
12 import bmx055
13 def main():
14
15 #インスタンスの作成
16     con = Controller.Controller()
17     l_servo = servo.servo(12)
18     r_servo = servo.servo(18)
19     gps = GPS.GPS()
20     bmx = bmx055.bmx055()
21     now_time = datetime.datetime.now()
22     filename = '/home/funsat/Desktop/FanSat2022.10.26/testtestdata'
23     for i in range(100):
24         filename_str=filename+'_'+str(i)+'.csv'
25         if(os.path.exists(filename_str)==False):
26             with open(filename_str, 'a',newline="") as f:
27                 writer = csv.writer(f)
28                 writer.writerow(['RUN_START'])
29                 writer.writerow(['time','latitude','longitude','acc/let
30                                     '])
31             break
32     gps.start()
33     count = 0
34 #GPS 確認
35     l_servo.write(180)
36     r_servo.write(0)
37     while True:
38         gps_x = gps.gps.longitude[0]
39         gps_y = gps.gps.latitude[0]
40         with open(filename_str,'a',newline="") as f:
41             writer = csv.writer(f)
42             with open(filename_str,'a',newline="") as f:
```

```

42         writer = csv.writer(f)
43         writer.writerow(['fall_junnbi',gps.gps.timestamp[0],gps.
            gps.timestamp[1],gps.gps.timestamp[2], 'latitude',gps.
            gps.latitude[0], 'longitude',gps.gps.longitude[0]])
44     if(gps_x!=0 or gps_y!=0):
45         l_servo.write(90)
46         r_servo.write(90)
47         time.sleep(3)
48         break
49     time.sleep(1)
50 #落下検知
51     while True:
52         acc = bmx.acc_value() #加速度の取得
53         #GPS のデータを csv に出力する.gps.data, gps.timestamp, gps.latitude
            , gps.longitude を使用(検証する)
54         #accno0,1,2=x,y,z
55         with open(filename_str,'a',newline="") as f:
56             writer = csv.writer(f)
57             writer.writerow(['fall_st',gps.gps.timestamp[0],gps.gps.
                timestamp[1],gps.gps.timestamp[2], 'latitude',gps.gps.
                latitude[0], 'longitude',gps.gps.longitude[0], 'acc',acc
                [0],acc[1],acc[2]])
58         if(-15 < acc[0] and acc[0] < 15 and -15 < acc[1] and acc[1] < 15
            and -15 < acc[2] and acc[2] < 15):#全部の値が規定値以下になれば
            開始
59             count += 1
60         else:
61             count = 0
62         if(count >= 15):
63             break
64         time.sleep(0.05)
65
66 #制御開始
67
68     gps_x = gps.gps.longitude[0]
69     gps_y = gps.gps.latitude[0]
70     acc = bmx.acc_value()
71     with open(filename_str,'a',newline="") as f:
72         writer = csv.writer(f)
73         writer.writerow(['fall_end',gps.gps.timestamp[0],gps.gps.
            timestamp[1],gps.gps.timestamp[2], 'latitude',gps.gps.
            latitude[0], 'longitude',gps.gps.longitude[0], 'acc',acc
            [0],acc[1],acc[2]])
74     old_x, old_y = gps_x, gps_y
75     i = 0
76
77     while True:
78         gps_x = gps.gps.longitude[0]

```

```

79     gps_y = gps.gps.latitude[0]
80
81     dir_x, dir_y = con.set_goal(gps_x, gps_y)
82     let = con.dis_gps(old_x, old_y, gps_x, gps_y, dir_x, dir_y)
83     with open(filename_str,'a',newline="") as f:
84         writer = csv.writer(f)
85         writer.writerow(['control',gps.gps.timestamp[0],gps.gps.
            timestamp[1],gps.gps.timestamp[2], 'latitude',gps.gps.
            latitude[0], 'longitude',gps.gps.longitude[0], 'acc',acc
            [0],acc[1],acc[2],let, 'dir',dir_x,dir_y])
86
87 #サーボモータを動かす
88     if let > 0:
89         l_servo.write(180)
90         r_servo.write(180)
91         print("hidari")
92     elif let == 0:
93         l_servo.write(0)
94         r_servo.write(180)
95         print("mae")
96     else:
97         l_servo.write(0)
98         r_servo.write(0)
99         print("migi")
100    print(old_x, old_y, gps_x, gps_y, dir_x, dir_y)
101
102    old_x, old_y = gps_x, gps_y
103    time.sleep(1.0)
104    i += 1
105    if(i > 180):
106        break
107    del l_servo
108    del r_servo
109    with open(filename_str, 'a',newline="") as f:
110        writer = csv.writer(f)
111        writer.writerow(['RUN_END'])
112    sys.exit()
113
114 if __name__ == '__main__':
115     main()

```

ソースコード A.2 bmx055.py

```

1 from smbus import SMBus
2 import time
3 import math
4
5 # I2C
6 class bmx055:

```

```

 7  ACCL_ADDR = 0x19
 8  ACCL_R_ADDR = 0x02
 9  GYRO_ADDR = 0x69
10  GYRO_R_ADDR = 0x02
11  MAG_ADDR = 0x13
12  MAG_R_ADDR = 0x42
13  i2c = SMBus(1)
14  def __init__(self):
15      # acc_data_setup : 加速度の値をセットアップ
16      self.i2c.write_byte_data(self.ACCL_ADDR, 0x0F, 0x03)
17      self.i2c.write_byte_data(self.ACCL_ADDR, 0x10, 0x08)
18      self.i2c.write_byte_data(self.ACCL_ADDR, 0x11, 0x00)
19      time.sleep(0.5)
20      # gyr_data_setup : ジャイロ値をセットアップ
21      """
22      i2c.write_byte_data(GYRO_ADDR, 0x0F, 0x04)
23      i2c.write_byte_data(GYRO_ADDR, 0x10, 0x07)
24      i2c.write_byte_data(GYRO_ADDR, 0x11, 0x00)
25      time.sleep(0.5)
26      # mag_data_setup : 地磁気値をセットアップ
27      #data = i2c.read_byte_data(MAG_ADDR, 0x4B)
28      if(data == 0):
29          i2c.write_byte_data(MAG_ADDR, 0x4B, 0x83)
30          time.sleep(0.5)
31      i2c.write_byte_data(MAG_ADDR, 0x4B, 0x01)
32      i2c.write_byte_data(MAG_ADDR, 0x4C, 0x00)
33      i2c.write_byte_data(MAG_ADDR, 0x4E, 0x84)
34      i2c.write_byte_data(MAG_ADDR, 0x51, 0x04)
35      i2c.write_byte_data(MAG_ADDR, 0x52, 0x16)
36      time.sleep(0.5)
37      """
38
39
40  def acc_value(self):
41      data = [0, 0, 0, 0, 0, 0]
42      acc_data = [0.0, 0.0, 0.0]
43      try:
44          for i in range(6):
45              data[i] = self.i2c.read_byte_data(self.ACCL_ADDR, self.
46                  ACCL_R_ADDR + i)
47          for i in range(3):
48              acc_data[i] = ((data[2*i + 1] * 256) + int(data[2*i] & 0xF0))
49                  / 16
50              if acc_data[i] > 2047:
51                  acc_data[i] -= 4096
52                  acc_data[i] *= 0.0098
53      except IOError as e:
54          print("I/O error({0}): {1}".format(e.errno, e.strerror))

```

Flying Autonomous Robot Project

```
53     return acc_data
54     """
55     def gyro_value(self):
56         data = [0, 0, 0, 0, 0, 0]
57         gyro_data = [0.0, 0.0, 0.0]
58         try:
59             for i in range(6):
60                 data[i] = i2c.read_byte_data(GYRO_ADDR, GYRO_R_ADDR + i)
61             for i in range(3):
62                 gyro_data[i] = (data[2*i + 1] * 256) + data[2*i]
63                 if gyro_data[i] > 32767:
64                     gyro_data[i] -= 65536
65                     gyro_data[i] *= 0.0038
66         except IOError as e:
67             print("I/O error({0}): {1}".format(e.errno, e.strerror))
68         return gyro_data
69     def mag_value(self):
70         data = [0, 0, 0, 0, 0, 0, 0, 0]
71         mag_data = [0.0, 0.0, 0.0]
72         try:
73             for i in range(8):
74                 data[i] = i2c.read_byte_data(MAG_ADDR, MAG_R_ADDR + i)
75             for i in range(3):
76                 if i != 2:
77                     mag_data[i] = ((data[2*i + 1] * 256) + (data[2*i] & 0xF8
78                                     )) / 8
79                     if mag_data[i] > 4095:
80                         mag_data[i] -= 8192
81                 else:
82                     mag_data[i] = ((data[2*i + 1] * 256) + (data[2*i] & 0xFE
83                                     )) / 2
84                     if mag_data[i] > 16383:
85                         mag_data[i] -= 32768
86         except IOError as e:
87             print("I/O error({0}): {1}".format(e.errno, e.strerror))
88         return mag_data
89     """
90     if __name__ == "__main__":
91         bmx = bmx055()
92         time.sleep(0.1)
93         while True:
94             acc = bmx.acc_value()
95             #gyro= bmx.gyro_value()
96             #mag = bmx.mag_value()
97             print("Acc1 -> x:{}, y:{}, z: {}".format(acc[0], acc[1], acc[2]))
98             #print("Gyro -> x:{}, y:{}, z: {}".format(gyro[0], gyro[1], gyro
99                 [2]))
```

```

98     #print("Mag -> x:{}, y:{}, z: {}".format(mag[0], mag[1], mag[2]))
99     print("\n")
100    time.sleep(0.1)

```

ソースコード A.3 Controller.py

```

1  import math
2
3  #返り値は左右どちらを引けばいいか
4  #
5
6  class Controller():
7
8      #最も近いゴール座標を返す。
9      def set_goal(self, gps_x, gps_y):
10         position = [
11             [41.862132,140.780907],
12             [41.842736,140.740163],
13             [41.799098,140.803688],
14             [41.764100,140.722771]
15         ]
16
17         n_dir = abs(position[0][0] - gps_x) + abs(position[0][1] - gps_y)
18         e_dir = abs(position[1][0] - gps_x) + abs(position[1][1] - gps_y)
19         s_dir = abs(position[2][0] - gps_x) + abs(position[2][1] - gps_y)
20         w_dir = abs(position[3][0] - gps_x) + abs(position[3][1] - gps_y)
21
22         go_dir = min(n_dir, e_dir, s_dir, w_dir)
23         if go_dir == n_dir:
24             return position[0][0], position[0][1]
25         elif go_dir == e_dir:
26             return position[1][0], position[1][1]
27         elif go_dir == s_dir:
28             return position[2][0], position[2][1]
29         else:
30             return position[3][0], position[3][1]
31
32
33     #ひとつ前のgps と今の gps、今の gps と goal、それぞれの傾きから右左を出す
34     def dis_gps(self, old_x, old_y, now_x, now_y, dir_x, dir_y):
35         dis_now_x = now_x - old_x
36         dis_now_y = now_y - old_y
37         dis_goal_x = dir_x - now_x
38         dis_goal_y = dir_y - now_y
39         asin_now = math.degrees(math.atan2(dis_now_y, dis_now_x))#傾きを
40             角度にする
41         asin_goal = math.degrees(math.atan2(dis_goal_y, dis_goal_x))#傾きを
42             角度にする
43         let = asin_goal - asin_now #ゴールが左右どちらにあるかを出す。この値が

```

正の値であればgoal は左にある。

```

42     return let
43
44 if __name__ == '__main__':
45     con = Controller()
46     goal = con.set_goal(41.833695,140.770596)
47     print(goal)
48     print(con.dis_gps(41.833695,140.770596,41.833699,140.770599,goal[0],
        goal[1]))

```

ソースコード A.4 GPS.py

```

1 import serial
2 import micropyGPS
3 import threading
4 import time
5
6 class GPS(threading.Thread):
7     def __init__(self):
8         super(GPS, self).__init__()
9         self.gps = micropyGPS.MicropyGPS(9, 'dd') #
            MicroGPS オブジェクトを生成する。
10                                     # 引数はタイムゾーンの時差と出力フォーマット
11
12 def run(self):#使うときはrun()じゃなくてstart()
13     s = serial.Serial('/dev/serial0', 9600, timeout=10)
14     s.readline() # 最初の1行は中途半端なデータが読めることがあるので、捨てる
15     while True:
16         sentence = s.readline().decode('utf-8') #
            GPS データを読み、文字列に変換する
17         #if len(sentence) > 0:
18         if sentence[0] != '$': # 先頭が'$'でなければ捨てる
19             continue
20         for x in sentence: # 読んだ文字列を解析して
            GPS オブジェクトにデータを追加、更新する
21             self.gps.update(x)
22
23 if __name__ == '__main__':
24     Gps = GPS()
25     Gps.start()
26     while True:
27         if Gps.gps.clean_sentences > 20:
28             h = Gps.gps.timestamp[0] if Gps.gps.timestamp[0] < 24 else
                Gps.gps.timestamp[0] - 24
29             print('%2d:%02d:%04.1f' % (h, Gps.gps.timestamp[1], Gps.gps.
                timestamp[2]))
30             x = Gps.gps.latitude[0]
31             y = Gps.gps.longitude[0]

```

```
32         print('緯度経度: %2.8f, %2.8f' % (x, y))
33         time.sleep(3.0)
```

ソースコード A.5 servo.py

```
1 import time
2 import RPi.GPIO as GPIO
3 import sys
4
5 class servo:
6     def __init__(self, pin):
7         GPIO.setwarnings(False)
8         GPIO.setmode(GPIO.BCM)
9         GPIO.setup(pin, GPIO.OUT)
10        self.servo = GPIO.PWM(pin, 50)
11        self.servo.start(0)
12
13    def __del__(self):
14        self.servo.stop()
15        GPIO.cleanup()
16
17    def write(self, angle): #0 ~ 180を受け取った時にその分だけ回す
18        if angle < 0:
19            angle = 0
20        elif angle > 180:
21            angle = 180
22        dc = (angle/180)*9.5+2.5
23        self.servo.ChangeDutyCycle(dc)
24        print(angle)
25
26 if __name__ == '__main__':
27     l_servo = servo(12)
28     r_servo = servo(18)
29     for i in range(1):
30         l_servo.write(180)
31         r_servo.write(180)
32         time.sleep(3)
33         l_servo.write(0)
34         r_servo.write(180)
35         time.sleep(3)
36         l_servo.write(0)
37         r_servo.write(0)
38         time.sleep(3)
39         l_servo.write(90)
40         r_servo.write(90)
41         time.sleep(3)
42     del l_servo
43     del r_servo
44     print('fin')
```

参考文献

- [1] CanSat 計画,”日米大学による手作り小型衛星への挑戦”,
https://www.jstage.jst.go.jp/article/kjsass/48/562/48_589/_pdf/-char/ja , (参照 2022-07-14)
- [2] 東京工業大学,”東工大ニュース”,
<https://www.titech.ac.jp/news/2019/043242>, (参照 2022-07-14)
- [3] 東京大学,”ARLISS2012 報告書”,
http://www.unisec.jp/history/arliss2012/report/NASU_rep.pdf, (参照 2022-7-15)
- [4] 植松電機株式会社. “スペースプローブコンテスト 2022”
<https://spc.uematsudenki.com/wp/>, (参照 2022-07-14)
- [5] Yahoo!ショッピング. “スタントパワーカイトインフレータブル耐久性サーフィンパラフォイル
レインボーパラシュート 2m”
<https://store.shopping.yahoo.co.jp/stk-shop/54051842.html>, (参照 2022-12-25)
- [6] 岐阜県立恵那高等学校. “CanSat(模擬人工衛星) 開発におけるパラシュート研究と本体開発”
<https://school.gifu-net.ed.jp/ena-hs/ssh/H22ssh/sc3/31003.pdf> , (参照 2022-07-14)