

ACTS報告書

提出日：2021年 12月 26日

- チーム情報

CanSatチーム名	電気通信大学 Carryber
CanSatチーム 代表者情報	前川 裕介 yusuke_maekawa_jr@cas.lab.uec.ac.jp, 080-9182-0912
UNISEC団体名	電気通信大学 高玉研究室
UNISEC団体 学生代表	白石 洋輝 hirowhite@cas.lab.uec.ac.jp, 090-1208-0322
責任教員	高玉 圭樹 keiki@inf.uec.ac.jp, 042-443-5808

- メンバー

役割	名前(学年, 経験年数)
リーダー兼ハード班	前川 裕介 (M1, 経験2年目)
ハード班	福本 有季子 (M2, 経験3年目)
ハード班	中理 怡誠 (B4, 経験1年目)
ハード班	原 亜美佳 (B1, 経験1年目)
ハード班	前川 恭介 (B1, 経験1年目)
ソフト班	千住 太希 (M2, 経験2年目)
ソフト班	谷津 直弥 (B4, 経験1年目)
ソフト班	堀越 健 (B1, 経験1年目)
回路班	嘉村 魁人 (B4, 経験1年目)
回路班	平野 晶也 (B1, 経験1年目)

- CanSatの製作目的・大会参加理由

本研究室では, 研究活動とは別に, エンジニアとしての開発技術の向上や課題解決能力の向上, チームマネジメントなどの教育を目的としてCanSatの開発をし, その有効性検証のため例年ARLISSに参加をしている.

目次

ミッションについて	5
ミッションステートメント(ミッションの意義と目的)	5
ミッションシーケンス	5
サクセスクライテリア	7
要求項目の設定	7
システム要求(安全確保, レギュレーションのための要求)	8
ミッション要求(ミッションを実現するにあたり要求される性能)	9
システム仕様	10
CanSat外観	10
CanSatの内観・機構	11
システム図(CanSat搭載計器仕様一覧)	19
アルゴリズム	24
試験項目設定	32
実施試験内容	33
システム要求を満たすための試験内容	33
質量試験	33
収納放出試験	34
通信機器電源 ON/OFF 試験	35
長距離通信試験	36
落下試験	37
開傘衝撃試験	40
準静的荷重試験	42
通信周波数変更試験	44
ナビゲーション試験	45
End-to-End 試験	46
制御履歴レポート作成試験	47
ミッション要求を満たすための試験内容	52
着地衝撃試験	52
電力耐久試験	54
反転・横転復帰試験	54
物体検知および運搬試験	56

ゴール検知試験	57
工程管理, ガントチャート	58
責任教員による自己安全審査結果のまとめ	62
安全基準審査	62
責任教員所感	63

第1章 ミッションについて

1. ミッションステートメント(ミッションの意義と目的)

タイトル: 物体検出×把持アーム機構による自律運搬CanSat

将来の惑星探査において、月面基地の建設など惑星上での物資の運搬はかかせない作業である。そこで、本ミッションは自律的に物資の置かれた目的地点に向かい、人の手を介すことなく物資を他の地点へ運搬可能なCanSatの構築を目指す。具体的にはカメラを用いた物体検出と、運搬物を把持するアームを取り付け、「手(アーム)を持ち、ものを認識できる」CanSatを開発する。特に、物体検出ではサイズの小さな物体に対しても検知できるモデルおよび接近動作の構築、アーム機構では物体の形状に依存しない把持性能を持つ機構を開発する。また運搬中に落とす等の物体のロスト対策をすることで確実な物資運搬が可能なCanSatを目指す。

2. ミッションシーケンス

ミッション全体の流れは以下の通りである(図1.1)。なお、具体的なアルゴリズムについては、4.4章のアルゴリズムにて説明する。

1. [待機状態]: CanSatはキャリアに収納され、上空で放出されるまで待機状態となる。
2. [落下状態]: 9軸センサ、気圧センサを用いてCanSatが降下中か着地したかの状態判定を開始。
3. [パラシュート分離状態]: 着地判定後、CanSatはパラシュートを切り離す動作を開始。サーボモータを複数回動かすことによりパラシュートを切り離す(パラシュート切り離し機構については4.3章を参照)。
4. [ナビゲーション状態]: GPS座標を用いるナビゲーションを行う。航法はCansatの現在地点とゴール地点から目標方位に向かってPID制御で走行する。
5. [物体検知状態]: 画像を用いる物体検知を行う。航法は画像中心に検知対象のバウンディングボックスの中心が配置されるように接近する。
6. [運搬設置状態 or 運搬物把持状態]: Cansat前方のアームによって運搬物の運搬・積み下ろしを行う。

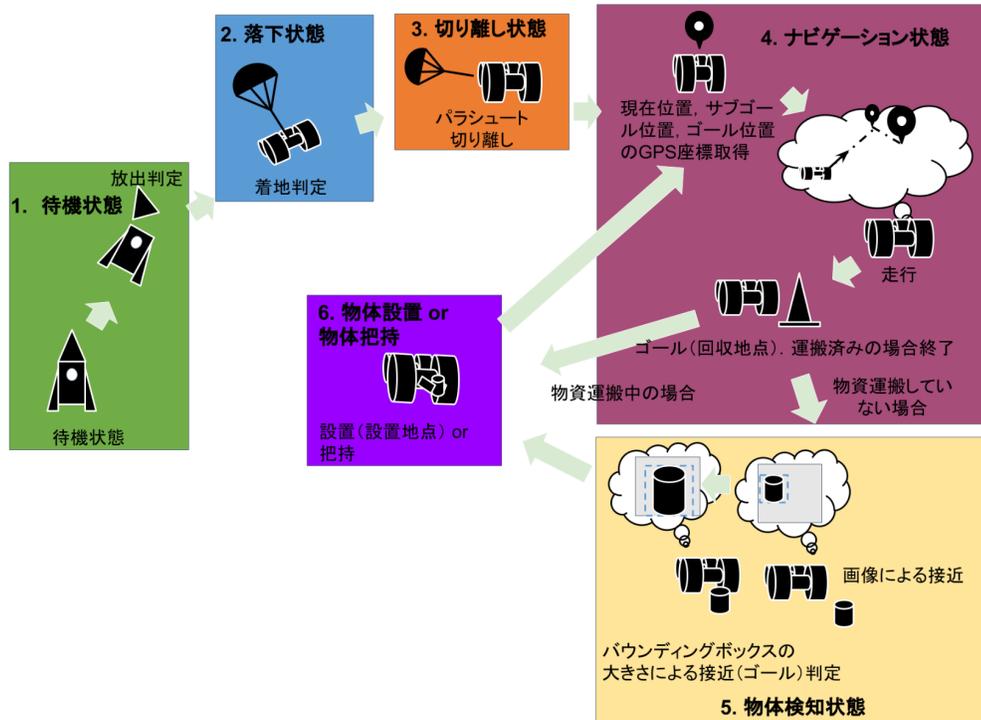


図1.1 ミッション全体の流れ

第2章 サクセスクライテリア

ミニマムサクセス	着地後, パラシュートを切り離し走行開始 ゴール地点付近に置かれた物資に画像処理による物体の検知
フルサクセス	どちらか達成で△, 両方達成で○ ①物資を収集・運搬し, 運搬先で物資を放出 ②ゴール地点へ帰還, 半径10cm以内に到達

また本ミッションでは以下のエクストラサクセスを設定する.

エクストラサクセス	物資を運搬途中で落としたとしても, ロスト判定により再度回収して運搬完遂
-----------	--------------------------------------

第3章 要求項目の設定

1. システム要求(安全確保, レギュレーションのための要求)

要求番号	自己審査項目
	ACTS安全基準
R1	質量と容積 がレギュレーションを満たすことが確認できている
R2	ロスト対策 を実施しており, 有効性が試験で確認できている (例: 地上局にダウンリンクする場合, ACTSで十分な通信距離が実現できると推測できる根拠が明確に示されていること.)
R3	地表近くで危険な速度で落下させないための 減速機構 を有し, その性能が試験で確認できている
R4	打ち上げ時の 準静的荷重 によって, 安全基準を充足するための機能が損なわれないことが試験で確認できている
R5	打ち上げ時の 振動荷重 によって, 安全基準を充足するための機能が損なわれないことが試験で確認できている
R6	分離時の 衝撃荷重 によって, 安全基準を充足するための機能が損なわれないことが試験で確認できている
R7	打ち上げ時の 無線機の電源OFF の規定を遵守できていることが確認できている (FCC認証かつ100mW以下の機器はOFFしなくて良い. また, スマートフォンを用いる場合はFCC認証かつソフトウェアまたはハードウェアスイッチでoffにできること(2017年追加))
R8	無線のチャンネル調整に応じる意思があり, また実際に調整ができることを確認できている
R9	R1-R8の充足を確認した設計の機体によって, ロケットへの装填から打ち上げ後の回収までを模擬したEnd-to-end試験を実施できている, 今後, 安全性に関わる大幅な設計変更はない
R10	CanSatの収納・投下準備が5分以内できている
	カムバックコンペティションルールの充足
R11	ミッション時に人間が介在しない 自律制御 を実施することが確認できている (注: 2014年のレギュレーション改定以降, 地上局設備に計算機能を持たせてアップリンクしても良い)
R12	ミッション後, 規定された 制御履歴レポートを運営者へ提出 する準備ができている (以下の根拠の項に制御履歴レポートの例を添付すること. ダミーデータを使用しても良い)

2. ミッション要求(ミッションを実現するにあたり要求される性能)

要求番号	自己審査項目(ミッション実現要求項目)
M1	着地時の衝撃荷重によって、ミッションを実現するための機能が損なわれていないことが試験で確認できている
M2	着地後にパラシュートが切り離せることを試験で確認できている
M3	競技時間を十分に走行可能であることを確認できている
M4	CanSatが反転・横転した時に通常の走行姿勢に復帰できることを試験で確認できている
M5	運搬機構を用いて物資の運搬・積み下ろしが可能であることを試験で確認できている
M6	カメラを用いて物資の検知が可能であることを試験で確認できている
M7	カメラを用いてゴール検知が可能であることを試験で確認できている
M8	カメラを用いて運搬物のロスト判定が可能であることを試験で確認できている

第4章 システム仕様

1. CanSat設計図

(1)ハード系

以下に公開可能な各モジュールの設計図を示す.

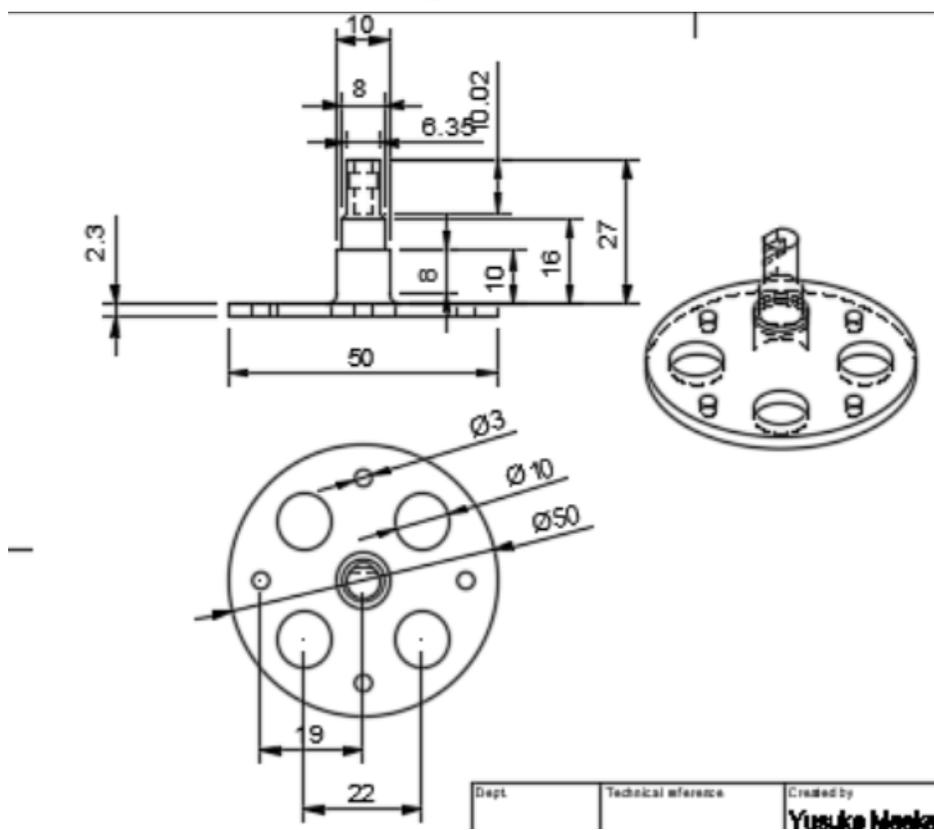


図4.1 車軸の図面

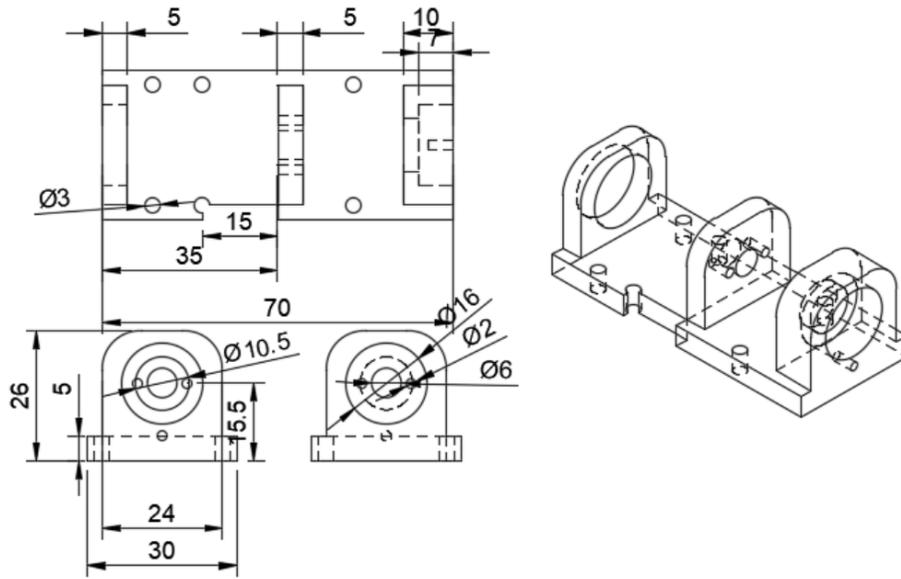


図4.2 モータホルダーの図面

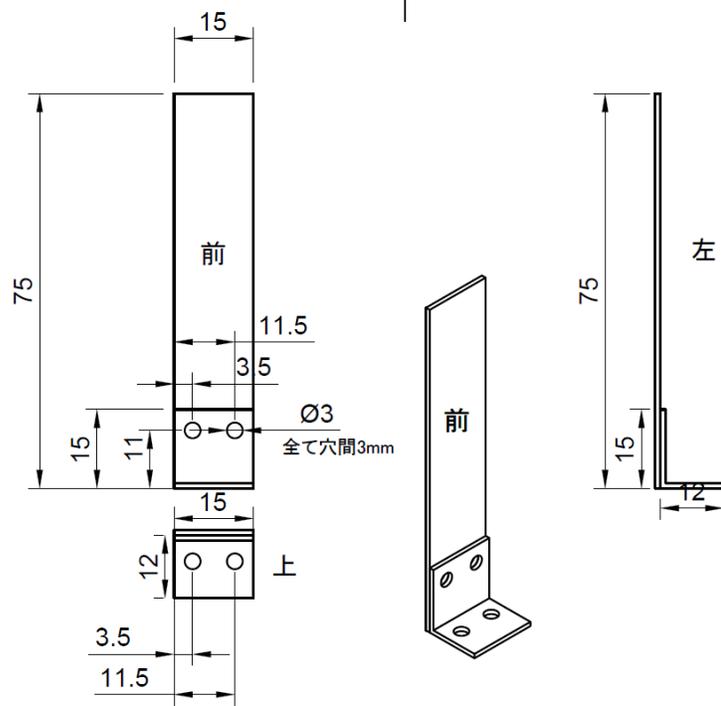


図4.3 運搬機構の図面1

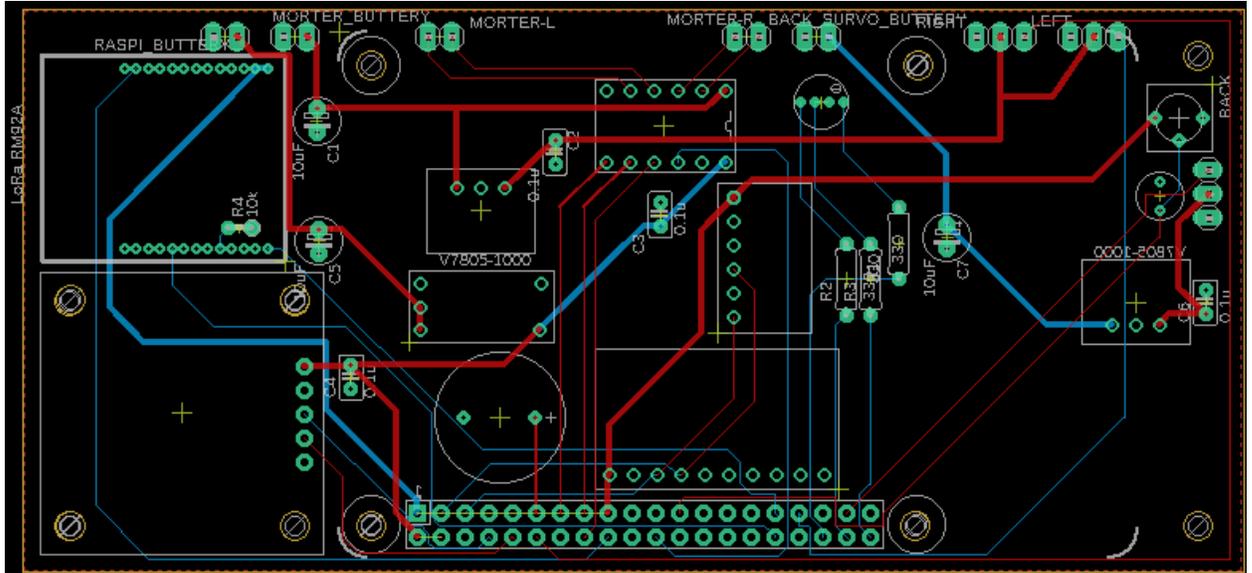


図4.6 回路の実際のレイアウト

2. CanSat外観

CanSatの全体像および後方からの様子を以下の図4.7, 図4.8に示す。本CanSatは2輪走行で物体を運搬するためのアーム機構(以降: 運搬機構)を有する。運搬用アーム機構は2つのサーボモータで構成されており, 具体的な構造や, 動作方法等の詳細は本章の2節で述べる。

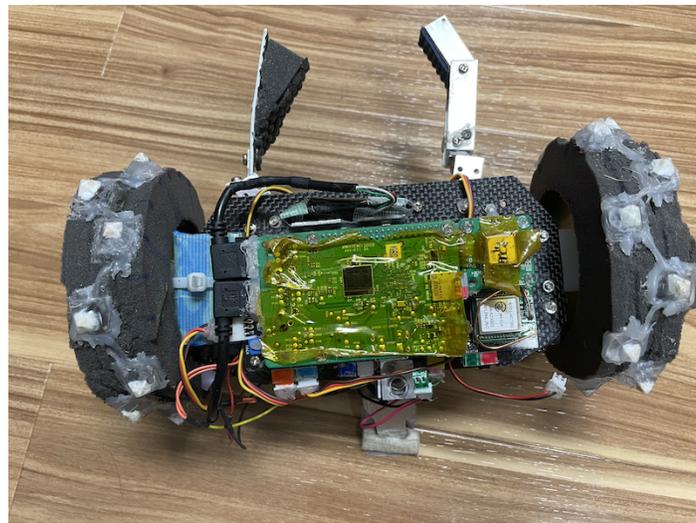


図4.7 CanSatの全体像



図4.8 後方からの様子

次に、CanSatの大きさを以下の表4.1に、計測の際の様子を図4.9、図4.10に示す。

表4.1 CanSatの全長と高さ

全長[mm]	240
高さ[mm]	130(タイヤのスパイクを含む)

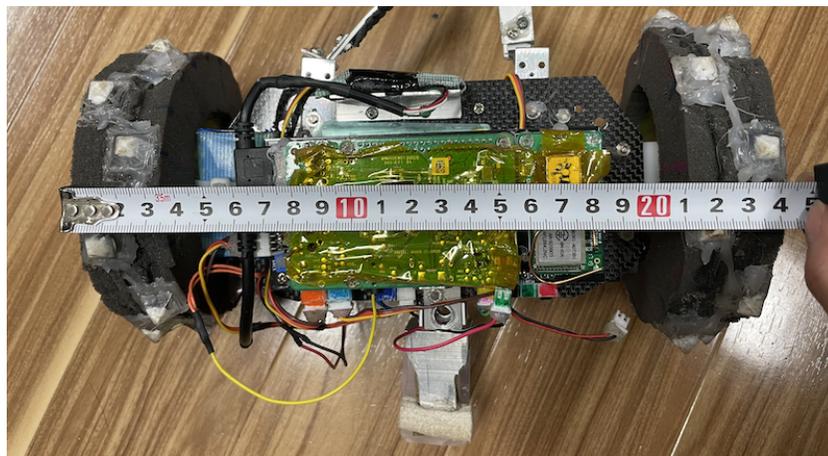


図4.9 CanSatの全長



図4.10 CanSatの高さ

3. CanSatの内観・機構

● 基本構造

- シャーシ上部(図4.11)
 - 回路基板(具体的に搭載されたセンサ類については3章で述べる)
 - カメラ
 - 回路用電源(リチウムイオンポリマー電池)
 - スタビライザーサーボモータ用電源(リチウムイオンポリマー電池)

- シャーシ下部(図4.12)
 - モータ2個
 - モータ用電源(エナジャイザー)
 - USBアクセラレータ(Raspberry Pi の推論処理高速化のための外付けデバイス)

- CanSat前方(図4.13)
 - 物体を運搬するための運搬機構

- CanSat後方(図4.14)
 - 姿勢制御およびパラシュート切り離しのためのスタビライザーサーボモータ

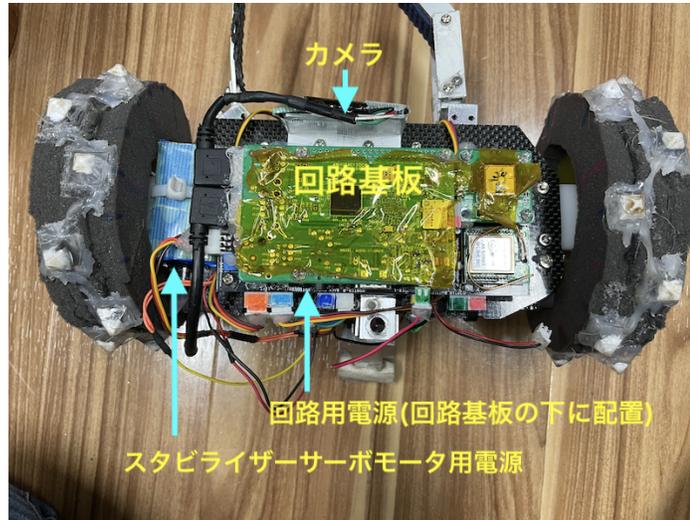


図4.11 CanSatシャーシ上部

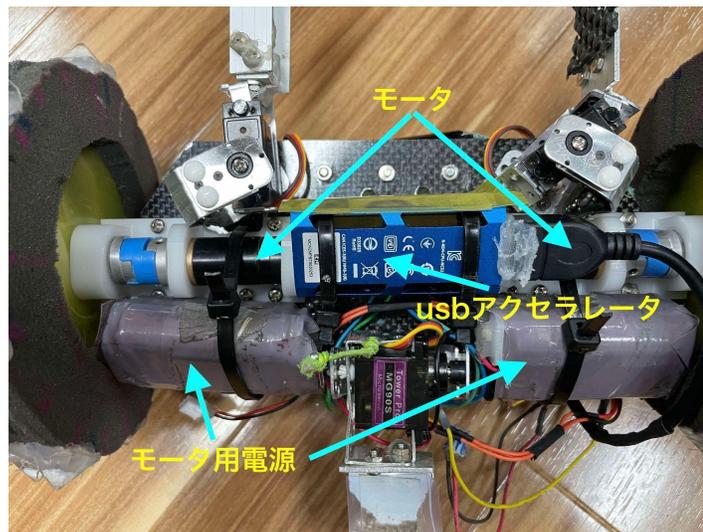


図4.12 CanSatシャーシ下部

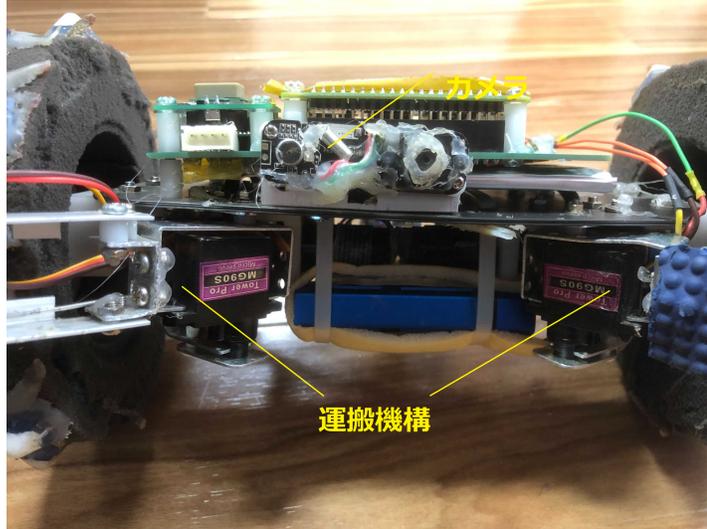


図4.13 CanSat前方

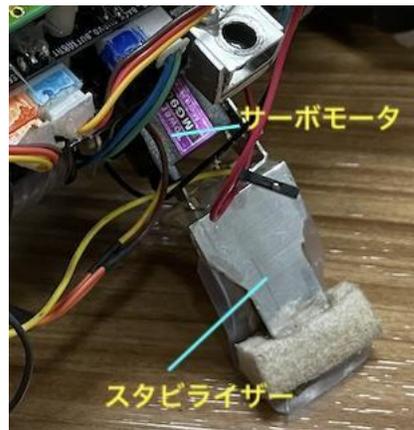


図4.14 CanSat後方

以下より本CanSatの特徴的な機構である、

- パラシュートを切り離すための機構(パラシュート切り離し機構)
- 運搬機構
- 着地衝撃から運搬機構を守るプロテクター
- スタビライザーサーボ
- タイヤのアタッチメント

について説明する。

- パラシュート切り離し機構

CanSatの着地後、減速機構であるパラシュートを切り離すための機構について説明する。

以下の図4.15は、パラシュート切り離し機構を示している。パラシュートを切り離すために、CanSat本体の後方に取り付けられているサーボモータを活用する。

パラシュート切り離す様子: <https://youtu.be/bWGyqtMXfDo>

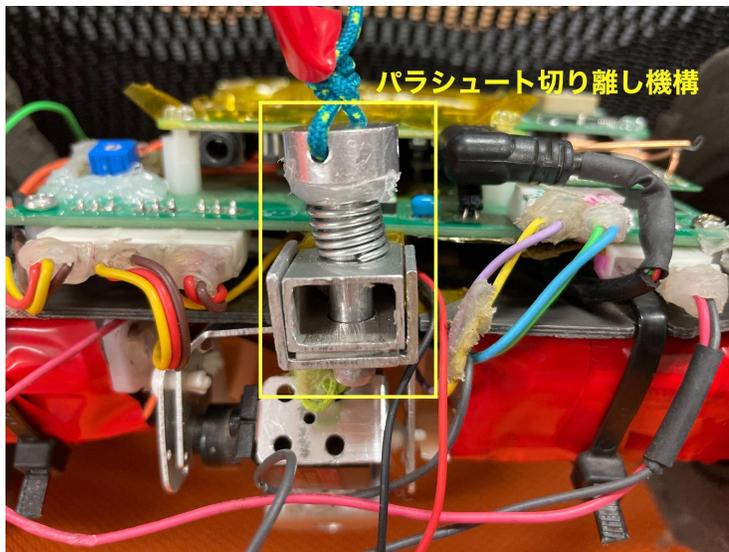


図4.15 パラシュート切り離し機構

パラシュート切り離し機構の模式図を図4.16に示す。この機構は、灰色で示されるパラシュートピン、濃い灰色で示されるスプリング、直方体アルミ部品、橙色で示されるクギで構成されている。パラシュートの紐(図中の青色で示される部分)はパラシュートピンにつながれており、パラシュートピンは直方体アルミパーツに空けた穴に挿入されている。また、パラシュートピンにはスプリングが取り付けられている。このため、直方体アルミパーツに挿入した際にスプリングが縮むことによって反発力が発生し、パラシュートピンには直方体アルミパーツから抜ける方向に力がかかるが、クギをパラシュートピンの先端の貫通した穴に挿すことによってこれを阻止している。クギについての紐を介してスタビライザーサーボモータに取り付けられている。

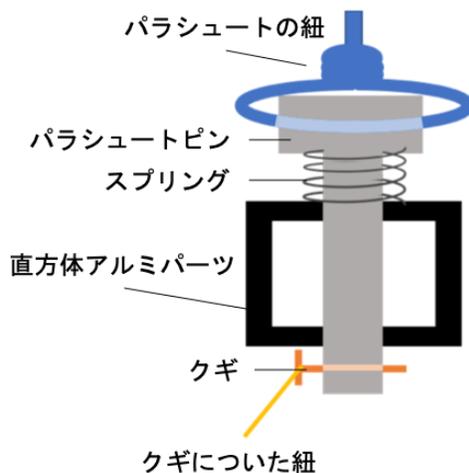


図4.16 パラシュート切り離し機構の模式図(切り離し前)

パラシュート切り離し時、スタビライザーサーボモータが動作し、クギについての紐を引くことによって、パラシュートピンに挿されていたクギを引き抜く(図4.17)。

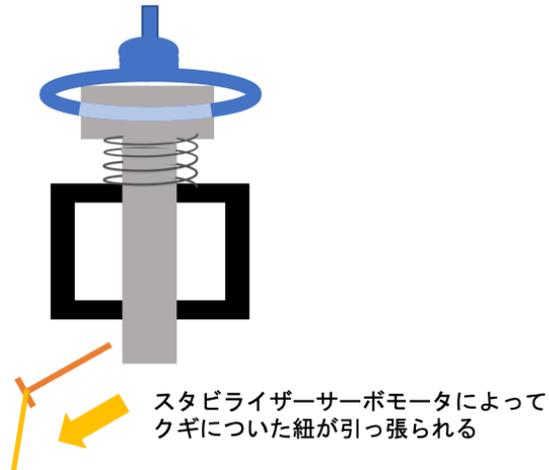


図4.17 パラシュート切り離し機構の模式図(切り離し途中)

クギによるストッパーが外れることによりスプリングの反発力が作用し、パラシュートが取り付けてあるパラシュートピンが直方体アルミ部品パーツから外れる(図4.18).
この際、スプリングの反発力によってパラシュートピンはCanSat本体から離れた位置まで飛ばされる.

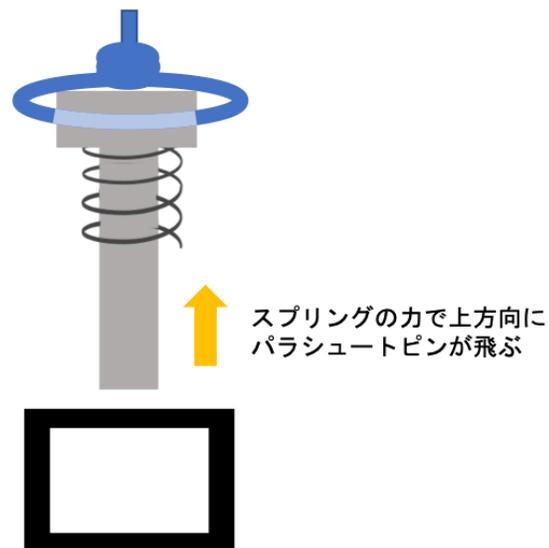


図4.18 パラシュート切り離し機構(切り離し後)

● 運搬機構

○ 運搬動作

図4.19に運搬機構の全体像を示す。運搬機構は2つのサーボモータとアームで構成され、2つのアームを閉じることで運搬物を把持し、アームを開くことで放出する。運搬物を挟む部分(アームの内側)には滑り止めを取り付け、走行時の振動から運搬物が滑り落ちることを防ぐ。

図4.19の右側のアームは折りたたみ式になっておりトーションパネの力によって開く。また左側のアームは直線であり、内側にスポンジが取り付けられている。

運搬物把持の例として、立方体を挟んだ様子と円柱を挟んだ様子を以下の図4.20に示す。

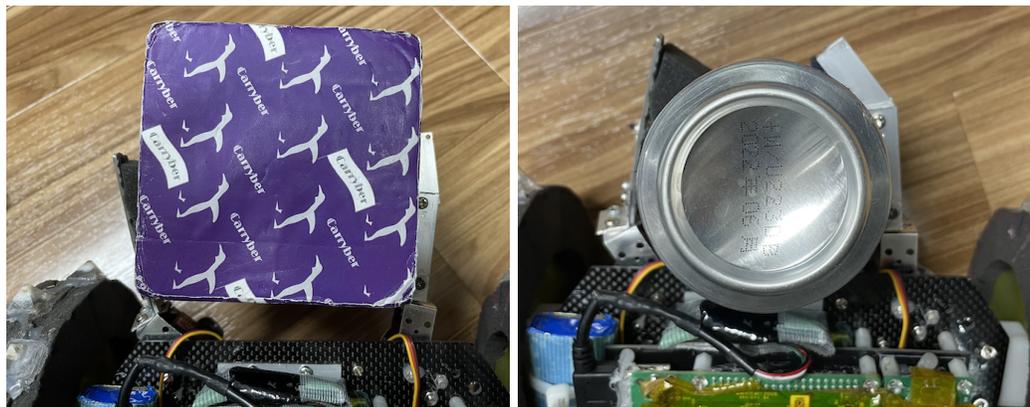
立方体の時は、右アームは立方体の面に沿って真っ直ぐになる。また左アームのスポンジの反発によって立方体の把持力をサポートする。

円柱の時は、右アームは円柱のカーブ面に沿い、左アームのスポンジの反発によって把持力をサポートする。

このように1つの運搬物の形状に寄らず、把持できる運搬機構となっている。



図4.19 運搬機構



立方体

円柱

図4.20 物体を挟んでいる様子

○ 運搬機構の収納・展開方法

図4.21, 4.22はアームの折り畳む様子とキャリアに収納された状態を示す。収納時は折りたたみアームの上部分を内側にたたみ(図4.21), その上からもう一方のアームを閉じる(図4.22)。

展開時は、直線のアームを開いたのち、折り畳みアームも開くと同時にトーションバネの力で折り畳まれた部分を自動的に開く。

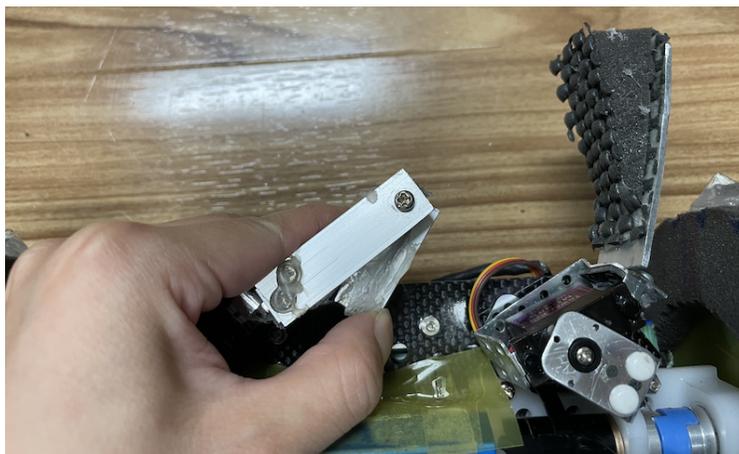


図4.21 運搬機構(折り畳んでいる様子)
※機体を裏返して撮影



図4.22 運搬機構(キャリア内の収納状態)
※機体を裏返して撮影

- プロテクター

前述した図4.22の状態での収納するためには、運搬機構のサーボモータを図4.22のような角度で固定することが考えられるが、サーボモータが駆動している状態に衝撃が加わるとモータ内部に大きな負荷を与えることになる。

そこでCanSat落下時にはサーボモータに電力を与えず、その代わりにプロテクターを用いて運搬機構が閉じた状態を維持する。

プロテクターの全体像を図4.23に示す。プロテクターはクリアファイルを加工して制作しており、パラシュートピンを直方体アルミパーツに挿入する際にプロテクターの穴、直方体アルミ

パーツ、プロテクターの穴の順に通すことで固定している。プロテクターはパラシュート切り離す際にパラシュートピンが外れることで展開する(図4.24)。



図4.23 プロテクターの全体像



図4.24 プロテクター固定時(左)と展開時(右)

- スタビライザー

CanSatの後方にあるスタビライザーの全体像を以下の図4.25に示す。スタビライザーはCanSatの姿勢制御に用いる。スタビライザーはサーボモータと連結しており、サーボモータはシャーシ後方に取り付けられている。サーボモータでスタビライザーの角度を変え、CanSatが常に地面と平行になるようにサーボモータを制御する。なお本大会は草地で走行するため、スタビライザーの地面との接地面には、クリアファイルを取り付けることで草との絡まりを軽減している。



図4.25 スタビライザーの全体像

- **タイヤのアタッチメント**

タイヤのアタッチメントの形状は底面が正方形の三角錐で、材質は紫外線硬化樹脂である。形状について、設置部分が尖った三角錐のもので大量についているため、スポーツシューズのスパイクのような役割をする。材質については、加工のしやすさ、耐候性、安価、変形のしやすさ、軽量、大量生産が可能、草に絡まりにくい点から、熱硬化性樹脂を用いた。また、このアタッチメントは図4.26の通りである。



図4.26 タイヤのアタッチメント

4. システム図(CanSat搭載計器仕様一覧)

システム図及び搭載計器仕様一覧表を図4.27と表4.28に示す。ミッションシーケンスのフェーズにおいて、GPSセンサ、9軸センサ(加速度、ジャイロ、地磁気)、気圧センサ、光センサから得られる値を入力としてCanSatの状態をRaspberry Piで処理し、モータ、サーボモータ、ブザー、LED、LoRaを制御する。図4.27は、センサからRaspberry Piに向いている矢印がセンサからの入力を表し、Raspberry Piから、センサに向いている矢印が各センサの制御を表す。赤の矢印は電源供給を表し、黒の矢印はセンサから動力系の制御、青の矢印は制御内容を表す。

使用電源は、下記のようにになっている。

- **乾電池(モータ・運搬機構用サーボモータ電源)**
 - 品名: Energizer Ultimate Lithium AA Batteries
 - 型番: 12-2037
 - 備考
 - 6本を直列(約10V)
 - サーボモータに対して必要な電圧は5Vであるため、スーパー三端子レギュレータを用いて降圧する。
- **リチウムイオンポリマー電池(スタビライザー用サーボモータ電源)**
 - 品名:リチウムイオンポリマー電池7.4V 860mAh
 - 型番:DTP603048-2S
 - 備考
 - サーボモータに対して必要な電圧は5Vであるため、スーパー三端子レギュレー

タを用いて降圧する.

- リチウムイオンポリマー電池(回路用電源)
 - 品名: リチウムイオンポリマー電池3.7V 2000mAh
 - 型番: DTP605068(PHR)
 - 備考
 - Raspberry Piに必要な電圧が5Vであるため, 昇圧型DC-DCコンバータモジュールを用いて昇圧する.

また, 本ミッションでは, 環境情報を取得する各センサ(入力)について, 以下のように用いる.

- GPSセンサ
 - 自己位置推定
 - 取得したGPS座標を用いて自己位置座標を更新
 - ナビゲーション処理
 - ゴール座標と毎回取得するGPS座標の差分から進行方向を調整
 - スタック判定
 - 毎回取得するGPS座標の差が一定以内ならばスタック状態と判定
- 9軸センサ
 - 着地判定
 - 落下中と着地後の加速度が異なることを利用
 - スタック判定
 - 進行方向の加速度の変化を使用
 - ナビゲーション処理
 - 地磁気を用いた進行方向の調整
 - 横転・反転判定
 - 重力の方向から, CanSatの状態を把握
 - 姿勢制御
 - 重力の方向から, 走行時の姿勢を制御
- 気圧センサ
 - 着地判定
 - 落下中に気圧が大きく変化し, 着地後は気圧変化が小さいことを利用
- 光センサ
 - キャリア放出判定
 - ドローンのキャリアから放出されたことを判定
- LoRa
 - ロスト対策
 - GPSセンサが取得した自己位置座標を定期的に発信
- webカメラ
 - 物体検知
 - 周囲の情報を画像データとして取得, 演算し物体を検知
 - 備考: 物体検知の演算にはUSBアクセラレータを補助として用いる

具体的なアルゴリズムについては, 4.4章のアルゴリズムにて説明する. また, モータ, サーボモータ, ブザーの使用目的については, 表4.2の使用目的の通りである.

動力系の制御については, 下記のように制御する.

- モータ
 - モータドライバに対してPWM制御することで左右のタイヤ駆動用の2つのモータを制御
- サーボモータ
 - Raspberry PiのPWM制御にて運搬機構用に2つ, スタビライザー用に1つのサーボモータ

タを制御

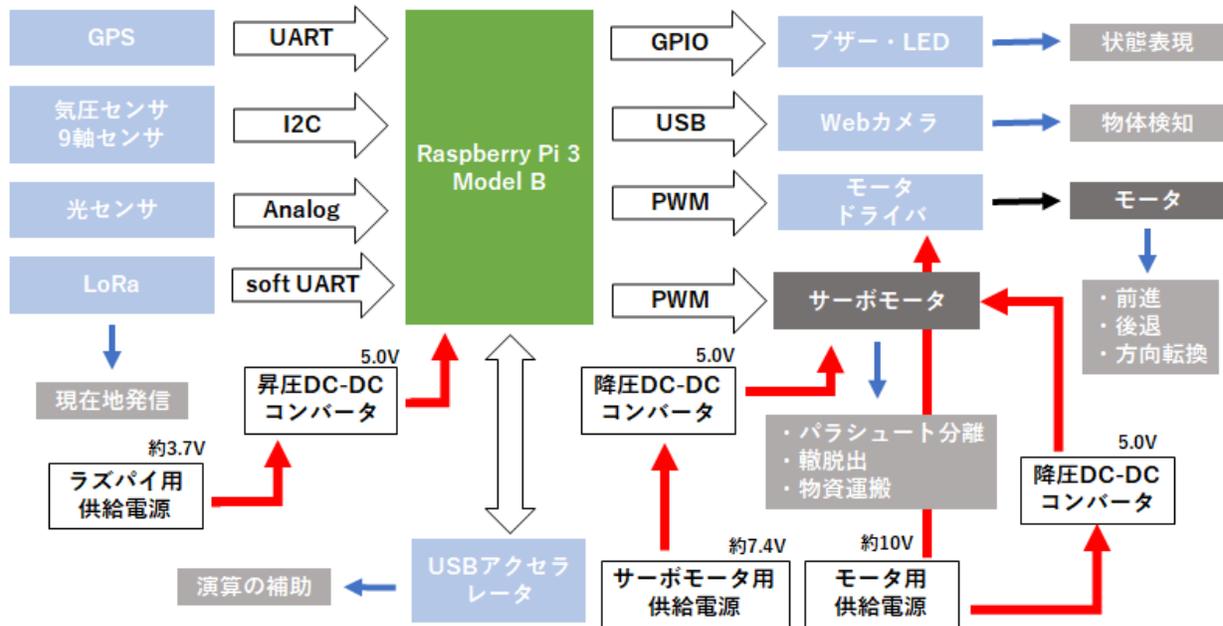


図4.27 システム図

表4.2 搭載計器仕様一覧表

計器名	型番	使用目的	参考情報
マイコン	Raspberry Pi 3 Model B	入出力値の計算, 及び, ログデータの記録	https://www.marutsu.co.jp/pc/i/23172192/
GPSモジュール	GMS-G9, PA6H	GPS座標の取得	http://akizukidenshi.com/catalog/g/gK-09991/
気圧センサ	BME280	気圧センサの取得	http://akizukidenshi.com/catalog/g/gK-09421/
9軸センサ	Hiletgo MPU9250	加速度, 地磁気, ジャイロ値の取得	https://www.amazon.co.jp/dp/B0154PM102/ref=cm_sw_em_r_mt_dp_U_Mzp.CbAD0E5HF

cdsセル(光センサ)	I-00110	ドローンのキャリアからの放出判定	http://akizukidenshi.com/catalog/g/gI-00110/
フルカラーLED	I-02476	CanSatの現状を色によって表現	https://akizukidenshi.com/catalog/g/gI-02476/
ブザー	PB04-SE12HPR	CanSatの現状を音によって表現	http://akizukidenshi.com/catalog/g/gP-04497/
マイコン用電源	12-2037	マイコンへの電源供給	https://www.marutsu.co.jp/pc/i/836419/
モータ・サーボモータ(運搬機構)用電源	DTP605068(PHR)	モータ・運搬機構用サーボモータへの電源供給	https://www.energizer.com/batteries/energizer-ultimate-lithium-batteries
サーボモータ(スタビライザー)用電源	DTP603048-2S	スタビライザー用サーボモータへの電源供給	https://www.marutsu.co.jp/pc/i/1555606/
昇圧型DC-DCコンバータモジュール(5V2.5A)	TPS61230	回路用電源を5Vに昇圧	https://strawberry-linux.com/catalog/items?code=16123
スーパー三端子レギュレータ	V7805-1000	サーボモータの入力可能電圧まで降圧	http://akizukidenshi.com/catalog/g/gM-06350/
ステッピング&DCモータドライバ	DRV8835	モータ制御	http://akizukidenshi.com/catalog/g/gK-09848/
マイクロサーボMG90S	MG90S	パラシュートの切り離し, 運搬機構制御	http://akizukidenshi.com/catalog/g/gM-13227/

マクソンDCモータ	463431(モータ118699,ギアヘッド118185)	走行	https://www.maxongroup.co.jp/maxon/view/service_search?query=463431
LoRa(ワイヤーアンテナ型)	RM-92AN	地上局との通信	https://www.green-house.co.jp/products/rm-92an/
webカメラ	UCAM-C310FBBK	物体検知	https://www.elecom.co.jp/products/UCAM-C310FBBK.html
USBアクセラレータ	INTEL-NCSM2485DK	物体検知演算の補助	https://www.switch-science.com/catalog/4104/

5. アルゴリズム

本ミッション達成のためのアルゴリズムを図4.28に示す.

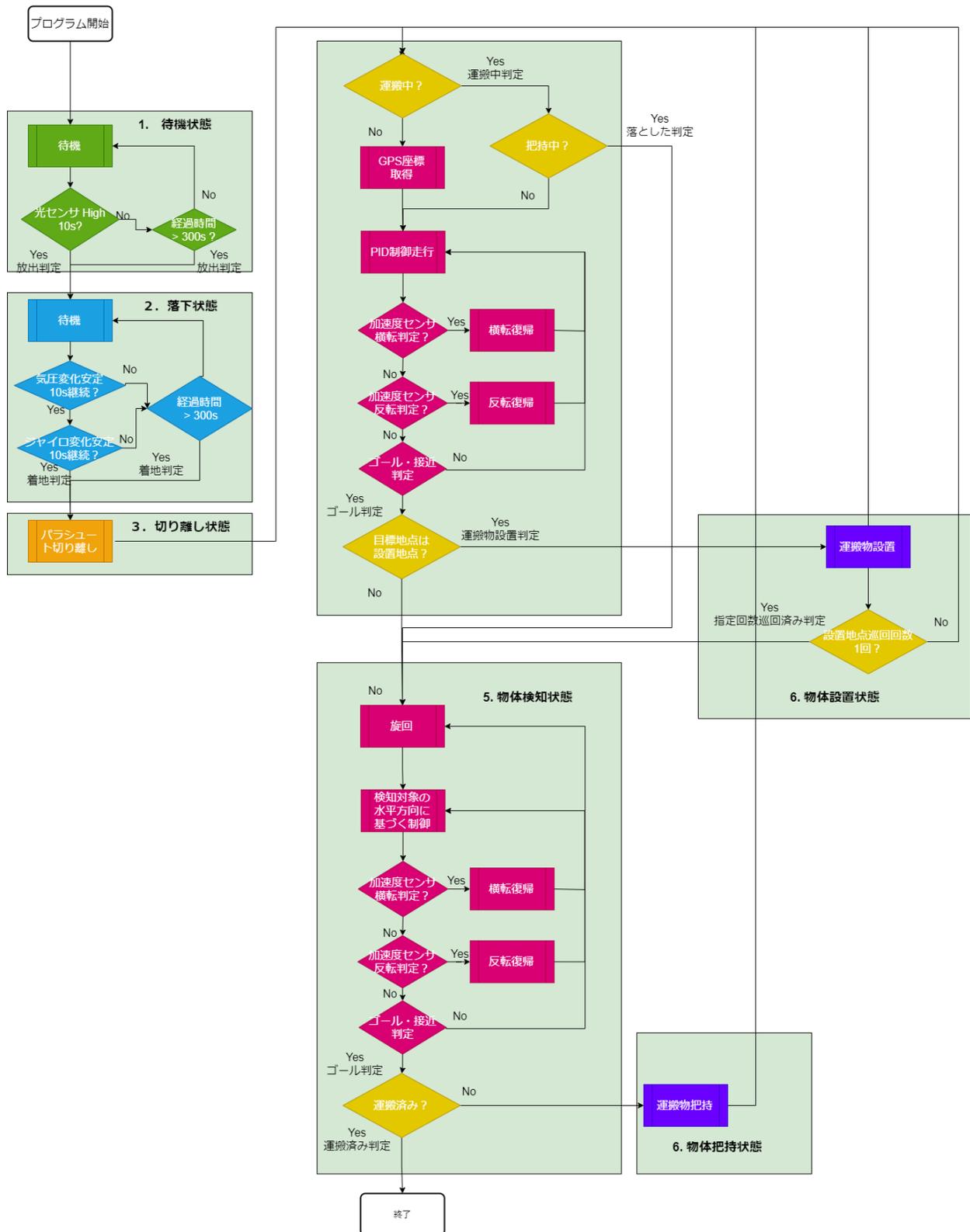


図4.28 フローチャート

ミッションを達成するアルゴリズムは、以下の5つの状態の遷移によって実現される。

1. 待機状態
2. 落下状態
3. 切り離し状態
4. ナビゲーション状態
5. 物体検知状態
6. 運搬物設置・運搬物把持状態

以下に詳細を述べる。

1. 待機状態

プログラムを開始すると、まずこの待機状態に遷移する。待機状態中、CanSatはキャリアから放出されているか否かの判定(放出判定)をする。放出判定には光センサを用いており、光センサが光を10秒間検知した場合に、「CanSatが放出された」と判定する。

プログラムは放出判定後にCanSatが空中落下していると仮定し、プログラムは次の状態である落下状態に移行する。放出判定されなかった場合でもタイムアウト判定(待機状態を開始してから20分)により、落下状態に移行する。

2. 落下状態

落下状態中に、CanSatが空中落下しているか、地面に着地しているかの判定(着地判定)をする。着地判定は加速度センサと気圧センサの値の変位が十分に小さい場合に、「CanSatが地面に着地した」と判定する。センサの値の変位の小ささは落下の際の各センサの値の取得を20回施行した際の各センサの値の平均より判断している。

プログラムは着地判定後に、CanSatについているパラシュートの切り離しのため、切り離し状態に移行する。また、着地判定に失敗しつづけた場合でもタイムアウト(落下状態を開始してから1分)により、切り離し状態に移行する。

3. 切り離し状態

切り離し状態では、CanSatは着地していると仮定し、パラシュート切り離し動作を開始する(パラシュート切り離し機構は3章参照)。パラシュートはCanSat本体に取り付けられたパラシュートピンをサーボモータで引き抜き、切り離す。なお、本CanSatは1~2回サーボモータを動かすことで切り離し可能であるが、CanSatの着地体制やその地面状況を考慮してサーボモータを2回動かすごとに片方のタイヤを駆動させCanSatの体制を変える。この切り離し動作を10回繰り返した後、ナビゲーション状態に移行する。

4. ナビゲーション状態

ナビゲーション状態ではGPSに基づいてCanSatを制御する。また、ナビゲーション中に運搬物を保有していた場合、**※運搬物のロスト対策**も行う(運搬物設置・運搬物把持状態にて述べる)。

GPSに基づくナビゲーション状態ではゴール地点と現在地点(CanSat)のGPS座標を基にPID制御により目標の方角へと走行する。ここで、ある時刻 t 、入力 u_t 、CanSatの正面方位 y_t 、目標方位 r_t とすると、偏差 e_t は以下の式で表せる。

$$e_t = r_t - y_t$$

上式両辺をラプラス変換することにより以下の式が得られる。

$$E_s = R_s - Y_s$$

ここで s は複素数である。PID制御はこの偏差量 E_s を入力にフィードバックすることで制御するものである。PID制御による入力は以下ようになる。

$$U_s = (K_p + K_i/s + sK_D)E_s$$

ここで K_p, K_i, K_D はそれぞれ比例ゲイン, 積分ゲイン, 微分ゲインを表している。また E_s は偏差のラプラス変換後の値である。この値に応じて入力のラプラス変換 U_s の値をフィードバック制御することで目標方位への走行が可能になる。実際には入力の正負と0の3通りに場合分けをしており、右回転・直進・左回転の3通りの行動をとる。また、加速度センサによる横転判定と反転判定が行われゴールコーン、それぞれの判定により横転復帰と反転復帰が呼ばれる。ゴール判定としてGPSセンサの誤差を考慮し、ゴール地点との距離が5m以内と5回判定された場合ゴールしたと判定する。ゴール判定された場合は目標地点が設置地点か回収地点かに応じて物体設置 or 物体把持状態に遷移する。

5. 物体検知状態

この状態では対象となる物体に対して、ローバーに搭載されているカメラで取得した画像を元に物体検知を行う。

対象となる物体はゴールコーンとチームロゴが印刷された立方体の二つであり、それぞれのサイズは以下の図4.29に示すように37×70cmと10×10cmである。



図4.29 ゴールコーンと立方体のサイズ

物体検知をする場合、物体の色を取得するような画像処理によって物体を認識することが考えられる。しかし、本ミッションで用いる立方体のような小さな物体(ゴールコーンに対して約1/26倍の面積)を認識する場合、物体の背景が一定である状況もしくは物体が近くにある状況でなければ(10×10cmの立方体の場合カメラから半径50cm以内でない)色の取得による物体検知は難しい。そこで本ミッションでは運搬物である10×10cmの立方体を検知するためにEfficientDetというCNNの技術を用いた深層学習による物体検知手法を用いる。

このモデルは画像を入力として画像内の物体に対して、その物体の種類と画像内における物体の場所を示すバウンディングボックス(図4.30)を出力することができる。このモデルを学習させる際に多様な環境に対応できるように、部分的に隠れている場合や、光によって色が変化した場合にも対応できるような物体検知モデルを構築する。



図4.30 バウンディングボックス(水色)の例

- 物体検知による航法
 - 撮影した画像でバウンディングボックスが取得できた場合
 - 水平方向のバウンディングボックスの中心(x_{bb})が画像の中心(x_m)になるようにPID制御を行う(図4.31)。ここで行うPID制御もナビゲーション状態同様正前方位 $y_t = x_m$ 、目標方位 $r_t = x_{bb}$ として3通りの行動を繰り返すフィードバック制御を行う。
 - 取得したバウンディングボックスの面積が閾値を超えた時、ゴール判定をする。その後対象物に対応して処理を遷移させる。
 - a. 対象物が立方体の場合
 - 運搬物設置・運搬物把持状態に遷移する。
 - b. 対象物がゴールコーンの場合
 - ミッション終了とし、プログラムを終了する。

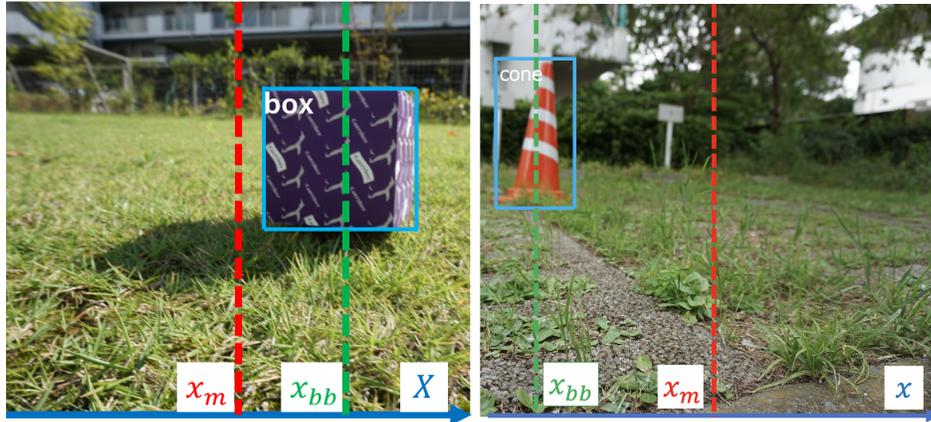


図4.31 制御方法の例(画像の中心(x_m)にバウンディングボックスの中心(x_{bb})が配置されるように接近する)

- バウンディングボックスが取得できなかった場合
 ゴールコーンのような物体への接近する場合、カメラの画角にうつらない状況は少ないが、立方体のような小さな物体へ接近する場合、接近方位の小さなずれや地形の状況によってカメラの画角から外れ、物体を見失うという問題がある(図4.32).

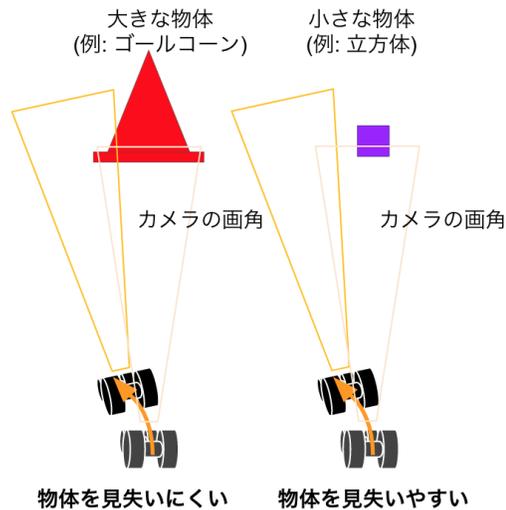


図4.32 物体のサイズ違いによるCanSatの制御問題

そこで、バウンディングボックスが取得できなかった場合には、以下の図4.33、図4.34のような接近動作をする。

- 直近に一度でも取得した場合
 4方向に対して物体検知をする。

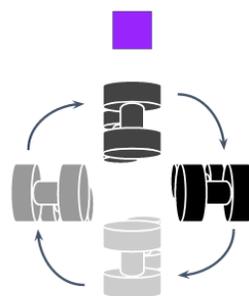


図4.33 直近に一度でも取得した場合

- それ以外
渦巻状に旋回移動する.

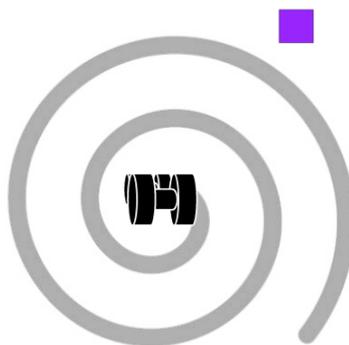


図4.34 それ以外の場合

なお、これらの制御時にはナビゲーション状態同様に横転・反転の判定もする。

6. 運搬物設置・運搬物把持状態

運搬物設置・運搬物把持状態では既に物体を把持しているかに応じて(1)運搬物設置(2)運搬物把持を行ったのちナビゲーション状態に遷移する。ただし、巡回済みの場合全ての運搬物を運搬済みと判断して、ゴール判定を行いミッションを終了する。

1)物体運搬では、ローバーの前面にある二本のアームで運搬物を掴む。この際、後部についているサーボモータの角度を変更し、姿勢を安定させる制御を行う。また、掴めない場合の想定も行う。ACTSのフィールドは芝生であることから、撮影した画像を画像処理によって緑色を検出する処理を入れることで掴めたかどうかの判定を行う。掴めた場合はカメラで撮影した画像が運搬物で覆われており、緑色を検出できないはずである。そこで掴めたかどうかの判定を行い、つかめていないと判定した場合は、再度物体を掴むように制御を行う。

運搬物をロストした際に再度運搬をする様子を以下の動画に示す。また、V15 物体検知および運搬試験の運搬物ロスト試験を参照されたい。

<https://youtu.be/sqsBPevzsno>



図4.35 運搬物把持状態の例

2)運搬物設置では、CanSatの前面にある2本のアームを開くことで、運搬物を地面に降ろす。また運搬物設置状態を経過したのち物体検知状態に遷移し、ゴールコーンを対象物体とする物体検知を行う。

6. 会計

CanSatは2回の打ち上げのための2機+予備パーツを作成した。部品購入費の総計を以下に示す。

ハード	回路	ソフト	総計
43,761円	35,774円	13,046円	92,581円

なお、上記の費用に加えてモータ(型番: maxon DC motor 46343)の代金は51,400円である。

第5章 試験項目設定

番号	検証項目名	対応する自己審査項目の 要求番号(複数可)	実施日	
V1	質量試験	R1	10/16	完了
V2	キャリア収納試験	R1, R10	10/16	完了
V3	通信機器電源 ON/OFF 試験	R2, R7	7/20	完了
V4	長距離通信試験	R2	7/23	完了
V5	落下試験	R3	6/15	完了
V6	開傘衝撃試験	R3	10/11	完了
V7	準静的荷重試験	R4	10/15	完了
V8	通信周波数変更試験	R8	7/18	完了
V9	ナビゲーション試験	R11	8/23, 26	完了
V10	End-to-End 試験	R9, R11	10/24	完了
V11	制御履歴レポート作成試験	R12	10/24	完了
V12	着地衝撃試験	M1, M2	10/10	完了
V13	電力耐久試験	M3	8/26	完了
V14	反転・横転復帰試験	M4	8/13	完了
V15	運搬および物体検知試験	M5, M6, M8	10/22-23	完了
V16	ゴール検知試験	M7	8/27	完了

第6章 実施試験内容

※衝撃がかかる実験(開傘衝撃試験・静的荷重試験・着地衝撃試験)において、センサ・動力系が正常に動いていることを確認するために以下を満たせば、正常に動作しているとみなす。

＜センサが正常である条件＞

1. GPSセンサ

- センサ値が取得できているかPC画面上で確認できる
- ナビゲーション状態においてCanSatがゴール方向に向かうことが確認できる

※ただし静的荷重試験は柔らかいマットのある室内で実施する(硬い地面と接触し、機体が破損することを避けるため)。そのため、GPSセンサの正常な値が得られないことから、センサ値の取得ができていることのみ確認する。

2. 光センサ

- センサ値が取得できているかPC画面上で確認できる
- 光が当たらないところでは待機状態、光が当たるところで待機状態から落下状態に移行ができる

3. 気圧センサ

- センサ値が取得できているかPC画面上で確認できる
- 気圧変動がある状態では落下状態、気圧変動がない状態で落下状態からパラシュート切り離し状態に移行ができる

4. 加速度センサ

- センサ値が取得できているかPC画面上で確認できる
- 加速度の大きさ(ノルム)が開傘衝撃前と後で変わっていないことを確認できる
- CanSatが加速度センサ値を用いて反転・横転からの復帰ができる(XYZ軸が正常であるといえる)

＜動力系が正常である条件＞

5. タイヤ用モータ

- パラシュート切り離し状態・ナビゲーション状態でアルゴリズム通りの動作をしていることが確認できる

6. スタビライザー用・アーム用サーボモータ

- パラシュート切り離し状態・ナビゲーション状態でアルゴリズム通りの動作をしていることが確認できる

7. システム要求を満たすための試験内容

(V1) 質量試験

- 実施日：10/16
- 実施責任者：前川
- 充足要求項目：R1
- 目的

- CanSat本体とパラシュートの合計質量がレギュレーションに記載されている最大質量(1050g)以下であることを確認する.
- 試験/解析内容
 - 電子はかりを用いてCanSat, パラシュート, プロテクター, ネジ緩み止めなど用のホットボンド1本の合計質量を計測する.
- 結果
 - 実験動画のURLを以下に示す.
<https://youtu.be/No30e2h-xus>
 - 図6.1は質量測定時の図である. 本CanSatの合計質量は1049gであった.



図6.1 CanSatの質量

- 結論
 - CanSat, パラシュート, プロテクター, ネジ緩み止めの合計質量はレギュレーション1050g以内をみたす.

(V2) 収納放出試験

- 実施日: 10/16
- 実験責任者: 福本
- 充足要求項目: R1, R10
- 目的
 - CanSatをキャリアに収納し, その後自重で放出可能であることを確認する. また, レギュレーションに記載された5分以内に収容可能であることも確認する.
- 試験/解析内容
 - 下記の手順で実施する. 収納時間は動画の時間で計測する.
 1. CanSatをキャリアに収納する
 2. キャリアを逆さにし, CanSatが自重で放出されることを確認する
- 結果
 - 試験の結果を以下の表6.1に示す.

表6.1 機体の収納放出試験の結果

回数	実験動画	収納時間[m:s]	放出判定
1	https://youtu.be/bJhikU57C6U	0:48	自重で放出できた
2	https://youtu.be/hxTTbzlr3h0	1:14	自重で放出できた
3	https://youtu.be/HkUsk5GMZmk	0:46	自重で放出できた

- 結論
 - CanSatをキャリアに収納後、自重で放出可能である。また、レギュレーションに記載された5分以内で収納できる。

(V3) 通信機器電源 ON/OFF 試験

- 実施日:
- 実施責任者: 谷津
- 充足要求項目: R2, R7
- 目的
 - ロケットの通信機器に影響を与えないために、キャリア内で通信機器が電波を発しない状態になることを確認する。
- 試験/解析内容

通信機器はRaspberry Piに搭載されているwifiモジュールとLoRaモジュールがあるため、それら両方を検証する。

 - wifi ON/OFF試験
 - Raspberry Piによって作成したWi-Fiアクセスポイント「carryber」が存在し、電波を発していることを確認する。その後、キャリア内にCanSatを格納した状態でプログラムを開始する。この時点でノートPCとの通信が切断されることを確認するとともに、CanSatが作成するWi-Fiアクセスポイント「carryber」が消えることを確認する。最後にキャリアからCanSatを放出し、放出と判定されたタイミングでCanSatが作成するWi-Fiアクセスポイント「carryber」が再度表示されることを確認する。
 - LoRa ON/OFF試験
 - LoRaモジュールはRaspberry PiのGPIOピンを制御することで、スリープモード(電波の出力が0)にすることが可能である。そのため、該当するGPIOピンを制御し、LoRaのスリープモードを自由にOn/Offできることを確認する。
ターミナルにコマンドを入力し、CanSat側のLoRaの通信状態をOnからOffに切り替える。LoRaの通信状態がOffに切り替わることを、地上局側のLoRaがCanSatに搭載されているLoRaからの信号を受け取れなくなることにより確認する。次に、Wi-Fiの時と同様にして、地上局側のLoRaがCanSatからの信号を受け取れることにより、LoRaの通信状態がOnに切り替わることを確認する。
- 結果

- wifi ON/OFF試験
 - 実験動画のURLを以下に示す.
 - <https://youtu.be/bjSA-dk081I>
 - CanSatが待機状態でwifiが正常に切れることが確認できた.
- LoRa ON/OFF試験
 - て実験動画のURLを以下に示す.
 - <https://www.youtube.com/watch?v=phBMUK1aSc4>
 - LoRaの通信のON/OFFが実施できることが確認できた.
- 結論
 - wifi
 - ロケット打ち上げ時にロケットの通信機器と干渉する可能性がある際に通信機器のon/offを適切に切り替えることが可能である
 - LoRa
 - wifi同様ロケット打ち上げ時に干渉する可能性がある際のon/offを適切に変えることが可能である

(V4) 長距離通信試験

- 実施日：7/23
- 実施責任者：前川
- 充足要求項目：R2
- 目的
 - CanSatとPCの通信可能な最大距離を測定し、放出後もCanSatをロストせずに捕捉可能であることを確認する.
- 試験/解析内容
 - 障害物が少ない河川敷で実験をする。まず、長距離無線通信モジュールLoRaを繋いだPCを地上局とし、CanSat側から地上局へCanSatの位置情報をLoRaで発信し、PC側がその位置情報を受信できることを確認する。次に、地上局の位置を固定し、CanSatを持った人が地上局から離れていき、地上局側がCanSatから位置情報を受信できなくなる地点で両者のGPS座標を取得し、2点間の距離を通信可能距離とする。Salamanderチームと共同で試験をした。
- 結果
 - CanSatとPC(地上局)の通信が切断された時のそれぞれのGPS座標を以下に示す。また、図6.2はその時の座標を地図上に示したものである。

表6.2 長距離通信試験結果

PC(地上局)の座標	緯度: 35.6425 経度: 139.52333
CanSatの座標	緯度: 35.6375033 経度: 139.5392733
最大通信可能距離	1.53 km



図6.2 長距離通信試験時の地上局とCanSatの位置

- 結論
 - 結果より地上局とCanSatの通信可能距離は1.53kmであり、会場の大きさを踏まえるとCanSatの投下後、地上局を介しCanSatのGPS座標を取得することは十分可能であるといえる。したがって、ロスト対策に十分な通信可能距離であるということが確認された。

(V5) 落下試験

- 実施日：6/15
- 実施責任者：前川
- 充足要求項目：R3
- 目的
 - キャリアから放出された後、減速機構であるパラシュートが問題なく開傘し、減速が可能であることを確認する。また、レギュレーションで定められている終端速度(5m/s以上)よりも早いことを確認する。
- 試験/解析内容
 - 電気通信大学西6号館7階(地面からの高さ約27.3m地点)から、パラシュートを付けた擬似CanSat(重量および形状・高さが実際のCanSatと同等になるように作成したモックアップ)を用いてキャリアに収納し降下させる。本試験では、パラシュートの性能の検証を目的としているため、本CanSatと同等の大きさ・質量・重心である擬似CanSatを用いる。(なお、着地時の衝撃についてはV13で検証する)。また、擬似CanSatと本CanSatの重心位置が同等であることを以下の図6.3に示す。重心の算出方法は、CanSatを縦方向と横方向で紐に吊るし、その紐の延長線を黄色い線で描く。縦方向と横方向の画像の重ね合わせ、黄色い線の交点を重心位置とした。さらに、減速の検証は実験動画から落下速度を算出し、落下時の速度が自由落下に比べて減速していることを確認する。

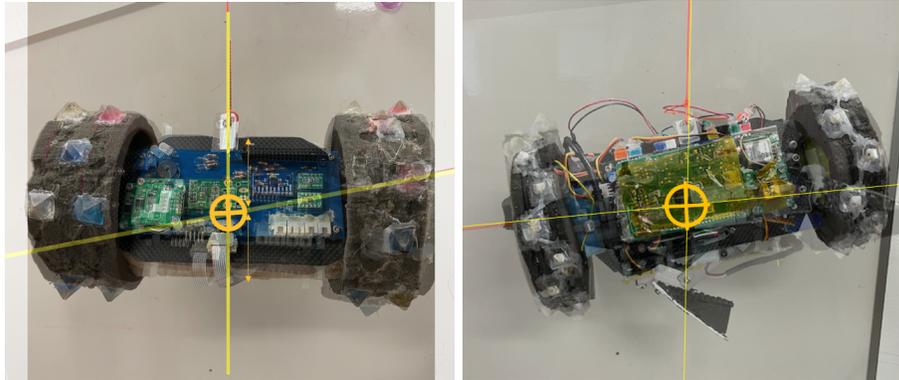


図6.3 擬似CanSat(左)と本CanSat(右)の重心位置

- 落下時の速度について、CanSatが地上から約8.4m(建物3階)の高さを通過したとき、落下速度が終端速度と等しくなると仮定した。表6.5の実験動画5本の建物3階から2階までの通過時間と2階から1階までの通過時間を測定したところ、表6.4のような結果が得られた。

表6.4 CanSatの大きさと重量

実験回数	3～2階の通過時間 [s]	2～1階の通過時間 [s]	通過時間の差 [s]
1回目	0.6	0.7	0.1
2回目	0.6	0.6	0
3回目	0.5	0.6	0.1
4回目	0.7	0.7	0
5回目	0.7	0.7	0

- 表6.4より、それぞれの通過時間の差の平均は0.04秒である。実験動画のフレームレートは29.97fpsであり、1フレームあたり約0.033秒を要するため、同じ速度で落下してもそれぞれの通過時間に0.03~0.07秒の差が生じる。したがって、この仮定は正しく、CanSatは地上から建物3階の高さ(約8.4m)間では落下速度と終端速度が等しくなるといえる。よって、疑似CanSatが地上から約8.4m(建物3階)の高さを通過してから着地するまでにかかった時間を動画から計測し、以下の式を用いて終端速度を計算した。

$$h = 8.4 \text{ m}$$

$$v = h/t \text{ m/s}$$

また、自由落下した際の速度は以下の式より、23.1m/sと求まる。

$$v = (2gh)^{1/2}$$

$$h = 27.3\text{m}, g = 9.8\text{m/s}^2$$

$$\therefore v = 23.1 \text{ m/s}$$



図6.4 地面から3階までの高さ

○ 結果

- 実験結果を以下の表6.5に示す.

表6.5 落下試験結果

回数	実験動画	結果	算出終端速度
1回目	https://youtu.be/2SKFiXERwbA	開傘成功	8.4m / 1.2 s \doteq 7.0 m/s (動画16:9~18:1)
2回目	https://youtu.be/e29We9-HHrQ	開傘成功	8.4m / 1.2 s \doteq 7.0 m/s (動画21:2~22:4)
3回目	https://youtu.be/3Fn7-ez7iXU	開傘成功	8.4m / 1.5 s \doteq 5.6 m/s (動画12:7~14:2)
4回目	https://youtu.be/3Fn7-ez7iXU	開傘成功	8.4m / 1.5s \doteq 5.6 m/s

	be/HgmGby2a3A		(動画13:5~15:0)
5回目	https://youtu.be/0FO46aPI0k	開傘成功	8.4m / 1.4s \doteq 6.0 m/s (動画7:3~8:7)

平均終端速度は6.24m/sであり、これは自由落下時の速度である 23.1m/s に比べて小さく、パラシュートによる減速が確認された。
また、最低終端速度は5.0m/s以上であった。

- 考察
 - キャリアから放出後、パラシュートの開閉は可能である。さらに、レギュレーションで定められている終端速度 (5m/s以上) を満たしている。

(V6) 開傘衝撃試験

- 実施日： 8/18
- 実施責任者： 前川
- 充足要求項目： R3
- 目的
 - キャリアからCanSatが放出され、落下中のパラシュート開傘衝撃によって、CanSat及びパラシュート連結機構の機能が損なわれていないことを確認する。またCanSatに異常が生じないことを確認する。
- 試験/解析内容
 - CanSatにパラシュートを取り付けた状態で、パラシュート紐の上端を固定し自由落下させることで、開傘時の衝撃を再現する。
待機状態からプログラムを開始し、開傘衝撃を与え、ナビゲーション状態まで人の手を介さずにシーケンスが移行できることを確認する。その後、全てのセンサの動作確認、CanSatに破損がないかを確認する。センサ・動力系に対しては6章冒頭で述べた条件を満たしていれば正常に動作しているとみなす。
 - また、今回用いるパラシュートは昨年使用したものと傘のサイズ、紐の長さが等しいため、開傘衝撃値は昨年のACTS本番のログを参考にする。
昨年のACTS本番のログ(図6.5)の開傘時に観測した最大加速度9.6Gを参考とし、10Gを超えるように実験する。

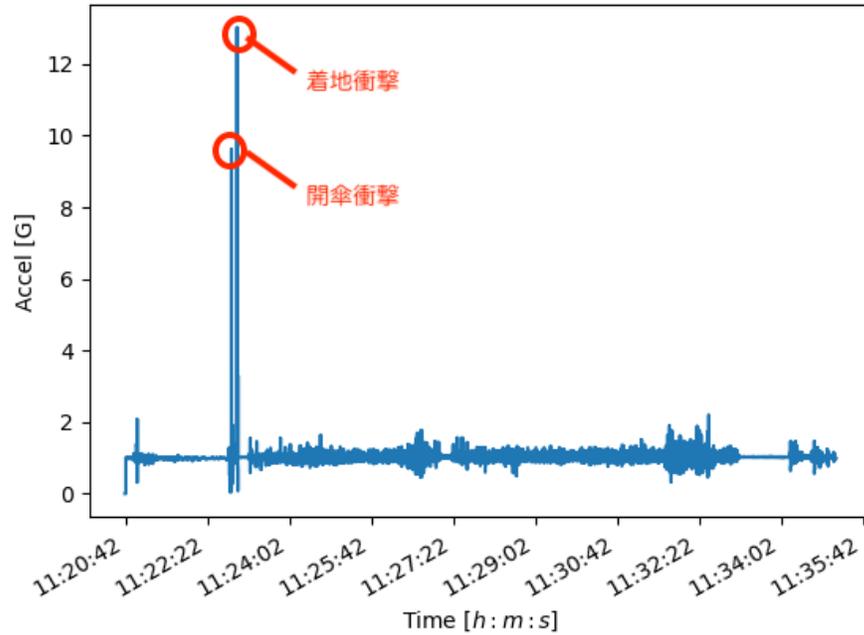


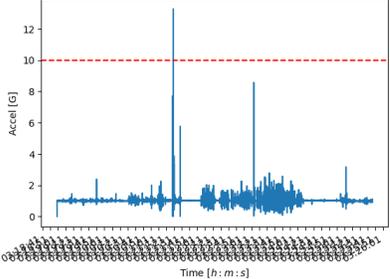
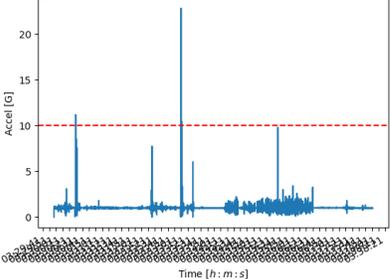
図6.5 昨年のACTS本番のログ

○ 結果

- 実験結果を以下の表6.6に示す. 動力系・センサ類が前述した条件を満たすように動作した場合は○, 動作しなかった場合は×と表記する.
- 加速度センサの取得値グラフは, 横軸は経過時間[s], 縦軸は加速度の大きさ(ノルム)が重力加速度の何倍であるかを示す. グラフ内の赤の点線は10Gを示す.

表6.6 開傘衝撃試験の結果

回数	実験動画	結果			外観
		動力系	センサ		
			その他センサ	加速度センサ取得値	
1	https://www.youtube.com/watch?v=YcKpQOFiDnk&list=PLJ7eKAd4pnIBucWFvp6JkaAJYRVgJITN&index=10	タイヤ用モータ: ○ スタビライザー用・アーム用サーボモータ: ○	GPSセンサ: ○ 光センサ: ○ 気圧センサ: ○	加速度の大きさ(ノルム)が開傘衝撃前と後で変わっていない 加速度センサ値を用いて反転・横転復帰可能 	破損なし

2	https://www.youtube.com/watch?v=iNbTFufdOm0&list=PLJ7eKAd4pnIBucWFvp6JkaAJYRVqJITN&index=13	タイヤ用モータ: ○ スタビライザー用・アーム用サーボモータ: ○	GPSセンサ: ○ 光センサ: ○ 気圧センサ: ○	加速度の大きさ(ノルム)が開傘衝撃前と後で変わっていない 加速度センサ値を用いて反転・横転復帰可能 	破損なし
3	https://www.youtube.com/watch?v=MlccwOWF6o0&list=PLJ7eKAd4pnIBucWFvp6JkaAJYRVqJITN&index=8	タイヤ用モータ: ○ スタビライザー用・アーム用サーボモータ: ○	GPSセンサ: ○ 光センサ: ○ 気圧センサ: ○	加速度の大きさ(ノルム)が開傘衝撃前と後で変わっていない 加速度センサ値を用いて反転・横転復帰可能 	破損なし

- 結論
 - 開傘衝撃がかかっても、パラシュート連結機構の機能および外観状態、センサ値の取得、動力系の動作に問題がない。

(V7) 準静的荷重試験

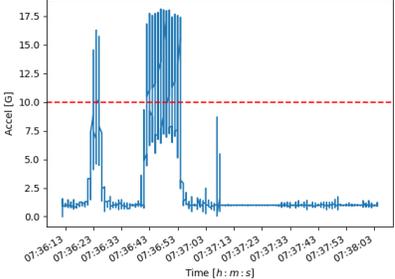
- 実施日: 8/25
- 実施責任者:
- 充足要求項目: R4
- 目的
 - ロケットによる打ち上げを想定した際、打ち出し時に生じる荷重によってCanSatに異常が生じないことを確認する。静的荷重ではなく準静的荷重にすることで実際にかかるであろう負荷を想定して実験する。
- 試験/解析内容
 - キャリアの縦方向(高さ)に約10Gを10秒間機体に与える。この状況を再現するために、紐につないだ袋にロケット搭載状態を想定した機体を入れてハンマー投げの加速時の要領で袋を回し、CanSatに搭載された加速度を用いて測定する。なお、

センサ値は0.1秒ごとに値を取得する。待機状態からプログラムを開始し、準静的荷重を与え、ナビゲーション状態まで人の手を介さずにシーケンスが移行できることを確認する。その後、全てのセンサの動作確認、CanSatに破損がないかを確認する。センサ・動力系に対しては6章冒頭で述べた条件を満たしていれば正常に動作しているとみなす。

- 実験結果を以下の表6.7に示す。動力系・センサ類が前述した条件を満たすように動作した場合は○、動作しなかった場合は×と表記する。加速度センサの取得値グラフは、横軸は経過時間[s]、縦軸は加速度の大きさ(ノルム)が重力加速度の何倍であるかを示す。グラフ内の赤の点線は10Gを示す。

表6.7 準静的荷重試験結果

回数	実験動画	結果			外観
		動力系	センサ		
			その他センサ	加速度センサ取得値	
1	https://youtu.be/C-5um_qSses	タイヤ用モータ: ○ スタビライザー用・アーム用サーボモータ: ○	GPSセンサ: ○ 光センサ: ○ 気圧センサ: ○	<p>加速度の大きさ(ノルム)が静的荷重試験前と後で変わっていない</p> <p>加速度センサ値を用いて反転・横転復帰可能</p>	破損なし
2	https://youtu.be/FUvTEB7CFJo	タイヤ用モータ: ○ スタビライザー用・アーム用サーボモータ: ○	GPSセンサ: ○ 光センサ: ○ 気圧センサ: ○	<p>加速度の大きさ(ノルム)が静的荷重試験前と後で変わっていない</p> <p>加速度センサ値を用いて反転・横転復帰可能</p>	破損なし

3	https://youtu.be/mvZ2Wa32pPU	タイヤ用モータ: ○ スタビライザー用・アーム用サーボモータ: ○	GPSセンサ: ○ 光センサ: ○ 気圧センサ: ○	加速度の大きさ(ノルム)が静的荷重試験前と後で変わっていない 加速度センサ値を用いて反転・横転復帰可能 	破損なし
---	---	--------------------------------------	----------------------------------	---	------

- 結論
 - 準静的荷重がかかっても、外観、センサ値、動力系の動作に問題がない。

(V8) 通信周波数変更試験

- 実施日: 10/23
- 実施責任者: 谷津
- 充足要求項目: R8
- 目的
 - 他のチームの通信と通信周波数が干渉する可能性がある場合に、使用する通信周波数を変更可能であることを確認する。
通信に利用する規格はLoRaとwifiの2種類のため、それぞれについて試験を実施する
- 試験/解析内容
 - LoRa通信周波数変更
 - 予め決められているポーレートでLoRaモジュールとノートPCでシリアル接続をする。LoRaモジュールで現在のLoRaが出している通信周波数を確認した後に、設定を変更し通信周波数を変更する。再度LoRaが出している通信周波数を確認して、変更前の周波数では通信ができず、変更後の周波数で通信が可能なことを確認する。
 - wifi通信周波数変更
 - はじめの使用チャンネルを決めておき、ノートPC側からRaspberry Piが出すアクセスポイントへ接続しておく。また、外部からの確認として、他の通信機器(タブレット)でWi-Fiの電波状況を表示するソフトウェアによってアクセスポイントの使用周波数を確認する。
その後、Raspberry Pi内の設定ファイル(/etc/create_ap.conf)の内容を変更し、Wi-Fiアクセスポイントのチャンネル番号を変更する。アクセスポイントを作成するためのサービス(create_ap)を再起動し、変更を反映する。その後、他の通信機器(タブレット)からアクセスポイントの使用周波数を確認し、変更前の周波数では通信ができず、変更後の周波数で通信が可能なことを確認する。

- 結果
 - LoRa通信周波数変更
 - 実験動画のURLを以下に示す
<https://www.youtube.com/watch?v=hMEpMIKqmgc>
 初めはLoRaが24チャンネルを出しており、CanSat側と通信が来ていることが確認できる。また周波数を24チャンネル以外の値に変更することで周波数が変更され通信が来ていないことが確認できる。以上からLoRaの周波数を変更できることが確認できる。
 - wifi通信周波数変更
 - 実験動画のURLを以下に示す
<https://www.youtube.com/watch?v=rxi8xYV4tyQ>
 デフォルトで設定されていたWiFiの周波数チャンネルCH1が設定ファイルの書き換えによりCH2の周波数チャンネルに変更されたことを確認した。
- 結論

他のチームの通信と通信周波数が干渉する可能性がある場合に、通信周波数を調整することが可能である。

(V9) ナビゲーション試験

- 実施日：8/24、8/26
- 実施責任者：
- 充足要求項目：R11
- 目的
 - CanSatが自律制御で目的地まで走行可能であることを確認する。
- 試験内容
 - 本ミッションは、CanSatが1つ目の運搬物がおいてある地点(サブゴール1)を経由し、運搬先である地点(最終ゴール)に向かう。さらに2つ目の運搬物がおいてある地点(サブゴール2)に向かい、最終ゴールに戻ってくることを確認する。本大会で想定される地面状況(砂地・草地)で走行制御できることを示すため、砂地・草地の環境で実験する。
 なお本試験は、CanSatが自律制御で3つの目的地に順に向かうことができるかを確認するため、物体の運搬はしない。(運搬に関しては、V15. 物体検知および運搬試験を参照されたい)
- 結果
 - 表6.8に実験結果を示す。

表6.8 ナビゲーション試験の結果

回数	地面状況	実験動画	結果
1	砂地	https://youtu.be/J0-XHiOeFpg	走行可能
2	砂地	https://youtu.be/Rty6CYSCx0o	走行可能

3	砂地	https://youtu.be/J5zLqIAgbO4	走行可能
4	草地	https://youtu.be/7AzNMyawMI0	走行可能
5	草地	https://youtu.be/2aSp7RqJ25Y	走行可能
6	草地	https://youtu.be/2MmyLj889zQ	走行可能

- 結論
 - GPSに従って自律制御で各目的地まで走行可能である。

(V10) End-to-End 試験

- 実施日：10/24
- 実施責任者：前川
- 充足要求項目：R9, R11
- 目的
 - レギュレーションを満たしたCanSatがキャリア収納からミッション終了までの全てのシーケンスを連続して問題なく自律的に制御できるか確認する。
- 試験/解析内容
 - CanSatが本ミッションのサクセスクライテリアを満たすことを確認する。キャリア放出後、着地判定、パラシュート切り離し、着地地点から運搬開始地点までのナビゲーション、運搬開始地点から運搬終了地点までの運搬、運搬終了地点からゴール地点までのナビゲーション、半径5m以内でのゴール判定の全てのシーケンスを自律的に制御できることを確認する。
- 結果
 - 実験動画のURLを以下に示す。
<https://youtu.be/1iHaq7C27iY>
 試験/解析内容に記述した全てのシーケンスを自律的に制御できた。
 ただし、運搬物を把持し運搬することはかなわなかった。これは運搬物方向からの太陽光が強く、運搬物の検知精度が下がり、運搬物への接近が困難であったことが原因である。
 太陽光がカメラに直接当たる場合の対策として、カメラに太陽光カットフィルターを被せること、カメラ画像内の露出を下げる等を検討しており、このような状況下でも運搬物の検知ができるような改善を今後もしていく。
- 結論
 - GPSに従って自律制御で各目的地まで走行可能である。

- ただし本ミッションのサクセスクライテリアを達成するためには、前述した改善が必要である。

(V11) 制御履歴レポート作成試験

- 実施日：10/24
- 実施責任者：千住
- 充足要求項目：R12
- 目的
 - ミッション後、規定の制御履歴レポートを運営へ提出する必要があることから、CanSatが保存したログデータから制御履歴レポートを作成できるかを確認する。
- 試験/解析内容
 - End-to-End試験で得られたログデータから、提出用の制御履歴レポートを作成する。
- 結果
 - 以下の図6.6, 6.7にGPSセンサの軌跡図と動作詳細を示す。
青い線はスタート地点とゴール地点の座標を示し、赤い線はCanSatの通った軌跡を示している。
また、CanSatの制御は、白丸地点で白線の伸びている方向にされている。なお、ナビゲーション走行に入る前ではゴール方向と異なる方向に制御されているが、これは放出判定、着地判定のためにCanSatを手にとっていたのである。

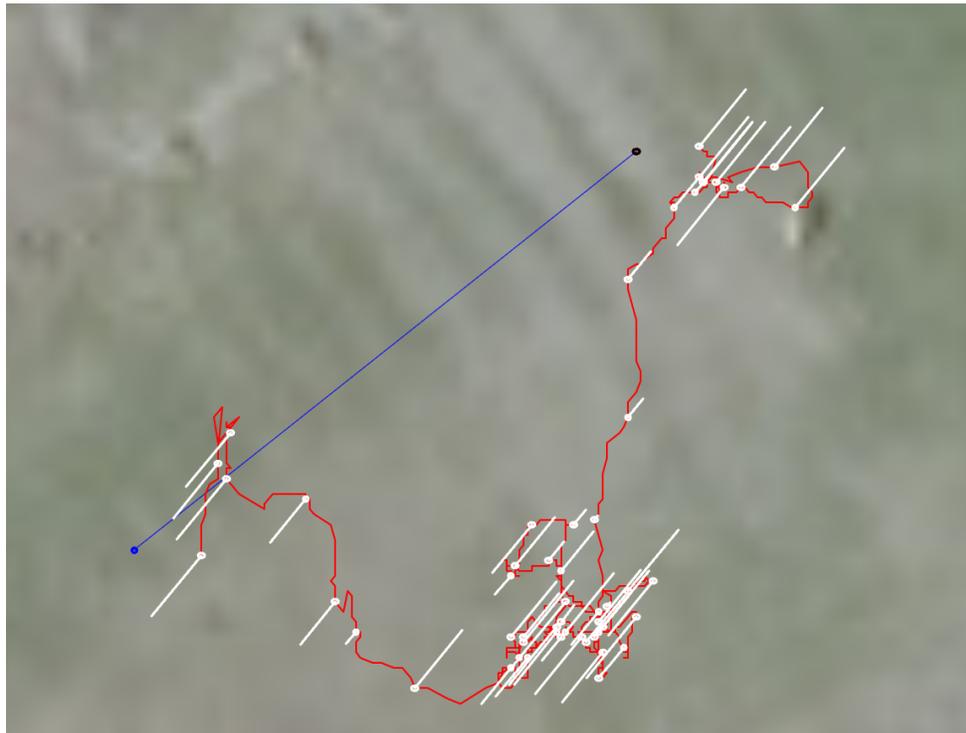


図6.6 GPSセンサの軌跡図

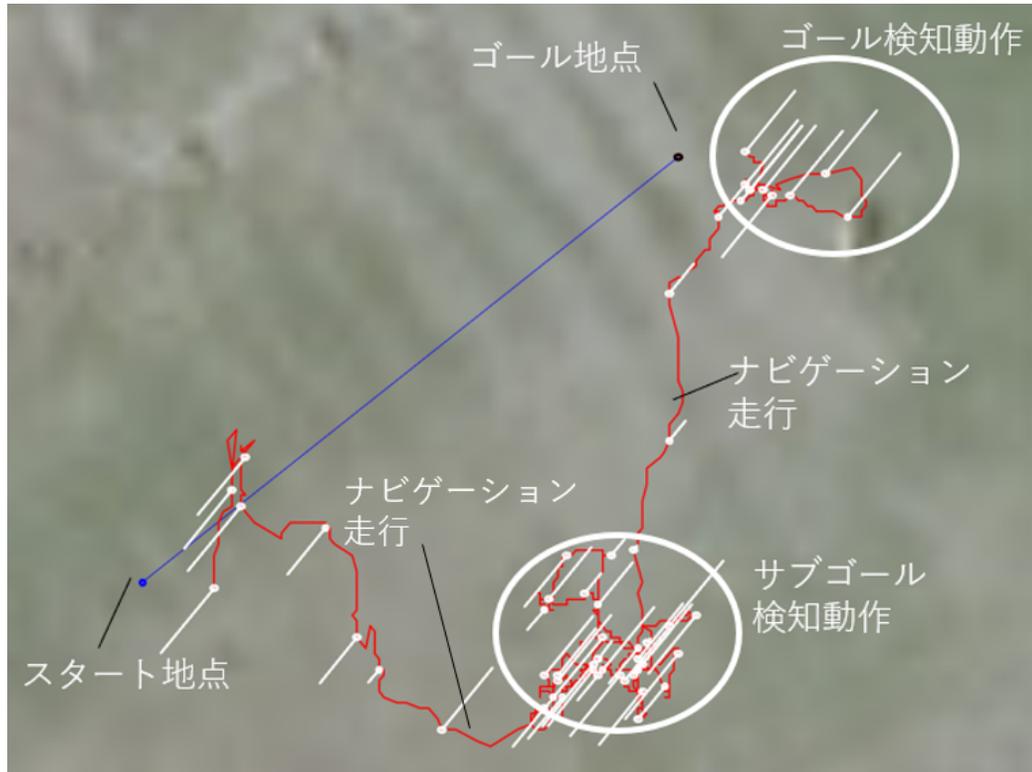


図6.7 GPSセンサの動作詳細

表6.9 End-to-End試験 制御履歴レポート

<p>※太字: ログの説明(追記部分), 赤色: シーケンス</p> <p>GPS Sensor is Ready! → GPSセンサの取得確認 Pressure Sensor is Ready! → 気圧センサの取得確認 Camera is ready → カメラの取得確認 ReadNetwork end → 物体検知モデル読み込み</p> <p><中略></p> <p>Log file: log/20211023_2044_log_nineaxis1_e2e9.csv → ログファイルの出力</p> <p><中略></p> <p>[20:51:08] Sequence: 'waiting' → 待機開始 Time: 20:51:09 → 時間[hh:mm:ss] Light count: 1/10 (HIGH) → 光センサによる放出判定</p> <p><中略></p> <p>Time: 20:51:17 → 時間[hh:mm:ss] Light count: 9/10 (HIGH) → 光センサによる放出判定</p> <p>[20:51:18] Sequence: 'falling' → 落下状態</p> <p>Time: 20:51:19 → 時間[hh:mm:ss] Pressure count: 1/10 (1023.537 hPa) → 気圧センサによる放出判定</p>

Gyro count: 0/10 → **9軸センサによる放出判定**
GPS position: 35.6431550, 139.5232333 → **放出地点のGPS座標**

Time: 20:51:20 → **時間[hh:mm:ss]**
Pressure count: 2/10 (1023.476 hPa) → **気圧センサによる放出判定**
Gyro count: 0/10 → **9軸センサによる放出判定**
GPS position: 35.6431583, 139.5232333 → **放出地点のGPS座標**

<中略>

Time: 20:51:28
Pressure count: 10/10 (1023.489 hPa) → **着地判定の条件1つ目(気圧センサによる判定)をクリア**
Gyro count: 0/10
GPS position: 35.6431700, 139.5232367

<中略>

Time: 20:51:34
Pressure count: 16/10 (1023.526 hPa)
Gyro count: 1/10 → **着地判定の条件2つ目(9軸センサによる判定)を開始**
GPS position: 35.6431683, 139.5232383

Time: 20:51:35
Pressure count: 17/10 (1023.522 hPa)
Gyro count: 2/10
GPS position: 35.6431667, 139.5232367

<中略>

Falling completed! (by gyro and pressure) → **着地判定の条件2つ目(ジャイロセンサで判定)をクリア**

[20:51:43] Sequence: 'para_separating' → **パラシュート切り離し動作開始**

Para separating... (1/10)
Para separating... (2/10)
Para separating... (3/10)
Para separating... (4/10)
Para separating... (5/10)
Para separating... (6/10)
Para separating... (7/10)
Para separating... (8/10)
Para separating... (9/10)
Para separating... (10/10)
Para separating finished! → **パラシュート切り離し動作終了**

[20:52:11] Sequence: 'navigating' > 'magnet_calibrating' → **八の字に旋回し地磁気センサを調整する状態**

MagnetCalibrating: Turning right...
MagnetCalibrating: Turning left...
MagnetCalibrating: Turning stopped!
min: (8.85, -24.75, -51.6)
max: (64.5, 37.05, -6.75)
Command:
nineaxis minmax 8.85 -24.75 -51.6 64.5 37.05 -6.75

filtered min: (10.8, -23.85, -48.9) → **正規化後の最小値**
filtered max: (63.45, 35.4, -8.1) → **正規化後の最大値**
Command:

nineaxis minmax 10.8 -23.85 -48.9 63.45 35.4 -8.1

MagnetCalibrating has been finished!

[20:52:26] Sequence: 'navigating' → **運搬物回収地点へのナビゲーションの開始**

Time: 20:52:26
GPS: (35.64310167,139.52329000) → 現在のGPS座標
target coord:(35.64313073,139.52337503) → 目標のGPS座標
Target Azimuth: 67.189 → ゴールへの方角計算
Current Azimuth: 142.438 → 現在の方角
Goal Distance: 8.345 m → 残りゴールへの距離を計算

Time: 20:52:28
GPS: (35.64310000,139.52328833)
target coord:(35.64313073,139.52337503)
Target Azimuth: 66.435
Current Azimuth: 117.912
Goal Distance: 8.557 m

(中略)

Time: 20:52:51
GPS: (35.64310167,139.52335500)
target coord:(35.64313073,139.52337503)
Target Azimuth: 29.252
Current Azimuth: 42.986
Goal Distance: 3.708 m

navigation goal has achieved → 運搬物回収地点に到達

Time: 20:52:52
GPS: (35.64310333,139.52335667)
target coord:(35.64313073,139.52337503)
Target Azimuth: 28.577
Current Azimuth: 45.306
Goal Distance: 3.474 m

finished init USB accelerator

[20:52:52] Sequence: 'object_detecting' → 運搬物の探索
get image → カメラから画像を取得(検知失敗)
height : 768width : 768 → 768x768の画像を取得
TURN MOVE powerL : 0.6 powerR : 0.2 → 時計回りに旋回して探索

<中略>

Object found! (rate: 0.0618)conf : 0.557GET LABEL :2 → 運搬物を画像内に発見. rate:画像内の割合,
conf:信頼度, LABEL;目標物体のカテゴリカル変数
get :right!! Object azimuth = 83.4 (current) + 30 (BB) = 113 → 画像右方向に運搬物を検知(画像と方位の
対応)
get image → カメラから画像を取得(検知失敗)
height : 768width : 768
getOutputData : 100 7
TURN!!

<中略>

[21:02:53] Sequence: 'navigating' > 'magnet_calibrating' → フェイルセーフ発動(物体検知開始から10分経過)

<中略>

[21:03:05] Sequence: 'navigating' → 運搬物設置地点(ゴールコーン)へのナビゲーション
Time: 21:03:05
GPS: (35.64310500,139.52338833)
target coord:(35.64325030,139.52344942)

Target Azimuth: 18.863
Current Azimuth: 186.067
Goal Distance: 17.093 m

<中略>

navigation goal has achieved → 運搬物設置地点(ゴールコーン)へ到着
Time: 21:04:00
GPS: (35.64323500,139.52341167)
target coord:(35.64325030,139.52344942)
Target Azimuth: 63.495
Current Azimuth: 49.993
Goal Distance: 3.817 m

[21:04:00] Sequence: 'object_setting' → 運搬物設置
setting finished! → 運搬物設置完了
[21:04:03] Sequence: 'object_detecting' → **ゴールコーン検知開始**
get image → 運搬物を画像内に発見

<中略>

TURN MOVE powerL : 0.6 powerR : 0.2 → コーン検知のため旋回

<中略>

Object found! (rate: 0.188)conf : 0.507GET LABEL :0 → 画像内にゴールコーン検知
label is : 1check_BB rate is : 0.603
go straight!! → 直進
Object approaching Goal [or Goal if object is goal cone]! → 画像内のBBサイズが閾値以上
Object detecting has finished !! → **ゴールコーン検知(及びゴールコーンへの接近)終了**

○ 結論

ミッション後, 規定された制御履歴レポートを提出するため, 得られたログデータから制御履歴レポートを作成できることが確認された.

8. ミッション要求を満たすための試験内容

(V12) 着地衝撃試験

- 実施日: 8/21
- 実施責任者: 前川
- 充足要求項目: M1, M2
- 目的
 - 着地の衝撃によってCanSatに異常が生じないことを確認する.
- 試験/解析内容
 - V7.落下試験で算出したパラシュートを用いて落下した時の最大の終端速度は7.00m/sである. この終端速度をもとに本番で想定される着地時の衝撃を与えるために必要な高さを以下の式で求める.

$$\frac{1}{2}mv^2 = mgh \Leftrightarrow h = \frac{1}{2g}v^2$$

$$h = \frac{1}{2*9.8} * 7.0^2 \approx 2.5[m]$$

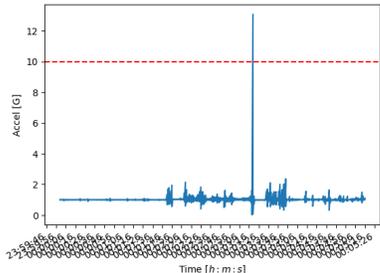
以上より本試験ではCanSatを2.5mから自由落下させることで、本番同等の着地衝撃を与える。待機状態からプログラムを開始し、着地衝撃を与え、ナビゲーション状態まで人の手を介さずにシーケンスが移行できることを確認する。その後、全てのセンサの動作確認、CanSatに破損がないかを確認する。

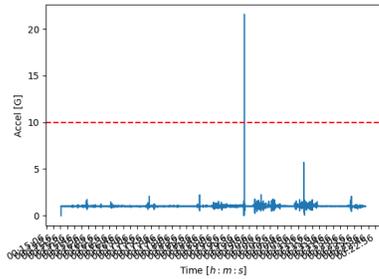
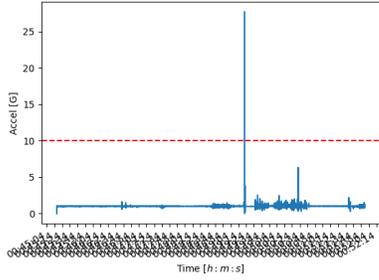
センサ・動力系がに対しては6章冒頭で述べた条件を満たしていれば正常に動作しているとみなす..

○ 結果

- 実験結果を以下の表6.10に示す。動力系・センサ類が6章冒頭で示した条件を満たすように動作した場合は○、動作しなかった場合は×と表記する。加速度センサの取得値グラフは、横軸は経過時間[s]、縦軸は加速度の大きさ(ノルム)が重力加速度の何倍であるかを示す。グラフ内の赤の点線は10Gを示す。

表6.10 着地衝撃試験結果

回数	実験動画	結果			外観
		動力系	センサ		
			その他センサ	加速度センサ取得値	
1	https://youtu.be/BaolDoSSmuM	タイヤ用モータ: ○ スタビライザー用・アーム用サーボモータ: ○	GPSセンサ: ○ 光センサ: ○ 気圧センサ: ○	加速度の大きさ(ノルム)が着地衝撃試験前と後で変わっていない 加速度センサ値を用いて反転・横転復帰可能 	破損なし
2	https://youtu.be/njRcKUAFp9g	タイヤ用モータ: ○ スタビライザー用・アーム用サーボモータ: ○	GPSセンサ: ○ 光センサ: ○ 気圧センサ: ○	加速度の大きさ(ノルム)が着地衝撃試験前と後で変わっていない 加速度センサ値を用いて反転・横転復帰可能	

				 <p>Accel [G]</p> <p>Time [h:m:s]</p>
3	https://youtu.be/i24sDmTbitI	タイヤ用モータ: ○ スタビライザー用・アーム用サーボモータ: ○	GPSセンサ: ○ 光センサ: ○ 気圧センサ: ○	加速度の大きさ(ノルム)が着地衝撃試験前後で変わっていない 加速度センサ値を用いて反転・横転復帰可能  <p>Accel [G]</p> <p>Time [h:m:s]</p>

- 結論
 - 着地の衝撃がかかっても、外観状態、センサ値の取得、動力系の動作に問題がない。

(V13) 電力耐久試験

- 実施日: 8/25
- 実施責任者: 前川
- 充足試験項目: M3
- 目的
 - CanSatにミッションを十分に遂行できるバッテリーが搭載されていることを確認する。
- 試験/解析内容
 - 草地でCanSatを走行させる。本大会の競技時間である20分を超えて走行できるよう、多めに見積もって本試験は1時間実施する。
 - 試験開始時と試験終了時(1時間後)のCanSatのモータやサーボの動きを比較し、変わらないことを確認する。
- 結果
 - 試験前の電力と試験後の電力を以下の表6.11に示す。

表6.11 電力耐久試験の結果

	回路用電源 (リチウムイオンポリ マー電池)	モータ・運搬機構サーボ モータ電源 (乾電池)	スタビライザ用サーボ モータ電源
試験前	4.08V	10.93V	8.14V
試験後	3.81V (-0.27V)	9.41V (-1.52V)	7.99V(-0.15V)

電力耐久試験の様子を以下の動画で示す。

<https://youtu.be/s6SEsyR4xIM>

試験開始時と試験終了時(1時間後)のCanSatのモータやサーボの動きを比較しても、目立った違いはみられない。

- 結論
本大会の競技時間は20分であるため、現バッテリー容量でミッションを十分に遂行可能である。

(V14) 反転・横転復帰試験

- 実施日：8/13
- 実施責任者：前川
- 充足要求項目：M4
- 目的
 - CanSatが走行中に反転・横転をした時に、走行可能な姿勢に復帰できることを確認する。
- 試験/解析内容
 - 反転復帰
 - CanSatを反転させた状態から反転復帰シーケンスのプログラムを実行し、走行可能な姿勢に復帰させる。
 - 横転復帰
 - CanSatを横転させた状態から横転復帰シーケンスのプログラムを実行し、走行可能な姿勢に復帰させる。
- 結果
 - 反転復帰
 - 実験結果を以下の表6.12に示す。

表6.12 反転復帰試験の結果

回数	実験動画	結果
1	https://youtu.be/w09uW-SJ-Ik	走行姿勢に復帰できた
2	https://youtu.be/bmcEh0Samg0	走行試験に復帰できた
3	https://youtu.be/-acS5Phae0k	走行試験に復帰できた

■ 横転復帰

- 実験結果を以下の表6.13に示す.

表6.13 横転復帰試験の結果

回数	実験動画	結果
1	右 https://youtu.be/D_B8Q8pT0qE 左 https://youtu.be/zjG0QnWad_Q	左右どちらに反転していても走行姿勢に復帰できた
2	右 https://youtu.be/FmLk_UALs1k 左 https://youtu.be/T-YQDjgS6n0	左右どちらに反転していても走行姿勢に復帰できた
3	右 https://youtu.be/0md7CGfMhn0 左 https://youtu.be/LkBrmYOU-2g	左右どちらに反転していても走行姿勢に復帰できた

○ 結論

- CanSatが左右どちらに横転しても、また反転しても走行可能な姿勢に復帰可能である.

(V15) 物体検知および運搬試験

- 実施日: 10/22-23

- 実施責任者: 前川

- 充足要求項目: M5, M6, M8

- 目的

■ 以下の性能をそれぞれ検証する.

- 物体検知性能の検証

CanSatが物資の運搬に必要な物体検知精度を有しており、カメラを用いた物体検知で物資を収集可能な距離まで近づき、運搬機構を用いて運搬物を挟むことが可能であることを確認する。

また、確実な運搬を実現するため、運搬物のロストを検知し、再度検知できることを確認する。

- 運搬性能の検証

運搬機構で運搬物を把持したまま走行可能であることを確認する。

- 試験/解析内容

- 運搬物と離れた地点から物体検知・接近し、運搬機構を用いて運搬物を把持、運搬物設置地点まで走行し、運搬物を放出する。本試験では、本番で用いる立方体を運搬物とする。

- また、物体を一度把持したのち、物体を一度取り上げ、再度運搬物を把持する。

○ 結果

実験結果を以下の表6.14、表6.15に示す。

表6.14 立方体の物体検知および運搬の結果

回数	実験動画	結果
1	https://youtu.be/AgtcAEb6Ez0	運搬物を検知し、収集・運搬・放出ができた
2	https://youtu.be/0LXwUYNXaPk	運搬物を検知し、収集・運搬・放出ができた
3	https://youtu.be/PLwgLk2TSEk	運搬物を検知し、収集・運搬・放出ができた

表6.15 ロスト検知の結果

回数	実験動画	結果
1	https://youtu.be/sEfJHfXCPQg	運搬物を把持した後ロストしても、再度運搬物を検知し、把持できた。
2	https://youtu.be/jaAGlf2or7g	運搬物を把持した後ロストしても、再度運搬物を検知し、把持できた。
3	https://youtu.be/oON7qvP1Pfk	運搬物を把持した後ロストしても、再度運搬物を検知し、把持できた。

(V16) ゴール検知試験

- 実施日: 8/27
- 実施責任者: 谷津
- 充足要求項目: M7
- 目的
 - CanSatがゴールに到達するために、ゴールコーンをカメラで検知しゴール判定可能な距離まで近づけることを確認する。
- 試験/解析内容
 - ゴールのGPS座標を設定し、その地点にゴールコーンを設定する。その後、離れた地点からCanSatがGPS座標とカメラによる物体検知を用いてゴール検知可能な距離まで近づけることを確認する。
- 結果
 - 実験結果を以下の表6.16に示す。

表6.16 ゴール検知試験の結果

回数	実験動画	ゴールからの距離
1	https://youtu.be/-ACdI_X4NbA	0m
2	https://youtu.be/m9ooqk3eYyE	0m
3	https://youtu.be/kfif7b85_y0	0m

○ 結論

- CanSatがGPS座標とカメラによる物体検知を用いてゴール検知可能である.
-

第7章 工程管理, ガントチャート

以下にCarryberの使用するガントチャートのリンクを掲載する. 項目は左からタスク内容・担当者・タスク重要度・開始日・終了予定日が記載されている.

[ガントチャート](#)

- チーム全体

<input type="checkbox"/>	本審査用実験				
<input type="checkbox"/>	振動試験				
<input type="checkbox"/>	能代試験項目の実験				
<input type="checkbox"/>	合同気球試験				
<input type="checkbox"/>	能代本審査書作成				
<input checked="" type="checkbox"/>	質量試験	全員	高	10/03	10/10
<input checked="" type="checkbox"/>	キャリア収納・放出試験	ハード班	高	10/03	10/17
<input checked="" type="checkbox"/>	開傘衝撃試験	ハード班	高	10/03	10/10
<input checked="" type="checkbox"/>	走行試験	全員	高	8/01	8/15
<input checked="" type="checkbox"/>	運搬試験	全員	高	10/10	10/17
<input checked="" type="checkbox"/>	物体検知試験	ソフト班	高	10/10	10/17
<input checked="" type="checkbox"/>	ゴール検知試験	ソフト班	高	10/10	10/17
<input checked="" type="checkbox"/>	反転・横転復帰試験		高	10/03	10/10
<input type="checkbox"/>	ACTS本審査書作成	全員	高	10/1	10/17
<input type="checkbox"/>	ACTS本審査_先生の所見をいただく	全員	高	10/17	10/24
<input type="checkbox"/>	チェックリストの作成	全員	高	10/24	11/01
<input type="checkbox"/>	End-to-End試験	全員	高	10/10	11/18
<input type="checkbox"/>	FM制作期間, 完成メ(予備パーツを含む)	全員	高	10/1	11/01
<input type="checkbox"/>	FM動作確認	全員	高	10/01	11/01
<input type="checkbox"/>	持ち物リスト作成	全員	高	11/01	11/12

- ハードウェア

ハード					
<input checked="" type="checkbox"/>	運搬機構構想・制作	ハード班	高	05/03	5/22
<input checked="" type="checkbox"/>	加工場講習 for B1	ハード班	中		
<input checked="" type="checkbox"/>	モータ選定	前川	高	05/10	5/17
<input checked="" type="checkbox"/>	BBM作成・完成	ハード班	高	05/17	5/28
<input checked="" type="checkbox"/>	タイヤ用スポンジ買い出し	ハード班	中	05/30	5/30
<input checked="" type="checkbox"/>	運搬機構実験・確定	ハード班	高	05/31	6/13
<input checked="" type="checkbox"/>	タイヤ設計・制作(木とスポンジ)	ハード班	高	6/7	6/13
<input checked="" type="checkbox"/>	パラシュート切り離し機構作成	ハード班	高	06/07	6/18
<input checked="" type="checkbox"/>	EM作成・完成	ハード班	高	06/06	6/13
<input checked="" type="checkbox"/>	実験を繰り返し品質・信頼性検証(設計確認・見直)	ハード班	高	06/14	7/11
<input checked="" type="checkbox"/>	GM完成(設計確定)	ハード班	高	07/11	7/18
<input checked="" type="checkbox"/>	車軸制作	ハード班	高	06/08	7/04
<input checked="" type="checkbox"/>	スタビライザ(後輪)制作	ハード班	高	06/08	7/04
<input checked="" type="checkbox"/>	シャーシ制作	ハード班	高	06/08	7/04
<input checked="" type="checkbox"/>	タイヤ制作	ハード班	高	06/08	7/04
<input checked="" type="checkbox"/>	タイヤスパイク選定	ハード班	高	06/08	8/01
<input checked="" type="checkbox"/>	モータホルダ制作	ハード班	高	06/08	7/04
<input checked="" type="checkbox"/>	PM完成	ハード班	高	06/08	7/04
<input checked="" type="checkbox"/>	電池ボックス作成	ハード班	高	07/04	7/11
<input type="checkbox"/>	2機目完成	ハード班	高	07/25	8/06
<input checked="" type="checkbox"/>	最終版スタビライザ(後輪)完成	ハード班	高	07/25	8/06
<input type="checkbox"/>	予備部品量産	ハード班	高	08/15	10/30
<input type="checkbox"/>	組み立て手順書の作成	ハード班	高	8/15	10/30
<input type="checkbox"/>	ハードチェックリストの作成	ハード班	高	8/15	10/30
<input type="checkbox"/>	新品ネジ・イモネジの見積もりと発注	前川	高	8/15	10/30

- 回路

回路					
<input checked="" type="checkbox"/>	センサの選定	回路班	▼ 中 ▼	05/02	05/03
<input checked="" type="checkbox"/>	センサの動作確認	回路班	▼ 中 ▼	5/02	5/03
<input checked="" type="checkbox"/>	拡張回路のセットアップ	回路班	▼ 中 ▼	5/02	5/03
<input checked="" type="checkbox"/>	回路図ver1 作成	回路班	▼ 中 ▼	5/02	5/12
<input checked="" type="checkbox"/>	配置図ver1 作成	回路班	▼ 中 ▼	5/02	5/12
<input checked="" type="checkbox"/>	ユニバーサル基板実装(配置決め、ピンソケット)	回路班	▼ 高 ▼	5/12	5/21
<input checked="" type="checkbox"/>	ユニバーサル基板実装(GND、電源の配線)	回路班	▼ 高 ▼	5/24	5/28
<input checked="" type="checkbox"/>	ユニバーサル基板実装(信号線の配線)	回路班	▼ 高 ▼	5/27	5/28
<input checked="" type="checkbox"/>	ユニバーサル基板実装(動作確認)	回路班	▼ 高 ▼	5/28	5/28
<input checked="" type="checkbox"/>	距離センサ実装、動作確認	回路班	▼ 高 ▼	6/01	6/10
<input checked="" type="checkbox"/>	回路図ver2 作成	回路班	▼ 中 ▼	5/28	6/14
<input checked="" type="checkbox"/>	配置図ver2 作成	回路班	▼ 中 ▼	5/28	6/14
<input checked="" type="checkbox"/>	print基板設計、発注	回路班	▼ 高 ▼	6/06	7/01
<input checked="" type="checkbox"/>	print基板作成	回路班	▼ 高 ▼	6/14	07/15
<input checked="" type="checkbox"/>	サーボコネクタ付け替え	回路班	▼ 中 ▼	07/01	08/01
<input checked="" type="checkbox"/>	カメラ・USBアクセラレータ用コネクタ作成	回路班	▼ 高 ▼	8/10	08/14
<input checked="" type="checkbox"/>	後方サーボ用小型回路作成	回路班	▼ 高 ▼	8/01	08/14
<input checked="" type="checkbox"/>	電池ボックス量産	回路班	▼ 低 ▼	8/14	08/31
<input checked="" type="checkbox"/>	新基盤回路図・配置図作成	回路班	▼ 高 ▼	9/1	09/14
<input checked="" type="checkbox"/>	新基盤作成・動作確認	回路班	▼ 高 ▼	10/04	10/08
<input type="checkbox"/>	新基盤量産	回路班	▼ 中 ▼	10/04	10/31

- ソフトウェア

ソフト					
<input checked="" type="checkbox"/>	led_matrixの削除	谷津	中	04/27	5/04
<input checked="" type="checkbox"/>	ディレクトリ名の変更	堀越	低	05/10	6/15
<input checked="" type="checkbox"/>	物体検知検証1	千住	高	04/27	5/21
<input type="checkbox"/>	アスキーアート変更or削除		低	06/16	8/22
<input checked="" type="checkbox"/>	カルマンフィルタの実装	千住	高	05/11	5/16
<input checked="" type="checkbox"/>	c++からpython呼び出し検証	谷津	中	05/20	6/11
<input checked="" type="checkbox"/>	SSD用学習データ叩き台	谷津	中	05/16	5/23
<input checked="" type="checkbox"/>	物体検知性能検証2(1で速度は検証済、精度検証)	谷津	高	05/16	5/21
<input checked="" type="checkbox"/>	ライブラリ整理	谷津	高	06/07	6/11
<input checked="" type="checkbox"/>	ゴール地点gps座標取得の自動化	千住	高	05/17	6/06
<input checked="" type="checkbox"/>	運搬物確定	全員	高	06/13	6/21
<input checked="" type="checkbox"/>	教師データ作成	全員	高	06/21	6/27
<input checked="" type="checkbox"/>	教師データクリーニング	堀越	中	06/27	7/2
<input checked="" type="checkbox"/>	新モデル作成&検証	谷津	高	06/27	7/2
<input checked="" type="checkbox"/>	物体検知距離測定	ソフト班	高	07/05	7/15
<input checked="" type="checkbox"/>	object detector作成	谷津	高	06/27	7/30
<input checked="" type="checkbox"/>	運搬物用ナビゲーションシーケンス	ソフト班	高	07/02	7/30
<input checked="" type="checkbox"/>	ミッションフロー動作確認	千住	高	08/05	8/18
<input type="checkbox"/>	モデル精度向上	ソフト班	高	07/15	8/22
<input type="checkbox"/>	catch時等のハイパラや動作調整	ソフト班	高	08/15	8/22
<input type="checkbox"/>	ディスクイメージ作成	ソフト班	高	08/01	8/22

第8章 大会結果報告

1. 目的

開発したCanSatの有効性を検証するため、キャリア放出判定からゴール判定およびミッション達成までのシーケンスを自律して制御できるかを検証するため

2. 結果

ACTS	パラシュート開 傘	壊れずに 着地	パラシュート分 離	走行開始	物体開始地 点到達	物体収集	半径10cm 以内にゴール
1回目	○	○	×	○	×	×	×
2回目	○	○	○	○	○	×	×

- 投下動画
 - 1回目
 - <https://youtu.be/bG0pzvD08QY>
 - <https://youtu.be/5FFuxUSkLZw>
 - 2回目
 - <https://youtu.be/XdmyCB7osMA>

- 故障原因説明・解決手段等
 - 1回目

落下時にパラシュート切り離しに失敗し、CanSatとパラシュートの分離ができなかった。パラシュートの取付に問題があったことと、草でタイヤが空転し、切り離し動作が十分にできなかったことが失敗の原因である。

パラシュートの取付に関しては事前実験では失敗はほぼない状態だったが、本番の気温が低い状況で精密な作業に支障がでたため、パラシュートを留めるピンが抜けなかったことが最も大きな要因であると考えられるため、コンディションを整えた状態で本番に臨むことが重要である。また、どのような状況でもしっかりと動作する頑健な機構を作成することが重要となる。

 - 2回目

パラシュート切り離し後に運搬物の設置場所付近まで移動したが、運搬物への接近中に旋回動作を挟んだ際に、遠距離の人混みを運搬物(Box)と誤検知してしまい、運搬物から離れた後、物体を3回連続で検知した際に物体が確実に近くにあると判断しその場での旋回動作を続けるプログラムが発動し、大会制限時間に到達してしまった。

この原因としては、誤検知が大きな要因であったが、ほぼ連続した誤検知はないものとしてプログラムを組んでいたために、その場での旋回から抜け出せなくなったことも要因となっている。連続して異なる物体を誤検知してしまった際に本来の物体の探索へと復帰するようなアルゴリズムが必要であった。また、物体検知シーケンスが長引いた場合に物体収集をあきらめ、ゴール到達を優先するようなプログラム(フェイルセーフ)になっていたが、大会の制限時間内にこのアルゴリズムが発動しなかったため、ゴールへの到達も時間切れとなった(図8.1の橙色矢印地点)。大会の制限時間超過後に、CanSatに人の手が触れることなく0mゴールを到達できたため、時間があればゴールはできていたことが分かる。これらのことから、このフェイルセーフの発動時間を大会に合わせてチューニングすることが十分にできていなかったため、各シーケンスのフェイルセーフのチューニングも重要である。

以下に、ナビゲーション軌跡及び制御履歴ログを示す。

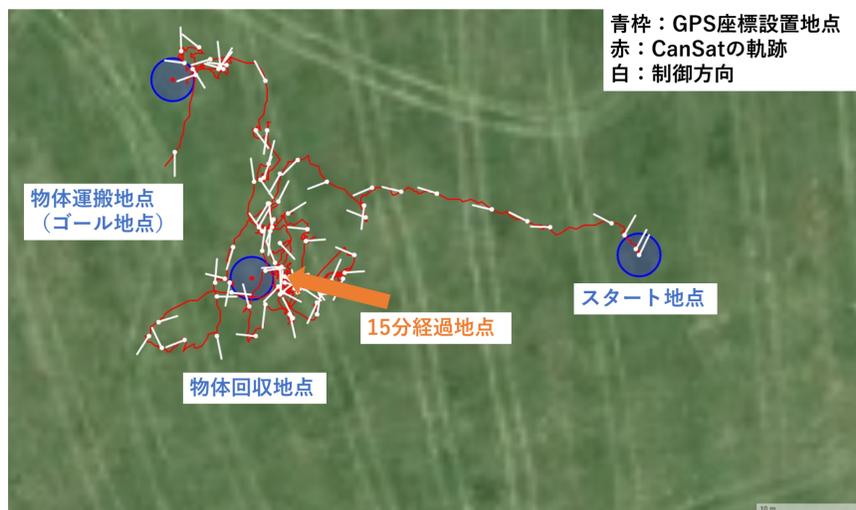


図8.1 ナビゲーション軌跡

制御履歴ログ

```

=== Team CarryBer ===

Takadama Lab.  ARLISS 2020

Waiting Mode :light    1 ->光センサを使用
Wi-Fi has been stopped!  >Wi-Fi off
Sequence:'waiting' ->待機状態
Time: 12:28:57
Light count: 0/10 (LOW)

Time: 12:28:58
Light count: 0/10 (LOW)

(略)

Time: 12:33:15
Light count: 9/10 (HIGH)

Wi-Fi has been restarted! >Wi-Fi on

Sequence:'falling' ->落下状態
[Time: 12:33:19
Pressure count: 1/10 (917.714 hPa) ->着地判定の条件1つ目(気圧センサによる判定). 10カウント

```

Gyro count: 1/10 ->着地判定の条件2つ目(ジャイロセンサによる判定). 10カウント
GPS position: 35.4128183 138.5932783

Time: 12:33:20
Pressure count: 2/10 (917.7507 hPa)
Gyro count: 2/10
GPS position: 35.4128200 138.5932883

(略)

Time: 12:33:26
Pressure count: 8/10 (917.747 hPa)
Gyro count: 8/10
GPS position: 35.4127983 138.59331

Time: 12:33:27
Pressure count: 9/10 (917.6626 hPa)
Gyro count: 9/10
GPS position: 35.4127967 138.5933117
Falling completed! (by gyro and pressure) ->着地判定完了

Sequence:para_separating' -> パラシュート切り離し状態開始
[0mPara separating... (1/20)
Para separating... (2/20)
(略)
Para separating... (19/20)
Para separating... (20/20)
Para separating finished! -> パラシュート切り離し状態完了

Sequence: para_separating' > 'init_waking' -> 反転復帰開始
Waking finished! -> 反転復帰完了
sequence: para_separating'

Sequence: 'navigating' > 'magnet_calibrating' -> 地磁気のキャリブレーション開始
MagnetCalibrating: Turning right...
MagnetCalibrating: Turning left...
MagnetCalibrating: Turning stopped!

min: (16.95 -32.4 -21)
max: (70.5 24.45 -3.45)
Command:
nineaxis minmax 16.95 -32.4 -21 70.5 24.45 -3.45

filtered min: (17.25 -30.75 -19.5)
filtered max: (68.4 20.4 -5.1)
Command:

nineaxis minmax 17.25 -30.75 -19.5 68.4 20.4 -5.1
MagnetCalibrating has been finished! -> 地磁気のキャリブレーション完了

Sequence: 'navigating' ->GPS座標に基づくサブゴールへのナビゲーション状態開始
GPS history of stuck_detection_by_gps has been reset because of subsequence interruption!

Time: 12:34:22
GPS: (35.41280333 138.59322500)
target coord:(35.41278363 138.59282614)
Target Azimuth: 266.531
Current Azimuth: 31.878
Goal Distance: 36.253 m

Time: 12:34:23
GPS: (35.41280500 138.59322500)
target coord:(35.41278363 138.59282614)
Target Azimuth: 266.239
Current Azimuth: 34.100
Goal Distance: 36.265 m

(略)

Sequence: 'navigating' > 'escaping' -> エスケイピング動作の実行

(略)

Sequence: 'navigating' ->GPS座標に基づくサブゴールへのナビゲーション状態再開
GPS history of stuck_detection_by_gps has been reset because of subsequence interruption!

Time: 12:34:51
GPS: (35.41282167 138.59320833)
target coord:(35.41278363 138.59282614)
Target Azimuth: 263.038
Current Azimuth: 3.301
Goal Distance: 34.932 m

Time: 12:34:52
GPS: (35.41282333 138.59320833)
target coord:(35.41278363 138.59282614)
Target Azimuth: 262.736
Current Azimuth: 319.949
Goal Distance: 34.955 m

(略)

Time: 12:36:21
GPS: (35.41285500 138.59294667)
target coord:(35.41278363 138.59282614)
Target Azimuth: 233.999

Current Azimuth: 256.188
Goal Distance: 13.517 m

SandMountain escaping.
Time: 12:36:22

GPS: (35.41285333 138.59294333)
target coord:(35.41278363 138.59282614)
Target Azimuth: 233.879
Current Azimuth: 272.212
Goal Distance: 13.162 m

Sequence: 'navigating' > 'turning_and_escaping_sand_mountain' -> エスケープ動作の実行

Sequence: 'navigating' ->GPS座標に基づくサブゴールへのナビゲーション状態再開
GPS history of stuck_detection_by_gps has been reset because of subsequence interruption!

Time: 12:36:39
GPS: (35.41283167 138.59294500)
target coord:(35.41278363 138.59282614)
Target Azimuth: 243.624
Current Azimuth: 353.045
Goal Distance: 12.037 m

Time: 12:36:40
GPS: (35.41283167 138.59294500)
target coord:(35.41278363 138.59282614)
Target Azimuth: 243.624
Current Azimuth: 278.875
Goal Distance: 12.037 m

(略)

Sequence: 'navigating' > 'escaping' -> エスケープ動作の実行

(略)

Sequence: 'navigating' >GPS座標に基づくサブゴールへのナビゲーション状態再開
GPS history of stuck_detection_by_gps has been reset because of subsequence interruption!

Time: 12:37:23
GPS: (35.41287167 138.59285500)
target coord:(35.41278363 138.59282614)
Target Azimuth: 194.958
Current Azimuth: 201.736
Goal Distance: 10.144 m

Time: 12:37:24
GPS: (35.41286833 138.59285500)
target coord:(35.41278363 138.59282614)

Target Azimuth: 195.519
Current Azimuth: 201.482
Goal Distance: 9.786 m

(略)

Time: 12:37:45
GPS: (35.41281667 138.59280833)
target coord:(35.41278363 138.59282614)
Target Azimuth: 156.285
Current Azimuth: 204.105
Goal Distance: 4.018 m

navigation goal has achieved ->GPS座標に基づくサブゴールへのナビゲーション状態完了

Time: 12:37:46
GPS: (35.41281333 138.59280667)
target coord:(35.41278363 138.59282614)
Target Azimuth: 151.884
Current Azimuth: 200.864
Goal Distance: 3.749 m

finished init USB accelerator
Sequence: 'object_detecting' ->物体検知状態開始
get image

WIPER TURN!! count 0
TURN WIPER powerL : -0.4 powerR : 0.4 ->首振り動作による箱へのアプローチ

(略)

get image
check param m_hasCarried : false find_count : 0 move_count : 0
TURN MOVE powerL : 0.7 powerR : 0.35 ->旋回移動による箱への接近アプローチ

(略)

restart navigation >検知可能範囲に箱がなかったため、再度ナビゲーションから探索開始

TURN MOVE powerL : 0.7 powerR : 0.35
Sequence: 'navigating' > 'magnet_calibrating' -> 地磁気のキャリブレーション開始
MagnetCalibrating: Turning right...
MagCalib stuck detection result: [OK -- -- --] (azimuth:40)
MagCalib stuck detection result: [OK OK -- --] (azimuth:91.6)
MagCalib stuck detection result: [OK OK OK --] (azimuth:181)
MagCalib stuck detection result: [OK OK OK OK] (azimuth:272)
MagnetCalibrating: Turning left...
MagnetCalibrating: Turning stopped!
min: (13.2 -32.2 -25.1)
max: (69.3 25.8 -11.7)
Command:
nineaxis minmax 13.2 -32.2 -25.1 69.3 25.8 -11.7

```
-----  
filtered min: (14.8 -29.8 -22.6)  
filtered max: (67.6 24 -12.9)  
Command:  
nineaxis minmax 14.8 -29.8 -22.6 67.6 24 -12.9  
MagCalib stuck detection result: [ OK OK OK OK ]  
---> Rover is not stuck!  
MagnetCalibrating has been finished! -> 地磁気のキャリブレーション完了  
Sequence: 'navigating' ->GPS座標に基づくサブゴールへのナビゲーション開始  
GPS history of stuck_detection_by_gps has been reset because of subsequence interruption!
```

```
Time: 12:40:13  
GPS: (35.41274333 138.59272333)  
target coord:(35.41278363 138.59282614)  
Target Azimuth: 64.315  
Current Azimuth: 55.419  
Goal Distance: 10.350 m
```

```
Time: 12:40:14  
GPS: (35.41274500 138.59272500)  
target coord:(35.41278363 138.59282614)  
Target Azimuth: 64.890  
Current Azimuth: 72.316  
Goal Distance: 10.133 m
```

(略)

```
Time: 12:40:31  
GPS: (35.41277167 138.59280500)  
target coord:(35.41278363 138.59282614)  
Target Azimuth: 55.226  
Current Azimuth: 89.105  
Goal Distance: 2.336 m
```

navigation goal has achieved ->GPS座標に基づくサブゴールへのナビゲーション完了

```
Time: 12:40:32  
GPS: (35.41277000 138.59281167)  
target coord:(35.41278363 138.59282614)  
Target Azimuth: 40.873  
Current Azimuth: 90.921  
Goal Distance: 2.007 m
```

```
finished init USB accelerator  
Sequence: 'object_detecting' ->物体検知状態開始  
WIPER TURN!! count 0  
TURN WIPER powerL : -0.4 powerR : 0.4 >首振り動作による箱へのアプローチ
```

(略)

```
get image
check param m_hasCarried : false find_count : 0 move_count : 0
TURN MOVE powerL : 0.7 powerR : 0.4 ->旋回移動による箱への接近アプローチ
```

(略)

```
get image
0 0.6 2 267 402 357 485
Object found! (rate: 0.0127)conf : 0.6GET LABEL :2 ->誤検知
get :center!! Object azimuth = 120 (current) + 0 (BB) = 120
```

```
get image
0 0.528 2 280 475 370 560
Object found! (rate: 0.013)conf : 0.528GET LABEL :2 ->誤検知
get :center!! Object azimuth = 121 (current) + 0 (BB) = 121
```

```
get image
0 0.604 2 136 343 230 429
Object found! (rate: 0.0137)conf : 0.604GET LABEL :2 ->誤検知
get :left!! Object azimuth = 123 (current) + -30 (BB) = 92.5
```

(略)

get image ->三回物体検知を行ったため、近くに物体があるとみなし、首振りと回転による接近アプローチをフェイルセーフするまで行う

```
pre azimuth[94.1 121 145
WIPER TURN!! count 0
TURN WIPER powerL : -0.4 powerR : 0.4
```

```
get image
pre azimuth[121 145 132
WIPER TURN!! count 1
TURN WIPER powerL : -0.4 powerR : 0.4
```

(略)

---競技時間15分経過---

```
get image
check param m_hasCarried : false find_count : 12 move_count : 174
pre azimuth[115 127 139
WIPER TURN!! leftTurn
TURN WIPER powerL : 0.6 powerR : -0.6
```

```
get image
check param m_hasCarried : false find_count : 12 move_count : 175
pre azimuth[127 139 157
WIPER TURN!! leftTurn
TURN WIPER powerL : 0.6 powerR : -0.6
```

Sequence: 'navigating' > 'magnet_calibrating' ->フェイルセーフ, 箱を諦め, コーンへのナビゲーション開始

MagnetCalibrating: Turning right... -> 地磁気のキャリブレーション開始

MagCalib stuck detection result: [-- -- OK --] (azimuth:255)

MagCalib stuck detection result: [-- -- OK OK] (azimuth:273)

MagCalib stuck detection result: [OK -- OK OK] (azimuth:2.54)

MagCalib stuck detection result: [OK OK OK OK] (azimuth:90)

MagnetCalibrating: Turning left...

MagnetCalibrating: Turning stopped!

min: (13 -34.4 -31.9)

max: (69.8 24.3 -10.3)

Command:

nineaxis minmax 13 -34.4 -31.9 69.8 24.3 -10.3

filtered min: (15.4 -31.5 -26.1)

filtered max: (67.5 21.4 -14.1)

Command:

nineaxis minmax 15.4 -31.5 -26.1 67.5 21.4 -14.1

MagCalib stuck detection result: [OK OK OK OK]

---> Rover is not stuck!

MagnetCalibrating has been finished! -> 地磁気のキャリブレーション完了

Sequence: 'navigating' ->GPS座標に基づくメインゴールへのナビゲーション状態開始

GPS history of stuck_detection_by_gps has been reset because of subsequence interruption!

Time: 12:50:44

GPS: (35.41275333 138.59289333)

target coord:(35.41295092 138.59274456)

Target Azimuth: 328.465

Current Azimuth: 347.366

Goal Distance: 25.806 m

Time: 12:50:45

GPS: (35.41275167 138.59289333)

target coord:(35.41295092 138.59274456)

Target Azimuth: 328.679

Current Azimuth: 340.211

Goal Distance: 25.965 m

(略)

Time: 12:51:28

GPS: (35.41283000 138.59283167)

target coord:(35.41295092 138.59274456)

Target Azimuth: 329.583

Current Azimuth: 36.816

Goal Distance: 15.609 m

Time: 12:51:29
GPS: (35.41282833 138.59283167)
target coord:(35.41295092 138.59274456)
Target Azimuth: 329.924
Current Azimuth: 35.461
Goal Distance: 15.769 m

Sequence: 'navigating' > 'escaping' -> エスケイピング動作の実行

(略)

Sequence: 'navigating' ->GPS座標に基づくメインゴールへのナビゲーション状態再開
GPS history of stuck_detection_by_gps has been reset because of subsequence interruption!

Time: 12:51:33
GPS: (35.41282167 138.59283667)
target coord:(35.41295092 138.59274456)
Target Azimuth: 329.853
Current Azimuth: 303.712
Goal Distance: 16.639 m

Time: 12:51:34
GPS: (35.41282333 138.59283667)
target coord:(35.41295092 138.59274456)
Target Azimuth: 329.529
Current Azimuth: 317.517
Goal Distance: 16.479 m

(略)

Sequence: [32m'navigating' > 'waking' -> 反転復帰動作の開始
Waking finished! -> 反転復帰動作の完了

(略)

Sequence: 'navigating' ->GPS座標に基づくメインゴールへのナビゲーション再開
[0mGPS history of stuck_detection_by_gps has been reset because of subsequence interruption!

Time: 12:52:55
GPS: (35.41290833 138.59283500)
target coord:(35.41295092 138.59274456)
Target Azimuth: 300.018
Current Azimuth: 64.921
Goal Distance: 9.477 m

Time: 12:52:56
GPS: (35.41291667 138.59283333)
target coord:(35.41295092 138.59274456)
Target Azimuth: 295.335

Current Azimuth: 27.806
Goal Distance: 8.911 m

Time: 12:53:19

GPS: (35.41296333 138.59277667)
target coord:(35.41295092 138.59274456)
Target Azimuth: 244.622
Current Azimuth: 262.821
Goal Distance: 3.223 m

(略)

navigation goal has achieved ->GPS座標に基づくメインゴールへのナビゲーション完了

Time: 12:53:20

GPS: (35.41296167 138.59277333)
target coord:(35.41295092 138.59274456)
Target Azimuth: 245.378
Current Azimuth: 248.650
Goal Distance: 2.872 m

Sequence: 'object_setting' ->物体の設置動作開始

(略)

setting finished! ->物体の設置動作完了

finished init USB accelerator

Sequence: 'object_detecting' ->メインゴールへの物体検知状態開始

get image

0 0.58 0 503 373 601 520

Object found! (rate: 0.0244)conf : 0.58GET LABEL :0

get :right!! Object azimuth = 247 (current) + 30 (BB) = 277 ->メインゴール検知

get image

0 0.514 0 176 273 284 442

Object found! (rate: 0.0309)conf : 0.514GET LABEL :0

get :left!! Object azimuth = 276 (current) + -30 (BB) = 246

(略)

get image

0 0.555 0 1 187 400 585

Object found! (rate: 0.269)conf : 0.555GET LABEL :0

get :left!! Object azimuth = 221 (current) + -30 (BB) = 191

get image

too close cone! ->至近距離判定

get image

too close cone! ->至近距離判定

```

get image
too close cone! ->至近距離判定
too close!! finish detecting!!! ->物体検知による接近完了

Sequence: 'Idling' ->ミツリヨ終了

```

3. 考察

- 1回目

サクセスクライテリア達成状況

ミニマムサクセス	フルサクセス	エクストラサクセス
×	×	×

原因は前述の通り, CanSatがパラシュートを切り離せなかったことにある. パラシュートの取付時に寒冷下で精密作業が十分にできていなかったことがこの要因となっており, 1日目の夜間に暖かい場所, 寒い場所で同様にパラシュート取付からの切り離しを複数回実験したところ, 暖かい場所では切り離しに失敗することはなかったが, 寒い場所では切り離しに失敗することがあった. これを解決するためには, 常にベストコンディションで作業をすることも重要だが, 何よりも, どのような状況でも動作する頑健な機構が求められる. 今回使用したパラシュート切り離し機構は, 金属パーツを人の手で切り出したものであるため, 個体差を無くすような精密な製作も重要である.

- 2回目

サクセスクライテリア達成状況

ミニマムサクセス	フルサクセス	エクストラサクセス
×	×	×

原因は前述の通り, まずCanSatが遠くの人混みを運搬物(箱)と誤検知したことにより本来の物体から離れてしまった. さらに, その後誤検知を3回連続で物体が確実に近くにあると判断しその場での旋回移動を繰り返すプログラムが発動し, そのまま時間切れとなった. ここでの要因は①物体の誤検知, ②誤検知した際の対応不足, ③緊急時のフェイルセーフの発動時間, の3点となる. ①の解決のためには予め箱に近い物体を箱と認識しないような物体検知モデルの作成が必要となる. ②の解決のためには仮に誤検知により近距離探索を繰り返すアルゴリズムに入っても, 対象が見つからなかった場合に再度対象の広域探索に戻るなどの対処が必要である. ③は大会の時間を加味してフェイルセーフに入り運搬物収集を諦めゴールへ向かうまでの時間を短くしていれば少なくともゴールは達成できフルサクセスも△とすることができた. 実際に競技時間終了後は, そのままCanSatに触れずに0mゴールを達成することができた. しかし, フェイルセーフの時間は短すぎると運搬物の収集の時間が足りなくなるため, CanSatの探索性能や競技時間を加味して慎重に設定

する必要がある。

第9章 まとめ

1. 工夫・努力したこと(ハード, ソフト, マネジメント面すべて)

【工夫・努力した点】

- ハード
 - 把持性能の高い長さのある運搬機構を担保するために、アームを折り畳み式にし、把持性能を維持しつつレギュレーションのサイズ内に収めた。
 - 走破性能を持ちつつ重量の少ない薄いタイヤを作成できた。
- 回路
 - センサやアクセラレータなどに応じて回路を設計し、レギュレーション内に収まるサイズで多機能な回路を作成した。
- ソフト
 - 小さな物体を草むらや不整地で検知するために学習時にデータにノイズやサイズの変更などの処理を加え、様々な環境に適応できる物体検知を実装した。

【苦勞した点】

- ハード
 - 落下衝撃に強い運搬機構の作成に苦勞した。(プロテクターにより解決した)
 - 加工場が平日日中しか使用できず、加工時間の確保に苦勞した。
- 回路
 - ハード面で変更があった際に回路を設計し直すのに苦勞した。
- ソフト
 - 大きな物体の検知は問題なかったが、小さな物体を検知するのに苦勞した。大きな物体と小さな物体の検知を両立させるのに苦勞した。

2. 良かった点・課題点

【良かった点】

- チーム
 - 機体に不具合が起きた時に原因を切り分けて的確に対処できていた。
- ハード班
 - 折り畳み式のアームなどの機構を簡単に量産できるように設計した。
- 回路班
 - 様々な外的要因(衝撃や荷重等)に対して頑強な回路を設計できた。
 - 故障に対して迅速に対応できた。

- ソフト班
 - 1日目投下後のプログラム修正で草地からのエスケージング問題を解決できた。

【課題点】

- ハード班
 - 手作りのパーツに個体差があった。
 - 機体の設計がなかなか固まらなかった。
- 回路班
 - 回路設計への理解が浅く、終始動作が安定しなかった。
 - 部品同士の干渉・機能の阻害等を考慮しきれていなかった。
- ソフト班
 - アルゴリズムの中身をチーム内で共有しきれていなかった。

3. 今後の展望

- COVID-19の影響により例年のスケジュール感が崩れ、全体的に出だしから開発が遅くなったが、来年以降に向けてまずは、スケジュールをしっかりと立てて計画通りに進めていくことが最も重要となる。2年続けて12月開催の大会となったため、夏の大会への大まかなスケジュールと直近の小スケジュールを見積もり、計画的に更に挑戦的なミッションを達成することを目指す。