

ARL ISS2022 報告書

提出日：2022 年 11 月 20 日

- 本審査書 1 次審査

- 提出期限：7 月 7 日

- 審査項目

- 第 3 章「要求項目の設定」

- システム要求の設定が打ち上げ場周辺および人員の安全確保及び設計/製作された機体の健全性の評価できているか

- ミッション要求の設定がミッション達成のための要求として適当であるか

- 第 6 章「試験項目」

- 「目的」と「試験内容」が書かれていて、要求項目を満たすための試験として有効であるか

- 本審査書 2 次審査

- 提出期限：8 月 8 日

- 審査項目

- 6[試験項目] 各実験の「試験結果」および「考察」が記入されていて、試験の結果から要求項目を満たしているといえるか

- End-to-end 試験でミニマムサクセスクライテリアが達成出来ていることを確認できるか

- 審査教員

審査教員氏名	
メールアドレス	
審査コメント	
その他一言	

- チーム情報

CanSat チーム名	東京情報大学 斎藤研究室 SuperNOVA
CanSat チーム 代表者情報	秋山実穂 Supernova.aut@gmail.com, 070-8384-5107
UNISEC 団体名	東京情報大学 斎藤研究室
UNISEC 団体 学生代表	秋山実穂
責任教員	斎藤卓也 T3saito22@rsch.tuis.ac.jp, 080-6625-5882
CanSat クラス	Open Class

- メンバー

役割	名前 (学年)
指導教員	斎藤卓也
PM, 機体, 回路, プログラム	秋山実穂

- CanSat の製作目的・大会参加理由

→CanSat を通じて勉強や研究を行うため

目次

第1章 ミッションステートメント	3
第2章 サクセスクライテリア	7
第3章 要求項目の設定	8
第4章 システム仕様	10
第5章 試験項目設定	17
第6章 実施試験の内容	18

第1章 ミッションステートメント

CanSat の走行時の振動の3軸加速度等を学習させ、
平坦な地面・凸凹な地面・スタック時を機械学習により推定し、その状態に適した速度やハンドル値の
走行制御を行う。さらに Object Detection を用いてリアルタイムでゴールを認識し 0m ゴールを目指す。

【振動を機械学習することによる状態推定】

CanSat カムバックの走行時には、様々な路面が想定される。例えば、ARLISS では割と平らで平坦な路面であり、所々轍が存在する。この場合、ゆっくり走ると、轍にはまりやすく、また路面にステアリングを取られてまっすぐ安定して走行するのが難しい。そのため、なるべく速いスピードに設定したほうが良い。逆に、能代宇宙イベントのような草の塊が多数存在するような凸凹した路面では、速いスピードで走行させると、草に乗り上げてひっくり返ってしまうことが多発し、安定して走行することができない。

また、スタック判定は GPS 位置情報を用いて判定しているが、この場合スタックしてはいないが、ゆっくりと草を乗り上げて左右に動いているような、単位時間当たりの移動距離が短い場合にスタックと誤判定されてしまうことがある問題がある。かといって、このような誤判定しないように移動距離を長く設定してスタック判定をなるべくしないようにすると、実際にスタックしたときに、なかなかスタック判定されない問題がある。

そこで CanSat の振動を利用し、加速度データから、ARLISS のような路面、能代のような路面等、様々な振動状態を学習することにより、CanSat 地面の状態を機械学習により推定し、その地面に適した走行を行うことで CanSat の従来より安定した走行制御を目指す。

振動データから、能代宇宙イベントなどの凸凹な地面を走行していると判定された場合は、走行速度が速いと機体が横転しやすいため速度を遅くして走行する。また凸凹でハンドルが取られるため、ハンドルのパラメータは大きく設定する。

同様に、振動データから ARLISS などの平坦な地面を走行していると判定された場合は、走行速度を速く設定し、ハンドルは大きく地面に取られないため小さくパラメータを設定する。時折轍が存在しても走行速度の速ければ、その勢いで乗り越えられるようになる可能性が高まる。また、振動データからスタックと判定されたときは、GPS 位置情報と合わせて判定することで、より確実にスタック判定ができる制御を目指す。

CanSat の振動を3軸加速度データおよび3軸ジャイロデータを、FFTにより time domain から frequency domain に変換し、それを学習させることにより、CanSat 走行時に機械学習により地面の状態推定を行う。

加速度データからの FFT 画像の例を Fig. 1 に示す。FFT 画像は縦軸 0-50Hz の周波数成分，横軸 0.1s ごとの 5 秒間の時間であり，50×50pixel カラー画像で表現される。画像の色及び明るさにより周波数成分の大きさを示しており，HSV の H を小さいときの青色から大きいときの赤色にかけ色変化により表現する。Fig. 1-1 では画像左側が CanSat 停止時の真っ青の状態であり，画像右側は走行開始により黄色や緑色を含んだ色変化が表れていることがわかる。

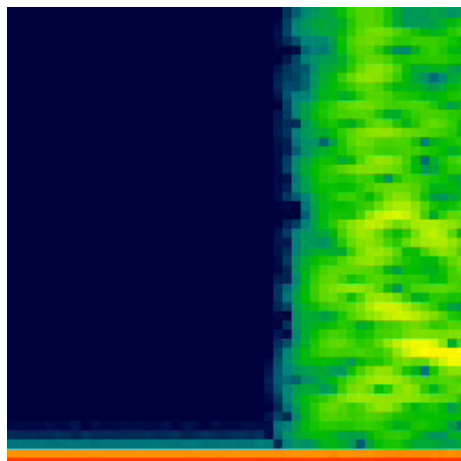


Fig. 1-1 FFT 画像の例

この FFT の画像を各状態における振動の学習データとして用いている。学習は FFT 画像での走行状態は 5 パターンを用意した。5 つの分類を下記の①ー⑤に示す。

- ① CanSat が完全に停止している状態の FFT 画像が真っ青の状態を halt と判定する。
- ② CanSat が走行開始時に FFT 画像が真っ青から走行状態の FFT 画像へと変化する時，start of running と判定する。
- ③ CanSat が固く平坦な地面を走行しているとき，hard と判定する。
- ④ CanSat が草が生えており凸凹な地面を走行しているとき，grass と判定する。
- ⑤ CanSat が走行停止時に走行状態の FFT 画像から停止時の真っ青へと変化する時，stop running と判定する。

実際の FFT 画像の走行状態の例を Fig. 1-2 に示す。

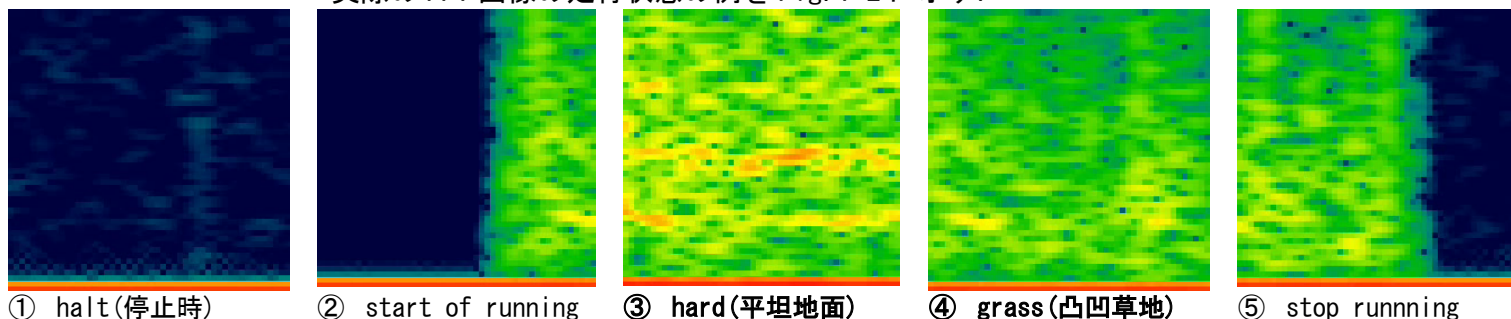


Fig. 1-2 各状態における振動の FFT 画像の例

FFT 画像から Deep Learning 画像認識により現在の状態を推定する。状態推定の結果 grass（草地での走行）と判定されれば場合、ローバの走行速度を横転しない遅い速度に設定する。同様に、状態推定の結果,hard（ARLISS のような硬い路面の走行）と判定された場合、滑らかに走行させるために、速い速度を設定する。つまり、走行開始後、路面が grass と判定されれば、能代宇宙イベントのような草地で glass と判定し速度を落とす、また ARLISS のような平坦な地面では hard と判定し速度を速める走行を目指す。

【Object Detection で 0m ゴール】

Object Detection(物体検知)では1枚の画像から物体を複数検知でき、画像中どこにあるかがわかる。1枚の画像でゴールの他にも人や気球なども同時に検知でき、それぞれ物体がどの位置にあるのかも座標情報で取得できる。Image Classification(画像分類)よりも1回の処理時間は長くなるが、従来はゴールを探すのに画面を分割する ROI を3分割、5分割や、さらに上下移動などを行っていたが、それを行う必要がなくなるため、特に CanSat のチルト方向の傾きに強くなると考えられる。

また、ゴール認識だけでなく、複数の他の物体を検知できるため、ゴールと同じ赤色の服を着た人や赤い気球が同じ画像内に写ってもゴールと識別することができ、認識精度の向上ができる。問題は、Object Detectionは従来の Image Classification よりも処理が複雑になり、計算量やメモリ使用量が増大するため、それに Raspberry Pi Zero で問題なく実装可能かどうかの懸念があるが、Raspberry Pi Zero でも Mobile Net SSD v2 モデルでは数秒間で認識可能であり、さらに Coral 等のアクセラレータを用いると、1秒以下で認識可能であるため、実装可能である。

Fig. 1-3 に Object Detection のゴール発見の例を示す。



Fig. 1-3 Object Detection でのゴール発見の例

【ミッションシーケンス】

本ミッションを達成するために必要なミッションシーケンスを以下に示す。

1. CanSat を気球のキャリアに収納する。
2. 気球のキャリアから CanSat 本体を放出する。
3. CanSat のパラシュートを開き、落下速度を減速させる。
4. 機体が破損無しで地上に着陸する。
5. CanSat 保護ケースを本体から切り離し、パラシュートと分離する。
6. GPS 位置情報をもとに、モータ回転数制御でゴールに向かって走行し、走行している地面の状態を加速度データにより推定し、その地面に適した制御を行う(ミッション)。
7. スタックになりそうな大きい草をタイヤで乗り越え、轍にスタックしたと判定した時、轍脱出動作を行い、轍からの脱出を行う。
8. GPS 情報によりゴール 10m 以内になったら動作速度を落とし、ゴールを目指して走行する。
9. 4m 以内になったら、カメラ画像を Deeplearning ObjectDetection により、ゴールを検知する。(ミッション)
10. カメラ画像によりゴールを発見したら、ゴールに向かって走行する。
11. カメラ画像からゴールに十分近づいたらゴール判定を行い、停止する。

以上のミッションシーケンスを Fig. 1-4 に示す。

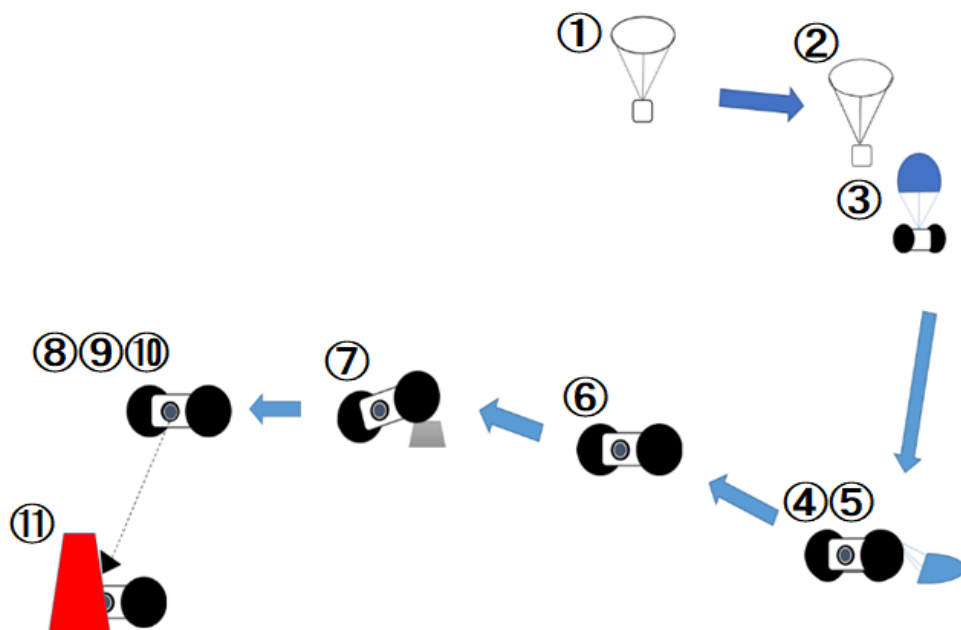


Fig. 1-4 ミッションシーケンス

第2章 サクセスクライテリア

このミッションステートメントに対して一つ及び複数のミッション項目・システム構成が存在し、それらの成功段階を詳細かつ定量的に述べたものがサクセスクライテリアとなります。

(参考資料: [成功基準（サクセスクライテリア）作成ガイドライン](#))

ミニマムサクセス	CanSat が放出落下後に破損なく着地し、パラシュートから分離し、GPS 位置情報からモータ回転数制御により 10m 以上走行する。
ミドルサクセス	CanSat がゴール付近に到達後、Object Detection で 0m ゴールする
フルサクセス	CanSat が地面を走行時、平坦な地面、凸凹な地面を判定し、その状態に合わせた速度やハンドルの走行制御を行い、Object Detection で 0m ゴールする。
アドバンスドサクセス	CanSat が地面を走行時、平坦な地面、凸凹な地面、 スタック状態 を判定し、その状態に合わせた走行制御を行い、Object Detection で 0m ゴールする。

第3章 要求項目の設定

3. 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	システム要求項目（ARLISS 打ち上げ <u>安全基準</u> ）
S1	投下する機体の質量が基準を満たしている
S2	容積がキャリア基準を満たしている
S3	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ていないことが試験で確認できている
S4	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ていないことが試験で確認できている
S5	ロケット分離時（パラシュート開傘時）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ていないことが試験で確認できている
S6	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
S7	ロストへの対策を実施しており，有効性が試験で確認できている （対策例：位置情報送信、ビーコン、蛍光カラーペイント等）
S8	打ち上げ時の無線機の電源 OFF の規定を遵守でき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）
S9	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
S10	ロケットへの装填からミッションの開始，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，大幅な設計変更はない

3. 2 ミッション要求

- ミッション達成のための CanSat への要求項目を書いてください。追加、削除する場合は、行を追加して書いてください。
- 各ミッション要求を満たすよう個別試験が必要な場合は各団体で適宜設定し、第 4 章に記載してください
- ミッション要求を分析することで記載が必要な要求項目が顕在化すると思いますので、各チームの独自性や詳細な要求設定の定義に期待します。

番号	ミッション要求項目
M1	ミッション後、規定された制御履歴レポートを運営及び審査員へ提出しログ・取得したデータについて説明が可能な状態である
M2	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
M3	環境の悪い地面における走行性能を試験で確認できている
M4	ミッションのゴール地点に設置されたコーンを Object Detection により検知し、0mゴールすることができることを試験で確認できている。
M5	ミッションの振動からの走行する地面状態の推定により、それに適した走行制御ができることを試験で確認できている。

第4章 システム仕様

4.1 機体外観

- 機体の外観を CAD 図や写真を用いて直径・高さの寸法を矢印などで簡単に説明してください。
(写真内に物差しを置く・図に数値を書き込むなど)
- 最低限、3 面以上から見た図を示してください。(例：正面図・平面図・右側面図・鳥瞰図)
- 図や写真を示す際は、余白などは最低限にコンパクトに示してください。
- 展開・収納などの機構変形により寸法が変化する場合、前後の寸法も示してください。

直径 [mm]	145
高さ [mm]	200
質量 [g]	850

機体は 2 輪のタイヤの間に電子回路を設置した形となっている。測定では、全長は左右のタイヤの外側に物差しを置き、高さはタイヤの上下間に物差しを置いて測定を行った。全長に物差しを置いて撮影した写真を Fig. 4-1-1 に示す。また高さを、物差しを置いて撮影した写真を Fig. 4-1-2 に示す。

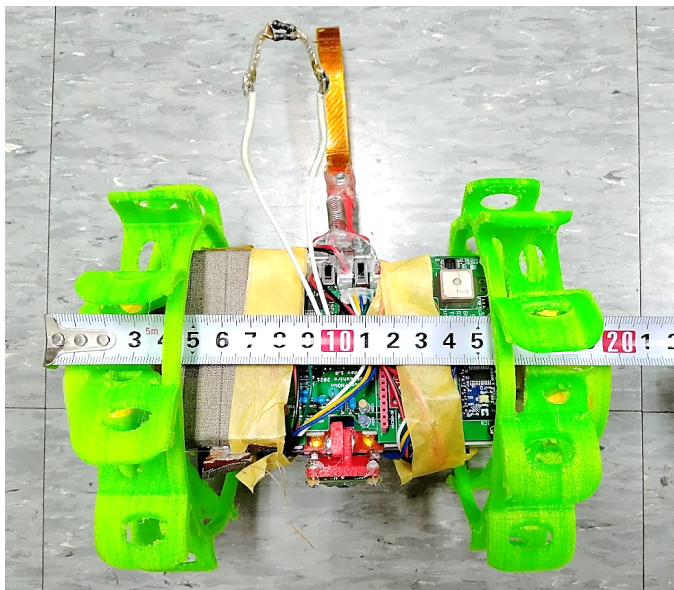


Fig. 4-1-1 CanSat 全長



Fig. 4-1-2 CanSat 高さ

4.2 機体内観・機構 【機体】

機体はポリカーボネート1枚の板に左右モータを取り付け、モータの裏側は電子回路を搭載している。また能代宇宙イベントのスタックしやすい草地でもスタックしないように草を乗り越えられるタイヤ内側に空洞になる形の設計のタイヤを取り付けた。CanSatを斜めから見た図をFig. 4-2-1に示す。CanSatを正面から見た図をFig. 4-2-2に示す。CanSatを上から見た図と下から見た図をFig. 4-2-3に示す。

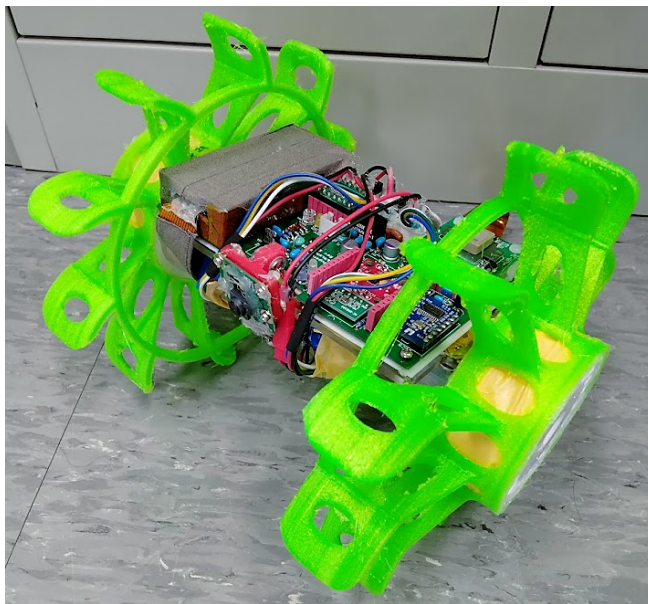


Fig. 4-2-1 CanSatを斜めから見た図



Fig. 4-2-2 CanSat正面から見た図

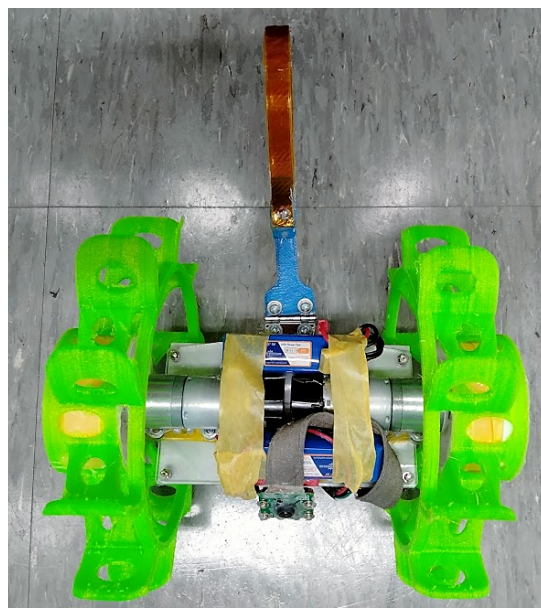
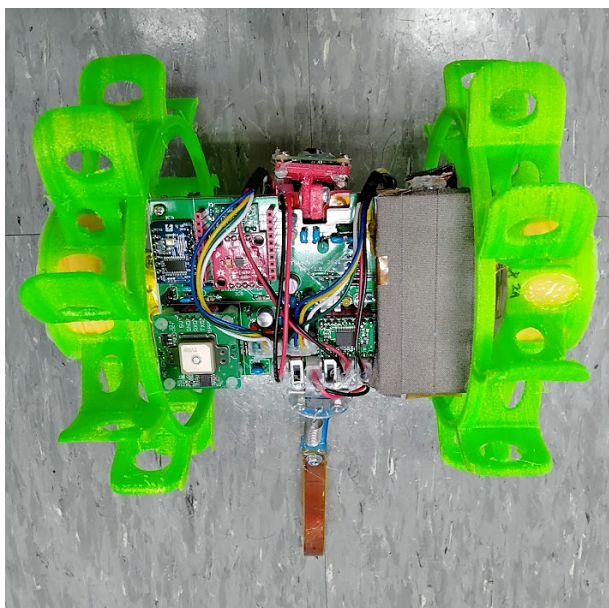


Fig. 4-2-3 CanSatを上から見た図(左), CanSatを下から見た図(右)

従来のタイヤには天然ゴムスポンジを利用することで着地衝撃に耐え、着地時に衝撃吸収を行う仕様となっていた。3Dプリンタの中でもゴムのような柔らかさを持つTPU素材(熱可塑性ポリウレタン)を利用することで、天然ゴムスポンジより複雑な造形ができる3Dプリンタを利用することにより機体への衝撃緩和と草へのスタック防止ができるタイヤを作成した。また、その柔らかさからキャリア収納のサイズも容易に小型化することができる。従来使用していたNRスポンジ素材と比較し、TPUタイヤが柔らかいことがわかる図をFig. 4-2-4に示す。

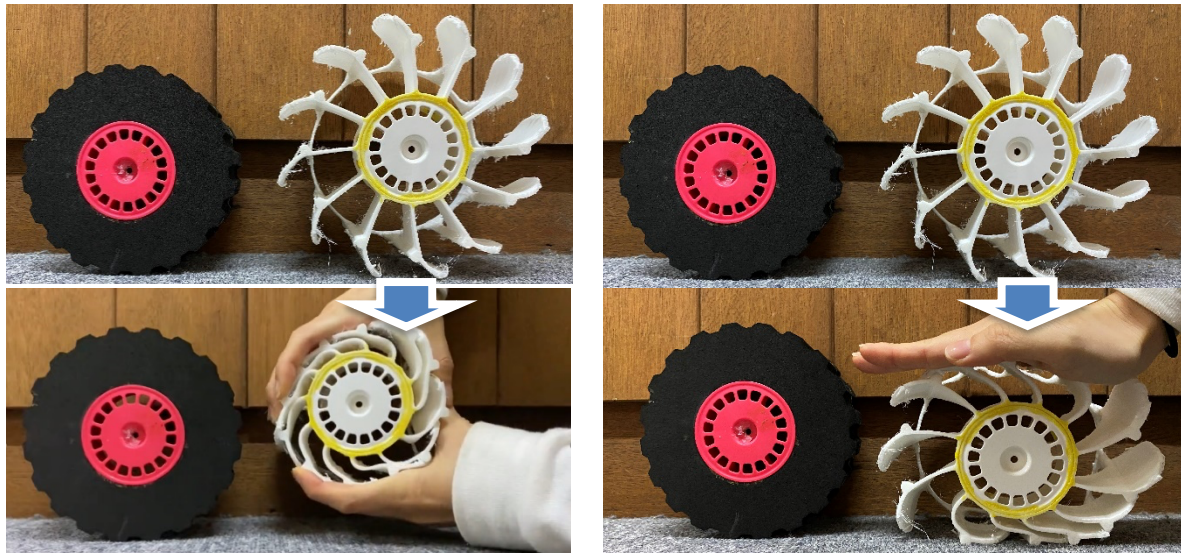


Fig. 4-2-4 TPUタイヤ(右)の収納性, TPUタイヤ(右)の柔らかさ

機体にはプラスチックの中で最も硬い素材のポリカーボネートを利用し、着地時に衝撃で少し曲がることにより衝撃吸収を行う仕様となっている。ポリカーボネートの柔らかさがわかる図をFig. 4-2-5に示す。

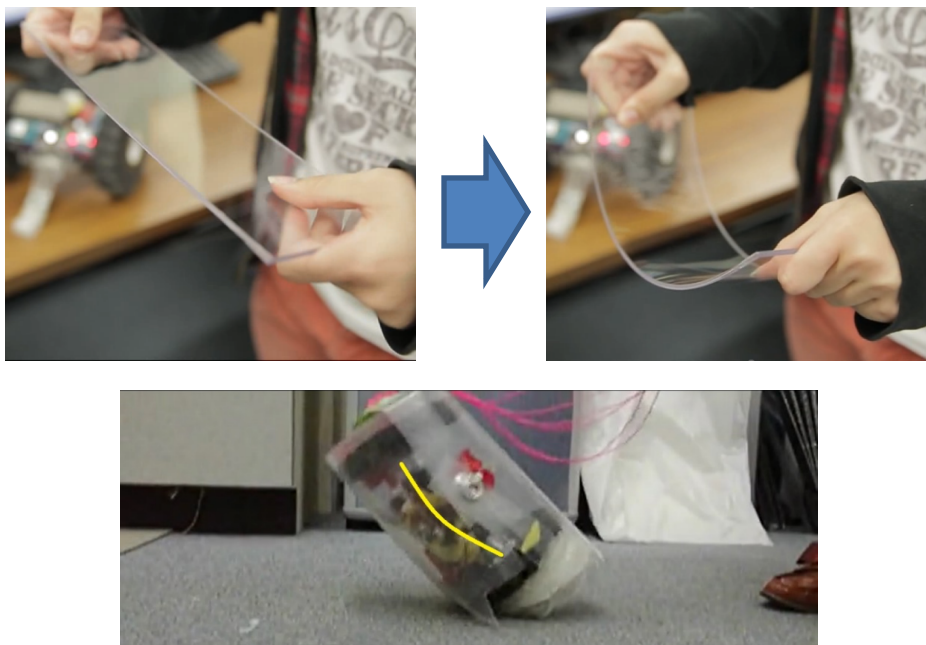


Fig. 4-2-5 ポリカーボネート衝撃吸収

【減速機構パラシュート】

減速機構のパラシュートには CanSat との絡まり防止のための素材や構造を取り入れている。

パラシュートの画像を Fig. 4-2-6 に示す。

・パラシュート生地

衝撃に耐えられるよう破れにくく、固くさらさらとした生地で CanSat が上を走行してもタイヤに巻き付きにくくした。

・ひも 1

曲がりにくい素材である金属ワイヤーを利用し、機体に巻き付くことを防いだ。生地との接続部は6つでクランプカンによる頑丈な圧着がしてある。

・ひも 2

強い力を加えないと曲がらない金属メジャーを利用することで、CanSat 上にパラシュートが落下せず、CanSat と遠ざかって着地ができる構造により、絡まり防止をした。

・CanSat ケース

機体上へパラシュートが落下しても、開いたとき弾かれることで直接の絡まりを防止した。また、CanSat とパラシュート紐との接続部と、分離機構部分をケース越しに2か所設けたことにより、開傘衝撃が CanSat 分離部分へ加わらないようにし、衝撃緩和を行った。

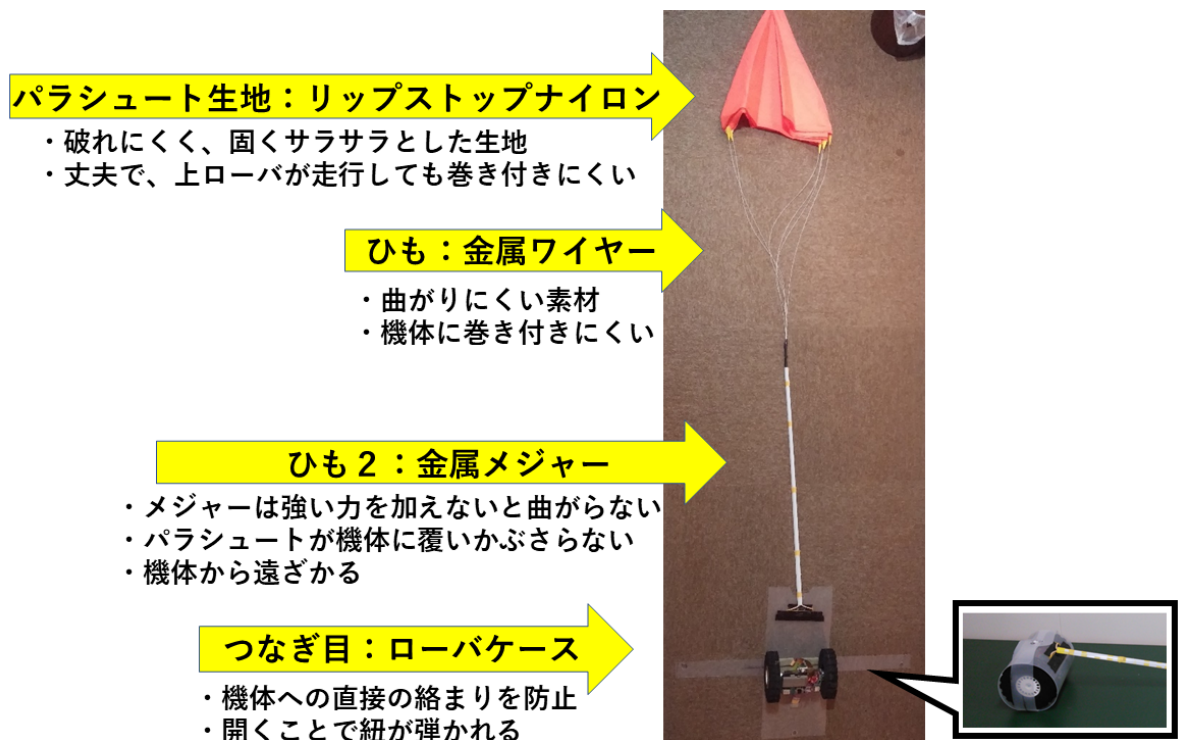


Fig. 4-2-6 パラシュート

4.3 システム図（機体搭載計器仕様一覧）

【マイコン、モータ】

RaspberryPi Zero 2 W をマイコンに利用し、Michibiki Compatible GPS で位置情報を取得し現在位置と目標位置の角度を計算し、TB6612 の Dual Motor Driver に目標位置への旋回 PWM 制御値を出力し、モータに出力する。

【センサ類】

LSM9DS1 で±16G の加速度・ジャイロを取得し、H3LIS331DL で 200G の加速度を取得、BME280 で気圧・気温を取得。加速度・ジャイロは CanSat の衝撃値や CanSat が上下反転して走行しているときに上下を戻すために用いて、気圧・温度は CanSat の高度とコンピュータに加わる熱の温度の参考にしている。

また、気圧センサが故障時には Photo IC Diode で放出されたかどうか明るさを MCP3208 A/D Converter で A/D 変換し測定している。

【長距離通信無線】

また、長距離へ落下した際に IM920 wireless Communication を用いて位置情報を送信し、現在位置特定に利用している。

【カメラ】

ゴールより数m接近後に機械学習による画像認識で誘導制御を行うために、Raspberry Pi Camera v2 から写真を撮影しゴールのパイロンが写っているか判定し誘導を行う。

Fig. 4-1 にシステム図を示す。また、実際に使用している電子部品の一覧を Table 4-3-1 に示す。

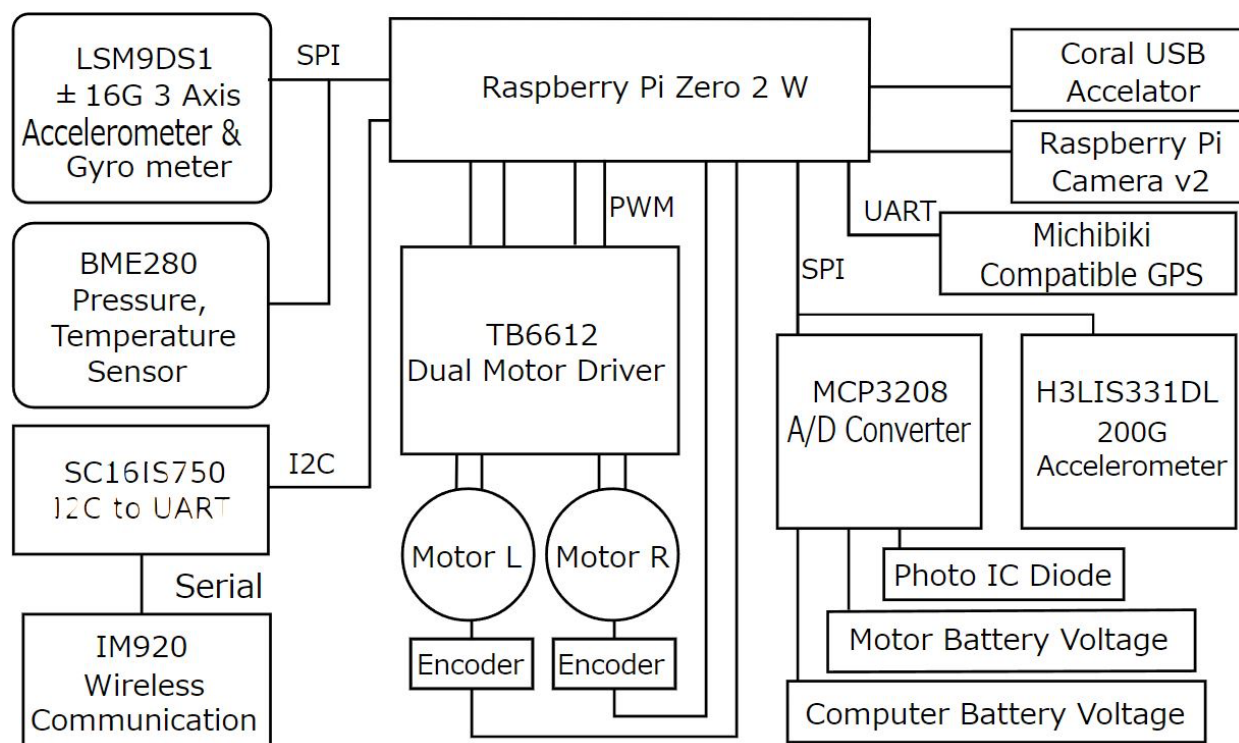


Fig. 4-3-1 システム図

Table 4-3 使用部品

分類	名称・型番	入手先・参考情報等	URL
----	-------	-----------	-----

GPS	AE-GYSFDMAXB	秋月電子	http://akizukidenshi.com/download/ds/akizuki/AE-GPS_manual_r1.06_s.pdf
マイコン	Raspberry Pi zero 2 W	SWITCH SCIENCE	https://www.switch-science.com/catalog/7600/
200G 加速度センサ	H3LIS331DL	SWITCH SCIENCE	https://cdn.sparkfun.com/assets/c/6/5/8/d/en_DM00053090.pdf
16G 加速度センサ	LSM9DS1	SWITCH SCIENCE	https://cdn.sparkfun.com/assets/learn_tutorials/3/7/3/LSM9DS1_Datasheet.pdf
I2C シリアル変換 A/D コンバータ	SC16IS750 MCP3208	SWITCH SCIENCE 秋月電子	https://www.switch-science.com/catalog/2310/ http://akizukidenshi.com/download/MCP3208.pdf
モータドライバ	TB6612	秋月電子	http://akizukidenshi.com/download/ds/Toshiba/TB6612FNG_datasheet_ja_20141001.pdf
5V 3 端子レギュレータ	V78-1000	秋月電子	https://www.marutsu.co.jp/contents/shop/marutsu/ds/v78xx-1000.pdf
3.3V 3 端子レギュレータ	48033	秋月電子	http://akizukidenshi.com/download/ta48033s.pdf
通信装置	IM920	秋月電子	https://www.interplan.co.jp/solution/wireless/im920/im920c.php
気圧センサ	AE-BME280	秋月電子	http://akizukidenshi.com/download/ds/bosch/BST-BME280_DS001-10.pdf
カメラ	Raspicam V2	SWITCH SCIENCE	https://www.switch-science.com/catalog/2713/
モータ	POLOLU ギアード モータ 99:1	POLOLU	https://www.pololu.com/product/4867
バッテリー (モータ用)	Hyperion G5 3S 850mAh LiPo 25C (本審査より記述 変更)	Hyperion	https://hyperion-world.com/en/p2601952-hp-g550-0850s3-14323
バッテリー (computer 用)	Bos Lipo リポバ ッテリ ー 7.4V 1100mAh 2S 25C (本審査より記述 変更)	Amazon	https://www.amazon.co.jp/gp/product/B09TF32517/ref=ox_sc_act_title_1?smid=A2RF2YPOTPHH3L&th=1

【アルゴリズム】

① 落下判定～パラシュート離脱

プログラム開始後、気圧センサの値を測定し、地面から上空に上がった後地面に落ちてきたかを高度で判定し、その後パラシュートを切り離し、CanSat 走行方向にパラシュートがあれば画像認識で旋回する。パラシュートを切り離すとデータが取得できているか確認し、できなければもう一度制御を切り離しから行い、GPS など制御に必要なデータが取得できない場合取得できるまでシーケンスを繰り返す。

② 走行制御～ゴール

GPS 位置情報から走行制御を開始し、パラシュートが走行方向にあれば旋回制御を行う。走行状態を加速度データにより推定し荒れた路面などの判定結果が出た場合にハンドルパラメータ値を調整する。スタックした場合はスタック処理のバック動作や回転動作から脱出動作を行う。ゴール 10m 以内まで近づくと減速しゴールに近づきやすい走行を行う。ゴール 4m 以内まで近づくと Deep Learning でゴールを発見するまで旋回走行する。ゴールを発見するとゴールの方向へ走行し、ゴール 0m 地点と Deep Learning で判定すると停止する。

フローチャートを Fig. 4-3-2 に示す。

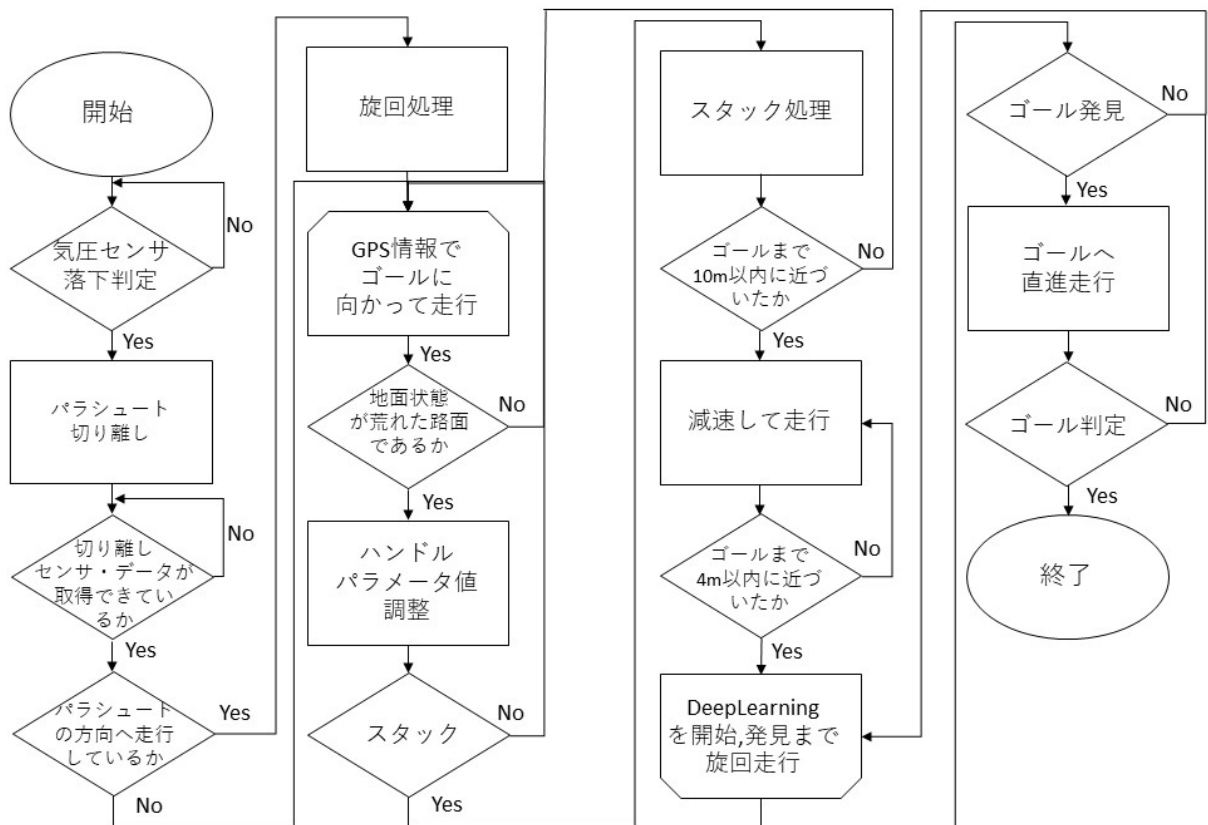


Fig. 4-3-2 フローチャート

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）	実施予定日
	《システム要求に関する項目》		
V1	質量試験	S1	6/25
V2	機体の収納・放出試験	S2	6/25
V3	準静的荷重試験	S3	7/9
V4	振動試験	S4	7/29
V5	分離（パラシュート開傘）衝撃試験	S5	7/17
V6	落下試験	S6	6/28
V7	GPS データダウンリンク試験	S7	6/11
V8	通信機電源 OFF/ON 試験	S8	6/15
V9	通信周波数 ch 変更試験	S9	6/14
V10	End-to-end 試験	S10	7/27
	《ミッション要求に関する項目》		
V11	制御履歴レポート作成試験	M1	7/20
V12	着地衝撃試験	M2	6/27
V13	走行性能確認試験	M3	7/8
V14	ゴール検知試験	M4	7/8
V15	走行状態推定試験	M5	8/4

第6章 実施試験の内容

〈注意点〉

- 準静的荷重試験，振動試験，分離（パラシュート開傘）衝撃試験の試験条件は別紙の「CanSat Regulations at ARLISS」を参考にしてください
- End-to-end 試験では，CanSat の投下からパラシュート展開，ミッションの実施（ランバックであれば），データの取り出しまでを本番と同じ手順で行い，各シーケンスを行えることを確認できる動画を提出してください。また動画内では，**少なくともミニマムサクセスクライテリア**を満たしていることが確認できるようにしてください。
- 他の試験で動画を用いる場合も同様に，URL を載せてください（**ファイル提出は避けてください**）

v1. 質量試験

- 目的
CanSat とパラシュートを合わせた質量が規定質量である 1050g 以下を満たすことを確認する。
- 試験内容
CanSat とパラシュートを質量計で計測し，レギュレーションに記載されている質量(1050g)以下であることを確認する。
本番当日キャリアに収納する全ての機材（CanSat 本体，回路部品，保護ケース，パラシュートなど）の質量を計測し，既定の質量以下であることを確認した。
CanSat は本番では保護ケースで覆った状態でキャリアに収納するため，保護ケースを覆った状態で計測器に載せ計測する。Fig. 6-1-1 に保護ケースを収納させた CanSat の図を示す。

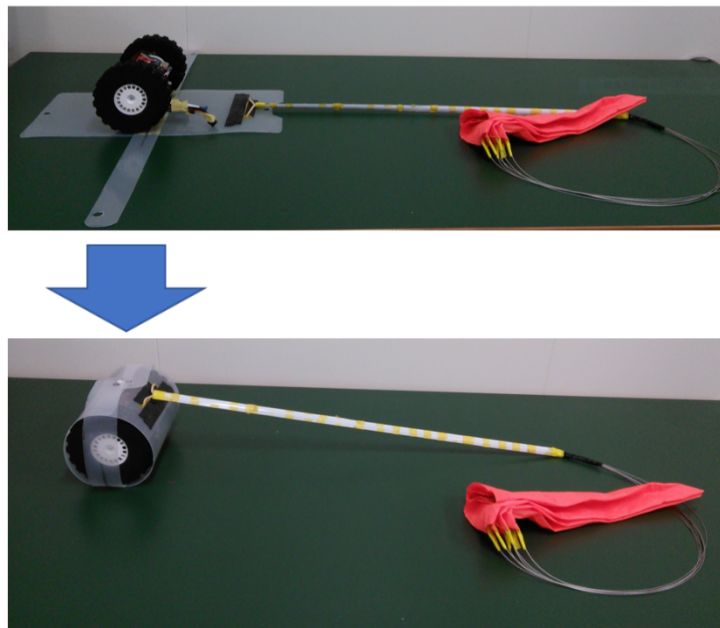


Fig. 6-1-1 保護ケースに収納させた CanSat

- 結果

CanSat とパラシュートの総重量は **850g** であり、レギュレーションの 1050g 以下であることを確認した。Fig. 6-1-2 に質量測定結果の様子を示す。



Fig. 6-1-2 パラシュートと CanSat の質量

- 考察

パラシュートの質量を含めた CanSat の総重量が、レギュレーションを満たしていることがわかった。

v2. 機体の収納・放出試験

● 目的

キャリア（内径 146mm，高さ 240mm）に収納でき，さらに，自重で放出できることを確認した．また，キャリア収納時間が 5 分以内であることを確認した．

● 試験内容

キャリアに収納できることを証明するため，ARLISS で購入したキャリアを収納することで規定寸法であることを確認した．キャリアの寸法奥行を Fig. 6-2-1，外径を Fig. 6-2-2 に示す．

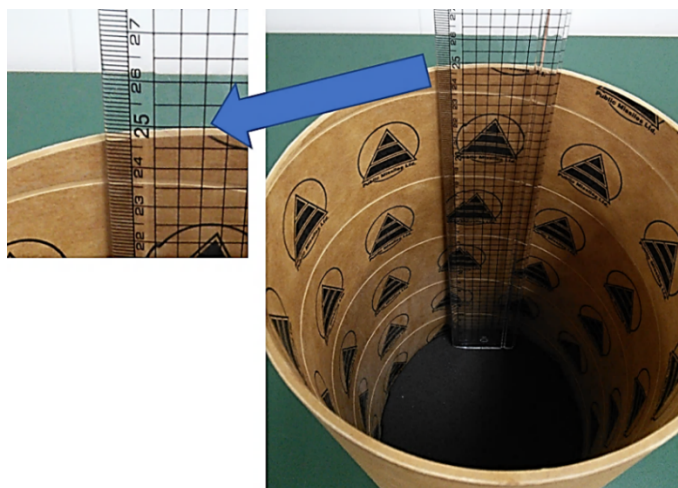


Fig. 6-2-1 キャリア収納ケースの奥行寸法

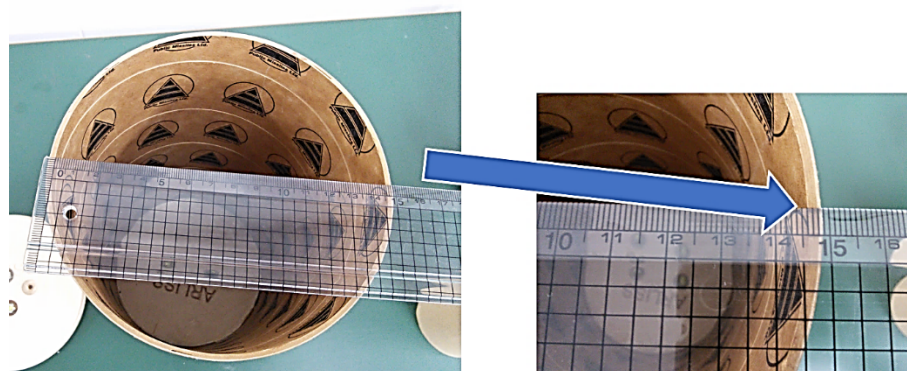


Fig. 6-2-2 キャリア収納ケースの外径寸法

Fig. 6-2-1, Fig. 6-2-2 から，キャリアが規定の奥行 240mm 外径 146mm であった．Fig. 6-2-1 では目盛りは奥行 245mm を指しているが，定規の余白が 5mm あるため，キャリアの奥行きは 240mm である．

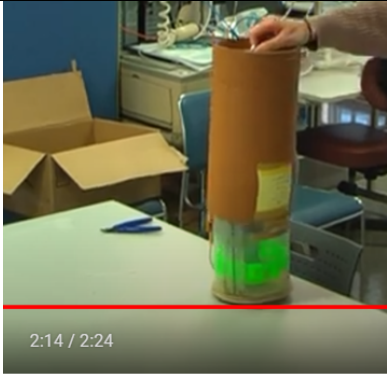
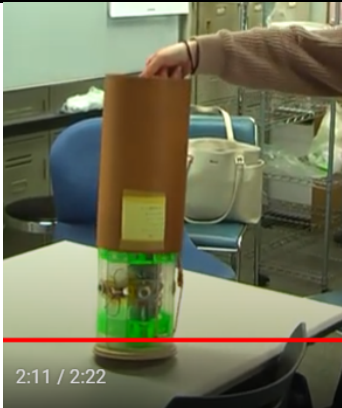
また，CanSat は保護ケースで覆った状態で，キャリアに収納するため，保護ケースに覆った状態で計測する．

規定サイズ内のキャリア収納ケースに収納後，CanSat が自重で落下できることを確認し，キャリア収納までの時間が 5 分以内であることを確認する．

● 結果

表 6-2-1 に示すように，CanSat を収納後，自重で放出することが 3 回中 3 回確認できた．

表 6-2-1 キャリア放出実験の結果

回数	実験動画	放出判定
1	 2:20 / 2:30 https://youtu.be/seNspw_fwRE	自重で放出できた また動画時間は 2:30 であり、キャリア収納が 5 分以内でできた
2	 2:14 / 2:24 https://youtu.be/8UtMT1DaQ00	自重で放出できた また動画時間は 2:24 であり、キャリア収納が 5 分以内でできた
3	 2:11 / 2:22 https://youtu.be/ejuggNlhzcg	自重で放出できた また動画時間は 2:22 であり、キャリア収納が 5 分以内でできた

● 考察

寸法測定結果より、機体は既定のキャリアに収納可能であること、機体収納 5 分以内であることが確認できた。

v3. 準静的荷重試験

- 目的

ロケットの打ち上げにかかる想定される大きさの静荷重を CanSat に与え、CanSat が耐えられることを実証する。

- 試験内容

レギュレーションでは準静的荷重の値は 10[G] が推奨値となっている。

また、荷重をかける時間を設定するために、自団体が実際に ARLISS2019 にて測定したロケットの打ち上げ時間を参考にする。ARLISS2019 3 回分の打ち上げにて実際にロケットの荷重がかかっている準静的荷重の値と時間はそれぞれ、

1 回目最大約 9.78[G] で 7[s]、2 回目最大約 8.93[G] で 8[s]、3 回目最大約 8.32[G] で 8[s] だった。準静的荷重の値はレギュレーション値 10[G] を下回っており、時間は 10[s] 以内であることから、**実験では 10[G] の準静的荷重を 10[s] 与えた**。実際に荷重がかかっている ARLISS2019 グラフを、ARLISS 1 回目準静的荷重を Fig. 6-3-1 に、ARLISS 2 回目準静的荷重を Fig. 6-3-2 に、ARLISS 3 回目準静的荷重を Fig. 6-3-3 に示す。

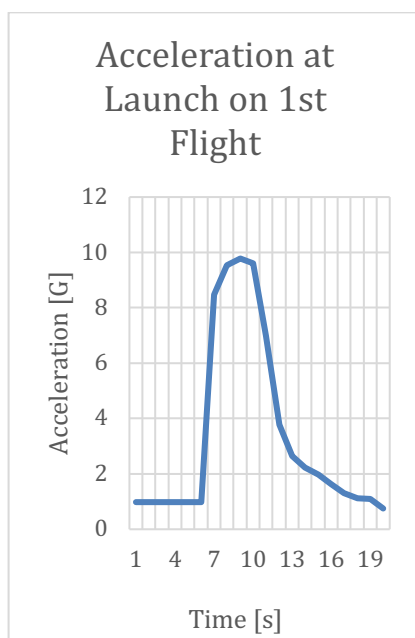


Fig. 6-3-1 1 回目準静的荷重

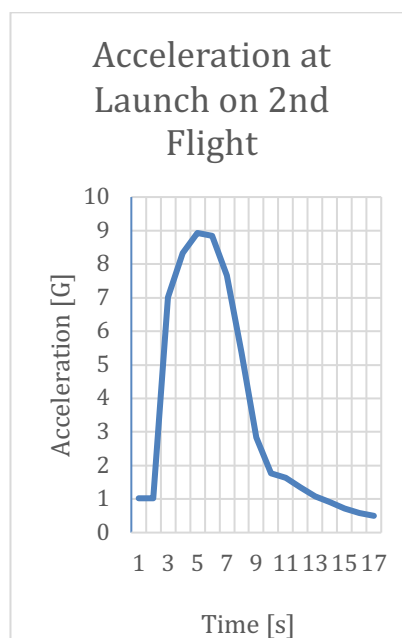


Fig. 6-3-2 2 回目準静的荷重

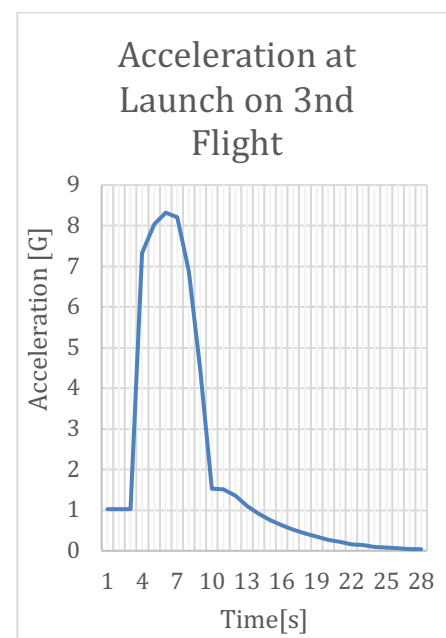


Fig. 6-3-3 3 回目準静的荷重

予めセンサや動力系が正常に動作することを確認し、その後 CanSat に搭載し、長いロープを取り付けたトートバッグに入れ、その紐を持ちトートバッグを振り回し回転運動の遠心力で大会の推奨値である 10G の加速度を与える。実際に荷重を加えている実験写真を Fig. 6-3-4 に示す。



Fig. 6-3-4 遠心力で 10G をかけた様子

キャリアに搭載した加速度センサの値を無線通信で PC 画面により確認し、10G を超えた時点から力を緩めず 10 秒間回転運動を継続し連続的な遠心力による加速度をかける。

その後、CanSat をキャリアから放出し、各種センサと動力系が正常に動作することをもって CanSat が静荷重に耐えうることを確認した。

● 結果

10G を 10 秒以上 3 回繰り返しかけた後に CanSat に搭載するセンサ(光・加速度・ジャイロ・温度・気圧・GPS)は正常に動作し、その後問題なくパラシュート離脱動作を行い、正常に走行することを確認した。

実際に 10G を 10 秒以上観測している加速度データログのグラフを Fig. 6-3-5 に示す。

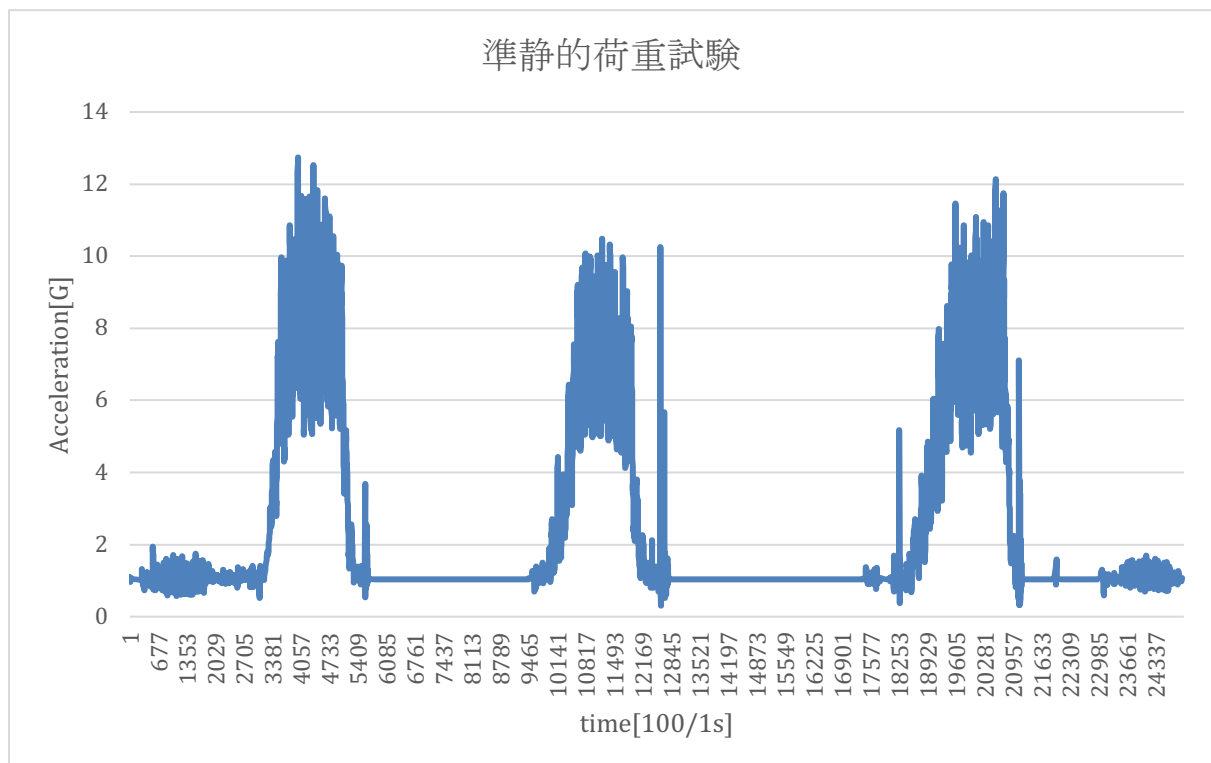


Fig. 6-3-5 加速度センサの数値（縦軸はかかった加速度[G]、横軸は時間[100/1s]）

静荷重試験の様子は動画に撮り、異常の有無の確認を記録した結果からも、機体の異常は見られなかった。（下記の URL を参照）

● 静荷重試験の動画 URL : <https://youtu.be/0zAlQskb808>

（センサ値事前確認 0 : 50～、静荷重運動① 2 : 27～、静荷重運動② 3 : 35～、静荷重運動③ 5 : 05～、センサ確認 7 : 37～、パラシュート離脱機構とモータ駆動と機体確認 9 : 36～）

● 考察

CanSat をロケットで打ち上げる際に想定される静荷重に CanSat が耐えられることが確認できた。

v4. 振動試験

- 目的

CanSat がロケット打ち上げの振動に耐えうるかを確認する。

- 試験内容

ロケット打ち上げ時にかかる CanSat への振動条件として、レギュレーションで推奨されているランダム振動をフラットパターンで 30Hz~2000Hz の 15G の振動を加える。振動時間は V3 準静的荷重試験の ARLISS2019 の 3 回分データでのロケット打ち上げ時間では 10 秒以下であることを確認したが、誤差や試験機のパラメータを確認するため考慮し多く設定した 1 分間の振動を設定し、CanSat を振動させた後、電子回路の動作、パラシュート離脱機構、モータ、機体に問題がないか確認する。

振動機は IMV 社 日本高度信頼性評価試験センターの振動機を利用した。実際の振動機写真を Fig. 6-4-1、振動条件設定の PC 画面を撮影した写真の基本・制御条件を Fig. 6-4-2 に、加振システム設定を Fig. 6-4-3、制御目標_全体を Fig. 6-4-4、入力チャネルを Fig. 6-4-5 示す。

また振動後、分離衝撃 40G の Shock 波（衝撃試験）を加えた。Shock 波は正弦半波規格 JIS 振動加速度 392 m/s^2 (40G 相当) 作用時間 0.01sec (0.05sec でブレークポイント到達、その後減衰) 制御周波数帯域 5000Hz を与えた。分離衝撃の加振システム情報を Fig. 6-4-6、目標波形を Fig. 6-4-7 に示す。



Fig. 6-4-1 実際の振動機 (茶色筒の中に CanSat 格納)

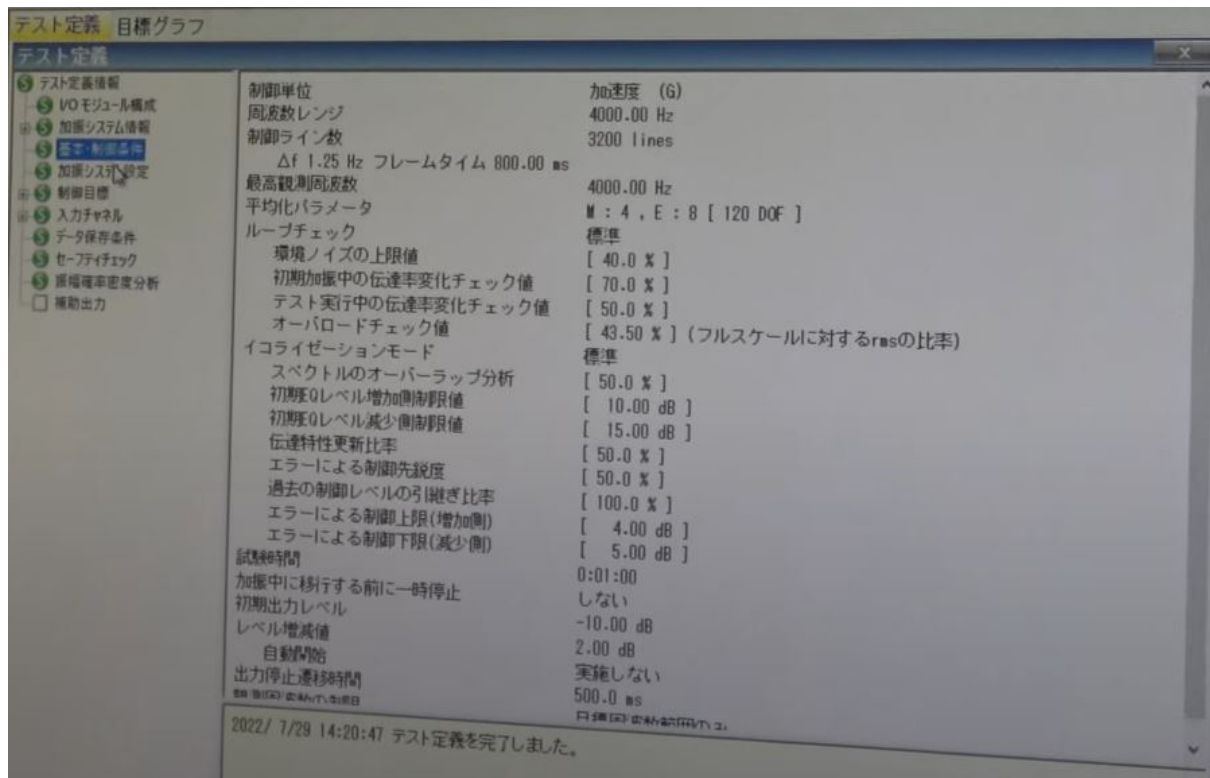


Fig. 6-4-2 基本・制御条件

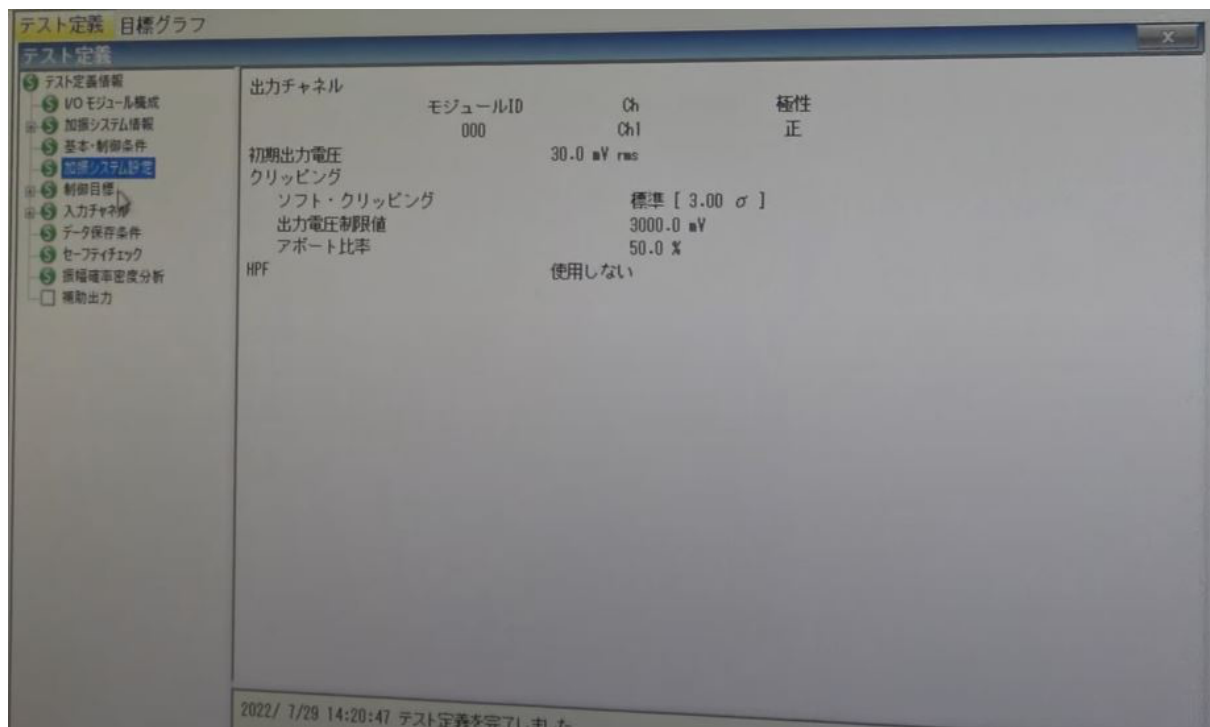


Fig. 6-4-3 加振システム設定

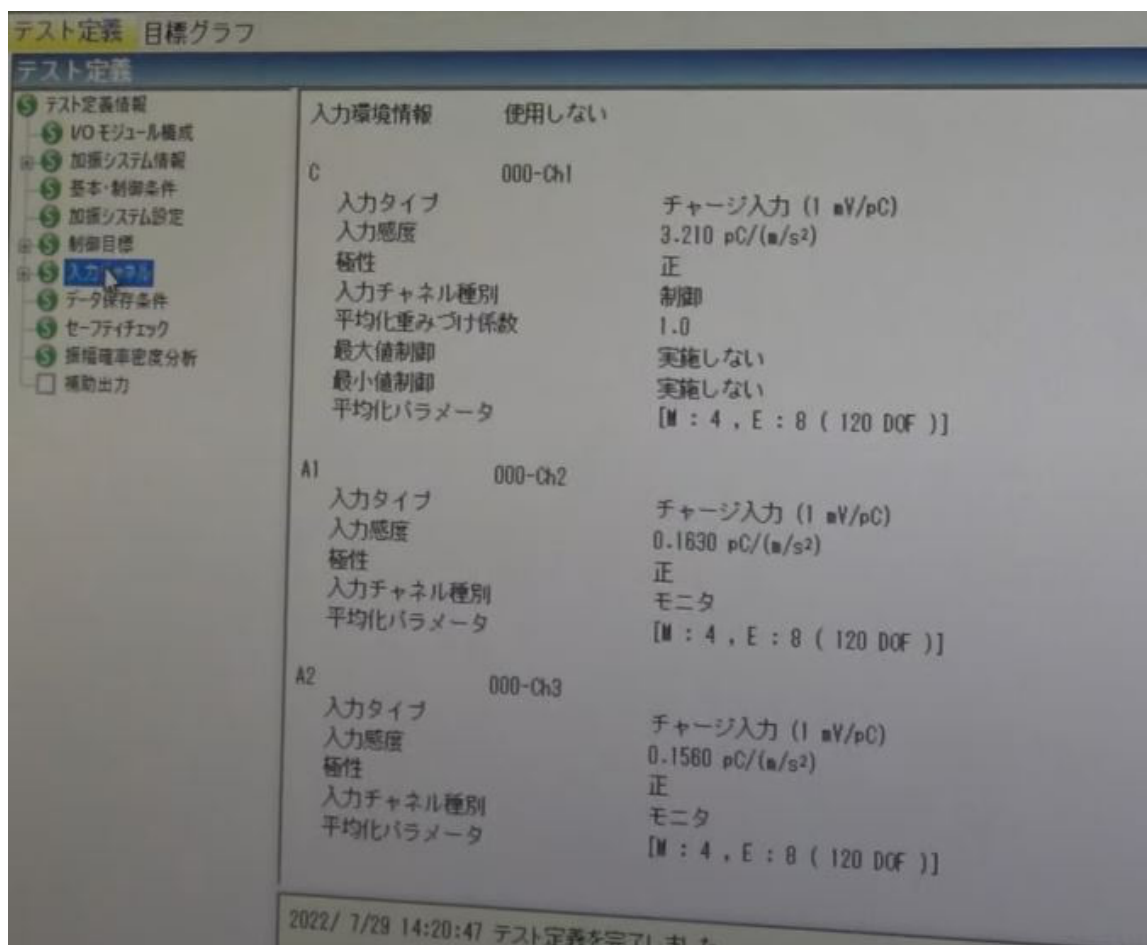


Fig. 6-4-4 制御目標_全体

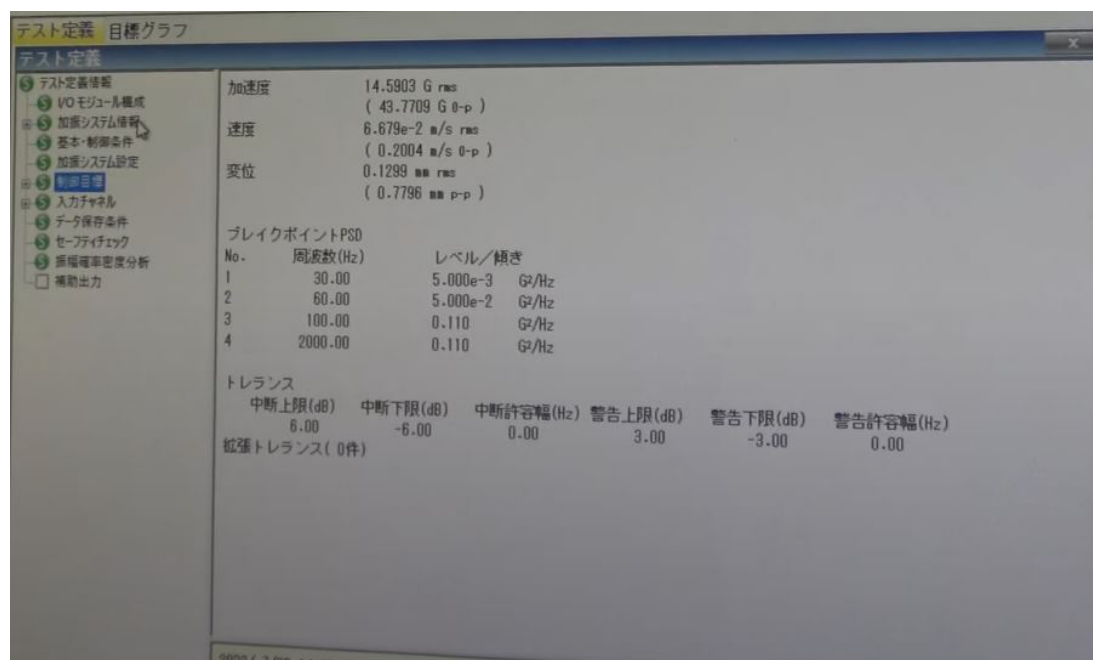


Fig. 6-4-5 入力チャネル

定義情報	加振システム情報名 A30/EM3HAM				
IO モジュール構成	出力チャンネル				
加振システム情報	モジュールID 000 Ch1 極性 正				
目標波形	初期出力電圧 既定値 30.0 mV rms				
制御条件	可動部質量 0.0 kg				
加振システム設定	定格情報				
入力チャンネル	制御周波数レンジ 制限しない				
セーフティチェック	加振力 加速度 速度 変位				
レベルスケジュール	SINE 1.0 kN 0-p 791.0 m/s ² 0-p 2.050 m/s 0-p 77.20 mm p-p				
タイミング信号	RANDOM 0.70 kN rms 554.0 m/s ² rms 2.050 m/s 0-p 77.20 mm p-p				
SRS分析条件	SHOCK 2.0 kN 0-p 1580.0 m/s ² 0-p 2.550 m/s 0-p 77.20 mm p-p				
	HV-SHOCK 2.0 kN 0-p 1317.0 m/s ² 0-p 3.60 m/s 0-p 77.20 mm p-p				
	接点制御				
	接点入力利用情報				
	端子	利用情報	極性		
	1	リモートコントロール許可	正		
	2	加振システム動作可能	正		
	3	加振開始	正		
	4	試験中止	正		
	5	一時停止	正		
	6	使用しない			
	7	使用しない			
	8	使用しない			

Fig. 6-4-6 (分離衝撃 Shock 波) 加振システム情報

定義情報	目標波形種別	Classical Shock 波形	
IO モジュール構成	波形種別	ハーフサイン波	
加振システム情報	サンプリング周波数	4096.00 Hz	
目標波形	ピーク振幅値	40.0 G	
制御条件	パルス幅	10.0 ms	
加振システム設定	パルス位置	0.0 %	
入力チャンネル	レストタイム パルス前	0.0 ms	
セーフティチェック	レストタイム パルス後	0.0 ms	
レベルスケジュール	トレランス種別	IEC(JIS C) 60068-2-27	
タイミング信号	補償波	左右非対称	
SRS分析条件	補償波形状	Type4	
	ピークレベル パルス前	90.0 %	
	ピークレベル パルス後	90.0 %	
	全長	124.7559 ms (512 ポイント)	
	加速度	最小 -6.5534 G	最大 40.0 G
	速度	-1.2415 m/s	1.2414 m/s
	変位	-9.3806 mm	7.1724 mm

Fig. 6-4-7 (分離衝撃 Shock 波) 目標波形

● 結果

CanSat に振動を与えた結果、電子回路動作、パラシュート離脱機構、モータ動作、

機体に問題はなかった。

また、実際に振動を与えた振動機からの IMV 社提供データである制御及び出力データを示す。出力のレベルを Fig. 6-4-8, 目標の G を Fig. 6-4-9, 制御応答の G を Fig. 6-4-10, 制御の m/s^2 を Fig. 6-4-11 に示す。

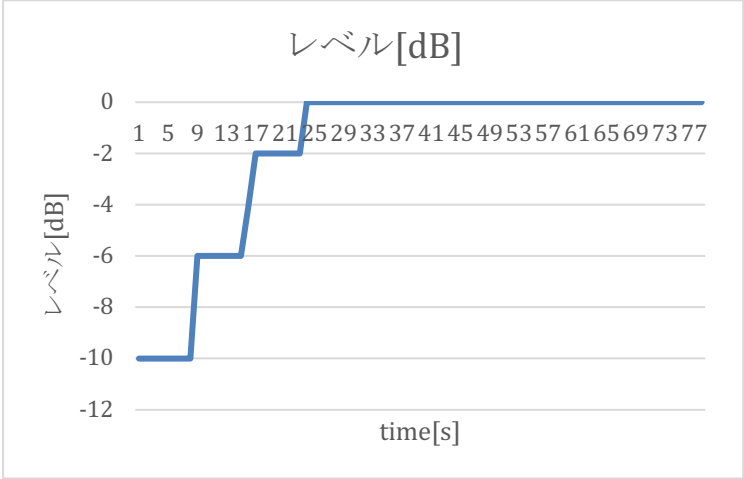


Fig. 6-4-8 出力レベル

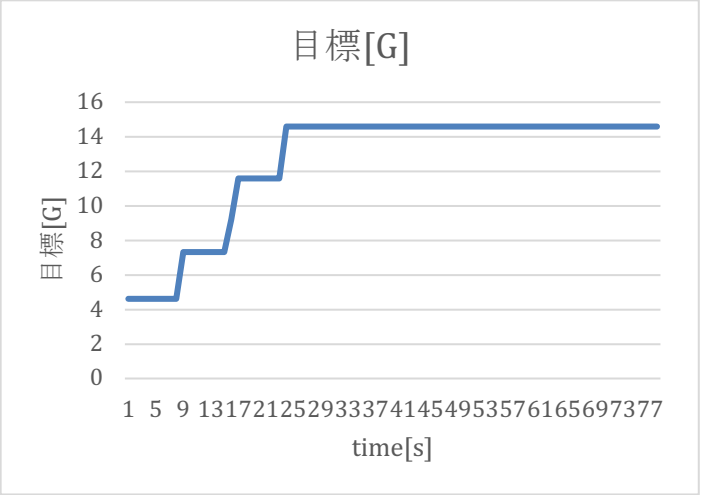


Fig. 6-4-9 目標 G

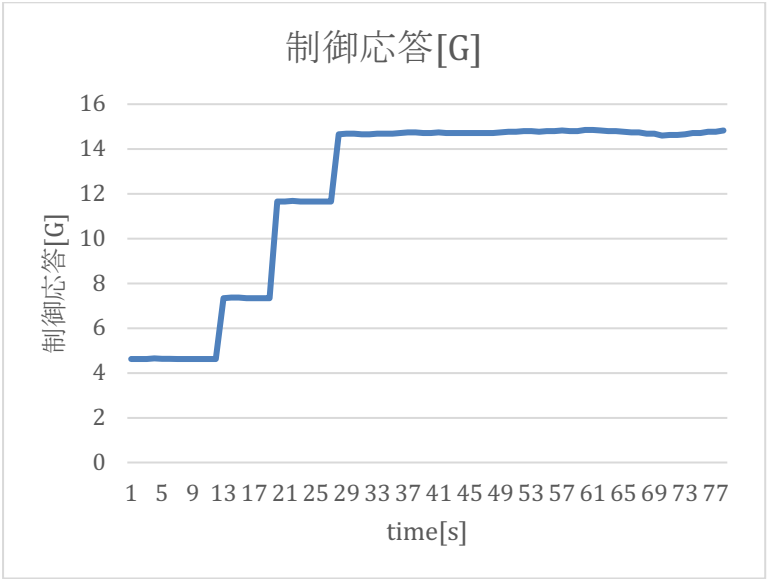


Fig. 6-4-10 制御応答 G

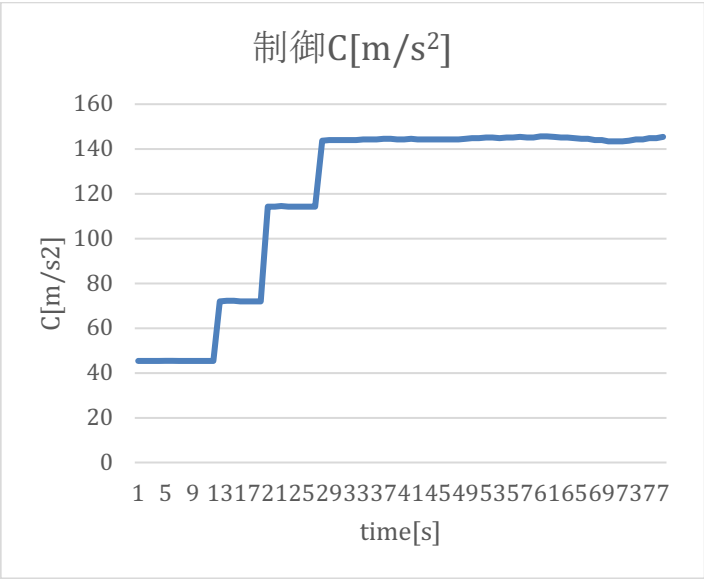


Fig. 6-4-11 制御 m/s^2

また振動試験時 IMV 社の加速度センサを回路マイコン上 A1 と CanSat 下面 A2 に設置した. その時の加速度センサのデータを, A1 加速度センサを Fig. 6-4-12, A2 加速度センサを Fig. 6-4-13 に示す.

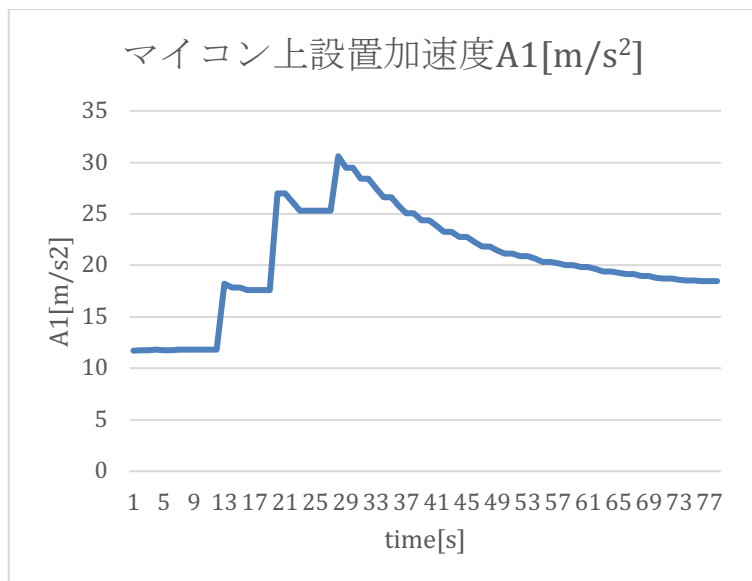


Fig. 6-4-12 マイコン上設置加速度 A1 [m/s²]

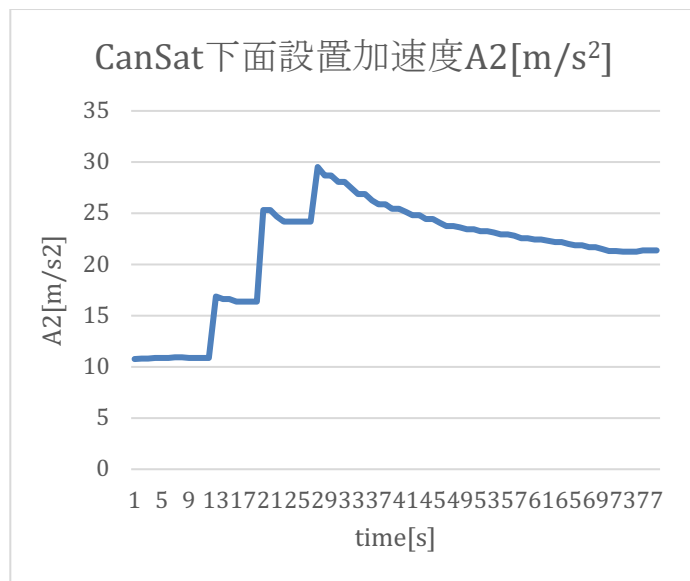


Fig. 6-4-13 CanSat 下面設置加速度 A2 [m/s²]

また, 分離衝撃 Shock 波の衝撃結果グラフを Fig. Fig. 6-4-14 に示す.

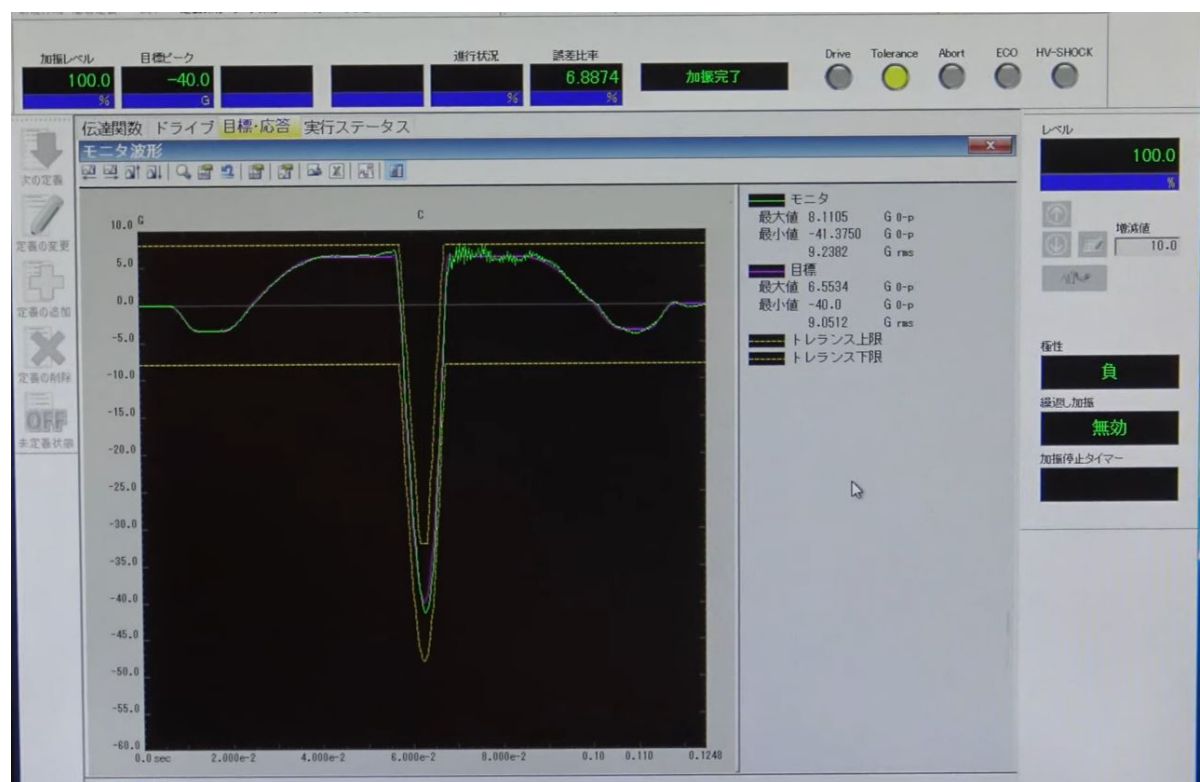


Fig. 6-4-14 分離衝撃 Shock 波の衝撃結果グラフ

また、CanSat に搭載した自団体の加速度センサの振動時データを Fig. 6-4-15 に示す。

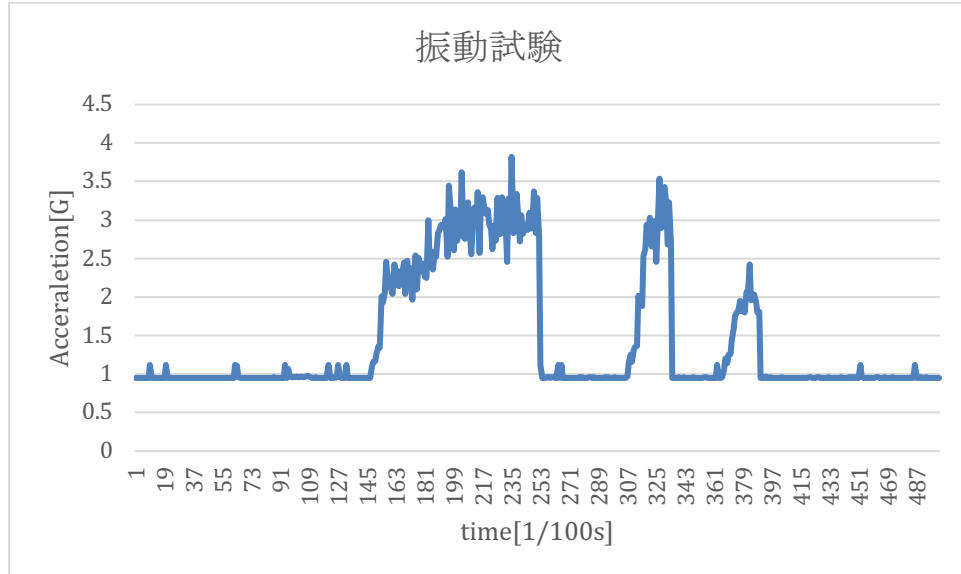


Fig. 6-4-15 (自団体の)加速度センサデータ

また、実際に振動試験を行った様子を撮影した動画を下記 URL に示す。

振動試験動画： https://youtu.be/aox9CFF_uos

※4K 画質なので視聴時は注意

(00:00 機体に IMV 提供の加速度センサ設置,
05:30 CanSat にパラシュート接続,
08:27 キャリア収納 (治具収納),
14:15 治具固定開始,
15:00 センサ動作 事前チェック,
33:33 振動条件設定,
41:45 振動開始
44:30 振動続き (7s),
48:10 分離衝撃 Shock 波条件設定,
53:18 Shock 波開始
56:56 治具取り外し,
58:22 センサ動作チェック,
1:02:30 CanSat 取り出し
1:05:13 パラシュート分離動作+モータ駆動確認
1:05:37 機体破損箇所チェック,
1:07:55 カメラ動作チェック
1:08:57 (試験終了)加速度センサ取り外し,
1:10:44 施設撮影)

また振動データをドライブで共有する

振動試験データ：

<https://drive.google.com/drive/folders/1WfYcEoprKB3MbaULERZaVwiM9W5cBHNP?usp=sharing>

● 考察

振動試験により、CanSat がロケット打ち上げ時の振動に耐えられることを確認した。

また提供されたデータから実験根拠の振動データを提出した。

v5. 分離衝撃試験

● 目的

CanSat がロケットから放出される時の衝撃に機体が耐えるかを衝撃試験により確認する。また、ロケットから放出され、パラシュートが開傘する際の衝撃に結合部が耐えられ、パラシュートや機体に損傷がないことを確認する。

● 試験内容

CanSat 機体がロケットから放出されパラシュートが開傘する際の瞬間的な分離衝撃の大きさを ARLISS2019 の打ち上げ 3 回分データから決定した。ARLISS2019 での分離衝撃はそれぞれ、1 回目分離衝撃約 23.48 [G]、2 回目分離衝撃約 25.41 [G]、3 回目分離衝撃約 21.67 [G]であった。3 回の分離衝撃では 30[G]を下回っていることから、分離衝撃は誤差を考慮し**最大 30[G]相当と想定し**、評価試験を実施した。

ARLISS2019 の放出時加速度データを、1 回目分離衝撃を Fig. 6-5-1、2 回目分離衝撃を Fig. 6-5-2、3 回目分離衝撃を Fig. 6-5-3 に示す。

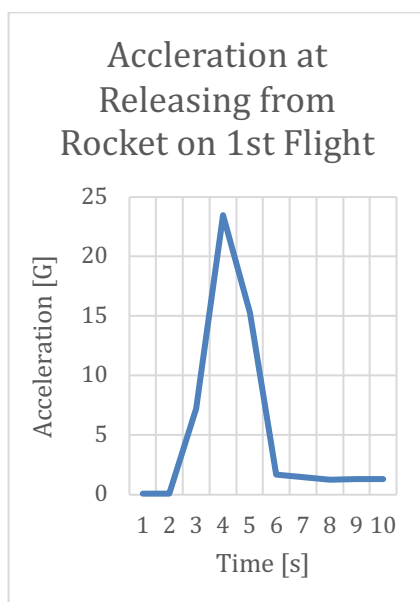


Fig. 6-5-1 分離衝撃 1 回目

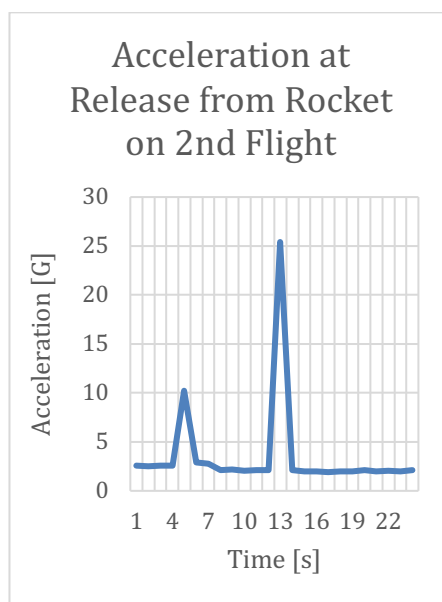


Fig. 6-5-2 分離衝撃 2 回目

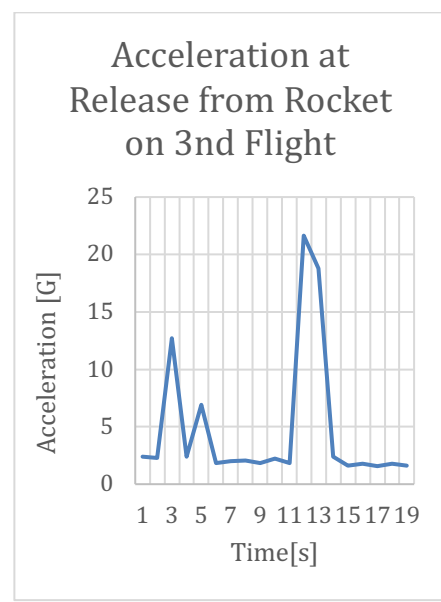


Fig. 6-5-3 分離衝撃 3 回目

30G の加速度を基準値として、CanSat のパラシュートを手で持ったまま、高い位置から自由落下させ、その際 CanSat の加速度センサを測定した結果 30G を超えた衝撃を観測したとき、30G の加速度が加わっていたときの CanSat の状態に異常が無いか評価し、問題が無ければ成功とする。

30G の加速度を基準値として、CanSat のパラシュートを手で持ったまま、高い位置から自由落下させ、その際 CanSat の加速度センサを測定した結果 30G を超えた衝撃を観測したとき、30G の加速度が加わっていたときの CanSat の状態に異常が無いか評価し、問題が無ければ成功とする。

実際の実験方法の図を Fig. 6-5-4 に示す。

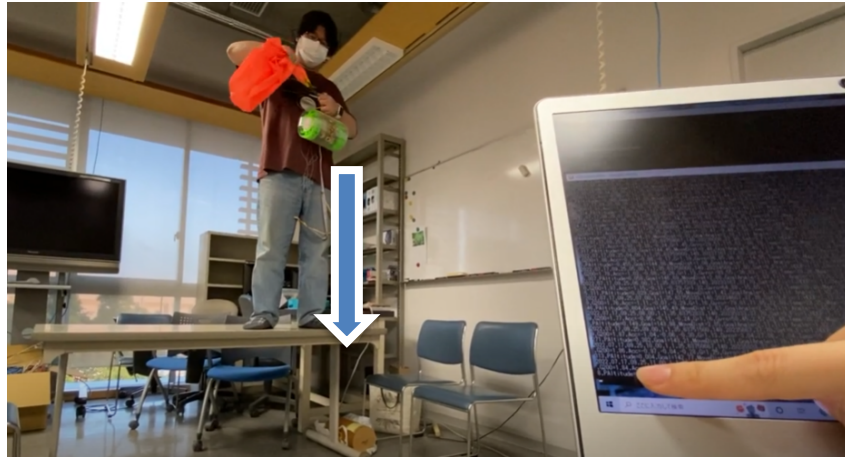


Fig. 6-5-4 実験のイメージ

● 結果

分離衝撃を回行い、約 30G の加速度がかかるようにした。実際にかかった加速度のログデータを Fig. 7-5-5 に示した。30G 以上の数値を観測している部分が分離衝撃を与えている部分となっている。

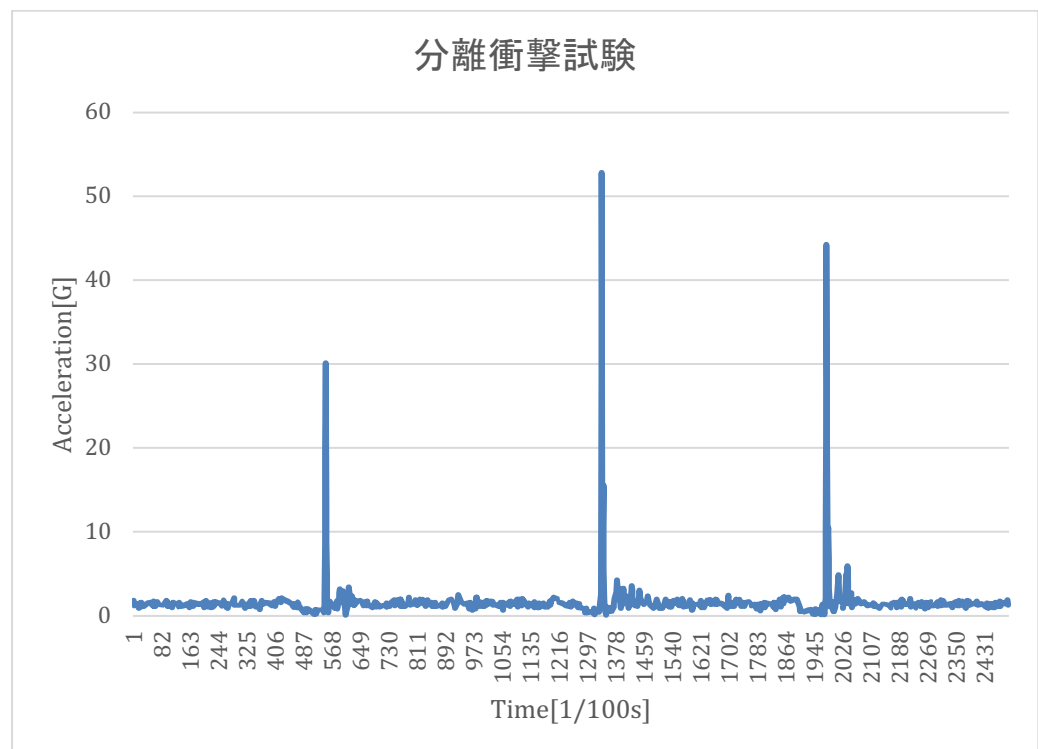


Fig. 7-5-5 分離衝撃の数値（縦軸はかかった加速度[G]，横軸は時間[1s]）

Fig. 7-5-5 から 30.1G, 52.80G, 44.26G の 30G 以上の分離衝撃を 3 回与えた。30G 以上の衝撃を 3 回以上与えた後、パラシュートの接合部、CanSat 保護カバーとの接合部に損傷は見られなかった。また機体・センサ等には損傷は見られず、その後パラシュート焼き切り機構とモータ駆動も正常に動作した。CanSat に搭載されているセンサの値や走行にも問題がないことが確認できた。

試験の様子は動画に撮り、異常の有無の確認を記録した結果からも、機体の異常は見られなかった。(下記の URL を参照)

○分離衝撃試験の動画 URL : <https://youtu.be/DW30i7LR9Nk>

(00:00 実験説明～, 00:20 分離衝撃 3 回～, 00:50 回路動作センサ確認～, 02:55 パラシュート離脱動作+モータ駆動確認～, 03:30 機体破損箇所確認～)

● 考察

試験結果より、センサなどの電子部品及び機体、モータなどが損傷した形跡はなかったため、ロケットから放出されパラシュートが開傘する際の衝撃にパラシュートとの結合部、パラシュート及び機体が耐えられると判断した。

v6. 落下試験

- 目的
パラシュートの開傘と減速が可能で、終端速度が 5m/s 程度であることを確認する。
- 試験内容
CanSat をつけたパラシュートを落下させ、落下時に測定した加速度が 5m/s 程度減速して終端速度になっているか確認した。
- 結果
実験結果を Table. 6 にまとめた。

回数	終端速度	5m/s 以上か ○or ×	実験データグラフ	動画 URL																								
1 回 目	5.089 [m/s]	○	落下試験1回目 <table><thead><tr><th>time[s]</th><th>Altitude[m]</th></tr></thead><tbody><tr><td>1</td><td>15.5</td></tr><tr><td>2</td><td>15.5</td></tr><tr><td>3</td><td>14.8</td></tr><tr><td>4</td><td>15.5</td></tr><tr><td>5</td><td>14.8</td></tr><tr><td>6</td><td>13.2</td></tr><tr><td>7</td><td>8.2</td></tr><tr><td>8</td><td>5.8</td></tr><tr><td>9</td><td>5.8</td></tr><tr><td>10</td><td>6.0</td></tr><tr><td>11</td><td>6.0</td></tr></tbody></table>	time[s]	Altitude[m]	1	15.5	2	15.5	3	14.8	4	15.5	5	14.8	6	13.2	7	8.2	8	5.8	9	5.8	10	6.0	11	6.0	https://youtu.be/xtlTvwk6RMk
time[s]	Altitude[m]																											
1	15.5																											
2	15.5																											
3	14.8																											
4	15.5																											
5	14.8																											
6	13.2																											
7	8.2																											
8	5.8																											
9	5.8																											
10	6.0																											
11	6.0																											
2 回 目	5.014 [m/s]	○	落下試験2回目 <table><thead><tr><th>time[s]</th><th>Altitude[m]</th></tr></thead><tbody><tr><td>1</td><td>14.8</td></tr><tr><td>2</td><td>15.0</td></tr><tr><td>3</td><td>15.0</td></tr><tr><td>4</td><td>15.5</td></tr><tr><td>5</td><td>10.5</td></tr><tr><td>6</td><td>6.2</td></tr><tr><td>7</td><td>6.0</td></tr><tr><td>8</td><td>5.5</td></tr><tr><td>9</td><td>5.5</td></tr><tr><td>10</td><td>5.6</td></tr><tr><td>11</td><td>5.5</td></tr></tbody></table>	time[s]	Altitude[m]	1	14.8	2	15.0	3	15.0	4	15.5	5	10.5	6	6.2	7	6.0	8	5.5	9	5.5	10	5.6	11	5.5	https://youtu.be/YK8LJwjvgM
time[s]	Altitude[m]																											
1	14.8																											
2	15.0																											
3	15.0																											
4	15.5																											
5	10.5																											
6	6.2																											
7	6.0																											
8	5.5																											
9	5.5																											
10	5.6																											
11	5.5																											

3 回 目	5.30 [m/s]	○	落下試験3回目 <table><thead><tr><th>time[s]</th><th>Altitude[m]</th></tr></thead><tbody><tr><td>1</td><td>14.2</td></tr><tr><td>2</td><td>13.8</td></tr><tr><td>3</td><td>14.2</td></tr><tr><td>4</td><td>13.4</td></tr><tr><td>5</td><td>8.1</td></tr><tr><td>6</td><td>4.7</td></tr><tr><td>7</td><td>4.2</td></tr><tr><td>8</td><td>4.5</td></tr><tr><td>9</td><td>4.2</td></tr><tr><td>10</td><td>4.7</td></tr><tr><td>11</td><td>4.2</td></tr></tbody></table>	time[s]	Altitude[m]	1	14.2	2	13.8	3	14.2	4	13.4	5	8.1	6	4.7	7	4.2	8	4.5	9	4.2	10	4.7	11	4.2	https://youtu.be/dArfhvVE2Uk
time[s]	Altitude[m]																											
1	14.2																											
2	13.8																											
3	14.2																											
4	13.4																											
5	8.1																											
6	4.7																											
7	4.2																											
8	4.5																											
9	4.2																											
10	4.7																											
11	4.2																											

【1回目】

グラフ中 6s で 13.272m と 7s で 8.183m の間 1s 間の落下高度は 5.089m であり,
 $5.089\text{m}/1\text{s}=5.089\text{m/s}$.

【2回目】

グラフ中 4s で 15.364m と 5s で 10.35m の間 1s 間の落下高度は 5.014 m であり,
 $5.014\text{ m}/1\text{s}=5.014\text{m/s}$.

【3回目】

グラフ中 4s で 13.444m と 5s で 8.144m の間 1s 間の落下高度は 5.30m であり,
 $5.30\text{m}/1\text{s}=5.30\text{m/s}$
 となり, 3回 5m/s 以上の終端速度であることが確認できた.

● 考察

パラシュートで減速したの終端速度が, 5[m/s]程度であることが確認できた.

v7. GPS データダウンリンク試験

- 目的
長距離通信が可能で、CanSat のロスト対策として有効である事を確認する。
- 試験内容
interplun IM920 (使用周波数: 920MHz 帯) を使用し、長距離での無線の受信を行い、CanSat の位置する方向の特定が可能か確認した。

Interplun IM920c(送信機)からの電波を Interplun IM920(受信機)に受信し、受信データを接続した PC に送信し Teraterm にて表示することができれば成功とする。今回の実験では、送信データは経過時間だが、本番では位置座標を送信することにより、ロスト対策を行う。

実験では 4km の通信を目標とした。4km の目標値根拠として、IM920 の最大受信距離は規格上 7km とされているが、打ち上げ高度は 4km であり、自団体のパラシュート色はオレンジ色であり空中で目視にて放出後の CanSat 方角を追跡した後に、そこから捜索をすることを目的に打ち上げ高度である 4km の追跡を目指した。また、実験を行うにあたり、地理的に見通しが取れる近隣場所は 4km が限界であるためである。

実験に使用した受信機 Interplun IM920 の写真を Fig. 6-7-1、送信機 Interplun IM920c を搭載した CanSat の写真を Fig. 6-7-2、実際に使用した送信機と受信機を PC につなげた状態の写真を Fig. 6-7-3 に示す。



Fig. 6-7-1 受信機 Interplun IM920

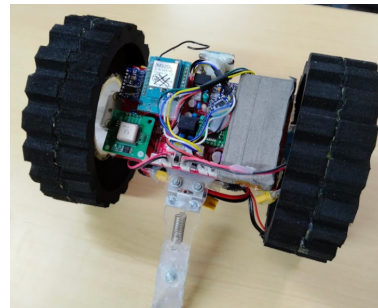


Fig. 6-7-2 送信機 Interplun IM920c 搭載 CanSat

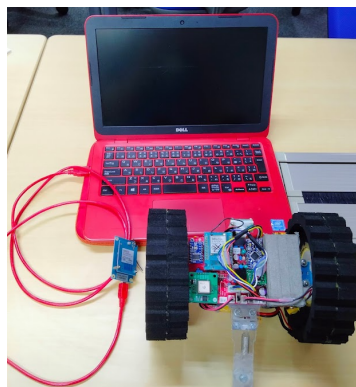


Fig. 6-7-3 実際に使用した受信機(左)と送信機(右)

また、実際に距離測定を行った位置座標を示す。

距離の指定場所：約 4.0km 【緯度 34.79416666 経度 137.13055555】
～【緯度 34.76916666 経度 137.16861111】の距離

● 結果

4km の無線受信

CanSat を組み込んだ送信機位置を【緯度 34.79416666 経度 137.13055555】に配置し、受信機を【緯度 34.76888888 経度 137.16861111】に配置し無線通信を試したところ、無線通信が成功した。

送信機位置写真を Fig. 6-7-11、送信機位置の緯度経度を Fig. 6-7-12、受信機位置写真を Fig. 6-7-13、受信機位置の緯度経度を Fig. 6-7-14、受信～送信位置の Google MAP を Fig. 6-7-15、実際に取得したデータを Fig. 6-7-16 示す。



Fig. 6-7-11 送信機の位置の写真

緯度	34.79416666		
経度	137.13055555		

⇒ 緯度経度をGoogleMapで確認

↓ ↑

緯度	34	度	47	分	39	秒	
経度	137	度	7	分	50	秒	

Fig. 6-7-12 送信機位置の緯度経度変換 (Fig. 6-7-11 の GPS ログ一時分秒からの変換)



Fig. 6-7-13 受信機の位置の写真

緯度	34.76888888			⇒ 緯度経度をGoogleMapで確認
経度	137.16861111			

↓

↑

緯度	34	度	46	分	8	秒	
経度	137	度	10	分	7	秒	

Fig. 6-7-14 受信機位置の緯度経度変換 (Fig. 6-7-12 の GPS ロガー一時分秒からの変換)

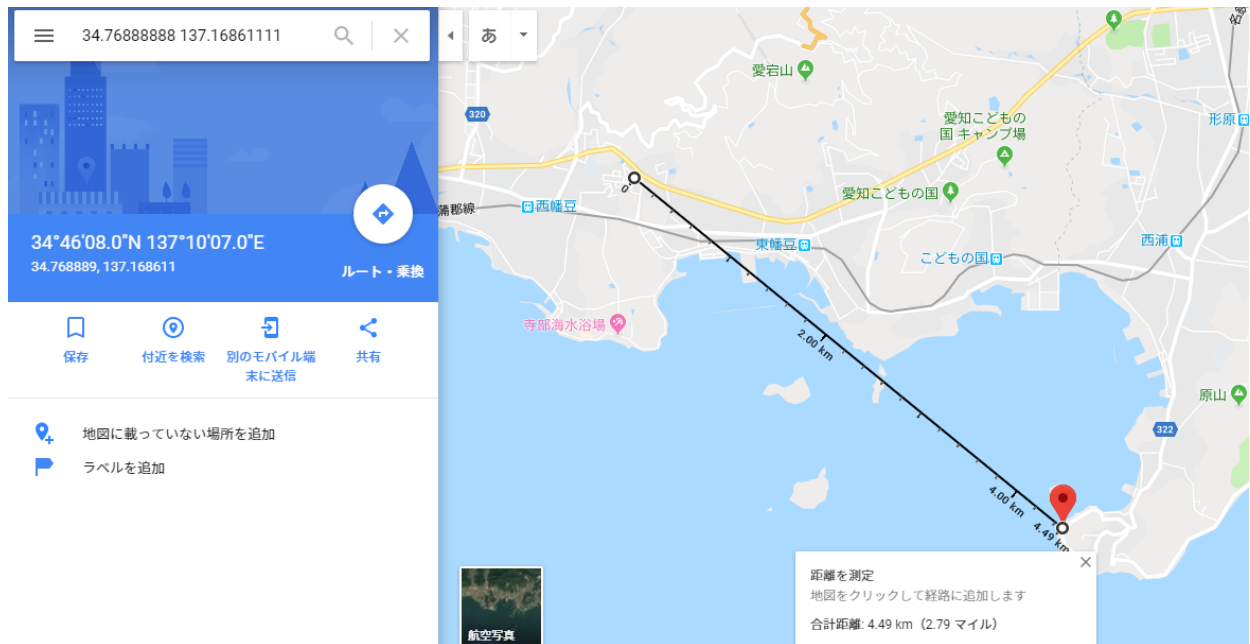


Fig. 6-7-15 4km 間の Google Map での距離測定

送信機から受信した GPS データをリアルタイムで表示されている動画を撮影した。

- 長距離通信試験：<https://youtu.be/yN6jo3j9JEs>

実験を行った結果、約 4km の長距離通信が可能であることが確認できた。

- 考察

IM920 を CanSat に搭載した状態であっても、同様に 4km 離れた地点でデータを受信することができた。これにより得たデータにより CanSat の位置を特定し、CanSat の位置を把握できることを確認した

v8. 通信機電源 OFF/ON 試験

● 目的

通信機の電波が、打ち上げ時に OFF、ロケットから CanSat が放出される際に ON となることを確認する。

● 試験内容

今回使用する通信機 Interplan IM920 は、既定の電波強度以下であるが FCC 確証でないため、気圧センサ値から求めた高度値を用いて通信機の電源供給を遮断する方式とした。

プログラムにて CanSat 電源を入れてから 20 秒後、小さい高度を検知している間はスリープモード(DSRX)に通信電波が OFF 状態になり、大きい高度を数秒間検知するとロケットが打ち上げられたと判断しその後 CanSat 電源入れた時の高度を数秒間検知するとロケットから放出され着地したと判断し、ジャイロセンサ値の変化が小さくなると地面に落ち停止している状態と判断し(ENRX)に入るプログラムを作成した。

実験では 4000m ほどのロケットの高度を観測する実験が不可能であるため、愛知工科大学 7 号館の建物の高さをロケットの高さと仮定した環境で行った。大学のエレベータを利用し、CanSat 電源 ON 時 1F にて高度を検知し、通信電波が OFF となっていることを確認し、その後 9F へ上昇しロケットが上昇した状態と仮定し、高度値が初期値から 15m 以上に上昇していることを 10 秒間検知したあと、1F へ降下し高度が初期値に近い値として 15m 以下を 10 秒間検知したあと、ジャイロセンサの変化が小さくなると、地面に落ち停止している状態と判断し通信電波が ON になることを確認した。

また、通信条件の手順を以下の①②③に示す。

- ① CanSat 電源 ON 時 20 秒間通信電波 ON、その後電波 OFF
- ② 高度 15m 以上を 10 秒間検知するとスリープモード(DSRX)で通信電波 OFF 状態
- ③ 高度 15m 以内を 10 秒間連続で検知したあと④のプログラムへ移行
- ④ ジャイロセンサの変化が小さくなると、地面に落ち停止している状態と判断し通信モード(ENRX)で無線 ON 状態

また、実験機器として、PC に無線ログと、CanSat へ有線で接続した時のプログラムのリアルタイムログを並列に表示させ、更に電波観測機としてスペクトラムアナライザで電波を測定し、電波の ON/OFF を目視で確認できるようにした。実際の実験機器を Fig. 6-8-1 に示す。

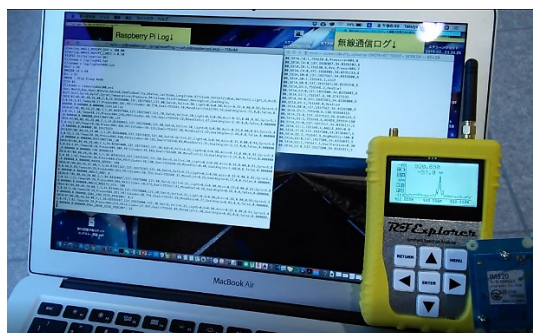


Fig. 6-8-1 電波 ON/OFF を目視で確認するための実験機器

● 結果

実際の結果を、通信機 OFF を①、設定した高い高度を観測したかを②、高い高度から落下し終え通信機が ON にしたかを③として結果をまとめた。

- ① 建物 1F で CanSat 電源 ON し 20 秒後、無線通信ログが停止し、スペクトラムアナライザが 920MHz 帯の電波を観測しなくなり、無線通信が OFF になった。実際の実験で PC に無線ログと、CanSat へ有線で接続した時のプログラムのリアルタイムログと、スペクトルアナライザの表示写真を Fig. 6-8-2 に示す。

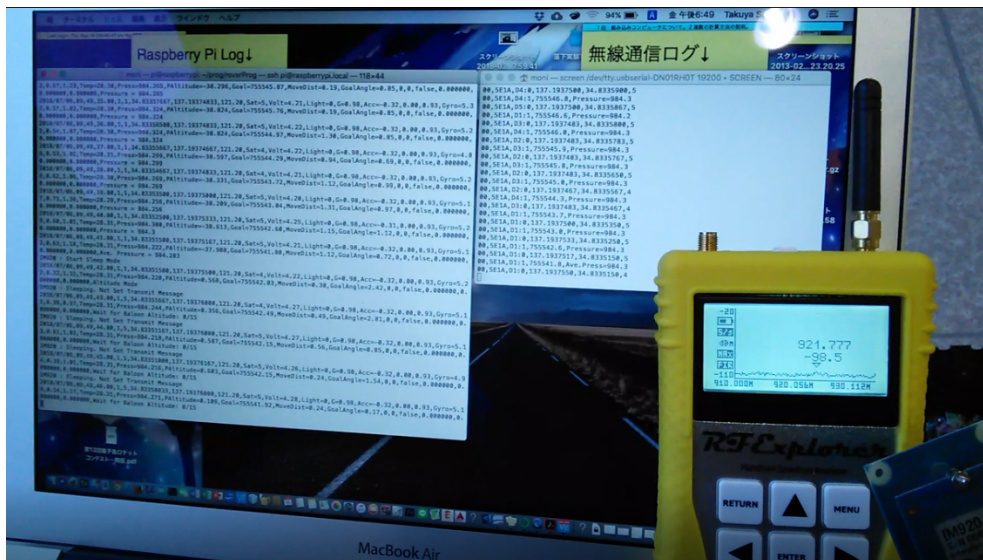


Fig. 6-8-2 電源 OFF 確認写真

- ② 建物 9F 上昇後、Raspberry Pi Log より、高度 31m を観測し、上昇したことを観測した。
実際の高度観測リアルタイムログの写真を Fig. 6-8-3 に示す。

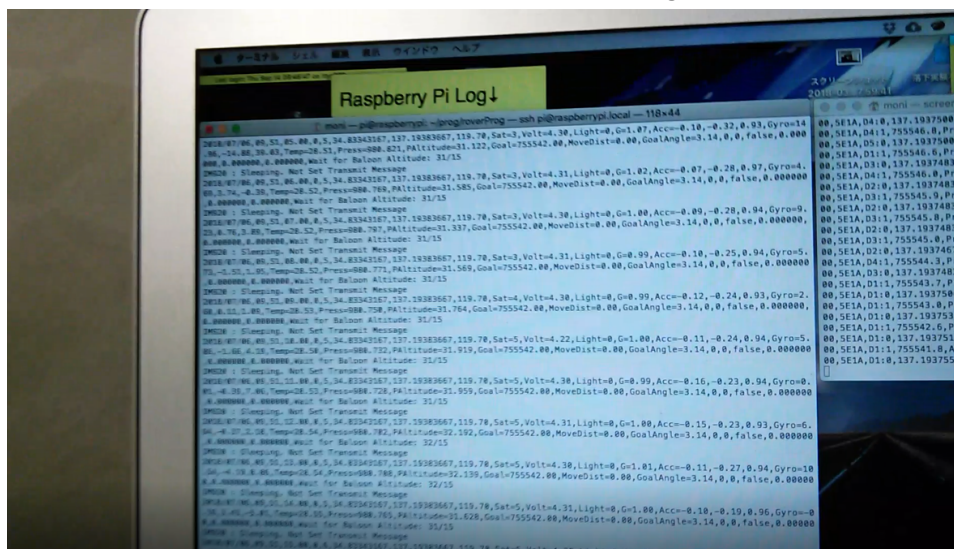


Fig. 6-8-3 高い高度の観測

- ③ 1F へ降下し、CanSat を地面に置き停止させ、高度が降下したことを観測し、ジャイロセンサ値の変化が小さくなったことから、地面に着地したと判断し、無線電波が ON になった。実際の実験で PC に無線ログと、CanSat へ有線で接続した

時のプログラムのリアルタイムログと、スペクトルアナライザの表示写真を
Fig. 6-8-4 に示す。



Fig. 6-8-4 高度降下後の無線 ON 確認写真

また、この試験の様子を動画にまとめた。（下記 URL を参照）

○通信機電源 ON/OFF 試験の動画 URL : <https://youtu.be/i2LnoMkIAzI>

試験により、通信機電波がロケット格納時に OFF、ロケットから CanSat が放出される際に ON となることを確認できた。

● 考察

結果、試験により打ち上げ時の無線機電波 OFF の規定を遵守できることを確認した。

v9. 通信周波数 ch 変更試験

● 目的

無線のチャンネル調整に応じる意思があり、実際に調整ができることを確認する。

● 試験内容

使用する予定の無線(interplan IM920)をプログラムにより、状況に応じて使用する通信機を切り替える措置を行った。

プログラム上でチャンネル変換を行い、データを受信する Tera Term 上で【stch (チャンネル数)】を入力後データが受信される仕様となっている。
これらの機能を用いて、無線チャンネル変更ができることを確認する。

● 結果

実際に利用している無線設定プログラムの説明を①、そのプログラムのチャンネル変更を②として実験の結果を記述する。

- ① M920 と接続された Raspberry Pi zero W メインプログラム上で
【sc16is750.setChannel();】の () 内の数字をチャンネル数に変更しコンパイルすることでチャンネルが変更される。

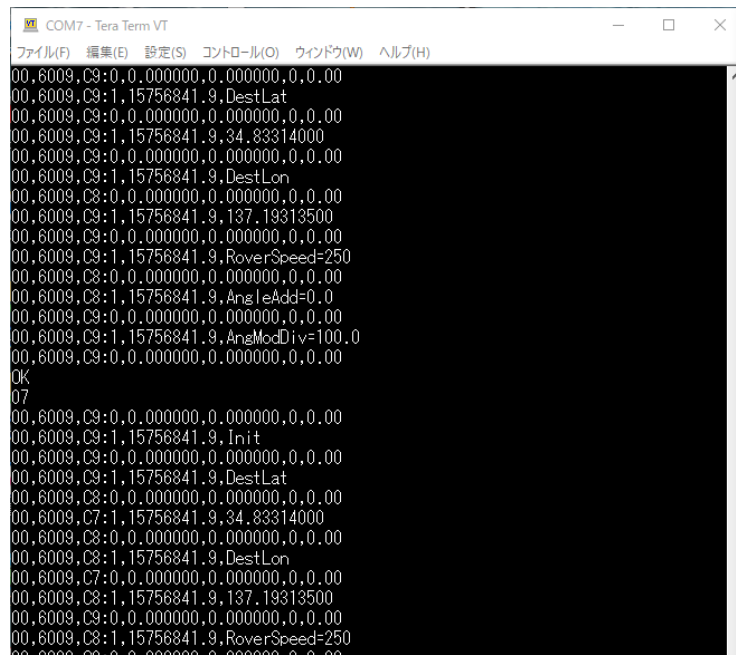
実験では 5 チャンネルから 7 チャンネルに設定変更を行った。実際のプログラムを Fig. 6-9-1 に示す。

```
1338
1339 // IM920の送信チャンネルを設定する
1340 // 1-15 (default:1)
1341 sc16is750.setChannel( 1 );
1342
1343 // IM920の送信出力を設定する
1344 // 1: -10dBm (0.1mW)
1345 // 2: 0dBm ( 1mW)
1346 // 3: 10dBm ( 10mW)
1347 sc16is750.setTxPower( TX_POWER_STARTUP );
```



Fig. 6-9-1 チャンネル変更のプログラム

- ② IM920 が接続された PC の TeraTerm 上で【STCH 07】とコンパイルすると、チャンネルが 7 になる。実際のデータログを Fig. 6-9-2 である。



```
COM7 - Tera Term VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
00,6009,C9:0,0.000000,0.000000,0,0.00
00,6009,C9:1,15756841.9,DestLat
00,6009,C9:0,0.000000,0.000000,0,0.00
00,6009,C9:1,15756841.9,34.83314000
00,6009,C9:0,0.000000,0.000000,0,0.00
00,6009,C9:1,15756841.9,DestLon
00,6009,C8:0,0.000000,0.000000,0,0.00
00,6009,C9:1,15756841.9,137.19313500
00,6009,C9:0,0.000000,0.000000,0,0.00
00,6009,C9:1,15756841.9,RoverSpeed=250
00,6009,C8:0,0.000000,0.000000,0,0.00
00,6009,C8:1,15756841.9,AngleAdd=0.0
00,6009,C9:0,0.000000,0.000000,0,0.00
00,6009,C9:1,15756841.9,AngModDiv=100.0
00,6009,C9:0,0.000000,0.000000,0,0.00
OK
07
00,6009,C9:0,0.000000,0.000000,0,0.00
00,6009,C9:1,15756841.9,Init
00,6009,C9:0,0.000000,0.000000,0,0.00
00,6009,C9:1,15756841.9,DestLat
00,6009,C8:0,0.000000,0.000000,0,0.00
00,6009,C7:1,15756841.9,34.83314000
00,6009,C8:0,0.000000,0.000000,0,0.00
00,6009,C8:1,15756841.9,DestLon
00,6009,C7:0,0.000000,0.000000,0,0.00
00,6009,C8:1,15756841.9,137.19313500
00,6009,C9:0,0.000000,0.000000,0,0.00
00,6009,C8:1,15756841.9,RoverSpeed=250
00,6009,C8:0,0.000000,0.000000,0,0.00
```

Fig. 6-9-2 実験でチャンネル変更したデータログ

無線 (interplan IM920) のチャンネル変更については、プログラムによりチャンネルを変更し実際に通信できることを確認した。

また、この試験の様子を動画にまとめた。(下記 URL を参照)

○通信周波数変更試験の動画 URL : <https://youtu.be/Kq6UdYgvaNI>

● 考察

無線のチャンネル調整が可能であることが確認できた。

v10. End to End 試験

● 目的

CanSat の動作を本番と同じ手順で行う。各シーケンスを自律で行えることを確認できる動画を提出する。

● 試験内容

- 1 キャリアから CanSat が自重で放出
 - 2 着地衝撃
 - 3 パラシュートからの離脱
 - 4 目的地まで GPS 位置情報で自律走行
 - 5 Deep Learning Object Detection ゴール接近&判定
 - 6 ログデータ取り出し
- までの一連の流れを通してできるか確認する。

● 結果

実験結果を Table. 15 に示す。

Table. 15 End to End 結果

回数	成功 Or 失敗	詳細	URL
1 回目	成功	【キャリア放出～GPS での走行】 キャリア放出後、サンセンサの値がキャリア内から外へ放出したと判定後、パラシュートと分離し、GPS 情報からゴール近くまでへ走行した。 【Object Detection でのゴール検知～ログ取り出し】 ゴール約 5m 付近から Object Detection でのゴール検知を開始。ゴールがカメラに映るまで旋回し、0m まで接近後ゴール判定した。その後 Object Detection ログ画像の取り出しと、走行軌跡を出力するプログラムを動作させ EndtoEnd の走行奇跡ログデータを取り出すことができた。	https://youtu.be/NTiNaUbnBk 00:24 キャリア収納 01:28 放出 01:37 着地衝撃 02:40 パラシュート離脱動作 03:00 GPS 走行開始 03:30 Object Detection 開始 04:14 ゴール判定 05:12 ログ取り出し走行軌跡 06:11 Object Detection 画像ログ
2 回目	成功	【キャリア放出～GPS での走行】 キャリア放出後、サンセンサの値がキャリア内から外へ放出したと判定後、パラシュートと分離し、GPS 情報からゴール近くまでへ走行した。 【Object Detection でのゴール検知～ログ取り出し】 ゴール約 5m 付近から Object Detection でのゴール検知を開始。ゴールがカメラに映るまで旋回し、0m まで接近後ゴール判定した。その後 Object Detection ログ画像の取り出しと、走行軌跡を出力するプログラムを動作させ EndtoEnd の走行奇跡ログデータを取り出すことができた。	https://youtu.be/pubwN5vwaWI 00:24 キャリア収納 01:01 放出 01:18 着地衝撃 02:26 パラシュート離脱動作 02:38 GPS 走行開始 03:32 Object Detection 開始 03:49 ゴール判定 04:51 ログ取り出し走行軌跡 05:33 Object Detection 画像ログ

また、制御履歴を V11 制御履歴レポート試験にて記載した。

● 考察

End to End 試験により、競技一連の流れに問題がないことが確認できた。本番までに繰り返し試験を行い、精度を高めて行く。

v11. 制御履歴レポート試験

● 目的

End to End 試験のデータから作成し、実験当日と大会終了後の大会報告書で制御履歴の提出できることを確認する。

● 試験内容

プログラムを起動した地点をスタート地点として、スタート地点からゴール地点へ走行した時のデータをマイコンのSDカードにファイルを保存する。実験後、GPS や制御値等のデータファイルをSDカードから読み取り、text 生データをExcel ファイルでまとめ見やすい形にして表示した。Excel データはサンプリング1s ずつ取得したファイルとサンプリング0.01s ずつ取得したデータを、まとめて提出する。また、走行軌跡を画像化し、走行軌跡が見やすくなるように色変化で表示した。

● 結果

実際に V15End to End で行ったデータを用いて、制御履歴作成を行った。

【走行軌跡】

制御履歴提出に色付き及び矢印付きの走行軌跡図を作成した図を、End to End 1 回目を Fig. 16-1 に、End to End 2 回目 Fig. 16-2 に示す。

走行軌跡図では、右旋回へのステアリング最大値 0.20 (黄色)、左旋回へのステアリング最大値 -0.20 (紫) として緯度経度から走行軌跡を色で示した。

また、スタート地点、ターゲット地点、ストップ地点を矢印で示し、その地点の緯度経度を示した。

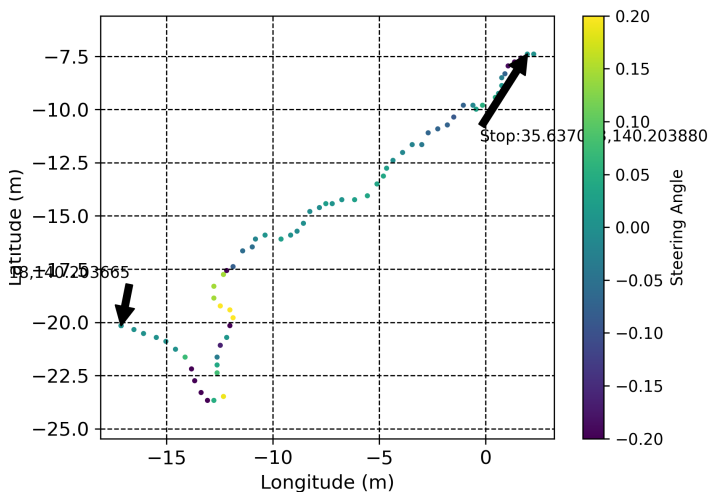


Fig. 16-1 1回目 End to End 走行軌跡

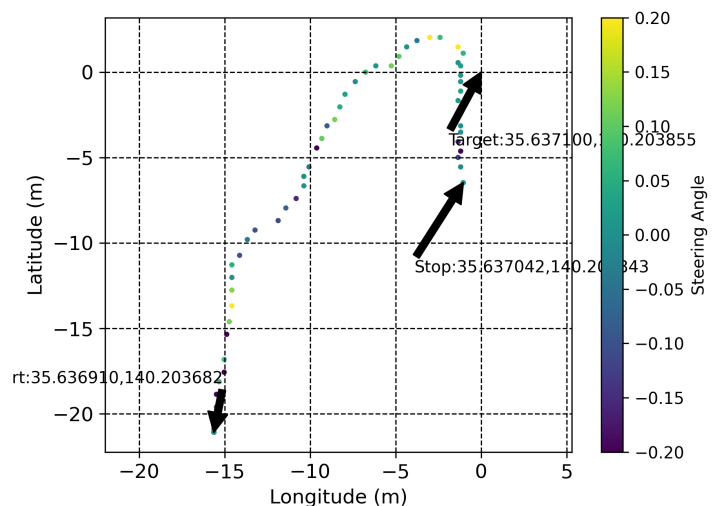


Fig. 16-2 2回目 End to End 走行軌跡

【制御履歴説明】

制御履歴 text 生データを, Excel ファイルで見やすくまとめた.

基本的な制御出力系データを確認できる. Excel ファイルは 5p 後のドライブ URL から確認できる.

制御項目は,

時系列データ : A~D 列

A	B	C	D	E	F
Year	Month	Day	Hour	Minute	Second
2019	8	15	7	22	10
2019	8	15	7	22	11
2019	8	15	7	22	12
2019	8	15	7	22	13

制御出力系データ : AF~AS 列

AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	
Goal Angle	Drive Speed	Speed Division	Steering Modify	Steering Angle	Steering Modify	Angle Actual	Steering Angle Target	Motor Rev R	Target Motor Rev L	Actual Motor Rev R	Actual Motor Rev L
-1.58	0	9	FALSE	0	0	0	0	110	66	111	76
-1.58	250	5	TRUE	-0.504014	-0.00504	-0.127263	0	0	44	36	
-1.59	250	5	TRUE	-0.505619	-0.010096	-0.128929	122	106	132	115	
0	250	5	TRUE	0	-0.010096	-0.002524	122	106	127	108	
-0.88	250	5	TRUE	-0.279795	-0.012894	-0.073172	122	122	124	113	
-0.39	250	5	TRUE	-0.125159	-0.014146	-0.034826	122	113	114	119	
-0.69	250	5	TRUE	-0.218401	-0.01633	-0.058683	122	118	144	93	

GPS 位置情報値 : I, J 列

I	J
Latitude	Longitude
40.14226333	139.9874683
40.14226333	139.98747
40.14226333	139.9874733
40.14226333	139.9874733

に記されています.

また, GPS による制御方法のアルゴリズムを Fig. 16-3 に示す.

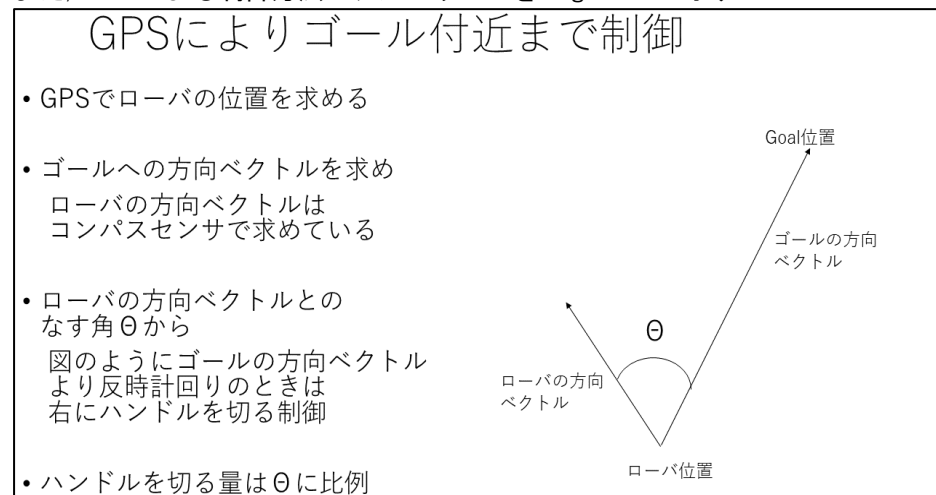


Fig. 16-3 GPS による制御方法

制御出力系データの項目の説明を下記に示す.

・ **AH 列 CanSat 進行方向とゴールとの角度[rad]** : 1 秒前と現在の CanSat の一から CanSat の進行方向とゴールとの角度をラジアン単位で求め, 制御に反映しています.

・ **AI 列モータ出力スピード(-255~+255)** : モータに加える出力を-255~+255(右~左)で与えています.

・ **AJ 列スピード分解能** : GPS を 1 秒ごとで制御していますが, 新しい速度にする際のノイズ対策として, 指定時間(0.1 秒)の線形変化でモータの出力を変換しています.(電圧の変化を少量ずつにする事で, ノイズ対策)

・ **AL 列ステアリング補正(TRUE or FALSE)** : 制御値から CanSat にステアリング補正を与えるかどうかを判定し, TRUE か FALSE で表しています.

・ **AC 列ステアリング角度(-1~+1) (-1 : 左, +1 : 右)** : 左旋回最大-1, 右旋回最大+1 として, -1~+1 でステアリング角度を表しています.

・ **AM 列修正舵(ステアリング I 制御)** : モータ損傷時を考慮し, 制御が片方のステアリング角度が著しく大きい時, もう片方のステアリング角度を大きくするよう I 制御を与えています.

・ **AN 列実際のステアリング角度(-1~+1)** : AC 列ステアリング角度を AD 列修正舵に適応させた, 実際にモータへ出力するステアリング角度を表している.

・ **AO 列右モータ目標回転数(-255~255)** : 回転数制御による右モータステアリング目標回転数.

・ **AP 列左モータ目標回転数(-255~255)** : 回転数制御による左モータステアリング実際の測定結果の回転数.

・ **AQ 列右モータ目標回転数(-255~255)** : 回転数制御による右モータステアリング目標回転数.

・ **AR 列左モータ目標回転数(-255~255)** : 回転数制御による左モータステアリング実際の測定結果の回転数.

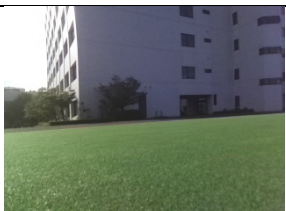
【Object Detection 画像ログ】

また、End to Endでの Object Detection の画像ログ 1 回目を Table 16-1 に、2 回目を Table 16-2 示す。

Table 16-1 End to End 1 回目 Object Detection

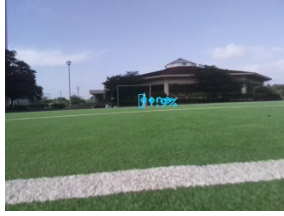
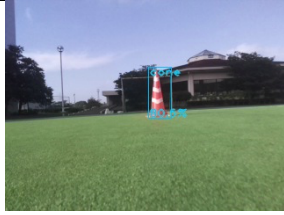
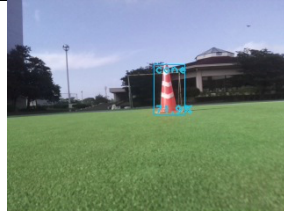
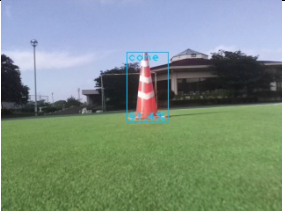
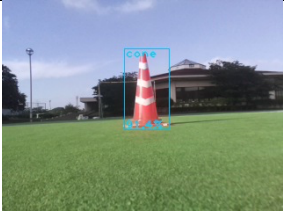
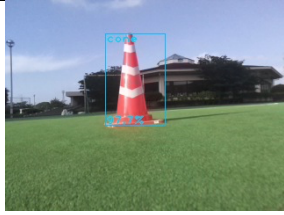
Object Detection での 0m ゴールまでの画像ログ & 詳細

ゴール発見まで 16 回巡回 (pic1-16)

				
pic1	pic2	pic3	pic4	pic5
				
pic6	pic7	pic8	pic9	pic10
				
pic11	pic12	pic13	pic14	pic15
				
pic16				

発見後、42 回の巡回 & 直進でゴール判定 (pic17-59)

				
pic17	pic18	pic19	pic20	pic21
				

pic22	pic23	pic24	pic25	pic26
				
pic27	pic28	pic29	pic30	pic31
				
pic32	pic33	pic34	pic35	pic36
				
pic37	pic38	pic39	pic40	pic41
				
pic42	pic43	pic44	pic45	pic46
				
pic47	pic48	pic49	pic50	pic51
				
pic52	pic53	pic54	pic55	pic56

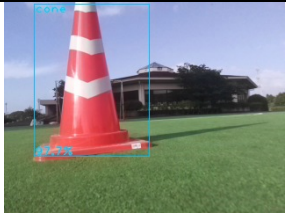
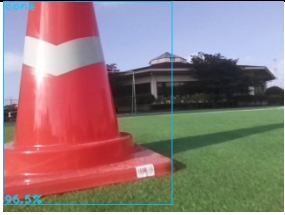
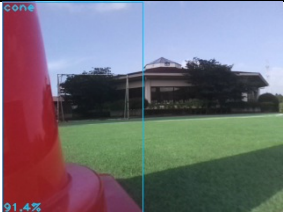
 pic57	 pic58	 pic59		
---	--	--	--	--

Table 16-2 End to End 2回目 Object Detection

Object Detectionでの0m ゴールまでの画像ログ&詳細

ゴール発見まで 42 回巡回 (pic1-42)



また, 実際の制御ログデータをまとめた URL を下記に示す.

OEnd to End 制御履歴 text 生データ, Excel ファイル, 走行軌跡, ゴール画像 :
<https://drive.google.com/drive/folders/172xsRoIKTe2KOWx2SyQnNpZD7oJEGBYi?usp=sharing>

● 考察

CanSat が目的地まで自律走行し, ゴール判定した後に停止するまでの詳細なデータ

を作成することが可能であり、制御履歴レポートを運営者に提出できることを確認した。

v12. 着地衝撃試験

● 目的

着地時に発生する衝撃に機体が耐えられるかどうかを確認する。

● 試験内容

【シュミュレーション_試験内容】

CanSat 落下し着地した際の荷重のシュミュレーションを行った。

ARLISS2019 では3回打ち上げ、1回目約 56.11[G]、2回目約 25.72[G]、3回目約 61.19[G]が実際に加わった。この3回分の着地衝撃の中から最大値である3回目約 61.19[G]を最大着地衝撃の数値として扱う。

ARLISS2019 1回目加速度のグラフを Fig. 6-12-1、ARLISS2019 2回目加速度のグラフを Fig. 6-12-2、ARLISS2019 3回目加速度のグラフを Fig. 6-12-3 に示す。

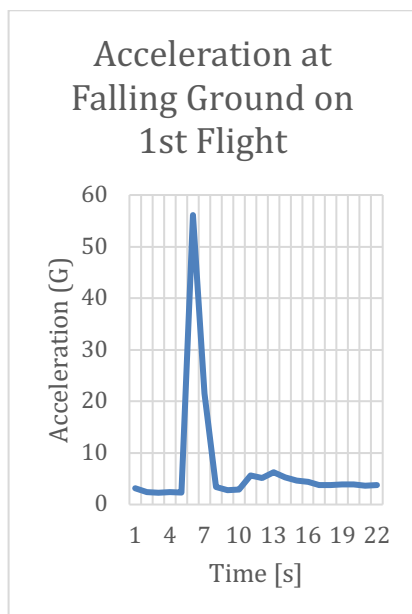


Fig. 6-12-1 1回目着地衝撃

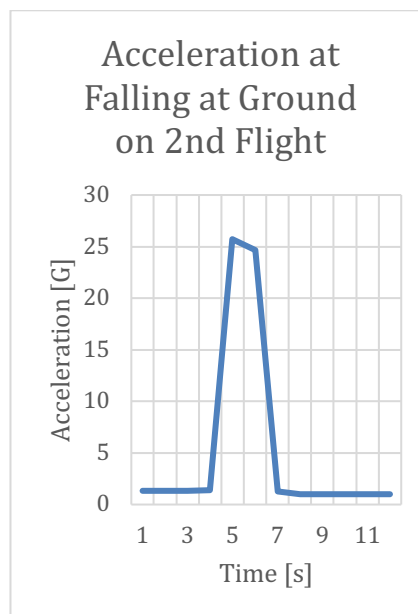


Fig. 6-12-2 2回目着地衝撃

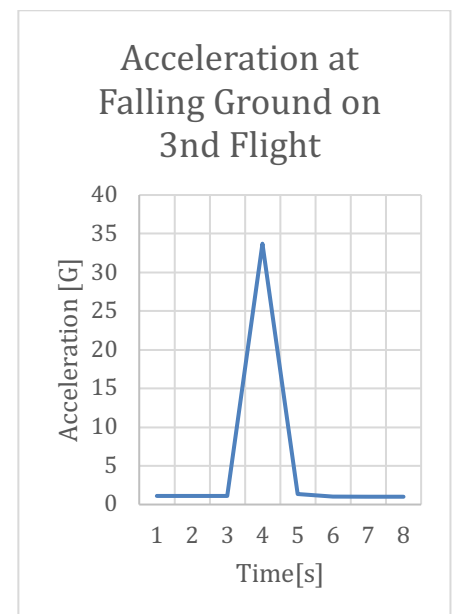


Fig. 6-12-3 3回目着地衝撃

CanSat の質量 0.850kg (V1 質量試験参照) で、ARLISS2019 で着地衝撃時に観測された最大約 61.19G を基準値として合計荷重計算をした。

機体は機体側面のタイヤから着地を行い、タイヤへ基板・モータ・バッテリーの荷重が加わるため、荷重はタイヤの質量を引いた機体の値でシュミュレーションを行う。

【実測_試験内容】

パラシュートの終端速度から自由落下したときと衝撃が同じになる高さを求め、その高さから自由落下させたときに CanSat に故障がないか確認する。

終端速度から自由落下の高さを以下に求めた。

(v^2 = パラシュート終端速度 5m/s) (a = 重力加速度 9.8g)

$$\text{終端速度 (パラシュート投下試験参照)} = \frac{v^2 - v_0^2}{2a} = \frac{5^2 - 0}{2 \times 9.8} = 1.275[m]$$

1.275[m] 以上の高さから CanSat を自由落下させ、その後電子回路の GPS やセンサ類の値に以上がないか、モータに以上がないか、CanSat が破損していないかの確認を行った。

● 結果

【シミュレーション_結果】

タイヤ1つの質量を測定した結果、タイヤは0.062kgであり、タイヤは2輪であるため、タイヤ2つ分の質量0.124 kgとなり、機体全体質量0.850 kg(質量試験参照)から引くと荷重の加わる質量は0.711 kgとなる。0.711 kgの質量に61.19 G加えると**合計荷重は44.42 kgf**となる。計算式を以下の6.12.1に示す。

$$\begin{aligned} 0.850 - (0.062 \times 2) &= 0.726 \\ 0.726 \times 61.19 &= \mathbf{44.42} \end{aligned} \quad \dots 6.12.1$$

合計荷重の 76.0 kgf をタイヤとモータ結束部分2つの部分へ分散させると、22.21 kgf ずつ加わる。計算式を下記の7.8.2に示す。

$$44.42 / 2 = \mathbf{22.21} \quad \dots 6.12.2$$

以上のシミュレーション結果をわかりやすく示した図をFig. 6-12-4に示す。
また、計算式 6.12.1 で示した 0.118 kg のタイヤの質量の測定根拠となる写真をFig. 6-12-5に示す。

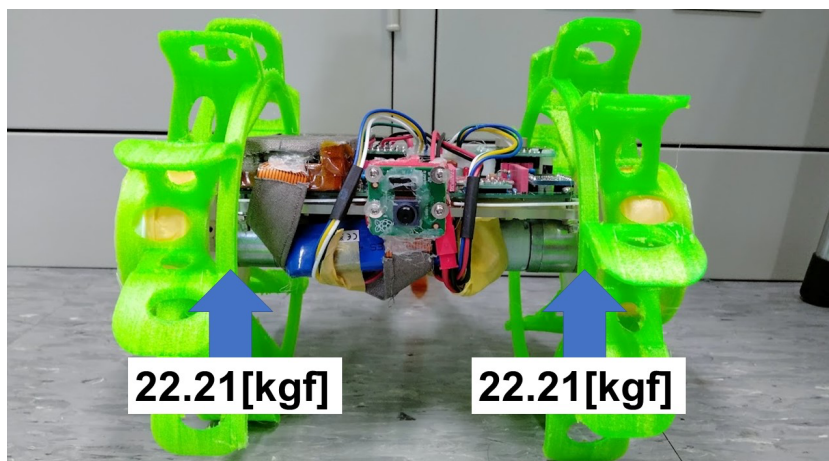


Fig. 6-12-4 シミュレーション結果



Fig. 6-12-5 タイヤの重量

【実測_結果】

着地衝撃を3回与えた。実際にかかった加速度のログデータをFig. 6-12-6に示す。

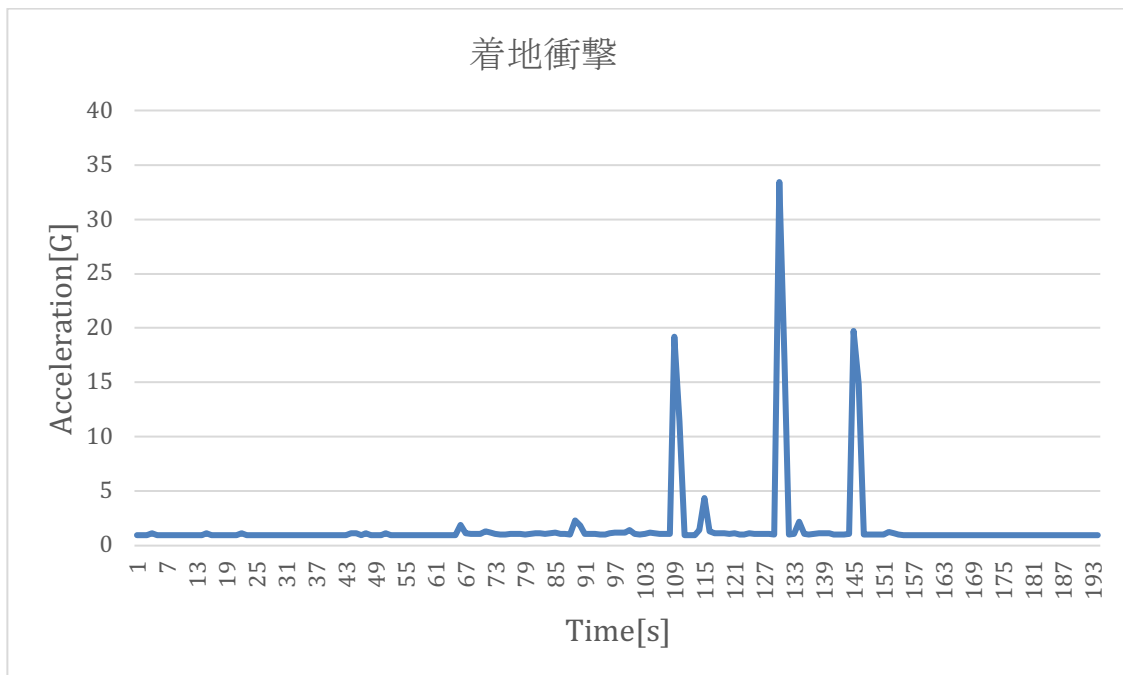


Fig.6-12-6 着地衝撃の数値グラフ(縦軸はかかった加速度[G], 横軸は時間[1s])

Fig. 6-12-6 から観測した着地衝撃は1 回目 19.23[G], 2 回目 33.42[G], 3 回目 19.74[G]だった

タイヤの設計変更(3.2. 機体の内観・機構 参照)により, タイヤの衝撃吸収機構が働き, ARLISS2019 の従来タイヤより着地衝撃値が小さくなっていることが確認できた.

また, 電子回路の GPS やセンサ類の値, パラシュート離脱機構, モータに異常がないか, CanSat 機体が破損していないかの確認を行い, 故障無く CanSat に問題がないことを確認した.

着地衝撃試験の動画に撮り, Youtube にて確認できるようにした. 実際の試験の様子を下記 URL に示す. (下記の URL を参照)

○着地衝撃試験の動画 URL : https://youtu.be/L044C xv_eVY

(0:33 回路センサ動作 事前確認, 2:12 落下衝撃 1 回目, 2:32 落下衝撃 2 回目, 2:49 落下衝撃, 3 回目, 3:05 回路センサ動作確認, 4:40 パラシュート離脱機構 動作確認, 4:53 モータ動作確認(直進走行), 4:13 機体破損 確認)

● 考察

CanSat が着地衝撃に耐え, 問題なく動作することを確認した. タイヤにより衝撃吸収できることが確認でき, タイヤの構造を今後様々な設計しより衝撃吸収ができるよう試していく.

v13. 走行性能確認試験

- 目的

本番のフィールド上で問題なく走行できることを確認する。

- 試験内容

能代宇宙イベントの大きな草の状態などに似た、大きな草の障害物のあるフィールドで CanSat を走行させ、CanSat がスタックしないか確認した。

能代宇宙イベント競技にて茎が太く大きな草にスタックし走行不能となる競技結果を過去に多く経験した。そのため競技フィールドに似た大きな草が生えている土地で CanSat を走行させ、今回のミッションであるタイヤの形状を草がスタックしにくいように変更することで、茎が太く大きな草にスタックしても離脱ができることを確認した。

実験での茎が太く大きな草への画像の例を、Fig. 6-13-1 に示す。



Fig. 6-13-1 実験のイメージ図

また、ARLISS での轍を走行できるか、地面に穴を掘り轍を再現し、走破ができるかの試験を行った。実験での轍の画像の例を Fig. 6-13-2 に示す。



Fig. 6-13-2 轍走破の実験のイメージ図

● 結果

走破性能確認試験の結果を Table 13-1 に示す。CanSat は長さ約 12cm 太めの草や、長さ約 34cm 細めの草や、長さ約 38cm 細めの草を走破した。

Table 13-1 走破性能確認試験結果

実験回数	走破	草の状態	結果詳細	動画 URL
1 回目	○	長さ約 12cm 太め 	約 12cm の長さ太い草を走破した。	https://youtu.be/cPIhGK1oQ1s
2 回目	○	長さ約 34cm 細め 	約 34cm の長さの細い草を走破した。	
3 回目	○	長さ約 38cm 細め 	約 38cm の長さの細い草を走破した。	

また、轍が乗り越えられるか轍の穴を掘り再現し実験を行った。
 今回の CanSat の拡張タイヤの経は 17cm であり、走破できる限界の段差の理論では半径の長さ 8.5cm の段差が走破ができるはずである。まず深さ 8.5cm より低い条件である 5cm の轍を乗り越えられるか確認を行った後、8.5cm で轍が走破できるか確認を行った。

Table 13-2 走破性能確認試験 轍の乗り越え

実験回数	走破	轍の大きさ	結果詳細	動画 URL
1 回目	○	深さ 5 cm 大きさ 30cm 走行しながら乗り越え 	深さ 5 cm 大きさ 30cm の轍を、走 行しながら 乗り越えた	https://youtu.be/HKwebfkbkw
2 回目	○	深さ 5 cm 大きさ 30cm 走行しながら乗り越え(1 回目逆方向から) 	深さ 5 cm 大きさ 30cm の轍を、(1 回目と逆方 向から)走 行しながら 乗り越えた	https://youtu.be/fI319fFgzD4
3 回目	○	深さ 5 cm 大きさ 30cm 轍の中から乗り越え 	深さ 5 cm 大きさ 30cm の轍を、轍 の中から乗 り越えた	https://youtu.be/ppsg0ew83eE
4 回目	○	深さ 5 cm 大きさ 30cm 轍の中から乗り越え(3 回目逆方向から) 	深さ 5 cm 大きさ 30cm の轍を、(1 回目と逆方 向から)轍 の中から乗 り越えた	https://youtu.be/ppsg0ew83eE

5 回目	○	深さ 8.5 cm 大きさ 33cm 走行しながら乗り越え  	深さ 8.5 cm 大きさ 33cm の轍を、走行 しながら乗り 越えた	https://youtu.be/Rr4SXES8bZg
6 回目	○	深さ 8.5 cm 大きさ 33cm 走行しながら乗り越え (5 回目逆方向から)  	深さ 8.5 cm 大きさ 33cm の轍を、(5 回目逆方向か ら)走行しな がら乗り越え た	
7 回目	○	深さ 8.5 cm 大きさ 33cm 轍の中から乗り越え  	深さ 8.5 cm 大きさ 33cm の轍を、轍の 中から乗り越 えた	https://youtu.be/42zvByzf3I
8 回目	○	深さ 8.5 cm 大きさ 33cm 轍の中から乗り越え (7 回目逆方向から)  	深さ 8.5 cm 大きさ 33cm の轍を、(7 回目と逆方向 から)轍の中 から乗り越え た	

● 考察

草の多い地面や轍での走破ができることが確認できた。よりスタックが無いようにタイヤのグリップ部分やサイズ、形状について検討を重ねていく

v14. ゴール検知試験

- 目的

ミッションである Deep Learning Object Detection による, 0m ゴールができるか確認する.

- 試験内容

目標位置から約 10m 以上離れた地点から, GPS で数 m まで接近後, 目標カラーコーンを Object Detection による画像認識で接近し 0m ゴールを行う. 下記に Object Detection の仕様詳細等を説明する.

【Object Detection による 0m ゴール高速化】

Object Detection では 1 枚の画像から物体を複数検知でき、画像中どこにあるかがわかる。1 枚の画像でゴールの他にも人や気球なども同時に検知でき、それぞれ物体がどの位置にあるのかも座標情報で取得できる。

昨年時に利用していた Image Classification は Raspberry Pi Zero W で可能な限り短時間で認識できるよう Tensorflow から Tensorflow Lite に変換すると、1 回ゴール認識に約 12 秒程度かかった。そこで Deep Learning 用計算アクセラレータの Coral USB Accelerator を Raspberry Pi Zero W に接続し、高速化した。これにより、1 回ゴール認識は約 12 秒から約 0.3 秒でできるようになっている。

【Object Detection 使用詳細】

○モデル : SSD MobileNet V1

○入力サイズ : 300×300pixel, RGB3 チャンネル

○学習方法 : coco 90 objects で学習済みモデルを約 5000 枚のゴール画像で cone, person, parachute, balloon, blue cone の 5 種類にアノテーションされたデータで転移学習を行った。学習は 1500 ステップ行った。

また、実際のアノテーションの例を Fig. 14-1 に示す。

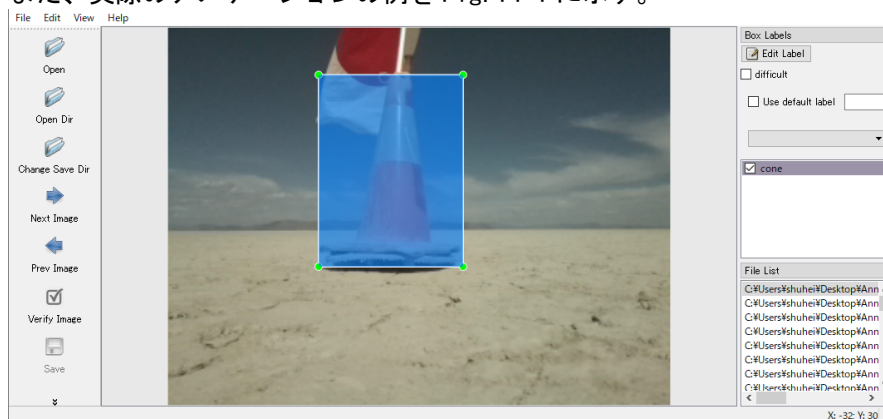


Fig. 14-1 Annotations の例

● 結果

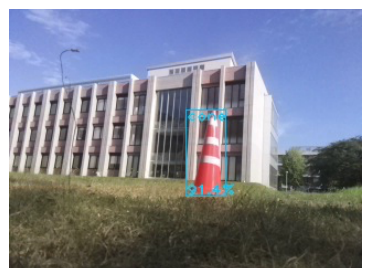
ゴール検知試験の結果を Table 10 に示す。画像ログでは Object Detection でゴールを発見すると水色の四角枠に画像で示されるようになっている。画像は上から下、左から右の順に、認識した 0m ゴールまでのログ画像を示す。

Table 10 ゴール検知試験 結果

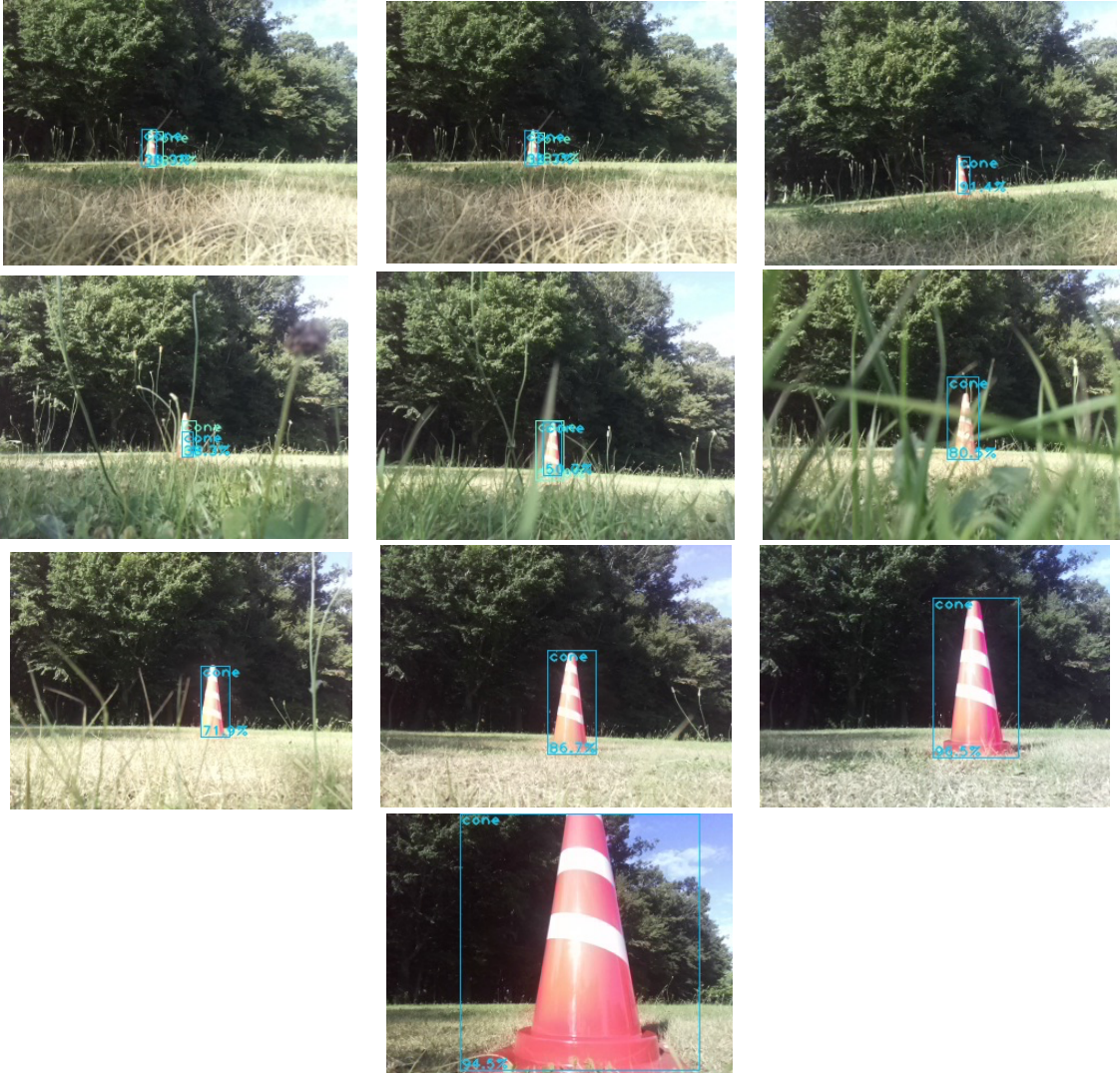
実験回数	0m ゴール	実際の撮影写真 & 詳細	動画 URL
1 回目	○	ゴール発見まで 2 回旋回 	https://youtu.be/ZBeBAcegJ2g
		発見後 11 回直進 & 旋回しゴール判定 	
2 回目	○	ゴール発見まで 2 回旋回	



発見後, 11 回の旋回&直進でゴール判定



https://youtu.be/MLe-Vh_op34

3 回目	○	発見後、10 回の旋回 & 直進でゴール判定 	https://youtu.be/y-N-4gq0k7c
------	---	---	---

また、画像と制御ログをまとめた URL を下記に示す。制御ログでは Object Detection による認識結果の認識率 (Score) やゴールまでの 4-0 で示した距離の制御状態 (distance) などがテキストログで確認できる。(制御状態 4: 最も長距離, 3: 長距離, 2: 短距離, 1: 最も短距離, 0: ゴール判定距離)

○ゴール検知試験画像ログ, 制御ログ:

https://drive.google.com/drive/folders/1jSap_Q-IAs683BYEWPg5Tb6G30LakVej?usp=sharing

● 考察

ミッションの Deep Learning での 0m ゴールが問題なくできることを確認した。

v15. 走行状態推定試験

● 目的

ミッションである走行状態の推定ができることを確認する。

● 試験内容

平坦な地面と草が大きい地面を交互に走行させ、その走行状態からハンドルパラメータが変化しているかどうか、走行状態を目視にて確認後、実験後ログにて適した変化が行われているか確認した。

● 結果

【走行振動状態の FFT データ】

ミッションの走行状態を推定し、その状態に適した制御をするため、走行時加速度データそれぞれ x, y, z の数値から FFT 画像データを生成し、その画像データを機械学習することで走行状態を推定しその走行状態に適した制御をする。

ACTS2020 の地面を走行している状態とスタックしている状態とで FFT データを生成した。能代宇宙イベント地面走行時 10s の FFT データを Fig. 15-1 に、スタックしている状態 10s の FFT データを Fig. 15-2 に示す。それぞれ走行状態は色の違いが表れている。

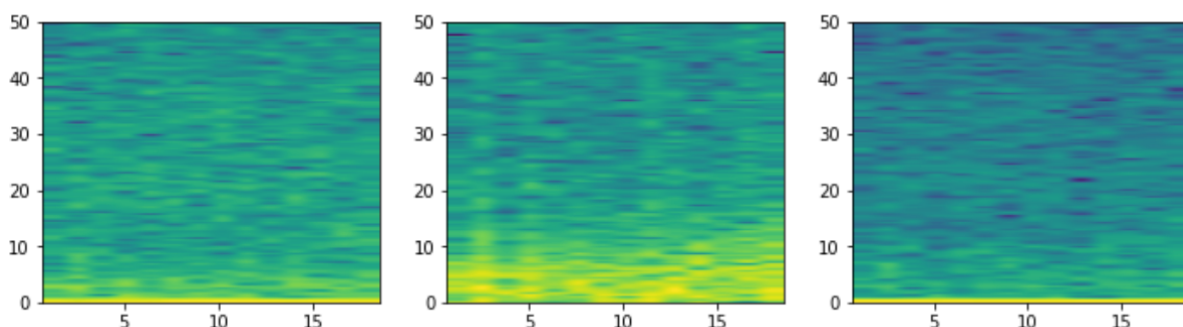


Fig. 15-1 ACTS2020 にて GPS で走行している状態の加速度 x, y, z の FFT データ (10s)

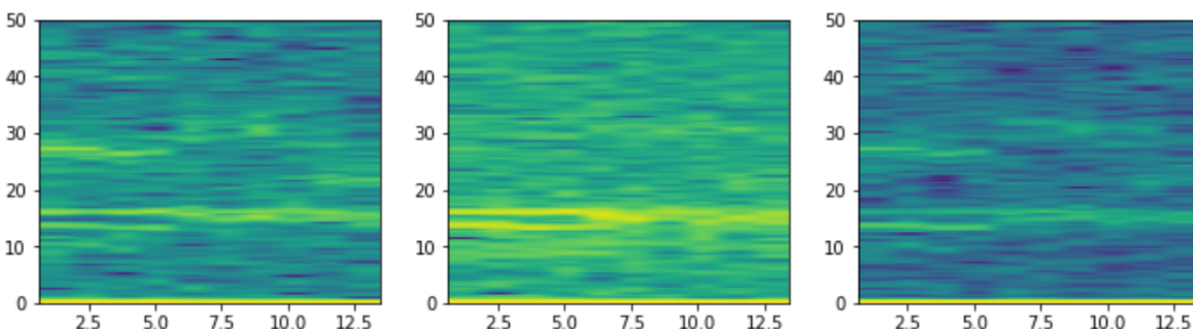


Fig. 15-2 ACTS2020 にてスタックしている状態の加速度 x, y, z の FFT データ (10s)

また本番で実際に使用するリアルタイムでの FFT 画像生成のシステムが完成したため、詳細を記述する。ACTS2021 投下 2 回目の加速度データ x, y, z の大きさ $\sqrt{x^2 + y^2 + z^2}$ に

てFFTのシュミュレーションを行った。競技の流れをそれぞれ気球で上げられている時、落下した時、パラシュート離脱動作をした時、走行時、ゴール判定時で加速度FFTの違いが大きく画像に反映されていることが確認できた。

Fig. 15-3 に気球上昇時のFFT画像である。この画像は縦軸 0—50Hz の加速度周波数成分の大きさで、横軸は秒数である。サンプリングは約 100Hz であり、下枠の一番左には時間[1/100s]があり、その右には年月日時刻、その右にはゴールまでの距離と 1s 間の移動距離[m]、その右には制御シーケンスが記載されている。

FFT 画像は周波数成分の強弱を色で表現しており、HSV の H の成分で青 0、赤最大として周波数成分の大きさで表示されている。例えば CanSat が完全に停止している時は青色で、着地衝撃など大きい加速度の時は赤色になる。その中間色は緑色で走行時などは緑色になる。

下の文字情報は 0s 地点の情報である。また秒数が 0-19 までであるが、横軸の数字は 0s を基準としてそれ以降の時間のFFTの結果を表している。例えば横軸 5s は 5s 後FFTの結果を表示している。

気球上昇時の場合は約 10Hz 程度が緑色で表示されていることがわかる。

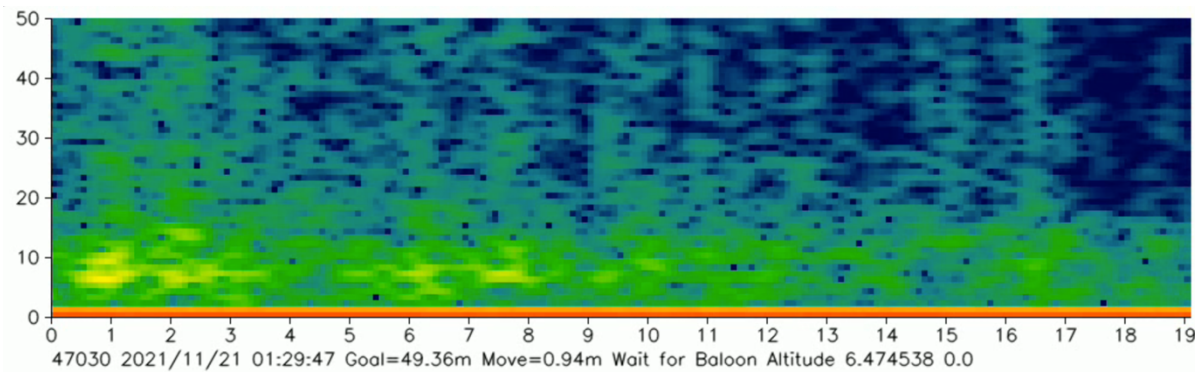


Fig. 15-3 ①気球上昇時_Wait for Baloon Alititude_動画内 8s

また、着地衝撃時のFFT画像をFig. 15-4に示す。横軸 12s 時に約 0-40Hz 程度まで赤色が表示されており、その後着地し静止しているため青色になっていることがわかる。

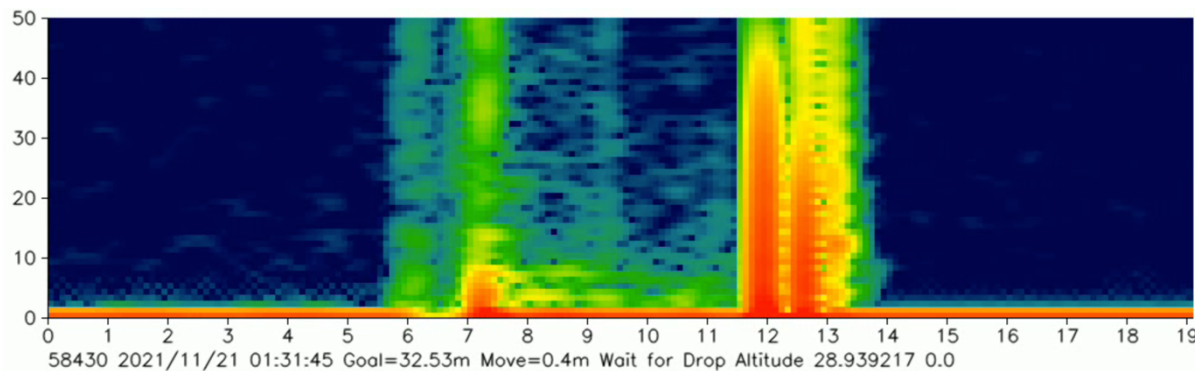


Fig. 15-4 ②着地衝撃時_Wait for Drop Altitude_動画内 56s

Fig. 15-5 にパラシュート分離時の FFT 画像を示す。着地後静止している状態からパラシュートケースと分離しわずかに CanSat に動きがあったことが、画像中の 9s 地点で確認できる。

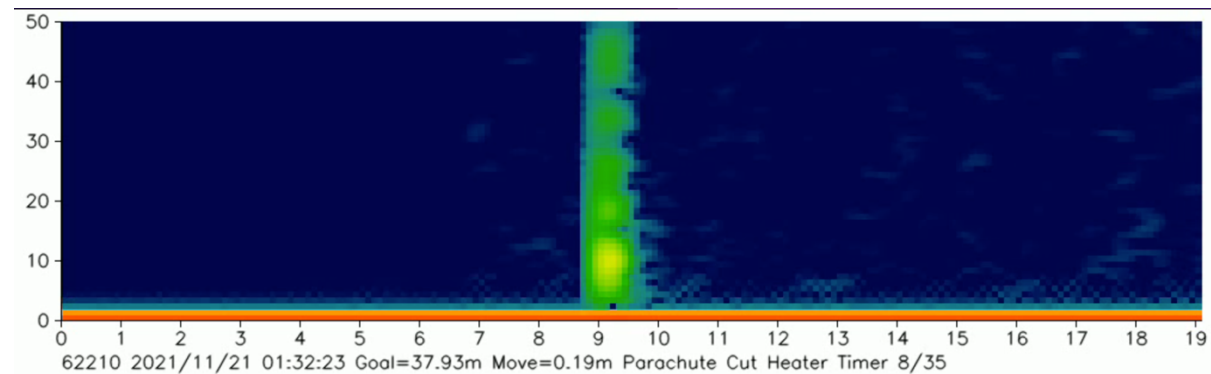


Fig. 15-5 ③パラシュート分離動作時_Parachute Cut Heater Timer_動画内 1m12s

Fig. 15-6 に走行開始時の FFT 画像を示す。パラシュート分離後静止状態から画像中 5s 地点で走行開始し、0-50Hz が全体的に緑色になっていることが確認できる。

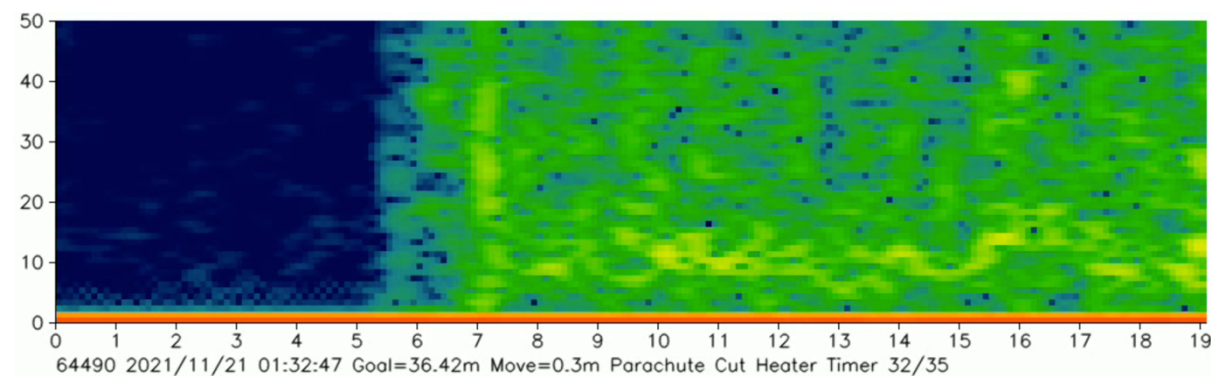


Fig. 15-6 ④走行開始時_Rover Drive Start_動画内 1m21s

Fig. 15-7 ゴール判定時の FFT 画像を示す。走行終了しゴール判定で CanSat が静止していることが横軸 4s 地点から確認できる。

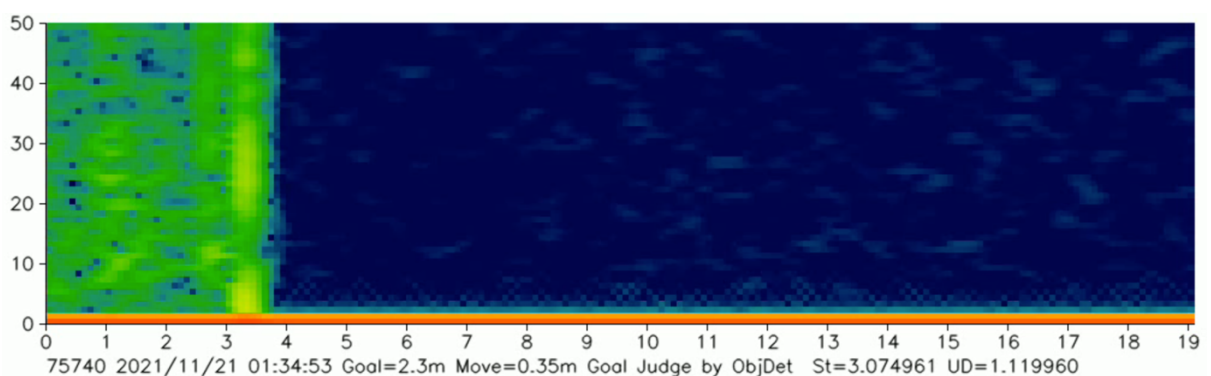


Fig. 15-7 ⑤ゴール判定_Goal Judge by Object_動画内 2m8s

このように各制御シーケンス時において全く違う画像になっていることが確認できた。走行時においてデータを収集し、Deep Learning の画像認識として学習させるこ

とで走行推定を行う.

また ACTS2021 でのリアルタイム FFT 画像を動画にまとめた.

○ACTS2021FFT 動画 URL : <https://youtu.be/M2HXDaGmhr4>

【走行試験】

FFT 画像での走行状態は 5 パターンに分類される。5 つの分類を下記の①－⑤に示す。

- ⑥ CanSat が完全に停止している状態の FFT 画像が真っ青の状態を halt と判定する。
- ⑦ CanSat が走行開始時に FFT 画像が真っ青から走行状態の FFT 画像へと変化する時, start of running と判定する。
- ⑧ CanSat が固く平坦な地面を走行しているとき, hard と判定する。
- ⑨ CanSat が草が生えており凸凹な地面を走行しているとき, grass と判定する。
- ⑩ CanSat が走行停止時に走行状態の FFT 画像から停止時の真っ青へと変化する時, stop running と判定する。

実際の FFT 画像の走行状態の例を Fig. 15-8 に示す。

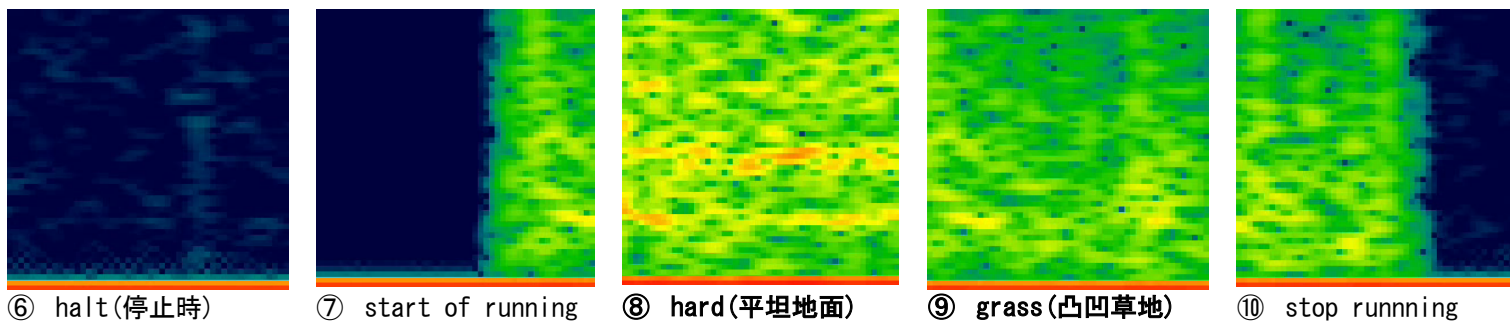
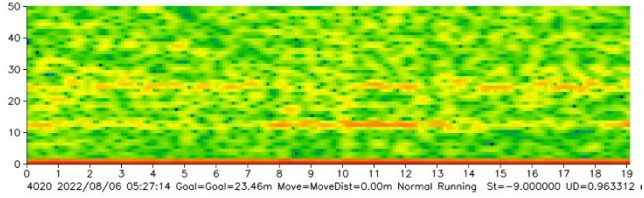
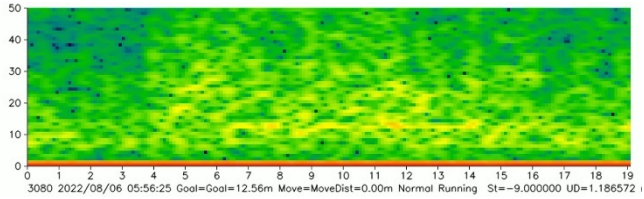
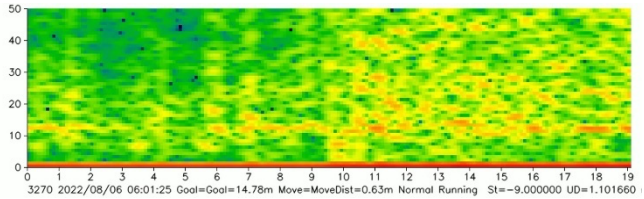


Fig. 15-8 各状態における振動の FFT 画像の例

走行状態の FFT 画像の分類により、実際に平坦な地面で走行時 CanSat が「hard」と判定するか、また凸凹の草地で「grass」と判定ができるか、走行試験を行った。また、停止時や走行開始、走行停止時における「halt」「start of running」「stop running」と判定されるか走行試験で確認を行った。

走行状態の判定結果を Table 15 示す。

Table 15 走行状態推定試験結果

回数	判定	地面の状態	結果詳細	動画 URL
1 回目	○	平坦な地面 (hard) 	平坦な地面 (hard) を判定した (実際の FFT 画像)  4020 2022/08/06 05:27:14 Goal=Goal=23.46m Move=MoveDist=0.00m Normal Running St=-9.000000 UD=0.963312	① 走行試験動画 https://youtu.be/ZIBfezloHBs ① FFT 画像 https://youtu.be/H1FxHRHDZrE
2 回目	○	草地の地面 (grass) 	草地の地面 (grass) を判定した (実際の FFT 画像)  3080 2022/08/06 05:56:25 Goal=Goal=12.56m Move=MoveDist=0.00m Normal Running St=-9.000000 UD=1.186572	② 走行試験動画 https://youtu.be/Fgpxu5hlvo8 ② FFT 画像 https://youtu.be/YKXuVpJRw2U
3 回目	○	草地の地面 (grass) & 平坦な地面 (hard)  	草地の地面 (grass) & 平坦な地面 (hard) を 判定した (実際の FFT 画像) 図中の横軸中央 9-10 で grass→hard 切り替わり  3270 2022/08/06 06:01:25 Goal=Goal=14.78m Move=MoveDist=0.63m Normal Running St=-9.000000 UD=1.101660	③ 走行試験動画 https://youtu.be/_D17GAHibtQ ③ FFT 画像 https://youtu.be/ESJMHUW_GvE

● 考察

FFT データを生成しその画像の違いから、FFT データの機械学習により走行状態を推定できることを確認した。

第7章 ガントチャート（工程管理）

全体的に祝日と土日を休みとして扱い、本来かかる時間の倍程度の日程で余裕を持ったスケジュールになっている。メンバーが複数人いないために担当ごとの項目と、大学院のため試験のスケジュールは無い。5月に新機能考案とプロトタイプ作成&テスト、6月にミッション&End to End 関係試験以外を行い、7月にEnd to End やミッション系の審査試験、8月にNSE 参加とリカバリー、9月にARLISS 準備及びリカバリーを行う。ガントチャートをFig.7に示す。

【5月】

5月はCanSatの新ミッション機能の考案、プロトタイプを製作し、テストを行い改良していく

第1-2週はCanSatへの搭載新機能の考案や事前調査を行う

第3-4週は搭載部品や材料の受注を行い、ミッション機能の開発を始める。

第5週からCanSatのプロトタイプモデルを製作し、テストを行い改良重ねる。

【6月】

6月はCanSat試作段階の試験を終了し、ミッション&End to End 関係試験以外を終了する

第1-2週はCanSatのプロトタイプモデルを製作し、テストを行い改良重ねる。

第3週は質量試験・キャリア収納試験、着地衝撃試験を行う（各2試験1週間）

第4週はパラシュート投下試験、開傘衝撃試験を行う

第5週は準静的荷重試験、分離衝撃試験を行う（NSE 予備審査提出）

【7月】

7月は全ての試験を終了し、本審査の提出を行う。（各2試験1週間）

第1週は分離衝撃試験の続きを行う。

第2週はミッションのゴール検知試験、走行性能確認試験。（ARLISS 予備審査書提出）

第3週は振動試験、End to End、制御履歴試験を行う。

第4週はNSE本番機体・回路作成を行う。（NSE本審査提出）

【8月】

8月はEndtoEndや本番への最終調整し、予定に余裕を持ち全体的なリカバリー可能にする

第1週NSE最終調整と準備を行う。（ARLISS本審査書提出）

第2-3週に能代宇宙イベントに参加しEnd to Endを行う。

第4週でNSE試験の結果を踏まえ、本番用のCanSatとスペアを製作する。

第5週にARLISSに向け試験のリカバリー及び最終調整や準備を行う。

【9月】

9月はARLISSへ向け最終調整や準備を行う

第1-2週はARLISSへ向け最終調整や準備を行う。

第3週よりARLISSへ参加する。

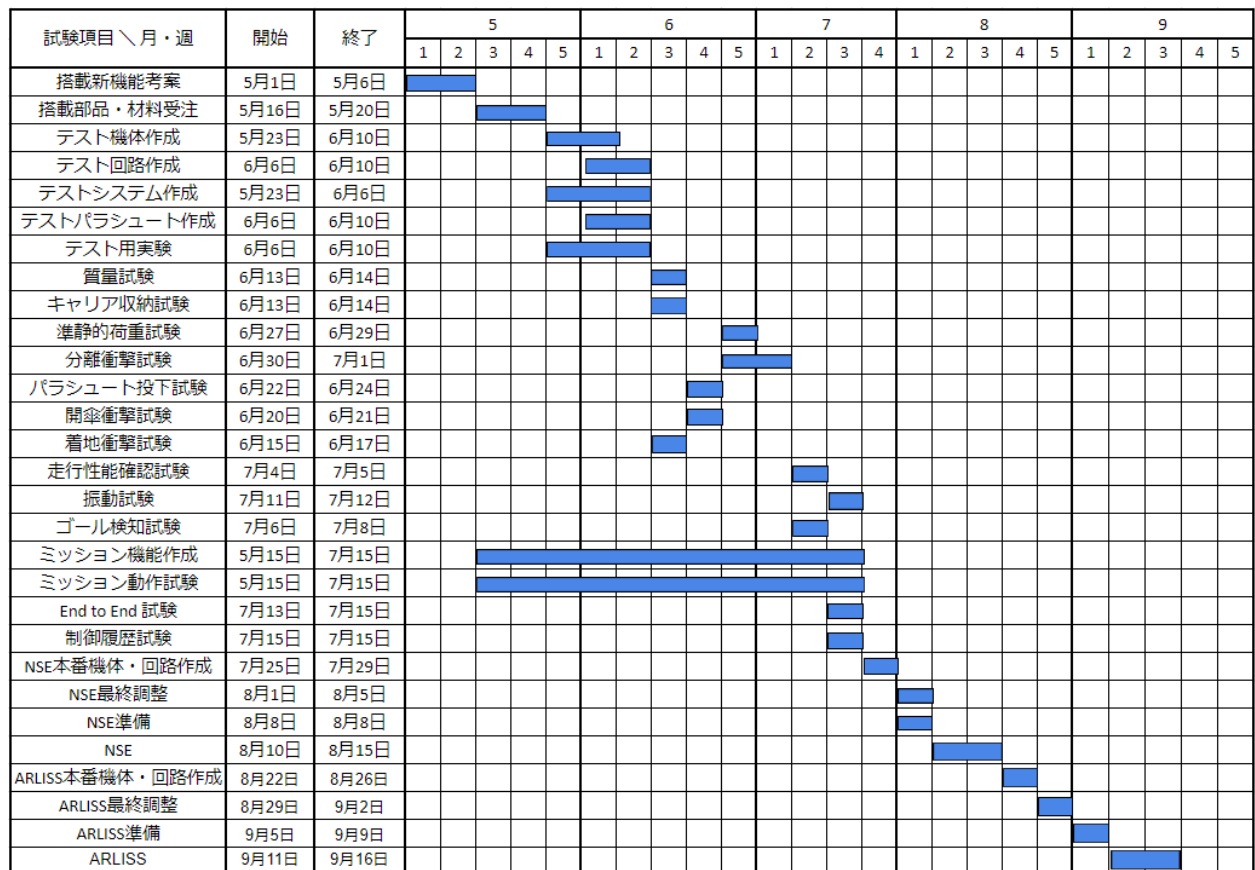


Fig. 7 ガントチャート

また、ガントチャートのスプレッドシートを以下に示す。

ガントチャートスプレッドシート

https://docs.google.com/spreadsheets/d/1WILs07sXlhtfopsxtKhZwL3XZM_a291au392DwweIrU/edit?usp=sharing

第8章 責任教員による自己安全審査結果のまとめ

(この章は必ず責任教員が記入してください)

・カムバックコンペティションに参加しない場合は、M3 と M4 は削除してください。

安全基準審査

要求番号	自己審査項目	自己審査結果	責任教員コメント（特筆すべき事項があれば）
	ARLISS2022 安全基準		
S1	投下する機体の質量が基準を満たしている	☒	
S2	容積がキャリア基準を満たしている	☒	
S3	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ていないことが試験で確認できている	☒	
S4	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ていないことが試験で確認できている	☒	
S5	ロケット分離時（パラシュート開傘時）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ていないことが試験で確認できている	☒	
S6	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	☒	
S7	ロストへの対策を実施しており、有効性が試験で確認できている (対策例：位置情報送信、ビーコン、蛍光カラーペイント等)	☒	
S8	打ち上げ時の無線機の電源 OFF の規定を遵守でき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)	☒	

S9	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	☒	
S10	ロケットへの装填からミッションの開始、打ち上げ後の回収までを模擬した End-to-end 試験を実施できており、今後、大幅な設計変更はない	☒	
カムバックコンペティションに参加する場合は以下の項目を必ず満たしてください			
M3	ミッション時に人間が介入しない自律制御を実施することを確認出来ている	☒	
M4	ミッション後、規定された制御履歴レポートを運営及び審査員へ提出しログ・取得したデータについて説明が可能な状態である	☒	

責任教員所感

今回は振動の加速度データを FFT し、その画像データを生成して、それを Deep Learning 画像認識により CanSat の状態推定を行うという新しいミッションにチャレンジしています。現在は 5 種類の状態推定に成功しており、今後さらに状態を追加する予定であり、興味深い結果が得られております。参加許可をいただければ幸いです。

第9章 大会結果報告

(i) 目的

CanSat の走行時の振動の 3 軸加速度等を学習させ、平坦な地面・凸凹な地面を機械学習により推定し、さらに Object Detection を用いてリアルタイムでゴールを認識し 0m ゴールを目指す。

(ii) 結果

【1 回目打ち上げ結果】

→4.5km 落下地点からゴールまで走行し、20cm ゴール

Fig. 9-1 に 1 回目結果の写真を示す。



Fig. 9-1 1 回目結果

1 回目打ち上げでは、機体は打ち上げやロケット分離、落下衝撃に耐え、ゴールから 4.5km 地点に落下した。パラシュート離脱を行い、GPS 位置情報でゴール付近まで約 2 時間半かけ走行した。走行中 ARLISS の地面は平坦な部分と土のさらさらとした柔らかい部分と様々存在し、ミッションの走行路面判定では各地点で FFT 画像による走行状態の推定に成功した。1 回目走行軌跡を Fig. 9-2 に示す。

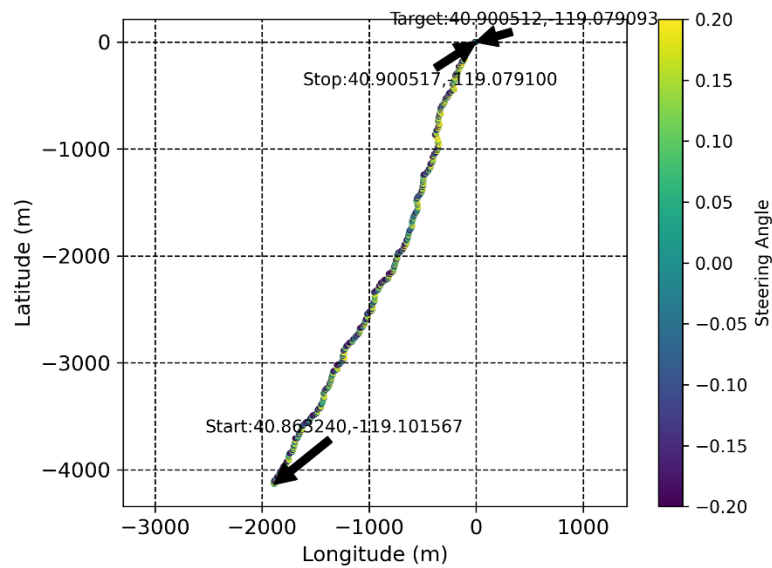


Fig. 9-2 1 回目走行軌跡

ゴール付近にて Object Detection による 0m ゴール誘導制御を行い、20cm でのゴールで停止した。実験では判定の画像の大きさを 3 回で停止する制御だったが、ゴールに衝突し横転することが多かったことから、打ち上げ直前で 2 回の制御へと変更したことが原因で、目標より 20cm 手前での停止となった。実際の Object Detection での GIF 画像ログを Fig. 9-3 に示す。



Fig. 9-3 Object Detection での画像ログ

また、モータ用バッテリーは容量の半分以上を消費し、コンピュータ用がほぼ空の消費量であったことから、コンピュータ用電源の容量を今後拡充する必要がある。

【2 回目打ち上げ結果】

→ロケット衝撃で機体に変形，片輪が動かず走行不能

Fig. 9-4 に 2 回目結果の写真を示す．



Fig. 9-4 2 回目打ち上げ結果

2 回目打ち上げでは，ロケットから分離後の落下中の通信で IM920 の電波からの通信が文字化けを起こした．着地した CanSat はパラシュートとの分離動作後，片輪が動かず回転しており，動作不能となっていた．

競技後に確認したところ，片方のモータ取付部は自由落下をしたときと同じように変形しており，大きな衝撃でモータが故障していることがわかった．IM920 の文字化けやパラシュートの焦げ付き，CanSat 放出高度が 2000m 地点であったことから何かしらのロケットの衝撃で故障した可能性がある．今後は規定の衝撃よりも機体を頑丈に設計する必要がある．実際のパラシュートの焦げ付きの写真を Fig. 9-5 に示す．



Fig. 9-5 パラシュートの焦げ付き

【3回目打ち上げ結果】

→3.5km 落下地点からゴールまで走行し、カメラが機能せずゴール判定せず

3 回目打ち上げでは記録外の打ち上げであることから、動画撮影機能を追加し打ち上げを行った。しかしこの動画撮影機能において、最終的にカメラが機能せずゴール不可能となった。

CanSat はロケットから分離後にゴールから 3.5km 地点へ着地し約 2 時間かけ GPS 位置情報による走行制御でゴール数 m 付近まで走行した。動画撮影を行ったことで、プログラムのバグによりカメラから画像が撮影できなくなり、ゴールすることができなくなった。

Fig. 9-6 に走行軌跡を示す。

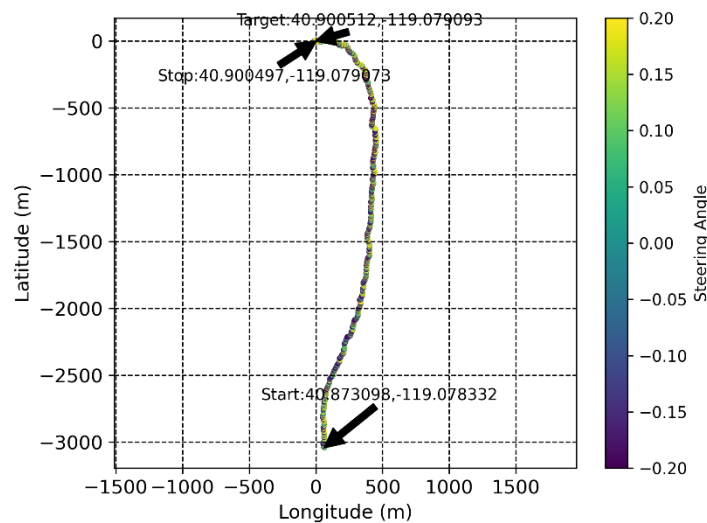


Fig. 9-6 3 回目走行軌跡

今回 Raspberry pi Zero 2 を利用したが、従来 Raspberry Pi Zero での動作を考慮しシステムでは Object Detection と動画撮影の並列処理を行わない設定としていた。Object Detection 動作時には負荷が大きいため動画撮影をしない設定としていた。かつ動画撮影のフラグが立っている状態では Object Detection をしない設定としていた。その相互で動画撮影フラグが立つと動画撮影及び Object Detection の処理が行われない制御となった。ゴール付近で写真撮影ができずゴール付近を周回している写真を Fig. 9-7 に示す。



Fig. 9-7 3 回目結果

また3回目において、ミッションの走行判定以外にも、ロケットの衝撃や着地時の衝撃をFFT画像でCanSatの状態判定が可能か実験を行った。FFT画像で①-⑧の種類の衝撃を検知し、Deep Learningの画像分類に成功した。実際のFFT画像による判定結果のGIF画像をFig. 9-8に示す。

- ① Rocket Launch(ロケット打ち上げ開始)
- ② Rocket Rising(ロケット打ち上げ上昇時)
- ③ Release from Rocket(ロケット分離衝撃)
- ④ Descending with Parachute(パラシュート開傘衝撃)
- ⑤ Landing(着地衝撃)
- ⑥ Open Cover (Parachute separation) (パラシュート分離)
- ⑦ Start of Running(走行開始)
- ⑧ Running ARLISS (ARLISS 走行地面)

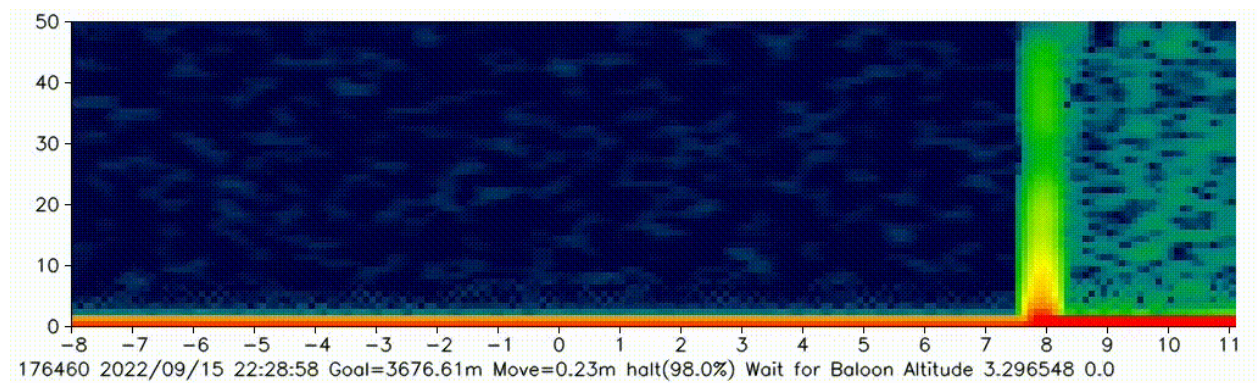


Fig. 9-8 打ち上げ3回目 ARLISS 競技全体の衝撃 FFT 画像による Deep Learning 分類結果

(iii) 考察

【サクセスクライテリア】

ミニмумサクセスは打ち上げ1回目と3回目に10m以上走行しゴールまで走行したことから、達成することができた。

ミドルサクセスは打ち上げ1回目でObject Detectionを開始後0mまで誘導制御を行ったが、手前の20cm地点で停止したため、完全な達成ではないが、部分的な達成となった。

フルサクセスは1回目と3回目地面の状態判定を行うことに成功し、3回目ではロケット衝撃の判定を行うことに成功した。しかし1回目と3回目では0mゴールを達成していないことから、部分的な達成となった。

アドバンスドサクセスはフルサクセスにスタック判定を追加した内容だが、今年度ARLISSでは平坦な地面が続きあまりスタックになる轍が存在しなかったことから、部分的な達成となった。

ミニмум サクセス	CanSat が放出落下後に破損なく着地し、パラシュートから分離し、GPS 位置情報からモータ回転数制御により10m以上走行する。	
ミドル サクセス	CanSat がゴール付近に到達後、Object Detectionで0mゴールする	 <u>0.2m</u>
フル サクセス	CanSat が地面を走行時、平坦な地面、凸凹な地面を判定し、その状態に合わせた速度やハンドルの走行制御を行い、Object Detectionで0mゴールする。	 <u>0.2m</u>
アドバンスド サクセス	CanSat が地面を走行時、平坦な地面、凸凹な地面、 スタック状態 を判定し、その状態に合わせた走行制御を行い、Object Detectionで0mゴールする。	 <u>0.2m</u>

【原因解析】

① 打ち上げ1回目 20 cm地点での停止

1 回目では 0m ゴールする手前の 20cm 地点で CanSat が停止した。事前調整では、CanSat はゴールに衝突時に横転し、0m ゴールができなかったことがあったため、打ち上げ直前でゴール判定条件を Object Detection によりゴール距離のゴールの大きさ 3 回で停止する制御から、2 回で停止するよう設定を変更した。また、タイヤの設計上は横転しても元の状態へ戻るようにタイヤ幅を狭めた結果ゴールへ衝突時に横転しやすい仕様であった。また、4.5km からの走行により、モータ用バッテリーが大きく減少し、従来遅いスピードで確実に誘導する速度に設定したことが原因で、遅い速度でパワーが足りず従来の走行距離とならず手前で停止した。

直前でのゴール判定設定変更と横転しやすいタイヤの設計、モータ用バッテリーの低下による走行距離の変化により、0m ゴールの手前での停止となった。

今後はゴール判定の制御を全面的に見直しより確実に 0m で停止できる仕様にするこ
と、横転しやすいタイヤの設計を見直すこと、モータ用バッテリーを考慮し走行速度を
遅くしすぎないことを改善していく必要がある。

② 打ち上げ2回目 ロケット衝撃による機体故障

2 回目では機体のモータ接続部が故障したが、パラシュートの焦げ付きや落下時の無線機 IM920 の文字化けから、想定よりロケット衝撃が大きかった可能性がある。想定より衝撃が大きいことを考慮し、推奨されている試験値の衝撃より大きい衝撃にも耐えられる機体を設計することが必要である。

③ 打ち上げ3回目 動画撮影によるカメラ動作不良

3 回目では動画撮影をシステムに追加した結果、カメラの動作不良が生じ、Object Detection ができなくなりゴールができなくなった。本番で組み込む設定は事前に試験しておく必要があり、直前での仕様変更は極力しないことが重要である。また、来年度からはデータ取り用に行う仕様変更を事前に列挙しておき、確実に動作できるか試験を行うことが必要である。

④ マネジメント面

マネジメント面ではスケジュールにおいて余裕を持つ必要があった。スケジュールではガントチャート作成後に複数の予定が入り、予定通りのスケジュールでの開発が困難となった。今後他の予定が追加されても余裕をもって開発できるようより初動を早め、リカバリー期間を多く設ける必要がある。

また今回半導体不足で部品入手が困難となったことから、早めの部品調達や入手しやすい部品での設計を考慮する必要がある。

また旅程の日程ではスケジュールが過密となったため、前日到着など日程で参加することが必要である。

第10章 まとめ

(i) エフ・努力した点（ハード、ソフト、マネジメント面すべて）

【ハード】

機体では主にシャーシとタイヤ、スタビライザーに改良点を加えた。シャーシは従来 3D プリンタの PETG 樹脂で作成したが、実験中に衝撃を伴うと破損したことから、ポリカーボネート 3 mm 板を用いて、モータの取り付け部であるシャーシを作成した。新たにモデリングマシンの monoFab SRM-20 導入し、切削により政策を行った。

タイヤの構造は従来の円弧型から角度を垂直にすることで、砂漠の硬い砂地への走行でグリップ効き、轍へのスタック時に砂地の角にグリップ部が食い込み離脱しやすい構造とした。また横転時に横向きのまま機体が回転し走行不能になることを防ぐため、タイヤ側面に角度をつけた設計にし、中心部が平坦にならないよう横転防止部品をホイールに取り付けた。

またスタビライザーではポリカーボネート板でばねを利用しキャリア格納時に折りたたむ構造だが、収納時ある程度の長さでは収納することができないため、ブロック部品を並べ走行時のみ直線状になる構造にし、キャリア格納時には折りたたみ格納可能となるように設計した。

実際の CanSat 機体を Fig. 10-1 に示す。



Fig. 10-1 CanSat 機体

【ソフト】

新規性のあるミッションとして、CanSat の振動による状態判定を行った。加速度データを FFT 画像にし、Deep Learning 画像分類で走行状態を判定させた。走行地面には硬く平坦な地面と草地のような凸凹で柔らかい地面の 2 パターンを分類させたが、どのような地面で走行状態の特徴が出るか色変化などを様々照らし合わせ試行錯誤を行い、この 2 パターンの走行状態の違いを発見した。FFT 画像の周波数成分ごとに色変化を行う画像では、色変化方法やサンプル数をどの程度の範囲で行うかで FFT 画像は大きく変化するが、最終的に Deep Learning により高い精度での画像分類が行える範囲での走行路面判定に成功した。また走行速度やタイヤ形状により振動が大きく変化するので、様々なパターンでの学習を行い、精度を高めるようにした。

また加速度データやジャイロデータ, x, y, z 方向をそれぞれ 3 つの FFT 画像か x, y, z 方向を 1 つの大きさとして FFT 画像を作るかで, どの方法での FFT 画像による画像分類の精度が高く出るか様々な条件でテストを行った。

【マネジメント】

今年度は団体では神奈川県と千葉県で開発や実験を行き来し行ったが, 実験日や場所に合わせ部品や機体の用意などを念入りに計画し行った。必要になる時を逆算し, 早めに開発や実験を行うようにした。また渡米の入出国ではコロナ禍であるため, 水際対策について念入りに調査を行った。

また今年度はガントチャート以外に様々な予定が追加で急に入ることが多かったが, できるだけ予定よりも早めに必要な作業を終了させるように終了日を設定し実行した。

(ii) 課題点

① ゴール判定制御改善

ARLISS 打ち上げ 1 回目にて 0m ゴールでなく 20 cm地点で停止した制御について全面的に見直し, ゴール判定のメカニズムを新しく設計する。

② 機体設計強化

ARLISS 打ち上げ 2 回目にて機体が故障したため, より衝撃に耐えられる構造に設計を行う。

③ カメラ動作不良改善

ARLISS 打ち上げ 3 回目にてカメラ動作不良にて, ゴール判定が行えなかったことから, まずはプログラムを修正し, 他にバグなどがないか見直しを行う。また動画撮影など, 本番で行う可能性のある動作は全て試験を行い, 本番で設定を変更することの無いようにする。

④ スケジュールに余裕を持つ

予定を立てても追加で他の予定が入ることを考慮し, 可能か限り早い段階で予定を終了できるようにスケジュールリングを行う。

(iii) 今後の展望

(ii)課題点の項目について改善を行う。ミッションの CanSat の振動（加速度データ）から FFT 画像による走行地面の画像分類では, 走行地面判定とロケット衝撃などの振動においても状態の判定ができたことから, パラシュート分離動作などをはじめとした全体的な FFT 画像判定を制御に組み込み, 確実な CanSat 制御を行えるかミッションを行う。