

ACTS 報告書

提出日：2021 年 12 月 26 日

- チーム情報

CanSat チーム名	湘南工科大学 SuperNOVA
CanSat チーム 代表者	秋山実穂 Supernova.aut@gmail.com, 090-8132-5165
UNISEC 団体名	湘南工科大学 斎藤研究室
UNISEC 団体 学生代表	秋山実穂
責任教員	斎藤卓也 Supernova.aut@gmail.com, 080-6625-5882

- メンバー

役割	名前 (学年)
PM,機体,回路,システム	秋山実穂

- CanSat の製作目的・大会参加理由
→研究活動, 学習のため

目次

第1章	ミッションについて	3
1.	ミッションステートメント（ミッションの意義と目的）	3
2.	ミッションシーケンス	4
第2章	サクセスクライテリア	5
第3章	要求項目の設定	5
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）	5
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）	6
第4章	システム仕様	7
1.	CanSat 外観	7
2.	CanSat 内観・機構	7
3.	システム図（CanSat 搭載計器仕様一覧）	10
4.	アルゴリズム	11
第5章	試験項目設定	12
第6章	実施試験内容	13
1.	システム要求を満たすための試験内容	13
第7章	工程管理、ガントチャート（スプレッドシートを推奨）	57
第8章	各担当（ハード・ソフト・全体などの進行状況を記入）	57
第9章	大会結果報告	59
1.	目的	59
2.	結果	59
3.	考察	63
第10章	まとめ	67
1.	工夫・努力した点（ハード、ソフト、マネジメント面すべて）	67
2.	課題点	70
3.	今後の展望	70

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）

ミッション内容

機械学習の画像認識の中で複数の物体とその距離を検知ができる **Object Detection** を利用し、 **Object Detection** によるゴール画像認識で 0m ゴールする。

【ミッション背景】

我々は例年 Deep Learning の Image Classification を用いた 0m ゴールをミッションとしていた。Image Classification による手法では、撮影した画像の分類はできても画像中どの位置にあるかは特定ができない。そこで、我々は画像全体を ROI で横方向に 3 分割、5 分割、それぞれ Image Classification させることにより、左右どこにあるかを判定して 0m ゴールの制御をしていた。3 分割でゴールを発見している例を Fig.1-1 に示す。



Fig.1-1 左右を判定するための 3 分割の画像分類の例

さらに昨年時のミッションでは、5 分割では 6~7m 程度までしかゴールを認識することができなかったため、さらに 7 分割を加えることにより、ゴールから 10m の距離からでも認識できるように改良を加えた。さらに、7 分割のように ROI が小さくなると、上下方向にも ROI が小さくなってしまったため、CanSat 本体のチルト方向への傾きにより、ROI が上下方向でゴールコーンを捉えられなくなる問題が生じた。そこで我々は、この問題を解決するため、7 分割時に ROI をさらに縦方向に上下に移動して画像分類を行うことにした。これにより、1 分割→3 分割→5 分割×3→7 分割×3(上下中)と 1 枚の画像に合計 40 回もの画像分類を行い、CanSat が写真を撮って次に走行方向を決定するのに最長約 12 秒程度かかっていた。そのため ACTS2020 での 0m ゴールは実験時間 15 分ギリギリのゴールとなった。今後はより制御の速度を上げられる手法を検討する必要があった。

【Object Detection による 0m ゴール】

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。Image Classification 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 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、である。現在我々がテストしている環境では、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、実装可能と判断した。

2. 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する。
7. 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. 5m 以内になったら、カメラ画像を Object Detection により、ゴールを物体として検出する。(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. 2 に示す。

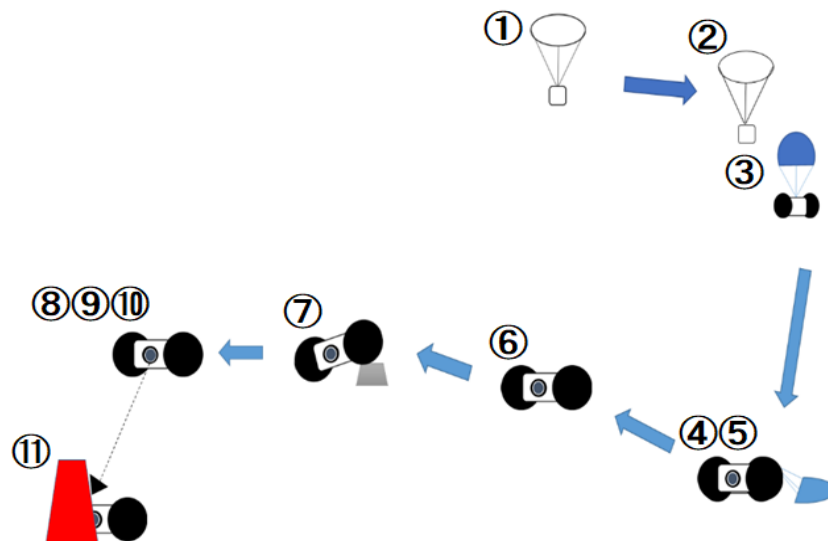


Fig. 2 ミッションシーケンス

第2章 サクセスクライテリア

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。
フルサクセス	ゴールから 5m 以内まで接近し、その後 Object Detection でゴールへ 0m まで近づきゴール判定をする。

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だろう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 分以内でできている

	<u>カムバックコンペティションルールの充足</u>
R11	ミッション時に人間が介在しない <u>自律制御</u> 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 <u>制御履歴レポートを運営者へ提出</u> 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（ <u>ミッション実現要求項目</u> ）
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	ゴール地点に設置されたコーンを検知し、10m以上GPS制御走行した後、Object Detection 画像認識で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。

第4章 システム仕様

1. CanSat 外観

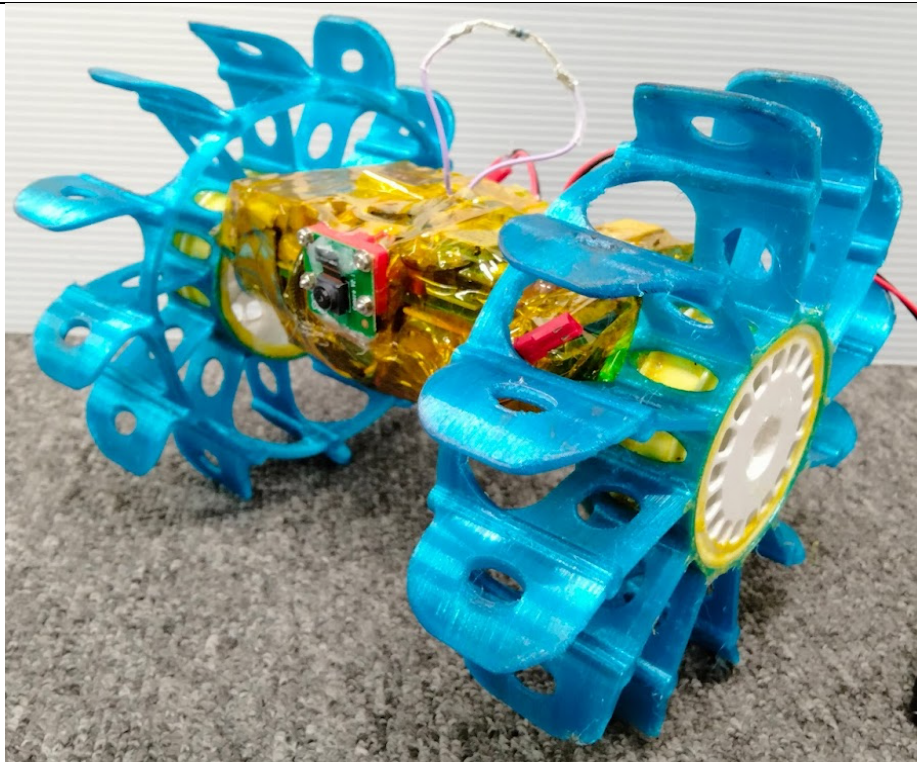


Fig.1 CanSat

2. CanSat 内観・機構

【機体】

機体は1枚の3Dプリンタで作成した板に左右モータと上下バッテリーを取り付け、モータの裏側は電子回路を搭載している。

3Dプリンタでの制作で利用したTPU樹脂は3Dプリンタで利用できる樹脂の中でもゴムのような柔らかさを持つ性質がある。また3Dプリンタでの設定が切削機を利用するよりも細かくカスタマイズできる素材であり、着地時に衝撃で少し曲がるような柔らかさの設計することにより、衝撃耐久し衝撃吸収を行う仕様となっている。またタイヤにも柔らかい素材のTPU樹脂を利用することで、より衝撃吸収ができる使用となっている。

CanSatを上から見た図を Fig.4-1, CanSatを下から見た図を Fig.4-2 に示す。

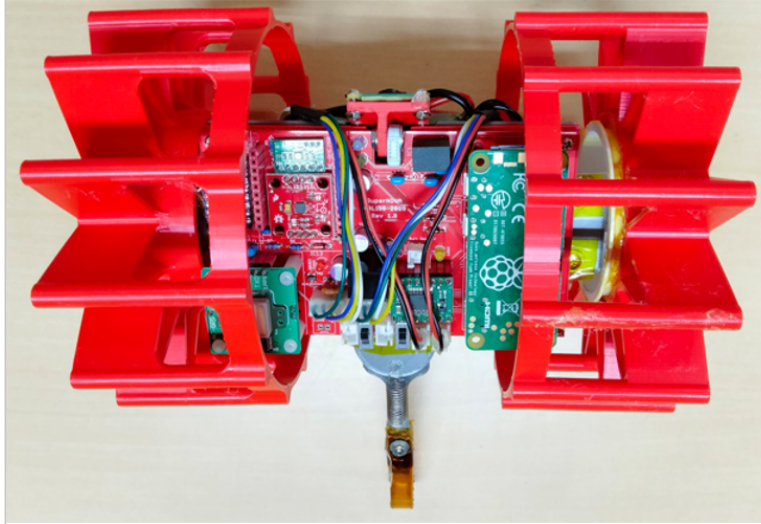


Fig.4-2-1 CanSat を上から見た図

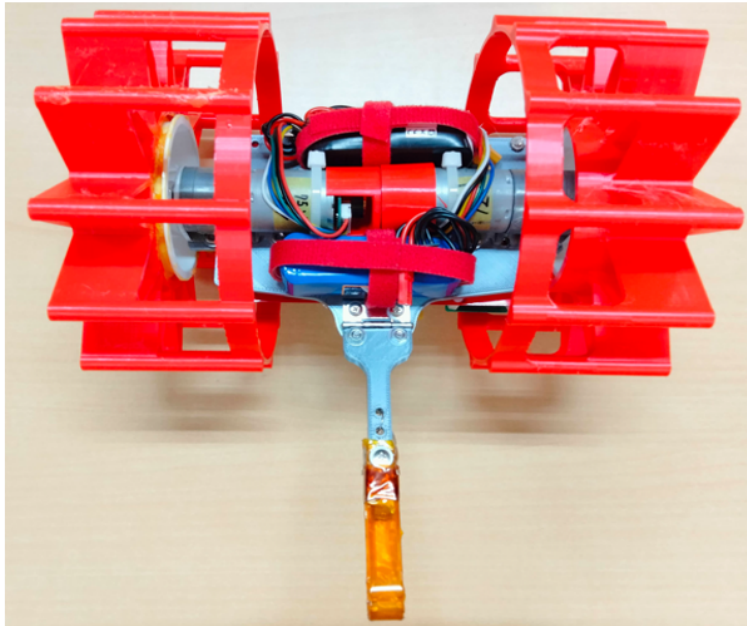


Fig.4-2-2 CanSat を下から見た図

【減速機構】

機体にパラシュートを付属した保護ケースを取り付ける。結束バンドを電熱線で焼き切ることによって展開する。パラシュート生地は頑丈なリップストップナイロンを利用し、紐には金属ワイヤーと金属メジャーを利用することで着地時に CanSat とパラシュートを遠ざけ絡まり防止ができる構造となっている。CanSat のパラシュート離脱機構と紐を直接繋げずに、保護ケースに紐を取り付けることにより、分離衝撃で離脱機構が故障することを防いだ。

パラシュートの保護ケース収納時写真を Fig.4-2-3 に示す。

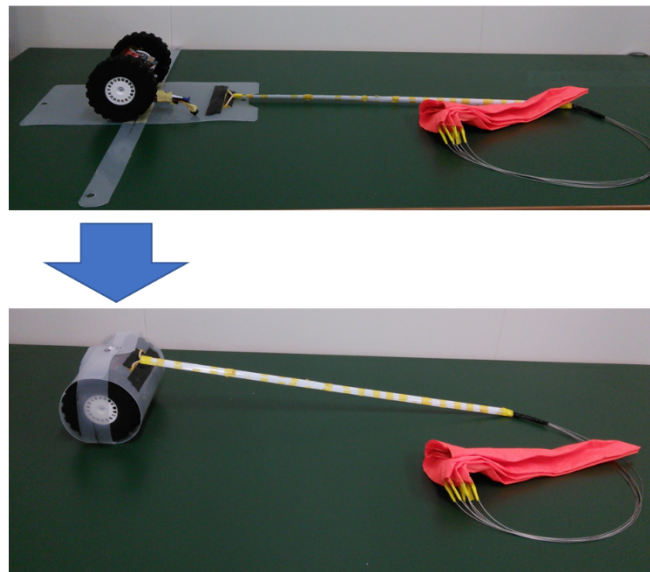


Fig.4-2-3 保護ケースに収納させたローバ

また、実際にパラシュートを展開した図を Fig.4-2-4 に示す。

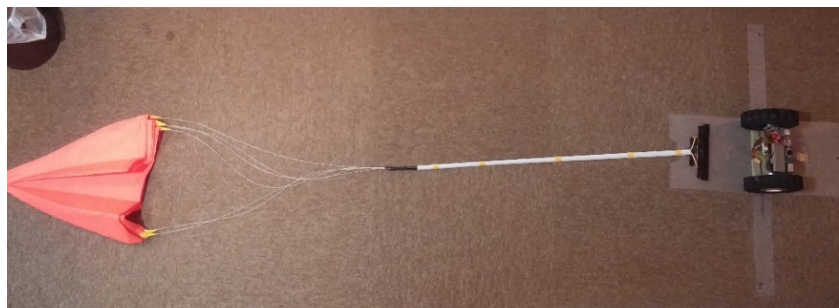


Fig.4-2-4 展開図

3. システム図 (CanSat 搭載計器仕様一覧)

【マイコン、モータ】

Raspberry Pi zero W をマイコンに利用し、Coral Accelator を Deep Learning の計算に利用し、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使っている。

【長距離通信無線】

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。

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.4-3-1 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 4-1 に示す。

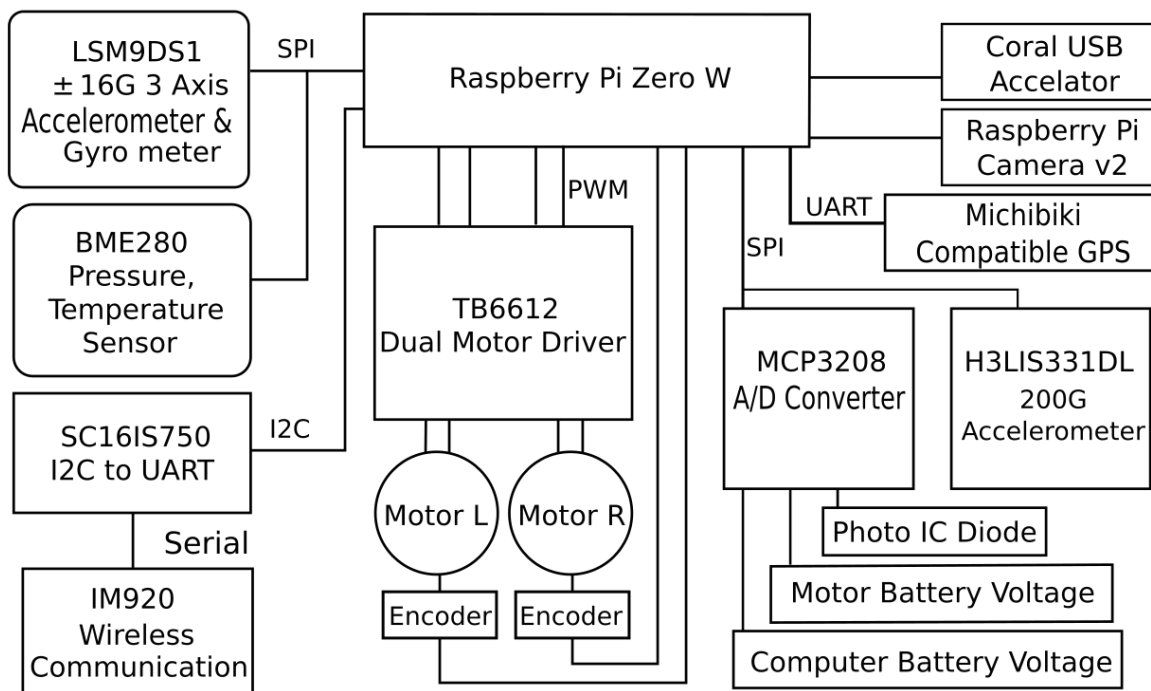


Fig.4-3-1 システム図

4. 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。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 Object Detection (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 に示す。

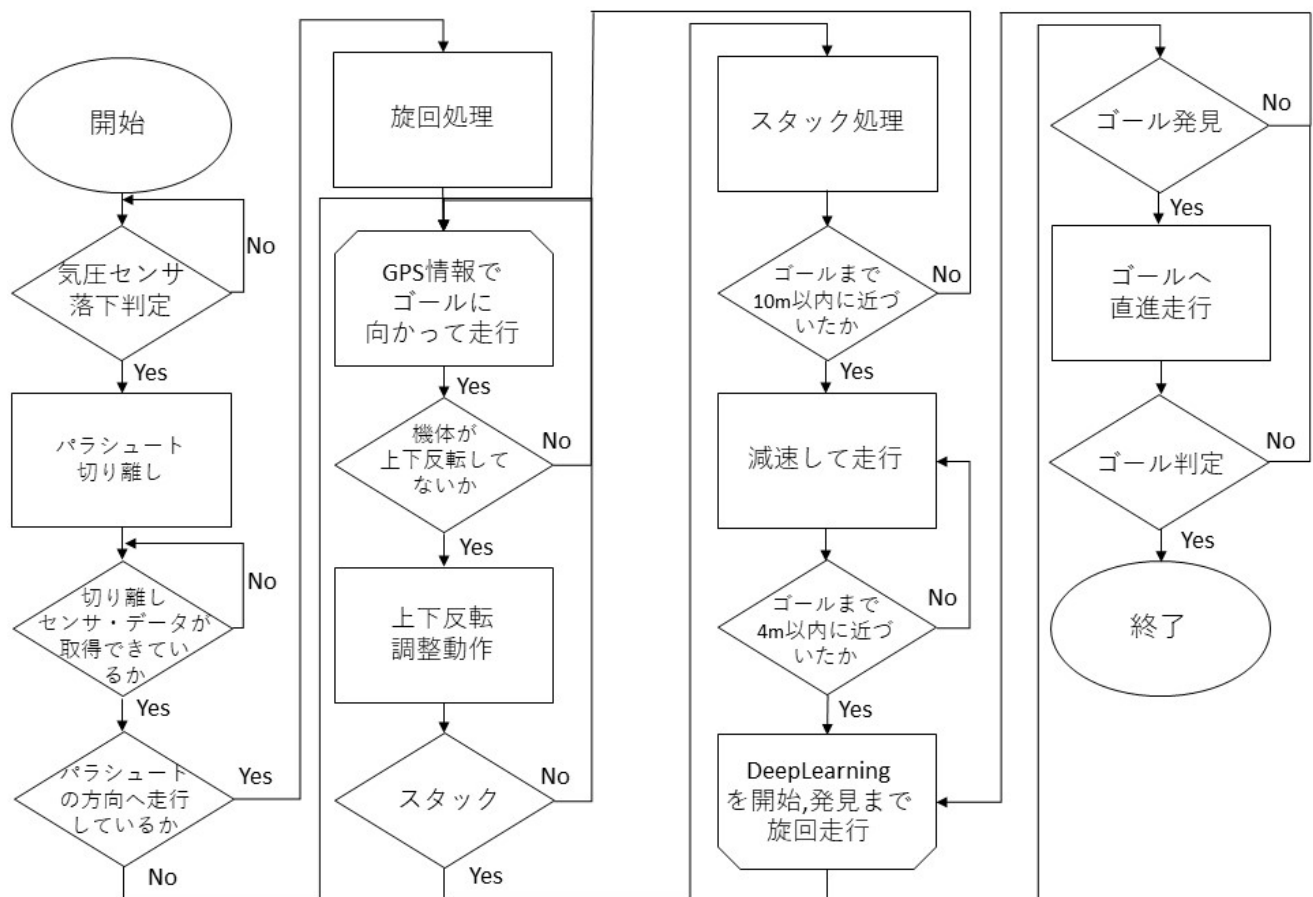


Fig. 4 フローチャート

第5章 試験項目設定

第3章の各要求項目に対して、基準を満たしているかどうかを確認する試験を設定してください

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 (予定日)
V1	質量試験	R1	7/9
V2	機体の収納・放出試験	R1, R10	7/9
V3	通信距離試験	R2	6/11
V4	落下試験	R3	7/12
V5	準静的荷重試験	R4	7/19
V6	振動試験	R5	8/18
V7	分離衝撃試験	R6	7/20
V8	通信機電源 OFF/ON 試験	R7	6/15
V9	通信周波数変更試験	R8	6/14
V10	End-to-end 試験	R9, R11	10/20
V11	制御履歴レポート作成試験	R12	10/21
V12	着地衝撃試験	M1	7/21
V13	ゴール検知試験	M2	10/20

第6章 実施試験内容

1. システム要求を満たすための試験内容

(V1) 質量試験

- 目的

CanSat の質量がレギュレーションの質量 1050[g] 以内であることを確認した。

- 試験内容

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 を示す。

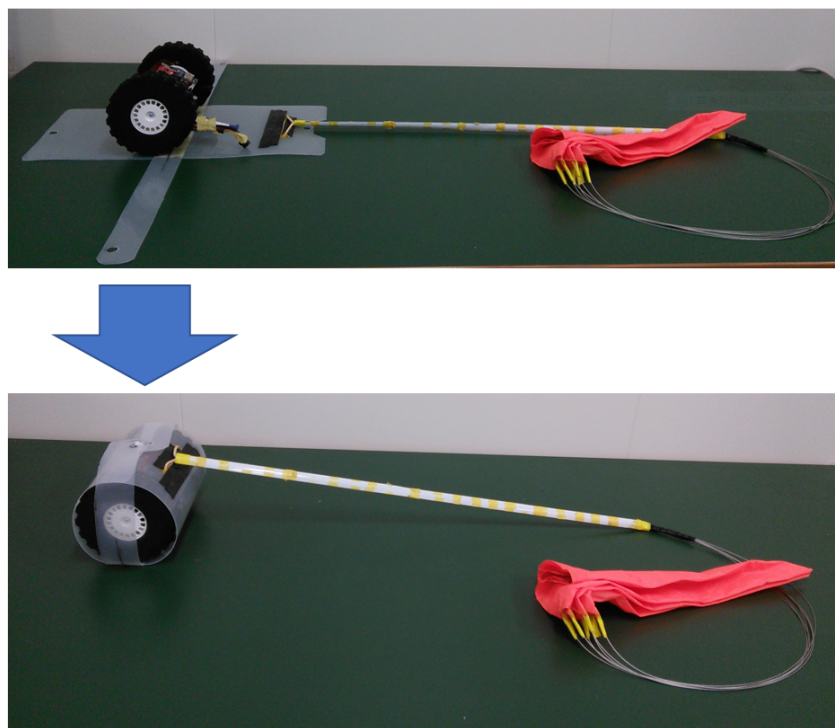


Fig. 6-1-1 保護ケースに収納させた CanSat

- 結果

質量計測結果を Fig. 6-1-2 に示す。

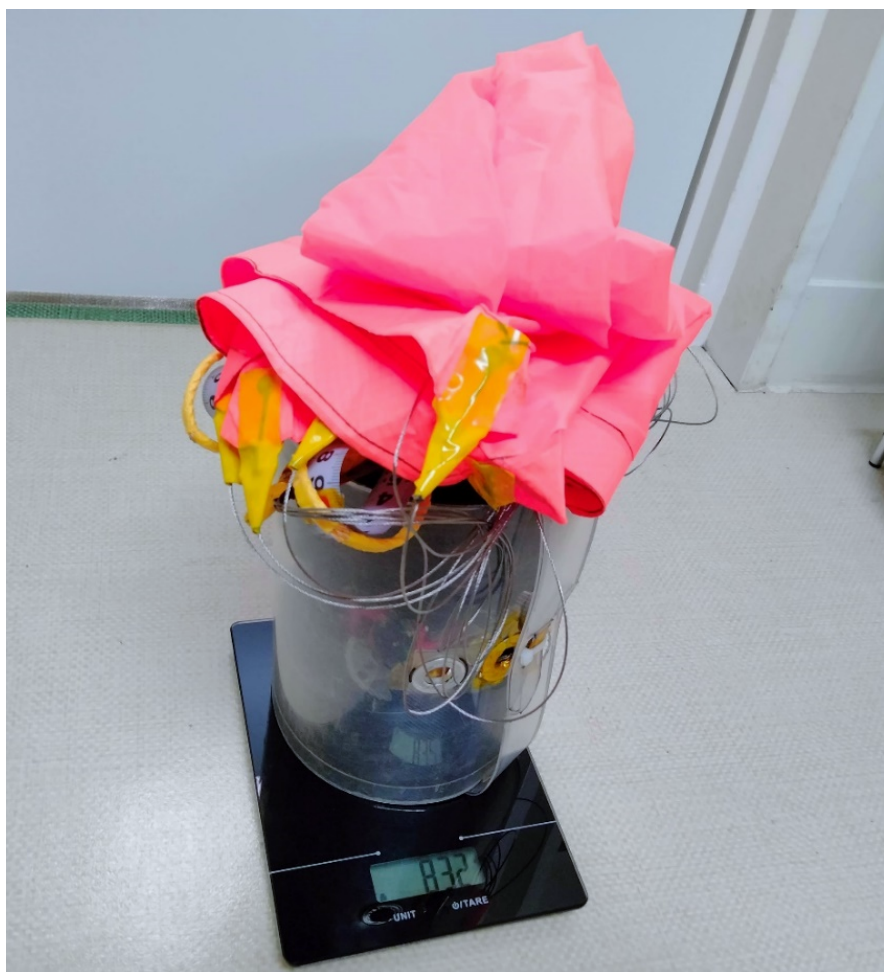


Fig. 6-1-2 質量測定結果

計測結果，目盛りは 832g だった．

CanSat の質量は、規定の重量 1050g を超えないことが確認できた。

【TPU タイヤ】

計測結果，目盛りは 947g だった。

CanSat の質量は、規定の重量 1050g を超えないことが確認できた。



Fig. 6-1-3 質量測定結果

● 考察

パラシュートを付けた質量測定の結果より、機体は既定の 1050g 以内であることが確認できた。

(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する．

- 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に収納することで規定寸法であること、また機体収納を 5 分以内にできることを確認した。キャリアの寸法奥行を Fig. 6-2-1, 外径を Fig. 6-2-2 に示す。

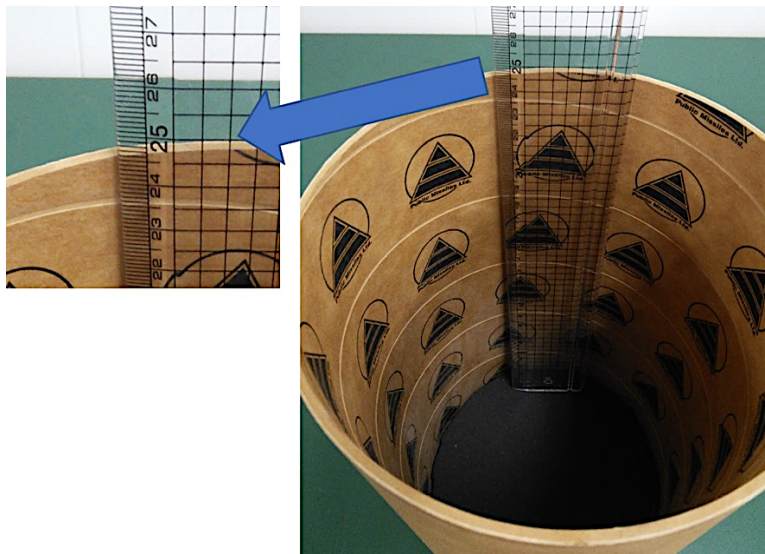


Fig. 6-2-1 キャリア収納ケースの奥行寸法

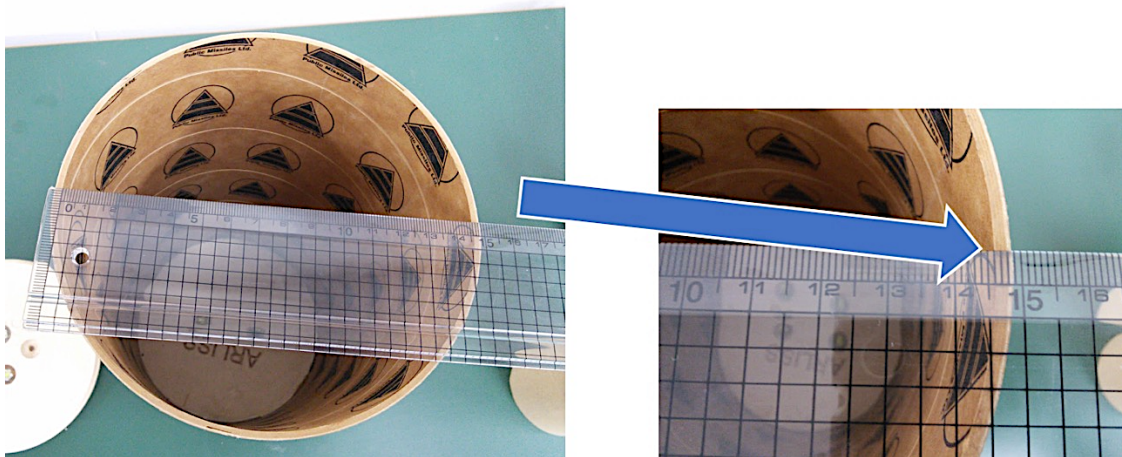


Fig. 6-2-2 キャリア収納ケースの外径寸法

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。保護ケースを覆った図を Fig. 6-2-3 に示す。

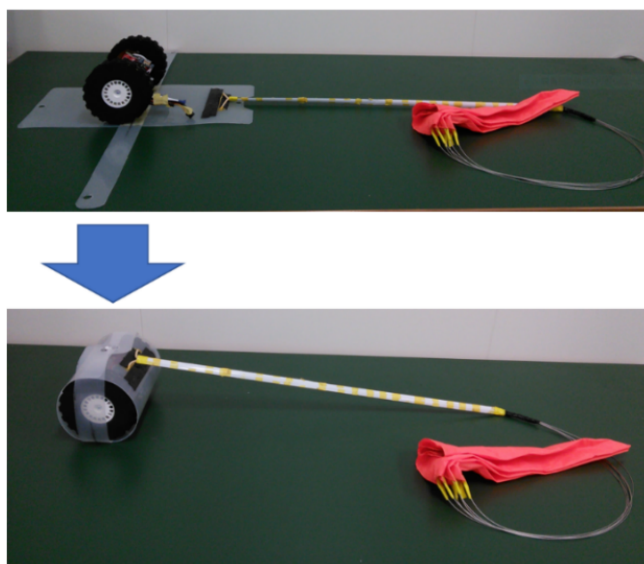


Fig. 6-2-3 保護ケースに収納させた CanSat

● 結果

規定の寸法のキャリアへ CanSat を収納できることが確認できた。
実験動画にてキャリアへ CanSat を収納した図を Fig. 6-2-4 に示す。



Fig. 6-2-4 実験動画でのキャリアへ収納した CanSat

また、収納後に自重で放出できることが確認できた。
実験動画にてキャリアから自重で放出した図をFig. 6-2-5に示す。

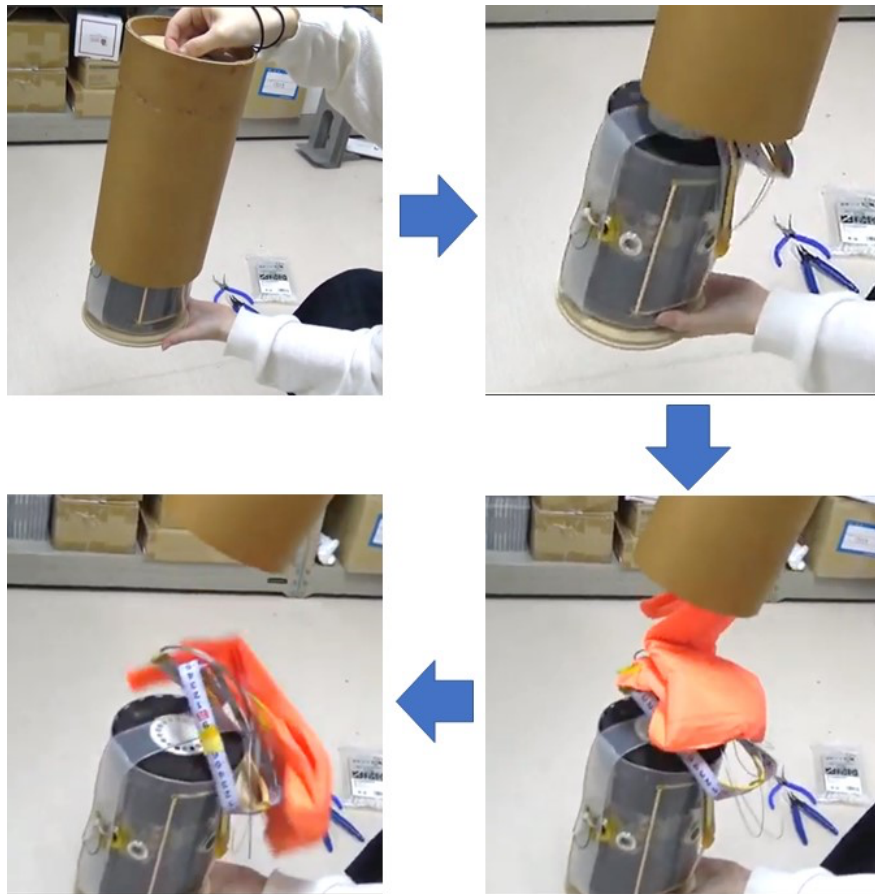


Fig. 6-2-5 実験動画でのキャリアから自重で放出した図

また放出までの時間を計測したところ2:59と3分以内であり、レギュレーションの機体収納時間が5分以内であることを確認した。

実験動画 URL : <https://youtu.be/8c5nkp1uJB8>

【TPU タイヤ】

Table. 6-2 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。

Table. 6-2 実験結果

回数	実験動画/画像	放出判定
1	 https://youtu.be/xys1eJziEeQ	自重で放出できた
2	 https://youtu.be/dJ7VQq4FPzg	自重で放出できた
3	 https://youtu.be/0apFImxjTqw	自重で放出できた

● 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) 長距離通信試験

- 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／解析の内容
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、そこから搜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. 6-3-1、送信機 Interplun IM920c を搭載した CanSat の写真を Fig. 6-3-2、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. 6-3-3 に示す。



Fig. 6-3-1 受信機 Interplun IM920

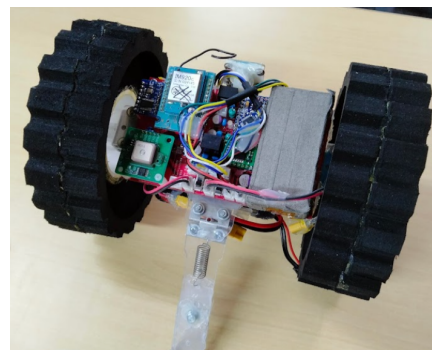


Fig. 6-3-2 送信機 Interplun IM920c 搭載 CanSat

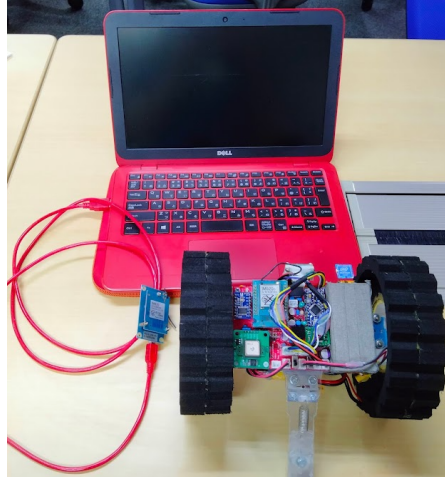


Fig. 6-3-3 実際に使用した受信機(左)と送信機(右)

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. 6-3-11、送信機位置の緯度経度を Fig. 6-3-12、受信機位置写真を Fig. 6-3-13、受信機位置の緯度経度を Fig. 6-3-14、受信～送信位置の Google MAP を Fig. 6-3-15、実際に取得したデータを Fig. 6-3-16 示す。



Fig. 6-3-11 送信機の位置の写真

緯度	34.79416666			⇒	緯度経度をGoogleMapで確認
経度	137.13055555				

↓

↑

緯度	34	度	47	分	39	秒	
経度	137	度	7	分	50	秒	

Fig. 6-3-12 送信機位置の緯度経度変換 (Fig. 6-3-11 の GPS ロガー一時分秒からの変換)



Fig. 6-3-13 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. 6-3-14 受信機位置の緯度経度変換 (Fig. 6-3-12 の GPS ロガー一時分秒からの変換)

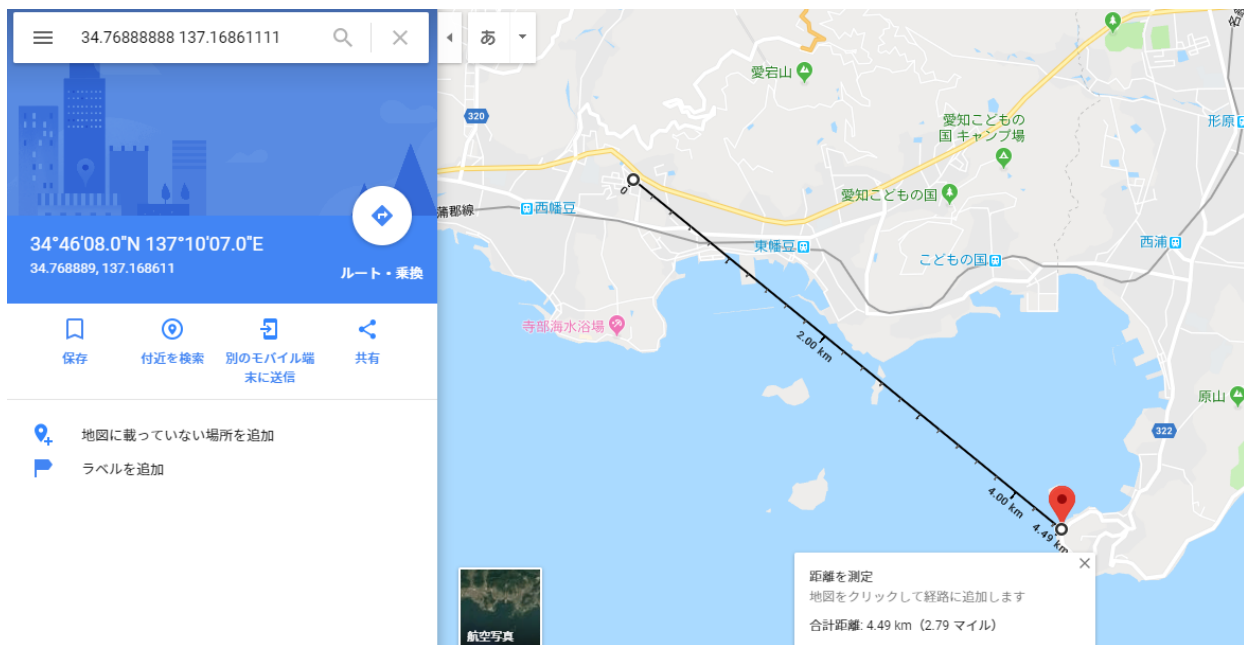


Fig. 6-3-15 4km 間の GoogleMap での距離測定

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。

●結論

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。

(V4) 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 程度であり減速機構として機能することを確認する。

● 試験内容

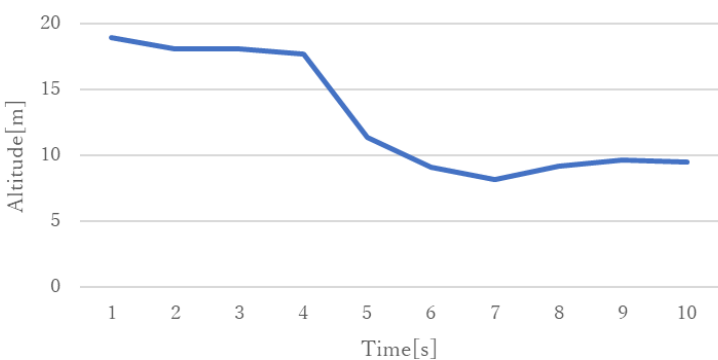
CanSat をつけたパラシュートを落下させ、落下時に測定した加速度が 5m/s の終端速度になっているか落下高度と時間から確認した。

● 結果

実験結果を Table. 4 にまとめた。

Table. 6 パラシュート投下試験結果

実験回数	終端速度	5m/s 以上か ○ or ×	実験データグラフ	動画 URL
1 回目	6.558 [m/s]	○	1回目 Altitude [m] Time [s]	https://youtu.be/5953Tbg-3go
2 回目	5.986 [m/s]	○	2回目 Altitude [m] Time [s]	https://youtu.be/t4yRBvbcNMk

3 回目	6.361 [m/s]	○	3回目  https://youtu.be/cxqcJjokrwA
------	----------------	---	--

【1 回目】

グラフ中 3s で 16.713m と 4s で 10.155m の間 1s 間の落下高度は 6.558m であり、 $6.558\text{m}/1\text{s} = 6.558\text{m/s}$.

【2 回目】

グラフ中 3s で 17.148m と 4s で 11.612m の間 1s 間の落下高度は 5.986m であり、 $5.986\text{m}/1\text{s} = 5.986\text{m/s}$.

【3 回目】

グラフ中 4s で 17.703m と 5s で 11.369m の間 1s 間の落下高度は 6.361m であり、 $6.361\text{m}/1\text{s} = 6.361\text{m/s}$

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

パラシュート減速の終端速度が平均で、5[m/s] 以上であることが確認できた。

(V5) 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 がそれに耐えられることを実証する。

● 試験内容

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レギュレーションの推奨値である 10G の加速度を与える。実際に行った実験写真を Fig. 6-5-1 に示す。



Fig. 6-5-1 遠心力で 10G をかけた様子

また、10G の根拠として、ARLISS2019 のロケット打ち上げ開始時に CanSat にかかった加速度を取得したデータ 1 回目を Fig. 6-5-2 に、2 回目を Fig. 6-5-3 に示す。サンプリング周波数は 100Hz、加速度センサの測定可能レンジはピーク 3 軸 16G 測定可能な LSM9DS1 を用いた。

2 回のデータの記録では、最大でも 10G に到達しておらず、また加速度がかかっている時間は 1 回目で約 5 秒、2 回目で約 6 秒間であった。このため、少し余裕を持ち、実験では 10G を 10 秒間に設定した。

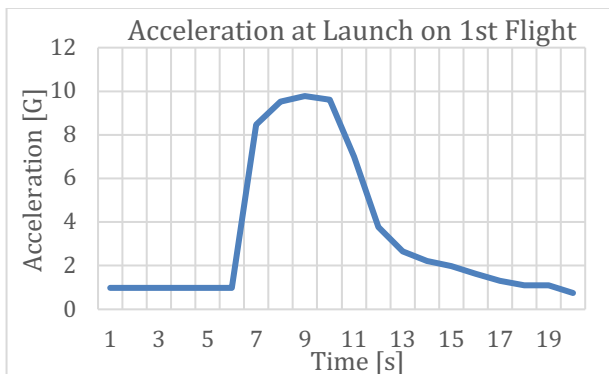


Fig. 6-5-2 ARLISS2019 ロケット打ち上げ時加速度 1

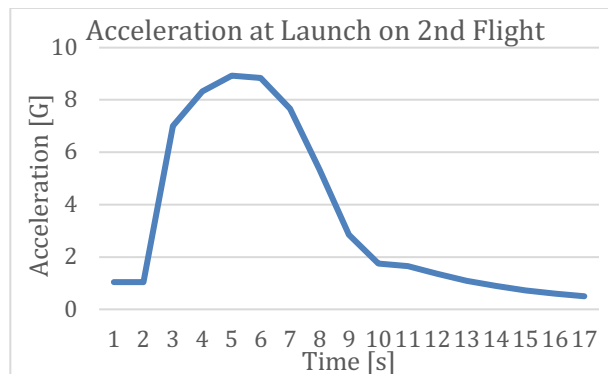


Fig. 6-5-3 ARLISS2019 ロケット打ち上げ時加速度 2

キャリアに搭載した 100Hz でサンプリングした加速度センサ (LSM9DS1) 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焼き切り動作を行い、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. 6-5-2 に示す

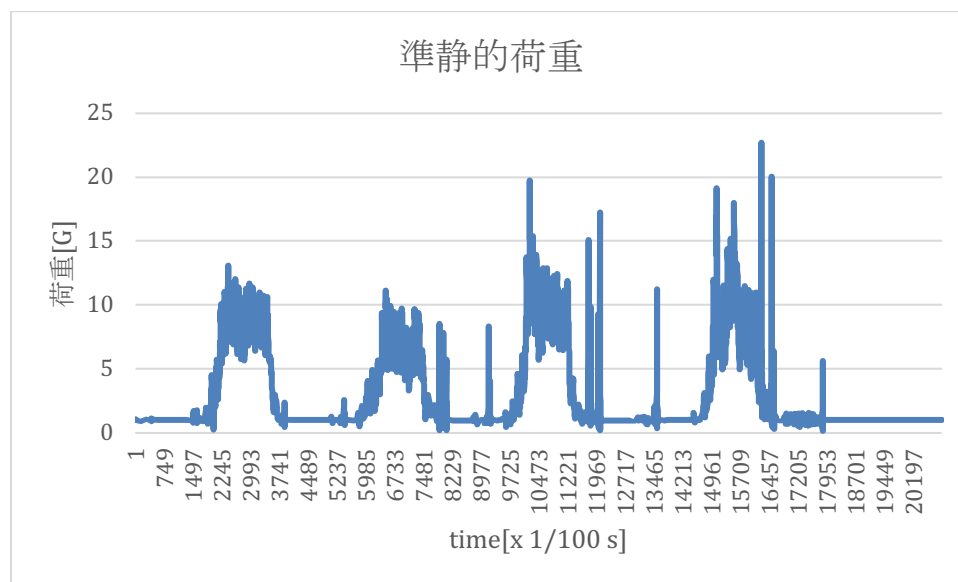


Fig. 6-5-2 加速度センサの数値（縦軸はかかった加速度[G]，横軸は時間[1s]）

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/amc7hidHvfY>

（センサ値事前確認 0:36～，キャリア収納 1:20～，準静的荷重運動 7:50～，センサ確認 11:00～，パラシュート離脱機構とモータ駆動 14:40～）

【TPU タイヤ】

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. 6-5-3 に示す



Fig. 6-5-3 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[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/v4YbdPTCzUo>

（センサ値事前確認 0 : 40～，静荷重運動① 6 : 45～，静荷重運動② 7 : 35～，静荷重運動③ 8 : 33～，センサ確認 9 : 10～，パラシュート離脱機構とモータ駆動と機体確認 12 : 45～）

● 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。

(V6) 振動試験

- 目的

ロケット打ち上げ時の振動に CanSat が耐えられることを確認する。

- 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 で、ロケット打ち上げ時間約 30 秒より誤差を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. 6-6-1、振動条件設定の PC 画面を撮影した写真を Fig. 6-6-2 に示す。

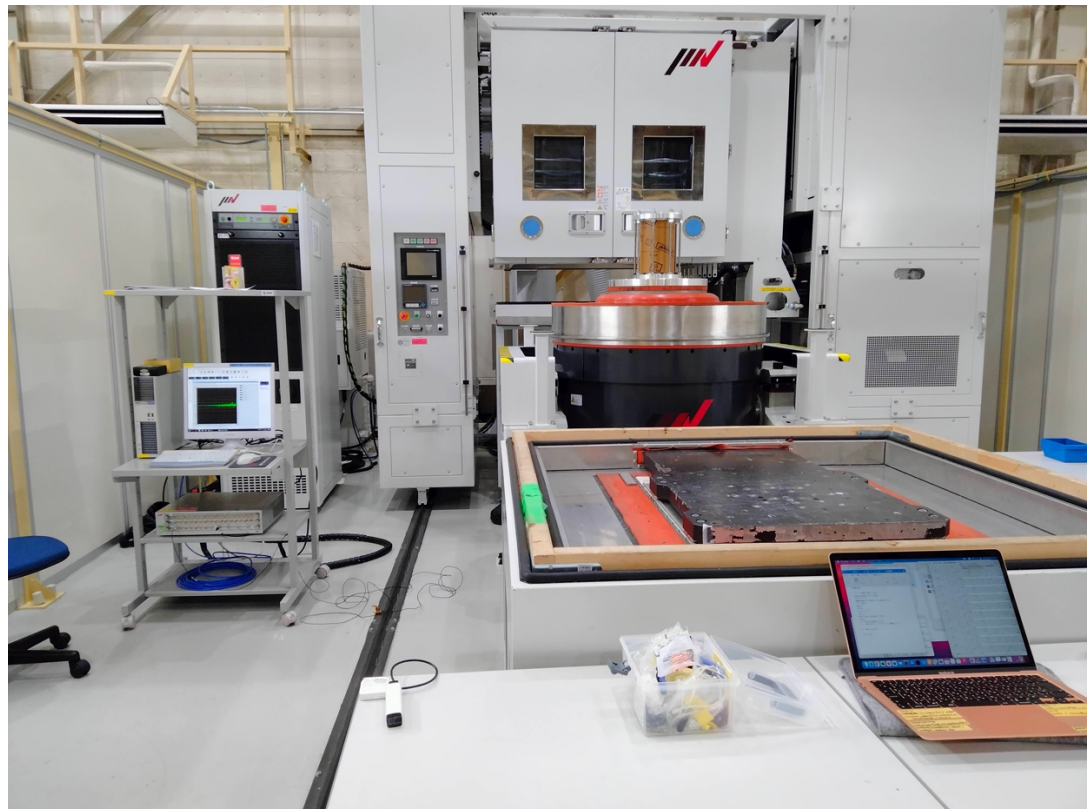


Fig. 6-6-1 実際の振動機 (茶色筒の中に CanSat 格納)

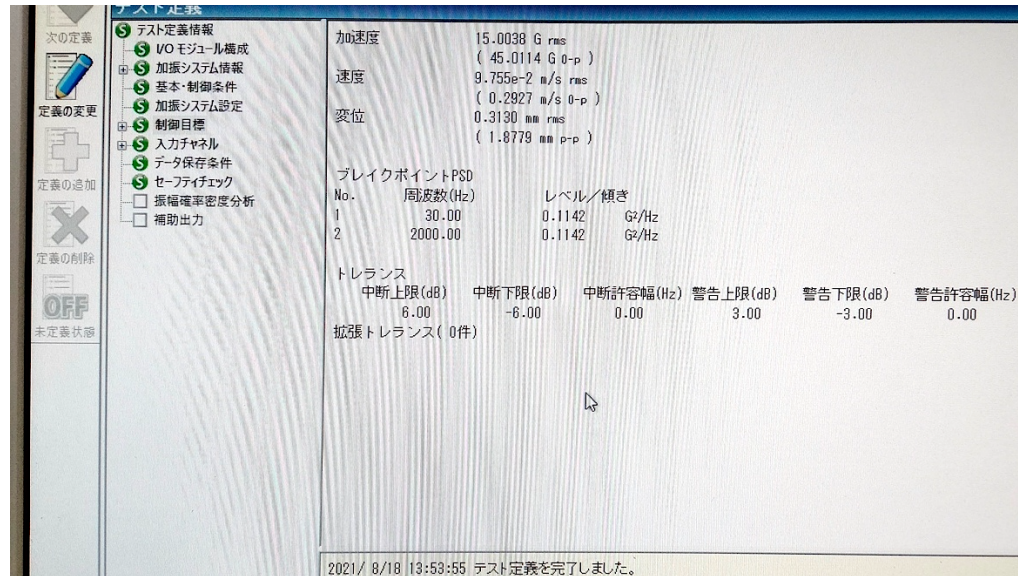


Fig. 6-6-2 振動条件設定画面

● 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、CanSat に搭載した加速度センサの振動時データを Fig. 6-6-3 に示す。

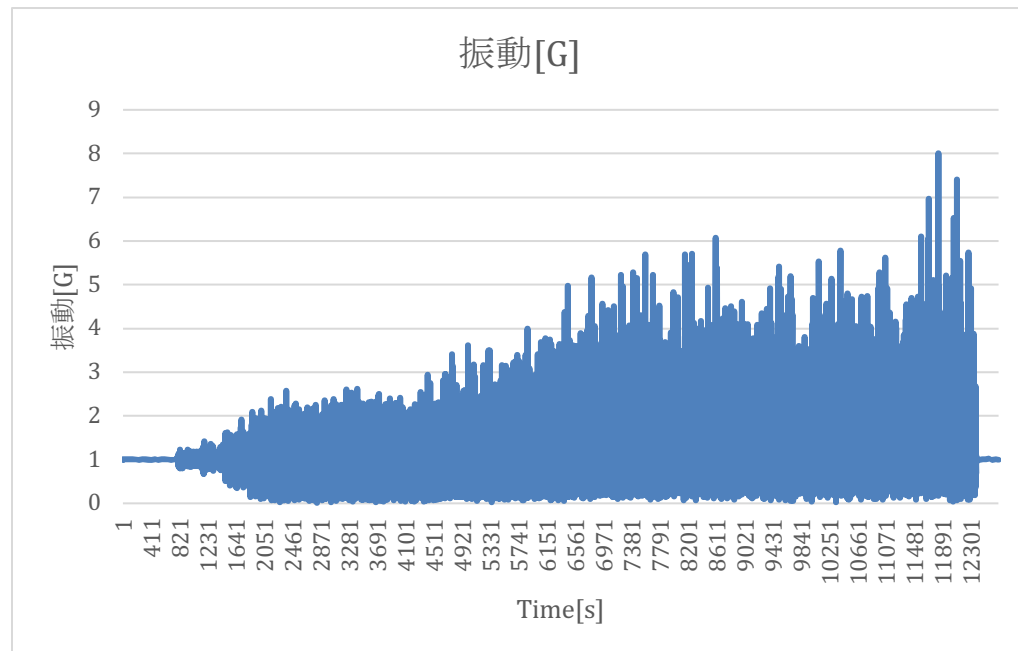


Fig. 6-6-3 加速度センサデータ

また、実際に振動を与えた振動機からの制御及び出力データを示す。出力のレベルをFig. 6-6-4、目標のGをFig. 6-6-5、制御応答のGをFig. 6-6-6、制御の m/s^2 をFig. 6-6-7に示す。（IMV 社提供データ）

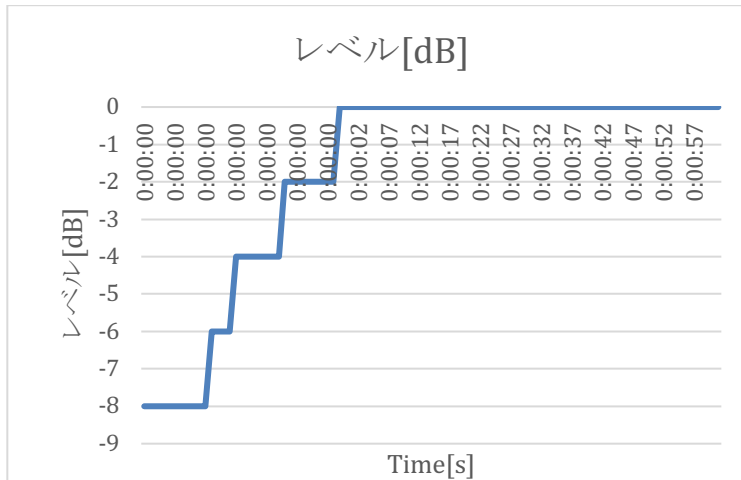


Fig. 6-6-4 出力レベル

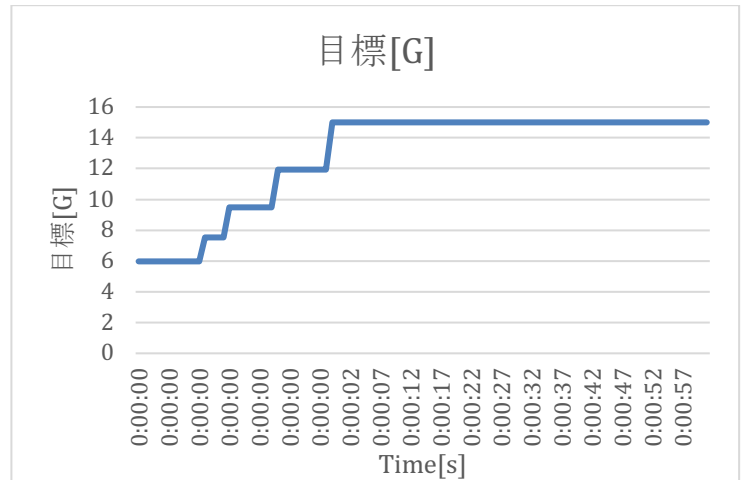


Fig. 6-6-5 目標G

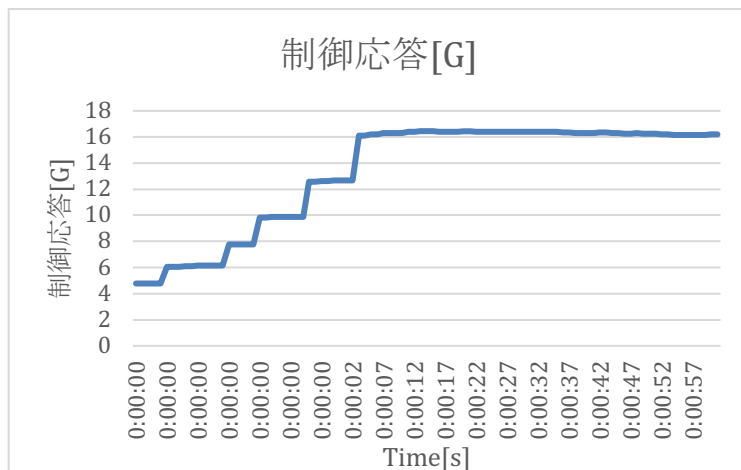


Fig. 6-6-6 制御応答G

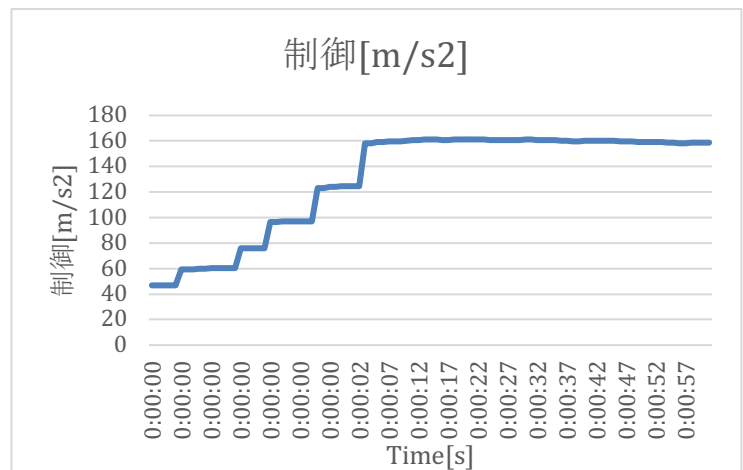


Fig. 6-6-7 制御 m/s^2

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/AZFDzesqxlI>

（電子回路動作事前確認 0:36～，キャリア収納 2:30～，治具固定 4:50～，振動～18:56，電子回路動作確認 21:11～，治具取り外し 21:52～，パラシュート離脱機構動作確認 29:30～，モータ動作確認 30:20～，機体破損確認 30:37～）

【TPU タイヤ】

CanSat に搭載した加速度センサの振動時データを Fig. 6-6-8 に示す.

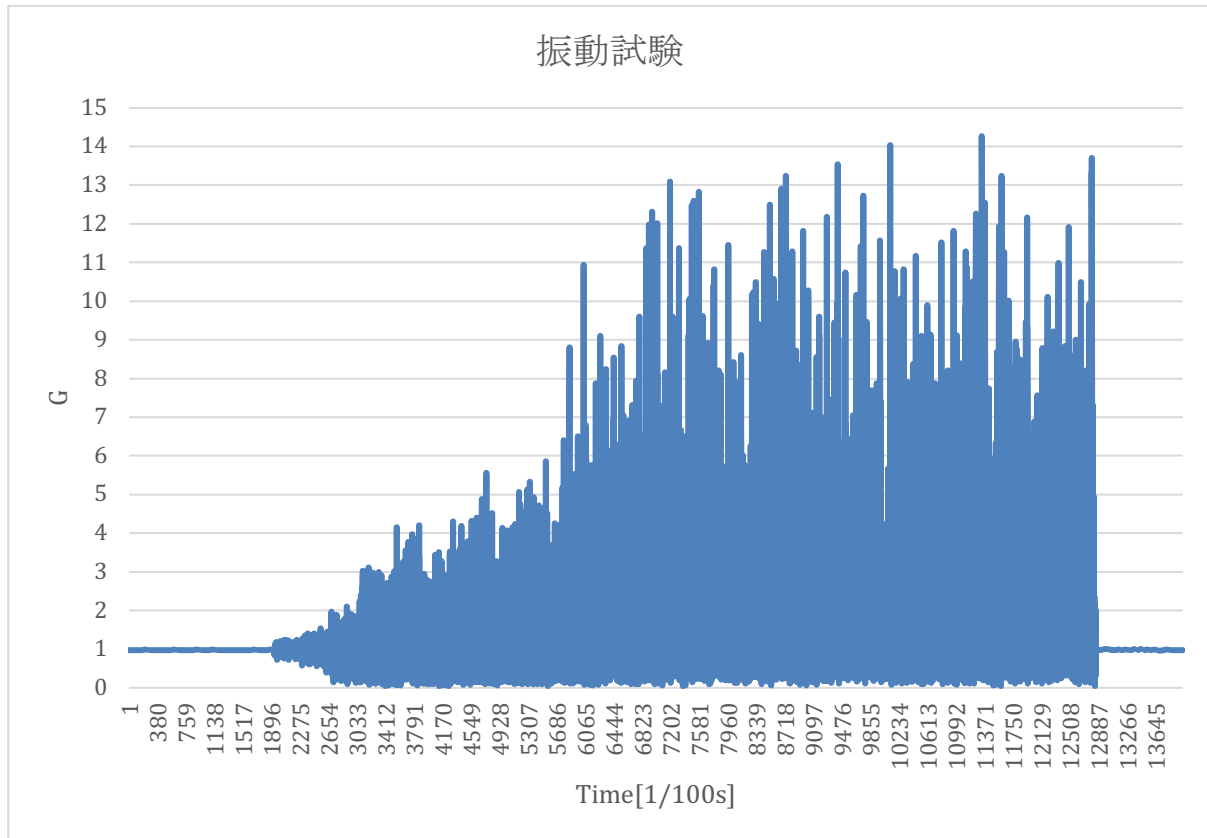


Fig. 6-6-8 加速度センサデータ

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/UZelq8FkNhY>

(電子回路動作事前確認 1:38~, キャリア収納 2:13~, 治具固定 3:18~, 振動 20:28~, 治具取り外し 23:23~, 電子回路動作確認 26:48~, パラシュート離脱機構動作確認 29:54~, モータ動作確認 30:28~, 機体破損確認 31:08~)

● 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.

(V7) 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う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大きな加速度を ARLISS2017 のデータから 30G 相当と想定し、評価試験を実施した。ARLISS2017 の放出時加速度データを Fig. 6-7-1 に示す。

この時のサンプリング周波数は 100Hz で加速度センサの測定可能レンジはピーク 200G まで 3 軸測定できる H3LIS331DL を利用した。

また、過去の ARLISS の本番データの記録では、ARLISS 2017 で取得したデータ及び ARLISS 2019 本番で取得した 2 回分のデータでは、最大で 30G を観測したため、30G を実験値とした。ARLISS2019 の加速度データを Fig. 6-7-2 に示す。

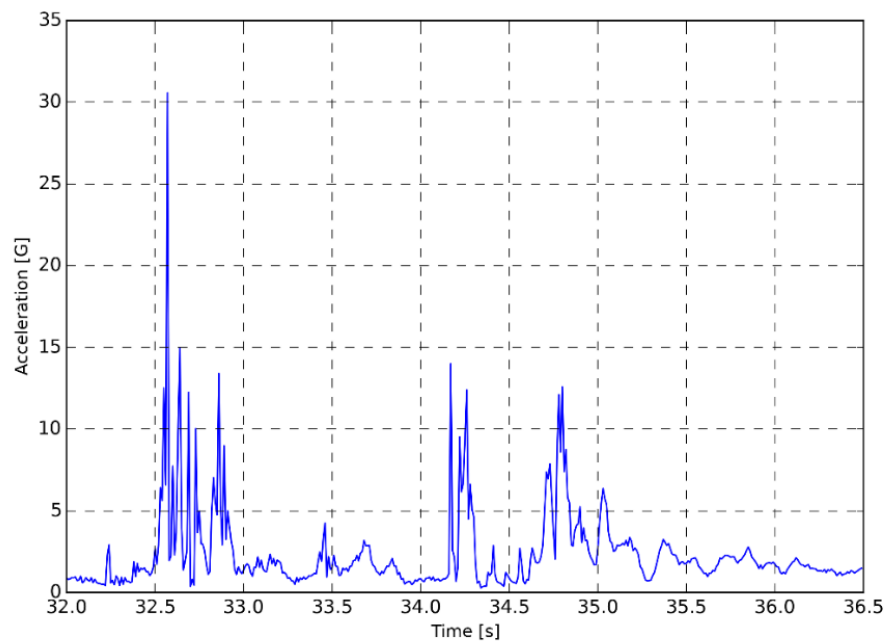


Fig. 6-7-1 ARLISS2017 の放出時加速度データ

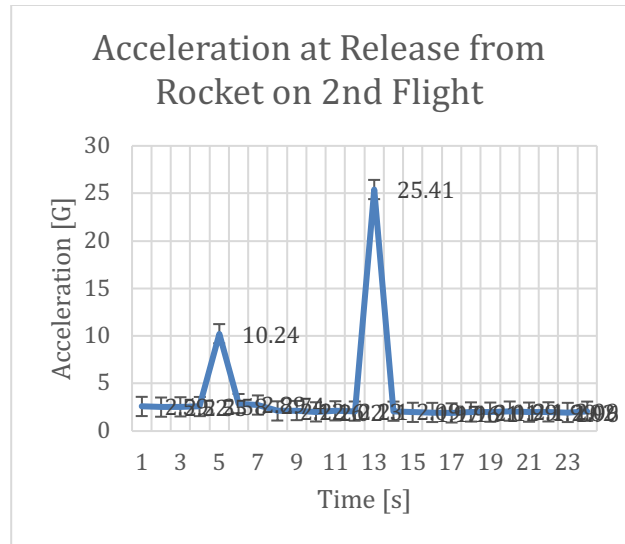
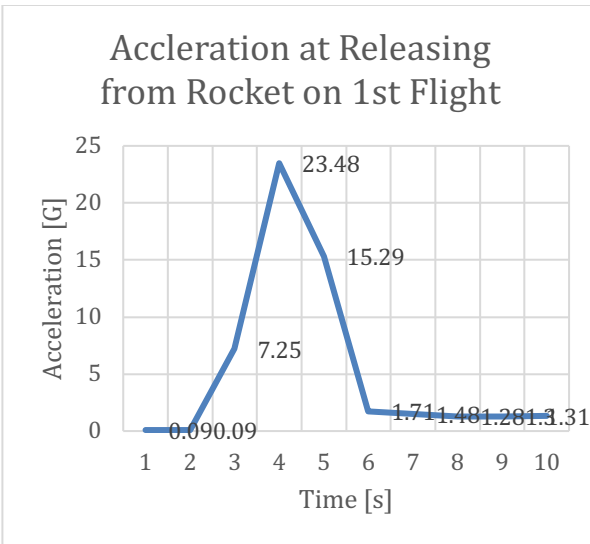


Fig. 6-7-2 ARLISS2019 加速度データ

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. 6-7-3 に示す。

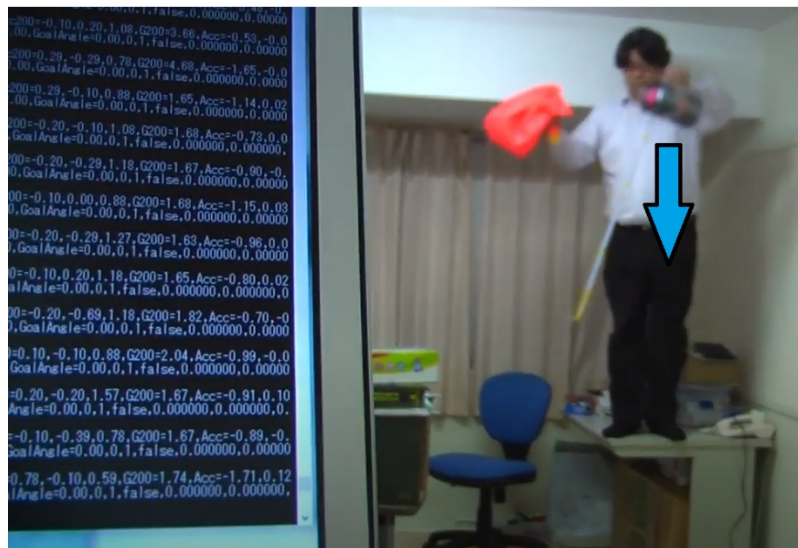


Fig. 6-7-3 実験のイメージ

● 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. 6-7-4 に示した。Fig. 6-7-4 では 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。

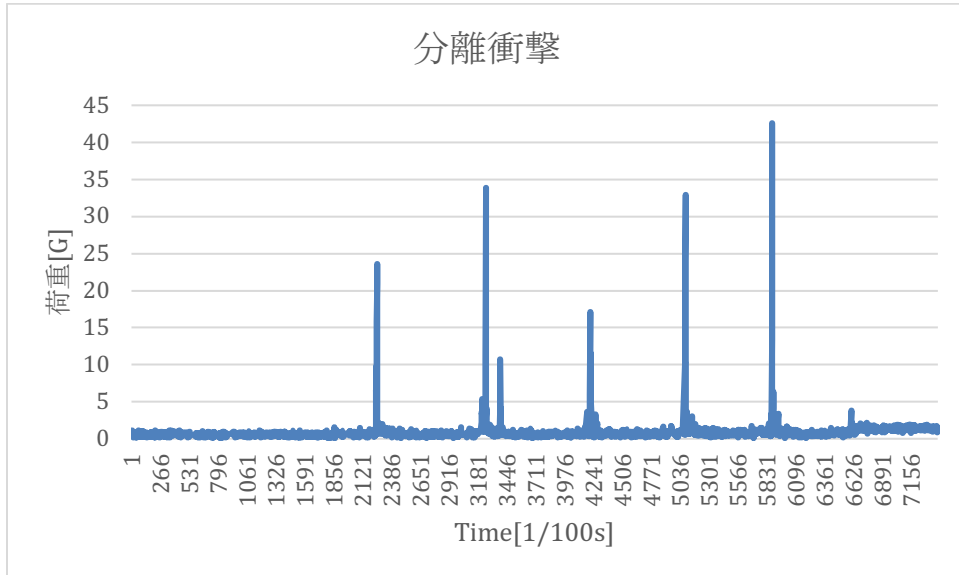


Fig. 6-7-4 分離衝撃の数値（縦軸はかかった加速度[G]，横軸は時間[1s]）

Fig. 6-7-4 から 23.63G, 33.9G, 17.13G, 32.9G, 42.58G の分離衝撃を与えた。30G 以上を観測した衝撃は 33.9G, 32.9G, 42.58G で観測できた。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/mRiDGR8nIJ0>

（センサ値事前確認 0:40～，キャリア収納 2:20～，分離衝撃運動 6:20～，パラシュート確認 8:00～，センサ確認 9:00～，パラシュート離脱機構とモータ駆動 10:49～）

【TPU タイヤ】

実際にかかった加速度のログデータを Fig. 6-7-5 に示した。

図では 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。

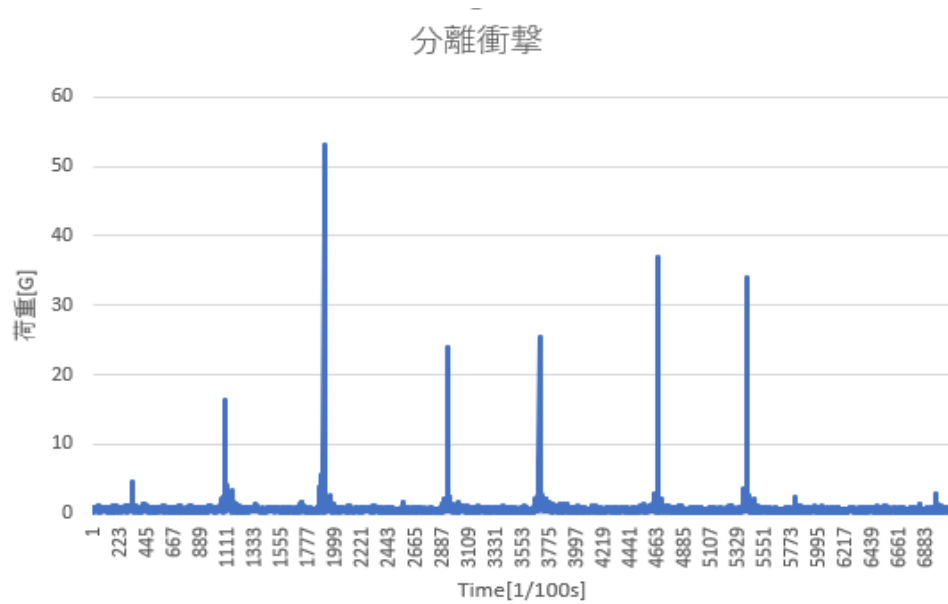


Fig. 6-7-5 分離衝撃の数値（縦軸はかかった加速度[G]，横軸は時間[1s]）

Fig. 6-7-5 から観測した衝撃は 16.35G, 53.16G, 24.05G, 25.47G, 39.95G, 33.96G だった。30G 以上を観測したのは 2 回目 53.16G, 5 回目 39.95G, 6 回目 33.96G で観測できた。

30G 以上を 3 回以上の衝撃を与えた後、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。

振動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/Le0Rr_j5CHq8

（センサ値事前確認 1 : 10～, 分離衝撃運動 4:40, パラシュート確認 5 : 44～, パラシュート離脱機構とモータ駆動と機体確認 6 : 45～）

● 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。

(V8) 通信周波数 ON/OFF 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 interplun 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。

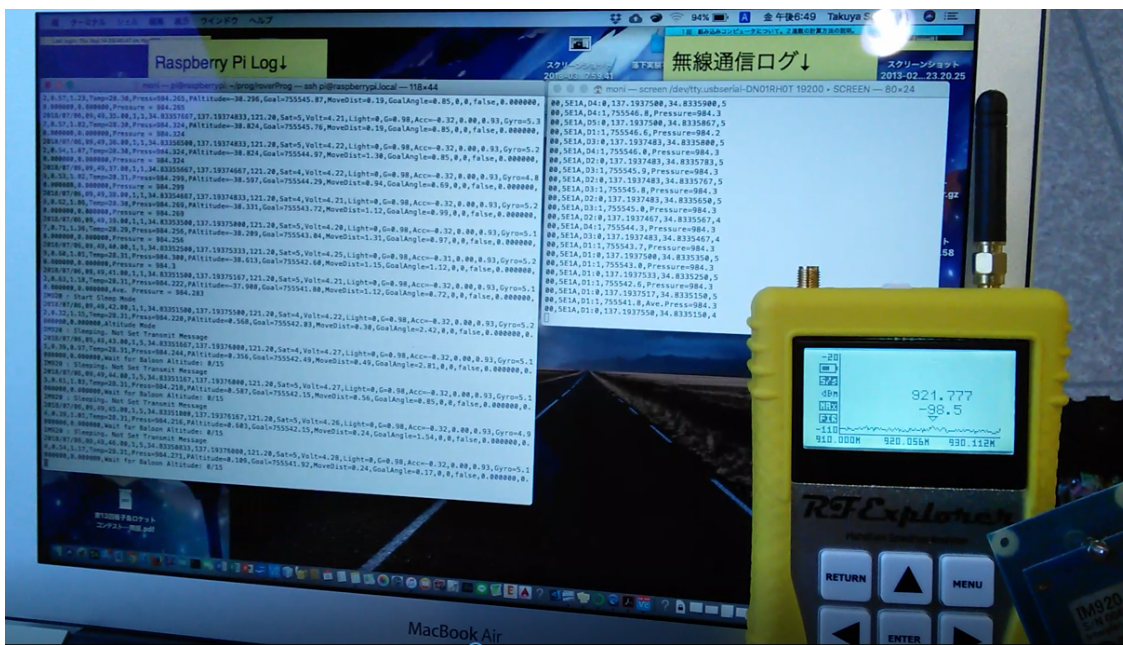


Fig. 6-8-1 電波 ON/OFF を目視で確認するための実験機器

● 結果

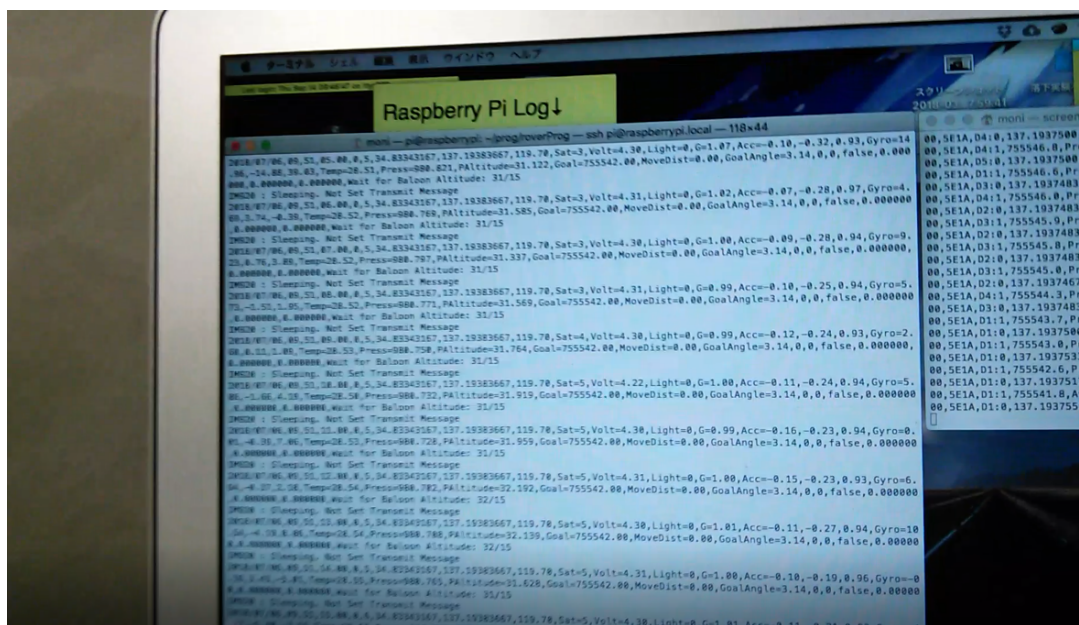
実際の結果を、通信機 OFF を①、設定した高い高度を観測したかを②、高い高度から落下し終え通信機が ON にしたかを③として結果をまとめた。

- 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。



② 建物 9F に上昇後、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。



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。

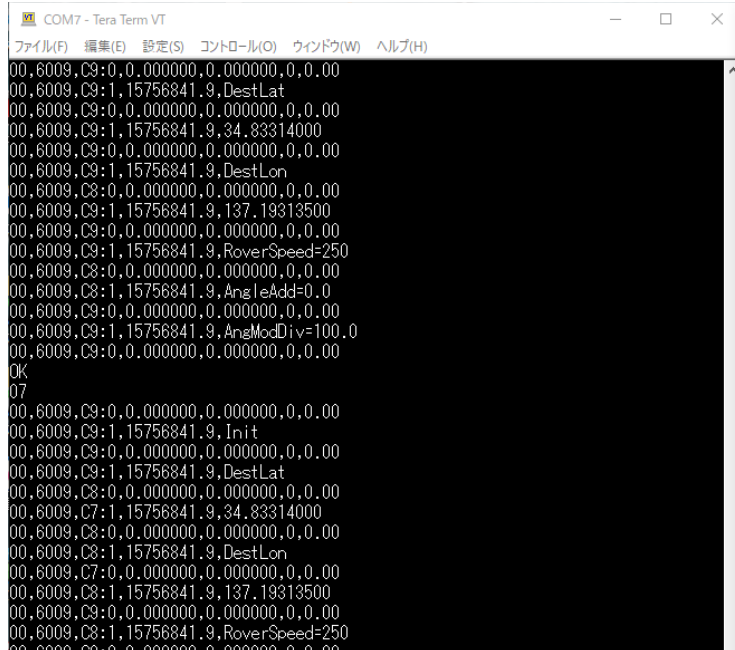
実験では 5 チャンネルから 7 チャンネルに設定変更を行った。実際のプログラムを Fig. 6-9-1 に示す。

```
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,AngleModDiv=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(動画提出)

● 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 サンセンサにより放出判定（本番では V8 の高度判定）
 - 3 着地衝撃
 - 4 パラシュートからの離脱
 - 5 目的地まで GPS 位置情報で自律走行
 - 6 Object Detection ゴール接近&判定
 - 7 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. 10-1 に示す。

Table. 10-1 End to End 結果

成功 Or 失敗	詳細	URL
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/C22BAId8d8g 4:13 キャリア放出&着地衝撃 5:38 パラシュート離脱& GPS 情報から走行開始 6:25 Object Detection での ゴール認識開始 7:20 ゴール判定 8:19 Object Detection ログ画像 10:52 走行軌跡ログ取り出し

また、制御履歴を V11 制御履歴レポート試験にて記載した。

【TPU タイヤ】

実験結果を Fig. 10-2 に示す.

Fig. 10-2 End to End 結果

成功 or 失敗	詳細	URL
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. 【Deep learning でのゴール検知～ログ取り出し】 ゴール約 3m 付近から Deep Learning 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/vNzpGcDBHFk 3:44 キャリア放出 & 着地衝撃 5:28 パラシュート離脱 6:06 GPS 情報から走行開始 7:13 Deep Learning でのゴール認識開始 8:40 ゴール判定 11:27 走行軌跡ログ取り出し

● 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) 制御履歴レポート作成試験

● 目的

End to End 試験のデータから作成し、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

CanSat の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 GPS データやセンサ値や制御値をサンプリング 1s と 0.01s でマイコンの SD カードに保存する。

ゴール判定後、ログデータファイルを読み取り、テキストデータを Excel ファイルでまとめ見やすい形にして表示した。また、走行軌跡が見やすくなるように、ステアリングを色で変化させた走行軌跡を表示する。

● 結果

実際に V12End 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 Fig. 11-1 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。

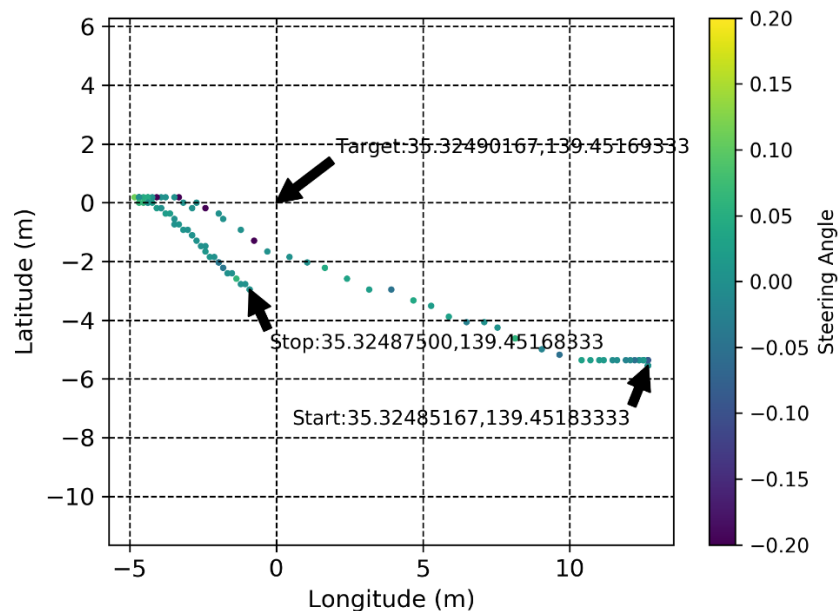


Fig. 11-1 End to End 走行軌跡

【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 3p 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. 13-2 に示す.

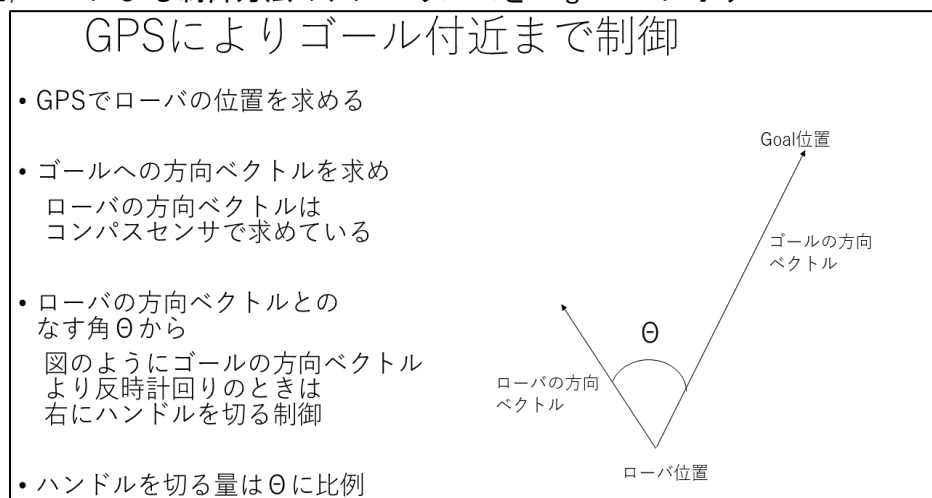


Fig. 13-2 制御方法

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.
- ・ **A0 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】

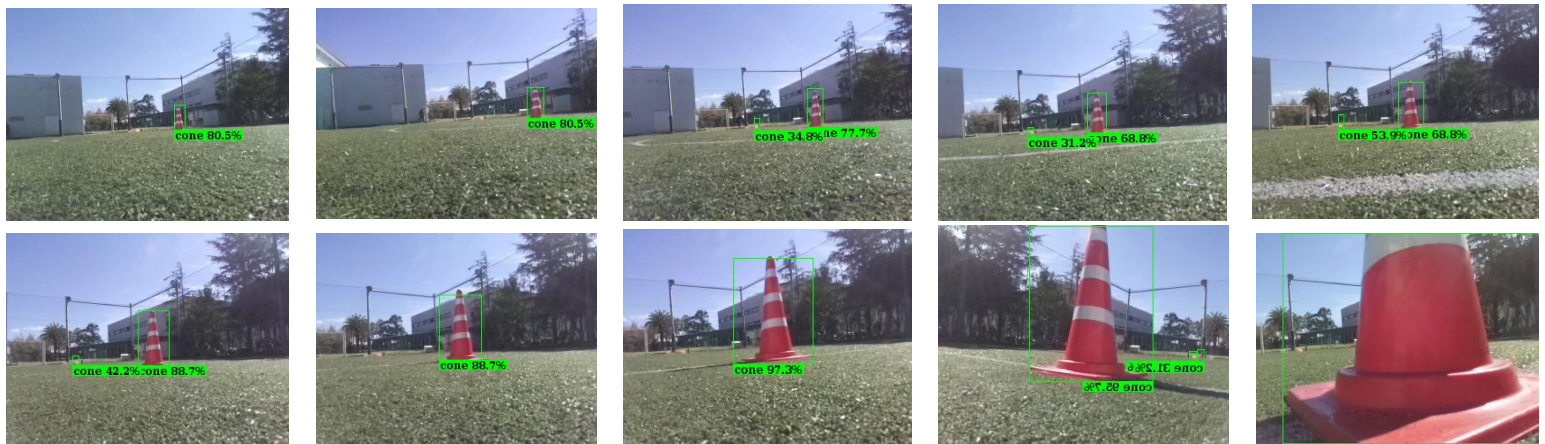
また、End to End での Object Detection の画像ログを Table 11 に示す。

Table 11 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 4 回巡回



発見後、10 回の巡回 & 直進でゴール判定



また、実際の制御ログデータをまとめた URL を下記に示す。

○End 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/1KQbfpw0UfTRibQ5sM4mzXXoUbd1CZASD?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。

1. 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

● 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

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 落下試験平均 6.2m/s) (a = 重力加速度 9.8g)

$$\text{終端速度 (パラシュート投下試験参照)} = \frac{v^2 + v_0^2}{2a} = \frac{6.2^2 - 0}{2 \times 9.8} = 1.9[m]$$

1.9[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

実験結果を Table 12-1 に示す。

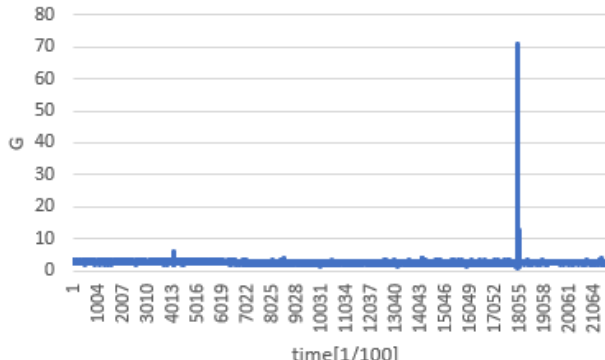
Table 12-1 着地衝撃試験結果

実験回数	衝撃 [G]	衝撃グラフ	破損 ○ or ×	詳細	動画 URL
1 回目	97.4	着地衝撃	○	回路センサ類: ○ パラシュート離脱機構: ○ モータ: ○ 機体: ○ 着地衝撃後、CanSat は正常であった。	https://youtu.be/GPaVqlyHe3g 0:30 センサ値事前確認 3:12 落下位置確認 4:50 落下 1-3 5:40 センサ値確認 7:13 パラシュート離脱機構確認 8:10 モータ動作確認 8:28 機体確認
2 回目	61.6				
3 回目	72.8				

【TPU タイヤ】

実験結果を Table 12-2 に示す.

Table 12-2 着地衝撃試験結果

破 損 ○ r x	着地衝撃グラフ	詳細	動画 URL
○	着地衝撃2 	回路センサ類 : ○ パラシュート離脱機 構 : ○ モータ : ○ 機体 : ○	https://youtu.be/YtgiXSjEs08 0:35 センサ値事前確認 2:44 落下位置確認 3:26 落下 1 6:37 センサ値確認 5:20 パラシュート離脱機 構確認 5:38 モータ動作確認 9:46 機体確認

約 70G の荷重が加わった CanSat の状態を調べた。結果、回路センサ類、パラシュート離脱機構、モータ、機体に問題はなかった。

● 考察

CanSat が着地衝撃に耐えられることを確認した。

(V13) ゴール検知試験

● 目的

ミッションである Deep Learning 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. 13-1 に示す.

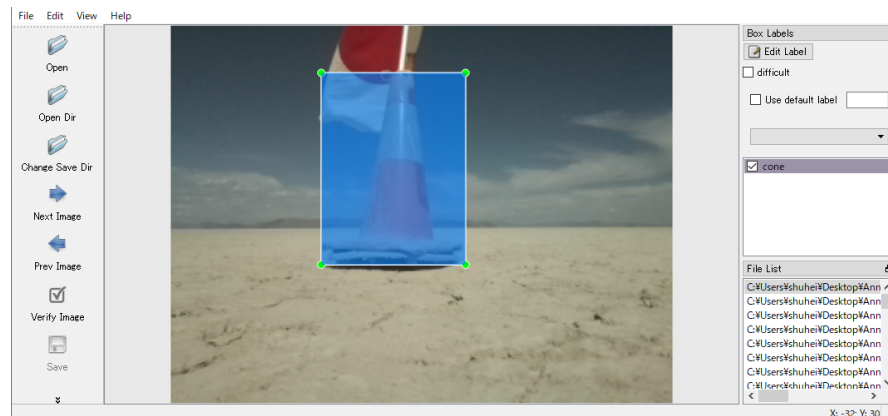


Fig. 13-1 Annotations の例

【走行制御方法】

Raspberry Pi Camera v2 から撮影された画像から、Object Detection で認識結果が Cone を 50%以上で見つけるとゴールが見つかったと判定し、認識範囲の重心座標の方へハンドルを切ってゴールへ接近走行している。また、複数ゴールが見つかった場合はコーンの可能性が 1 番高いものだけを採用している。ゴールが見つからない場合は左旋回を繰り返す。

ゴールが見つかったときの例の画像を Fig. 13-2 に示す。

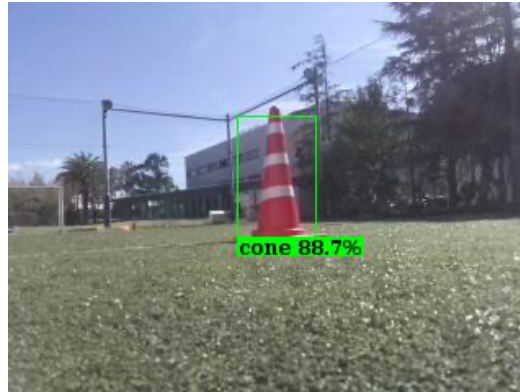


Fig. 13-2 Object Detection でのゴール発見の例

ゴール判定では、横幅が 250 ピクセル以上のときゴール判定をしている。ゴール判定をした例の画像を Fig. 13-3 に示す。



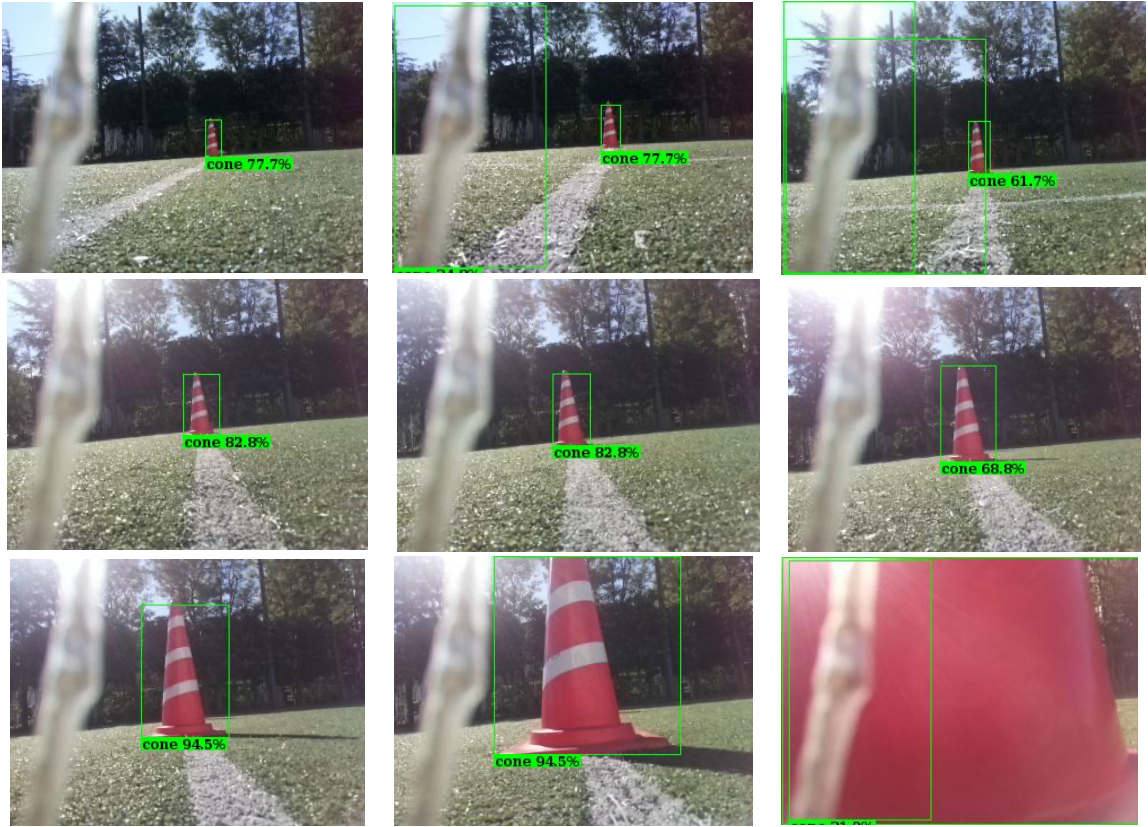
Fig. 13-3 Object Detection でのゴール判定の例

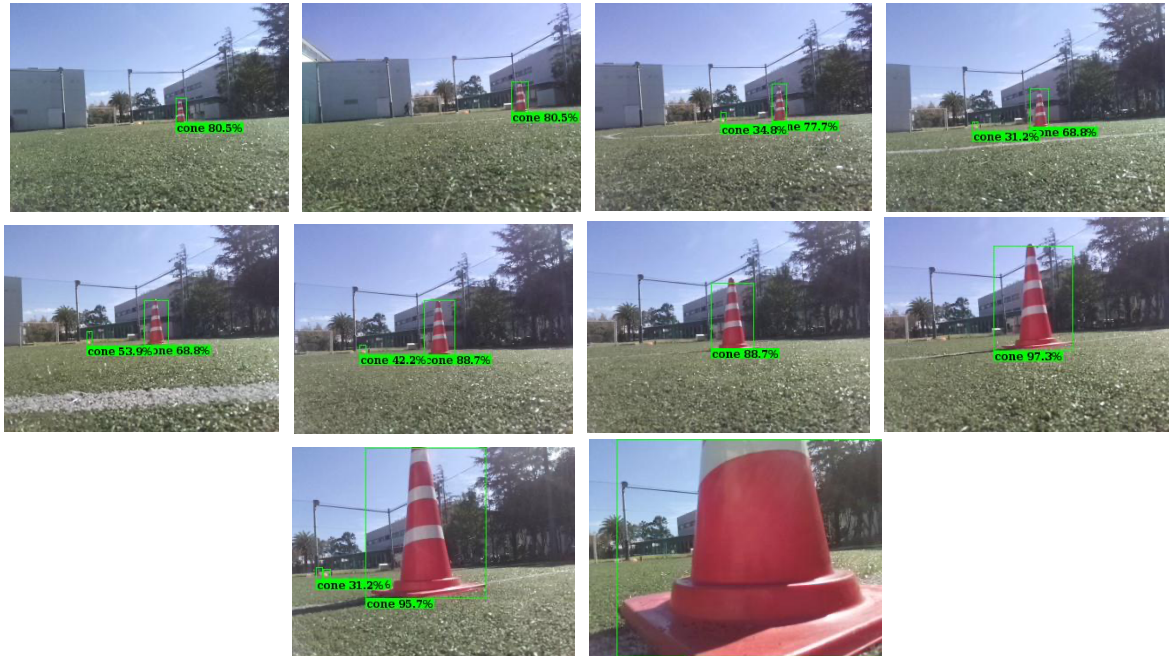
● 結果

結果を Table 13 に示す.

Table 13 ゴール検知試験結果

実験回数	0m goal	実際の撮影写真&詳細	動画 URL
1回目	○	ゴール発見まで5回巡回  発見後5回の巡回&直進しゴール判定 	https://youtu.be/Wzyks52hrss

		ゴール発見まで 6 回巡回 	
2 回 目	○	発見後, 9 回の巡回 & 直進でゴール判定 	https://youtu.be/IE6J0sJyuCE

		ゴール発見まで 4 回巡回 	
3 回 目	○	発見後、10 回の巡回 & 直進でゴール判定 	https://youtu.be/nokhP7sH7gI

また、画像と制御ログをまとめた URL を下記に示す.

○ゴール検知試験画像ログ, 制御ログ :

<https://drive.google.com/drive/folders/1X6ckDw7wWwkSn3dRpMSOs3WIk00DerYV?usp=sharing>

● 考察

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

第7章 工程管理、ガントチャート（スプレッドシートを推奨）

第8章 各担当（ハード・ソフト・全体などの進行状況を記入）

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。全体的には6月にプロトタイプ作成&テスト、7月に EndtoEnd 以外の審査試験、8月に EndtoEnd&最終調整とリカバリーを行う。ガントチャートを Fig.7に示す。

【6月】

6月中は CanSat のプロトタイプを作成し、問題点などテストし確認し、改良を重ねていく。

第1週は従来より改良する点を考案し部品や材料の発注をする。

第2週から4週にテスト CanSat 作成及び問題点など確認し改良を重ねる。

第5週はリカバリーができるようできるだけスケジュールを空け、テストを重ねる

【7月】

7月中は予備審査に向け、EndtoEnd 以外の試験をできる限り終える。（各2試験1週間）

第1週は質量試験，キャリア収納試験。

第2週は準静荷重試験，分離衝撃試験。

第3週は落下試験，走行性能確認試験。

第4週は振動試験。（NSE 予備審査書提出）

第5週はミッションのゴール検知試験。（ACTS 予備審査書提出）

【8月】

8月中は 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～第2週 EndtoEnd，制御履歴試験を行う。（NSE 本審査書提出）

第2週までに6月から始めたミッションのシステムを確立する。

第3週で試験の結果を踏まえた本番用の CanSat とスペアを製作する。（ACTS 本審査書提出）

第4週～第5週にリカバリー及び最終調整や準備を行う。

試験項目 \ 月・週	開始	終了	6					7					8					9
			1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1
搭載新機能考案	6月1日	6月5日																
搭載部品・材料受注	6月1日	6月5日																
テスト機体作成	6月6日	6月26日																
テスト回路作成	6月6日	6月26日																
テストシステム作成	6月6日	6月26日																
テストパラシュート作成	6月6日	6月26日																
テスト用実験	6月6日	6月30日																
質量試験	7月1日	7月3日																
キャリア収納試験	7月1日	7月3日																
準静的荷重試験	7月5日	7月9日																
分離衝撃試験	7月5日	7月9日																
落下試験	7月12日	7月16日																
走行性能確認試験	7月12日	7月16日																
振動試験	7月19日	7月21日																
ゴール検知試験	7月26日	7月30日																
ミッション機能作成	6月1日	8月18日																
ミッション動作試験	6月21日	8月18日																
End to End 試験	8月2日	8月12日																
制御履歴試験	8月10日	8月12日																
本番機体・回路作成	8月10日	8月13日																
最終調整	8月16日	9月1日																
準備	8月16日	9月1日																

Fig.7 ガンチャート

ガンチャートスプレッドシート：

<https://drive.google.com/file/d/13xo0SLPcvLZRlZg1tPhHdXlwntUTl-U-/view?usp=sharing>

第9章 大会結果報告

1. 目的

機械学習の画像認識の中で複数の物体とその距離を検知ができる **Object Detection** を利用し、 **Object Detection** によるゴール画像認識で **0m** ゴールする。

2. 結果

投下試験を行ったときに目視で確認できたこと、計器類より取得できたデータ、動画など総じて投下試験で得られたものを示す。投下機会ごとに分けてください。

【1回目結果】

→**CanSat** が縦向きになり、スタックで走行不可

1回目の投下では、パラシュートが開傘し減速した速度で落下し、その後離脱機構の溶断によりパラシュートから離脱し走行を開始した。**ACTS** 会場での長い草の多い地点に落下したため、**CanSat** が頻繁にスタックしスタック離脱動作で抜ける動作を繰り返していた。

スタック離脱後に走行していると、**CanSat** が縦向きになりその状態からの復帰ができず、走行不可となった。

実際のスタックしてる **CanSat** を Fig.9-1 に示す。



Fig.9-1-1 CanSat 縦向きによるスタック

また、競技での走行軌跡を Fig.9-1-2 に示す。

縦軸が **Latitude** で横軸が **Longitude** で、それぞれ走行した座標を **ls** ごとにプロットし、スタートした位置とストップした位置を矢印にて示してある。また、プロットの色変化により、それぞれの地点でモータのステアリングがどの方向に舵を切ったか制御値の可視化ができるようになっている。**0.20(黄色)**が右旋回の最大値で、**-0.20(紫)**が左旋回の最大値となっている。

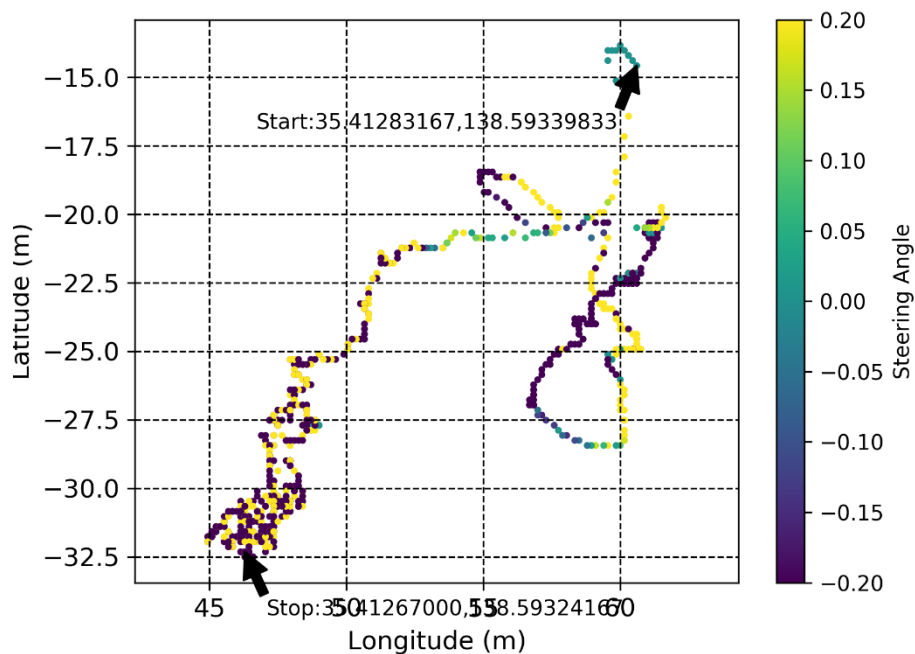


Fig.9-1-2 ACTS1 回目走行軌跡

【2回目結果】

→Object Detection による 0m ゴール達成！

ACTS2 回目の投下では、パラシュートが開傘し減速した速度で落下し、その後離脱機構の溶断によりパラシュートから離脱し走行を開始した。走行開始後、1 回目に使用した従来のタイヤから、TPU 樹脂製の拡張タイヤに変更することにより、ACTS 会場の長い草が走破できなかった問題を解決し、安定走行して GPS 位置情報と磁気センサによるモータ回転数制御でゴール数 m 範囲まで接近し、ミッションである Object Detection による 0m ゴールを達成した。ゴールした際の写真を Fig.9-2-1 に示す。



Fig.9-2-1 ACTS2 回目結果

また、ACTS2 回目の走行軌跡を Fig.9-2-2 に示す。

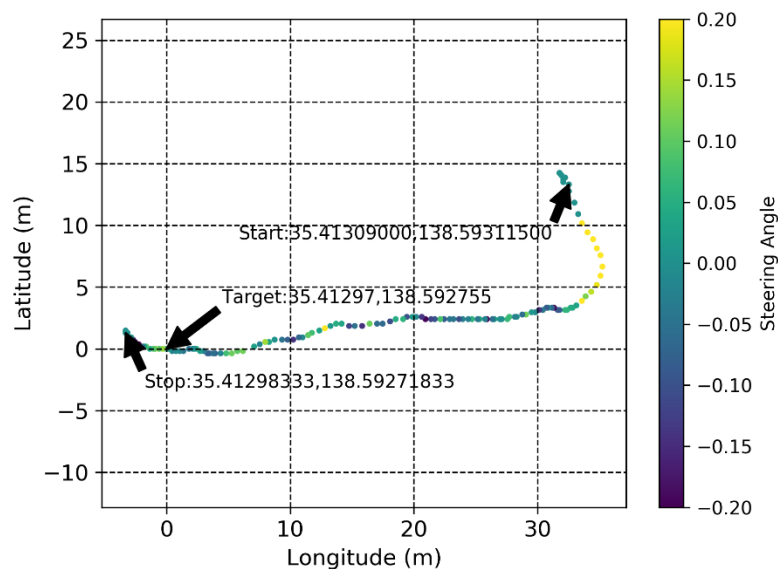


Fig.9-2-2 ACTS2 回目走行軌跡

また、**Object Detection** の結果、最初遠くからコーンを発見しつつ、予め学習させた会場の青色のコーンも同時に別物体として多数認識し、ゴールの赤色コーンだけに走行して行き、**0m** 地点でゴール判定をした。

実際の **Object Detection** での認識結果を GIF 化した画像を **Fig.9-2-3** に示す。



Fig.9-2-3 Object Detection 結果

制御履歴が認められ、カムバックアワードを受賞した。表彰状を **Fig.9-2-4** に示す。

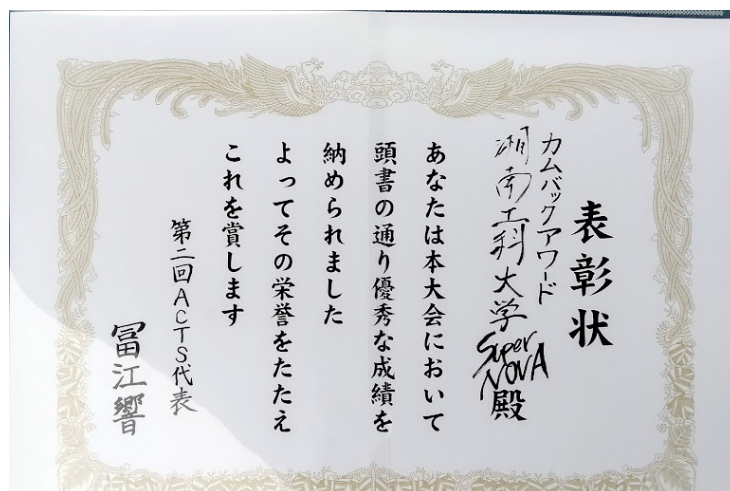


Fig.9-2-4 0m ゴールでカムバックアワード受賞

3. 考察

【ACTS1回目 考察】

サクセスクライテリア達成度

ACTS1 回目結果は **CanSat** が縦向きになり走行不可となった。

また、ACTS1 回目結果から、サクセスクライテリアの達成度を示した表を **Table9-1** に示す。

ACTS1 回目の結果はミニмумサクセスである「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．」を達成した。しかし、フルサクセスの「ゴールから 5m 以内まで接近し、その後 **Object Detection** でゴールへ 0m まで近づきゴール判定をする。」は達成できなかった。

Table 9-1 ACTS1 回目結果サクセスクライテリア

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。	◎
フル サクセス	ゴールから 5m 以内まで接近し、その後 Object Detection でゴールへ 0m まで近づきゴール判定をする。	×

原因追及

原因の 1 番の理由は、スタックの可能性のある従来タイヤで走破できると思い込み、従来タイヤで 1 回目投下を行ったことである。

従来タイヤとは、NR スポンジ素材の約 3mm 程度溝のグリップがある直径約 130mm 程度タイヤである。サイズ制限ギリギリの設計ではあるが、大きい草の多い路面では絶対に走破できないことが、今回を含み過去の多数の投下でも確認されていた。今年度能代宇宙イベントで利用したタイヤでは、走破性能を大きく向上させた、柔らかい衝撃吸収や伸縮可能な TPU 樹脂素材の 3D プリンタで凸凹に設計した直径 170mm まで拡張させたタイヤによって、大きい草を走破する必要がある。

実際のタイヤの画像を Fig.9-1-3 に示す。

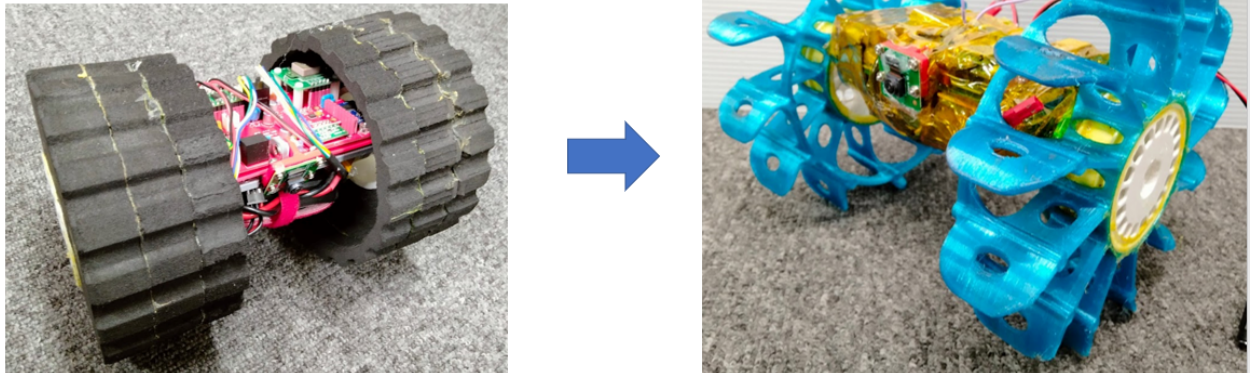


Fig.9-1-3 従来タイヤ(左)と TPU 拡張タイヤ(右)

また、走破に加えてタイヤが縦向きになる問題に対して、事前に対策できなかった原因として、実際の CanSat 投下会場に疑似した大きい草での実験ができておらず、起こりうる問題のリスクを考慮できなかったため、疑似した実験地を探すべきだった事が挙げられる。また、縦向きになった時の対処法として、タイヤの側面を丸くする事により、縦向きで静止せず横向きに倒れるような構造にする必要がある。また、TPU タイヤでの造形時間は 1 個当たり約 24 時間であり、切削時間を考慮したスケジュールでもっと早く完成を終え、より多くのパターンのタイヤの実験を行うべきだった。

次回以降では、CanSat 投下会場に疑似した環境での実験を行うこと、タイヤ縦向きの対策で側面を丸く設計する事、タイヤの造形時間を考慮したスケジュールにする必要がある。

【ACTS2 回目 考察】

サクセスクライテリア達成度

また、ACTS2 回目結果から、サクセスクライテリアの達成度を示した表を Table9-2 に示す。

ACTS2 回目の結果はミニマムサクセスである「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。」を達成した。しかし、フルサクセスの「ゴールから 5m 以内まで接近し、その後 Object Detection でゴールへ 0m まで近づきゴール判定をする。」は達成できなかった。

Table 9-2 ACTS1 回目結果サクセスクライテリア

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。	
フル サクセス	ゴールから 5m 以内まで接近し、その後 Object Detection でゴールへ 0m まで近づきゴール判定をする。	

原因追及

原因追及ではフルサクセスを達成できた要因を分析する。

1 回目投下では大きい草を走破することができない従来タイヤによりスタックしたが、拡張タイヤによりスタックせず、CanSat がゴールまで走行できた。拡張タイヤにより、従来の問題が解決できた。

また Object Detection ではゴールを認識するとき、疑似した物体を誤認識しないために、「パラシュート」「青コーン(会場範囲の仕切り用)」「気球」「人間」を別物体としてアノテーションし学習を行った。その結果、実際に能代宇宙イベントでは競技中に「人間」を認識し、ACTS2 回目では「青コーン」を認識し、別物体として認識し誤認識を防いだ。また、従来の 0m ゴール手法 Image Classification では CanSat チルト角度変化による認識がうまくいかなかったが、Object Detection によりどんな角度でも画像内のゴールを認識できるようになった。また、10m など距離の遠い場所からの認識率も高く、画像内に小さく映っているだけでも認識ができており、認識率を向上することができた。

実際の能代宇宙イベントで遠方の画面端でもゴール及び人間を認識した認識結果画像を Fig. に示す。また、実際の ACTS で遠方の画面端でもゴール及び青コーンを認識した認識結果画像を Fig. に示す。



Fig. 遠方から画像どの位置でも青コーン(黄)とゴール(水色)を認識している認識結果



Fig. 遠方から画像どの位置でも人間(オレンジ)とゴール(水色)を認識している認識結果

第10章 まとめ

1. 工夫・努力した点（ハード、ソフト、マネジメント面すべて）

スケジュール（マネジメント面）について

今回のスケジュールでは「予備審査で8割程度試験を行う」ことでスケジュールの余裕を持つことができた。スケジュールに余裕を持った結果、仮完成を大会の1か月前に終え、そこから新たな問題点が起こらないか徹底的に実験で調査することにより精度を高めることができた。また、1つ1つ部品を丁寧に組み立て故障を予防したり、プログラムの見直しによりバグの修正を行うことができた。

パラシュート離脱までについて

パラシュートは離脱までの動作で着実にできるよう設計した。

パラシュート生地はリップストップナイロンで、丈夫で硬く破れにくい。生地上を CanSat が走行しても絡まりにくい生地となっている。リップストップナイロンは軽く丈夫な生地でアウトドア用品やパラグライダーのパラシュートにも採用されている。

パラシュートの紐は金属ワイヤーを利用することで、あまり折れ曲がる状態にならず、紐が極力真っすぐになることで絡まりを防ぐ構造になっている。キャリア収納時には丸めて収納できる。また、生地との接着にアルミクランプカンでの強力な圧着をすることにより、接着部は今までの実験や大会で1度も故障したことがない構造になっている。固定する際は皮膚に触れないよう注意が必要になる。

パラシュート紐に金属メジャーを両面合わせて利用することで、強い力が加わらないと曲がらない紐となり、着地時に必ず真っすぐ落下し、CanSat 上にパラシュートが落下することができる構造となっている。キャリア収納時は折り曲げて収納するため金属疲労を考慮し、定期的に取り換えが必要である。

紐と CanSat のつなぎ目に pp シート(ポリプロピレン)を利用し、CanSat を覆うようにケースとして取り付け、CanSat 上に紐が落ちないようにし、pp シートケースを介して CanSat 離脱機構と紐を別々に取り付けることにより離脱機構への衝撃を緩和し、故障を防止している。また、ダクトテープをケースに貼り付けケース故障対策をしている。

実際のパラシュート画像を Fig.10-1 に示す。

また、走行開始後パラシュートを色画像認識し旋回させパラシュート絡まりを防いだ。

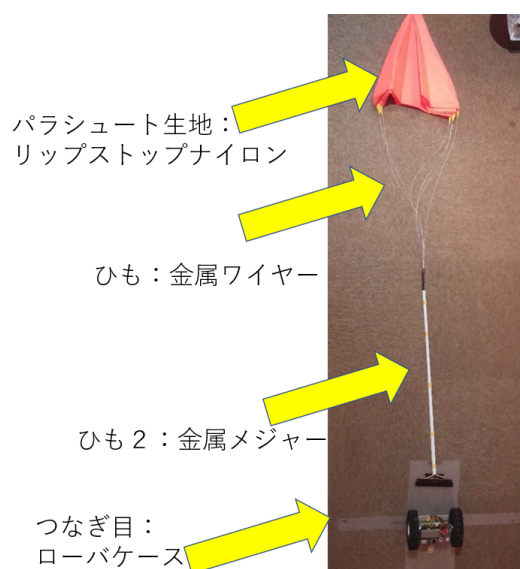


Fig.10-1 パラシュート

スタック防止 TPU タイヤについて

今回は走破性を重視した 3D プリンタ TPU 樹脂製の拡張タイヤを利用し、ACTS2 回目では荒れた草地を走破した。

TPU 拡張タイヤの性能

CanSat において従来サイズ制限ギリギリのタイヤを、最低地上高 50mm ほどの機体では走破できないことが確認できていた。そこで全く新しい構造の拡張タイヤであり、柔らかく衝撃吸収でき収納性がある TPU 樹脂製のタイヤを造形した。

タイヤの構造を考案する際は、まずギザギザのタイヤを設計したが、造形後に柔らかくありながら丈夫で硬い特性を確認し、ギザギザから斜め曲線に拡張させたような構造で走破性と収納性と衝撃吸収ができることを確認できた。設計段階から完璧ではなくても、実際にプロトタイプモデルを試作し、実物の監修から新たな機能を考案するきっかけを作ることができることを確認した。今後の設計考案にも応用できるようにする。

実際の TPU タイヤの設計画像を Fig. に示す。

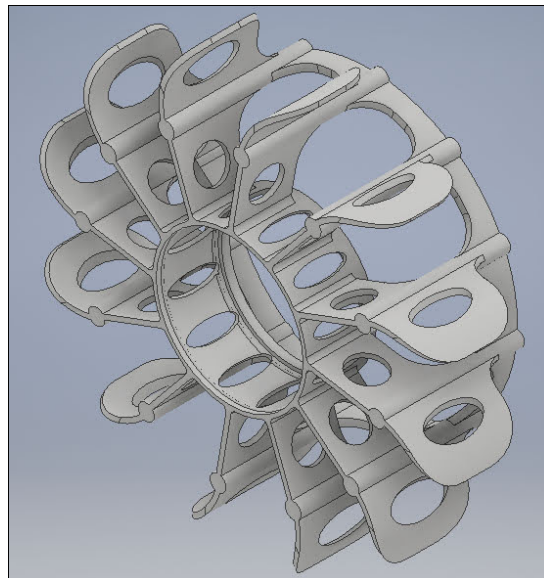


Fig. 拡張タイヤの設計

3D プリンタでの TPU 樹脂の利用方法

3D プリンタで柔らかい TPU 樹脂を利用する場合には TPU 樹脂に対応している型番のものを利用しないと樹脂の送り出しができない型番もあるので注意が必要である。自団体 ANYCUBIC MEGA PRO を利用したが、問題なく造形することが確認できた。

また、TPU 樹脂は造形時間を極力低速に設定する必要がある、今回利用したタイヤでは 1 個 24 時間の造形時間がかかった。1 セットに最低でも丸 2 日かかるため、造形時間を考慮した開発スケジュールが必要である。

また、TPU 樹脂を始め、3D プリンタによって綺麗に造形するためには温度や速度の設定がそれぞれ異なり、実際にテスト造形から最適値を調べる必要がある。

Object Detection 0m ゴールについて

Object Detection の 0m ゴールの認識精度の高さ

これまで 0m ゴール手法には、超音波センサや色画像認識や Image Classification(Deep Learning)による 0m ゴール誘導手法を取り入れてきた。超音波センサではノイズでうまく動作しない場合もあったり、色画像認識には日光の明るさ変化に色変化が大きく左右されることがや似た色の誤認識があったり、Image Classification ではゴールに似た物体を間違えたり、ゴール方向を画像から取得するために何度も認識して時間がかかったりする問題などがあった。

Object Detection では、日光の色変化には様々な日光状態のゴールを学習することで変化に対応し、同時に複数認識できるためゴールに似た物体も別に学習し識別でき、画像中のどこにあるかもわかるのでゴール方向が取得でき、Google Coral TPU で時間短縮ができる。

今回で Object Detection による 0m ゴールで問題点を克服した性能を確認できた。

Raspberry Pi Zero W での Object Detection の動作

また、Object Detection を Raspberry Pi で利用する場合では、CPU が ARM V8 以上の RaspberryPi で YOLO を利用するライブラリが多数存在しているが、マイコンのサイズや電力の最小化の観点から Raspberry Pi Zero W を利用し、速度短縮の観点から Google Coral TPU 非対応の YOLO でなく Mobile Net SSD v2 を利用することで、サイズと電力が短縮されながら、速度向上ができた。

Raspberry Pi Zero W の ARMv6 CPU で MoblieNet SSD v2 を動作させた 1 回の認識時間は 12 秒だった。従来の Image Classification では最大 10 秒と速度向上はできていないが、Google Coral TPU で計算させることにより 1 回 0.3 秒まで認識速度の向上ができた。

また、Google Coral TPU は公式には Raspberry Pi CPU ARMv8 以上には対応しているが、Raspberry Pi Zero W CPU ARM v6 には対応しておらず、公式のライブラリも存在しないが、非公式のライブラリからの Raspberry Pi Zero W 用のライブラリを利用し、エラーも起こりながらデバックしていき、Raspberry Pi Zero W と Google Coral TPU の動作を成功させた。非対応であることから、今後 API が無くなってしまっている場合もある。

学習について

Object Detection では、画像中の複数の物体を認識できるためゴール他にも、似た物体として写りうる、「人間」「パラシュート」「気球」「他の色(青)のコーン」も別物体として学習し、似た物体の誤検出の対策をした。

学習データとして 1 枚 1 枚の画像をマウスのドラックによりアノテーションを行うが、今回アノテーションを行った枚数は約 5000 枚と作業量が多くなるため約 1 ヶ月前後は学習データ収集のスケジュールを考慮する必要がある。また、長時間の単純作業ではあるが、開発で集中力が途切れたタイミングなどで休憩がてら作業を行うと効率の良いスケジュールでアノテーション作業を行うことができた。

学習回数は 1000 STEP で、PC スペックは Ryzen 7 1700(8 core 16 thread)でメモリ 64GB の学習で約 5 時間の学習時間で、学習結果は距離 5m まではほぼ 100%の高い認識率となっている。

2. 課題点

・縦向きタイヤの対策

今後はタイヤ側面を丸型に設計することにより、縦向きになっても横向きに動作を戻すことができるようにすること。

・Object Detection 誤検知の対策

今回はゴールの他に、「人間」「パラシュート」「気球」「他の色(青)のコーン」も別物体として学習し、似た物体の誤検出の対策をしたが、0m ゴール時の画像では画面1面がパイロン表面で暗くなっている事から、「空」を誤検知することがあるため、別物体として空の学習データを追加学習すること。

また、ARLISS のブラックロック砂漠走行をする際に、「車」などを誤検知しないために、別物体として追加学習すること。

3. 今後の展望

スケジュールについて、予備審査で8割程度試験を行う。大会1か月前程度で仮完成を目指し、調整や試験で精度を高める。

また、3D プリンタでの TPU 樹脂などの造形では造形時間が長時間であるため、考慮したスケジュールにすることや、Deep Learning での学習データ収集に考慮したスケジュールにする。

課題点で述べた改良を行うため、縦向きでスタックする対策にタイヤの形状の変更や、Object Detection の誤検知対策の学習データ追加を行う。

今回の実験結果から、TPU タイヤの設計の妥当性や Object Detection による 0m ゴール手法の妥当性について、CanSat の動作をデータから詳細に分析し、論文投稿を行う。

今後の CanSat の新ミッションでは、まだ考案段階で決定は出来てないが、プロトタイプモデルを早めに試作し、動作を確認する。