

ARLISS2022 大会報告書

提出日:2022年 11月 21日

- チーム情報

CanSatチーム名	電気通信大学 高玉研究室 Dolphins
CanSatチーム 代表者情報	戸板 佳祐 kawawa792526@gmail.com, 070-6962-3654
UNISEC団体名	電気通信大学 高玉研究室
UNISEC団体 学生代表	空閑智也 t_kuga@cas.lab.uec.ac.jp, 090-1268-2024
責任教員	高玉 圭樹 keiki@inf.uec.ac.jp, 042-443-5808
CanSatクラス	Open Class

- メンバー

役割	名前(学年, 経験年数)
リーダー 兼 回路班*	戸板 佳祐(M1, 経験 2年目)
ソフト班*	松田 尚也(M2, 経験 3年目)
ソフト班	石澤 竜希(B4, 経験 1年目)
ソフト班	原 亜美佳(B2, 経験 2年目)
ソフト班	堀越 健(B2, 経験 2年目)
ソフト班	中理 怡誠(M1, 経験 2年目)
回路班	空閑 智也(M1, 経験 1年目)
ハード班*	加藤 駿(M2, 経験 2年目)
ハード班	坂上 凜矩(B4, 経験 1年目)
ハード班	富澤 千香(B1, 経験 1年目)

- CanSatの製作目的・大会参加理由

弊団体はチーム開発を通して協調性を上げること, 開発に必要な技術力を身につけること, および研究室におけるCanSatに関する技術向上を目的にCanSat開発および大会への参加を実施している。

目次

ミッションステートメント	5
サクセスクライテリア	6
要求項目の設定	8
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)	8
3.2 ミッション要求	8
システム仕様	10
4.1 機体外観	10
4.1.1.親機	10
4.1.2.子機	13
4.2 機体内観・機構	15

4.2.1親機	15
4.2.2子機	21
4.3 システム図	23
4.3.1 親機	23
4.3.2 子機	26
4.3.3 使用電源	28
4.4 アルゴリズム	29
プログラム全体のフローチャート	29
試験項目設定	40
実施試験の内容	42
システム要求を満たすための試験内容	42
質量試験	42
機体の収納・放出試験	44
準静的荷重試験	47
振動試験	50
分離衝撃試験	53
開傘衝撃試験	54
パラシュート投下試験	58
長距離通信試験	61
通信機電源ON/OFF試験	62
通信周波数変更試験 (※安全面)	64
End-to-end試験	65
ミッション要求を満たすための試験内容	66
着地衝撃試験 (※ミッション面)	66
走行性能確認試験	69
電力耐久試験	70
反転横転復帰試験	71
親機・子機間通信確立試験	73
最良経路導出試験	73
子機経路評価試験	75
子機初期位置配置試験	76
子機数変動試験	77
指定経路走行試験	78
制御履歴レポート作成試験	79
ガントチャート(工程管理)	88
責任教員の所感	90

第1章 ミッションステートメント

ミッションステートメントを以下に示す。(ミッションの性質上「CanSat」という呼称を「親機」「子機」と書き分けている。)

ミッションの内容を以下に説明する。

親機が子機群を用いて広域未踏のエリアを安全に探索する手法の提案

現在の惑星探査、特に地表面の探査は単独機体での遂行が基本である。しかし、1機体での探査は非効率的であり、より安全な道のりの見落としによるミッションの失敗などの問題がある。このような観点から、我々は①複数の子機で親機から指定された領域*1の探査を行いながら経路および各地点の安全性を評価するためのデータ*2を収集し、②親機がこのデータから安全な探索経路*3を導出し、その経路を実際に走行する、という2つのフェーズを繰り返すことにより、親機が子機群を用いて広域未踏のエリアを安全に探索する手法の実現を目的とする。本目的達成のために、複数機体間における相互通信技術、各子機の探索により収集した安全評価データから最良の経路*4を導出する技術を確立し、ARLISSの場において有効に作用することを実証する。

*1指定された領域(以下、指定領域)とは、親機が探索を望む方位と親機・子機が通信可能な距離から生成した、親機が子機に探索を依頼する領域。(詳細は4章4節 アルゴリズム)

*2子機の加速度、GNSS座標から計算している。(詳細は4章4節アルゴリズム)

*3安全な探索経路とは、機体の突っかかりや反転・横転が生じる可能性のある障害を避け、比較的小さい振動で走行可能な経路を指す。(詳細は4章4節 アルゴリズム)

*4ここでの最良の経路とは、各子機の探索により収集した安全評価データから算出された親機が横転、反転、スタックをしづらいかつ走行による振動を最小限に抑えられる経路である。

ミッションの流れの概要を図1に示す。投下するのは親機のみであり、待機、放出後の落下、パラシュートの分離、そして探索を含むナビゲーション、それぞれの動作をシーケンスとして遂行する。なお、ミッション考案当初は親機と子機の両方を投下することを計画していたが、指定領域を効率的かつ十分に探索するためには多くの子機が必要であり、必要数に対し柔軟に対応することを考えた場合に子機は親機投下とは別のタイミングで投下されるのが適切であると判断したため、今回は親機のみでの投下とした。結果、子機は親機の着地後、その周辺に設置させることとする。その後は、1)ナビゲーションから子機が親機のもとに集合、2)各子機が親機から指定された領域・経路を探索、3)子機が走行中に収集した安全評価のためのデータを親機に転送して親機側で最良経路の導出、4)導出経路の走行、5)これらの手順を繰り返すことによって、仮の探索目標としてカムバックミッション用のゴールを目指す。また、シーケンスの具体的な動作は下記の通りである。

- I. 【待機シーケンス】: 親機はキャリアに収納され、上空で放出されるまで待機状態となる。親機はキャリアから放出後、光センサを用いて放出判定を行う。放出後にはパラシュートも開傘。
- II. 【落下シーケンス】: 9軸センサ、気圧センサを用いて親機が降下中か着地したかの状態判定を開始。
- III. 【パラシュート分離シーケンス】: 着地判定後、親機はパラシュートを切り離す動作を開始。サーボモータを複数回動かすことによりパラシュートを切り離す(パラシュート切り離し機構については3章を参照)。
- IV. 【ナビゲーションシーケンス】: 予め設定されたゴール地点のGNSS地点情報を取得し、その地点に向けて親機が安全に走行できると推定される最良経路を以下の手順で求める

1. 子機を親機の周辺に設置*5する。

2. 子機と親機間の通信が確立するまで待機する。
3. 子機が親機のもとに集合する。(子機初期配置誘導とする。具体的な配置関係については4.4節アルゴリズムを参照)
4. 親機が進みたい方向に対して指定した親機前方の長方形の領域を複数台の子機で探索し、走行中の座標と加速度ノルムやスタック判定履歴などの安全評価データを収集する。(子機探索命令とする)
5. 子機が収集したそれぞれの安全評価データを親機に送信し、親機が最良経路の導出を行う。(最良経路導出とする)
6. 導出した最良経路に沿って親機を走行させる。(親機・子機移動とする)
7. 以上の手順のうちに、手順3.～6.をゴールにたどり着くまで繰り返し行う。

*5子機の設置に関しては、親機の着地以降で親機の着陸地点を発見したときに、人の手によって親機から15 [m]程度離れた周辺に設置する。

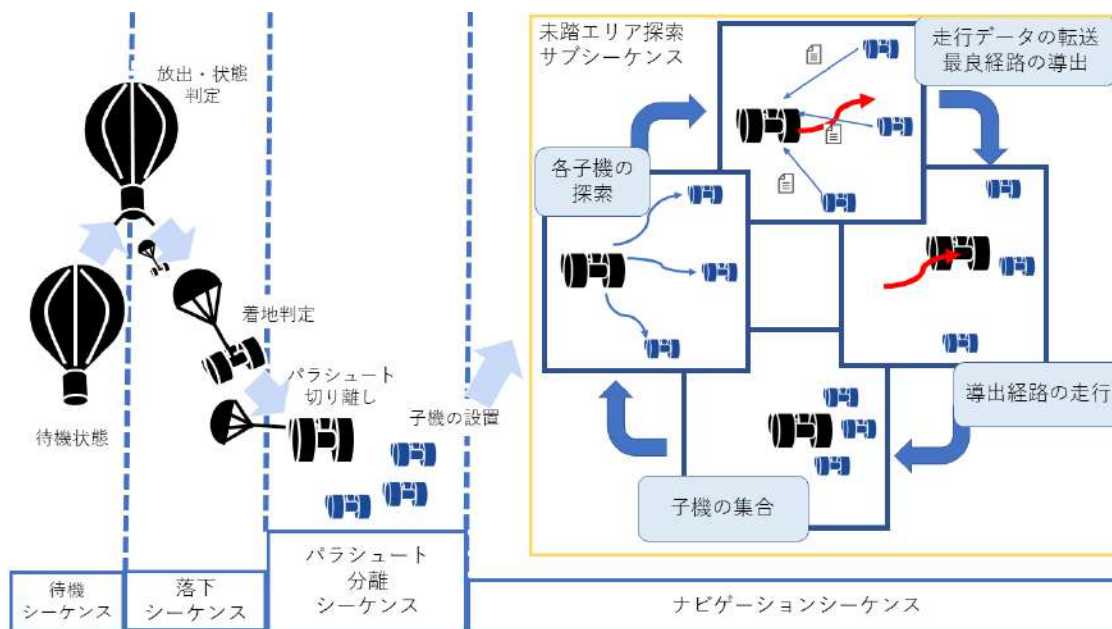


図1 ミッションの流れの概要

第2章 サクセスクライテリア

ミニマムサクセス	親機着陸後、各子機との双方向通信を確立し、各子機が最良経路探索に向けて走行を開始する。
フルサクセス	以下の4つの項目を順に達成することにより、フルサクセス達成とする。 1. 各子機体が親機からの指定領域を走行後、親機に走行データ*6を転送する

	<ol style="list-style-type: none"> 2. 親機は子機体から受信した走行データをもとに指定領域での最良経路を算出 3. 導出した最良経路を親機が完走する 4. 子機が親機のもとに集合する
アドバンスドサクセス	フルサクセスの1フロー($X=100$ [m]) x ($N=5$)セット*7を導出した経路に沿って親機が走行する。その際、子機が走行困難または走行不能と判定した地点は確実に避けるようにする。

*6加速度センサz軸値(起伏判定), GNSSセンサおよびジャイロスコプ(スタック・反転判定)

*7X: 機体同士が通信可能な距離かつ、用意した子機数で十分に探索可能かつ、それぞれの子機の状態確認のために追従するメンバー同士のやりとりが可能な距離を考慮し決定、N: 電力耐久試験の結果に応じて設定

また全てのサクセスクライテリアの評価方法は以下のように設定する

ミニマムサクセス	親機と各子機間で通信確立および走行開始のコマンド文字列を送受信したことをログで確認する。親機と各子機が正常に受信した文字列および各子機が走行開始したことを目視で確認する。
フルサクセス	<ol style="list-style-type: none"> 1. 指定した区間を走行した全ての子機が走行データを親機に転送していることをログで確認する。 2. 親機が子機群の走行データから最良経路を算出していることをログで確認する。 3. 親機が導出した経路と常に10 [m]以内*8の距離を保ちながら走行していることを確認する。 4. 親機が指定した子機の集合座標10 [m]以内*8に走行可能な子機が全て集まることを確認する。
アドバンスドサクセス	<ul style="list-style-type: none"> ・親機が着地した地点からの走行距離が500(=100x5) [m]以上 ・子機が走行困難となった地点のログおよびその地点を親機が回避した経路を導出し、更に走行しなかったことが分かるログ

*8GNSSセンサの誤差を考慮

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	S1-9の充足を確認した設計のCanSatによって, 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, 安全性に関わる大幅な設計変更はない

3.2 ミッション要求

番号	ミッション要求項目
M1	着地時の衝撃荷重によって, ミッションを実現するための機能が損なわれていないことが試験で確認できている
M2	親機と子機が環境の悪い地面に置ける走行性能を試験で確認出来ている

M3	ミッション達成に必要な時間を十分に走行可能であることを試験で確認できている
M4	親機と子機が反転・横転した時に通常の走行姿勢に復帰できることを試験で確認できている
M5	各子機体と親機の通信が確立できることを試験で確認できている
M6	親機・子機が同時に走行時、経路が被っているなど衝突可能性がある際に衝突を回避するような走行をすることを試験で確認できている
M7	子機体が自身の走った経路について評価・記録できることを試験で確認できている
M8	各子機体が複数経路を走行し、走行データを親機に反映後、親機は最良経路を導出することができることを試験で確認出来ている
M9	親機の最良経路走行後において、次の経路探索に向けて子機が親機の付近に集合できることを試験で確認できている
M10	着地後にパラシュートを分離できることを試験で確認できている
M11	子機からの通信が途絶えた場合、あるいは、走行が不可能である場合、残りの走行可能な機体を認識し、その機体数に応じてミッションを続行できることと通信が途絶えた子機についてこれ以上の動作を行わないよう停止できることを試験で確認できている
M12	親機、子機それぞれが決められた経路を走行できることを試験で確認できている
M13	ミッション時に人間が介在しない自律制御を実施することを確認出来ている
M14	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

第4章 システム仕様

4.1 機体外観

本ミッションでは、主要機体の安全な広域探査実現に向けた実証実験のため、それぞれ役割が異なる『親機』と『子機』を用いる。両機体は以下の点に留意して設計を行った。

- 親機：探査の根幹を担う重要な機体であるため、走行性能を高めることで安定した走行が可能な機体にする必要がある。そこで、可能な限り機体幅を大きくすると共に機体の重心を調整することで、横転・反転が起きるリスクを可能な限り低減する。
- 子機：親機が安全に探査を進めるためのサブ機としての役割を果たす。親機の経路探索のためのデータを取得する必要があるため、軽量化することで敏感に起伏を感知し、小型化[†]することで小さいトルクでも軽快な動きを可能にする。

[†]：機体の小型化により、従来の機体では走行出来ていた所が走行できなくなるリスクが高まることが考えられる。しかし、本ミッションではこうした走行不可能となった機体をもとに環境を評価したり、走行不可能となった機体を切り離してミッションを続行することが可能となっている。これより、機体の小型化に関するリスクについては十分に考慮しており、これらのリスクに対して講じている対策も本ミッションの醍醐味の一つである。

4.1.1.親機

親機の正面を撮った画像を図4.1.1.1に示す。回路基盤下の緑の実線で囲まれた部分にOrange Piを起動するためのリチウムポリマー電池が設置されている。また、胴体下部の赤色の実線で囲まれた部分に駆動輪及びサーボモータ用の乾電池が装着されている(詳細は4.2.1(機体内観・機構 - 親機)にて記述する)。親機上面からの画像を図4.1.1.2、親機右側面からの画像を図4.1.1.3、俯瞰図を図4.1.1.4に示す。図4.1.1.2の機体上部中央にある回路基板に搭載したセンサ・モジュールについては4章3節1項のシステム図にて詳細を後述する。また、図4.1.1.5中の赤色の実線で囲まれたスタビライザが地面に接地することで機体が反転することを防ぐと共に機体のバランスを取っている。このスタビライザはサーボモータで制御することができ、図4.1.1.6のように機体下部に折りたたむことでレギュレーションを満たした状態でキャリアに収納することができる。また、先端に半径10 [mm]の円柱型スポンジをクロロプレン製の布でコーティングしたものを接着することでスタビライザに直接かかる衝撃を緩和している(スタビライザの詳細は4.2.1に記載している)。さらに、タイヤには走破性を向上させるため、木片(5 [mm]×5 [mm]×5 [mm])をホットボンドにより接着している。なお、炎天下の中、3時間に及ぶ電力耐久試験を終えてもホットボンドが溶けることは確認されなかった。

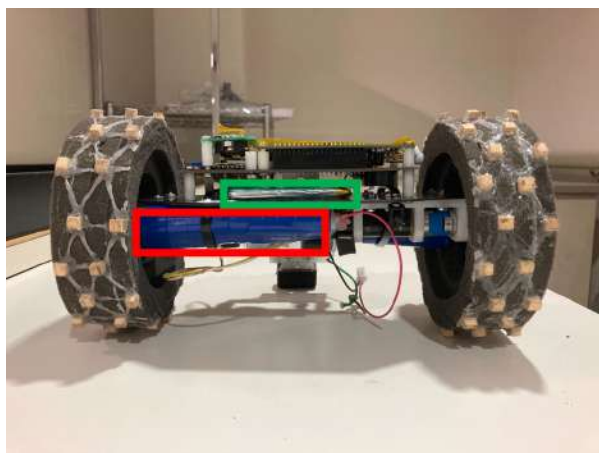


図4.1.1.1 親機(正面)

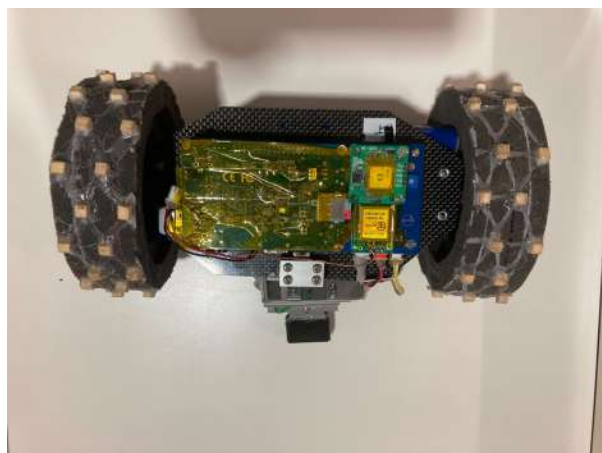


図4.1.1.2 親機(上面)

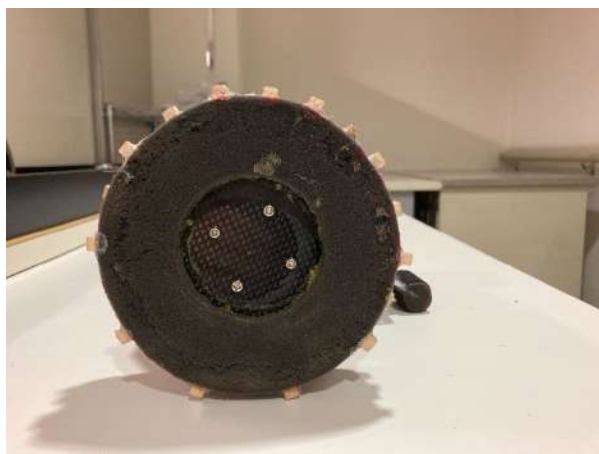


図4.1.1.3 親機(右側面)

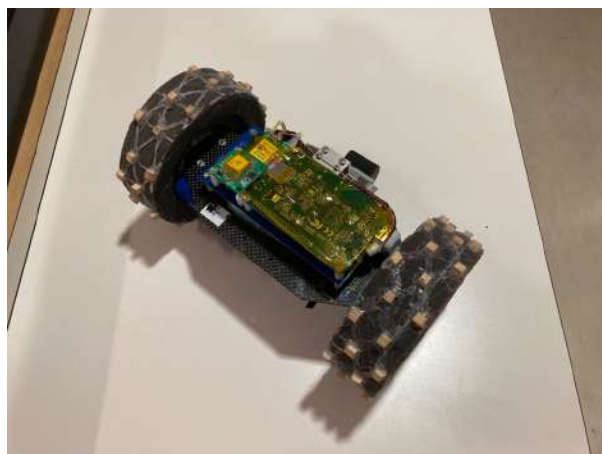


図4.1.1.4 親機(俯瞰)

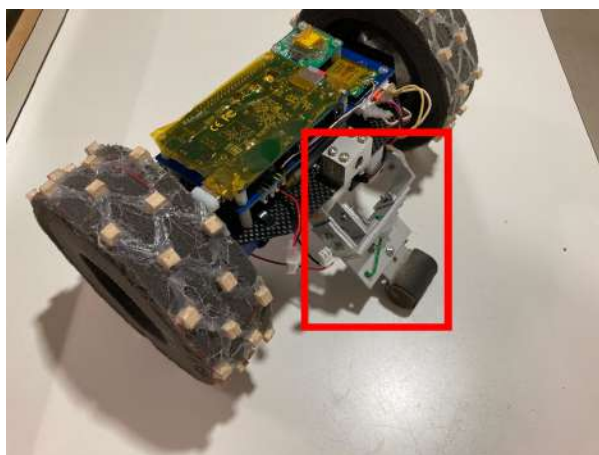


図4.1.1.5 親機(スタビライザ展開)

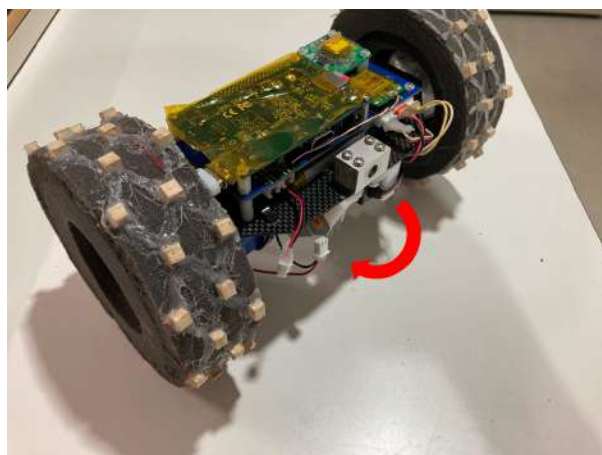


図4.1.1.6 親機(スタビライザ収納)

親機の機体幅の計測のために親機の正面にメジャーをあてた画像を図4.1.1.7, 親機の高さを計測するために親機の側面にメジャーを当てた画像を図4.1.1.8, 親機の全長を計測するために親機の側面に地面と平行にメジャーを当てた画像を図

4.1.1.9, 親機の質量を計測した画像を図4.1.1.10に示す。また, 表4.1.1に親機のサイズならびに質量についてまとめる。実際にレギュレーションを満たしていることは7章の質量試験および収納放出試験の結果を参考にされたい。

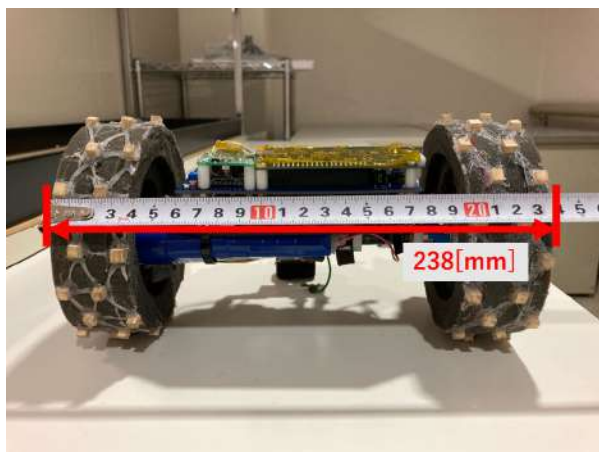


図4.1.1.7 親機の機体幅測定

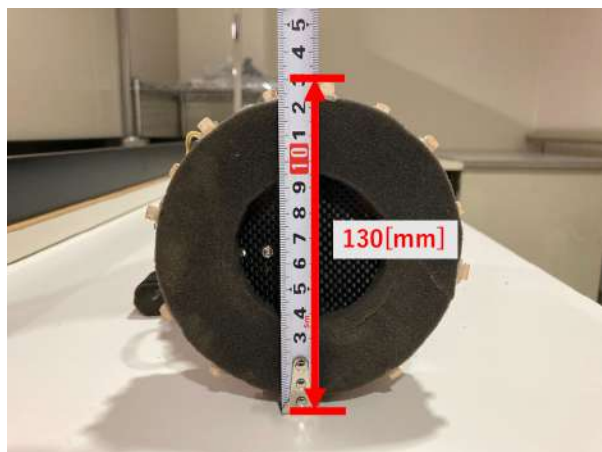


図4.1.1.8 親機の高さ測定



図4.1.1.9 親機の全長測定



図4.1.1.10 親機のための質量測定

表4.1.1 親機サイズ

機体幅 [mm]	238
高さ [mm]	130
全長 [mm]	153
質量 [g]	783

4.1.2.子機

子機を正面から撮った画像を図4.1.2.1に示す。胴体下の中央の赤色で囲まれた部分に前輪駆動モータ用の単4電池が機体水平方向に2本ずつ2組設置されている(詳細は4.2.2(機体の内観・機構 - 子機)にて記述する)。子機上面からの画像を図4.1.2.2, 子機右側面からの画像を図4.1.2.3に示す。図4.1.2.2の機体上部中央にある回路基板に搭載したセンサ、モジュールについては4章3節2項のシステム図にて詳細を後述する。また、同図の緑の破線で囲まれた部分には回路基盤の下にRasberry Pi Zeroを起動するためのリチウムポリマー電池が装着されている。さらに、胴体下部から下側には、親機同様の役割を果たすスタビライザーが固定されている。図4.1.2.3の前輪はスポークに竹(2 [mm]×5 [mm]×35 [mm])を8本用いて軽量化をはかると共に、木片(親機と同様のもの)をホットボンドで接着することで走破性を向上させている(車軸部分については子機の機体内観で詳しく説明する)。最後に、俯瞰図を図4.1.2.4に示す。

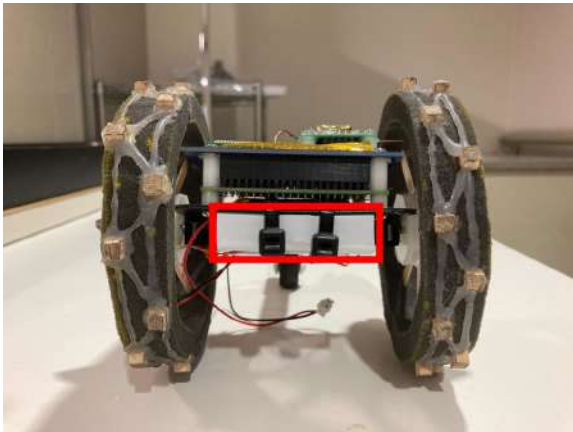


図4.1.2.1 子機(正面)



図4.1.2.2 子機(上面)

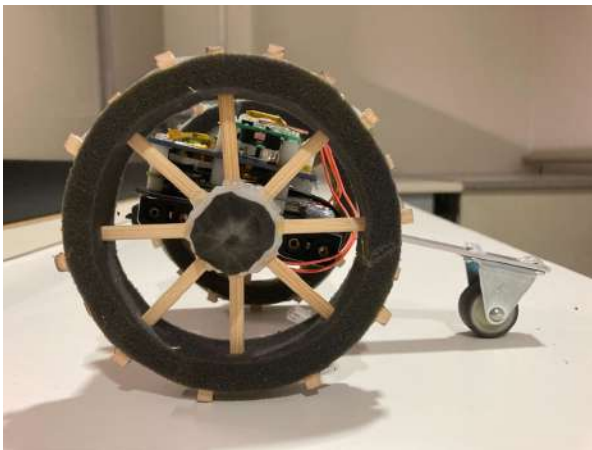


図4.1.2.3 子機(右側面)



図4.1.2.4 子機(俯瞰)

子機の幅の計測のために子機の上面にメジャーを当てた画像を図4.1.2.5, 子機の高さの計測のために子機の側面にメジャーを当てた画像を図4.1.2.6, 子機の全長

を計測するためにスタビライザを含めてメジャーを当てた画像を図4.1.2.7, 子機の質量を計測した画像を図4.1.2.8に示す. また, 表4.1.2に子機のサイズならびに質量をまとめる.

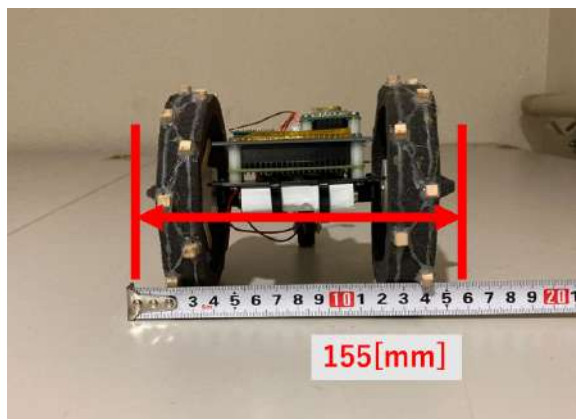


図4.1.2.5 子機の機体幅測定

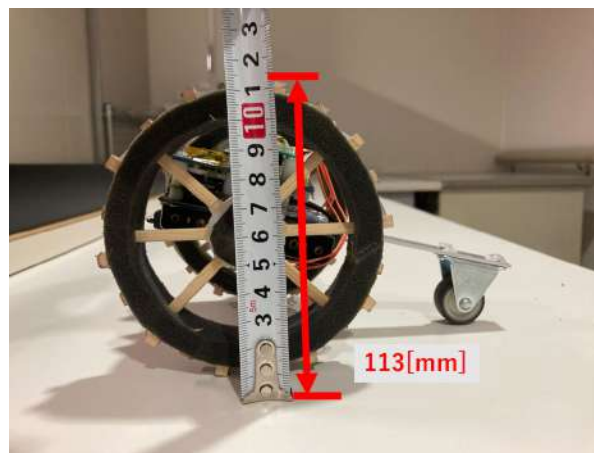


図4.1.2.6 子機の高さ測定

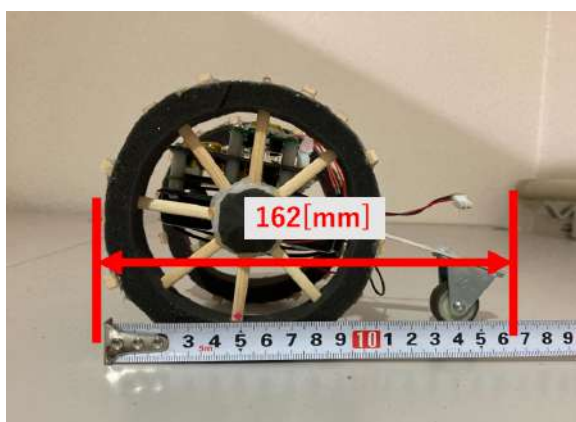


図4.1.2.7 子機の全長測定



図4.1.2.8 子機の質量測定

表4.1.2 子機サイズ

機体幅 [mm]	155
高さ [mm]	113
全長 [mm]	162
質量 [g]	280

4.2 機体内観・機構

4.2.1 親機

- モータ電源

図4.2.1.1に親機を裏面から撮った画像を示す。赤の実線で囲まれた部分に駆動輪及びサーボモータ用の単3乾電池6本(進行方向前方側4本, 後方側2本)が装着されている。図のように機体前後に分離して配置することで, 機体前方に6本をまとめて配置するよりも, 機体の重心位置が機体の進行方向前方に偏り過ぎないように調整している。

- スタビライザ

図4.2.1.2と図4.2.1.3に親機のスタビライザの画像を示す。スタビライザはL字型及び平板の金属を図4.2.1.4に示すように組み合わせて形成している。

図4.2.1.3の中央付近赤色で囲まれた部分にスタビライザを制御するためのサーボモータを装着した。また, パラシュートピン装着をスムーズに行うためにスタビライザを胴体に固定するポリアセタール同士の距離を離すように配置した(図4.2.1.4)。これにより, ポリアセタール間に見える穴へパラシュートピンを固定するための釘の装着を容易にしている。

最後に, スタビライザL字部分は折り畳み時にタイヤ駆動用モータに接触する。着地衝撃時等では, これらが衝突しモータの破損に繋がる。そこで, 図4.2.1.6の緑の実線で囲まれた部分のように金属L字とモータが接触する部分にスポンジを貼り付けることで衝撃を緩和する。

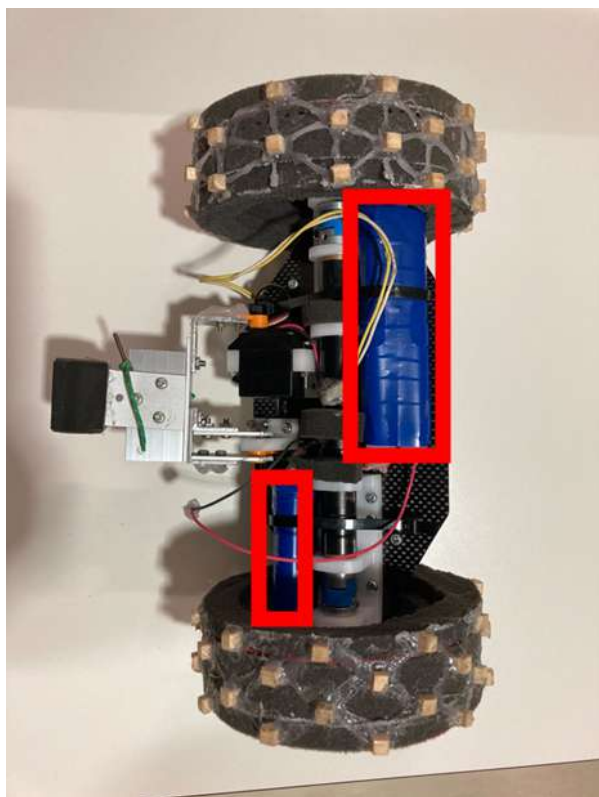


図4.2.1.1 親機(裏面)

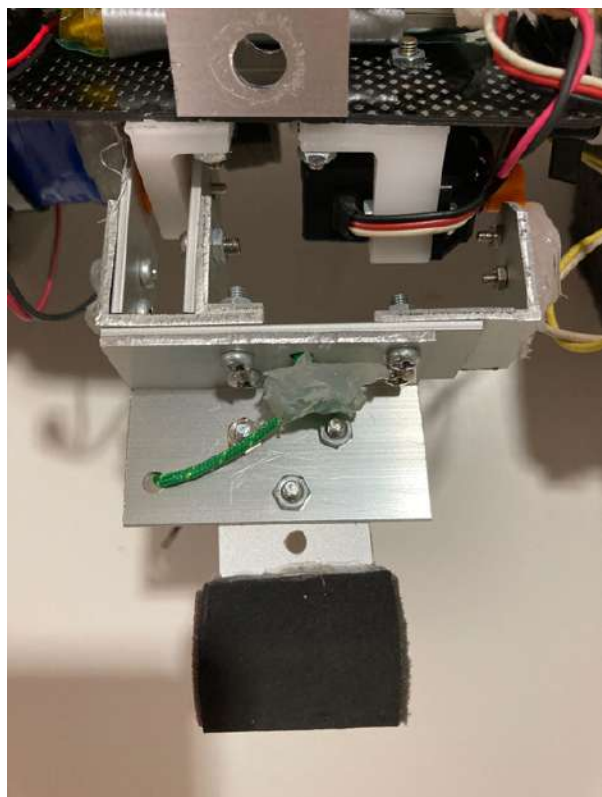


図4.2.1.2 親機スタビライザ(上面)

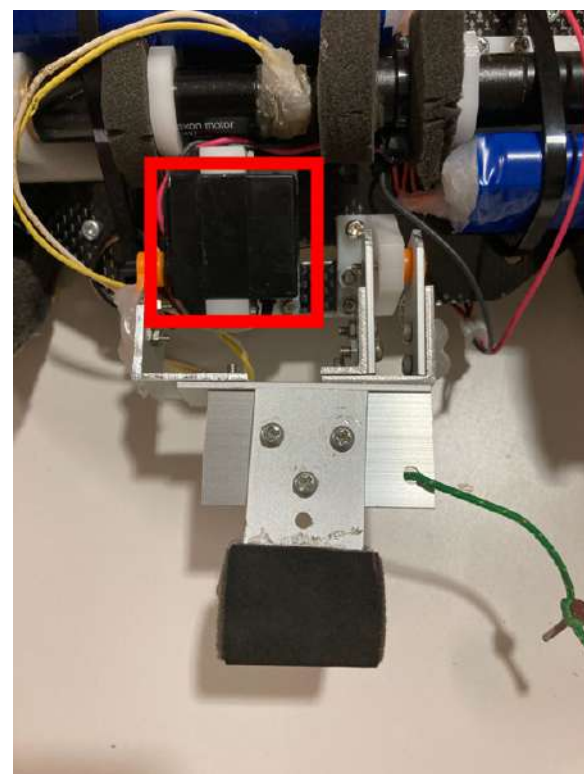


図4.2.1.3 親機スタビライザ(下面)

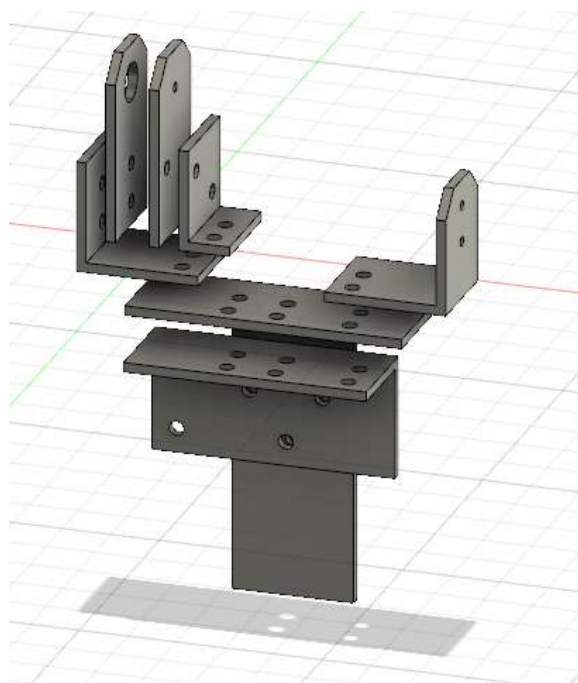


図4.2.1.4 親機スタビライザ(設計図)

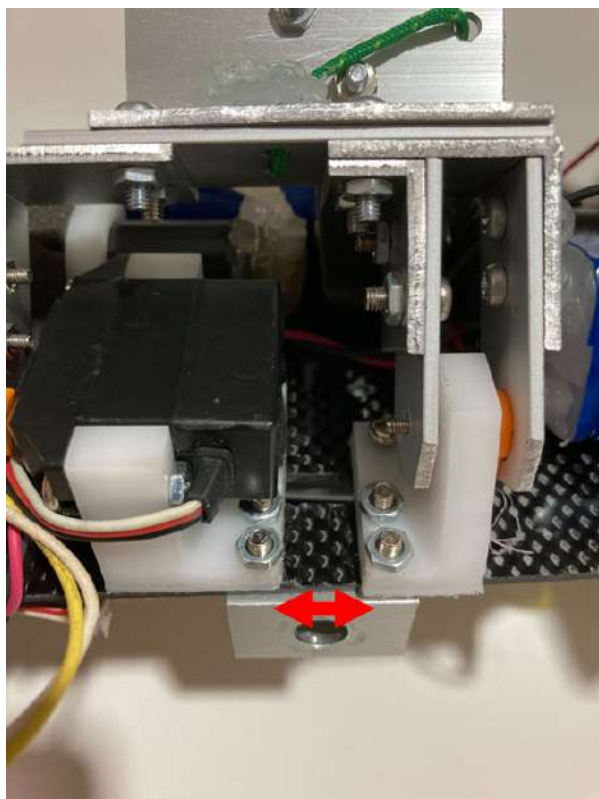


図4.2.1.5 ポリアセタールを離して配置

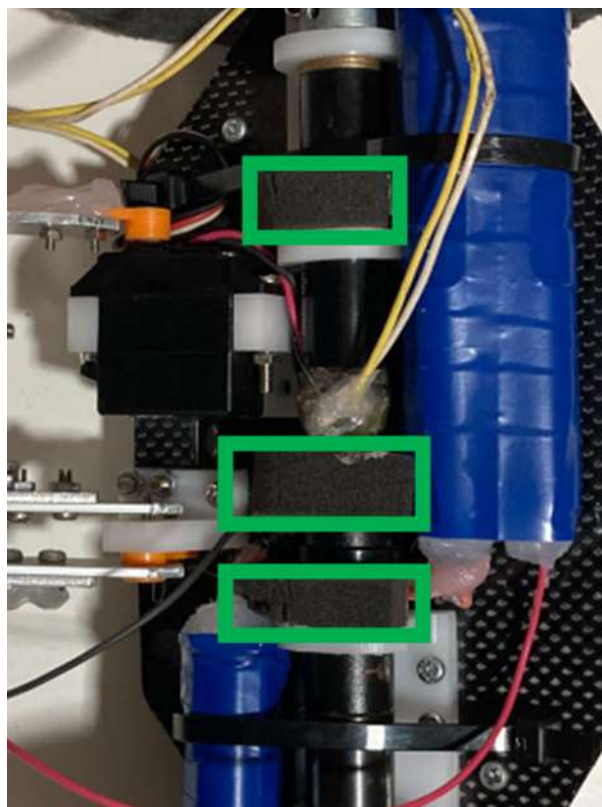


図4.2.1.6 衝撃緩和スポンジ

- パラシュート分離機構

図4.2.1.7, 図4.2.1.8は親機のパラシュート切り離し機構の画像である。図4.2.1.7はパラシュートが親機と接続された状態, 図4.2.1.8はパラシュートが切り離された状態を示す。なお, パラシュートは図4.2.1.7の橙色の紐の先に接続されている。

ここにスタビライザ
に着いた緑色の紐の
先の釘を差し込むこ
とで固定されている

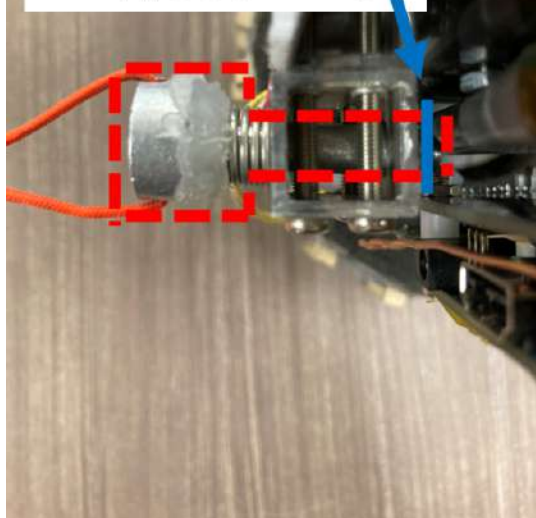


図3.2.1.7 パラシュート分離機構(装着済)



図4.2.1.8 パラシュート分離機構(未装着)



図4.2.1.9 パラシュートを接続するピン

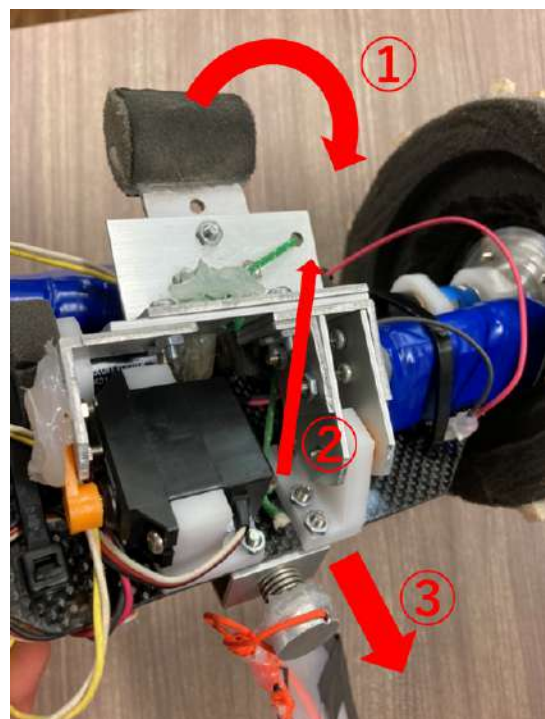


図4.2.1.10 パラシュート分離の流れ

パラシュートと親機を接続するピンは図4.2.1.9で示されるように、先端付近に穴が開いており、ここに釘を差し込むことでピンと親機が接続された状態(図4.2.1.7)を維持している。図3.2.1.10にはパラシュートが分離する流れを示す。①親機下面に固定

されているサーボモータを駆動させると、スタビライザーが同図の矢印方向に開く。②緑色の紐がスタビライザーの動きによって引っ張られることで釘が外れる。③パラシュートのピンが親機から切り離される。ピンには切り離しをスムーズにするためにバネが設置されており、釘が外れた際にはこのバネが伸びることで親機からピンが勢いよく切り離される。

なお、この設定は経験上、3秒間サーボモータを上下に駆動し、親機を前進、という動作を1, 2回程度繰り返すことでピンが切り離されることが分かっている。フローチャートではより高い確率でピンが抜けるよう、上記動作を10回繰り返すこととしている。10回という数字は、不確実性のある動作の確実性を上げるために、動作が成功すると期待される試行回数に安全率Sをかけた回数を設定するという機械工学の考えに基づき、鉄などの金属部品の繰り返し動作における安全率S=6をかけた値として設定した。

ここで、パラシュートピンがバネの弾性力によって放出されることを数値計算によって確認する。パラシュートピンがばねの弾性力によって放出される条件は、下式の条件で表される。ここで、 k はバネ定数、 E は縦弾性係数である。なお、今回使用しているピンの材質はばね鉄鋼材であり、JIS B 2704によれば $E = 206 \times 10^3$ [N/mm]である。

$$k < E$$

ここからは、今回採用したばねがこの条件を満たすことを確認する。まず、実際にパラシュートピンに使用しているばねのばね定数 k は次の関係式をもとに求めることができる。なお、現実的には摩擦のエネルギー分が大きく影響することが考えられる。しかし、実験を通して繰り返しばねをスムーズに切り離すことに成功しており、ばねの選定が適切であると考えられるため、今回は摩擦エネルギーの議論を省略する。左辺はばね定数の弾性エネルギー、右辺はばねの位置エネルギーの差分を示す。

$$\frac{1}{2} k x^2 = m g \{h - (l - x)\}$$

ここで、 l はばねの自然長 [mm]、 h はピンの長さ、 x は設置時のばね先端からのばねの変位 [mm]である。図4.2.1.11、図4.2.1.12は、これらの寸法を実際の画像上に示したものである。

実測より、 $h = 30$ [mm]、 $l = 18$ [mm]、 $x = 13$ [mm]、ピンの質量 $m = 8$ [g]であった。なお、重力加速度 $g = 9.8$ [m/s²]とする。これらの値を上式に代入すると、 $k = 23.2 \times 10^3$ [N/mm]と求められる。従って $k < E = 206 \times 10^3$ [N/mm]であり、最

初に述べたパラシュートピンがばねの弾性力によって放出される条件 $k < E$ を満たしていることが確認された。

また、パラシュート切り離し時に釘に加わるトルクはサーボモータから発生する53 [N]($\div 5.4$ [kgf])である(サーボモータのデータシートを参照)。また、JIS規格より釘の素材であるSUS304を参照すると釘の強度 σ は205 [N/mm²]であることが示されている。ここで、サーボモータの回転によって釘が引っ張られる際に力が加わる面積 S はサーボモータと釘を繋ぐ紐の釘に対する接地面の半分(力が加わる方向と真反対の部分)であるので、釘の胴径 $2r$ が1.25 [mm]、紐の幅 w が2 [mm]であることを踏まえると以下の式より求められる。

$$S = \frac{2\pi rw}{2} = 3.14 \times 0.625 \times 2 = 3.925[\text{mm}^2]$$

以上よりパラシュート切り離し時に加わることが想定される釘の面への力 σ' は

$$\sigma' = S\sigma = 3.925 \times 205 = 804.6[\text{N}]$$

となる。従って、使用する釘はパラシュート切り離し時に加わるトルクに対して変形することなく耐えることが可能であり、パラシュート切り離し機構に採用するものとして適切である。

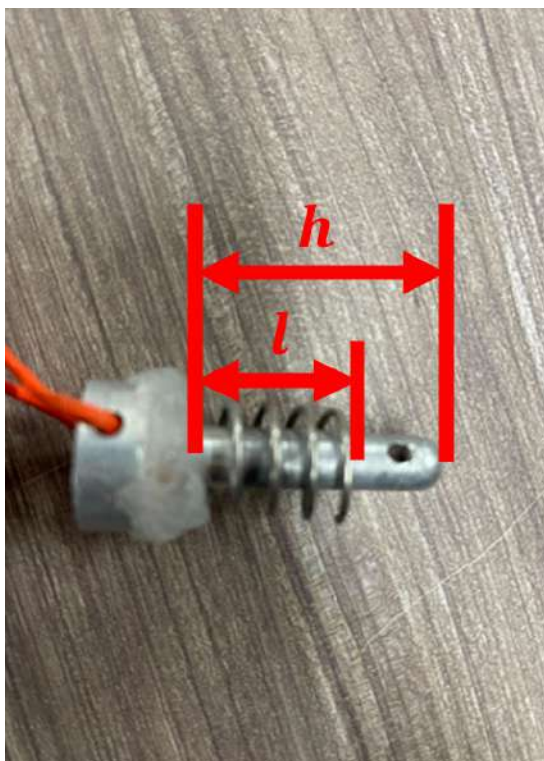


図4.2.1.11 ばねとピンの寸法

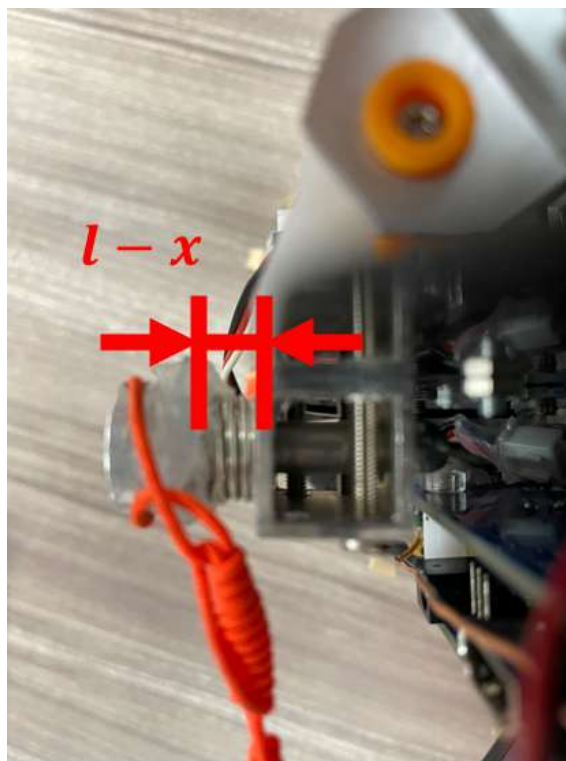


図4.2.1.12 縮めたときのばねとピンの寸法

4.2.2 子機

- モータ電源

子機裏面を図4.2.2.1に示す。画像中の赤色で囲まれた部分に前輪駆動モータ用の単4電池が2本ずつ2組設置されている。

実験により、モータの出力について十分な性能を発揮できておらず、回転数が少ないことが確認できた。これは走行・復帰性能に関わる問題で探索性能を低下させる要因となり得る。そこで、十分な電圧を確保するため、図4.2.2.4に示すように機体後方の電池ボックスに収納されている電池の数を2本から3本に変更した。

- 車軸部分

車軸は図4.2.2.2に示すような2種類の部品(材質:ポリアセタール, 3D切削加工により製作)で同図中のカップリング(一部)を挟みこむことでモータに直結している。また、車軸側面にはスポークを挿し込むための溝ができるようになっている。また、モータを固定するためのイモネジを操作するための穴が図の上部に空いている。

- スタビライザ

スタビライザを図4.2.2.3に示す。スタビライザにはアルミニウム板(15 [mm] x 120 [mm] x 2 [mm])を採用した。先端には振動を抑制するために車輪を装着している。この車輪は、サーボモータを採用していないために懸念される反転復帰の問題への対策として、重心を後方にずらす役割も担っている。

実験により、スタビライザの先端に取り付けられている後輪が一方向にしか回転で

きないことから、機体が急な回転を行ったときに取得するデータにノイズが乗ってしまうことが確認できた。これは親機が最良経路を導出する際のデータの精度を下げる要因となり、ミッションの成否に関わると考えられる。そこで、図4.2.2.5に示すように後輪を360度回転可能なものに変更した。当日は、現地実験の結果に応じて使用する後輪の種類を決定する。

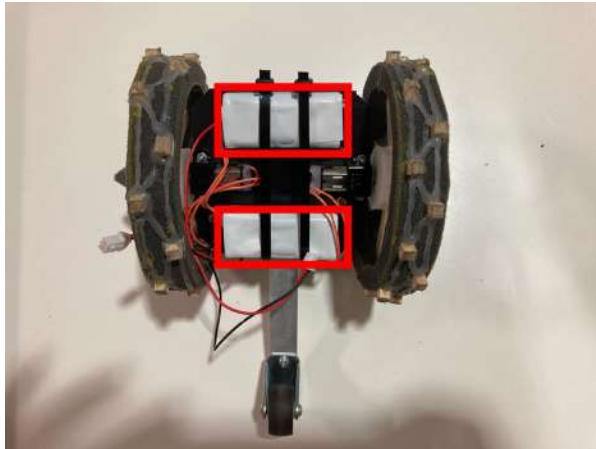


図4.2.2.1 子機裏面(変更前)

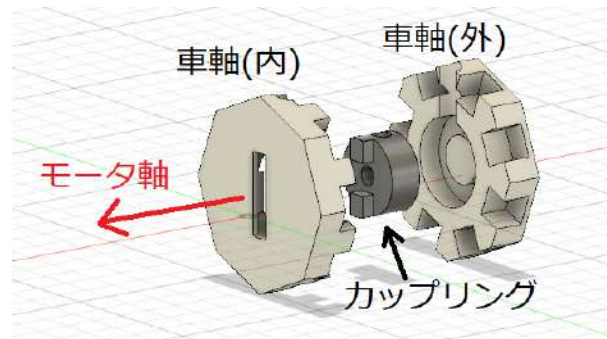


図4.2.2.2 車軸

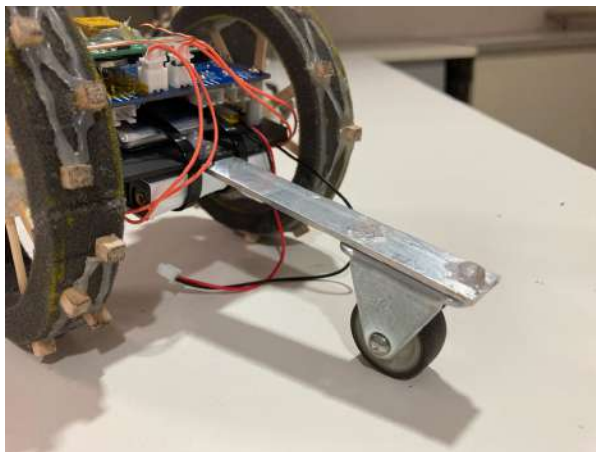


図4.2.2.3 子機スタビライザ



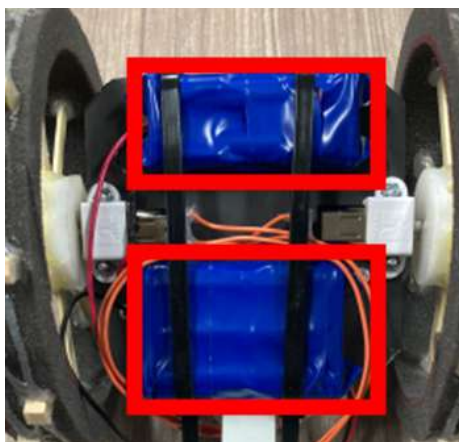


図4.2.2.4 子機裏面(変更後)



図4.2.2.5 子機スタビライザ
(ボールベアリング付キャスタ)

※なお、子機は上記変更を施した場合、32[g]増加し、総重量は280[g]から312[g]となる。

4.3 システム図

4.3.1 親機

図4.3.1に親機を制御するシステムの構成を示す。親機には9軸センサ(加速度, ジャイロ, 地磁気)と気圧センサ, 光センサ, GNSSセンサが搭載されており, これらのセンサから得た情報からモータやサーボモータの動作を制御する。また, LoRaモジュールを通して機体間の通信はこのモジュールを通して行う。親機の位置情報を地上局に適宜送信する。使用する各センサやモジュールの名称および使用目的を表4.3.1にまとめ, その詳細を以下に示す。

また, 本ミッションでは, 環境情報を取得する各センサ(入力)について, 以下のように用いる。(具体的な値の利用方法については4.4節ミッションシーケンスおよびアルゴリズムの方で記述する)

- GNSSセンサ
 - 自己位置推定
 - 取得したGNSS座標を用いて自己位置座標を更新
 - ナビゲーション処理
 - ゴール座標*9と毎回取得するGNSS座標の差分から進行方向を調整
 - スタック判定
 - 毎回取得するGNSS座標の差が一定以内ならばスタック状態と判定
- 9軸センサ
 - 着地判定
 - 落下中と着地後の加速度が異なることを利用
 - ナビゲーション処理
 - 地磁気を用いた進行方向の調整
 - 横転・反転判定

- 重力の方向から、親機の状態を把握
 - 姿勢制御
 - 重力の方向から、走行時の姿勢を制御
- 気圧センサ
 - 着地判定
 - 落下シーケンスに気圧が大きく変化し、着地後は気圧変化が小さいことを利用
- 光センサ
 - キャリア放出判定
 - ロケットのキャリアから放出されたことを判定
- LoRa
 - 機体間の相互通信
 - 投下時ロスト対策
 - GNSSセンサが取得した自己位置座標を定期的に発信、目視で親機と通信可能な範囲まで追従した後に基地局LoRaで受信

*9”ゴール座標”とは、仮の探索目標であるカムバックミッション用のゴールのGNSS座標のことを指す(4.4章で後述).

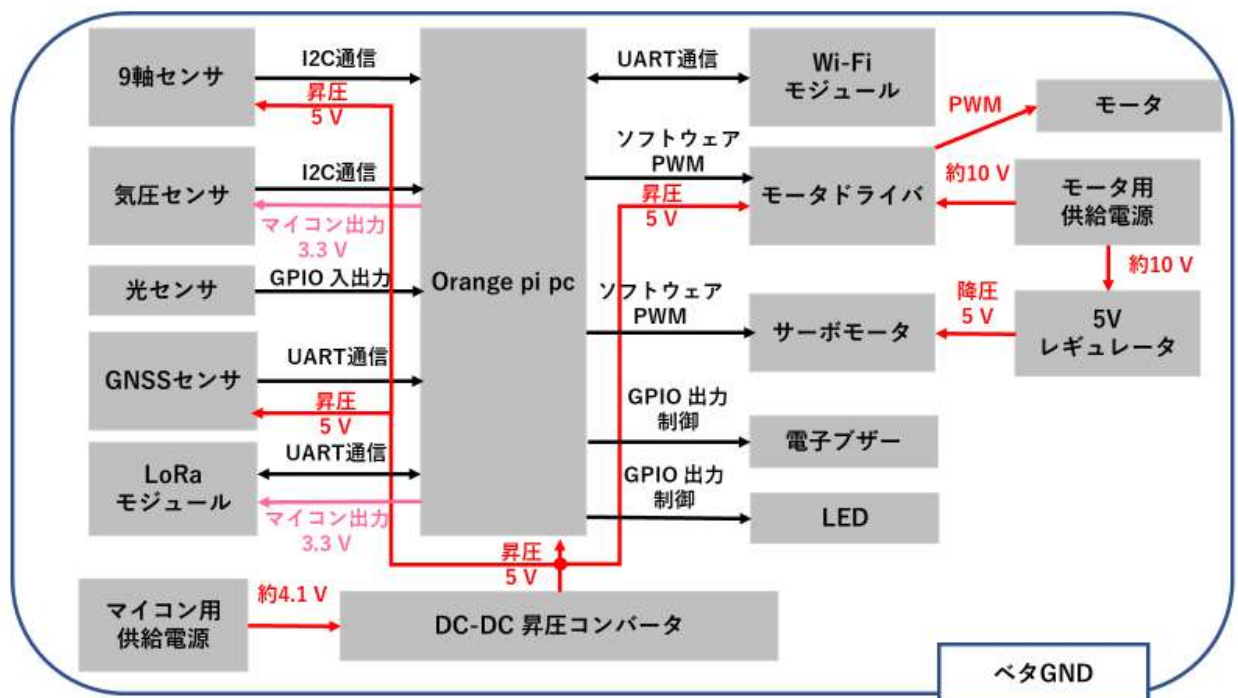


図4.3.1 親機のシステム図

表4.3.1 親機搭載計器使用

分類	名称・型番	使用目的	入手先URL・参考情報等
マイコン	Orange pi pc	入出力値の計算, ログデータ	https://ja.aliexpress.com/item/10

		の記録	05001689152635.html
GNSSセンサ	AE-GYSFDM AXB	ゴール地点へ の方向・距離 の取得, スタッ ク判定 (GPS(米国), QZSS(日本)に 対応)	https://akizukiden shi.com/catalog g/gK-09991/
モータドライ バ	DRV8835	モータの出力 制御	http://akizukiden shi.com/catalog/ g/gK-09848/
モータ	DC motor RE EBCLL 3.2W SL 2WE	タイヤの回転	https://www.max ongroup.co.jp/m axon/view/produ ct/motor/dcmoto r/re/re16/11869 9
サーボモータ	GWSMIRMG FA S03T/2BBM G/J	パラシュート切 り離し, 親機の 姿勢が水平に なるように制 御	http://akizukiden shi.com/catalog/ g/gM-01908/
モータ用供 給電源	Energizer L91Ultimate Lithium AA Batteries	駆動系(サーボ , モータ)への 電源供給	https://www.wal mart.com/ip/Ene rgizer-Ultimate- Lithium-AA-Batt eries-12-Pack-D ouble-A-Batterie s/139060065
マイコン用供 給電源	DTP605068(PHR)	マイコン及びセ ンサ類への電 源供給	https://www.mar utsu.co.jp/pc/i/8 36419/
DC-DC昇圧 コンバータ	TPS63070	マイコン用の 供給電圧を5 [V]に昇圧	https://strawberr y-linux.com/cata log/items?code =16370
スーパー三 端子5Vレ ギュレータ	V7805-1000	モータ用の供 給電圧をサー ボモータの許 容電圧範囲ま で降圧する	http://akizukiden shi.com/catalog/ g/gM-06350/

9軸センサ	AE-BMX055	加速度(横転, 反転判定), 地磁気(方位推定の制御履歴), ジャイロ(着地判定)の値取得	https://akizukidenshi.com/catalog/g/gK-13010/
気圧センサ	AE-BME280	着地判定	http://akizukidenshi.com/catalog/g/gK-09421/
電子ブザー	PB04-SE12H PR	光検知中であるかどうかの把握	http://akizukidenshi.com/catalog/g/gP-04497/
光センサ	MI527/MI5527	キャリアからの放出判定	http://akizukidenshi.com/catalog/g/gI-00110/
LED	OSTA5131A-R/PG/B	実行中のシーケンスの把握	http://akizukidenshi.com/catalog/g/gI-02476/
LoRa通信モジュール	RM92-AN	機体間通信, ロスト対策としての地上局との長距離通信, 自己位置座標を地上局へ送信(ダウンリング)	https://www.green-house.co.jp/products/rm-92an/
Wifi 無線 LAN 子機	TL-WN725N	Orange pi pc でWifiを使用可能に拡張	https://www.tp-link.com/jp/home-networking/adapter/tl-wn725n/

4.3.2 子機

図4.3.2に子機を制御するシステムの構成を示す。子機には9軸センサ(加速度, ジャイロ, 地磁気)とGNSSセンサが搭載されており, これらのセンサから得た情報からモータの動作を制御する。また, LoRaモジュールを搭載しており, 機体間の通信はこのモジュールを通して行う。ロスト対策でもLoRaモジュールを通して子機の位置情報を地上局に適宜送信する。使用する各センサやモジュールの名称および使用目的を表4.3.2にまとめ, その詳細を以下に示す。

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

．(具体的な値の利用方法については4.4節ミッションシーケンスおよびアルゴリズムの方で記述する)

- GNSSセンサ
 - 自己位置推定
 - 取得したGNSS座標を用いて自己位置座標を更新
 - ナビゲーション処理
 - 目標座標と毎回取得するGNSS座標の差分から進行方向を調整
 - スタック判定
 - 毎回取得するGNSS座標の差が一定以内ならばスタック状態と判定
- 9軸センサ
 - ナビゲーション処理
 - 地磁気を用いた進行方向の調整
 - 横転・反転判定
 - 重力の方向から, 子機の姿勢を把握
 - 姿勢制御
 - 重力の方向から, 走行時の姿勢を制御
- LoRa
 - 機体間の相互通信に使用

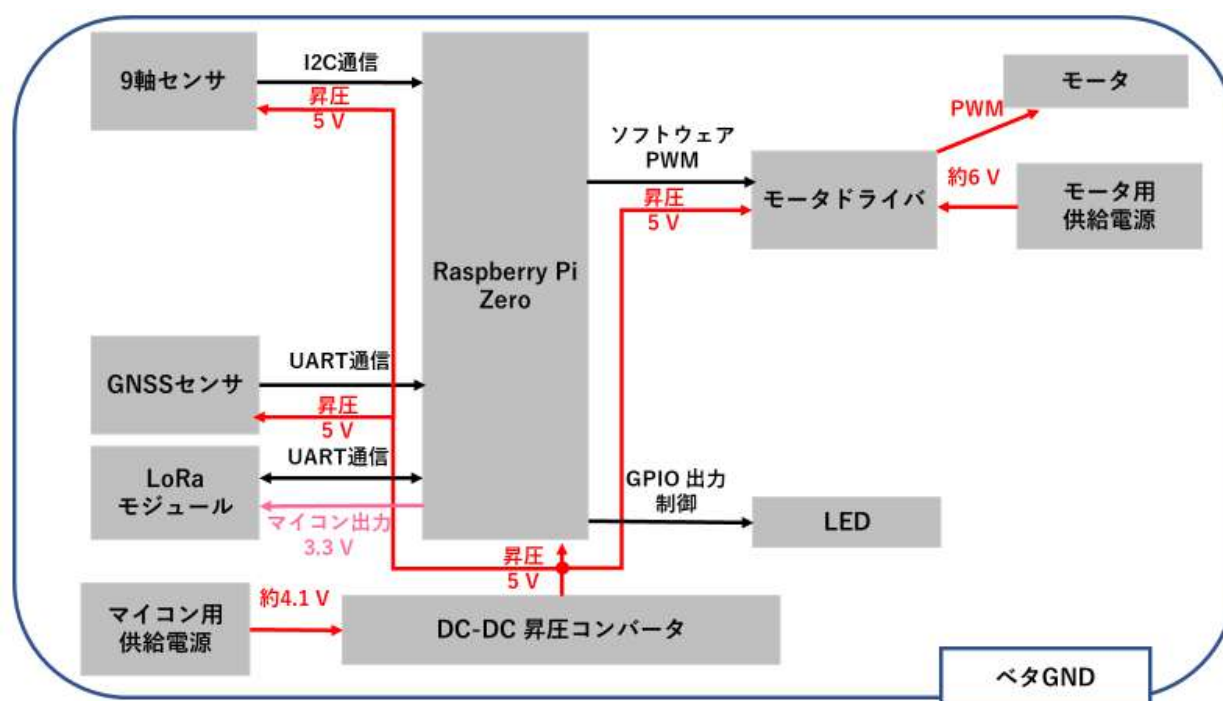


図4.3.2 子機のシステム図

表4.3.2 子機搭載計器使用

分類	名称・型番	使用目的	入手先URL・参考情報等
マイコン	Raspberry Pi Zero WH	入出力値の計算, ログデータの記録	https://www.switch-science.com/catalog/3646/
GNSSセンサ	AE-GYSFDMAXB	目標地点への方向・距離の取得, スタック判定 (GPS(米国), QZSS(日本))に対応)	https://akizukidenshi.com/catalog/g/gK-09991/
モータドライバ	DRV8835	モータの出力制御	http://akizukidenshi.com/catalog/g/gK-09848/
モータ	Pololu 210:1 Micro Metal Gearmotor HP 6V	タイヤの回転	https://store.shopping.yahoo.co.jp/suzakulab/pololu-996.html
カップリング	ASJ12-3-3	車軸部品	https://jp.misumi-ec.com/vona2/detail/221006230444/
モータ用供給電源	eneloop pro BK-4HCD/4SA 単4形充電電池	駆動系(サーボ, モータ)への電源供給	https://www.amazon.co.jp/dp/B07FQPFBHM?ref=cm_sw_r_cp_ud_dp_H23A3AZQWRBQVF_MHD9M4&th=1
マイコン用供給電源	DTP603048(PHR)	マイコン及びセンサ類への電源供給	https://www.marutsu.co.jp/pc/i/836350/
DC-DC昇圧コンバータ	TPS63070	マイコン用の供給電圧を5 [V]に昇圧	https://strawberry-linux.com/catalog/items?code=16370
9軸センサ	AE-BMX055	加速度(横転, 反転判定のため), 地磁気(方位推定の制御履歴のため), ジャイロ(着地判定のため)の値取得	https://akizukidenshi.com/catalog/g/gK-13010/
LED	OSTA5131A-R/PG/B	実行中のシーケンスの把握	http://akizukidenshi.com/catalog/g/gI-02476/
LoRa通信モジュール	RM92-AN	ロスト対策としての地上局との長距離通信, 自己位置座標を地上局へ送信(ダウンリンク)	https://www.green-house.co.jp/products/rm-92an/

4.3.3 使用電源

- 親機電源
 - 乾電池(モータ・サーボモータ用電源)
 - 品名: Energizer Ultimate Lithium AA Batteries
 - 型番: 12-2037
 - 備考
 - 6本を直列(約9~10 [V])
 - サーボモータに対しては、必要な電圧が5Vであるため、スーパー三端子レギュレータを用いて降圧する。
 - リチウムイオンポリマー電池(回路用電源)
 - 品名: リチウムイオンポリマー電池3.7 [V] 2000 [mAh]
 - 型番: TP605068(PHR)
 - 備考
 - Orange Pi PCに必要な電圧が5 [V]であるため、昇圧型DC-DCコンバータモジュールを用いて昇圧する。
- 子機電源
 - 乾電池(モータ用電源)
 - 品名: エネルーブ ENELOOP PRO ニッケル水素電池単4
 - 型番: BK-4HCD/4SA
 - 備考
 - 5本を直列(約6~7 [V])
 - リチウムイオンポリマー電池(回路用電源)
 - 品名: リチウムイオンポリマー電池3.7 [V] 860 [mAh]
 - 型番: DTP603048(PHR)
 - 備考
 - Raspberry Piに必要な電圧が5 [V]であるため、昇圧型DC-DCコンバータモジュールを用いて昇圧する。
- 安全対策
 - 回路用電源には基本的にリチウムイオンポリマー電池を使用している。この電池の発火する事故を防ぐため、使用時以外は基本的にリポバッテリーのセーフティバッグに入れて管理する
 - また、保管時は他の荷物からの衝撃が加わらないような位置に常に置く。

4.4 アルゴリズム

アルゴリズムのフローチャートを図4.4.1に示す(後の詳細事項はフローチャート内の番号に対応している)。特に図中の未踏エリア探索サブシーケンスは今回のミッションで最も重要である。各子機の探索により収集した安全評価データから親機の最良の経路を導出する手順に該当する。

今回の手法を端的に述べると、子機が走行した際の位置情報と加速度情報と方位角から、荒い地面では加速度の分散が大きく上昇したり、機体が突っかかるスタックの可能性のある地

面では分散が大きく減少したりすることに注目した地形の評価(安全評価データ)を生成する。親機がそのデータの位置情報を点ではなくGNSSセンサの誤差分の円として捉えることで、データをファジィに解釈し広域的なマップを効率的に作成、同時に複数子機によって親機の要探索点の発見を狙うものとなっている。これは親機が動かず(事前情報なき移動に伴うリスクを負うことなく)して、探索のためのハザードマップを要探索点も含め効率的に調査&生成する試みとも言える。

図4.4.2は親機の探索経路を生成するために使用可能な一般的なマッピング手法を比較したものである。本手法で重要なのは情報量の少なさと効率的な情報収集とコストの低さである。LiDAR SLAMや衛星写真では3Dマップデータや画像などの大きいサイズのデータのやり取りが必要であるが、加速度は分散などを用いることで圧縮ができる。またLiDARは高精度なスキャンのためには探査機を低速もしくは停止させながら観測する必要がある、単純に時間を要する他、探査機の走行性能に対する地形の評価ができない。衛星写真は地形の安全性を評価するには解像度が不足しており、永久影は評価できないなどの問題もある。精度に関しても、LiDAR SLAMは確かに比較的高精度な位置推定ができる可能性があるが、同一座標を通るという保証がない限りは、上記で述べた誤差分の円をどの程度大きく捉えるか(どれぐらいファジィに考えるか)という議論の延長線上にあり、機体の性能か3Dマップのどちらを安全評価に用いるかを効率性とコストの観点からトレードオフで考えることになる。我々は、機体の性能に対する地形の評価と探索の効率性、コスト削減の観点から今回の手法を選択した。

4.4.1 プログラム全体のフローチャート

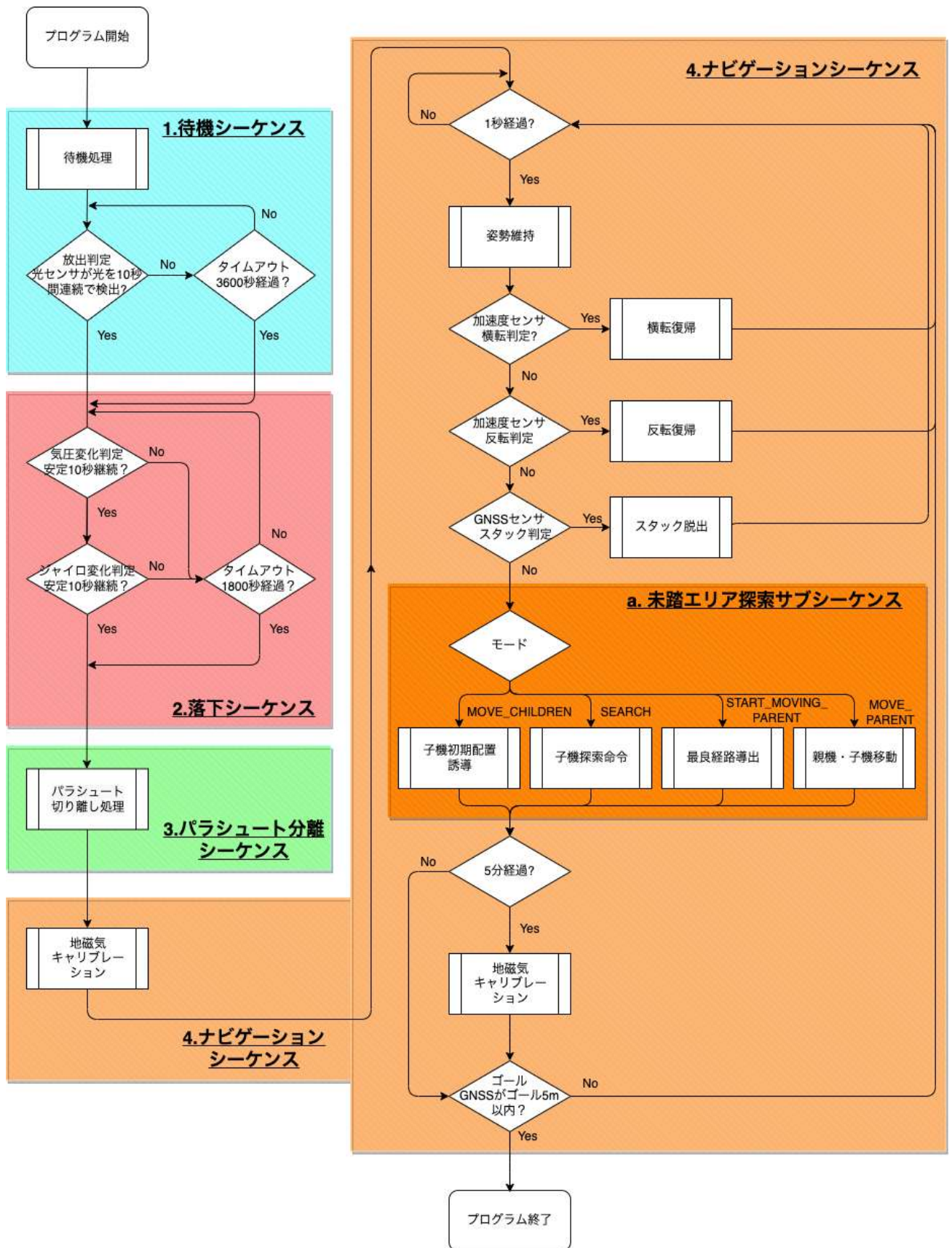


図4.4.1 アルゴリズムのフローチャート

	主要素	位置精度	情報量	惑星探査 (屋外)利用
ファジーな 地形評価 (本手法)	<ul style="list-style-type: none"> GNSS 加速度 	約5m (GNSSセンサの精度)	少ない	○
LIDAR SLAM	<ul style="list-style-type: none"> GNSS LIDAR SLAM 	約数十cm (屋内)	非常に多い	×
衛星写真	<ul style="list-style-type: none"> GNSS 衛星写真 	約50cm以下 (地上分解能)	多い	○

図4.4.2 手法比較マトリクス表

1. 待機シーケンス

キャリア搭載から親機放出までの間は親機を待機させる必要があるため、プログラム開始からは待機状態とする。可変抵抗の調整により、あらかじめ検出の閾値を調整した光センサが光を検出している状態を10秒継続した場合は親機が放出されたと判断し次のフローに移る。

光センサの異常により、光を検出しない場合のために、以下の条件を同時に満たした場合は親機が放出されたと判断し次のフローに移る。

- 条件A: 加速度センサの値がしきい値以下
 - 加速度センサのL2ノルム値 < 0.8 [deg/s]
- 条件B: 条件Aを満たしてから3秒以内に加速度センサ値が5.0 [G]を観測する

※単位Gは重力加速度 (≈ 9.81 [m/s²])

2. 落下シーケンス

親機が着地したことを判定するため、以下のすべての条件を同時に満たした状態が一定時間継続するまで待機する。

- 条件1: 気圧センサの値の変化量がしきい値以下
 - $|\text{気圧センサの値の変化量}| < 6.0$ [hPa] を10秒継続。
- 条件2: ジャイロセンサの値の絶対値がしきい値以下
 - 以下の3つの条件を同時に10秒継続。
 - $|\text{ジャイロセンサx軸の値}| < 35.0$ [deg/s]
 - $|\text{ジャイロセンサy軸の値}| < 35.0$ [deg/s]
 - $|\text{ジャイロセンサz軸の値}| < 35.0$ [deg/s]

条件を満たした場合、着地したと判断して次のフローに移る。

3. パラシュート分離シーケンス

サーボモータを駆動し、パラシュート接合部の切り離しを行う。
具体的には、3秒間サーボモータを上下に駆動し、親機前進。これを10回繰り返す。
終了後、次のフローに移る。

4. ナビゲーションシーケンス

GNSS座標(現在地およびゴール地点)と地磁気センサによる親機前方方向の情報から親機の進むべき方向と長方形(横幅 50 [m] × 縦幅 100 [m])の探索領域を指定(指定領域とする)する。そして、未踏エリア探索サブシーケンスにより親機前方領域の安全であると推定された最良経路を走行する。走行後に再び進むべき方向と指定領域を決定し未踏エリア探索サブシーケンスを実行することを繰り返す。

また、このシーケンスを開始した直後および5分経過ごとに、機体を8字で走行させ各方位の地磁気データを取得することにより、地磁気のキャリブレーションを行う。

更に走行中は加速度センサやGNSSセンサから得られる値をもとに横転判定・反転判定・スタック判定を実行する。それぞれの具体的な判定条件は以下の通りである。

- 横転判定:
x軸(左右方向)の加速度ノルム > 0.65 [G]
1秒ごとに計測
- 反転判定:
z軸(基板鉛直方向)の加速度センサ値 < 0.8 [G]
1秒ごとに計測
- スタック判定:
5秒ごとのGNSS記録で過去5回分の記録の平均 < 1.0 [m]

横転、反転判定された場合はミッションを一時中断してタイヤとサーボモータによる復帰動作を実行する。

a. 未踏エリア探索サブシーケンス

- 概要

親機となる打ち上げたローバによって安全に未踏エリアを探索するために、複数台の子機による安全評価データの収集とそのデータを用いて生成された最良経路の走行を行う。具体的なフローを以下に、そのイメージ図を図4.4.3に示す。

1. 親機が子機群を、親機の周辺でかつ親機が進行を望む方角(ゴール方向)に対してゴール方角と垂直となるように配置するために、GNSS座標を算出しそれぞれ子機に送信、子機は走行する。
2. 親機がゴール方向に対して設定した未踏エリアである特定範囲内から、子機用の探索座標点を生成、子機に座標点を送信することで探索させ、安全評価データを収集する。
3. 集まった安全評価データを元に親機が安全に走行可能と考えられる最良経路探索を行い、親機をその経路に沿って走行させる。子機は

親機に自身の座標を送信し、親機にその地点からの最良経路を導出して子機に返信してもらう。そして子機はその経路に沿って次の初期位置まで走行する。

4. 1. ~ 3. を親機がゴールに到達するまで繰り返す。

なお、このサブシーケンスの実行中に子機が親機との距離が離れ過ぎてしまったことにより通信不可能な状態(何らかのデータが受け取れない状態を120秒継続した後、接続確認のコマンドを送信して返信がない状態を120秒継続した時に通信不可能と判定)になってしまったり、子機が横転・反転してしまったことによりミッション続行不可能な状態になってしまったりした場合は、親機と子機のペアリングを解除し子機のプログラムを停止させる。

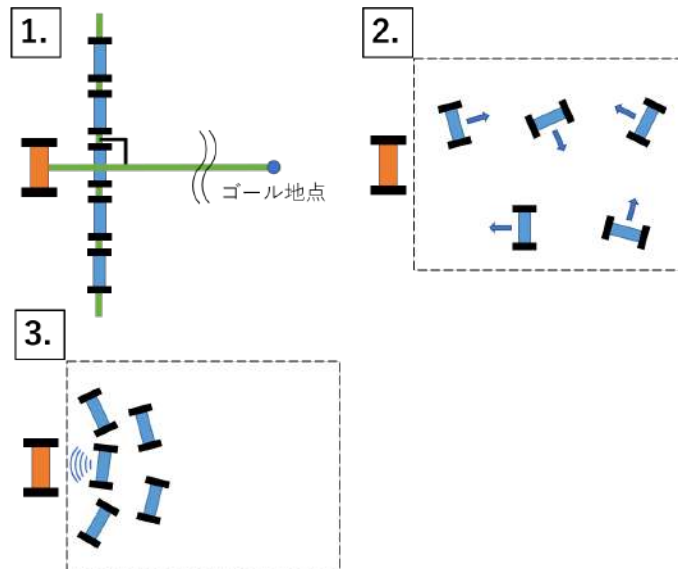


図4.4.3 未踏エリア探索サブシーケンスのイメージ図(橙: 親機, 青: 子機)

- 子機初期配置誘導

探索前に子機を親機から見たゴール方向かつ親機とゴールを結んだ直線と垂直に並ぶように親機が命令を送る(図4.4.4)。それぞれの子機の緯度経度を指定して子機の位置を指定する。具体的なフローを以下に示す。

0. 子機の数, 親機-中央の子機(以下中央子機)間距離, 子機-子機間距離はあらかじめ決めてある(図4.4.4)。
1. 親機からみたゴールの方位角 θ と親機とゴールを結んだ直線と親機と子機が配置される地点を結んだ直線のなす角 φ から, 親機からみた子機が配置される地点の方位角を求める(図4.4.5)。
2. 子機が配置される地点の方位角, 親機-中央子機間距離, 子機-子機間距離から, 親機を原点として緯線に平行なx軸と経線に平行なy軸を用いた座標系(ローバ座標系と呼ぶ)での子機が配置される地点の座標を求める。(図4.4.6)
3. この座標と親機の緯度経度から子機が配置される地点の緯度経度を求める。
4. 親機がその緯度経度の子機を向かわせる。

5. 1. ~ 4. をすべての子機に対して行う.

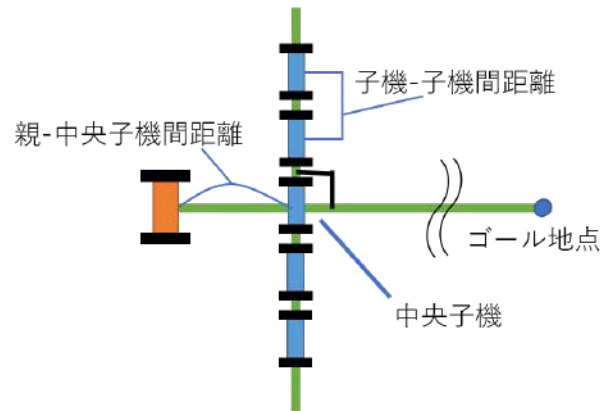


図4.4.4 初期配置(橙:親機, 青:子機)

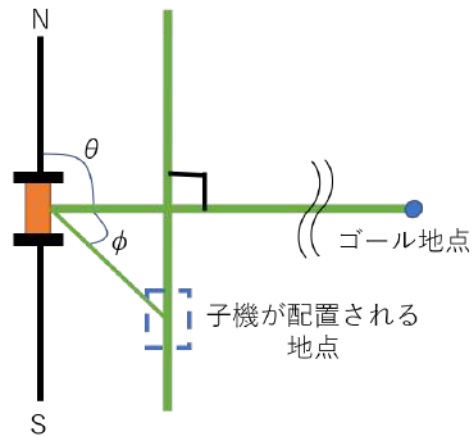


図4.4.5 親機からみた子機が配置される地点の方位角(橙:親機)

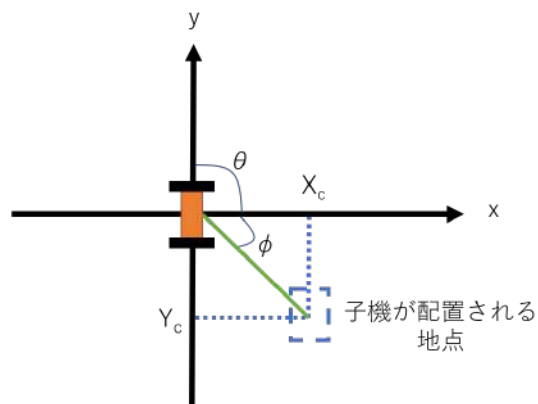


図4.4.6 ロボット座標系(橙:親機)

- 子機探索命令

子機の探索経路はなるべく子機同士の衝突を回避し, 探索箇所のムラを無くすために子機の情報を集約した親機より目標座標を次々に指定し, 子機は

各目標地点へ下記の安全評価を実施しながら走行する。具体的なフローを以下に示す。

1. 初期位置到達後または以前に親機に指定された目標座標に到達した子機(Aとする)は、親機に次の目標座標を要求する。
2. 親機はAの座標から複数の方位・距離の目標座標候補を生成。具体的には乱数で生成したある方位角を起点に一周分(360度)を17等分した17個の方位角と5,10,15 [m]を組み合わせた計51個の現在座標からの目標座標候補として生成する。
3. 目標座標候補から下記の条件に該当するものを除外する。
 - 目標点が探索の指定領域を外れるもの
 - Aの座標と目標点を繋ぐ線分が、その時点のA以外の子機の座標と目標座標を繋ぐ線分に一つでも交わるもの(更にはGNSS座標誤差を考慮し、実際には線分に5 [m]の幅を持たせている)
4. 残った目標座標候補のうち、目標座標周辺の安全評価データが少ない座標を選択。これを次の目標座標としAに送信する。
5. 次の目標座標を受け取ったAは探索を開始する。(目標地点に到達次第、再び1.を実行)

また、図4.4.7は上記のフローを図示したもので、1.~4.が図中番号に一致する。(5.は省略)

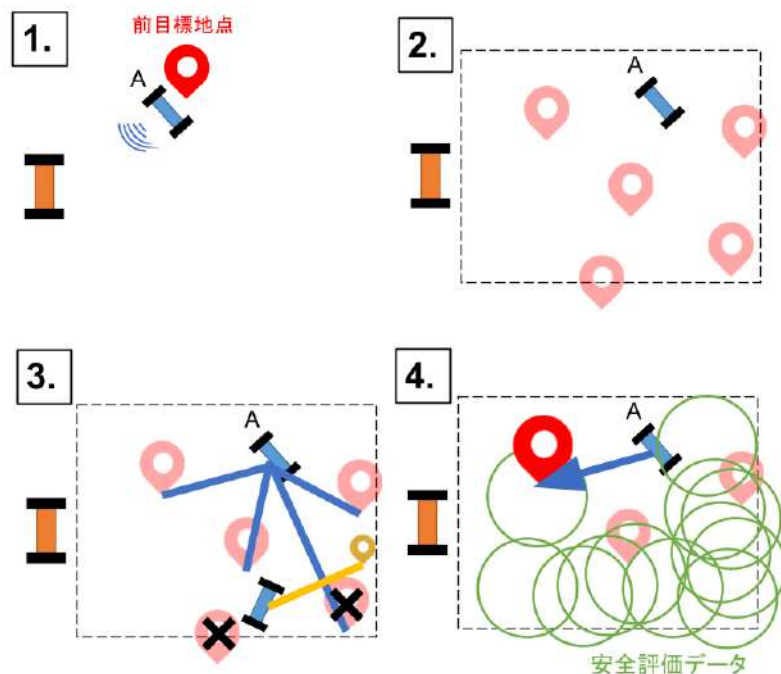


図4.4.7 子機探索命令
(橙:親機, 青:子機(特にAが次目標を求める子機))

- 安全評価

目標座標に向かっている子機は、5秒間程度(※)に一度、その間の加速度センサの値から子機が走行してきた経路を評価する。評価の参考になっているのは予備実験で確認した以下の結果に基づいている。

「荒れた地形」なら走行時の加速度の分散 σ が大きい

「突っかかる地面」なら突っかかっている間の加速度の分散が小さい

また、背丈の高い草など走行不能な領域に侵入した際に発生するスタック判定なども安全性の評価として重要であるため、スタック判定・反転判定・横転判定は上記の評価式とは別途確認し、大きな問題がある地面として評価する。

予備実験について、大学周辺の4種類の地形(以後それぞれ「荒いコンクリート(図4.4.8)」、「平らなコンクリート(図4.4.9)」、「平らな砂地(図4.4.10)」、「丘のある砂地(図4.4.11)」とする)においての走行実験データから求めた加速度の値の分散を求めた。分散が大きいほど速度が安定しないため荒れた地形とした。また、突っかかるとその場にとどまるため一定時間速度の分散が低いと突っかかりやすい地形とした。



図4.4.8 荒いコンクリート



図4.4.9 平らなコンクリート



図4.4.10 平らな砂地



図4.4.11 丘のある砂地

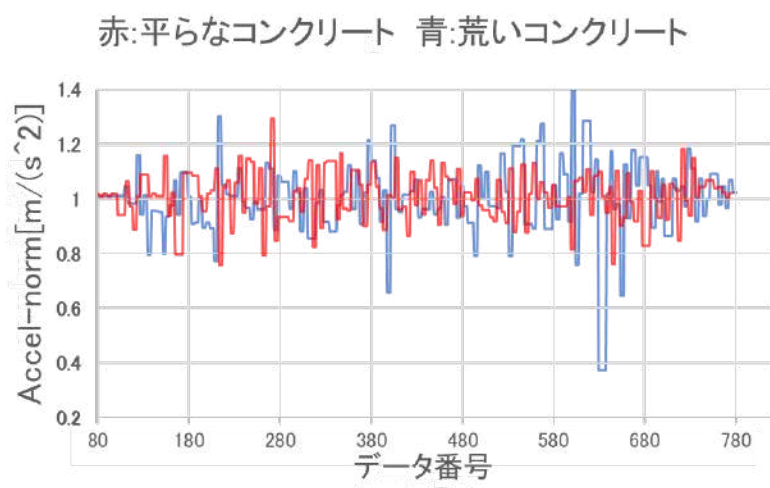


図4.4.12 走行実験で得られた加速度の推移(コンクリート)

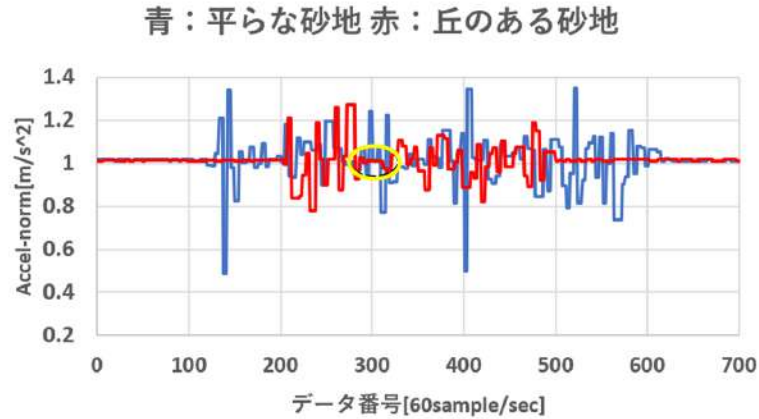


図4.4.13 走行実験で得られた加速度の推移(砂地)

図4.4.12は荒さ異なる2種類のコンクリート環境における走行実験で得られた加速度の推移を示す。縦軸は加速度のノルム、横軸はデータ番号である。同図からわかる通り、2つの環境を比較すると「荒いコンクリート」では、特に後半500～680サンプルにかけて、加速度のノルムが大きくぶれて不安定な走行になっていることが顕著に分かる。

同様に、図4.4.13に起伏の異なる2つの砂地環境における走行実験で得られた加速度の推移を示す。この図の黄色い円に囲まれた範囲(270～320サンプル)における「丘のある砂地」(赤)の加速度に着目すると、円の直前で1.3の加速度になった後にしばらく1.0付近、その後50サンプルほど0.9～1.1付近を推移していることがわかる。これは、子機が砂地の丘部分を走行し、砂にはまっているタイミングである。子機が砂にはまると、速度が出せずその場に停滞気味になるため必然と加速度のノルムも1に近づく。この「荒いコンクリート」、「丘のある砂地」のような場所を走行に適さない土地とし、親機の走行ルートから外すために今回の予備実験データから評価値の基準式を設定する。

今回設定している評価関数は以下の通りとなっている。

$$Safety\ Evaluation = \begin{cases} -5000 \left(\frac{a - M}{M} \right)^3 & \text{if } a \leq M \\ 4000(a - M)^3 M^{0.1} & \text{otherwise} \end{cases}$$

aは過去30サンプル(約1秒)ごとの加速度の分散値の過去1秒の平均(平均を用いているのは瞬間的な突っかかりを確実に安全評価データに反映させるため)、Mは閾値を表しM=0.5である。このグラフをプロットすると図4.4.14のようになる。

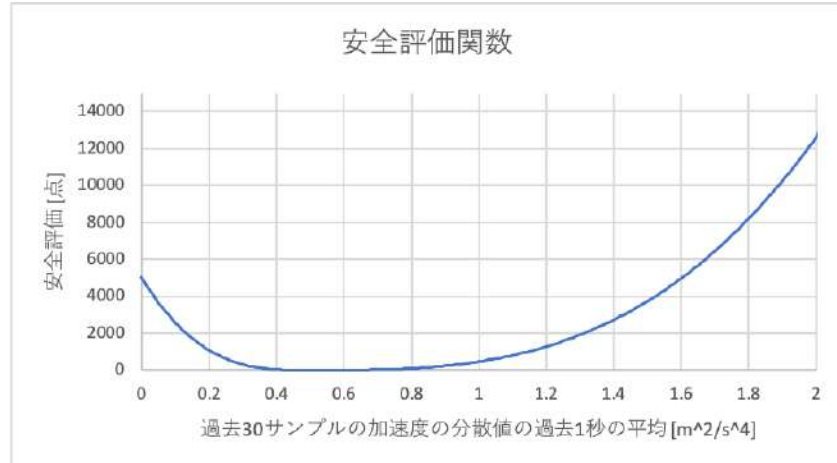


図4.4.14 安全評価関数

この図における安全評価値は値が低いほど安全性が高いことを意味する。図を見ると分散値の過去1秒の平均が0に近い時や1.0を超えてから値が上昇しているが、これがそれぞれ「突っかかる地面」と「荒い地面」に該当し、それぞれで安全評価値が大きく(安全性が低く)出ているようになっている。なお、安全性が低いほど安全評価値が高くなるようにしている理由は、最良経路導出で用いるダイクストラ法における移動コストとして利用するためである。

また、この評価はその評価時のGNSS座標に紐づけられるが、GNSS座標の誤差である半径約5 [m]を考慮して、半径5 [m]の円内(以下、安全評価円とする)は同一評価であると扱う。安全評価円は図4.4.15のような要素で表される。円は評価値、座標(緯度・経度)、方位角の情報を持つ。このデータを親機に送信することで最良経路探索の情報源とする。安全評価円の重複などについては次項である”最良経路導出”について詳細を記述する。

安全評価円 = (評価値, 緯度, 経度, 方位角)

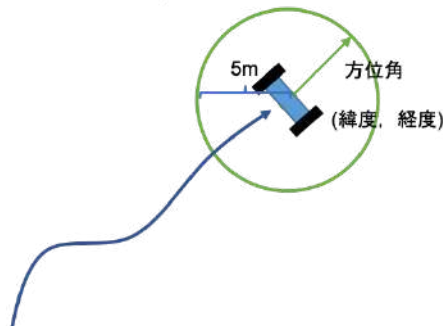


図4.4.15 安全評価円

※”5秒間程度”というのは複数機体同士で通信し合うLoRaの連続通信が可能な時間間隔に依存する値である。高速であることが望まし

いが、GNSSの誤差と子機の走行速度を考慮すると、極端に早い必要はない。

- 最良経路導出

子機が収集した安全情報をもとに親機的最良経路探索を行う。今回はダイクストラ法を用いて最短経路を求めた。

ダイクストラ法とは、ノードと辺によるグラフにおいて各辺に重みがあるとき、その重みが最小になる経路を求めるアルゴリズムの一つである。総当たりで経路を調べる場合、規模が大きくなるにつれ計算が膨大になることが知られているが、ダイクストラ法の以下の手順で求めていくことで効率的に経路を求めることができる。

1. 頂点集合全体をVとし、始点からの最短経路(最短距離)が既に確定した頂点集合をSとする。
2. 現時点で経路が判明している頂点のうち、始点からの距離が最小となる頂点pを新たにSに移動させる。(図4.4.15.①)
3. Vに残っている頂点pと接する各頂点について、既に求めた最短距離をもとに始点からの最短距離を更新する。(図4.4.15.②)
4. 始点から全ての頂点への最短経路が確定するまで(V が空になるまで)2と3を繰り返す。

今回のノードは親機からの指定探索エリアを格子状に割り、GNSSにてノードを設置している。また、辺に直接重みがあるのではなく、2点間におけるノードの評価の差をその2点の辺の重みとしている。なお、重みが負になる場合今回の手法は適用できない。以下の図4.4.16は一般的なダイクストラ法に関する手順であり、始点Aから終点Eまでの経路で最も移動の重みの合計が小さい最短経路を算出している様子を表す。図4.4.17は今回の最良経路探索においてダイクストラ法を適用するためのノードと辺の生成イメージである。指定探索エリアにおいて、縦横5mずつで格子状に分割し、交差した点を緯度経度に変換したものがノード、各ノードの最も近くにある最少2つ最多4つのノードを連結するのが辺である。また、各辺の重みは前項の安全評価円を求めることができる。具体的には各ノードの緯度経度と被っている安全評価円を取り出し、評価値の平均を求める。この際、直前のノードと現在のノードから出した理論上の方位角と取り出した安全評価円の方位角の差分から重み付き平均で算出する。特に、安全評価円の方位角と理論の方位角が近いほど、その評価結果が重視されるようにする。これらのノード、辺、そして安全評価による各辺の重みを入力に図4.4.18のようにダイクストラ法を適用することで最良経路を算出する。この場合のスタート地点とは現在の親機座標が最も近い格子点であり、ゴールは右辺の格子点全てが候補である(算出された最良経路の中でも最も良いものが採用される)。

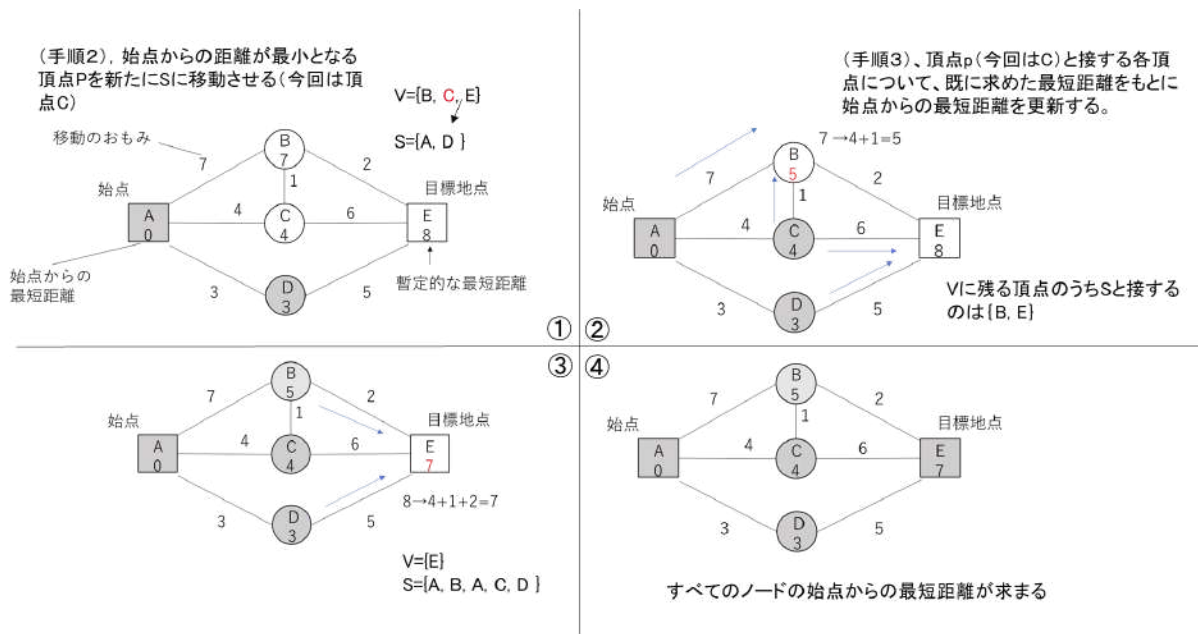


図4.4.16 ダイクストラ法

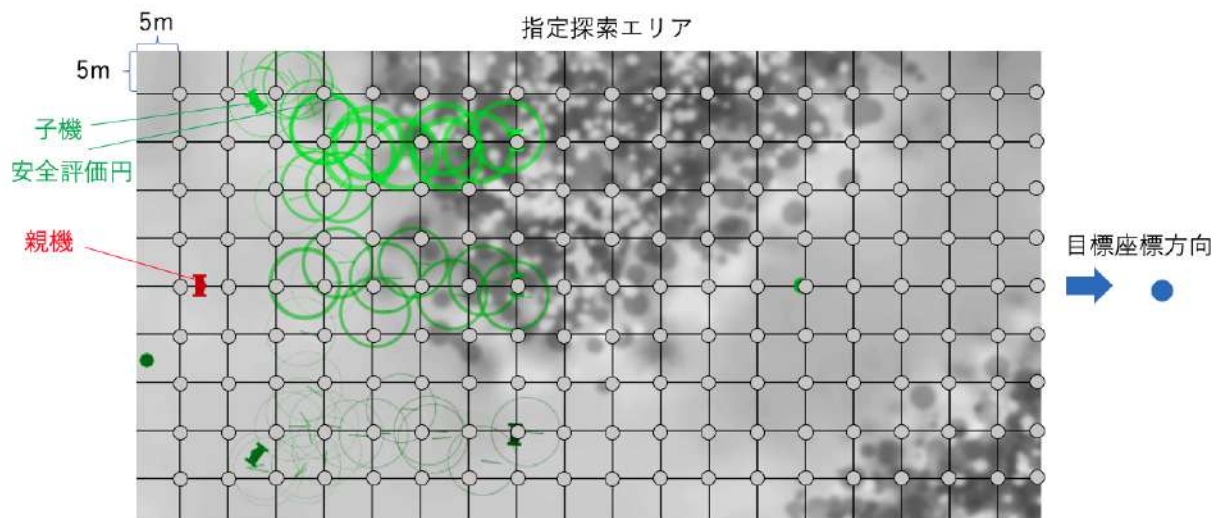


図4.4.17 探索におけるダイクストラ法のノードと辺生成のイメージ

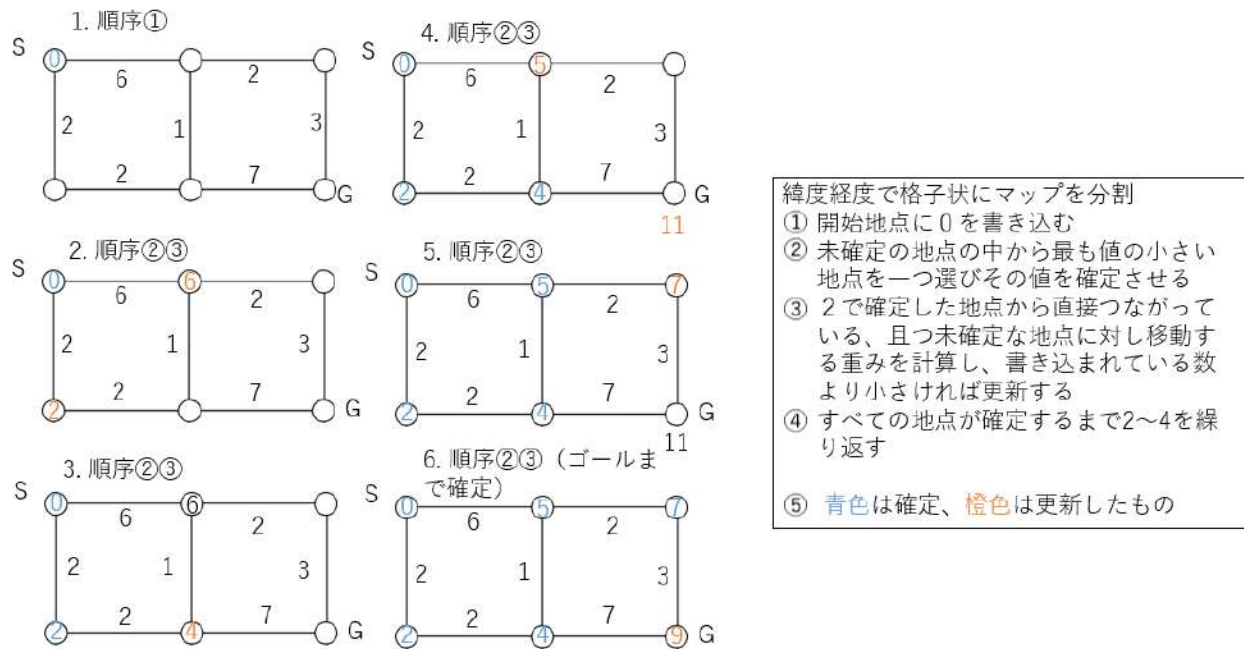


図4.4.18 格子点へのダイクストラ法適用

- 親機・子機移動

親機は最良経路探索で求めた経路を通るような複数の中継目標座標を設定し、最終目標座標まで走行する。子機は自身の座標を親機に送信し、そこから次の初期位置への最良経路を通るような複数の中継目標座標を親機から子機に送信、その経路に沿った走行を行う。

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
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, M10	7/1
V4	振動試験	S4	7/29
V5	分離衝撃試験	S5	7/29
V6	開傘衝撃試験	S5, M10	7/1

V7	パラシュート投下試験	S6	6/1
V8	長距離通信試験	S7	7/31
V9	通信機電源ON/OFF試験	S8	6/23~6/25
V10	通信周波数変更試験	S9	6/23~6/25
V11	End-to-end試験	S10	8/5
V12	着地衝撃試験	M1, M10	6/28
V13	走行性能確認試験	M2	7/22
V14	電力耐久試験	M3	7/1
V15	反転横転復帰試験	M4	7/2
V16	親機・子機間通信確立試験	M5	7/19
V17	最良経路導出試験	M6, M8	7/30
V18	子機経路評価試験	M7	7/29
V19	子機初期位置配置試験	M9	7/30
V20	子機数変動試験	M11	8/3
V21	指定経路走行試験	M12	7/29
V22	制御履歴レポート作成試験	M13, M14	8/5

第6章 実施試験の内容

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
親機とパラシュートを合わせた質量が規定質量である1050 [g]以下を満たすことを確認する.
- 試験内容
親機とパラシュートを質量計で計測をし、レギュレーションに記載されている質量(1050 [g])以下であることを確認する.
- 結果
親機とパラシュートの総重量は完成した機体を補強するためのホットボンド2本分を加えても計911 [g]であり、レギュレーションの1050 [g]以下であることを確認した. 図6.1.1~図6.1.4に質量測定結果の結果の様子を示す.



図6.1.1 パラシュートと親機の質量



図6.1.2 親機のための質量



図6.1.3 パarachuteのための質量



図6.1.4 ホットボンド2本分の質量

また、以下に質量試験の動画リンクを示す。

- リンク
https://youtu.be/q3JYM_SeqOU
- 考察
 パラシュートの質量を含めた親機の総重量が、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
 以下の3点を確認することを目的とする。
 - 親機のレギュレーション(内径146 [mm], 高さ240 [mm])を満たすこと
 - キャリアに収納可能であり、円滑に放出可能であること
 - 親機を収納するまでの工程が5分以内に収まっていること
- 試験内容
 以下の4つの項目を実施する。
 - (1) 親機を入れるキャリアの内径を計測する。
 - (2) 親機の高さを計測する。
 - (3) 減速機能を有する親機がキャリアに収納・放出可能であることを確認する。これを5回繰り返し実施し、問題ないことを確認する。
 - (4) 親機収納の際に収納までの手順の時間を計測し、5分以内に収納可能であることを確認する。

- 収納手順
 1. パラシュートを畳む
 2. パラシュートの紐を絡みにくいようにまとめる
 3. 親機の側面に畳んだパラシュートをあてがう
 4. キャリアに収納する
- 結果

図6.2.1はキャリアの内径を測定した様子を表す。使用しているキャリアは大会規定のキャリアと同程度の大きさ(146[mm])であり、開発した親機が問題なくキャリアに収納できていることを確認した。



図6.2.1 キャリアの内径

また、親機測定した様子を図6.2.2に示す。親機の高さは238 [mm]であり、レギュレーションに記載された240 [mm]以内の制約を満たしている。

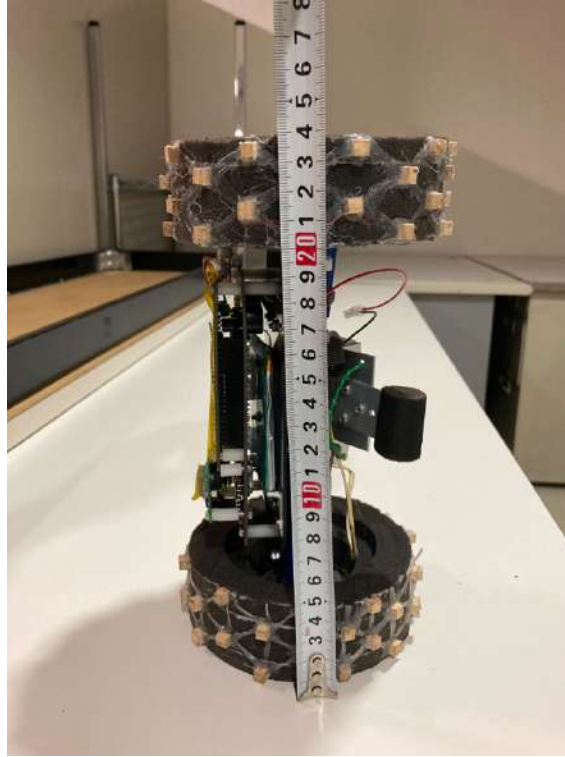


図6.2.2 親機の高さ

表6.2.1に示すように、親機を収納後、自重で放出することが5回中5回確認できた。

表6.2.1 キャリア放出実験の結果

試行回	収納時間	自重での放出	YouTubeリンク
1回目	1分32秒	成功	https://youtu.be/XzavsdYKfLQ
2回目	1分31秒	成功	https://youtu.be/o-OOvW863vA
3回目	1分45秒	成功	https://youtu.be/U7KAbnW2FTs
4回目	1分32秒	成功	https://youtu.be/c4JoEtLezEE
5回目	1分20秒	成功	https://youtu.be/XpcPtM7DigE
成功率	100%(5/5)	100% (5/5)	

- 考察
 - 以上の結果より、以下の3点が確認された。
 1. キャリアの内径と高さに関するレギュレーションを満たすこと
 2. 円滑なキャリア収納・放出が可能であること

3. 親機の収納までの工程が5分以内に収まっていること

v3. 準静的荷重試験

- 目的
打ち上げ時の準静的荷重(以下、静荷重と呼ぶ)によって親機のハードウェア・ソフトウェア共に問題が発生せず、正常に動作することを確認する。
- 試験内容
ロケット搭載状態を想定した親機を紐を繋いだ袋に入れハンマー投げの要領で回すことでロケットによる静荷重を再現する。静荷重はレギュレーションの5.2項に記載されている通り、親機の高さ方向に10 [G]を10秒間親機に与える。その後、ハードウェアが破損していないことを確認する。また、放出判定からパラシュート切り離しまでの各シーケンス動作が正常に動作するかを確認することで、モータやサーボモータに損傷がないことを確認する。また、本試験におけるセンサ、モータの出力は以下の基準を満たしていれば正常と扱う。

<センサ類>

- ・気圧センサ: 実験開始前に計測した気圧 ± 1 [hPa]
- ・9軸センサ: 機体の姿勢が静安時の加速度のノルムが0.95 [G]~1.05 [G](重力加速度1 [G] \pm 許容測定誤差5%, および、機体の姿勢によってxyz軸の値が変化している)
- ・光センサ: 光が照射されている時に光があたっていると判定(センサ値がHIGHを示す)し、手で光センサを覆ったときに光があたっていないと判定(センサ値がLOWを示す)する
- ・ブザー: 音のON/OFFができる
- ・GNSS: 1つ以上の衛星からGNSS情報を受信し、その座標が試験の実施地点を指している
- ・LoRa: 親機から子機と子機から親機への文字列の送受信を確認する

<モータ類>

- ・サーボモータ: 上下方向にスタビライザを制御できる
- ・モータ: 機体が走行可能な程度に回転する

本試験は全3回の試行を実施し、それぞれについて親機にかかっている加速度の大きさを時系列データとして取得することで、10 [G]の静荷重が与えられていることを確認する。また、静荷重の負荷開始時点からシーケンス動作の確認までの一連の動作を映像で記録する。

- 結果
9軸センサから得られた加速度の推移を図6.3.1~図6.3.4に示す。これらの図は横軸が時間、縦軸は加速度のL2ノルム [G]を示す。これらのグラフから、親機を振り回すことでおよそ10 [G]以上の静荷重が10秒以上与えられていることが確認できる。

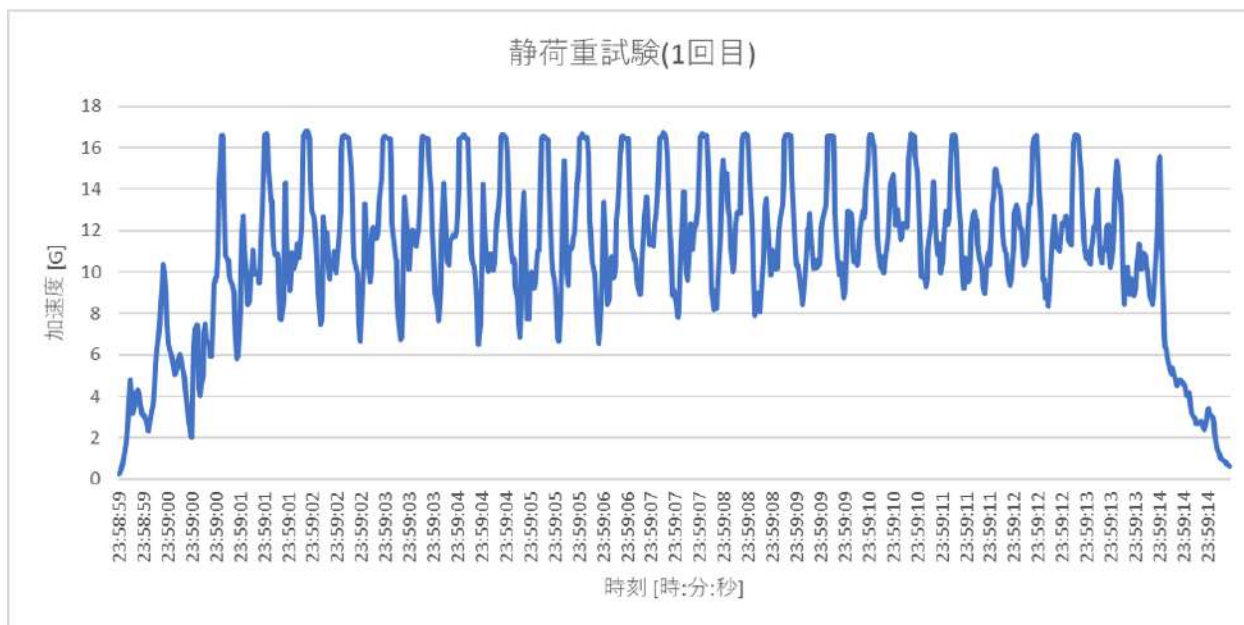


図 6.3.1 静荷重試験1回目の加速度のグラフ

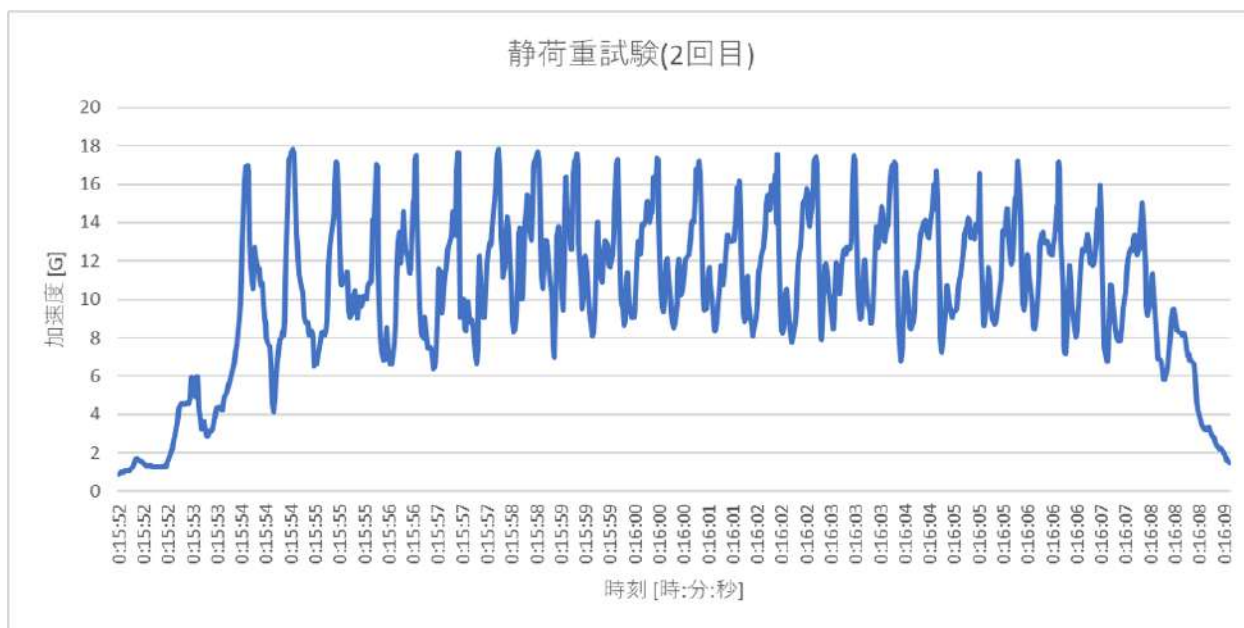


図 6.3.2 静荷重試験2回目の加速度のグラフ

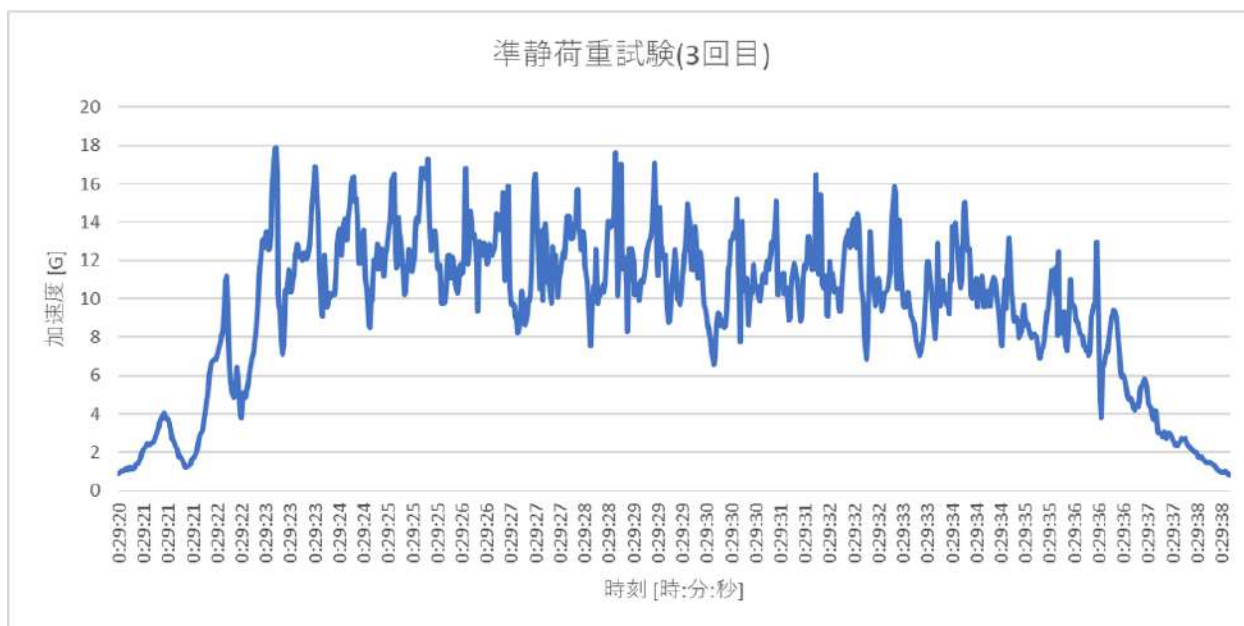


図 6.3.3 静荷重試験3回目の加速度のグラフ

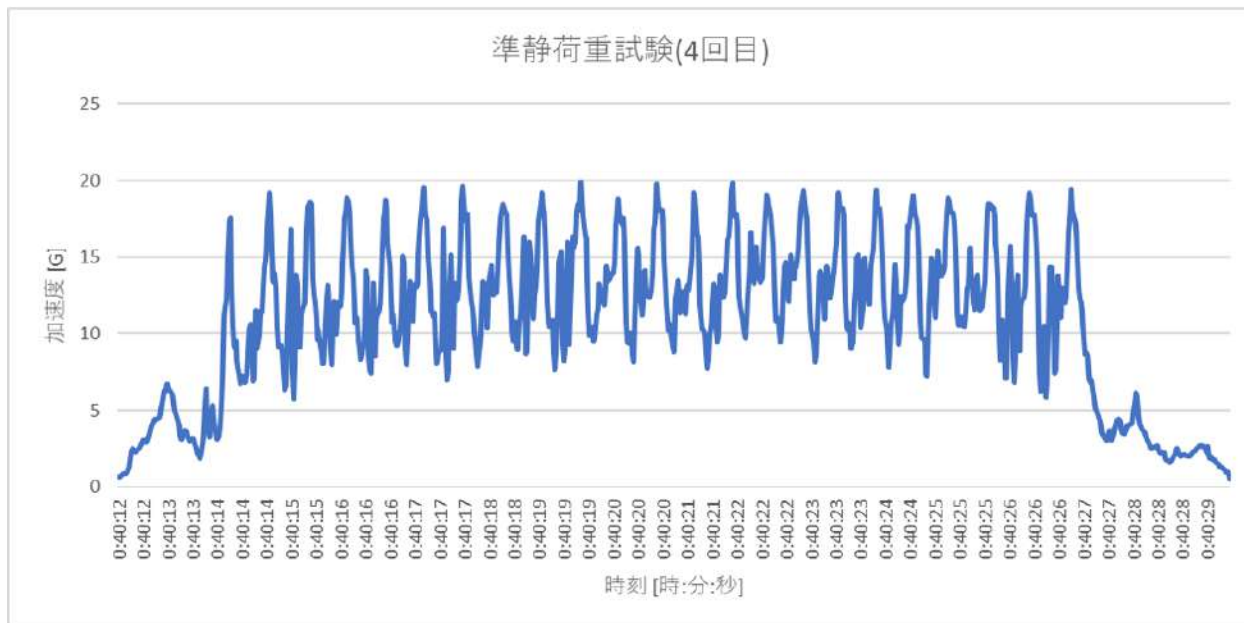


図 6.3.4 静荷重試験4回目の加速度のグラフ

静荷重を加えた後の親機の損傷および各部品の故障の有無を表6.3.1に示す。

表 6.3.1 静荷重試験の結果

回数	シーケンス動作	親機の外的損傷	モータ類	センサ類	YouTubeリンク
1回目	正常に遷移	正常	正常	問題なし	https://youtu.be/ZO7qLMWg8FE
2回目	正常に遷移	正常	正常	問題なし	https://youtu.be/bDRoay1Y1Z8
3回目	正常に遷移	正常	正常	問題なし	https://youtu.be/txsgxLfw4dk
4回目	正常に遷移	正常	正常	問題なし	https://youtu.be/E3x6AcTqckI
成功率	100% (4/4)	100% (4/4)	100% (4/4)	100% (4/4)	

※準静荷重試験後のGNSSセンサが正常な値を取得していることを確認した。その時の動画を以下に載せる。

GNSSセンサ確認動画: <https://youtu.be/qvrRHaB9VMc>

- 考察
以上の結果より、親機は10 [G]の静荷重に耐久可能であり、打ち上げ時の衝撃によってハードウェア・ソフトウェアに問題が発生せず、正常に動作することが確認された。

v4. 振動試験

- 目的
ロケットによる打ち上げを想定した際、打ち出し時に生じる振動によって親機に異常が生じないことを確認する。
- 試験内容
親機をキャリアに収納し、振動試験装置を用いてロケット上昇時に生じるランダムな振動を与える。試験後、パラシュート切り離しから走行までのシーケンスを実施し、全てのセンサ・動力系の動作確認、親機に破損がないかを確認する。加える振動の内容は以下のとおりである。

30Hz～2000Hzの範囲の10Gのランダム振動

(これについては、最新のレギュレーションのバージョンでも15Gと書かれているが、

参加団体MTGにおいて正弦波振動15Gを出すのはかなり厳しいとされ、10Gの振動に変更されている認識です。運営側に確認お願いいたします。)

実際に加振器に上記の設定を入力して、機体に負荷を与えた。

- 結果

[1回目(7/29実施)]

与えたランダム振動の加速度推移を図6.4.1に示す。グラフの横軸は時間 [sec]、縦軸は加速度 [G]を表している。また、Cは目標となる加速度、A1は実際に計測された加速度を示す。

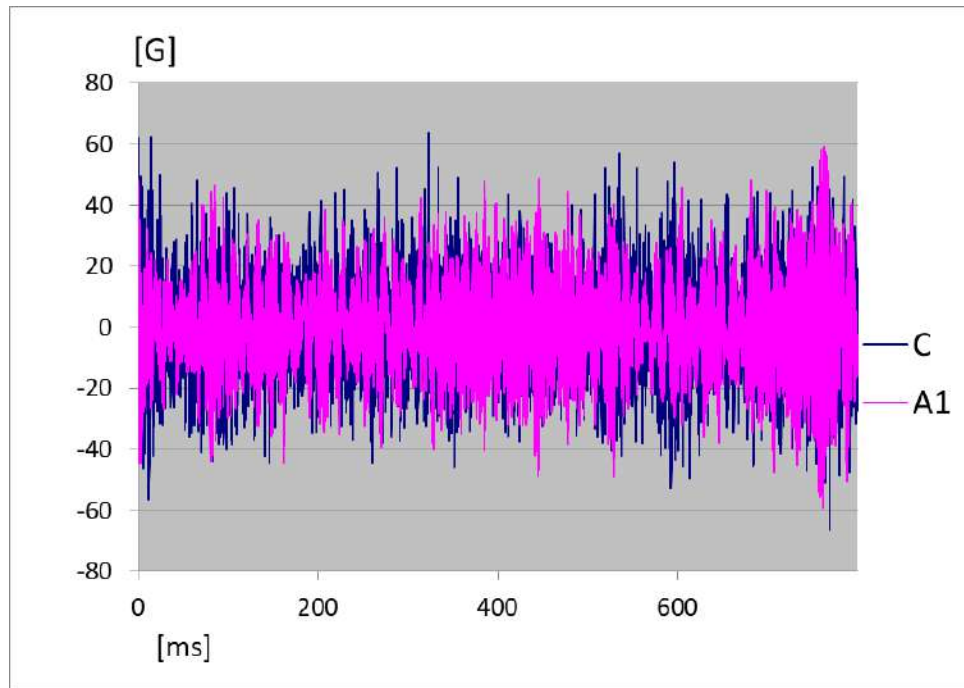


図6.4.1 ランダム振動の加速度(2回目)

また、図6.4.2にランダム振動におけるPSD(パワースペクトル密度)グラフを示す。縦軸は加速度rms $[G^2/Hz]$ であり、横軸は各周波数 $[Hz]$ である。また、今回のランダム振動の実効値は14.6972 $[Grms]$ であり、ARLISS CanSat部門の基準(試験内容に追記)を設定した時の実効値の目標値である 14.5903 $[Grms]$ を満たす基準で試験を実行できている。

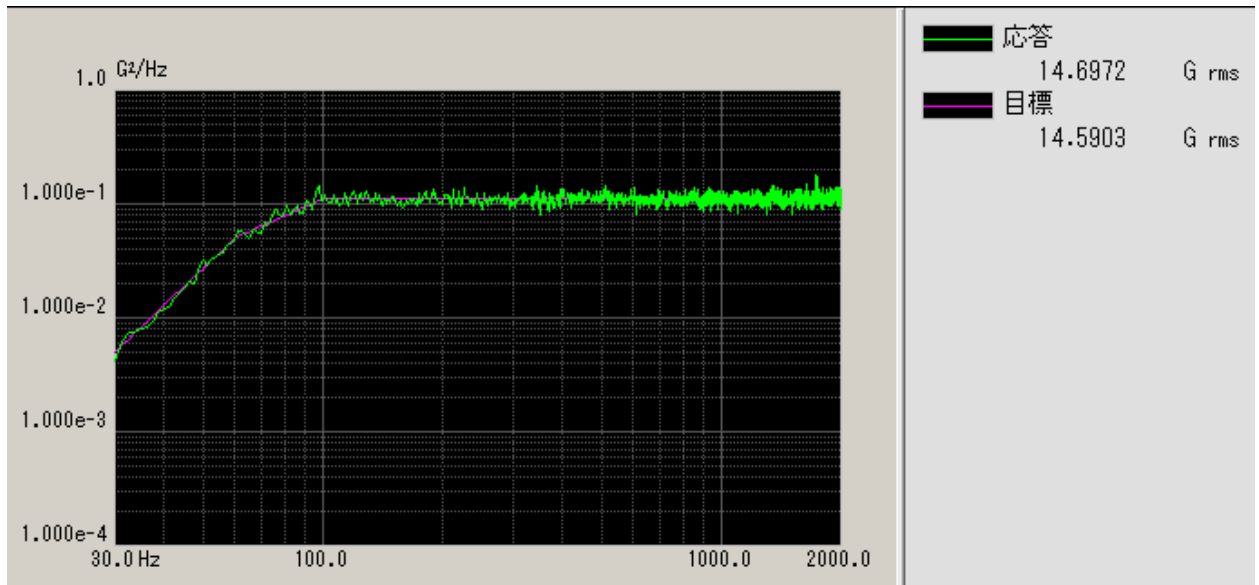


図6.4.2 PSDグラフ(2回目)

キャリア収納から振動試験装置で振動を加えた後、親機が正常に動作するかどうかの一連の流れを撮った実験動画を以下に示す。

以下に動画中の各イベントの時間を示す。

<https://youtu.be/WcV-ZAsKR-8>

- 0:00～ キャリア収納
- 2:50～ 重さ計測
- 3:07～ 振動試験装置の準備
- 18:04～ ランダム振動開始
- 25:30～ ショック振動開始
- 31:50～ キャリア放出
- 32:05～ 落下判定
- 33:14～ 着地判定
- 33:35～ パラシュート切り離し(失敗)
- (光, 加速度, ジャイロセンサ値・気圧センサ値の取得)
- 34:40～ ミッションシーケンスへの移行確認(LoRaの正常な動作を確認)
- 36:09～ 親機に損傷がないことを確認
- 37:25～ 駆動系の確認(動かない)
- 38:00～ 駆動系が動作しない原因を探る
- 40:45～ 原因がモータバッテリーからの電力供給をするコネクタの接触不良
である可能性を確認
- 41:38～ 駆動系について、動かなかった原因の現在の見解を口頭説明
- 42:45～ 各センサ(GNSS, 9軸センサ, 気圧センサ, 光センサ, ブザー, LoRa,)

以下の表6.4.1に各種動作の確認結果を示す。

表 6.4.1 振動・分離衝撃試験の結果

	放出判定	着地判定	パラシュート切り離し	ナビゲーション(通信の確立まで)	センサ類の動作	モータ及びサーボモータの動作
結果	成功	成功	失敗 (モータ, サーボともに動作せず)	成功	正常	失敗 (モータ, サーボともに動作せず)

[2回目(8/19実施)]

- 2回目の与えたランダム振動の加速度推移を図6.4.3に示す。

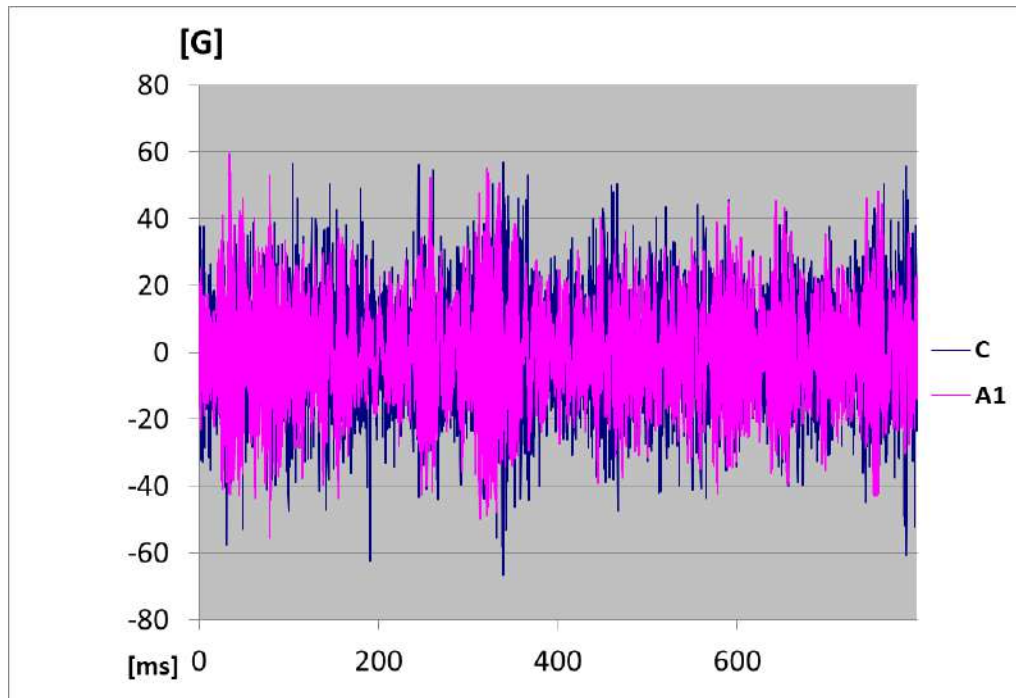


図6.4.3 ランダム振動の加速度(2回目)

また、図6.4.4にランダム振動におけるPSD(パワースペクトル密度)グラフの2回目の結果を示す。2回目のランダム振動の実効値は14.6528 [Grms]であり、ARLISS CanSat部門の基準(試験内容に追記)を設定した時の実効値の目標値である14.5903 [Grms]を満たす基準で試験を実行できている。

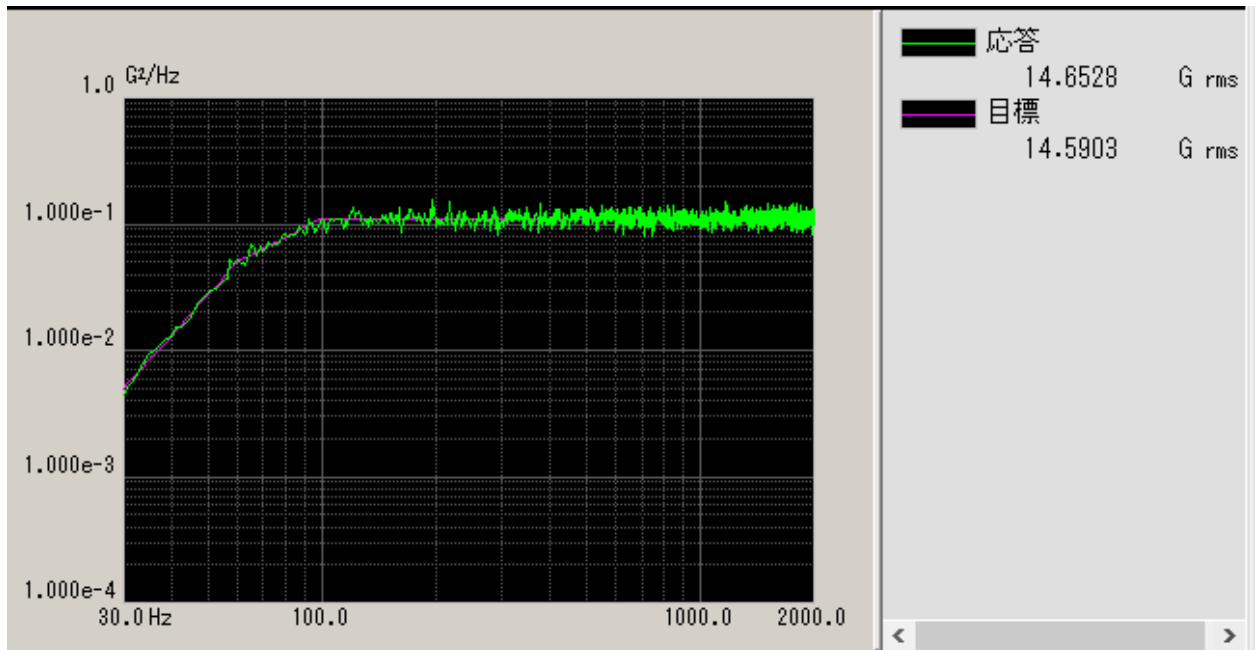


図6.4.4 PSDグラフ(2回目)

キャリア収納から振動試験装置で振動を加えた後、親機が正常に動作するかどうかの一連の流れを撮った実験動画を以下に示す。

以下に動画中の各イベントの時間を示す。

https://youtu.be/sL6krjd4_aw

- 0:00～ キャリア収納
- 2:10～ 重さ計測
- 3:05～ 振動試験装置の準備
- 29:40～ ランダム振動開始
- 34:40～ ショック振動開始
- 41:45～ キャリア放出
- 42:40～ 落下判定
- 42:50～ 着地判定
- 43:05～ パラシュート切り離し
- 43:50～ ミッションシーケンスへの移行確認(LoRaの正常な動作を確認)
- 44:35～ 親機に損傷がないことを確認
- 45:40～ 駆動系の確認
- 46:02～ 各センサ(GNSS, 9軸センサ, 気圧センサ, 光センサ, ブザー, LoRa,)
- 48:44～ 長距離走行確認
- 54:10～ GPSの確認

以下の表6.4.1に各種動作の確認結果を示す。

表 6.4.2 振動・分離衝撃試験の結果(2回目)

	放出判定	着地判定	パラシュート切り離し	ナビゲーション(通信の確立まで)	センサ類の動作	モータ及びサーボモータの動作
結果	成功	成功	成功	成功	正常	正常

- 考察

今回、振動試験におけるランダム振動＋分離衝撃試験におけるショック振動を加えた結果、駆動系における不具合が見られた。原因としては、動画の中でも述べている通り、モータバッテリーからの電力供給ができていなかったことが考えられる。以下に、コネクタ部とバッテリー内部の電池の位置に微調整を施した後の機体の駆動系確認の動画のリンクを示す。

振動試験後 駆動系確認：<https://youtu.be/vJnb6g50QvI>

この動画にある通り、モータバッテリーのコネクタとバッテリー内部の電池の位置に微調整を施した機体はモータとサーボモータについて、正常に動作することを確認できた。

以上より、今回の試験で不備があったのはモータバッテリーからの電力供給部分であると考えられる。そのため、ARLISS本番では供給部における電池の位置ズレやコネクタの接触不良が生じないように、コネクタの修理、新しい電池ボックスの作成などの対策を行い、接続が安定することを確認する。

また、8月19日に再試験を予定しており、そこで再度振動試験(v4)と分離衝撃試験(v5)を行い、システム要求を満たすことを確認する。

[2回目(8/19)試験後追記]

1回目の試験をうけて、機体の電池ボックスを見直し、電池のずれが生じないように工夫を施した。具体的には、ボックス内に電池のずれが生じて空間が生まれそうな箇所に銅板をはさむことによって、電力供給の頑健性を増加させた。コネクタ部分についてもより、念入りにホットボンドなどによる補強を行った。

その結果、上にもある通り、すべてのチェック項目において問題なく動作し、駆動系、センサ類のチェック後における長距離走行も問題ないことが確認できたため、レギュレーションにて記載された振動と分離衝撃の負荷に耐える機体となったことが確認できた。

v5. 分離衝撃試験

- 目的

ロケットによる打ち上げを想定した際、ロケット分離時の衝撃によって親機に異常が生じないことを確認する。

- 試験内容

(V4)振動試験と同様の振動試験装置を用いて40 [G]のショック振動を与える。試験後、キャリアからの放出、着地、パラシュート分離、子機との通信確立までのシーケンス、親機が正常に動作することを確認する。衝撃の種類はARLISSのレギュレー

シヨンの1.4項目に従い, 負の方向にショック振動:[目標]392 [m/s²]->392/9.8 [1G]=40 [G]を与える.

- 結果
[1回目(7/29(金)実施)]
機体が得たショック振動の加速度推移を以下, 図6.5.1に示す. 横軸は時間 [sec], 縦軸は加速度 [G]である.

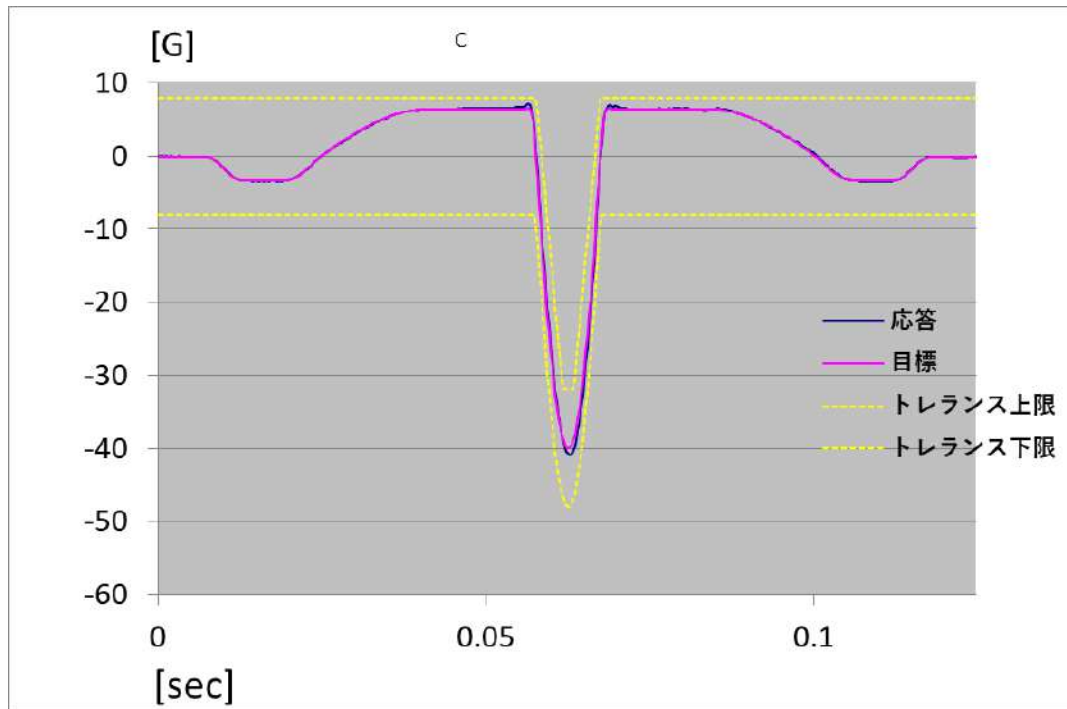


図6.5.1 ショック振動の加速度(1回目)

ショック振動を与えた後, 親機が正常に動作するかを確認した. 結果は(V4)振動試験と同様であるため, ここでは省略する. また, 本試験の実験動画は(V4)振動試験と同一である.

[2回目(8/19(金)実施)]

2回目の実施における機体得たショック振動の加速度推移を以下, 図6.5.2に示す.

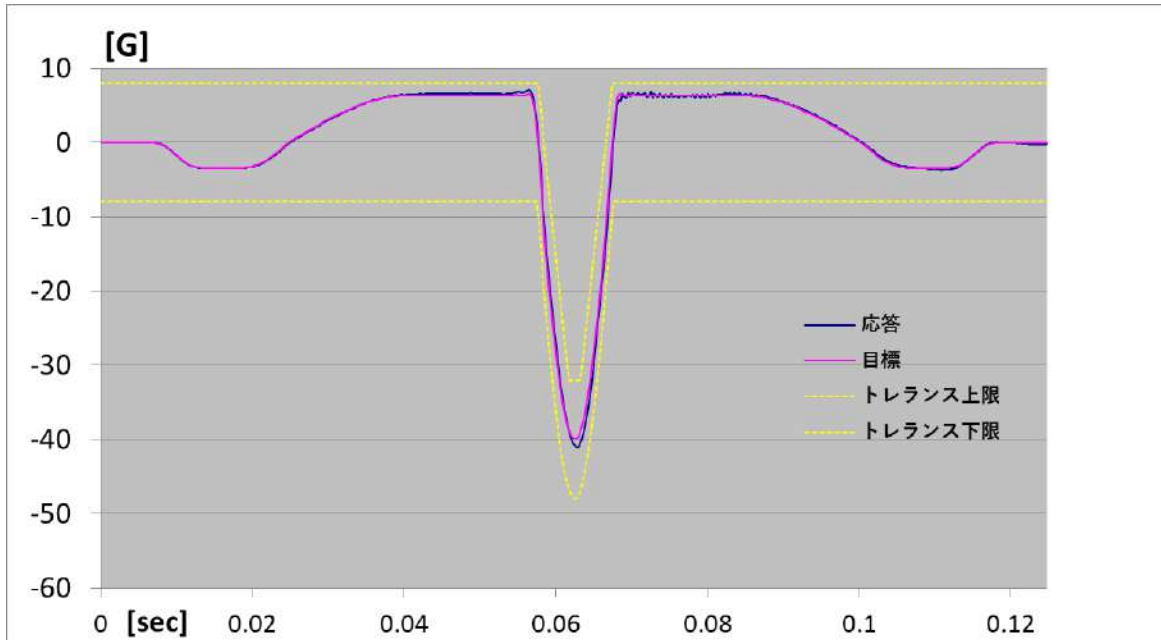


図6.5.2 ショック振動の加速度(2回目)

ショック振動を与えた後、親機が正常に動作するかを確認した。結果は(V4)振動試験と同様であるため、ここでは省略する。また、本試験の実験動画は(V4)振動試験と同一である。

- 考察
振動試験と同時にを行ったため、考察の記述は振動試験 (v5)に統合する。

v6. 開傘衝撃試験

- 目的
パラシュート開傘時の衝撃に親機本体とパラシュートとの結合部分が耐えられることを確認する。
- 試験内容
親機を固定し、パラシュートの紐を付けて自由落下させる。この時加速度センサのログを確認しパラシュート開傘時の衝撃(12 [G])に耐えられたかを確認する。衝撃を加えた後、パラシュート切り離しから走行までのシーケンスを実施し、全てのセンサ・動力系の動作確認、親機に破損がないかを確認する。なお、12 [G]という値は過去のARLISS・ACTSでの開傘時の加速度ログをもとに独自に設定したものである。また、本試験におけるセンサ、モータの出力は以下の基準を満たしていれば正常と扱う。

<センサ類>

- ・気圧センサ: 実験開始前に計測した気圧 ± 1 [hPa]
- ・9軸センサ: 機体の姿勢が静安時の加速度のノルムが0.95 [G]~1.05 [G](重力加速度1 [G] \pm 許容測定誤差5%, および、機体の姿勢によってxyz軸の値が変化している)
- ・光センサ: 通常時に光があたっていると判定(センサ値がHIGHを示す)し、手で光センサを覆ったときに光があたっていないと判定(センサ値がLOWを示す)する
- ・ブザー: 音のON/OFFができる
- ・GNSS: 1つ以上の衛星からGNSS情報を受信し、その座標が試験の実施地点を指している

- ・LoRa: 親機から子機と子機から親機への文字列の送受信を確認する

<モータ類>

- ・サーボモータ: 上下方向にスタビライザを制御できる
- ・モータ: 機体が走行可能な程度に回転する

(参考) 開傘衝撃試験で与える加速度(12 [G])の根拠

開傘衝撃のを決定するにあたって, ACTS2021のSalamanderチームの2回目の投下試験のログ(図6.6.1)を参考にした. グラフ内の時刻[11:27:12]における加速度の大きさを確認すると, 11.16 [G]であり, これをもとに12 [G]と設定した. なお, Salamanderチームが使用したパラシュートは本投下試験で使用しているパラシュートと同一であり, 同程度の開傘衝撃が加わることが想定される.

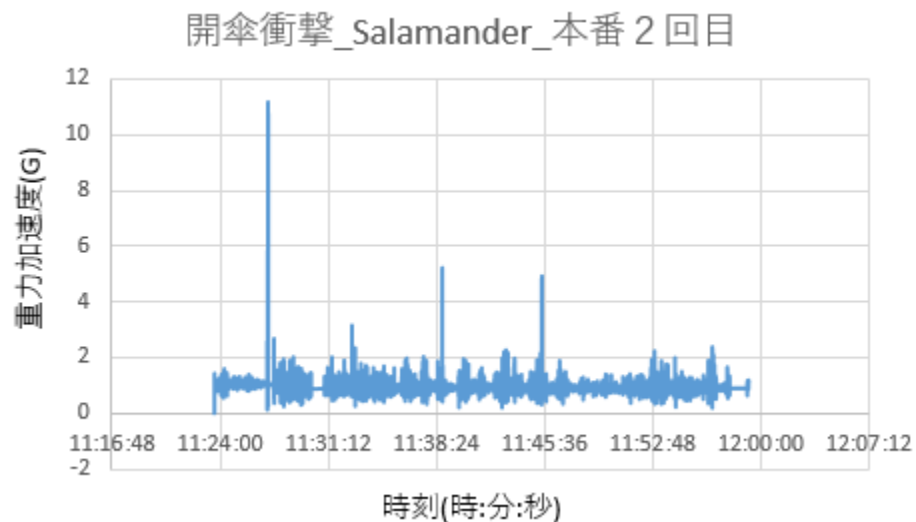


図6.6.1 開傘衝撃のログ ACTS2021 Salamander 2回目の投下試験より

- 結果
各試験におけるグラフを図6.6.2~図6.6.4に示す. 横軸は経過時間 [s], 縦軸は加速度のL2ノルム [G]を示す.

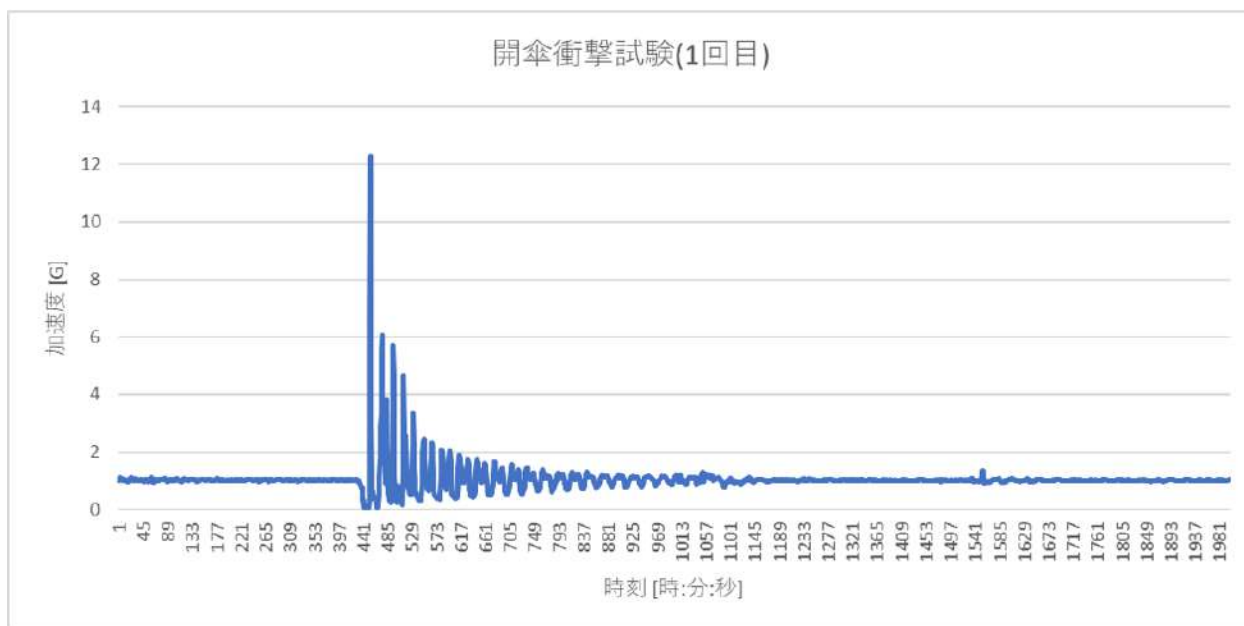


図 6.6.2 開傘衝撃試験1回目の加速度のグラフ

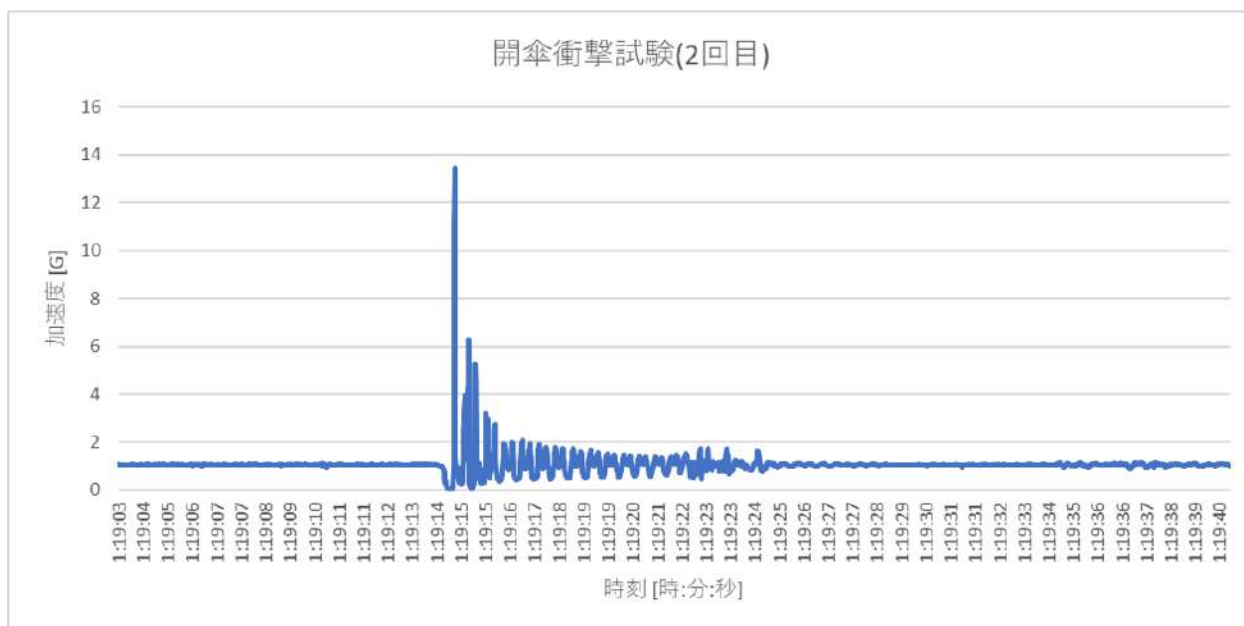


図 6.6.3 開傘衝撃試験2回目の加速度のグラフ

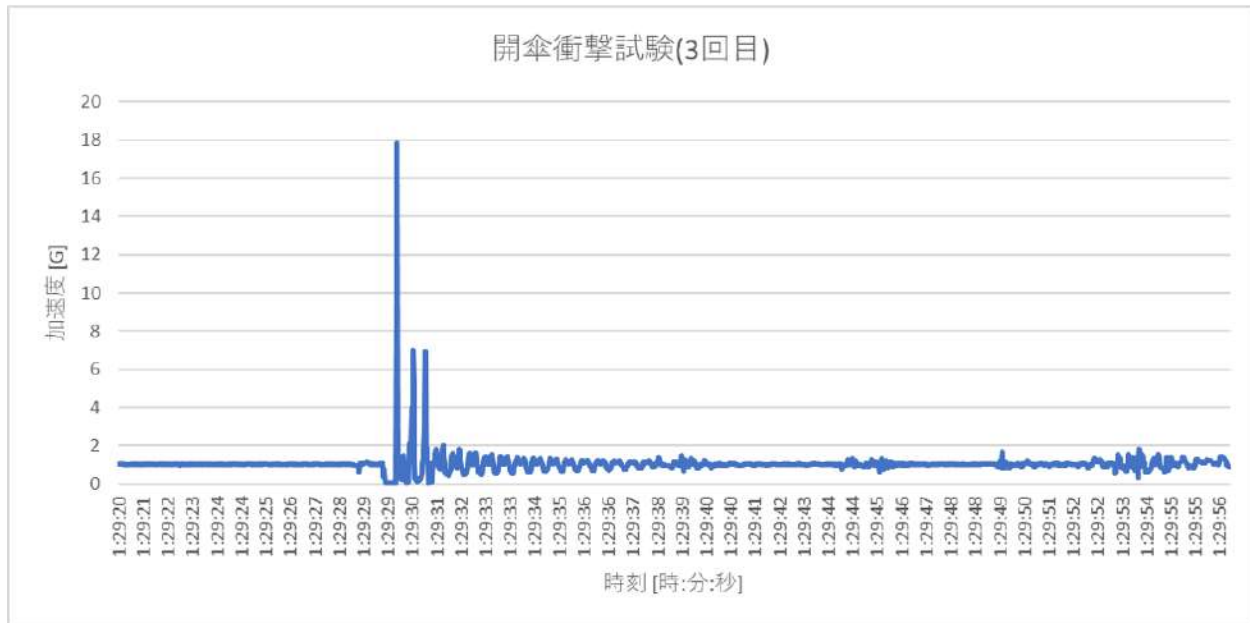


図 6.6.4 開傘衝撃試験3回目の加速度のグラフ

以下の表6.6.1に試行ごとの結果を示す。

表6.6.1 開傘衝撃試験の結果

回数	シーケンス動作	親機の外的損傷	センサ類	モータ類	最大加速度 [G]	YouTubeリンク
1回目	正常に遷移	なし	正常	正常	12.2592	https://youtu.be/crlCuh0b8v8
2回目	正常に遷移	なし	正常	正常	13.4387	https://youtu.be/Ns5SfvBZa0
3回目	正常に遷移	なし	正常	正常	17.8367	https://youtu.be/MMc1XjPR6M
成功率	100% (3/3)	100% (3/3)	100% (3/3)	100% (3/3)		

※開傘衝撃試験後のGNSSセンサが正常な値を取得していることを確認した。その時の動画を以下に載せる。

GNSSセンサ確認動画: <https://youtu.be/lwPy0Rfxjjw>

- 考察
以上より、親機はパラシュート開傘時にかかる最大予想衝撃である10 [G]の衝撃に耐えることが確認された。

v7. パラシュート投下試験

- 目的
キャリアから放出されたときに減速機構であるパラシュートが問題なく展開されて減速すること、また、落下速度のレギュレーション(5.0 [m/s]以上)を満たすことを確認する。
- 試験内容
電気通信大学西6号館7階地面からの高さ約27.3 [m]地点から、あらかじめキャリアに収納済みの「疑似親機」を放出する。この「疑似親機」は、実際に開発中の親機と同等の重量、形状、高さである。「疑似親機」を図6.7.1に示す。
本試験は親機自体の回路やソフトウェアには関与しない、あくまでパラシュートの性能および親機のハードウェア性能に対する試験であるため、本試験はこの「疑似親機」を用いて行っても実際の親機を用いた場合と同等の結果が得られるといえる。

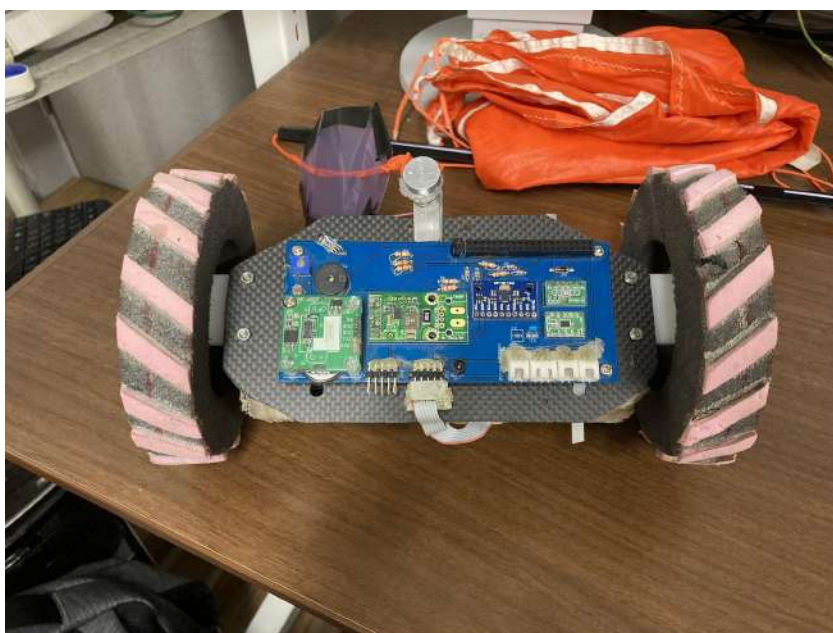


図6.7.1 疑似親機

本試験で確認することは以下の通りである。

- パラシュートの正常な開傘
- パラシュートによる減速

表6.7.1 各階間の通過時間

階	通過時間
7 → 6	00:09:56-00:10:37

6 → 5	00:10:37-00:11:12
5 → 4	00:11:12-00:11:30
4 → 3	00:11:30-00:12:04
3 → 2	00:12:04-00:12:40
2 → 1	00:12:40-00:13:19

(本審査時, 追記) 追試験によりパラシュートの減速により終端速度になっている試行が確認できたため, 表6.7.1をその試行のものに変更している. その試行の様子は, 下の表6.7.2と表6.7.3に5回目として, 結果を記載した.

これら2項目はビデオカメラで撮影した動画をもとに確認を取る. 特に落下時の速度に関しては, 図 6.7.2に示すように, パラシュートをつけた状態でキャリアから放出して落下し, 終端速度に達している高さからの通過時間を計測する. 実験動画から, 各階間を通過する時間を動画から算出した. (表6.7.1) これより, 4階以下では, パラシュートの減速により速度が同程度になっていることから, 終端速度になっている試行が確認できた. 建物3階相当(地上から8.4 [m])を親機が通過してから, 着地までにかかった時間を動画から計測し, 以下の式を用いて実験時の終端速度を計算する.

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

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

また, 自由落下時の終端速度は次の式より, 23.1 [m/s]と求められる.

$$v = \sqrt{2gh}$$

$$h = 27.3 \text{ [m]}, \quad g = 9.81 \text{ [m/s}^2\text{]}$$

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



図6.7.2 地面から3階までの高さ

- 結果
試験の結果を表6.7.2にまとめる. 4回全てでパラシュートの開傘が確認された.
表6.7.2 パラシュート投下試験の結果

試行回	パラシュートの開傘	YouTubeリンク
1回目	成功	https://youtu.be/NmcMuzZN7-c
2回目	成功	https://youtu.be/OwRJrntttz0
3回目	成功	https://youtu.be/-bRitHR0sgw
4回目	成功	https://youtu.be/hnDE0Lf5w5E
5回目	成功	https://youtu.be/8UIhkgmfAFI
成功率	100% (5/5)	-

また、落下速度の計算結果を表6.7.3に示す。各落下において動画から落下にかかるまでの時間を見積もり、この時間から速度を算出した。

表6.7.3 パラシュートの落下速度

試験回数	算出速度	動画該当箇所
1回目	8.4 [m] / 1.57 [s] \div 5.4 [m/s]	00:08:88~00:10:45
2回目	8.4 [m] / 1.27 [s] \div 6.6 [m/s]	00:10:73~00:12:00
3回目	8.4 [m] / 1.15 [s] \div 7.3 [m/s]	00:16:18~00:17:33
4回目	8.4 [m] / 1.63 [s] \div 5.2 [m/s]	00:11:67~00:13:30
5回目	8.4 [m] / 1.40 [s] = 6.0 [m/s]	00:11:16~00:12:56

落下速度の最大値は 7.3 [m/s] であり、これは自由落下時の速度である 23.1 [m/s] に比べて小さく、パラシュートによる減速が確認された。なお、パラシュートの落下速度で機体が耐えることができるかをv6の開傘衝撃試験とv12の着地衝撃試験にて確認する。

- 考察
キャリアから放出されたときに減速機構であるパラシュートが問題なく展開し、減速することが確認された。

v8. 長距離通信試験

- 目的
親機・子機に搭載されているLoRaの最大通信距離を把握し、機体のロスト対策としてLoRaが有効かどうか確認する。
- 試験内容
2台のLoRa間*で10秒ごとに通信を行っている状態から、通信不可能となるまでそれらの間隔を開ける。そして、通信が途絶えた時点でそれぞれのLoRaが存在する地点のGNSS情報から最大通信距離を導出し、ロスト対策としてLoRaが有効かどうか確認する。
*動画内ではPCと親機にLoRaが1台ずつ接続されている。この2台は本番で親機と子機に載せるものと同じである。
- 結果
CanSatとPC(地上局)の通信が切断された時のそれぞれのGNSS座標を表6.8.1に示す。図6.8.1はその座標を地図上に示したものである。

表6.8.1 長距離通信試験結果

PC(地上局)の座標	北緯:35.6424317 東経:139.5234383
CanSatの座標	北緯:35.6382133

	東経:139.5497117
最大通信可能距離	2.43km
試験開始時	https://youtu.be/xNb9ci-zh1M
試験終了時	https://youtu.be/WEVOyvStdpw



図6.8.1 長距離通信試験時のPCと親機の位置

- 考察
LoRaの最大通信距離を確認できた。ARLISS本番において、CanSatは上空約4 kmの高さにおいてロケットから放出され、上空約2.43 [km]の高さからGNSS座標を受信できると考えられる。v7. パラシュート投下試験よりパラシュート開傘後の終端速度の最大値は7.3 [m/s]であるから、上空2.43 [km] から着地までにかかる時間は約5分である。車で時速50 km程の速さ(5分で4 km走行可能)でCanSatを追跡するため、着地する時間前後までにCanSatを視認することが可能であると考えられ、ロスト対策できていると言える。

v9. 通信機電源ON/OFF試験

- 目的
ロケットの通信機器に悪影響を与えないよう、キャリア内でWi-FiおよびLoRaの電源が切れることを確認する。
- 試験内容
 - Wi-Fi
親機に搭載されているOrange Pi PCに差してあるWi-Fiモジュールをアクセスポイント化し、PCと通信している状態から、Wi-Fiの通信状態をONからOFFに切り替える。Wi-Fiの通信状態がOFFに切り替わることを、ターミナル上で自由にON/OFFできることを確認する。
まず、ローバプログラムのWi-Fi通信モードを変更する。ローバプログラムを起動し待機シーケンス(Waiting)に遷移するとWi-Fiの通信状態がOFFに切り替わることを、Wi-Fi Analyzerという周囲のアクセスポイントを可視化するソフトに親機のアクセスポイントが表示されなくなるにより確認する。次に、

ターミナル上でシーケンスをWaitingから移行させる。すると、Wi-Fiの通信状態がONに切り替わり、親機とPCが通信できていることを、Wi-Fi Analyzerに親機のアクセスポイントが表示されることにより確認する。

- LoRa

LoRaはOrange Pi PCのGPIOピンを制御することにより、スリープモード（電波の出力を停止）にすることが可能である。そのため、該当するGPIOピンを制御し、LoRaのスリープモードを自由にON/OFFできることを確認する。

まず、ターミナルにコマンドを入力し、地上局から機体側のLoRaにデータが送受信できることを確認する。次にローバのプログラム上からLoRaを通信状態をONからOFFにして送受信できないことを確認する。最後に通信状態をOFFからONにして再び送受信ができることを確認する。

- 結果

Wi-FiおよびLoRaの通信機電源ON/OFF試験の結果を表6.9.1に示す。具体的な結果は下記ようになった。

- Wi-Fi

Waitingのシーケンス内で本番と同様の手順でマイコンのWi-Fiの通信をOFFにした。実際にOFFにされたかをスマートフォンアプリのWi-Fiの電波を確認するツール(Wi-Fi Analyzer)を用いて確認した。また、Waitingシーケンスが終了した後、自動的にWi-Fiの通信機能がONになることも同様に確認した。

- LoRa

はじめは、親機のLoRaから地上局側のLoRaに向けてメッセージの送信が可能であった。親機側のローバプログラムにLoRaをスリープモードにするコマンドを入力したのち、親機側のLoRaからメッセージを送信すると地上局側に届いていないことが確認できた。次に、親機側のローバプログラムにLoRaのスリープモードを解除にするコマンドを入力したのち、親機のLoRaから地上局側のLoRaに向けてメッセージの送信が可能であることを確認した。

表6.9.1 通信機電源ON/OFF試験の結果

試験	結果	YouTubeリンク
Wi-Fiの電源ON/OFF	成功	https://youtu.be/cfm3YVHNuL0
LoRaの電源ON/OFF	成功	https://youtu.be/giuXQaihVw

- 考察

- Wi-Fi

問題なくWi-Fiの通信状態をONからOFFに切り替えと、シーケンス移行後OFFからONへできる。

- LoRa
問題なくLoRaの通信状態をONからOFF, OFFからONに切り替えができる。

v10. 通信周波数変更試験 (※安全面)

- 目的
他の無線通信と通信周波数が干渉する可能性がある場合に, 使用する通信周波数を変更できることを確認する。
- 試験内容
 - Wi-Fi
親機に搭載されているOrange Pi PCに差してあるWi-Fiモジュールに立ててあるアクセスポイントの周波数を, Wi-Fi Analyzerという周囲のアクセスポイントを可視化するソフトにより確認する。その後, 親機のアクセスポイントの周波数を変更し, Orange Pi PCを再起動する。そして, Wi-Fi Analyzerにより親機が立っているWi-Fiの周波数を変更されていることを確認する。
 - LoRa
周波数の一致している2つのLoRa(AとB)を用意する。Aの周波数を変更しBとデータ送受信ができなくなることを確認する。次にBのLoRaの周波数の変更操作をしてAと合わせることにより, 再びデータ送受信が可能になることを確認する。
- 結果
Wi-FiおよびLoRaの周波数変更試験の結果を表6.10.1に示す。また, 試験のYouTube動画の解説をそれぞれまとめた。

表6.10.1 通信周波数変更試験の結果

試験	結果	YouTubeリンク
Wi-Fiの周波数変更	成功	https://youtu.be/DAmqArNHHtC
LoRaの周波数変更	成功	https://youtu.be/giuXQaihVw

Wi-Fiに関する試験のYouTube動画の解説

0:00 ~ 試験の説明

0:30 ~ 親機のアクセスポイントの周波数(チャンネル番号)確認

→ チャンネル番号 7 であることを確認

0:56 ~ Wi-Fi周波数(チャンネル番号)の変更操作

→ チャンネル番号 7
から チャンネル番号 1 に変更

1:42 ~ create_apサービスの再起動

3:07 ~ 親機との接続が復帰

3:12 ~ Wi-Fi周波数が変更されていることを確認

→ チャンネル番号 1 であることを確認

LoRaに関する試験のYouTube動画の解説

0:00 ~ 試験の説明

0:30 ~ データ送受信の確認

1:27 ~ 片方のLoRa(A)の周波数を変更して送受信ができないことを確認

→ Aの周波数チャンネルを25(0x19) に変更

2:25 ~ もう一方のLoRa(B)の周波数を変更した周波数に合わせてLoRa間の通信ができることを確認

→ Bの周波数チャンネルを25(0x19) に変更

- 考察
Wi-Fiモジュール, LoRa通信モジュールともに周波数を変更できることを確認した。

v11. End-to-end試験

- 目的
親機がミッションを行う上での全シーケンスを一連の流れとして遂行できることを確認する。
- 試験内容
親機がミッションを行う上で質量測定, キャリア収納, 放出判定, 着地判定, パラシュート切り離し, 未踏エリア探索サブシーケンスを含むナビゲーションまでの全シーケンスを一連の流れとして自律的に制御できることを確認する。なお, 未踏エリア探索サブシーケンスの試行回数は2回とする。
- 結果
以下の表6.11.1に試行ごとの結果を示す。なお, 動画内におけるEnd-to-End試験の各段階(質量測定, キャリア収納など)の再生秒数は動画の概要欄に記載している。図6.11.1は試験における未踏エリア探索シーケンスの結果を著したものである。子機から親機に送信された地形の安全評価データは記録を取った座標を中心に半径5[m]の円で表され, 走行が難しいと判定された地点であるほど線が太く表示される。なお, 青・橙・紫色は3機それぞれの子機のデータである。(紫色の円は見辛いが一歩左にある橙色の円の下に存在している)。これらのデータを元に親機が算出した最適経路が青色のピンであり, 左上の点から右下に進行するように打点している。そして実際に親機が走行した点が茶色の線で表される。

表6.11.1 End-to-End試験の結果

試行回	質量	キャリア	放	着	パ	ナ	ミニマ	フル	YouTubeリ
-----	----	------	---	---	---	---	-----	----	----------

	測定	ア 収納・ 放出	出 判定	地 判定	ラ シ ュー ト 切 り 離 し	ビ ゲ ー シ ョ ン	ムサク セス判 定	サク セス判 定	ンク
1回目	成功	成功	成功	成功	成功	失敗	○	×	https://youtu.be/MZ6cigwoeOk



図 6.11.1 End-to-End試験1回目における子機の評価記録(子機別で青、橙、紫色の円)と導出した最適経路(青色ピン)と親機の走行経路(茶色の線)

- 考察

1回目に関して質量測定からパラシュート切り離しまでは順調に成功し、ナビゲーションに遷移した。しかし、ナビゲーションの未踏エリア探索シーケンスにおいて、通信確立から初期位置配置、子機の探索を経て親機が最良経路を導出するところまで成功したが、親機が正常にGNSSを取得できず、最良経路に沿った走行をすることができなかった。具体的な問題点としては、下記の3点が挙げられる。(※1.と2.に関しては最良経路導出試験1回目で確認されているものと同様である)

1. 紫色の円の子機と安定した通信が確立されず親機が受信した安全評価のデータが1点のみであった。
2. 親機から送信される子機の探索点が最初の数点のみで、子機が次の探索点を親機から送信されるのを待っていたため、探索を途中で停止しているような状態に陥った。

3. 親機が長期待機したことによりGNSSの平均点が待機点に集中し、走行開始後のGNSSが上手く更新されず、記録上のGNSSは待機点付近に残っているのに、動画上で図 6.11.1で左下に位置するグラウンドの端(草地)に突入してしまった。また、これは最良経路探索で生成された目標点が近いため走行中に座標更新が追いつかないことで起こる問題とも言える。

1.と2.についてはLoRaの通信環境に問題がある。具体的には複数LoRaによる輻輳した送受信能力に限界があることが分かっている。これを解決するために現在、親子機間のコマンドのデータ形式変更による文字長の圧縮やアルゴリズム内の不要なコマンド送信タイミングを検討し送受信回数の削減を試みている。3.についてはGNSSセンサが返す座標が複数衛星から受信した推定座標の平均である特性上、停止している状態から走行を開始してからしばらくは過去に受信した推定座標が平均座標を引っ張ってしまうためである。これに対し、走行開始時にGNSSの記録を一旦リセットすることでこの引っ張りを無くすことを検討している。

2. 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 (※ミッション面)

- 目的
着地衝撃の付与後であっても、親機に損傷が無くミッション継続可能であることを確認する。
- 試験内容
V7.パラシュート投下試験で算出したパラシュートを用いて落下した時の最大終端速度は7.30 [m/s]である。この終端速度をもとに本番で想定される着地時の衝撃を与えるために必要な高さを以下の式で求める。

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

$$h = \frac{1}{2 \times 9.8} \times 7.30^2 = 2.72 \text{ [m]}$$

以上より実際の落下衝撃に相当する高さは2.72 [m]である。本試験ではこの高さより高い約3 [m]から親機を自由落下させることで、本番同等の着地衝撃を与える。親機の着地後、全てのセンサ・動力系の動作確認、親機に破損がないかを確認する。

また、本試験におけるセンサ、モータの出力は以下の基準を満たしていれば正常と扱う。

<センサ類>

- ・気圧センサ: 実験開始前に計測した気圧 ± 1 [hPa]
- ・9軸センサ: 機体の姿勢が静安時の加速度のノルムが0.95 [G]~1.05 [G](重力加速度1 [G] \pm 許容測定誤差5%, および、機体の姿勢によってxyz軸の値が変化している

- ・光センサ: 通常時に光があたっていると判定(センサ値がHIGHを示す)し, 手で光センサを覆ったときに光があたっていないと判定(センサ値がLOWを示す)する
- ・ブザー: 音のON/OFFができる
- ・GNSS: 1つ以上の衛星からGNSS情報を受信し, その座標が試験の実施地点を指している
- ・LoRa: 親機から子機と子機から親機への文字列の送受信を確認する

<モータ類>

- ・サーボモータ: 上下方向または左右方向にスタビライザを制御できる
- ・モータ: 機体が走行可能な程度に回転する



図7.12.1 投下位置の計測の様子



図7.12.2 地面から投下位置までの距離

- 結果
以下の表7.12.1に試行ごとの結果を示す。

表7.12.1 着地衝撃試験の結果

試行回	親機の外的損傷	センサ類	モータ類	シーケンス動作	YouTubeリンク
1回目	なし	正常な値が取得できた。	正常に動作した。	正常に動作した。	https://youtu.be/trlRL8l-8hw
2回目	なし	正常な値が取得できた。	正常に動作した。	正常に動作した。	https://youtu.be/HyG4J3OMdbE
3回目	なし	正常な値が取得できた。	正常に動作した。	正常に動作した。	https://youtu.be/GWT85gvdU2Q
4回目	なし	正常な値が取得できた。	正常に動作した。	正常に動作した。	https://youtu.be/SnpKzFLbWNY
5回目	なし	正常な値が取得できた。	正常に動作した。	正常に動作した。	https://youtu.be/qdXpdXnCEPM
成功率	100% (5/5)	100% (5/5)	100% (5/5)	100% (5/5)	

※着地衝撃試験後のGNSSセンサが正常な値を取得していることを確認した。その時の動画を以下に載せる。

GNSSセンサ確認動画: https://youtu.be/rPdgc_NBD8k

- 考察
現状としては着地の衝撃がかかっても、外観状態、センサ値の取得、動力系、シーケンスの動作に問題がない。

v13. 走行性能確認試験

- 目的
CanSatが目的地に到達するために必要な走行性能を有することを確認する。
- 試験内容
砂地、砂利上での走行及び轍からの脱出走行を行い、途中で停止せずに走行をできることを確認する。なお、親機・子機でそれぞれ確認する。
- 結果
実施した結果を次の表7.13.1と7.13.2に示す。

表7.13.1 親機の走行性能試験の結果

試行回	走破性を測る状況	地面の走破	YouTubeリンク
1回目	平坦な砂地	成功	https://youtu.be/VYS-blr4yJc
2回目	砂利道	成功	https://youtu.be/be3KT8ci1qY
3回目	轍脱出(8cm)	成功	https://youtu.be/SsTVtdyb_m8
4回目		成功	https://youtu.be/1QhAfsMdl6s
5回目		成功	https://youtu.be/hdEFr-Z7N5c
成功率		100%	

表7.13.2 子機の走行性能試験の結果

試行回	走破性を測る状況	地面の走破	YouTubeリンク
1回目	平坦な砂地	成功	https://youtu.be/qghQRhu236E
2回目	砂利道	成功	https://youtu.be/xZZ6Bi_rV4I
3回目	轍脱出(8cm)	成功	https://youtu.be/OLzQ3DuHpKs
4回目		成功	https://youtu.be/9ym2kbtoLbs
5回目		成功	https://youtu.be/Yy8pSi0IVKw
成功率		100%	

- 考察
実験結果より、異なる路面状況においても走行を継続可能であり、CanSatがミッションを継続するために必要な走行性能を有していることが確認された。

v14. 電力耐久試験

- 目的

機体に搭載しているバッテリー容量が十分大きく、ミッション中に電池切れが発生しないことを確認する。

- 試験内容

円を描くように移動するようモータの強さの左右比を調整し、その場で機体を進行させる。制御開始のタイミングから制御が停止するまで(いずれか一方のバッテリーが切れるまで)の時間を計測する。十分なバッテリー容量を決定するための目安はローバープログラム起動から探索を含むナビゲーションシーケンスにおける動作の総和時間とする。具体的には、

- 親機起動から打ち上げを経て親機を発見するまでの時間: 120分
 - 待機シーケンス・落下シーケンス・パラシュート切り離しシーケンス・ナビゲーションシーケンスの通信確率のための待機はこれに含まれる
- ナビゲーションシーケンス
 - a. 子機を親機の周辺に配置してから通信確立: 最大3分(タイムアウト式)
 - b. 子機の初期配置: 最大3分(タイムアウト式)
 - c. 探索時間: 7分(タイマー式)
 - d. 次回初期配置: 目安5分
 - e. 予備時間: 10分

の目安時間を設定した。その上で、フルサクセスを満たすために探索を5回実行することを考えると、親機は $120 + 3 + 7 + 5 + 10 = 142$ 分で3時間を満たせば、親機を発見するまでの最大時間+予備時間が $120 + 10 = 130$ 分であることも加味して十分であるとする。子機はナビゲーションシーケンスの時間だけ考慮すればいいので、 $3 + 7 + 5 + 10 = 25$ 分で2時間を満たせば、予備時間が10分であることも加味して十分であるとする。

本番で使用するものと同じ以下の電池を使用する。二次電池は十分に充電したものを使用し、一次電池は未使用のものを使用する。

- 親機
 - 制御回路用電源
リチウムバッテリー(二次電池) x 1
 - モータ用電源
「Energizer Ultimate Lithium」(リチウムイオン一次電池, 単3形) x 6
- 子機
 - 制御回路用電源
リチウムバッテリー(二次電池) x 1
 - モータ用電源
「eneloop pro BK-4HCD/4SA」(二次電池, 単4形) x 4

- 結果

表7.14.1~7.14.2に試験開始前と試験終了後の親機、子機の電源電圧を示す。

表7.14.1 親機電力耐久試験の結果

	回路用電源	モータ, サーボモータ用電源	YouTubeリンク
試験開始前	4.18 V	10.78 V	https://youtu.be/Z-QDgQy0c-k
試験終了後	3.48 V	7.16 V	https://youtu.be/1w9pi3yZRP0

表7.14.2 子機電力耐久試験の結果

	回路用電源	モータ, サーボモータ用電源	YouTubeリンク
試験開始前	4.17 V	5.53 V	https://youtu.be/s7sMSoPppSA
試験終了後	3.62 V	5.10 V	https://youtu.be/xGwN4ZOI9Vg

- 考察
試験の結果から、機体に搭載しているバッテリー容量が十分大きく、ミッション中に電池切れが発生しないといえる。

v15. 反転横転復帰試験

- 目的
親機および子機が走行中に反転・横転した際に、走行可能な姿勢に復帰できることを確認する。
- 試験内容
 - 反転復帰
親機もしくは子機を反転させた状態から反転復帰シーケンスのプログラムを実行し、走行可能な姿勢に復帰させる。
 - 横転復帰
親機もしくは子機を横転させた状態から横転復帰シーケンスのプログラムを実行し、走行可能な姿勢に復帰させる。
- 結果

表7.15.1 親機反転復帰試験の結果

回数	結果	YouTubeリンク
1回目	成功	https://youtu.be/qsuu787t2Kk
2回目	成功	https://youtu.be/IA1RM64nS-o
3回目	成功	https://youtu.be/IFS1K2YdssE
4回目	成功	https://youtu.be/JpbGIzcGwOQ
5回目	成功	https://youtu.be/LyiKLAs9kvA
成功率	100%(5/5)	

表7.15.2 子機反転復帰試験の結果

回数	結果	YouTubeリンク
1回目	成功	https://youtu.be/sYdKEupKLtc
2回目	成功	https://youtu.be/mptq77DDVfk
3回目	成功	https://youtu.be/t5ml5aN6eWs
4回目	成功	https://youtu.be/FjEEODQTzqc
5回目	成功	https://youtu.be/bXqRHQ8ot7Q
成功率	100%(5/5)	

表7.15.3 親機横転復帰試験の結果

回数	結果	YouTubeリンク
1回目	成功	https://youtu.be/T2a0kWutgEw
2回目	成功	https://youtu.be/tt0EXa8T5SQ
3回目	成功	https://youtu.be/VX7PtN3Am-4
4回目	成功	https://youtu.be/lvO0R53l41g
5回目	成功	https://youtu.be/1gi5YOeRCok
成功率	100%(5/5)	

表7.15.4 子機横転復帰試験の結果

回数	結果	YouTubeリンク
1回目	成功	https://youtu.be/J7M3zRHFoXo
2回目	成功	https://youtu.be/LmPI3wZujxl
3回目	成功	https://youtu.be/362E2HdOqB4
4回目	成功	https://youtu.be/mO4lrXtFIW8
5回目	成功	https://youtu.be/rDTJx_eDUoc
成功率	100%	

- 考察
親機と子機の両方が反転・横転から走行可能な状態に 復帰できることが確認できた。

v16. 親機・子機間通信確立試験

- 目的
親機がそれぞれの子機と通信を確立し、ナビゲーションなど通信を要するシーケンスが実行可能な状態へ移行できることを確認する。
- 試験内容
親機を通信確立シーケンスに設定し、その周囲に子機群を親機と同じように通信確立シーケンスに設定した際に全ての子機が親機と通信確立できたことを、親機・子機双方のログを見て確認する。
- 結果

表7.16.1 親機・子機間通信確立試験の結果

回数	結果	YouTubeリンク
1回目	https://youtu.be/sK3JMy4JX0g	成功
2回目	https://youtu.be/rb5y9IAUPUY	成功
3回目	https://youtu.be/bGjLWa6RsTM	成功

- 考察
ミッション開始時の親機と子機の通信確立が行えることを試験で確認できた。

v17. 最良経路導出試験

- 目的
複数の子機が親機が未踏エリアを互いに衝突せずに探索し、路面状態や子機が走行困難な状況になってしまった状態などから得られた安全評価情報により親機が最良経路を導出できたか確認する。
- 試験内容
複数の子機に多数の軽微な障害物を設置したグラウンドなどの指定領域内を探索させ、安全評価情報を親機に送信する。そして、そこで得られた情報から親機が路面状態の悪い地点走行時の加速度ノルムの分散が大きい地点、子機が横転・反転・スタックした地点を除いて最良経路を導出出来ているかログにより確認する。なお、少なくとも1回目の試験は人の手を介さない状態で試験をするが、数回目の試験では意図的に子機を横転・反転状態にするなど、様々な状況においての経路算出などを確認する。

- 結果

以下の表7.17.1に試行ごとの結果を示す。また、図7.17.1は試験における未踏エリア探索シーケンスの結果を著したものである。子機から親機に送信された地形の安全評価データは記録を取った座標を中心に半径5mの円で表され、走行が難しいと判定された地点であるほど線が太く表示される。なお、青・橙・紫色は3機それぞれの子機のデータである。(紫色の円は見辛いが一番左にある橙色の円の下に存在している)。これらのデータを元に親機が算出した最適経路が青色のピンであり、左上の点から右下に進行するように打点している。そして実際に親機が走行した点が茶色の線で表される。

表7.17.1 最良経路導出試験の結果

回数	YouTubeリンク	結果
1回目	https://youtu.be/0q7QVasFxQE	成功 (ただし、子機の探索アルゴリズムに要改善点あり)



図 7.17.1 最良経路導出試験1回目における子機の評価記録(子機別で青、橙、紫色の円)と導出した最適経路(青色ピン)と親機の走行経路(茶色の線)

- 考察

1回目の結果に関して、図7.17.1を見ると、親機は子機からの安全評価データを受信し、それらの内、安全性が高いと評価された地点(円の線が細いもの)の中をなるべく通るような最良経路が算出されていることが分かる。そのような意味でこの試行を成功としている。しかし、いくつかの問題点も確認されている。具体的な問題点としては、下記の2点が挙げられる。(※これらの問題はEnd-to-End試験1回目で確認されている問題点の1.と2.と同様である)

1. 紫色の円の子機と安定した通信が確立されず親機が受信した安全評価のデータが1点のみであった
2. 親機から送信される子機の探索点が最初の数点のみで、子機が次の探索点を親機から送信されるのを待っていたため、探索を途中で停止しているような状態に陥った。

これらについてはLoRaの通信環境に問題がある。具体的には複数LoRaによる輻輳した送受信能力に限界があることが分かっている。これを解決するために現在、親機子機間のコマンドのデータ形式変更による文字長の圧縮やアルゴリズム内の不要なコマンド送信タイミングを検討し送受信回数の削減を試みている。

v18. 子機経路評価試験

- 目的
子機が地面を走行した際、路面状態や機体が走行困難な状況を踏まえて正しく安全評価出来ているか確認する。
- 試験内容
 - 路面状態評価
子機を様々な条件の地面を走行させ、路面状況や機体が突っかかってしまい走行できない状況など機体状態を正しく安全評価出来ているか機体のログを確認する。以下に試験で用いる路面の状態を示す。
 - 平らなコンクリート
 - 凹凸の激しいコンクリート
 - 突っかかりやすい地面(スタック状態)
 - 機体走行状態評価
子機を横転・反転状態にし、走行してはいけない地点としてその状態を正しく安全評価出来ているか機体のログを確認する。
- 結果
以下の表7.18.1に走行させた路面状況ごと、表7.18.2に機体状態ごとの結果を示す。

表7.18.1 路面状態評価の結果

路面状況	YouTubeリンク	安全評価(5秒ごと、低いほど安全)	結果(安全評価からの結論)
平らな地面	https://youtu.be/nLZSL1EnAmA	評価1: 563, 評価2: 1185, 評価3: 10	“安全性が比較的高い”ので正しい
凹凸の激しい地面	https://youtu.be/pRW28vHJxml	評価1: 4833, 評価2: 4913, 評価3: 6487	“安全性が比較的低い”ので正しい
突っかかりやすい	https://youtu.be/Jo	評価1: 4054,	“安全性が比較的低

地面(スタック状態)	dZ_vKQhLg	評価2: 2559	い”ので正しい
------------	-----------	-----------	---------

表7.18.2 機体走行状態評価の結果

機体状況	YouTubeリンク	結果(安全評価への記録)
横転状態	https://youtu.be/ZuzifOfKO7w	記録成功
反転状態	https://youtu.be/keTzHDUGET4	記録成功

- 考察
 - 路面状態の評価に関して、機体が走行しやすい”平らな地面”と機体が走行しにくい”凹凸の激しい地面”と”突っ掛かりやすい地面”では、後者の方が安全評価の記録が大きく出ていることが分かる。評価が大きい方が走行しにくい地面という設定であるため、機体に対して路面状況を区別した安全評価が出来ていることが分かる。
 - 走行状態の評価に関して、反転状態・横転状態の検知および安全評価の記録が出来ていることを確認した。また、路面状況の評価と区別して、走行してはいけない地点として判定できている。

v19. 子機初期位置配置試験

- 目的

子機が未踏エリア探索サブシーケンス開始時に親機に指定された位置に配置できるか確認する。
- 試験内容

親機の周辺に子機をバラバラに設置し、子機を初期位置に配置するよう親機から命令を送信する。そして、親機が指定した位置に子機が配置されたかどうか、親機が指定した座標が分かるログと子機が停止した座標を照らし合わせ確認する。
- 結果

以下の表7.19.1に試行ごとの結果を示す

表7.19.1 子機初期配置試験の結果

回数	YouTubeリンク	結果
1回目	https://youtu.be/lc1TuFtaFkg	成功
2回目	https://youtu.be/op9a9Tvz0_Q	成功

3回目	https://youtu.be/5L9xBnx5wrE	成功
成功率	-	100% (3/3)

- 考察

アルゴリズムの章で述べた子機初期配置誘導のアルゴリズムに沿った、子機が親機の前方で、親機とゴールを結んだ直線と垂直に並ぶように、親機が走行命令を送ることおよび子機がそれを遂行できることが確認できた。

v20. 子機数変動試験

- 目的

子機が反転状態や轍にはまるなどミッション続行不可能となってしまった場合、もしくは子機のバッテリー切れや親機との距離が離れすぎてしまったことにより親機との通信が困難となってしまった場合に親機がそれぞれの状態を判断しミッション続行可能な子機数を変更出来ているか確認する。また、通信が切断された子機が以降の動作を行わないよう停止される事も同時に確認する。

- 試験内容

親機と子機の通信を確立した後、子機を以下の状態にし親機がそれに応じてミッション続行可能な機体数を変更できたか確認する。また、通信切断された子機がナビゲーションシーケンスを停止していることをログにより確認する。

- 反転状態

子機を意図的に反転させる。

- 走行困難状態

子機を自力で脱出できないような窪みに設置する。

- 通信切断状態

子機のLoRaを抜き取り、通信不可能となってしまった状態を再現する。

- 結果

以下の表7.20.1に試行ごとの結果を示す。

表7.20.1 子機数変更試験の結果

	YouTubeリンク	結果
走行困難状態	https://youtu.be/ZvJkIgL2OwA	成功
通信切断状態	https://youtu.be/6an0_a-iCYs	成功

- 考察

子機が走行困難あるいは通信切断状態になった際に、前者では子機が走行困難

になったことを親機に伝達し記録すること、後者では時間経過で親機が子機と接続ができなくなったことを認識し切断状態にすること、そして、両者において親機がその子機を命令対象から外した上でミッションを続行できることを確認した。また、子機においても両者においてミッションを終了し、これ以上は動作をしないようにテスト用シーケンスに遷移したことが確認された。

v21. 指定経路走行試験

- 目的
親機が導出した最良経路を走行する際や、子機が親機が指定した座標まで走行する際などそれぞれの種類の機体が指定した経路を走行できるか確認する。
- 試験内容
走行させたい経路の座標に沿って親機・子機を走行させ、走行した際の座標ログと予め定めた経路を比較し走行経路が大きく外れていないことを確認する。
- 結果
以下の表7.21.1, 表7.21.2に親機, 子機それぞれの結果を示す。

表7.21.1 親機指定経路走行試験の結果

YouTubeリンク	結果
https://youtu.be/DNtxA4So4GQ	成功

表7.21.2 子機指定経路走行試験の結果

YouTubeリンク	結果
https://youtu.be/6JaJyv01fB8	成功

- 考察
図7.21.1は親機指定経路走行試験の結果である。指定した経路を青色のピンで表しており、S→1→2→3→4のGNSS座標の順で経路を指定した。茶色の線は実際に親機が指定経路を走行した際のGNSS記録であり、橙色の円がGNSSの誤差である半径約5[m]の円を表している。これを見ると、経路はS→1→2→3→4の点を辿るような軌跡を描いており、指定経路のGNSS座標は実際に走行した経路の誤差の円の中に含有されるようなものになっていることが分かる。つまり、親機は指定した経路を辿るような走行を実現している。
図7.21.2は子機指定経路走行試験の結果である。指定した経路を青色のピンで表しており、S→1→2→3→4→SのGNSS座標の順で経路を指定した。茶色の線は実際に子機が指定経路を走行した際のGNSS記録であり、橙色の円がGNSSの誤差である半径約5[m]の円を表している。これを見ると、経路はS→1→2→3→4→Sの点を辿るような軌跡を描いており、指定経路のGNSS座標は実際に走行した経路の誤差

の円の中に含有されるようなものになっていることが分かる。つまり、子機は指定した経路を辿るような走行を実現している。



図7.21.1 親機指定経路走行試験の指定経路(青)と実際の走行経路(茶・橙)



図7.21.2 子機指定経路走行試験の指定経路(青)と実際の走行経路(茶・橙)

v22. 制御履歴レポート作成試験

- 目的
ミッション後、規定された制御履歴レポートを提出するため、得られたログデータから制御履歴レポートを作成する。
- 試験内容
End-to-End試験で得られた制御履歴に関するログデータから提出用の制御履歴レポートを実際に作成する。
- 結果
表7.22.1はEnd-to-End試験1回目の親機の制御履歴をログデータから一部抜粋したものである。ミッションを進めるためのシーケンス移動やミッションのためのシーケンスの分析で大事となる箇所は赤字でコメントを加えた。また、この履歴やその他センサの記録から生成したEnd-to-End試験の未踏エリア探索サブシーケンスの可視化記録はEnd-to-End試験(V11)の結果の図7.22.1として掲載しているため、そちらを参照されたい。

表7.22.1 End-to-End試験1回目 制御履歴レポート

```

uccess: Setup WiringPi finished! -> プログラム起動
GPS Sensor is Ready!
Pressure Sensor is Ready!
PoseJudgment: ( rollOverThresholdY: 0.65, rollOverThresholdZ: 10, turnOverThresholdX: 9999,
turnOverThresholdZ: 0.5 )
Log file: log/20220805_0426_log_nineaxis1_e2e_0000_2.csv
Log file: log/20220805_0426_log_gps1_e2e_0000_2.csv
Log file: log/20220805_0426_log_pressure1_e2e_0000_2.csv
Log file: log/20220805_0426_log_lora1_e2e_0000_2.csv
This rover is used as PARENT rover!ld rovers num =
Log file: log/20220805_0426_log_safty_evaluation1_e2e_0000_2.csv
[1;34m

```



```

=== Team Dolphins ===

```

```

[0m[30m[43m Takadama Lab. ARLISS 2022 [0m

```

```

alias 'ww' = 'motor 100'
alias 'hh' = 'motor 0'
alias 'ss' = 'motor -100'
alias 'aa' = 'motor 0 100'
alias 'dd' = 'motor 100 0'
alias 'test' = 'start testing'
alias 'wait' = 'start waiting'
alias 'fall' = 'start falling'
alias 'sepa' = 'start para_separating'
alias 'poli' = 'start policy_learning'
alias 'navi' = 'start navigating'
alias 'detect' = 'start goal_detecting'
alias 'posetest' = 'start pose_detection_testing'
alias 'lighttest' = 'start light_detection_testing'
alias 'pidww' = 'magnet_pid drive'
alias 'pidhh' = 'magnet_pid stop;motor 0'
alias 'kaisan' = 'start testing;servo 1;nineaxis monitor start'
alias 'kaisan2' = 'nineaxis monitor stop'
alias 'seikaju' = 'start static_load_testing'
alias 'encoder' = 'encoder_l;encoder_r'
alias 'goto' = 'call moving_to_coordinate drive'

```

```

[5C[F

```

> wait
= start waiting
Waiting Mode :light, 1
[1m[04:28:45] [36mSequence: [32m'waiting' -> 待機シーケンス起動
[0mTime: 04:28:46
Light count: 0/10 (LOW)

Time: 04:28:47
Light count: 0/10 (LOW)

Time: 04:28:48
Light count: 0/10 (LOW)

Time: 04:28:49
Light count: 0/10 (LOW)

Time: 04:28:50
Light count: 0/10 (LOW)

Time: 04:28:51
Light count: 0/10 (LOW)

Time: 04:28:52
Light count: 0/10 (LOW)

Time: 04:28:53
Light count: 0/10 (LOW)

Time: 04:28:54
Light count: 0/10 (LOW)

Time: 04:28:55
Light count: 0/10 (LOW)

Time: 04:28:56
Light count: 0/10 (HIGH) -> 放出後で光を検知

Time: 04:28:57
Light count: 1/10 (HIGH)

Time: 04:28:58
Light count: 2/10 (HIGH)

Time: 04:28:59
Light count: 3/10 (HIGH)

Time: 04:29:00
Light count: 4/10 (HIGH)

Time: 04:29:01
Light count: 5/10 (HIGH)

Time: 04:29:02
Light count: 6/10 (HIGH)

Time: 04:29:03
Light count: 7/10 (HIGH)

Time: 04:29:04
Light count: 8/10 (HIGH)

Time: 04:29:05
Light count: 9/10 (HIGH)

[1m[04:29:05] [36mSequence: [32m'falling' -> 落下シーケンス起動

[0mTime: 04:29:06
Pressure count: 1/10 (1.01e+03 hPa)
Gyro count: 0/10
GPS position: 35.6404483, 139.5442433

Time: 04:29:07
Pressure count: 2/10 (1004.947 hPa)
Gyro count: 0/10
GPS position: 35.6404483, 139.5442450

Time: 04:29:08
Pressure count: 3/10 (1005.016 hPa)
Gyro count: 0/10
GPS position: 35.6404483, 139.5442450

Time: 04:29:09
Pressure count: 4/10 (1005.039 hPa)
Gyro count: 0/10
GPS position: 35.6404467, 139.5442467

Send data 35.6404483,139.5442467 by gps sender!

Time: 04:29:10
Pressure count: 5/10 (1004.894 hPa)
Gyro count: 0/10
GPS position: 35.6404483, 139.5442467

Time: 04:29:11
Pressure count: 6/10 (1005.104 hPa)
Gyro count: 0/10
GPS position: 35.6404500, 139.5442483

Time: 04:29:12
Pressure count: 7/10 (1005.077 hPa)
Gyro count: 0/10
GPS position: 35.6404567, 139.5442500

Time: 04:29:13
Pressure count: 8/10 (1005.009 hPa)
Gyro count: 0/10
GPS position: 35.6404583, 139.5442567

Time: 04:29:14
Pressure count: 9/10 (1005.116 hPa)

Gyro count: 0/10
GPS position: 35.6404600, 139.5442600

Send data 35.6404583,139.5442583 by gps sender!
Time: 04:29:15
Pressure count: 10/10 (1004.927 hPa)
Gyro count: 0/10
GPS position: 35.6404583, 139.5442583

Time: 04:29:16
Pressure count: 11/10 (1005.126 hPa)
Gyro count: 0/10
GPS position: 35.6404583, 139.5442600

Time: 04:29:17
Pressure count: 12/10 (1005.096 hPa)
Gyro count: 0/10
GPS position: 35.6404600, 139.5442550

Time: 04:29:18
Pressure count: 13/10 (1005.08 hPa)
Gyro count: 0/10
GPS position: 35.6404617, 139.5442567

Time: 04:29:19
Pressure count: 14/10 (1005.18 hPa)
Gyro count: 0/10
GPS position: 35.6404617, 139.5442550

Send data 35.6404617,139.5442550 by gps sender!
Time: 04:29:20
Pressure count: 15/10 (1005.058 hPa)
Gyro count: 0/10
GPS position: 35.6404617, 139.5442550

Time: 04:29:21 -> 着地後で静止状態を検知
Pressure count: 16/10 (1005.134 hPa)
Gyro count: 1/10
GPS position: 35.6404617, 139.5442533

Time: 04:29:22
Pressure count: 17/10 (1005.195 hPa)
Gyro count: 2/10
GPS position: 35.6404617, 139.5442500

Time: 04:29:23
Pressure count: 18/10 (1005.142 hPa)
Gyro count: 3/10
GPS position: 35.6404600, 139.5442483

Time: 04:29:24
Pressure count: 19/10 (1005.178 hPa)
Gyro count: 4/10
GPS position: 35.6404583, 139.5442467

Send data 35.6404583,139.5442450 by gps sender!

Time: 04:29:25

Pressure count: 20/10 (1005.103 hPa)

Gyro count: 5/10

GPS position: 35.6404583, 139.5442450

Time: 04:29:26

Pressure count: 21/10 (1005.231 hPa)

Gyro count: 6/10

GPS position: 35.6404567, 139.5442433

Time: 04:29:27

Pressure count: 22/10 (1005.094 hPa)

Gyro count: 7/10

GPS position: 35.6404567, 139.5442417

Time: 04:29:28

Pressure count: 23/10 (1005.168 hPa)

Gyro count: 8/10

GPS position: 35.6404550, 139.5442417

Time: 04:29:29

Pressure count: 24/10 (1005.138 hPa)

Gyro count: 9/10

GPS position: 35.6404533, 139.5442400

Send data 35.6404533,139.5442400 by gps sender!

Falling completed! (by gyro and pressure)

[1m[04:29:30] [36mSequence: [32m'para_separating' -> **パラシュート切り離しシーケンス**

[0mPara separating... (1/5)

Send data 35.6404500,139.5442400 by gps sender!

Para separating... (2/5)

Para separating... (3/5)

Send data 35.6404517,139.5442417 by gps sender!

Para separating... (4/5)

Send data 35.6404567,139.5442383 by gps sender!

Para separating... (5/5)

Para separating finished!

Error: Failed to get parameter 'navigating_method' from config file!

The config value of 'navigating_method' is

PolicyLearning Method is Magnet.

This rover is used as PARENT rover!ld rovers num =

Error: Failed to get parameter 'wait_avoid_child_time' from config file!

Log file: log/20220805_0426_log_safty_evaluation5_e2e_0000_2.csv

Log file: log/20220805_0426_log_policy_learning1_e2e_0000_2.csv

[1m[04:29:45] [36mSequence: [32m'policy_learning' > 'magnet_calibrating' -> **ナビゲーションシーケンス, 地磁気のキャリブレーション開始**

[0m[1m[04:29:45] [36mSequence: [32m'policy_learning' > 'magnet_calibrating' > 'waking'

[0mWaking finished!

[1m[04:29:51] [36mSequence: [32m'policy_learning' > 'magnet_calibrating'

[0mMagnetCalibrating: Turning right...

MagnetCalibrating: Turning left...

MagnetCalibrating: Turning stopped!

min: (-81, -231, -343)
max: (95, -58, -266)
Command:
nineaxis minmax -81 -231 -343 95 -58 -266

filtered min: (-81, -227, -338)
filtered max: (93, -59, -268)
Command:
nineaxis minmax -81 -227 -338 93 -59 -268
MagnetCalibrating has been finished!
[1m[04:30:03] [36mSequence: [32m'policy_learning' -> 未踏エリア探索サブシーケンス開始
[0mTime: 04:30:03
Type/Mode/Id: parent/START_PAIRING/0000 -> 親機・子機間通信確立開始
Current Azimuth: 280.889

** Start pairing **
Time: 04:30:04
Type/Mode/Id: parent/WAIT_PAIRING/0000
Current Azimuth: 281.151

Time: 04:30:05
Type/Mode/Id: parent/WAIT_PAIRING/0000
Current Azimuth: 283.066

[1CTime: 04:30:06
Type/Mode/Id: parent/WAIT_PAIRING/0000
Current Azimuth: 281.981

[3CTime: 04:30:07
Type/Mode/Id: parent/WAIT_PAIRING/0000
Current Azimuth: 284.821

Success! lora cmd push: rssi
** Successfully paired with 0001. (2 unit left) **
Success! lora cmd push: rssi
** Successfully paired with 0003. (1 unit left) **
Success! lora cmd push: rssi
** Successfully paired with 0002. (0 unit left) **
** Checked connection from children(3/3) ** -> 親機・子機間通信確立成功
[31m Minimum success clear! [m -> ミニマムサクセス成功
** Child(LoRaID: 0001) go to (35.6405108, 139.5443683) ** -> 子機初期位置配置開始
** Child(LoRaID: 0002) go to (35.6404272, 139.5443279) **
** Child(LoRaID: 0003) go to (35.6403436, 139.5442875) **
** Move children to initial position **
Success! lora cmd push: rec
Execute retransmission process toward 0001.
Success! lora cmd push: rec
Execute retransmission process toward 0002.
Success! lora cmd push: rec
Success! lora cmd push: done
** Moved ID: 0003 to init pos! (1/3) **
Success! lora cmd push: done
** Moved ID: 0001 to init pos! (2/3) **
Success! lora cmd push: done

```
** Moved ID: 0002 to init pos! (3/3) **
** Moved all children! ** -> 子機初期位置配置成功
essfeff95.28723, 15.00015
** Set search area! [origin: (35.6405760, 139.5442838), azimuth: 110.7878804, goal: (35.6402906,
139.5447417)] **
** Child(LoRaID: 0001) go to (35.6405348, 139.5443238) **
** Child(LoRaID: 0002) go to (35.6402058, 139.5446430) **
** Child(LoRaID: 0003) go to (35.6404073, 139.5447371) **
** Start children search! ** -> 子機探索命令開始
Success! lora cmd push: rec
Success! lora cmd push: done
** Child(LoRaID: 0001) finish searching! **
** Child(LoRaID: 0001) go to [LatLng: (35.6402756, 139.5445301), Vec: (32.6938110, 23.3523601)] **
Execute retransmission process toward 0002.
Success! lora cmd push: rec
Execute retransmission process toward 0001.
Success! lora cmd push: saft
Execute retransmission process toward 0001.
** Get safty evaluation data! [LoRaID: 0002, LatLNg: (35.6404000, 139.5443333), Vec: (11.1350339,
16.7264245), data: 87] ** -> 子機(2機目)の評価データを初受信
Success! lora cmd push: rec
Success! lora cmd push: saft
** Get safty evaluation data! [LoRaID: 0003, LatLNg: (35.6403250, 139.5442717), Vec: (8.8879923,
26.5093977), data: 281] ** -> 子機(3機目)の評価データを初受信
Success! lora cmd push: saft
** Get safty evaluation data! [LoRaID: 0002, LatLNg: (35.6404000, 139.5443333), Vec: (11.1350339,
16.7264245), data: 87] **
Success! lora cmd push: saft
** Get safty evaluation data! [LoRaID: 0002, LatLNg: (35.6403783, 139.5443867), Vec: (16.5089339,
17.2702939), data: 956] **
Success! lora cmd push: saft
** Get safty evaluation data! [LoRaID: 0002, LatLNg: (35.6403550, 139.5444450), Vec: (22.3603286,
17.8231904), data: 15] **
Success! lora cmd push: saft
** Get safty evaluation data! [LoRaID: 0002, LatLNg: (35.6403300, 139.5445000), Vec: (27.9998247,
18.6590253), data: 227] **
Success! lora cmd push: saft
** Get safty evaluation data! [LoRaID: 0002, LatLNg: (35.6402950, 139.5445450), Vec: (33.1887694,
20.8567587), data: 27] **
Success! lora cmd push: saft
** Get safty evaluation data! [LoRaID: 0002, LatLNg: (35.6402950, 139.5445450), Vec: (33.1887694,
20.8567587), data: 27] **
Success! lora cmd push: saft
** Get safty evaluation data! [LoRaID: 0002, LatLNg: (35.6402950, 139.5445450), Vec: (33.1887694,
20.8567587), data: 27] **
Success! lora cmd push: saft
** Get safty evaluation data! [LoRaID: 0001, LatLNg: (35.6405000, 139.5444100), Vec: (13.6713188,
3.8564611), data: 46] ** -> 子機(1機目)の評価データを初受信
Success! lora cmd push: saft
** Get safty evaluation data! [LoRaID: 0001, LatLNg: (35.6404750, 139.5444650), Vec: (19.3106577,
4.6922843), data: 931] **
Success! lora cmd push: saft

<中略>
```

```

** Get safty evaluation data! [LoRaID: 0002, LatLng: (35.6404900, 139.5444883), Vec: (21.7858382,
4.1365525), data: 94] **
Execute retransmission process toward 0003.
** Get safty evaluation data! [LoRaID: 0001, LatLng: (35.6403883, 139.5445417), Vec: (30.5259860,
12.8034762), data: 61] **
** Child(LoRaID: 0001) finish searching! ** -> 子機(1機目)が親機の指定した探索点の一つに到着
Execute retransmission process toward 0001. -> ここから子機(1機目)との接続が不安定に
Execute retransmission process toward 0003. -> ここから子機(3機目)との接続が不安定に
Execute retransmission process toward 0003.
Execute retransmission process toward 0001.
Execute retransmission process toward 0003.
Execute retransmission process toward 0003.
** Get safty evaluation data! [LoRaID: 0002, LatLng: (35.6404900, 139.5444883), Vec: (21.7858382,
4.1365525), data: 94] **

<中略>

** Get safty evaluation data! [LoRaID: 0002, LatLng: (35.6404900, 139.5444883), Vec: (21.7858382,
4.1365525), data: 94] **
** Get safty evaluation data! [LoRaID: 0002, LatLng: (35.6404900, 139.5444883), Vec: (21.7858382,
4.1365525), data: 94] **
Execute retransmission process toward 0001.
Execute retransmission process toward 0001.
Execute retransmission process toward 0001.
Execute retransmission process toward 0001.
Execute retransmission process toward 0001.
Execute retransmission process toward 0001.
Success! lora cmd push: rec
Success! lora cmd push: done
** Child(LoRaID: 0002) finish searching! ** -> 子機(2機目)が親機の指定した探索点の一つに到着
** Finished to search unexplored area! (search record num: 21) ** -> 親機が指定した探索時間が過ぎ、
子機の探索が完了したのでここから最適経路導出

## Cost ##
10000000, 46, 448, 449, 515, 308, 246, 10000000, 10000000, 10000000
10000000, 46, 488, 449, 515, 208, 88, 10, 10, 10000000
10000000, 87, 376, 303, 362, 173, 102, 12, 13, 10000000
29, 58, 376, 286, 399, 48, 39, 21, 16, 10000000
89, 88, 277, 349, 285, 142, 82, 79, 46, 10000000
89, 89, 142, 260, 168, 138, 76, 86, 56, 10000000

## Minimum cost ##
10000046, 92, 534, 983, 1387, 1280, 1140, 10000894, 10000894, 10000894
10000000, 46, 534, 871, 1241, 1166, 972, 894, 894, 10000894
0, 87, 422, 725, 984, 1029, 957, 884, 894, 10000894
29, 58, 434, 621, 1009, 855, 871, 892, 900, 10000897
119, 118, 335, 610, 807, 831, 910, 951, 939, 10000900
208, 208, 260, 521, 689, 827, 904, 990, 995, 10000939

## Shortest route ##
(35.6404920, 139.5442446)[(-0.0005206, 10.0001700)] -> (35.6404500, 139.5442250)[(-0.0002600,
14.9997280)] -> (35.6403921, 139.5442571)[(15.0006852, 19.9995869)] -> (35.6403182,
139.5443408)[(14.9999745, 24.9999609)] -> (35.6403023, 139.5443925)[(19.9998718, 24.9999413)] ->
(35.6403055, 139.5444343)[(29.9999622, 14.9999622)] -> (35.6403804, 139.5446063)[(35.0000299,
10.0000023)] -> (35.6403645, 139.5446580)[(40.0000645, 9.9999218)] -> (35.6402964,

```



```
139.5446087)[(39.9999290, 20.0000352)] -> (35.6403210, 139.5446808)[(45.0001148, 14.9999471)] ->
(35.6403041, 139.5447320)[(50.0004911, 14.9997195)] -> GOAL -> 導出された最適経路
** Start to move! ** -> 親機が最適経路移動を開始

[1m[04:43:31] [36mSequence: [32m'policy_learning' > 'escaping' -> 草地にスタック, サーボの草地用のフィ
ルム装着忘れから脱出不能と判断し中断
[0mRetrying escaping... (1/4)
Retrying escaping... (2/4)
Retrying escaping... (3/4)
```

- 考察

ミッション後, 規定された制御履歴レポートを提出するため, 得られたログデータから制御履歴レポートを作成できることが確認された. また, シーケンスのログから今後の改修すべき点(どの段階で問題が発生しているか, 原因となっている動作は何か)が分析しやすいログが得られていることが分かる. このログからシーケンスのアルゴリズムの改修をしEnd-to-End試験の成功率を高めていくことを目指す

第7章 ガントチャート(工程管理)

1. 各担当(ハード・ソフト・全体などの予定)

1.1. [チーム全体]

- 定例MTG: 毎週水曜日
- BBM完成: 5/24
- PM完成: 6/20
- 静荷重試験: 6/23~6/30
- 振動試験: 7月上旬
- 予備審査書提出: 7/7
- End-To-End試験: 7/10~9/7
- 本審査書提出: 8/8
- FM完成:

1.2. [ソフトウェア]

- 画像処理系完成: 6/20
- 制御系完成: 6/20
- 通信系完成: 6/1
- 反転・横転復帰試験: 7/2
- 通信機電源ON/OFF試験: 6/23~6/30

- 通信周波数変更試験:6/23~6/30
- 最良経路導出試験:7/1~7/7
- 親機・子機間通信確立試験:7/4
- 子機経路評価試験:7/8
- 子機初期位置配置試験:7/13
- 指定経路走行試験:7/8
- 子機数変更試験:7/15
- 長距離通信試験:7/15
- 制御履歴レポート試験:7/15

1.3. [ハードウェア]

- 概形決定:~6/2
- BBM用機体作成:~5/21
- PM用機体作成:6/2~6/20
- 質量試験:6/22~6/23
- キャリア収納試験:6/22~6/23
- パラシュート投下試験:6/23~6/30
- 走行性能確認試験:6/25
- 開傘衝撃試験:6/23~6/30
- 着地衝撃試験:6/23~6/30

1.4. [回路]

- 使用センサの選定:4/30
- BBMユニバーサル基板実装:5/24
- PMプリント基板設計:6/2
- PMプリント基板発注:6/6
- プリント基板実装:6/15
- 回路量産:8/10

2. ガントチャート

- チーム内の予定は以下のガントチャートで管理している
https://docs.google.com/spreadsheets/d/1Liq_D6biy94feRNC_DsqdkIPMmMJ1u1S0WUqk_Ri7f3M/edit?usp=sharing
- “本体”シートの方に現在のタスクが入力されている。
 各項目の進捗は進捗状況の欄で管理されており、それぞれ意味は以下のとおりである。

Backlog: まだ作業を開始していない

InProgress: 作業中

Test & Confirming: タスクが完了したかチェック

Done: 終了(灰色で塗られる)

Canceled: タスク中止(黒塗り, 不必要になったタスクの意)

第8章 責任教員所感(本審査時点)

本年は、複数の子機ローバが親機ローバの進む予定の場所を事前に探査し、その情報を親機ローバに送信することで、親機ローバが安全かつ最良の経路を計算して走行する方法に取り組んでいます。実験は一通り完了しており、ミッションの成功率が高まって参りましたが、子機ローバの個数が少なく、複数の子機ローバと親機の通信が不完全の部分があるので、その点の対策が必要かと思っています。この部分に関しては継続して指導して参りますので、ご審査の方、よろしくお願い致します。

第9章 大会結果報告

(1) 目的

本CanSatがキャリア放出判定からゴール判定およびミッション達成までのシーケンスを自律して制御できるかを確認する。それとともに、CanSat(親機)の着地後、子機との通信を確立し、シーケンスに沿って子機の先行探索時の走行データを活用しながら安全な経路の走行制御ができるか検証をした。

(2) 結果

ARLISSにおける大会結果をまとめる。表9.1はサクセスクライテリアの達成度を示している。

表9.1 サクセスクライテリアの達成度

	ミニマムサクセス	フルサクセス
1回目	○	-
2回目	○	△

- 投下1回目

＜サクセスクライテリア達成状況＞

ミニマムサクセス: ○

フルサクセス: X

＜制御概要＞

2022年9月14日

(OrangePiにおける時刻設定と実際の時刻に差があり、OrangePi内での投下時刻は07:23となってしまう。以下、OrangePiにおけるタイムスタンプを記載する)

07:08:20 プログラム起動

07:34:17 親機放出判定

07:44:28 親機着地判定/パラシュート切り離し処理開始

07:44:44 パラシュート切り離し処理終了/最適経路探索シーケンス開始

07:44:56 子機とのペアリング待機

(親機のタイヤが外れてしまったため、ここでミッション終了)

<機体の軌跡>

以下にGPSデータからプロットした機体の軌跡のデータを示す。赤い線が親機の軌跡を示す。また、右にある追加実験の部分にはタイヤが取れていたものの、回路が機能していたため、親機の実行プログラムを利用し、ミッション終了後に行った追加実験の軌跡である(説明は省略)。

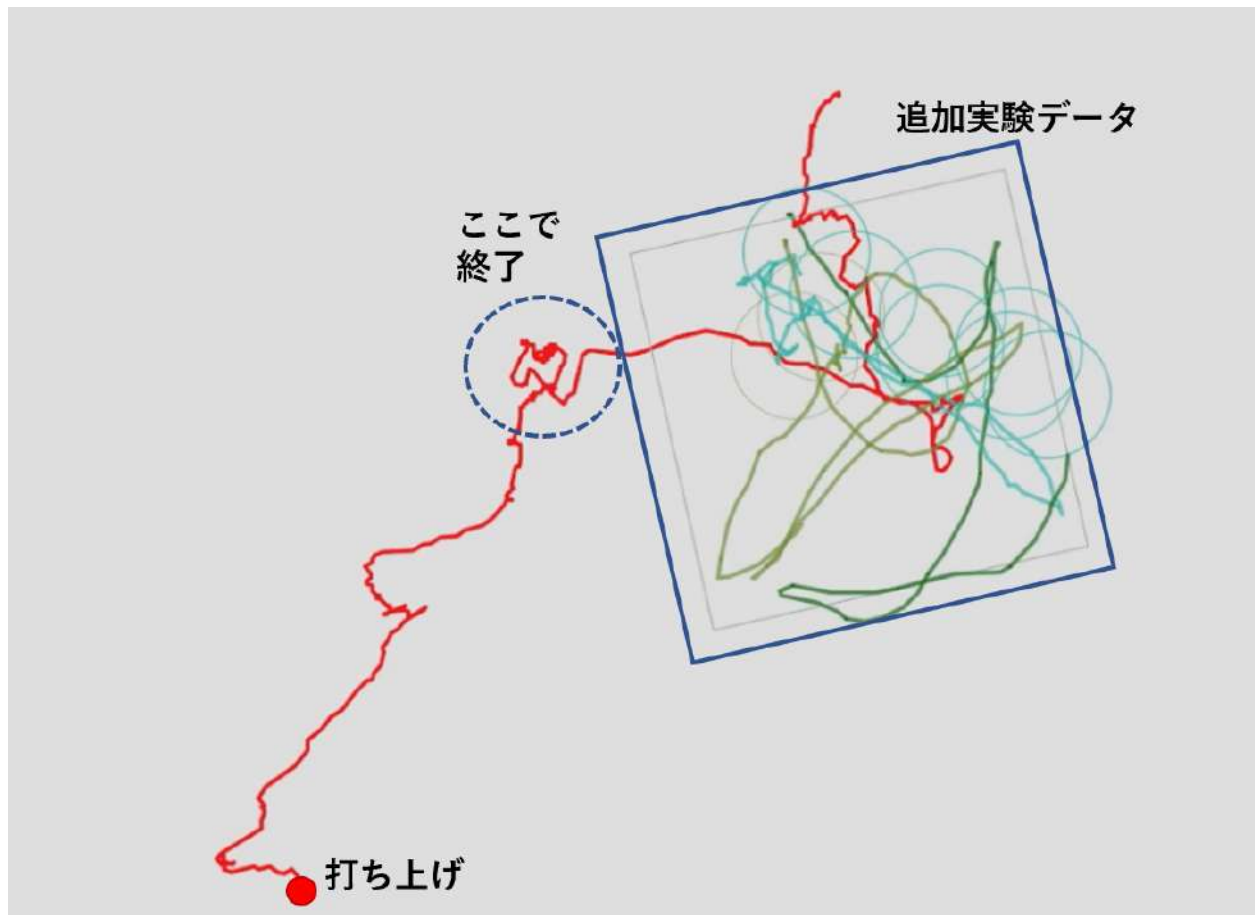


図9.1 投下1回目の軌跡

<制御履歴ログ>

```
=== Team Dolphins === ←プログラム起動
[0m[30m[43m   Takadama Lab.  ARLISS 2022
```

...設定表示のため省略...

[1m[07:08:19] [36mSequence: [32m'waiting' ← waitシーケンス(発射待ち)

[0mTime: 07:08:20

Light count: 0/10 (LOW)

Time: 07:08:21

Light count: 0/10 (LOW)

Time: 07:08:22

Light count: 0/10 (LOW)

...省略...

Time: 07:34:05

Light count: 0/10 (LOW)

Time: 07:34:06

Light count: 0/10 (LOW)

Time: 07:34:07

Light count: 0/10 (HIGH) ←放出, 光センサのカウント開始

Time: 07:34:08

Light count: 1/10 (HIGH)

Time: 07:34:09

Light count: 2/10 (HIGH)

...省略...

Time: 07:34:15

Light count: 8/10 (HIGH)

Time: 07:34:16

Light count: 9/10 (HIGH)

Wi-Fi has been restarted!

Warning: LoRaM92a has already stopped sleep mode!

LoRa has been restarted!

[1m[07:34:17] [36mSequence: [32m'falling' ←光センサによる放出判定完了 falling
シーケンススタート

[0mTime: 07:34:18

Pressure count: 1/10 (578.607 hPa)
Gyro count: 0/10
GPS position: 40.8750783

Time: 07:34:19
Pressure count: 2/10 (579.0068 hPa)
Gyro count: 0/10
GPS position: 40.8750783

...省略...

Time: 07:44:18
Pressure count: 600/10 (878.9586 hPa)
Gyro count: 0/10
GPS position: 40.9119950

Send data (40.9119950
Time: 07:44:19
Pressure count: 601/10 (878.8744 hPa)
Gyro count: 0/10
GPS position: 40.9120000

Time: 07:44:20
Pressure count: 602/10 (878.8754 hPa)
Gyro count: 1/10 ←着地, ジャイロカウントスタート
GPS position: 40.9120000

Time: 07:44:21
Pressure count: 603/10 (878.8691 hPa)
Gyro count: 2/10
GPS position: 40.9119983

...省略...

Time: 07:44:27
Pressure count: 609/10 (878.8713 hPa)
Gyro count: 8/10
GPS position: 40.9120333

Time: 07:44:28
Pressure count: 610/10 (878.8312 hPa)
Gyro count: 9/10
GPS position: 40.9120417


```

Falling completed! (by gyro and pressure) ←ジャイロカウント終了, パラセパレーティン
グシーケンス開始
[1m[07:44:29] [36mSequence: [32m'para_separating'
[0mPara separating... (1/5)
Para separating... (2/5)
Para separating... (3/5)
Para separating... (4/5)
Send data (40.9122200
Para separating... (5/5)
Para separating finished!
Error: Failed to get parameter 'navigating_method' from config file!
The config value of 'navigating_method' is
PolicyLearning Method is Magnet.
Error: Failed to get parameter 'wait_avoid_child_time' from config file!
Log file: log/20220914_0708_log_safty_evaluation2.csv
Gps auto reset is truned ON.
Log file: log/20220914_0708_log_policy_learning1.csv
[1m[07:44:44] [36mSequence: [32m'policy_learning' > 'magnet_calibrating' ←パ
ラセパレーティングシーケンス終了, 最適経路導出シーケンススタート
[0mMagnetCalibrating: Turning right...
MagnetCalibrating: Turning left...
MagnetCalibrating: Turning stopped!
min: (53
max: (85
Command:
nineaxis minmax 53 -199 26 85 -154 79
-----
filtered min: (54
filtered max: (82
Command:
nineaxis minmax 54 -196 29 82 -156 66
MagnetCalibrating has been finished!
[1m[07:44:56] [36mSequence: [32m'policy_learning'
[0mPush message: 6f626f6f74
** Start pairing ** ←親機 ペアリング待ち状態 (親機のタイヤが破損していたためミッション
終了)

```

- 投下2回目
 <サクセスクライテリア達成状況>
 ミニマムサクセス: ○

フルサクセス: △

<制御概要>

2022年9月15日

(OrangePiにおける時刻設定と実際の時刻に差があり, OrangePi内での投下時刻は11:24となってしまう。以下, OrangePiにおけるタイムスタンプを記載する)

10:57:12 プログラム起動

11:24:57 親機放出判定

11:33:43 親機着地判定/パラシュート切り離し処理開始

11:33:58 パラシュート切り離し処理終了/最適経路探索シーケンス開始

11:34:21 子機とのペアリング待機

11:40:18 子機03と通信確立確認

11:40:36 子機01と通信確立確認

11:40:42 子機02と通信確立確認(ミニマムサクセス達成)/初期位置配置サブシーケンススタート

11:42:19 子機02初期位置到着確認

11:44:14 子機03通信切断/子機探索開始

11:44:26 子機01通信切断

11:49:11 全ての子機と通信切断されたため, ミッション終了

<機体の軌跡>

GPSの不具合により, 正しいログが出力できず.

<制御履歴ログ>

```
=== Team Dolphins ===      ←プログラム起動
[0m[30m[43m   Takadama Lab.  ARLISS 2022

...設定表示のため省略...

[1m[10:57:12] [36mSequence: [32m'waiting' ←waitシーケンス
```

```
[0mTime: 10:57:13
Light count: 0/10 (LOW)

Time: 10:57:14
Light count: 0/10 (LOW)

...省略...

Time: 11:24:47
Light count: 0/10 (LOW)

Time: 11:24:48                                ←光センサカウントスタート
Light count: 0/10 (HIGH)

Time: 11:24:49
Light count: 1/10 (HIGH)

Time: 11:24:50
Light count: 2/10 (HIGH)

...省略...

Time: 11:24:55
Light count: 8/10 (HIGH)

Time: 11:24:56
Light count: 9/10 (HIGH)

Wi-Fi has been restarted!
Warning: LoRaM92a has already stopped sleep mode!
LoRa has been restarted!
[1m[11:24:57] [36mSequence: 'falling' ←放出判定DONE fallingシーケンススタート
[0mTime: 11:24:58
Pressure count: 1/10 (623.319 hPa)
Gyro count:      0/10
GPS position:    40.8708983    -119.102085

Time: 11:24:59
Pressure count: 2/10 (623.7852 hPa)
Gyro count:      0/10
GPS position:    40.8710217    -119.1020417

...省略...
```

```
Time: 11:33:33
Pressure count: 515/10 (879.14 hPa)
Gyro count:      0/10
GPS position:    40.8760000    -119.08356

Time: 11:33:34
Pressure count: 516/10 (879.141 hPa)
Gyro count:      1/10          ←ジャイロカウントスタート
GPS position:    40.8760367    -119.083515

Time: 11:33:35
Pressure count: 517/10 (879.1457 hPa)
Gyro count:      2/10
GPS position:    40.8760650    -119.0834867

...省略...

Time: 11:33:41
Pressure count: 523/10 (879.1013 hPa)
Gyro count:      8/10
GPS position:    40.8762783    -119.0831667

Time: 11:33:42
Pressure count: 524/10 (879.1646 hPa)
Gyro count:      9/10
GPS position:    40.8763083    -119.0830733

Send data (40.8763383    -119.0829817) by gps sender!
Falling completed! (by gyro and pressure)
[1m[11:33:43] [36mSequence: [32m'para_separating'    ←パラセパレーティングシーケ
ンススタート
[0mPara separating... (1/5)
Para separating... (2/5)
Para separating... (3/5)
Para separating... (4/5)
Send data (40.8761617    -119.0837650) by gps sender!
Para separating... (5/5)
Para separating finished!    ←パラセパレーティング終了
Error: Failed to get parameter 'navigating_method' from config file!

[1m[11:33:58] [36mSequence: [32m'policy_learning' > 'magnet_calibrating'←最
適経路探索シーケンススタート
```

```

[0m[1m[11:33:58] [36mSequence: [32m'policy_learning' > 'magnet_calibrating' >
'waking'
[0mWaking finished!
[1m[11:34:03] [36mSequence: [32m'policy_learning' > 'magnet_calibrating'

[0mMagnetCalibrating: Turning right...
MagnetCalibrating: Turning left...
MagnetCalibrating: Turning stopped!
min: (-36  44  -426)
max: (113  177  -368)
Command:
nineaxis minmax -36 44 -426 113 177 -368
-----
filtered min: (-32  48  -420)
filtered max: (98 172  -379)
Command:
nineaxis minmax -32 48 -420 98 172 -379
MagnetCalibrating has been finished!
[1m[11:34:16] [36mSequence: [32m'policy_learning'
[0mPush message: 6f626f6f74
** Start pairing **
Time: 11:34:21
Type/Mode/Id: parent/WAIT_PARING/0000 ←ペアリング待ち

Time: 11:34:27
Type/Mode/Id: parent/WAIT_PARING/0000

Send data (40.8761333 -119.0838300) by gps sender!
Time: 11:34:33
Type/Mode/Id: parent/WAIT_PARING/0000

...省略...

Time: 11:39:47
Type/Mode/Id: parent/WAIT_PARING/0000

Send data (40.8771467 -119.0854183) by gps sender!
Time: 11:39:53
Type/Mode/Id: parent/WAIT_PARING/0000

04000001234567000123456700180d
RoverID: 3 Mode: SYN_RECEIVED AckNum: 1 SynNum: 0
SndUna: 0 SndNxt: 1 RcvNxt: 1 canReceive: 0

```

```

Ack: 0      Syn: 1      Fin: 0      RecAckNum: 0      RecSynNum: 0
RecSec: 0
pause
Send!
Time: 11:39:59
Type/Mode/Id: parent/WAIT_PARING/0000

Time: 11:40:06
Type/Mode/Id: parent/WAIT_PARING/0000

04000001234567000123456700180d
RoverID: 1   Mode: SYN_RECEIVED   AckNum: 1   SynNum: 0
SndUna: 0    SndNxt: 1   RcvNxt: 1   canReceive: 0
Ack: 0      Syn: 1      Fin: 0      RecAckNum: 0      RecSynNum: 0
RecSec: 0
Time: 11:40:12
Type/Mode/Id: parent/WAIT_PARING/0000

04000001234567000123456700180d
RoverID: 2   Mode: SYN_RECEIVED   AckNum: 1   SynNum: 0
SndUna: 0    SndNxt: 1   RcvNxt: 1   canReceive: 0
Ack: 0      Syn: 1      Fin: 0      RecAckNum: 0      RecSynNum: 0
RecSec: 0
Time: 11:40:18
Type/Mode/Id: parent/WAIT_PARING/0000

80000101234567010123456701050f727373698b1e
Connection established with LoRaID 0003! ←子機03と通信確立確認
RoverID: 3   Mode: ESTABLISHED   AckNum: 2   SynNum: 1
SndUna: 1    SndNxt: 2   RcvNxt: 2   canReceive: 0
Ack: 1      Syn: 0      Fin: 0      RecAckNum: 1      RecSynNum: 1
RecSec: 1
Time: 11:40:24
Type/Mode/Id: parent/WAIT_PARING/0000

Send!
Time: 11:40:30
Type/Mode/Id: parent/WAIT_PARING/0000

Time: 11:40:36
Type/Mode/Id: parent/WAIT_PARING/0000

80000101234567010123456701050f727373698b1e

```



```

Connection established with LoRaID 0001! ←子機01と通信確立確認
RoverID: 1   Mode: ESTABLISHED       AckNum: 2   SynNum: 1
SndUna: 1    SndNxt: 2   RcvNxt: 2   canReceive: 0
Ack: 1       Syn: 0      Fin: 0      RecAckNum: 1   RecSynNum: 1
RecSec: 1
Time: 11:40:42
Type/Mode/Id: parent/WAIT_PARING/0000

80000101234567010123456701050f727373698b1e
Connection established with LoRaID 0002! ←子機02と通信確立確認
RoverID: 2   Mode: ESTABLISHED       AckNum: 2   SynNum: 1
SndUna: 1    SndNxt: 2   RcvNxt: 2   canReceive: 0
Ack: 1       Syn: 0      Fin: 0      RecAckNum: 1   RecSynNum: 1
RecSec: 1
** Checked connection from children(3/3) **
[31m Minimum success clear! [m ←通信確立したことにより, ミニマムサクセスを満たした

Push message: 1f676f746f2a34302e383737303539362a2d3131392e30383532363836

** Child(LoRaID: 0001) go to (40.8770596   -119.0852686) ** ←各子機へ探索方向
を指示
Push message: 2f676f746f2a34302e383737303432342a2d3131392e30383531353139

** Child(LoRaID: 0002) go to (40.8770424   -119.0851519) **
Push message: 3f676f746f2a34302e383737303235332a2d3131392e30383530333534

** Child(LoRaID: 0003) go to (40.8770253   -119.0850354) **
** Move children to initial position ** ←初期位置配置サブシーケンススタート
Time: 11:40:48
Type/Mode/Id: parent/WAIT_TO_MOVE_CHILDREN_INIT_POS/0000

1.01235E+24
RoverID: 3   Mode: ESTABLISHED       AckNum: 2   SynNum: 1
SndUna: 1    SndNxt: 2   RcvNxt: 2   canReceive: 1
Ack: 0       Syn: 0      Fin: 0      RecAckNum: 1   RecSynNum: 2
RecSec: 0
Time: 11:40:54
Type/Mode/Id: parent/WAIT_TO_MOVE_CHILDREN_INIT_POS/0000

Time: 11:41:00

```

Type/Mode/Id: parent/WAIT_TO_MOVE_CHILDREN_INIT_POS/0000

Send!

Time: 11:41:06

Type/Mode/Id: parent/WAIT_TO_MOVE_CHILDREN_INIT_POS/0000

8000020123456702012345670007f6

RoverID: 1 Mode: ESTABLISHED AckNum: 2 SynNum: 2

SndUna: 2 SndNxt: 3 RcvNxt: 2 canReceive: 1

Ack: 1 Syn: 0 Fin: 0 RecAckNum: 2 RecSynNum: 2

RecSec: 0

Time: 11:41:12

Type/Mode/Id: parent/WAIT_TO_MOVE_CHILDREN_INIT_POS/0000

8000020123456702012345670007f6

RoverID: 2 Mode: ESTABLISHED AckNum: 2 SynNum: 2

SndUna: 2 SndNxt: 3 RcvNxt: 2 canReceive: 1

Ack: 1 Syn: 0 Fin: 0 RecAckNum: 2 RecSynNum: 2

RecSec: 0

Time: 11:41:18

Type/Mode/Id: parent/WAIT_TO_MOVE_CHILDREN_INIT_POS/0000

Time: 11:41:24

Type/Mode/Id: parent/WAIT_TO_MOVE_CHILDREN_INIT_POS/0000

Time: 11:41:30

Type/Mode/Id: parent/WAIT_TO_MOVE_CHILDREN_INIT_POS/0000

Send!

Time: 11:41:36

Type/Mode/Id: parent/WAIT_TO_MOVE_CHILDREN_INIT_POS/0000

Time: 11:41:42

Type/Mode/Id: parent/WAIT_TO_MOVE_CHILDREN_INIT_POS/0000

```
Time: 11:41:48
Type/Mode/Id: parent/WAIT_TO_MOVE_CHILDREN_INIT_POS/0000

8000020123456702012345670007f6
RoverID: 2   Mode: ESTABLISHED       AckNum: 2   SynNum: 2
SndUna: 2    SndNxt: 3   RcvNxt: 2   canReceive: 1
Ack: 1       Syn: 0      Fin: 0      RecAckNum: 2   RecSynNum: 2
RecSec: 0
Time: 11:41:55
Type/Mode/Id: parent/WAIT_TO_MOVE_CHILDREN_INIT_POS/0000

Time: 11:42:01
Type/Mode/Id: parent/WAIT_TO_MOVE_CHILDREN_INIT_POS/0000

Send!
Time: 11:42:07
Type/Mode/Id: parent/WAIT_TO_MOVE_CHILDREN_INIT_POS/0000

Time: 11:42:13
Type/Mode/Id: parent/WAIT_TO_MOVE_CHILDREN_INIT_POS/0000

Time: 11:42:19
Type/Mode/Id: parent/WAIT_TO_MOVE_CHILDREN_INIT_POS/0000

000002012345670201234567021f050f7372756e2a332a34302e383738303036372a2d3131392
e303836313238330f646f6e654e37
RoverID: 2   Mode: ESTABLISHED       AckNum: 3   SynNum: 2
SndUna: 2    SndNxt: 3   RcvNxt: 3   canReceive: 1
Ack: 0       Syn: 0      Fin: 0      RecAckNum: 2   RecSynNum: 2
RecSec: 1
Success! lora cmd push: srun
Success! lora cmd push: done
** Moved ID: 0002 to init pos! (1/3) **  ←1台初期位置到着
Time: 11:42:25
Type/Mode/Id: parent/WAIT_TO_MOVE_CHILDREN_INIT_POS/0000

...省略...
```

Time: 11:44:14

Type/Mode/Id: parent/WAIT_TO_MOVE_CHILDREN_INIT_POS/0000

Warning: Forced disconnection due to timeout! (LoRaID: 0003) ←1台通信切断

Time: 11:44:20

Type/Mode/Id: parent/WAIT_TO_MOVE_CHILDREN_INIT_POS/0000

Time: 11:44:26

Type/Mode/Id: parent/WAIT_TO_MOVE_CHILDREN_INIT_POS/0000

2.01235E+24

RoverID: 2 Mode: ESTABLISHED AckNum: 3 SynNum: 2

SndUna: 2 SndNxt: 3 RcvNxt: 3 canReceive: 1

Ack: 0 Syn: 0 Fin: 0 RecAckNum: 2 RecSynNum: 3

RecSec: 0

Warning: Forced disconnection due to timeout! (LoRaID: 0001) ←もう1台通信切断

** Moved all children! ** ←2台通信切断したため、子機1台で経路評価サブシーケンスへ

comebackGoal: 2.68e+03 15

** Set search area! [origin: (40.8768714 -119.0851468) azimuth:
10.5826873 goal: (40.8771116 -119.0849062)] **

Push message: 6f706f732a34302e383736383436372a2d3131392e30383439373137

** Start children search! ** ←子機探索スタート

Time: 11:44:32

Type/Mode/Id: parent/CHILDREN_SEARCH/0000

Time: 11:44:38

Type/Mode/Id: parent/CHILDREN_SEARCH/0000

Time: 11:44:44

Type/Mode/Id: parent/CHILDREN_SEARCH/0000

Send!

Time: 11:44:50

Type/Mode/Id: parent/CHILDREN_SEARCH/0000

Time: 11:44:56
Type/Mode/Id: parent/CHILDREN_SEARCH/0000

2.01235E+24
RoverID: 2 Mode: ESTABLISHED AckNum: 3 SynNum: 2
SndUna: 2 SndNxt: 3 RcvNxt: 3 canReceive: 1
Ack: 0 Syn: 0 Fin: 0 RecAckNum: 2 RecSynNum: 3
RecSec: 0
Time: 11:45:02
Type/Mode/Id: parent/CHILDREN_SEARCH/0000

Time: 11:45:08
Type/Mode/Id: parent/CHILDREN_SEARCH/0000

Time: 11:45:14
Type/Mode/Id: parent/CHILDREN_SEARCH/0000

Push message: 6f706f732a34302e383736393330302a2d3131392e30383439393137

** Send my position data! **
Send!
Time: 11:45:20
Type/Mode/Id: parent/CHILDREN_SEARCH/0000

Time: 11:45:26
Type/Mode/Id: parent/CHILDREN_SEARCH/0000

Time: 11:45:32
Type/Mode/Id: parent/CHILDREN_SEARCH/0000

00000201234567030123456701126f736166742a312a2d332a3332322a313537cc67

RoverID: 2 Mode: ESTABLISHED AckNum: 3 SynNum: 2
SndUna: 2 SndNxt: 3 RcvNxt: 3 canReceive: 1
Ack: 0 Syn: 0 Fin: 0 RecAckNum: 2 RecSynNum: 3
RecSec: 0
Success! lora cmd push: saft
** Get safty evaluation data! [LoRaID: 0002 LatLng: (40.8780664
-119.0861667) Vec: (114.9950320 -108.8124561) RoverState:
STUCKED] ** ←子機から経路評価のデータを受信
Time: 11:45:39
Type/Mode/Id: parent/CHILDREN_SEARCH/0000

```
Time: 11:45:45
Type/Mode/Id: parent/CHILDREN_SEARCH/0000

...省略(いくつかデータを受信しながら探索を継続)...

Time: 11:48:46
Type/Mode/Id: parent/CHILDREN_SEARCH/0000

Time: 11:48:52
Type/Mode/Id: parent/CHILDREN_SEARCH/0000

Warning: Forced disconnection due to timeout! (LoRaID: 0002)← 最後の1機との通信が切断
** Terminate because the number of connected children has reached 0 **

Error: Connection does not exist!
Time: 11:48:58
Type/Mode/Id: parent/ABORT/0000

Send!
Time: 11:49:04
Type/Mode/Id: parent/ABORT/0000

Time: 11:49:11
Type/Mode/Id: parent/ABORT/0000

** Abort policy learning sequence! **
** Closed connections with all children! **← すべての子機との通信が切れたためミッションを終了
```

- 追加実験時の軌跡

2回目のミッション終了後、GPSの機能が正常になっていることを確認した上で行った追加実験における軌跡を示す。本番の試行とは関係ないがあくまで、われわれが実現しようとした到達点として結果を示す。

投下1回目で示した図同様、赤い線は親機、緑の濃淡がある3つの線がそれぞれの子機の軌跡を示す。また、大きな四角のエリアが親機が走行したい方向のエリアを示しており、薄い緑の円は各子機が送信した地形の経路評価のデータである。円が濃いほど走行が難しいことを示す。図に示す通り、3つの子機で親機が走行したいエリアをまんべんなく評価で来ていることがわかる。また、赤い点は導出した最適経路の途中点を示しており、親機がその付近を走行できていることがわかる。

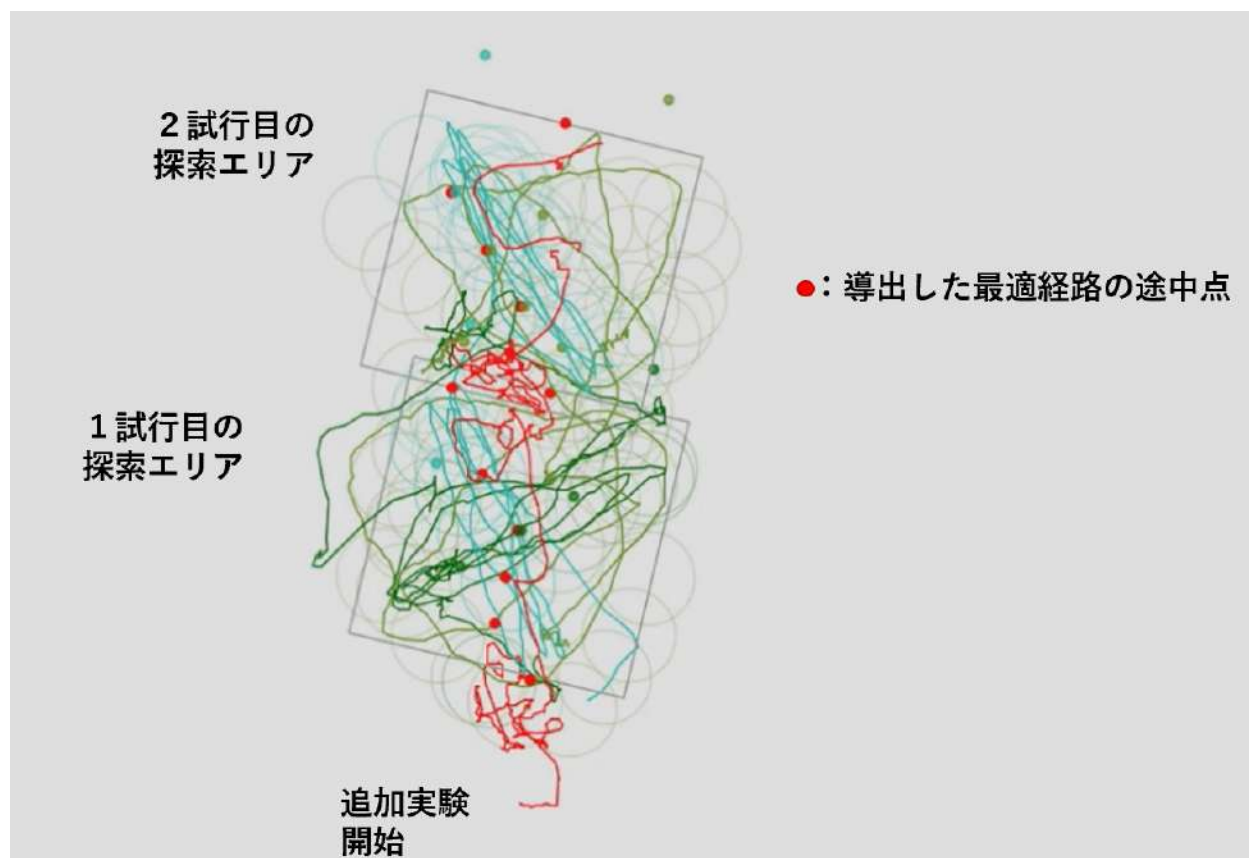


図9.2 投下2回目後追加実験における機体の軌跡

(3) 考察

- 投下1回目
投下1回目は制御履歴の最後の方にある通り、我々が機体を発見した時点で機体のタイヤがとれており、ミッション続行不可能であった。この理由としていくつか考えたが、着地したタイミングで外れたというよりは分析を進めるとパラシュートがうまく外すことができず強い風を受けたことによって、大きく引きずられる形になってしまったと考えられる。この結果を受けて、機体のタイヤ接続部分(特にイモネジがタイヤの軸をかむ部分)を見直し、より頑健なタイヤ接続になるように改良した。その結果下に述べる投下2回目の試行ではタイヤが取れることは無かった。
- 投下2回目
投下1回目の反省を踏まえ、タイヤ接続をより頑健にして臨んだ。2回目の試行ではタイヤ周りの破損は見られず、改良がうまくいったと考えられる。しかし、結果的には初期位置配置の段階から徐々に子機との通信が切断されていき、最終的には探索の段階で全ての子機との通信が切断されてしまった。これは、実際の現地での観測より、子機が一斉に親機より遠い方向に走行を始めたことが原因である。また、目指した方向は、本来われわれが想定していた初期位置とはかなりかけ離れたものであった。この座標系の不具合はGPSセンサから生じたと考えられる。おそらく2回目の投下において、初期位置配置の座標計算時に偶発的にGPSセンサが取得した値が変化した。そのため、本来の位置からかなり遠い位置に親機自身がいるという認識の元初期配置の座標を計算したため、子機は親機から遠く離れるように走行したと考える。また、初期位置配置の後、1機だけで探索を開始し、その後通信が切断されたのも同様の理由である。GPSセンサはミッション終了後に確認したところ、特に異なる出力はしなかった。そのため、追加で行った実験(図9.2の軌跡)においては、子機に正しい座標を送信することで探索が正常に行えたことがわかる。この追加実験の結果からも、偶発的なGPSの不具合による座標計算ミスが2回目の投下の失敗の要因であったと考える。

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)

1 ハード・回路面

- ハードについて
- 本ミッションは、着地してパラシュートを展開してからが本題であるため、そこまでが確実にできるようにパラシュート機構を接続する部分を工夫した。具体的には接続部が地面に接地せず、パラシュートの末端の機体につなぎ留めてある釘部分が抜けやすいようにした。その結果、2回目の投下において、着地してからパラシュート切り離しまで問題なく行うことができた。
- 回路について
- 毎年、機体の開発や試験が滞る原因として、回路の不具合がボトルネックとなり、作業が止まるということが多くあった。そのため今回の機体における回路は不具合が起こらないように細心の注意を払って設計した。また、子機に搭載する回路は、回路の小型化を実現するため回路の両面を活用して設計し、幅が大きくならないようにした。

2 ソフト面

- 本ミッションでは、安全評価データ等の伝達とために必須となる機体間通信にLoRaを使用した。しかし、この通信規格は低頻度でやり取りを行うことが前提となっているため、数秒単位で複数機体との通信が必要となる本ミッションでは相性が悪かった。そのため、既存のAPIの上に新たに独自の通信規格に則ったAPIを実装することにより、従来よりも安定した通信を行えるようになった。

3 マネジメント面

- 今回、我々がマネジメント面で特に意識しなければならなかったこととして、機体数のある程度確保しておかなければならなかったことが挙げられる。親機はもちろんのこと、子機も最低3機万全に動く状態であることが求められるミッションであったため、実際アメリカには親機を3機、子機を7機分用意して持って行った。
結果的に、最後の打ち上げ時に正常に動いたのは親機1機、子機3機のみであったため、これほどまで機体を製作してもぎりぎりであったことを考えるとかなり必要機体製作数が膨大なミッションだったといえる。

(2)課題点

1 ハード・回路面

- 今回のミッションでは、複数子機の走行データを親機に送信し、親機がそのデータを用いて走行する枠組みを実現することに重きを置いたため、その子機の形状が親機が使う走行データとして適切であったか、適切なセンサを選定出来ていたかというところには発展の余地がある。親機が安全な経路を選択できることをより担保するために回路、ハード面において改善をかけることが今後の課題といえる。

2 ソフト面

- ミッションを行う上で必須となる通信を行うLoRa関連の実装を8月末くらいまで行っており、肝心の子機の安全評価方法や安全評価データを元に親機が走行するアルゴリズムなどを考え、実装する期間が短くなってしまった。そのため、次回では細かいデバイス関連の実装を早期から行うようにしたいと考える。

3 マネジメント面

- 本ミッションはとにもかくにもコストのかかり方が尋常ではなかった。多くの機体を生産するため、その分比例してコストが増大していった。ミッションがどのようにすれば上手くいくかに重点を置き、コスト面をないがしろにしていたため、もう少し安い経費で実現できた可能性は否めない。また、機体数が必要であった分、ミッションに必要な本格的な試験を始めるのが本来の予定よりかなり後ろに倒れてしまったことは大きな反省点であった。

(3)今後の展望

- ミッション面

今回の我々のミッションは、本来のCanSatの意図とは異なるかもしれないが、地上に設置する形で、3体以上のローバーの相互協力を実現した。本プロジェクトを通して、複数機体が相互協力することで実現可能になる事柄はかなり大きいと感じた。本プロジェクトにて行った広範囲の探索にとどまらず、より高度なミッション設定につなげられるように技術継承をしていきたい

- 開発面

我々が行ったような3機以上の機体の相互協力をテーマにしたミッションをするのであれば、1年完結ではなく数年単位で見積もって流用していくと良いかと思われる。毎年同形状の子機を使いながらもやることを変えたりすると、全体としてコストが抑えられるかと思われるので、複数機体の同時制御をミッションとして考えている団体はぜひとも検討してみしてほしい。