大会報告書

電気通信大学 高玉研究室
HighBall

内容

2016 年 10 月 12 日

梅内祐太

1 チームについて

指導教員 高玉圭樹

学年	名前	製作担当箇所
M1	梅内祐太	プロジェクトマネージャー
M1	石井晴之	ハード
M2	上野史	ソフト(轍検知，轍回避，動作)
M2	齋藤嶺	回路(メイン・モータードライバ)
B4	土橋功治	回路(メイン・モータードライバ)
B4	仲田航也	ハード(設計・製作)
B3	高谷美穂	ソフト(轍回避)
B3	高橋来夏	ハード(製作)
B2	楊坤	ソフト(轍検知)
B2	鄭佳健	ソフト(轍検知)

毎年 ARLISS に参加しており，プロジェクト活動を通してプロジェクトをマネジメントする感覚や実際にものつくりを行うなかで様々な技術や知識そして経験を身に着けることを目的に活動している．

2 CanSat 機体概要

2.1 ミッションステートメント

火星有人探査に向けた火星の情報取得を目指し，ルートと画像を対応させたマップ作成に加え物体の 3 D モデル作成のための情報を取得する機能と走行性を有した CanSat を用いて ARLISS の場で実証する．

2.2 サクセスクライテリア

	内容
ミニマムサクセス	CanSat の走行経路とその安全性を示すルートマップを形成する．具体的には，CanSat が行動を開始し行動を終了するまでの GPS 情報からルートを割り出し，画像処理を用いて判別した危険地域を提示するマップが生成できたかを測る．
ミドルサクセス	なし
フルサクセス	CanSat が目的地まで安全にたどり着くためのルートマップを形成する．具体的には，CanSat が目的地まで到達できることを確認し，その後目的地到達までのルートマップを作成し，それを用いて CanSat を安全に目的地へ到達させることが出来るか検証する．検証は別の CanSat にルートマップデータを取り込み，目的地へ 3 回中 2 回到達できるかを調査する．
アドバンスドサクセス	目的地到着後，対象物の 3D モデリングを行う．具体的に，対象物を測距センサ，画像処理により形状情報を取得し，ログデータから対象物の 3D モデルが完成するか評価する．検証方法については現在検討中である．

2.3 ミッションシーケンス

ミッションシーケンス図を図 2.3-1 として以下に示した.

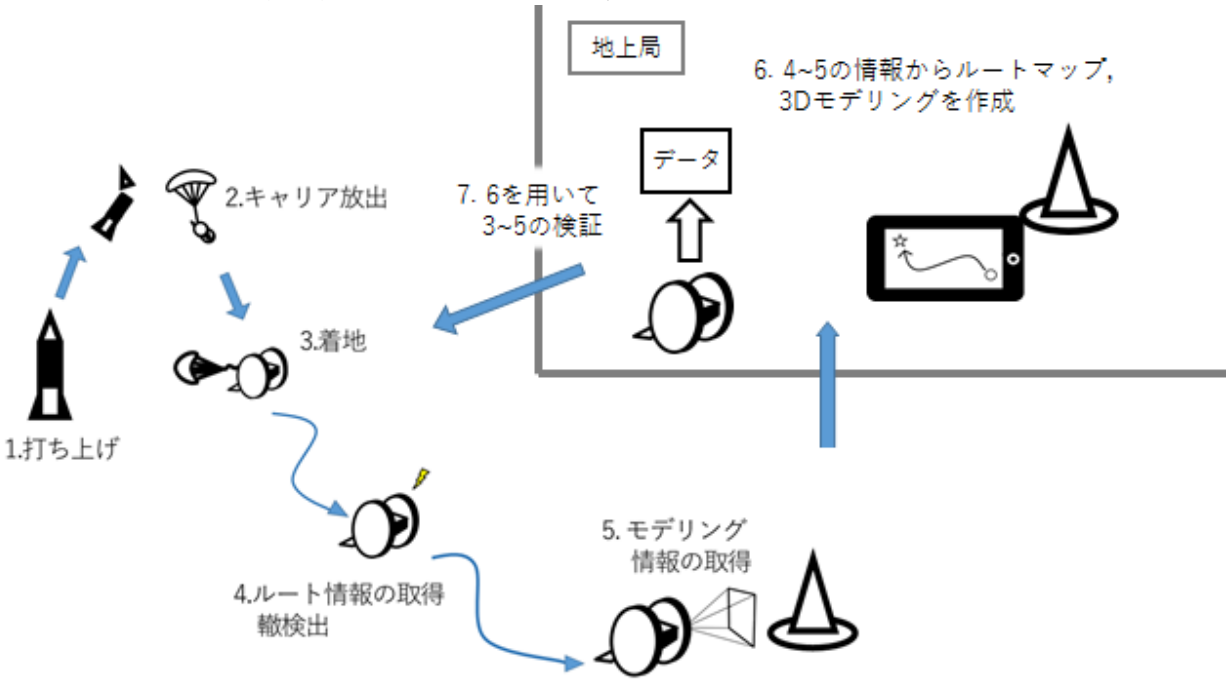


図 2.3-1 . ミッションシーケンス図

アルゴリズムの詳細な流れは 2.6 で詳細に記述する.

2.4 製作スケジュール

チーム内ミーティングの頻度・・チーム全体での会議(各班の進捗報告，困ったことなど)を週 1 回程度，それとは別に団体内での会議を週に 1 回(各チームの進捗報告，困ったことなどの共有，渡航関係の連絡など)
審査会等・・安全審査(全 2 回)

構想開始・・2016 年 3 月
開発開始・・2016 年 4 月
BBM(機能モデル)製作開始・・2016 年 4 月 25 日~
BBM(機能モデル)検証完了・・2016 年 7 月 20 日
安全審査第 1 回・・2016 年 7 月 20 日
EFM(実物大モデル、負荷検証)完了・・2016 年 8 月 19 日
安全審査第 2 回・・2016 年 8 月 22 日

ガントチャートを図 2.4-1 としていかに示す.

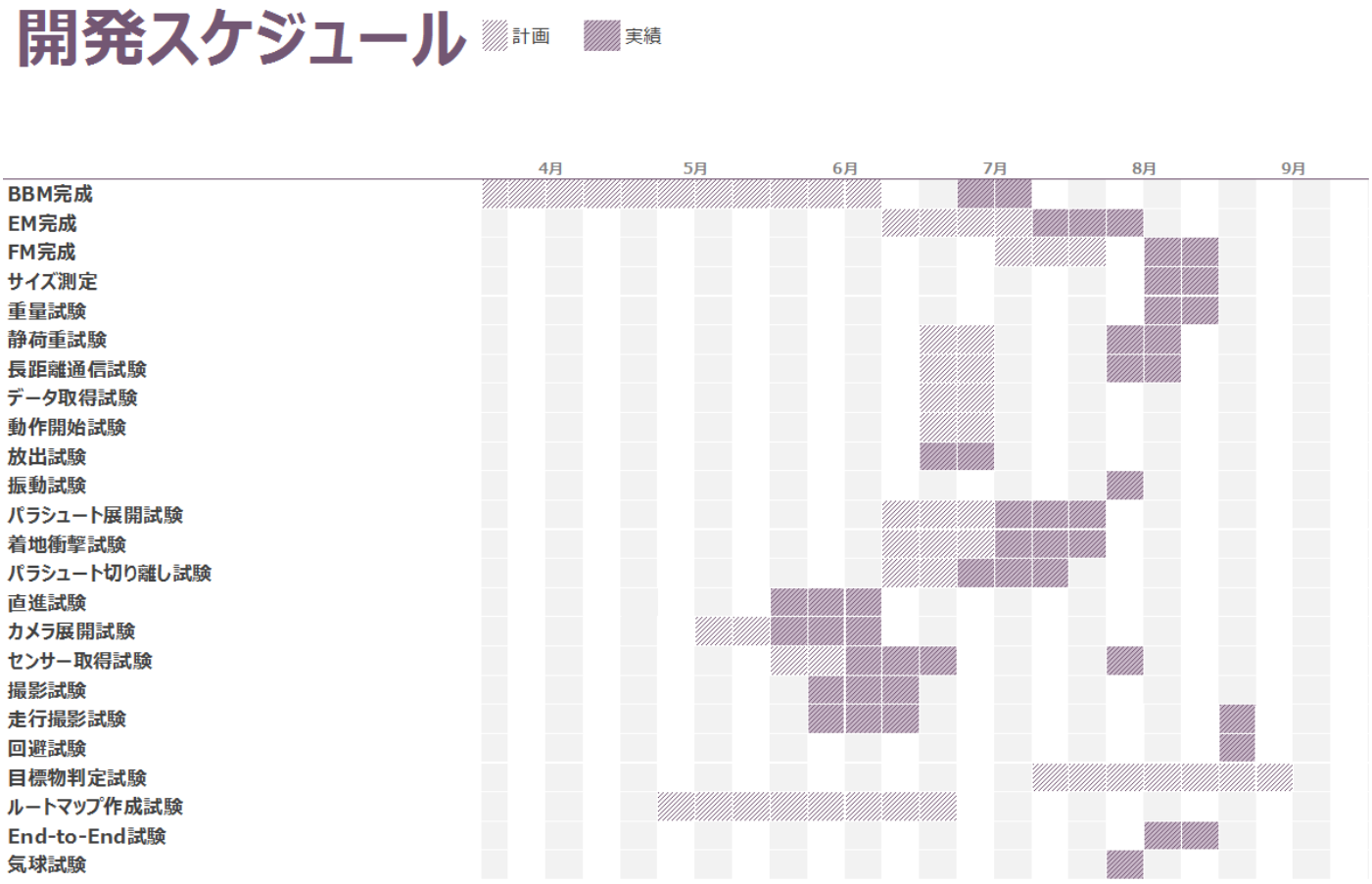


図 2.4−1 ガントチャート

2.5 システム図

機体のシステム概念図を図 2.5−1 に示す.

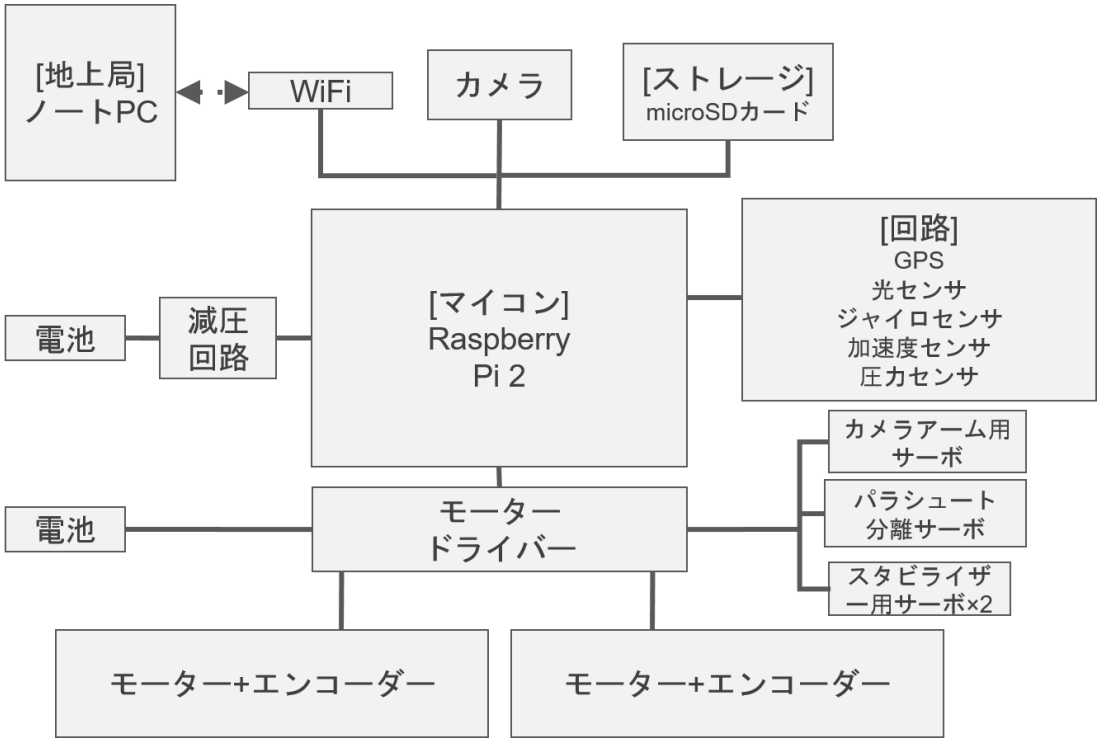
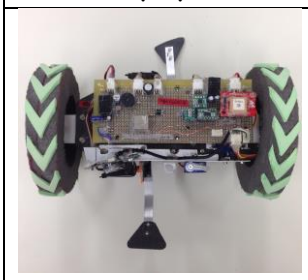
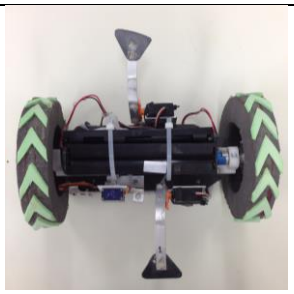
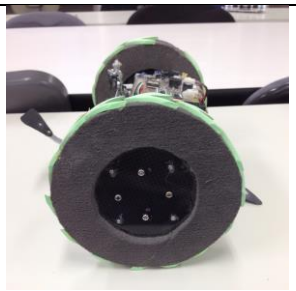
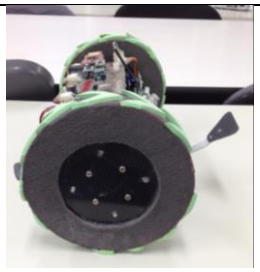


図 2.5−1. システム図

2.6 機体外観

(上)	(下)	(左)	(右)
			

2.7 機体構造・仕組み

機体の基本的な構造は 2.5 および 2.6 に示す通りである．ここではパラシュート分離機構を図と動画で，カメラアームとスタビライザーの展開の様子を動画で説明する．

パラシュート分離機構の概念図を図 2.7−1 に示す．

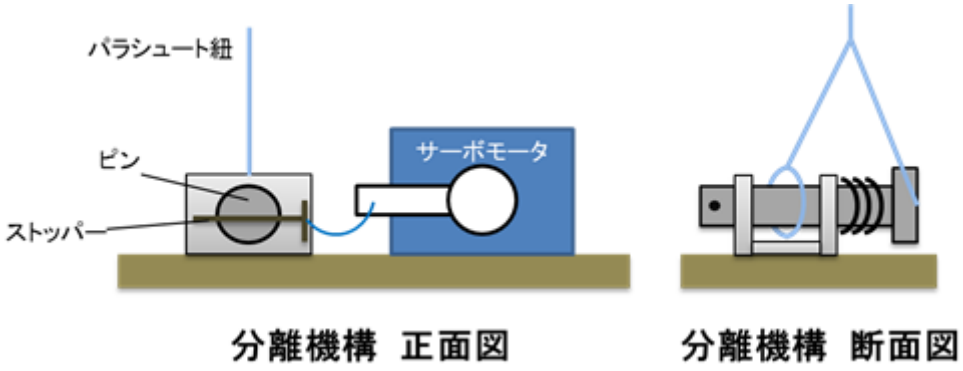


図 2.7−1　パラシュート分離機構

図右側のように，パラシュートの紐の先端にはバネ付きの金具(ピン)が取り付けられている．ローバー側に設けられたピン取り付けようの穴にピンを通し，ピン先端にストッパーを挿し込むことでピンが抜けないように固定する．着陸後，サーボモーターを駆動させることでストッパーを引き抜き，ばねの勢いでパラシュートを分離する．以下の動画はパラシュート分離の様子を示す．
https://drive.google.com/open?id=0B4rebpN36U14STNDTTRreE4yaGs

また，以下のリンクは着陸後，スタビライザーとカメラを展開し走行姿勢に入る様子を示す．
https://drive.google.com/open?id=0B4rebpN36U14RUdqdXRMcngxN0U

2.8 プログラム・アルゴリズム

使用言語: c++

プログラムのフローチャートを図 2.8−1 に示す．ローバープログラムは主に 5 つのモードで成り立つ．モードは Waiting，Falling，Separating，Navigating，Modeling と呼ばれそれぞれ下記のシーケンスを実行する．

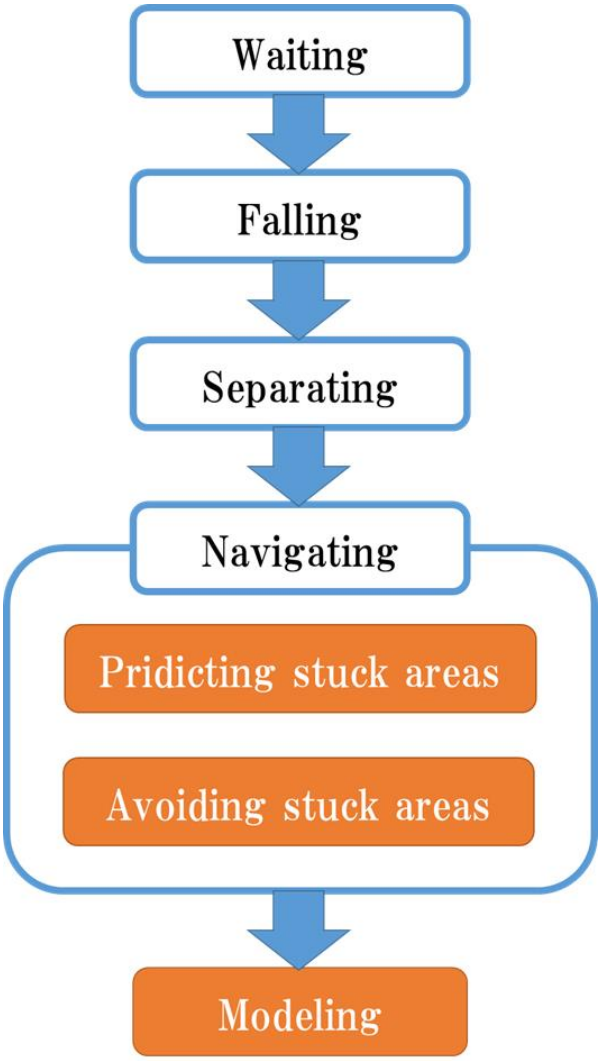


図 2.8-1. プログラムのフローチャート

- Waiting
  キャリア内での動作を行うシーケンス．詳しくは光センサを用いて数秒間光が当たり続けると放出判定を行う
- Falling
  パラシュートによる降下中の動作を行うシーケンス．詳しくはジャイロセンサ，大気圧センサ，ロータリーエンコーダを用いてジャイロセンサの値が小さく，大気圧センサの 1 秒毎の差が小さい，またロータリーエンコーダの値が小さい状態が続いたとき着地判定を行う．
- Separating
  パラシュート切り離しの動作を行うシーケンス．パラシュートのサーボを左右合計 16 回動かし，パラシュートの
  ピンを抜く．
- Navigating
  ゴールまでナビゲーションを行うシーケンス．ナビゲーションは現在の GPS 座標と目的の GPS 座標を比較し向きを決定し，ジャイロセンサによる PID 制御を行いゴールまで進む．今回はナビゲーション中に轍検知と轍回避の動作を加えている．これについては後述するが，ナビゲーションでは 5 秒おきに画像を撮影し，その画像を基にして轍検知を行う．
- Modeling
  今回の新規開発要素の一つであるモデリングの動作を行うシーケンス．詳しくはその場で写真を撮り，その後遠隔操作によりどの位置でも写真を取ることができるようにする．

- 轍検知(Pridicting stuck areas)
  轍検知の方法をここでは説明する．図 2.8−2 はその検知の結果を表している．図 2.8−2 の左の画像はローバーが撮影した画像で，右の画像は検知結果を表している．轍検知はまず，画像にガウシアンフィルタをかけてぼかし，Canny フィルタと呼ばれるフィルタでエッジを抽出する．なお，Canny フィルタでエッジを抽出すると図 2.8−3 の右の図の様にエッジが抽出できる．図 2.8−3 の右の図において白い部分がエッジを表している．エッジを抽出した後は画像を 8x9 のエリアに区切って各エリアのエッジの量を計算する．図 2.8−2 の右図の数値がそのエッジの量を示し，赤は 200 以上，青は 80 以上 200 より小さい，緑はそれ以外の数値を示している．そして，赤字の領域が画面の数%を占めるとき轍を検知する（ARLISS では 30%で設定）．検知した後轍回避動作へ移行し，その際も 5 秒に 1 回画像を撮影し続け轍が無いと検知されるまで轍回避動作を続ける．轍が無いと判定されればナビゲーションに移行する．
  この轍検知アルゴリズムは下記のような凸凹した硬い轍の検知を想定しており，エッジの量が大きいところを轍として検知している．

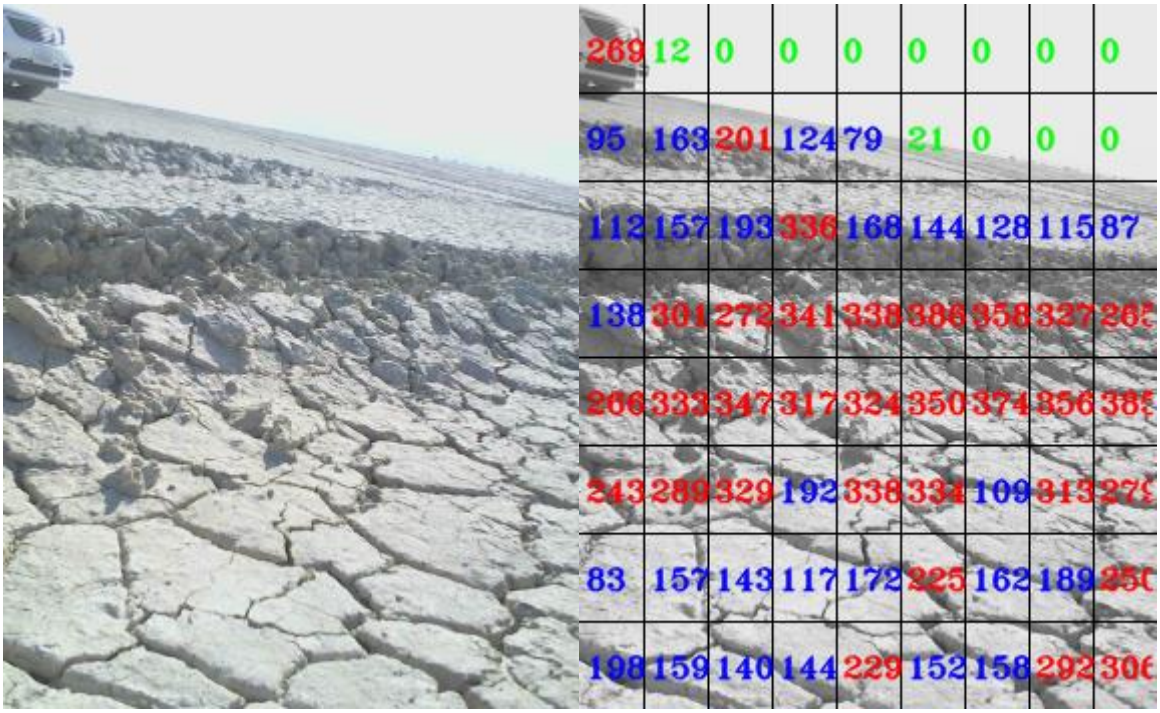


図 2.8−2. 轍検知の結果

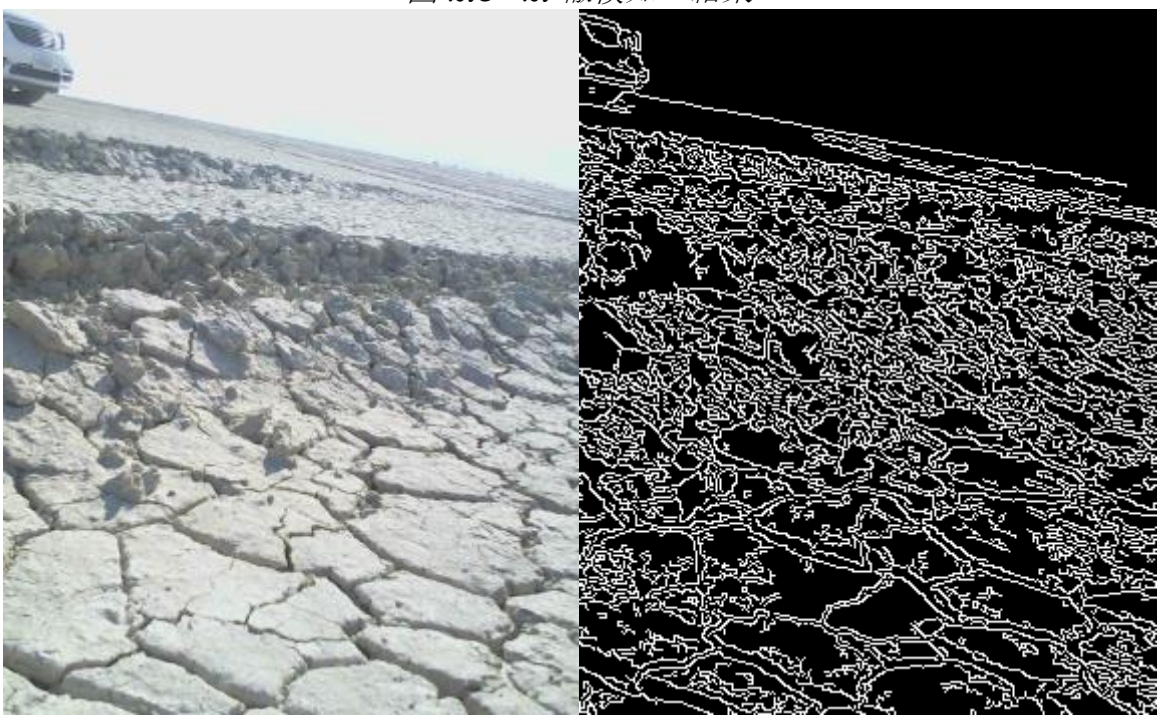


図 2.8-3. Canny フィルタ

- 轍脱出(Avoiding stuck areas)
  轍を検知した際にとる行動を図 2.8-4 に示す．まず轍を検知した後ローバーは PID 制御により横 90 度の方向へ直進する（図の①に相当）．その後検知が終わるまで直進する（図の②に相当）．検知が終わった後はナビゲーションに移行する（図の③に相当）．検知が終了するときその付近は轍が無いということを表すのでナビゲーションにより進むべき方向へ進ませることで，本アルゴリズムは自然に轍を回避して進むことができる．
  本アルゴリズムは轍を想定しているため，横 90 度へ直進している．そうすることで轍に入ること無く轍の無い地点を探索できることが特徴である．

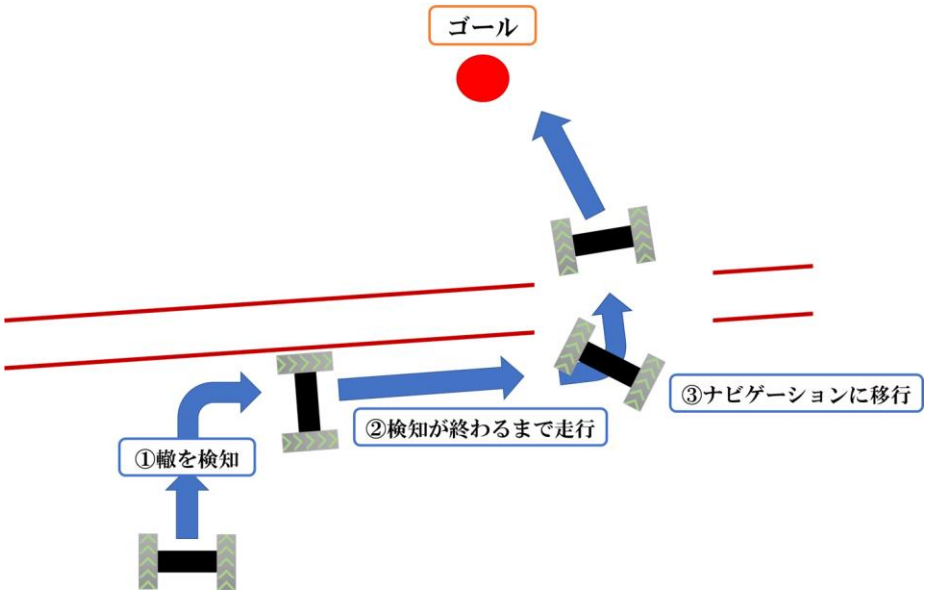


図 2.8-4. 轍回避行動

2.9 特に工夫した点・苦労した点

・工夫した点

ハード

空間的，重量的な制約が大きい中，展開・格納が可能な可動式カメラアームを搭載した．これにより従来より高い視点から地形を撮影することが可能となり，轍検知に貢献した．

タイヤやモーターホルダ，シャーシなど基本システム部分の設計を極力流用したことで新規要素であったカメラアームの開発に注力できた．

ソフト

轍検知には，撮影した画像に対してエッジ検出を施すことで，危険地帯かどうかの判断をしたこと．

回路

去年の回路基板では，ＧＰＳセンサーとＷｉｆｉモジュールを近くに配置してしまったため，ＧＰＳセンサーの取得がスムーズに行われませんでした．そのため，今年の回路を設計する際には，GPS センサーと Wifi モジュールを離して設計しました．

・苦労した点

ハード

ミッションおよびハード要求の確定に非常に時間がかかった，不慮の事態で人員が不足したなどの理由により製作が遅れたことは非常に苦労した．しかしながらこのような困難な状況を乗り越えてなんとか打ち上げることができた点は評価できる．可動式カメラアームの搭載にあたっては，スペースの確保と軽量化および強度の確保に非常に苦労した．打ち上げ直前まで強度の問題を解決できず，土壇場の工夫で打ち上げにこぎつけたことは，本来そうならないように準備すべきではあるが，努力した点である．

ソフト

機体に搭載したカメラから撮影した画像にどのような処理を行なえば，轍検知できるかはチーム全体で苦労したことです．本番時には，エッジ検出を施すことで，轍検知することに成功しました．

回路

サーボへ供給する電力が周期的に変動してしまう問題があり，それへの対処に苦労しました．電源とサーボとの間にダイオードを挟むなど試しましたが，それでも解決できませんでした．最終的にプログラム的に変更を加えて解決しました．

3　大会結果

3.1 ARLISS

3.1.1 目的

CanSat が未知の環境において危険を検知し自律的に回避行動を行うことを確認する

3.1.2 結果

ARLISS 本番では機体の整備の都合上 1 度のフライトとなってしまった．さらに打ち上げたフライトでは着地の衝撃で後ろのスタビライザーが破損してしまったため転倒してしまい走行不可能になりリタイアしてしまった(図 3.1.1)．しかし，ARLISS での目的である"CanSat が未知の環境において自律的に回避行動を行うこと"を達成でき，その結果と次年度への期待もあり，UNISEC 賞を受賞することができた．

具体的には，地面の轍を検知してその轍を回避する行動．悪路に置かれたときその状態を検知し悪路を抜け出す行動．目の前の障害物を検知し回避する行動を確認することができた．以下はそれぞれの行動を撮影した動画のリンクである．

リンク 1:轍
リンク 2:悪路
リンク 3:障害物

表 3.1.2-1　ミッション達成率

	達成率
ミニマムサクセス	50%
フルサクセス	0%
アドバンスドサクセス	0%

表 3.1.2-2　Flight 結果

	パラシュート開傘	壊れずに着地	パラシュート分離	走行開始	10m 以上制御走行	100m 以上制御走行	1km 以上制御走行	ゴール
1 回目	○	×	○	△	×	×	×	×

図 3.1.1 破損したスタビ

3.1.3取得データ

本番で取得されたデータの内，制御履歴を図 3.1.3−1，GPS を図 3.1.3−2，ジャイロセンサーを図 3.1.3−3，大気圧センサーを図 3.1.3−4，加速度センサーを図 3.1.3−5 にそれぞれ示す．図 3.1.3−2 では x 座標，y 座標，z 座標を表し，図 3.1.3−3 ではコリオリ力の x 方向，y 方向，z 方向，dx 方向，dy 方向，dz 方向，図 3.1.3−4 では大気圧，図 3.1.3−5 では x 方向，y 方向，z 方向へのそれぞれの加速度を示している．

```
Start waiting (stopped)                          ←キャリア内モードへ移行
Waiting... Time --> 18:24:20
Waiting Finished!
Falling... Time −−> 18:24:26                     ←パラシュート降下中のモードへ移行
Waiting Finished!
Pressure Sensor is Ready!: (2079.375000 -2.455933 -1.028748 0.000844)
GPS Firmware Version:21
Pressure Count  1 / 5 (888 hPa)                  ←大気圧センサによる着地判定
Gyro Count      0 / 1000                         ←ジャイロセンサによる着地判定
MotorPulse Count 0 / 10 (0,0)                    ←エンコーダーによる着地判定
GPS Position    (40.870525 -119.106192 1187.000000)
Pressure Count  2 / 5 (889 hPa)
Gyro Count      0 / 1000
MotorPulse Count 0 / 10 (0,0)
GPS Position    (40.870525 -119.106195 1187.000000)
Pressure Count  3 / 5 (888 hPa)
Gyro Count      0 / 1000
MotorPulse Count 0 / 10 (0,0)
GPS Position    (40.870523 -119.106190 1187.000000)
Pressure Count  4 / 5 (889 hPa)
Gyro Count      0 / 1000
MotorPulse Count 0 / 10 (0,0)
GPS Position    (40.870520 -119.106190 1187.000000)
Pressure Count  0 / 5 (885 hPa)
Gyro Count      0 / 1000
MotorPulse Count 0 / 10 (0,0)
GPS Position    (40.870525 -119.106182 1187.000000)
Pressure Count  1 / 5 (886 hPa)
Gyro Count      0 / 1000
MotorPulse Count 0 / 10 (0,0)
GPS Position    (40.870530 -119.106173 1188.000000)
 (中略)
Gyro Count      248 / 1000
MotorPulse Count 0 / 10 (0,0)
GPS Position    (40.870547 -119.106245 1189.000000)
Pressure Count  0 / 5 (889 hPa)
Gyro Count      458 / 1000
MotorPulse Count 0 / 10 (0,0)
GPS Position    (40.870547 -119.106247 1189.000000)
Pressure Count  1 / 5 (890 hPa)
Gyro Count      678 / 1000
MotorPulse Count 0 / 10 (0,0)
GPS Position    (40.870548 -119.106248 1189.000000)
Pressure Count  2 / 5 (889 hPa)
Gyro Count      898 / 1000
MotorPulse Count 0 / 10 (0,0)
GPS Position    (40.870548 -119.106250 1189.000000)
Pressure Count  3 / 5 (889 hPa)
Gyro Count      1000 / 1000
```

MotorPulse Count 0 / 10 (0,0)
GPS Position (40.870550 -119.106250 1189.000000)
Pressure Count 4 / 5 (888 hPa)
Gyro Count 1000 / 1000
MotorPulse Count 0 / 10 (0,0)
GPS Position (40.870550 -119.106250 1189.000000)
Pressure Count 5 / 5 (887 hPa)
Gyro Count 1000 / 1000
MotorPulse Count 0 / 10 (0,0)
GPS Position (40.870550 -119.106250 1189.000000)
Falling Finished!
Separating... Time --> 18:25:16 ←パラシュート切り離しモードへ移行
Separating...(1/16) ←パラシュートのサーボを動かしている
Separating...(2/16)
Separating...(3/16)
(中略)
Separating...(15/16)
Separating...(16/16)
Waking Timeout : unable to spin
Waking will be retried (1 / 5) by power 50.000000
Waking Timeout : unable to spin
Waking will be retried (2 / 5) by power 55.000000
Waking Timeout : unable to spin
Waking will be retried (3 / 5) by power 60.000000
Waking Timeout : unable to spin
Waking will be retried (4 / 5) by power 65.000000
Waking Timeout : unable to spin
Waking will be retried (5 / 5) by power 70.000000
Waking Timeout : unable to spin
Waking Failed!
Para ratio: 0.000000
Para check: Not Found!!
Separating Finished!
Captured image was saved as capture92.png
Navigating...
Starting navigation...Time --> 18:25:51 ←ナビゲーションモードへ移行
Control Start Point:(40.870543 -119.106242)
PID is Started (0.000000, 100)
Captured image was saved as pic_gps_0.png ←カメラ撮影
NAVIGATING: Last 31 samples (40.870542 -119.106246) Current(40.870545 -119.106262)
↑GPS 座標の取得
distance = 1967.351408 (m) delta angle = -45.328966(RIGHT) ←目的地までの距離と角度を計算
PID is Started (-45.328966, 50)
Captured image was saved as pic_gps_1.png
NAVIGATING: Last 36 samples (40.870573 -119.106299) Current(40.870612 -119.106312)
distance = 1958.469209 (m) delta angle = 15.877480(LEFT)
PID is Started (-29.451486, 50)
NAVIGATING: GPS Error value detected
Captured image was saved as pic_gps_2.png
NAVIGATING: Last 4 samples (40.870686 -119.106231) Current(40.870685 -119.106232)
distance = 1956.683852 (m) delta angle = -90.000000(RIGHT)
PID is Started (-119.451486, 50)
NAVIGATING: GPS Error value detected
Captured image was saved as pic_gps_3.png
(中略)
PID is Started (81.499810, 50)
Captured image was saved as pic_gps_12.png
NAVIGATING: Last 31 samples (40.870909 -119.106415) Current(40.870942 -119.106465)
distance = 1918.533827 (m) delta angle = -22.588636(RIGHT)

```
PID is Started (58.911174, 50)
> start testing                    ←実験終了
```

図 3.1.3−1.　制御履歴

```
40.870533,-119.106232,1188.000000
40.870533,-119.106232,1188.000000
40.870533,-119.106232,1188.000000
40.870533,-119.106232,1188.000000
40.870533,-119.106232,1188.000000
40.870533,-119.106232,1188.000000
40.870533,-119.106232,1188.000000
40.870533,-119.106232,1188.000000
40.870533,-119.106232,1188.000000
40.870533,-119.106232,1188.000000
```

図 3.1.3−2.　GPS

```
36.404325,-38.319353,3.588829,-0.156236,-0.209462,0.037078
42.424325,-48.749353,-6.841171,4.547427,-3.485991,-0.676091
0.000000,-3.529353,-1.031171,4.434479,-4.292787,-0.289739
0.000000,-9.969353,-17.411171,4.157613,-4.758956,-0.774225
44.454325,-68.559353,-56.401171,6.950107,-9.651009,-4.949848
40.604325,-53.789353,-15.381171,10.900852,-16.791055,-8.957451
13.654325,-35.239353,5.338829,13.650304,-21.545331,-9.164652
8.894325,-27.539353,-9.151171,14.761562,-24.543048,-9.278388
21.774325,-47.489353,-27.071171,16.751536,-28.612827,-11.232760
3.504325,-39.579353,-20.911171,18.392886,-33.452959,-14.048366
```

図 3.1.3−3.　ジャイロセンサー

```
894.947754
890.126770
889.725037
887.533264
887.533264
888.027527
891.423584
888.428833
888.428833
888.830078
```

図 3.1.3−4.　大気圧センサー

```
-0.164062,-0.078125,0.515625
-0.411133,-0.132812,0.929688
-0.459595,-0.127930,0.967773
-0.547684,-0.004272,0.953003
```

```
-0.470804,-0.148972,0.951157
-0.336194,-0.397528,1.009520
-0.292024,-0.549691,1.010956
-0.251347,-0.568711,1.018948
-0.200279,-0.576560,0.985559
-0.157847,-0.706836,0.941554
```

図 3.1.3−5.　加速度センサー

3.1.4故障原因解析

打ち上げ実験では，着地衝撃によりバックスタビが破損してしまいました．そのため，機体が走行不可能になり，ミッションが達成できませんでした．この原因を探るため FTA を用いた(図 3.1.4.)．強度不足と着地姿勢が考えられます．強度不足が原因となり得るかを判断するのに，強度不足を設計不良と組み立て不良の二つに分けました．設計不良に関しては，バックスタビを中心から離れた位置に取り付けたため，着地したとき，落下衝撃によりタイヤがしぼんでしまい，バックスタビが地面に接触してしまう可能性がありました．次に，組み立て不良について原因を探ったところ，1 号機用と 2 号機用のバックスタビの曲線の傾きが異なっていました．具体的には，2 号機用のバックスタビは反りがあまくなっていました．しかし，これを考慮せずに 1 号機と 2 号機ともに同様の組み立て方をしたため、バックサーボが外側に反りかった状態になってしまい、着地時に地面と接触する可能性がありました．最後に着地姿勢ですが，目視で見た限りでは，ほぼ垂直に落下しており，理想的な着地姿勢でした．以上より，バックスタビが故障した原因は，1 号機用と 2 号機用のバックスタビの設計が異なっていたにも関わらずに，同様の取り付け方をしてしまったこと，バックサーボの取り付け位置が中心から離れていたことが考えらます．

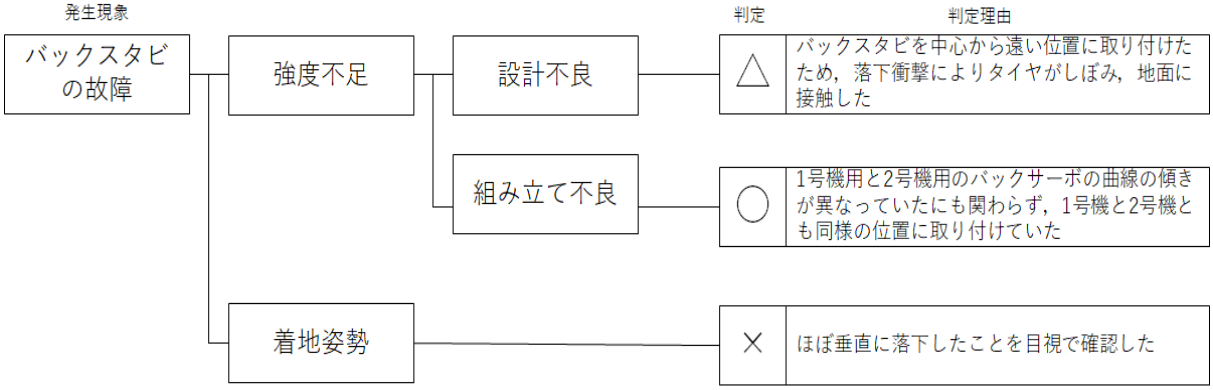


図 3.1.4.　原因分析

4　まとめ

今回の結果から惑星探査のような未知の環境において CanSat が自律的に危機を感知して回避行動を行うことを確認できたことはとても意義のある結果だったと思う．すべてのリスクを事前に知りうることはできないためこのような技術が確立することで新しい探査の形を生み出す可能性を秘めている．しかし，上記の原因によりミッションとしては失敗してしまったことが悔やまれる．衝撃がかかる状況を十分に網羅できていなかったこと，打ち上げ前に十分にチェックが行えなかったことが失敗の原因であったがこれらは事前に準備していれば解決できた問題である．このようなことが今後起こらないよう十分スケジュールに余裕をもってプロジェクトを進めることの重要さを感じた．やはりスケジュールに余裕がないと十分にチェックリストを作成できていなかったり見落としをしてしまう危険性が増える．全体を通して十分な要求分析や必要な実験をリストアップしてそれを一つ一つこなすことがうまくいかず，全体的にスケジュールが押してしまった．しかもスケジュールが予定通り進行していないにも関わらずあまり危機感を感じている様子が見られない部分もあった．　またチームの全体の一体感を作り出すことがうまくできなかった点が反省点の一つとして挙げられる．さらにどのように行動していいのかわからない人が発生したときにうまく対応できなかったことでせっかくの人的リソースを十分に活用できなかった．この問題を解決するために事前にドキュメントを作っておく，教える環境を十分に用意しておくことを怠ってはいけないことを学んだ．今年の ARLISS プロジェクトではそのような意味も含めてとても反省点が多くいい意味でも悪い意味でもいい経験することができたと思う．次年度は今回の反省を十分活かせるようにする．