

大会報告書

所属(慶應義塾大学 高橋研究室)

Keio Team Wolve'Z

内容

1	チームについて	2
2	CanSat 機体概要.....	2
3	大会結果	22
3.1	能代宇宙イベント	22
3.2	ARLISS.....	24
4	まとめ.....	27
5	責任教員所感.....	27

2016年10月12日

作成者：杉原 有理花

1 チームについて

指導教員 高橋 正樹

学年	名前	製作担当箇所
M1	杉原 有理花	プロジェクトマネージャー
M1	黒須 純	ハードウェア主任
M1	小澤 僚太郎	ソフトウェア主任
B4	小林 大介	制御・誘導、機体製作
B4	森 光汰	機体・タイヤ製作、電源系統
B4	吉田 直人	地上局
B4	渡邊 幹大	Web デザイン、庶務
M2	波田野 恭祥	メンター

CanSat を作る目的：

ものづくりの知識と技術向上のため。また、プロジェクトマネジメントの学習のため。

2 CanSat 機体概要

2.1 ミッションステートメント

カムバックコンペティションで優勝する。そのため、二輪型ローバーで気球からパラシュートで降下し、着地後にゴールへ向かって自律走行し、ゴールから半径 10m 以内に到達して停止する。

明確な目的のあるランバックコンペティションで技術を競い実証することで、定量的に完成度を評価することができるため、ランバックコンペティションに参加した。

2.2 サクセスクライテリア

	内容
ミニマムサクセス	着地後 1m 走行
ミドルサクセス	—
フルサクセス	ゴールから半径 10 m 以内に到達
アドバンスドサクセス	ゴールから半径 5 m 以内に到達

2.3 ミッションシーケンス

図 1 にミッションシーケンスを示す。詳細な移行条件等は 2.7.1 節で述べる。

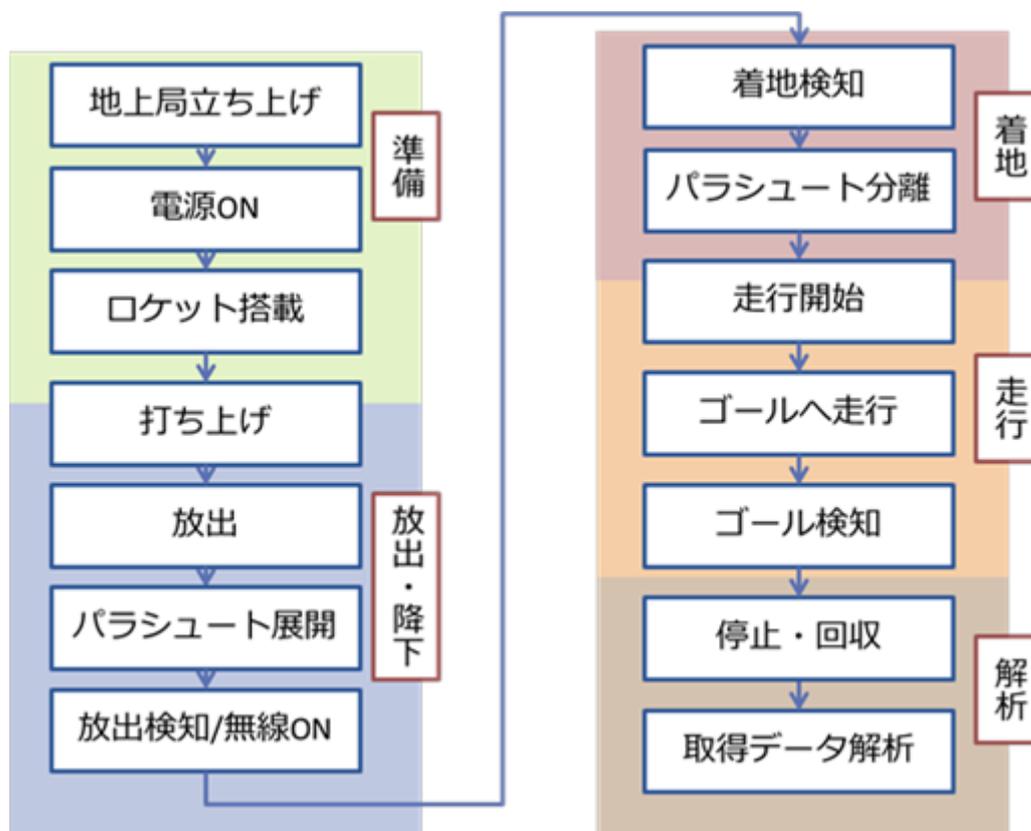


図 1 ミッションシーケンス

2.4 製作スケジュール

チーム内ミーティングの頻度・・・週 1～2 回

審査・・・能代審査 2 回、ARLISS 審査 2 回

構想開始・・・2016 年 3 月

設計開始・・・2016 年 4 月

BBM(機能モデル)製作開始・・・2016 年 4 月 18 日

BBM(機能モデル)検証完了・・・2016 年 5 月 14 日

EM(実物大モデル、負荷検証)完了・・・2016 年 8 月 13 日

FM(実物大モデル、負荷検証)完了・・・2016 年 9 月 1 日

図 2 から図 4 に、各段階における詳細なスケジュールを記載したガントチャートを示す。

・BBM



図 2 ガントチャート (BBM)

・EM

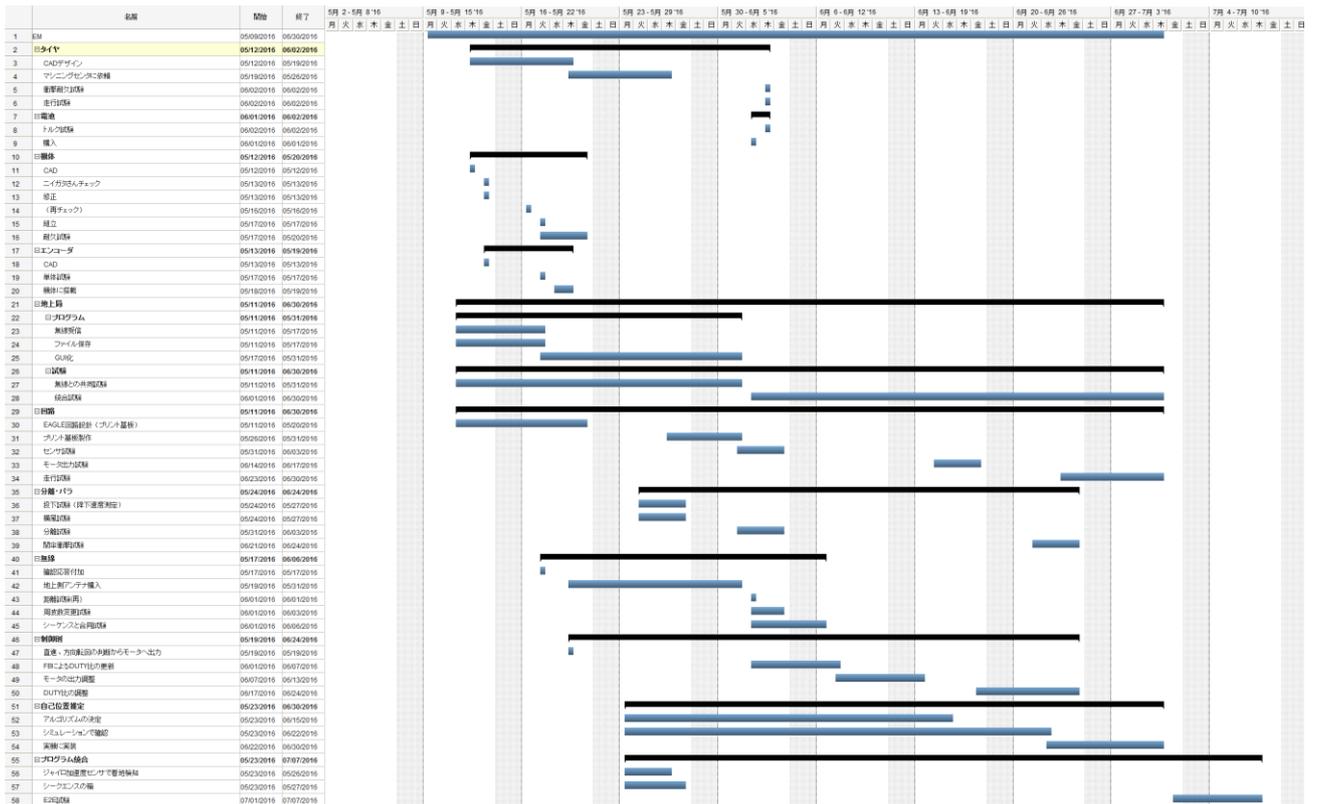


図 3 ガントチャート (EM)

・FM



図 4 ガントチャート (FM)

2.5 システム図

図 5 にシステム図を示す。

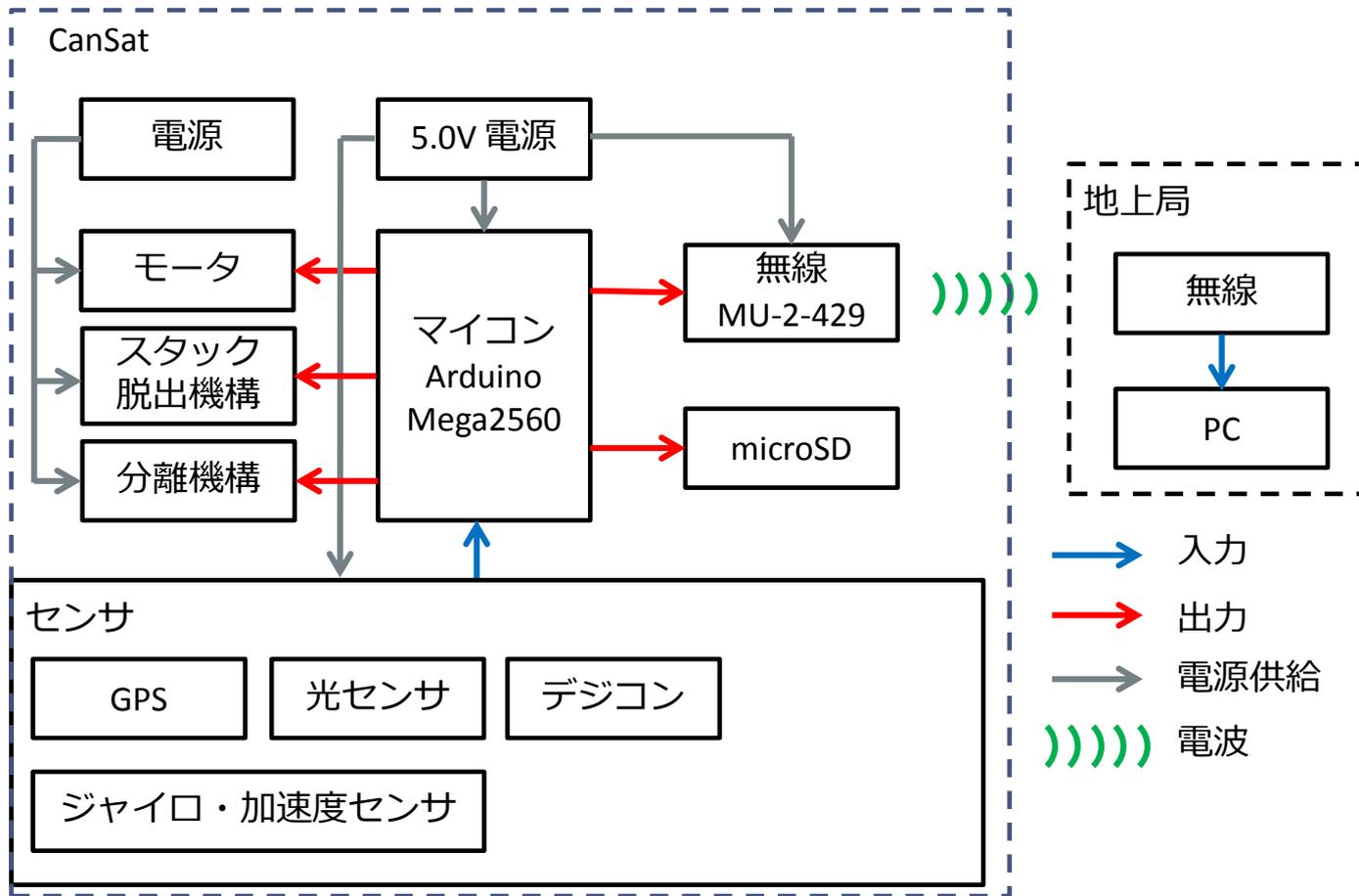
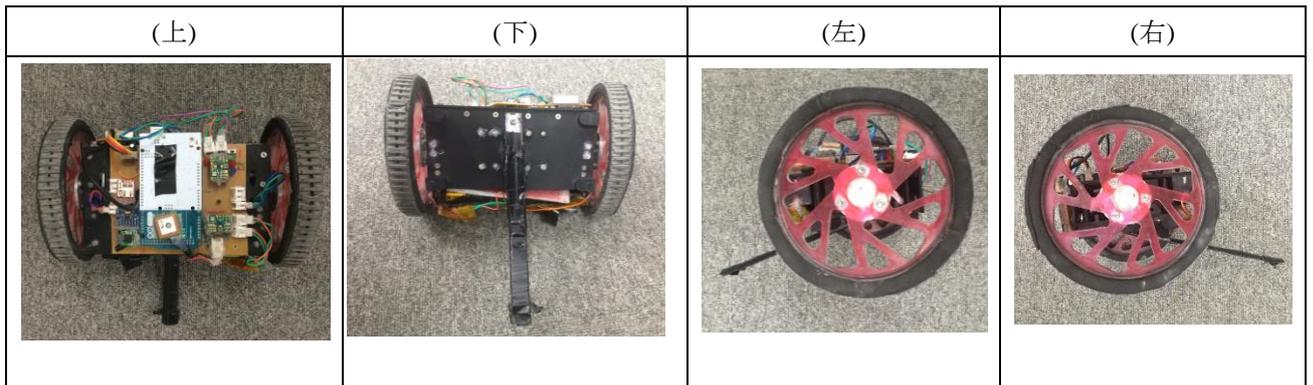


図 5 システム図

2.6 機体外観



2.7 機体構造・仕組み

2.7.1 機体構造

機体は二段構成とした。上段に回路とセンサ、下段にモータ、電池を配置した。機体がスタックするのを防ぐために、図 6 のように機体と地面の間を広く設けた。落下の衝撃に強くするため下段にリブ

をつけた。

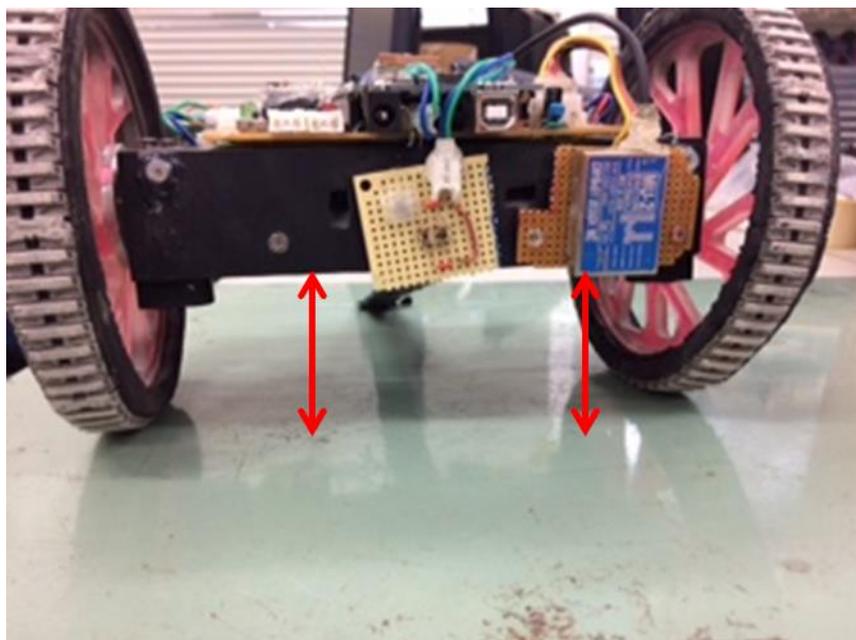


図 6 機体の外観

2.7.2 タイヤ



図 7 タイヤの外観

図 7 にタイヤの外観を示す。ホイール部分はポリカーボネート板から削り出し、直径は 11.5cm である。その周りに 1cm の黒いスポンジを巻き付け、衝撃に強い機構にしている。また、滑り防止にタミヤモータ製のキャタピラを巻いている。

耐衝撃に関しては、図 8 のように応力解析を行うことでこのタイヤ案が十分耐えうることを確認し

てから作成した。

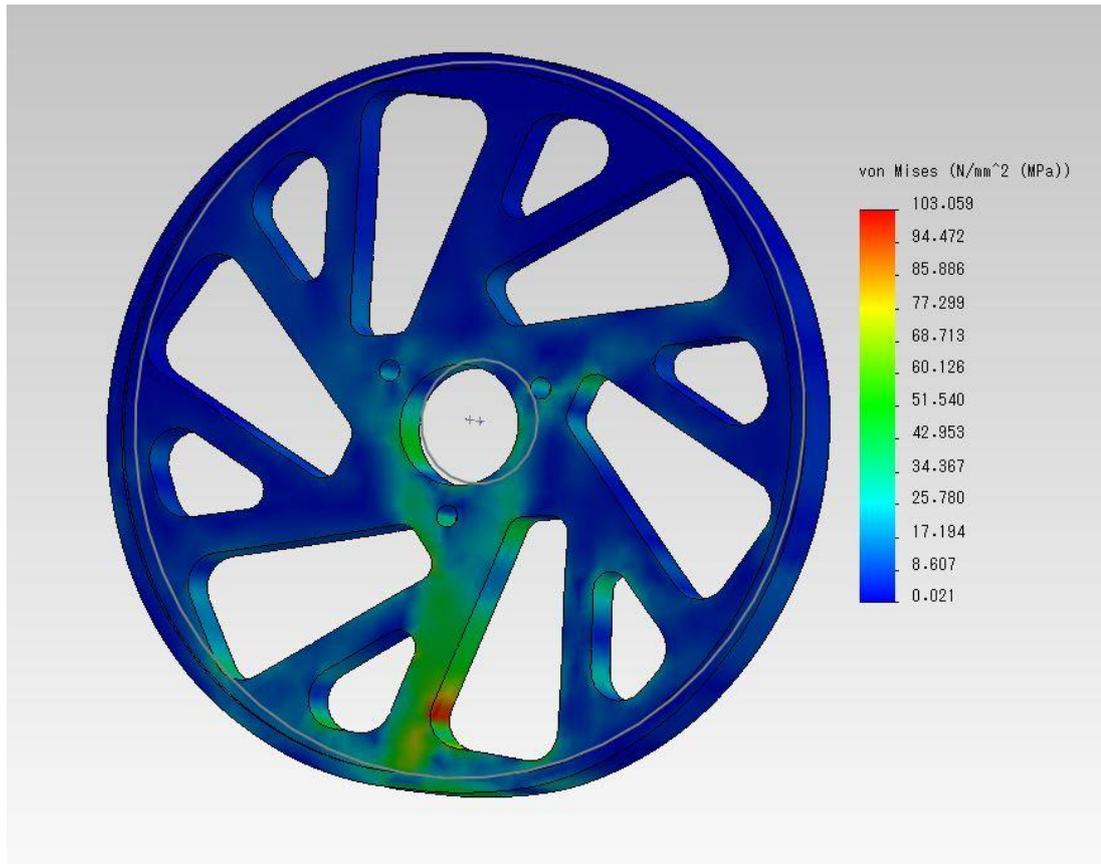


図 8 ホイールの応力解析

2.7.3. パラシュート展開・分離

パラシュートおよびそれらに付随する構造を図 9 に示す。下図に示した機構は以下の 3 点で構成される。

- A) キャリア収納時に機体を包む半透明のポリエチレン板（以下、分離シート）
- B) 機体とパラシュートを接合し、分離後にパラシュートが機体に被さってしまうことを防ぐ機構（以下、メジャー機構）
- C) パラシュート本体（以下、パラシュート）

以下、各々の機構の構造および特徴について述べる。



図 9 パラシュート周りの外観

A) 分離シート

この機構の有効な点として、キャリアからのスムーズな脱出の他に、開傘・着地衝撃からの機体保護が挙げられる。分離シートは、開傘時にパラシュート糸からの荷重を直接受けることで、機体へ開傘時の衝撃を緩和し、機体が破損することを防ぐ。

前述の分離シート機構を用いる場合、ロケット格納時には機体を分離シートで包むことで固定し、分離時には分離シートを外しパラシュートと分離する必要がある。そこで今年度は、抵抗のジュール熱を用いたテグスの切断による分離方法を用いた。テグス切断機構を図 10 に示す。

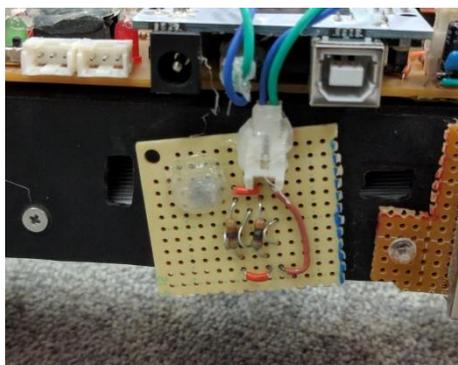


図 10 分離機構

B) メジャー機構

この機構には市販の伸縮可能なメジャーを用い、キャリアへの収納を可能にした。メジャー機構の長さは 1.20[m]とし、パラシュートの糸 0.80[m]よりも長く設定することで、パラシュートが機体に被さることを防止した。

C) パラシュート

パラシュートは視認性を高めるためオレンジ色のリップストップ生地を使用した。リップストップ生地はナイロンから構成される生地であり、特殊な強化技術を利用し、碁盤の目状にナイロンの繊維を縫い込み、生地が裂けても、それ以上に裂けることを防ぐ。この生地を利用することで開傘時の衝撃で生地が破れることを防止した。

以上に述べた 3 つの機構を組み合わせた機構をキャリアに収納可能な状態にした様子を図 11 に示す。本年度は、機体下部の空間にメジャー機構およびパラシュートを収納した。このため分離シートには図 12 に示す収納用の穴を開けた。



図 11 収納状態



図 12 分離シート

2.8 プログラム・アルゴリズム

使用言語: C++

2.8.1 シークエンス

本節では、シークエンスのステートごとの内容について述べる。

今年度用いたマイコンは Arduino Mega 2560 であり、開発言語は C++ である。Arduino IDE を用いてプログラミングを行った。

はじめに、図 13 にステートとシークエンス内の状態との対応を示す。

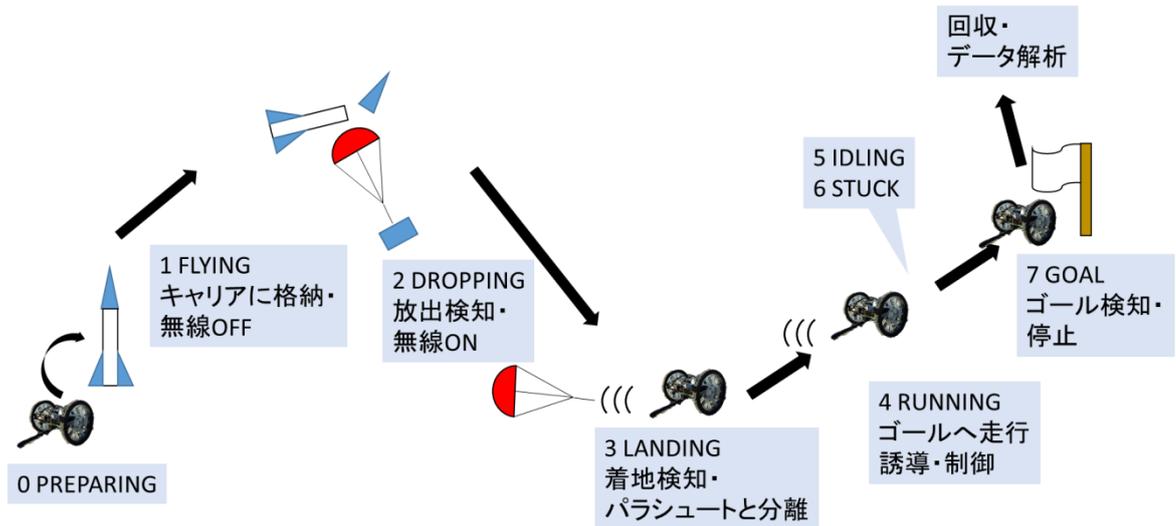


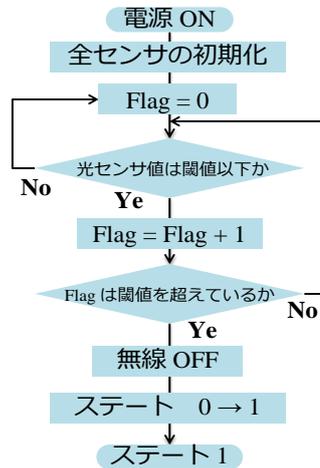
図 13 シークエンスとステート

次に、各制御シークエンスの概要を以下に述べる。

・ステート 0 : PREPARING

電源を ON にし、搭載したセンサ・モータの初期化を行うステートである。この時点で、SD への記録を開始し、MU-2 (無線) の送信を開始する。本ステートから次のステート (FLYING) への移行には光センサを用い、センサ値が閾値を下回る (キャリアに格納されたと判断される) 場合には、ステートを FLYING に移行する。

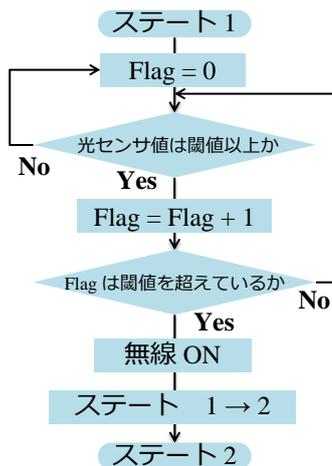
本ステートにおけるフローチャートを次に示す。



・ステート 1 : FLYING

CanSat がキャリア内に格納されたと検地した際に、本ステートに移行する。本ステートでは、ロケット内部に格納されている状態である。CanSat のステートの移動を検知するために光センサ（フォトリフレクタ）と GPS センサ、ジャイロ加速度センサが駆動している。また、レギュレーション通り、キャリアに格納された状態では無線を OFF にするように設定されている。光センサのセンサ値が閾値を超える（キャリアから放出されたと判断される）場合には、ステートを DROPPING に移行する。

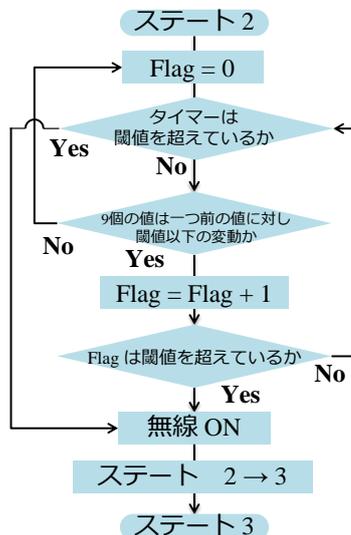
本ステートにおけるフローチャートを次に示す。



・ステート 2 : DROPPING

CanSat がキャリアから放出されたと検地された際に移行するステートである。本ステートでは、無線が動作を再開し、着地判定を行う。着地判定では、ジャイロ加速度センサを用いる。このとき、角速度・角度・並進加速度の 9 軸の変動が各センサ値に対する閾値以下に 3 分間収まり続けた場合に着地検知をし、次のステート（LANDING）に移行する。さらに、地上の風等の影響で着地しているが、検知ができない場合への対策とし、冗長系のタイマーによるステートの移動を用いた。本年度の競技では、能代・ARLISS の両競技において、ジャイロ加速度センサによる着地検知がうまく動作したため、冗長系が必要とされる場合は無かった。

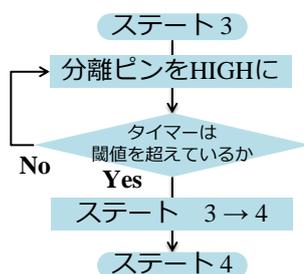
本ステートにおけるフローチャートを次に示す。



・ステート 3 : LANDING

CanSat が着地したと判断された後に、パラシュートと機体を分離するためのシーケンスである。パラシュートが接続している分離シートと機体の分離のために、接合材であるテグス糸を熱で溶かす。このとき、熱で溶かすため抵抗に電流を流した際に発生する熱を利用して分離を行う。熱分離はタイマーによるソフトウェア制御で行い、設定された時間を経過した際に次のステート (RUNNING) に移行する。

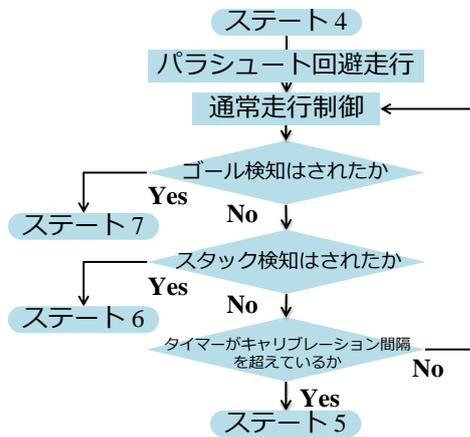
本ステートにおけるフローチャートを次に示す。



・ステート 4 : RUNNING

本ステートは、着地点からゴール座標までの走行を行うステートである。基本的にはこのステートがメインとなるが、走行状況により後に述べるステート 5、6、7 へ移行する、

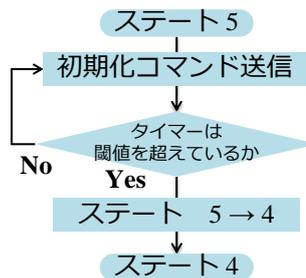
本ステートにおけるフローチャートを次に示す。パラシュート回避走行と通常走行制御については後に詳細を述べる。



・ステート 5 : IDLING

本ステートは、デジタルコンパスセンサやジャイロ加速度センサのキャリブレーションを行うステートであり、定期的に行うことにより精度が向上すると考えられるため、経験的に3分間隔でステート 4 から本ステートに移行させ、上記のセンサのキャリブレーションを行う。事前解析では、モータが駆動している際にデジタルコンパスセンサのセンサ値にオフセットが生じることやジャイロ加速度センサのz軸回りのオイラー角の補正が困難であることにより生じるオフセットとドリフト誤差を確認しているため、これらの影響を最小限に抑えることを目的としている。事前解析の結果は項目 0、 0 に示す。

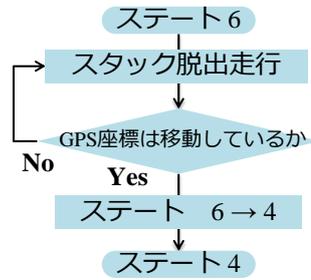
本ステートにおけるフローチャートを次に示す。



・ステート 6 : STUCK

CanSat の GPS 座標が一定時間、指定した範囲から移動しなかった場合にスタックしたと判断された場合にスタック状態であると検知し、本ステートへ移行する。本ステートでは、スタックから脱出するためのタイヤ駆動を行う。具体的には、スタックから脱出するために、左右のモータへの PWM 出力を交互に上げ下げすることにより、うねうねと移動させる制御を行う。この制御を行うのと同時に、スタック検知を行い、スタック条件と合致しなかった場合には、ステート 4 の走行ステートへと移行する。

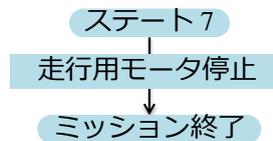
本ステートにおけるフローチャートを次に示す。



・状態 7 : GOAL

本状態は、ゴール座標を中心とした矩形のゴール範囲内に到達した際に状態 4 から移行する状態である。

本状態におけるフローチャートを次に示す。



2.8.2 走行制御と誘導

本節では、走行中の誘導と制御方法について述べる。

A) パラシュート回避走行

能代競技では、着地してから走行を始める際に、パラシュートが機体の側にあったことと、パラシュートが機体の進行方向に被ってしまっていたことから、失敗に終わってしまった。従って、この問題への対策として、図 14 のようにゴールがどこに存在するかで領域分割を行い、パラシュートの回避を行う走行を行った。なお、領域 A ではパラシュートに機体が絡まることはないため、通常走行を行うこととする。

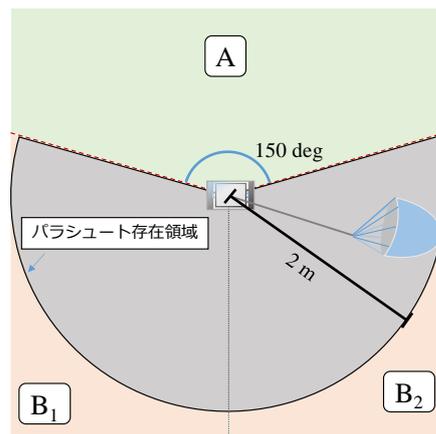


図 14 収納状態

パラシュート回避を陽に行う必要があるのは領域 B₁、B₂ である。通常走行では、後に述べるように左右のモータへの PWM 出力が舵角によって決定されるため、旋回半径が可変となる。このとき、機体進行方向とゴールまでの方位角が大きいほど舵角も大きくなるように設計されている。従って、領域

B₁、B₂ではゴール座標が機体の広報に存在するため、舵角が大きくなる。ゆえに、小さな旋回半径で曲進するような制御入力となるため、パラシュートに絡まってしまう可能性が高い。

この問題への対策として、パラシュート存在半径である2 mの3倍の旋回半径で曲進（例えば、領域B₁の場合は左旋回、半径6 mで曲進）し、ゴール座標がAの領域に入るように走行を行うことを考える。このとき、旋回半径や走行用モータへのPWM出力やその出力継続時間等は、実験により試行錯誤的に決定した。A領域にゴールが入った場合、通常走行へと切り替える。

B) 通常走行

通常走行では、位置をGPS、方位をデジタルコンパスで測定し、その値を誘導に使用した。ターゲットの座標(x_g, y_g)と自分の座標(x, y)、デジタルコンパスθの値を用いて誘導した。機体を原点とし、機体の進行方向をy軸とした座標系に変換する。(式(1))その座標系を3つの領域に分けた。座標変換後のゴールの座標(X, Y)の位置でそれぞれで右、左、直進の指示を出す(図15)。その際のゴールへのベクトルと、y軸との角度のずれφを用いて出力調整を行った。右に曲がる際の左右のモータへ与える電圧は式(2)で決定される。

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x_g - x \\ y_g - y \end{bmatrix} \quad (1)$$

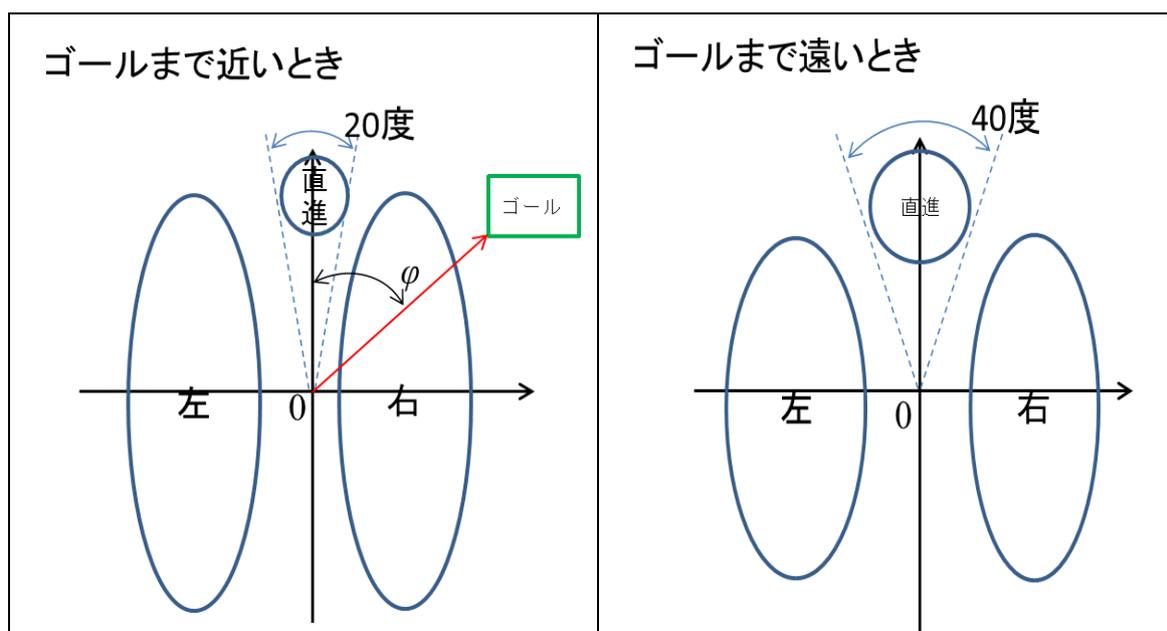


図 15 誘導の概要

$$\begin{bmatrix} V_r \\ V_l \end{bmatrix} = \begin{bmatrix} 190 \times \left(1 - \frac{\phi}{180}\right) \\ 250 \end{bmatrix} \quad (2)$$

2.7.3 センサ補正

本節では、走行する際に使用する制御量であるオイラー角や慣性座標系での機体進行角の補正方法について述べる

A) ジャイロ加速度センサによるオイラー角補正

まず、事前解析として、静止状態でのオイラー角のドリフトの様子を図 16 に示す。

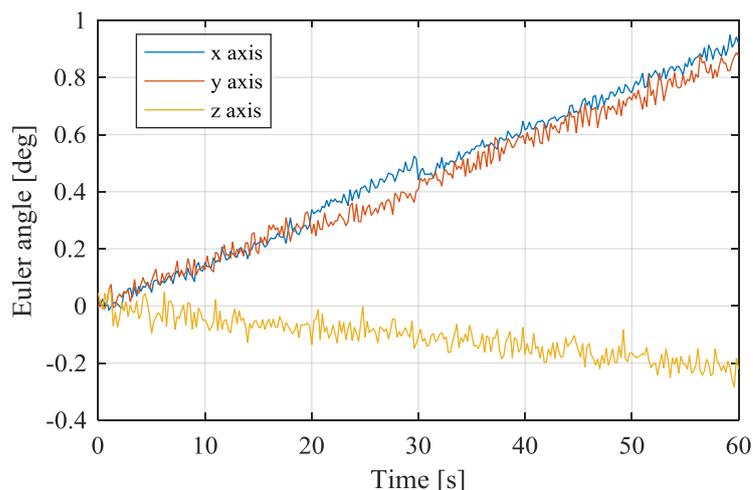


図 16 静止状態でのオイラー角のドリフト

上図から、静止状態であるがオイラー角が徐々に変化（ドリフト）してしまっていることが確認される。この問題の大きな原因として、センサが取得した角速度を単純積分してオイラー角を算出していることが考えられる。

上記のドリフト誤差に対し、今年度用いたジャイロ加速度センサである MPU-6050 は 6 軸センサであることから、並進加速度を用いることによりカルマンフィルタ (KF) で対策することが可能である。ただし、6 軸センサであるため z 軸回りの補正は、ジャイロ加速度センサ単体では行えないこと、そして z 軸回りのオイラー角のドリフト量が他の軸に比べ小さく、定期的にキャリブレーションを行うことにより影響は無視できると考えたため、x 軸と y 軸に関する補正に関して述べる。

KF は、誤差を持った観測値を用いて動的システムの状態を推定するための無限インパルス応答フィルタである。実装した KF は、線形 KF であり、オイラー角の推定をするために、角速度の単純積分値、並進加速度の傾きから得られた角度を用いる。理論に関しては割愛し、実装に関してのみ述べ、コードを次ページに記載する。ただし、`newRate` に角速度の単純積分値を代入、`newAngle` に並進加速度の傾きから得られた角度を代入することにより、返り値としてフィルタリングされたオイラー角である `angle` を得る。この操作を制御周期 0.2 s ごとに計算を行うことにより時々刻々と推定値を更新していく。

次ページに示すようなプログラムを実装した際の静止状態におけるオイラー角のドリフトの様子と KF によって算出したオイラー角を図 17 に示す。

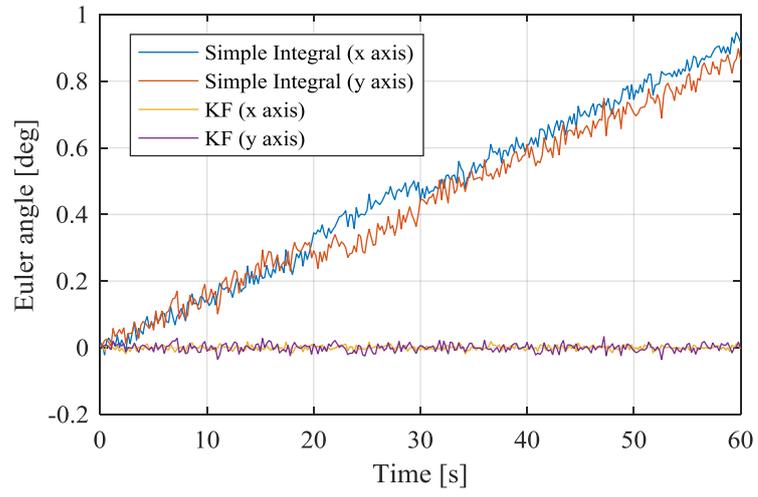


図 17 補正後のオイラー角のドリフト

```

Kalman::Kalman() {
    /* 設計変数 */
    // 駆動雑音-----
    Q_angle = 0.001f;
    Q_bias = 0.003f;
    // 雑音共分散-----
    R_measure = 0.03f;

    angle = 0.0f; // Reset the angle
    bias = 0.0f; // Reset bias

    // 誤差共分散行列-----
    P[0][0] = 0.0f;
    P[0][1] = 0.0f;
    P[1][0] = 0.0f;
    P[1][1] = 0.0f;
};

float Kalman::getAngle(float newAngle, float newRate, float dt) {
    /* Step 1 */
    rate = newRate - bias;
    angle += dt * rate;

    // Update estimation error covariance
    /* Step 2 */
    P[0][0] += dt * (dt * P[1][1] - P[0][1] - P[1][0] + Q_angle);
    P[0][1] -= dt * P[1][1];
    P[1][0] -= dt * P[1][1];
    P[1][1] += Q_bias * dt;
    /* Step 4 */
    float S = P[0][0] + R_measure; // Estimate error

    /* Step 5 */
    float K[2]; // Kalman gain
    K[0] = P[0][0] / S;
    K[1] = P[1][0] / S;

    //Update estimate with measurement zk (newAngle)
    /* Step 3 */
    float y = newAngle - angle; // Angle difference

    /* Step 6 */
    angle += K[0] * y;
    bias += K[1] * y;

    //Update the error covariance
    /* Step 7 */
    float P00_temp = P[0][0];
    float P01_temp = P[0][1];
    P[0][0] -= K[0] * P00_temp;
    P[0][1] -= K[0] * P01_temp;
    P[1][0] -= K[1] * P00_temp;
    P[1][1] -= K[1] * P01_temp;

    return angle;
}

```

B) ジャイロ加速度センサによるデジタルコンパスセンサ補正

本節では、ジャイロ加速度センサによるデジタルコンパスセンサの補正を行う必要性とそれに対する対策として使用した相補フィルタに関して述べる。

デジタルコンパスセンサは温度によるドリフトや、2次元で方位角を算出していることから路面状況によってはセンサ値が正常な値を示さず、それがオフセットとして残ってしまう場合がある。機体は炎天下を走行するため温度変化が非常に大きいこと、また、砂漠という起伏の激しい地面を走行することから、上記のような問題が大きく影響してくると考えられる。この影響を低減するために、相補フィルタの適用を考える。

デジタルコンパスセンサは慣性座標系に対する方位角を取得する。これを相補フィルタにより補正するために、次式で示すような1次の相補フィルタとデジタルコンパスセンサの生データに重みづけして足した重みづけ相補フィルタを定義する。デジタルコンパスセンサの生データに重みをかけて足しているのは、制御周期が遅く相補フィルタのみでは応答が遅れる傾向にあったためである。また、次式の各変数に対する係数や重みは実験により試行錯誤的に算出した。

$$d_i = 0.9 \left(0.93 \left(d_{i-1} + \Delta T \omega_{z,gyro} \right) + 0.07 \theta_{z,accel} \right) + 0.1 d_{compass}$$

$\left(\text{ただし, } d_i: \text{補正方位角, } d_{compass}: \text{デジタルコンパスの生データ, } \Delta T: \text{制御周期}(=0.2 \text{ s}) \right)$
 $\left(\omega_{z,gyro}: z \text{軸回りの角速度, } \theta_{z,accel}: \text{加速度の傾きから算出したオイラー角とする.} \right)$

相補フィルタを用いず生データを用いた場合と重みづけした相補フィルタを用いた場合において、z軸回り 90 deg / 20 s で回転させた実験の結果を図 18 に示す。本実験では、キャリブレーションをモータが駆動しているときに行い、モータがデジタルコンパスセンサに与える影響を最小限にした上で実験を行った。約 8 s 時に 3 センチ程度の段差を乗り越えて機体が一時的に傾いた。下右図は下左図を拡大して表示したものである。

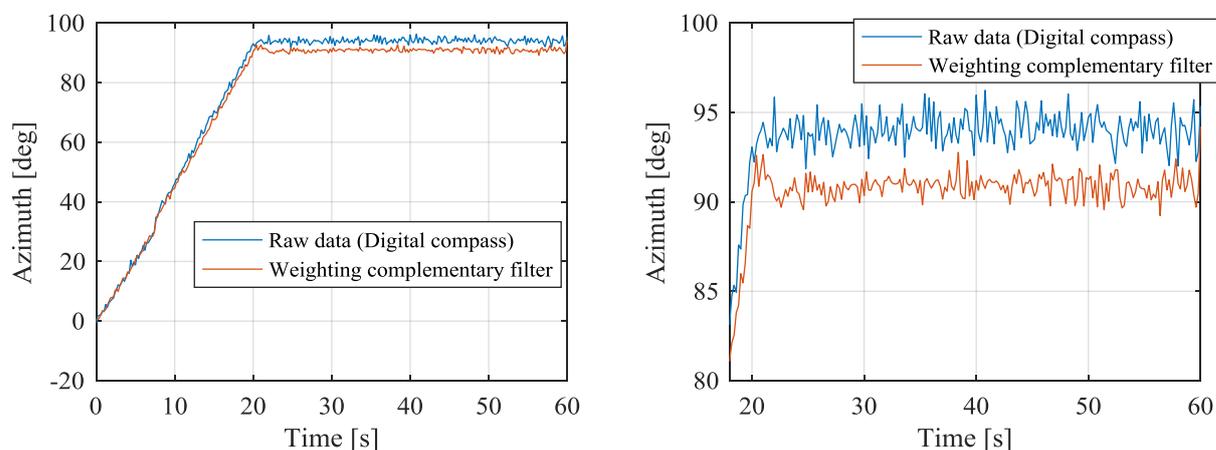


図 18 フィルタリング前後のデジタルコンパスの値

図 18 から、デジタルコンパスの生データを使用した場合には、4 deg 程度のオフセットが生じてしまっているが、重みづけ相補フィルタを用いた場合にはオフセットが 1 deg 程度と約 3 deg 程度の誤差を抑制できている。今回は短距離の走行であったが、長距離走行を考えた場合では誤差がより多く蓄積していくと考えられるため、補正の必要性は高いことが確認された。

2.9 特に工夫した点・苦労した点

Wolve'Z はプリント基板を用いた回路の設計と製作を行っている。設計時に信号線と電源線の配置が近い場合や信号線が長い場合には電磁誘導によるノイズが発生し、モジュールが動作不良を起こすことが多くあった。また、プリント基板は焼いたあとに修正することが困難であるため、設計時点での思考錯誤が重要であった。レギュレーションを満たすよう小さな基板上に素子を収めつつ、ノイズが発生する要因を排除するよう配線設計を行った。また、通信線がモータ駆動による電磁波を受けてシステムがハングアップしてしまう問題が発生した。これに対処するため、ハードウェアとソフトウェアの両方から対策を行った。

また、チームは B4 と M1 で成っており、研究や授業のスケジュールによっては学年全員が作業できない時期が発生することが予想された。したがって、モジュールごとに開発が忙しくなる時期を考慮してスケジュール作成や仕事の振り分けを行った。また、昨年度と比較してメンバーの数が半分であったため、設定するミッションの要素数やスケジュールを昨年度から大幅に変更した。

3 大会結果

3.1 能代宇宙イベント

3.1.1 目的

シーケンスが動作することを確認する。

3.1.2 結果

サクセスクライテリアの達成度

1回目・・・達成なし

2回目・・・達成なし

ランバックの結果を以下の表に示す。

能代	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	50m以上制御走行	ゴール
1回目	○	○	○	○	×	×	×
2回目	○	○	×	×	×	×	×

落下時に CanSat が跳ねてパラシュートの上に落ち、紐に絡まったため、走行を開始できなかった。

3.1.3 取得データ

走行していないため、なし。

3.1.4故障原因解析

パラシュート紐がからまった原因として、パラシュートと機体の距離が近かった事が挙げられる。パラシュートと機体の距離を離すために、分離シートとパラシュートの間に細長いプラスチックのシートを挟んでいたが、シートが柔らかく長さが短かったため、機体のはねたときにシートが曲がり、パラシュートに機体がついてしまった。これは、落下後にシートが伸びるよう素材を固くし、セパレーションを取るために長さを伸ばすことで防ぐことができる。従って、能代後はプラスチックの代わりにメジャーを用い、長さをパラシュートの紐の長さ以上になるよう伸ばした。しかし、シートの長さを長くするとパラシュートが開くのが遅くなる。能代の落下距離は 100 m 前後であり、シートの長さが長くなると着地までに開かないため、能代においては別の解決策が必要である。

3.2 ARLISS

3.2.1 目的

ロケット搭載から目的地へ到達するまでの一連のシーケンスが実行できることを確認する。

3.2.2 結果

サクセスクライテリアの達成度

1回目・・・ミニマムサクセス達成

2回目・・・ミニマムサクセス達成

ランバックの結果を以下の表に示す。

	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	○	○	○	○	×	×	×	×
2回目	○	○	○	○	○	○	○	×

・1回目

1回目の打ち上げでは、走行開始後約1分でソフトウェアがハングアップし、制御不能になった。また、制御中もデジタルコンパスの値が正しくとれておらず、正しい制御指令が出ないことがあった。

・2回目

2回目の打ち上げでは、モータを回した状態で定期的にデジタルコンパスとジャイロセンサのキャリブレーションを行うようにした。落下後にパラシュートと分離し、ゴールへ向かって1.3km走行した後、ゴールから400mの地点でソフトウェアがハングアップした。

3.2.3 取得データ

打ち上げ2回目の制御履歴と位置履歴を図19に示す。また、microSDにより取得したデータを添付する。

画像ファイルの左側は各データの時刻歴、右側は位置履歴を示す。左側のグラフは上から状態、緯度、経度、左右に出力したPWM、制御入力（duty比、正が右、負が左）、CanSatの向き（デジタルコンパスの値）を表す。

また、走行中は3分ごとにデジタルコンパスとジャイロセンサのキャリブレーションを行う。キャリブレーションはモータを回した状態で行うため、その場で旋回しながら行う。

すべてのシーケンス移行をセンサで検知しているため、位置履歴には飛行中、落下中、着地状態の軌跡も色分けして示す。走行フェーズではゴールへ向かって制御できていることがわかる。

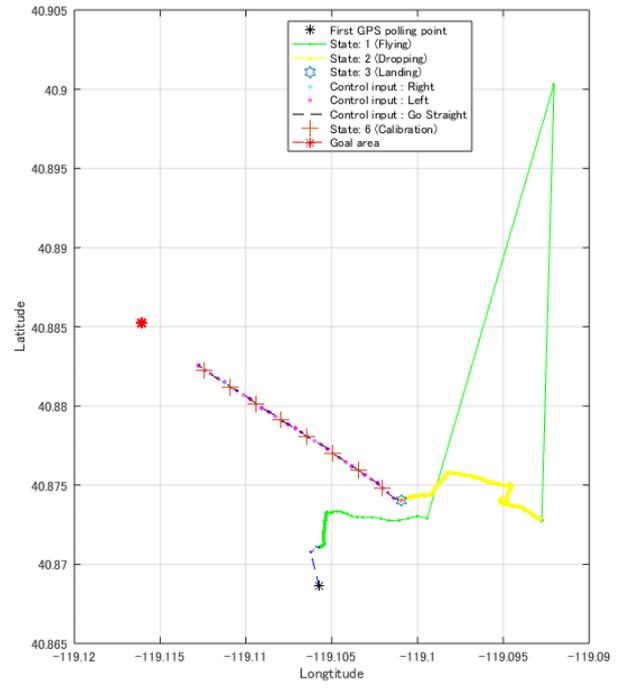
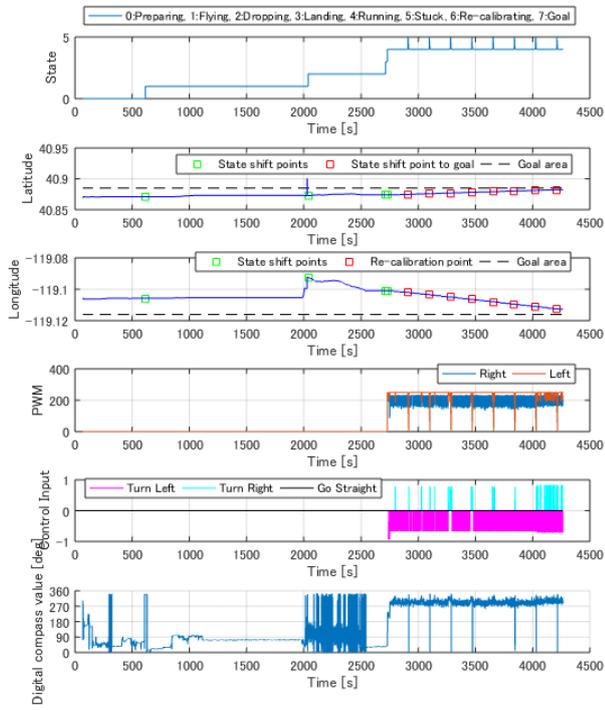


图 19 制御履歴

3.2.4故障原因解析

システムがハングアップしたことについて原因解析を行った。

—回路にノイズが乗る

- 回路上の素子配置により電磁誘導ノイズが発生した可能性
 - 狭い基板内に素子を配置しているため配置の自由度が少ない
 - マイコン (Arduino Mega) の面積が大きく基板を覆っている
 - I2C 通信の素子が 2 つあるなどして面積を占めていた
- 去年購入したセンサを今年に使いまわしたのもあったため性能が悪化していた
- 構造上モータが至近距離にあるため磁気ノイズが乗った
 - モータをスチールのシールドで囲っていたが、そこに電気がたまった可能性
- 電池やモータの導線からの電磁誘導を受けた

—長距離を走ると出る

- 回路にラップをして長距離試験をしたときはハングアップがなかった
 - 砂塵によりショートした可能性
 - 回路がむき出しで裏面のプリントパターンが外にでていた

—ソフトウェアに問題がある可能性

- 例外が起きた際に **While** 文に入り抜け出せない可能性がある

—問題に気づくのが遅れたため、大会中まで問題が長引いた

- BBM の検証時は USB でマイコンに給電しており、回路電池を用いた検証をしていなかった

—走行用電池の出力が不安定であり回路に影響を与えた可能性

- 電池出力の安定性を試験していなかった
- 電源からのノイズを遮断するコンデンサの計算を厳密に行っていなかった

4 まとめ

●故障解析の結果から導き出された対策/今後の課題

今後の課題は、回路のハングアップを解決する事である。原因分析の結果、以下の対策が挙げられた。

- ・回路を基板を2つに分け、配線に余裕をつくり、ベタグラウンドの面積を増やす。
- ・マイコンの変更を検討。
- ・デジコンとジャイロを統合し、9軸センサを使用する。また、センサも最新のものを使用する。
- ・回路周りにシールドをつける（砂塵対策、磁気ノイズの対策）、あるいは回路を内部にしまう
- ・BBMの試験を、電池を用いて行う。使用する電池の安定性を試験で評価する。

●今回の CanSat 機体の良かった点・うまくいかなかった点

- ・良かった点：6 m/s の落下に耐えうる点。センサフュージョンによる誘導を実証した点。
- ・うまくいかなかった点：ソフトウェアのハングアップが起きやすかった点。duty 比が電池によって変化するため、試行錯誤で決定する必要があった点。無線の送信が不安定であった点。

●総括

今年度の ARLISS では回路のハングアップという明確な課題を見出すことができたため、来年度の活動につなげていきたい。スケジュールとしては、BBM 完成から発注した機体が完成するまであまり作業のない期間があったため、来年度は本年度の機体の部品を使用して試験等を行えるようにしたい。

また、今年度は外部の大会には出場しなかった分、Facebook のページを作成して外部へ向けた活動報告を行った。

5 責任教員所感

能代イベントでの実験的検証で明らかになった課題を短い期間で解消し、ARLISS において打ち上げから着地、分離、自律走行の一連の実装が実現できたことは何よりである。ただし、サクセスクライテリアの設定を順位にしたことは反省点であり、次年度以降は数値目標をしっかりと立てて欲しい。さらに、コンセプトあるいはハードウェアの面で新しいことを少なくとも1つは入れた開発を進めてほしい。