

博 士 論 文

CanSat における Deep Learning を用いた
誘導制御手法に関する研究

令和 5 年 3 月

湘南工科大学大学院
工学研究科 電気情報工学専攻 博士後期課程

20 T 2501

秋山 実穂

目次

第 1 章序論	1
第 2 章研究背景・研究目的	10
2-1CanSat とは	10
2-2CanSat の起源	11
2-2-1CanSat と ARLISS の誕生	11
2-2-2 ARLISS の始まり	11
2-2-3 カムバック部門の誕生	12
2-2-4 能代宇宙イベントの誕生	13
2-2-5 種子島ロケットコンテストの誕生	13
2-3 レギュレーション	14
2-3-1 サイズ制限	14
2-3-2 衝撃荷重の耐久	14
2-2-3 減速機構での落下	15
2-2-4 ロスト対策	15
2-3-5 通信条件	16
2-3-6 自律制御	16
2-3-7 審査書の通過	16
2-3-8 表彰	17
2-3-8-1 ARLISS 表彰	17
2-3-8-2 能代宇宙イベント表彰	18
2-3-8-3 種子島ロケットコンテスト表彰	18
2-3-9 その他	19
2-4 事前審査	20
2-4-1 安全審査と設計計画書	20
2-4-2 ミッションステートメントとサクセスクライテリア	20
2-4-3 ミッションシーケンス	21
2-4-4 システム要求項目と試験項目設定	22

2-4-5 設計詳細	22
2-4-6 質量試験	22
2-4-7 機体収納・放出試験	22
2-4-8 準静的荷重試験	23
2-4-9 振動試験	23
2-4-10 分離衝撃試験	23
2-4-11 落下試験	24
2-4-12GNSS データダウンリンク試験	24
2-4-13 通信機電源 OFF/ON 試験	24
2-4-14 通信周波数 ch 変更試験	25
2-4-15 着地衝撃試験	25
2-4-16 走行性能確認試験	25
2-4-17End to End	25
2-4-18 制御履歴レポート試験	26
2-4-19 その他ミッション要求試験	26
2-4-20 ガントチャート	26
2-5 競技部門	27
2-5-1 カムバック部門	27
2-5-1-1 競技内容	27
2-5-1-2 ランバック部門	27
2-5-1-3 フライバック部門	28
2-5-2 ミッション部門	28
2-5-3 遠隔操作部門	28
2-6 大会	29
2-6-1ARLISS	29
2-6-1-1ARLISS とは	29
2-6-1-2 大会スケジュール	29
2-6-1-2 AeroPAC ロケット打ち合わせ協力	30
2-6-1-3 審査書と報告書の提出	30
2-6-1-4 競技環境	31

2-6-1-5 大会の注意事項など	32
2-6-2 能代宇宙イベント	33
2-6-2-1 能代宇宙イベントとは	33
2-6-2-2 スケジュール	33
2-6-2-3 審査書と報告書の提出	34
2-6-2-4 競技環境	34
2-6-3 種子島ロケットコンテスト	35
2-6-3-1 種子島ロケットコンテストとは	35
2-6-3-2 スケジュール	35
2-6-3-3 設計計画書の提出	36
2-6-3-4 競技環境	36
2-6-4 その他の大会	37
2-7 研究目的	37
2-7-1 ARLISS2016 の GNSS 精度によるゴール差	37
2-7-1 研究目的	39
第 3 章 再現実験による安全審査と設計評価	41
3-1 安全審査と設計計画書	41
3-2 ミッションステートメントとサクセスクライテリア	42
3-2-1 ミッションステートメント	42
3-2-2 ミッションステートメントの例	42
3-2-3 サクセスクライテリア	44
3-2-4 サクセスクライテリアの例	45
3-3 ミッションシーケンス	45
3-3-1 ミッションシーケンス内容	45
3-3-2 ミッションシーケンスの例	46
3-4 システム要求項目と試験項目設定	47
3-4-1 システム要求項目	47
3-4-2 システム要求項目の例	47
3-4-3 試験項目設定	49

3-4-4 試験項目設定の例	50
3-5 設計詳細	51
3-5-1 CanSat 機構	51
3-5-2 システム図とアルゴリズム	55
3-6 質量試験	59
3-6-1 質量試験内容	59
3-6-2 質量試験結果	59
3-7 機体収納・放出試験	60
3-7-1 機体収納・放出試験内容	60
3-7-2 機体収納・放出試験結果	61
3-8 準静的荷重試験	62
3-8-1 準静的荷重試験内容	62
3-8-2 準静的荷重試験結果	65
3-9 振動試験	66
3-9-1 振動試験内容	66
3-9-2 振動試験結果	68
3-10 分離衝撃試験	71
3-10-1 分離衝撃試験内容	71
3-10-2 分離衝撃試験結果	74
3-11 落下試験	75
3-11-1 落下試験内容	75
3-11-2 落下試験結果	75
3-12 GNSS データダウンリンク試験	77
3-12-1 GNSS データダウンリンク試験内容	77
3-12-2 GNSS データダウンリンク試験結果	79
3-13 通信機電源 OFF/ON 試験	82
3-13-1 通信機電源 OFF/ON 試験内容	82
3-13-2 通信機電源 OFF/ON 試験結果	84
3-14 通信周波数 ch 変更試験	86
3-14-1 通信周波数 ch 変更試験内容	86

3-14-2 通信周波数 ch 変更試験結果	86
3-15 着地衝撃試験	88
3-15-1 着地衝撃試験内容	88
3-15-2 着地衝撃試験結果	89
3-16 走行性能確認試験	90
3-16-1 走行性能確認試験内容	90
3-16-2 走行性能確認試験結果	91
3-17End to End	93
3-17-1End to End 内容	93
3-17-2End to End 結果	94
3-18 制御履歴レポート試験	95
3-18-1 制御履歴レポート試験内容	95
3-18-2 制御履歴レポート試験結果	95
3-19 その他ミッション要求試験	100
3-19-1 ゴール検知試験内容	100
3-19-2 ゴール検知試験結果	102
3-20 ガントチャート	106
第 4 章 ゴール色の画像認識による誘導制御手法	110
4-1GNSS による誤差の問題点	110
4-2 通信機器における GNSS 電波干渉調査	110
4-3 Xbee による GNSS 測位精度影響実験	111
4-4 GNSS 測位精度影響実験	113
4-5 浮動小数点数精度による測位位置分解能	116
4-6 電波干渉による GNSS 測位精度	117
4-7 カムバックにおける 0m ゴール手法	117
4-8 ゴール画像認識手法の検討	118
4-9 画像認識手法	120
4-10 画像認識による CanSat の誘導制御手法	120
4-11 種子島ロケットコンテストにおける評価	121

4-12 結論	125
参考文献	126
第 5 章 Deep Learning 画像分類を用いたゴール認識制御	128
5-1 色の画像認識による日照条件の色変化	128
5-2 従来の誘導制御手法	128
5-2-1 GNSS による誘導制御	128
5-2-2 色によるゴール画像認識手法	128
5-2-3 色によるゴール画像認識の問題点	131
5-3 提案手法	132
5-3-1 Deep Learning によるゴール認識	132
5-3-2 使用する Deep Learning モデル	133
5-3-3 Deep Learning プログラム構造	136
5-3-4 ゴール画像の学習	136
5-3-5 ゴールの位置認識手法	138
5-4 評価	140
5-4-1 学習結果の評価	140
5-4-2 Deep Learning による認識計算時間	142
5-4-3 メモリ使用量	144
5-4-4 ゴール認識率評価	145
5-4-5 ゴール判定認識率評価	148
5-4-6 ARLISS 2019 における実証評価実験	150
5-5 結論	153
参考文献	154
第 6 章 10m 長距離からの Deep Learning を用いた認識手法	155
6-1 長距離における Deep Learning を利用したゴール画像認識	155
6-2 第 5 章による従来手法	157
6-2-1 CanSat における Deep Learning 画像認識	157
6-2-2 Deep Learning モデル	158

6-2-3 Deep Learning モデルの学習	160
6-2-4 ゴール位置認識手法	162
6-2-5 ROI 分割手法	164
6-2-6 ROI 計算方法	165
6-2-7 画像認識によるゴール誘導制御手法	167
6-3 提案手法	168
6-3-1 従来手法の問題点	168
6-3-2 水平 ROI 分割数の増加手法	169
6-3-3 上下 ROI3 段階移動手法	169
6-4 評価実験	170
6-4-1 ROI 6 分割から 9 分割の認識率	173
6-4-2 上下移動あり ROI 7 分割時の認識率	175
6-4-3 ROI 分割数とゴールからの距離との関係	177
6-4-4 計算時間	179
6-4-5 通信時間	181
6-5 Object Detection との比較	182
6-5-1 認識率の比較	182
6-5-2 計算時間の比較	183
6-6 実証評価実験	184
6-6-1 投下 1 回目	185
6-6-2 投下 2 回目	186
6-6-3 投下 3 回目	187
6-6-4 実証実験結果の考察	189
6-7 結言	191
参考文献	193
第 7 章 Deep Learning を用いた高速目標認識手法の提案	194
7-1 長距離における Image Classification ゴール認識の速度	194
7-2 第 6 章による従来手法	195
7-2-1 Image Classification モデル	195

7-2-2 仕様の整合性の維持	198
7-3 提案手法	200
7-3-1 エッジデバイスでの機械学習アクセラレータ	200
7-3-2 Object Detection 学習	200
7-3-3 コンパイル手法	203
7-4 評価	204
7-4-1 認識率の評価	204
7-4-2 トラフィックコーンに白線がないときの認識率	205
7-4-3 トラフィックコーンに白線があるときの認識率	207
7-4-4 従来手法との認識率の比較	208
7-4-4-1 白線ありの場合の従来手法との比較結果	210
7-4-4-2 白線ありの場合の従来手法との比較結果	210
7-4-5 認識の計算時間	211
7-4-6 アクセラレータを使わないときの計算時間	211
7-4-7 従来手法との認識時間の比較	213
7-5 実証評価	213
7-5-1 能代宇宙イベント 2021	213
7-5-2 あさぎり CanSat 投下試験(ACTS)2021	217
7-6 結論	221
参考文献	222
第 8 章 結論	233
【研究業績】	230
【謝辞】	234

第 1 章 序論

現在，実践的な宇宙開発を体験できる CanSat コンテストが注目を集めている．従来宇宙開発は NASA や JAXA 等の専門機関でないと行えないものであり，大学生がその開発体験をすることは困難であった．1998 年にハワイで行われた University Space System Symposium (USSS) において，大学で開発されたナノサイズの人工衛星を宇宙にロケットで打ち上げる構想が，スタンフォード大学の Bob Twiggs 教授により提案された．この提案はその後 1999 年にアメリカネバダ州ブラックロック砂漠で開催された ARLISS 大会として実現することになる [1][a]．これが CanSat の始まりである．その後 CanSat 等宇宙工学の分野で，実践的な教育活動の実現を支援することを目的とする NPO 法人である University Space Engineering Consortium (UNISEC) [b] が設立され，ARLISS や能代宇宙イベント等の CanSat コンテストを開催している [2]．その他にも JAXA 等が主催する種子島ロケットコンテスト [c] などが日本国内で開催されており，大学生のみならず，高校生から社会人まで参加者数が増えている．実際に ARLISS 2019 では世界中から 30 を超える団体以上が参加し，また種子島ロケットコンテスト 2019 では，参加希望団体数が 50 団体を超えるなど，CanSat に対する関心が高まっている．

CanSat とは缶 (Can) サイズの衛星 (Satellite) の略であり，現在は 350 ml サイズの缶 3 個分のサイズ内で設計された惑星探査ローバを指すものとなっている．CanSat コンテストでは，各チームで独自に開発した惑星探査ローバ (CanSat) を，ロケットや気球，ドローン等により上空へ上げ，そこから放出しパラシュート等で地上へと落下させる．落下後 CanSat はパラシュートを切り離し，あらかじめ地上に定められた目標地点へと向かい自律走行をする．そして最終的に CanSat がどれだけ目標地点まで近づくことができたかを競い合う Comeback Competition 部門 [3-4] がメイン競技となっている [d]．その他にも人工衛星技術に関する様々な独自ミッションを行う Mission Competition 部門もある．独自ミッションの例として，Uwano らは CanSat が不整地をスタックせず走行するミッションを行っており，[5]，Coin らは気候モニタリングをミッションとして行っている [6]．本研究では CanSat コンテストの Comeback Competition に焦点を当てる．

CanSat は宇宙開発を想定しているため、全て自律制御で動作する必要がある、競技終了まで一切手を触れることはできない。数ある CanSat 大会の中で、最も難しいと言われているのが ARLISS である。ARLISS ではロケット打ち上げ時に CanSat に大きな加速度が加えられ、この衝撃に耐えられる必要があるが、我々はこの加速度について詳細に測定した。ロケットはアメリカロケット愛好家団体 AeroPac[e]により提供される。このロケットに CanSat を搭載し、上空約 4,000m まで打ち上げる。このときに CanSat には持続的に約 8 G の加速度が加えられ、打ち上げ後約 5 s で時速約 1,500 km/h に達する。その後、高度約 4,000 m で CanSat は放出されるが、このときに約 30 G の加速度が加えられる。さらにそこからパラシュートで地表へと落下するが、地上との衝突時に CanSat には約 100 G の衝撃が加えられる [7]。CanSat はこの衝撃に耐え、問題なくパラシュートを切り離し [8]、ゴール地点へと向けて走行していくことを要求される。ARLISS は広大なブラックロック砂漠で実施されるため、CanSat はゴール地点から数 km 離れた地点に落下することが多く、時には 10 km 以上離れた地点に落下することもある。このため、CanSat は数 km の距離を問題なく走行できる能力が必要となる。

ARLISS では 2015 年まではゴール地点数 m まで到達できる CanSat はほとんどなく、ロケット打ち上げや地上への落下衝撃で壊れたり、さらに数 km 走行することができず壊れたりするものが多かった。しかし各チームの技術が進歩し、ARLISS 2016 においては、上位 3 チームがゴール地点から約 3 m, 4 m, 5 m まで近づくことができるようにまで進歩した。この時 CanSat のゴールへの誘導制御は GNSS を用いて行われていたが、この 3 チームのゴール地点までの距離の差は GNSS の測位誤差によるものであるため、ARLISS は 2016 年で GNSS 誘導の限界に達したことになる [9-11]。本論文の第 2 章では研究背景・目的として以上の内容について述べる。

CanSat は、実際の宇宙開発と同様にロケットに搭載可能であるか JAXA などの機関が定めた安全審査により必要とされる設計要件を備えているか実験により徹底的に調査を行い、設計評価を行う。CanSat に必要とされる性能を再現実験により調査を行う。本論文第 3 章では再現実験による安全審査と設計評価として以上の内容について述べる。

CanSat は ARLISS2016 以降, GNSS の測位誤差を超え, いかにもゴールまで距離 0 m を目標に近づけるかが競われるようになった. GNSS の測位誤差は搭載されている回路の無線電波により干渉しているか調査を行った. また搭載可能なサイズの小型マイコンで動作可能である 0m まで誘導制御手法として, 目的地に設置されたパイロンの色を画像認識する手法において, 画像認識制御手法を提案した. 我々 Super NOVA チームは種子島ロケットコンテスト 2018 において, 同様にカメラによりゴールの色を認識してゴールまでのゼロ距離誘導を達成して優勝した. この後, 他のチームもカメラを用いてゴールの赤色を認識することにより距離 0 m を目標とするチームが複数現れるが, 次第にこのゴールの赤色を認識する手法の問題点が明らかになってきた. 本論文の第 4 章では GNSS 測位精度を改善するゴール誘導制御手法として以上の内容について述べる.

ゴール地点に置かれたパイロンの赤色を YCrCb や HSV 色空間で範囲指定し, 2 値画像を生成し, ゴール地点を画像認識により求める手法[12]は計算量も少ないため, CanSat のようにサイズの制約により使えるコンピュータのリソースが限られている環境においては有効な手法ではあるが, 実際に CanSat に適用してみると様々な問題がある.

例えば天候の変化により照明の色温度が変化するため, 晴天時にゴールの色範囲を合わせると曇天時には認識できず, 曇天時に合わせれば晴天時に認識できなくなる. 両方に対応するため認識範囲を広げると, 枯れ草等もゴールと誤認識してしまうなど誤判定が増加する. またゴールからの距離約 4 m が認識距離の限界であるが, 実際の競技では GNSS の誤差により 5~6 m 程度離れた地点から認識する必要が多々生じるが, その場合はゴールを見つけることができない問題がある. このため, コンテスト実施直前にゴール地点にて色範囲を調整する必要がある, それをしないとゴール認識精度が十分ではなく, ゼロ距離ゴールできなくなるなど, 問題点が多い.

このような問題を解決するため, 我々は Deep Learning 画像分類を用いてゴールを認識させる手法を提案した. サイズや重量等制約が厳しい CanSat において, 計算量の多い Deep Learning を適用するための CanSat の回路構成及び学習,

認識に用いる適切な Deep Learning モデルの提案, 及びその計算時間, メモリ使用量, 認識率について調査を行った. そして ARLISS 2019 にて 1 回目 0.05 m, 2 回目 0 m, 3 回目 0 m を達成し, Accuracy Award 1 位, Best Mission Award 1 位, Technical System Award 2 位を獲得し, Overall Award で 1 位となり総合優勝を獲得した[13]. 本論文の第 5 章では Deep Learning 画像分類を用いたゴール認識制御として以上の内容について述べる.

Deep Learning 画像分類を用いたゴール認識は, 天候による照明条件の変化に強く, 無調整で高精度にゴールを発見できることから, CanSat においても非常に有効である. 我々は Deep Learning を用いて画像分類を行っているため, そのままではゴールの位置がカメラ画像のどこにあるのかがわからない. そこで水平方向に画像全体の部分領域である Region of Interest (ROI)を 3 分割, 5 分割として複数回画像認識を行うことにより, ゴールの位置を求めている. この手法では, ゴールから約 6 m までが問題なく認識できる限界であり, それを超えると急速に認識率が低下し, ゴールを見つけられなくなって行く. 特に GNSS の測位誤差に加えて強風により CanSat が流される等, いくつかの問題が重なると, ゴール地点から最悪の場合 10m 近く離れた地点からゴールを認識する必要が生じることがある. しかしそのような場合はゴールの認識ができず, ゴール付近をさまよい続けることになる.

そこで, 我々は従来の ROI 5 分割ではゴールが見つからない時に, さらに ROI 分割数を増やして認識を行うことにした. ROI 分割数が増えると ROI の上下範囲も狭くなるため, 5 分割以上ではさらに ROI を上下方向に 3 段階に変化させて認識させることにし, この手法の有効性について評価実験をおこなった. また, ROI 分割数を増やすことにより計算量が増大し, 計算時間が増えるため, 認識率との関係から適切な分割数及び上下移動のあり, なしについて実験により明らかにした. さらに提案手法を実証評価するため, コロナ禍のため中止された ARLISS 2020 の代替大会として開催されたあさぎり CanSat 投下試験 (ACTS)に出場し, 約 10 m の距離から提案手法の画像認識によりゴールまでのゼロ距離までの誘導に 3 回連続で成功し, 提案手法の有効性を確認した. 本論文の第 6 章では 10m 長距離からの Deep Learning を用いた認識手法として以上の内容について述べる.

Deep Learning を用いた画像認識手法には、提案手法のように Deep Learning 画像分類を用いて、その ROI を変化させることにより、ゴールの位置を認識させる手法の他に、Object Detection を用いる手法がある。我々は軽量 Object Detection として代表的な SSD MobileNet V1 [14]を用い、提案手法とゴールからの距離における認識率の比較を行った。その結果提案手法の方が SSD MobileNet V1 よりもゴールからの距離が遠い場合に認識率が高く、また計算時間も短いことを確認した。

一般的に Object Detection は計算量が多いため、組込み機器に用いる場合は、Y. Yoshimoto らのように処理の重い計算の一部を FPGA に行わせ高速化する [15] ことが多いが、専用のハードウェアを用意する必要がある問題がある。我々は CanSat のようにサイズ制約が非常に厳しい環境において、Linux が動作する組込みシステムとして最低性能クラスである Raspberry Pi Zero を用い、単純なモデルを用いて ROI 分割を工夫することにより、CPU の計算能力のみで SSD MobileNet V1 よりも、ゴールから遠距離において高い認識率かつ高速で認識する手法を提案する。サイズや重量等制約条件が厳しい CanSat において、計算量の多い Deep Learning 画像認識を適用し、従来手法では困難であったゴールまで 10m の距離からゴール認識を可能にする手法について述べる。

CanSat はサイズや重量制限が厳しいため、搭載できるコンピュータの種類は限られており、我々は Raspberry Pi Zero W を使用している。サイズ制約によりこれ以上大きなコンピュータの搭載は困難である。Raspberry Pi Zero は性能が低いため、Deep Learning に使用できるモデルも小さなモデルに限られる。そのため、我々は可能な限り小さなサイズの Image Classification モデルを用いた。

Image Classification モデルであるため、画像のどの位置にゴールがあるかは分からないため、画像を水平方向に複数分割した Region of Interest (ROI)を用い、それぞれ Image Classification 処理を行うことにより、画像の左右どの位置にゴールがあるのかを判定している。ゴールが見つからないと ROI を増やしていくため、1回のゴール認識に、最大で 40 回ゴール認識処理を行うことになる。このため、1回の認識で 10 秒以上の計算時間がかかることになる [16]。

実際の CanSat では、GNSS でゴール付近まで近づいた後、停止してカメラ画像の Deep Learning 画像認識を行い、ゴールが見つからない場合は少し左へ旋

回して停止し、再びカメラ画像の認識を繰り返し行うという動作を行う。ゴールが見つかった場合、そちらの方向へステアリングを向け、数秒走行したのち停止して、上記を繰り返し、画像認識結果がゴール判定可能な画像となったと判定したところでゴール判定を行い、動作を停止する、という制御を行っている。

このため、Deep Learning によりゴール認識は誤認識が少なく、高い精度でゴールの認識が可能であり、より確実にゴールまで0mまで誘導可能になったが、認識時間がかかるため、ゴールにたどり着くまでに時間がかかるのが欠点であった。特に日本で行われている CanSat 大会には 15 分間の制限時間が設けられているため、計算時間によりこの制限時間をオーバーしてしまうことがある問題があった。

そこで、我々は従来よりも圧倒的に計算量が大きいため採用が困難であった、Object Detection モデルである MobileNet SSD V1 及び V2 を用いながら、Coral USB Accelerator[g]を Raspberry Pi Zero[h]と連携させることで従来手法を超える高い認識率を実現しながら、約 0.3 秒で認識させることに成功した。また提案手法の実証評価のため能代宇宙イベント 2021 及びあさぎり CanSat 投下試験 (ACTS)2021 に参加し、どちらも最短時間記録でゴールまでの距離 0m を達成し、優勝した。Coral USB Accelerator を Raspberry Pi Zero と連携させることにより、高速でかつ高い精度でのゴール認識を可能にし、実証評価でも高い成果を示した手法について述べる。本論文の第 7 章では 10m 長距離からの Deep Learning を用いた高速目標認識手法の提案として以上の内容について述べる。

本論文は 8 章で構成される。第 2 章では研究背景・目的について述べる。第 3 章では再現実験による安全審査と設計評価について述べる。第 4 章では GNSS 測位精度を改善するゴール誘導制御手法について述べる。第 5 章では Deep Learning 画像分類を用いたゴール認識制御について述べる。第 6 章では 10m 長距離からの Deep Learning を用いた認識手法について述べる。第 7 章では Deep Learning を用いた高速目標認識手法の提案について述べる。第 8 章で結論を述べる。

【参考文献】

- [1] N. Sako, Y. Tsuda, S. Nakasuka et al., “CanSat Suborbital Launch Experiment – University Educational Space Program Using Can Sized Pico-Satellite,” *Acta Astronautica*, Vol.48, Issues 5-12, pp. 767-776, 2001.
- [2] Nakasuka, Shinichi. “Students' Challenges towards New Frontier-Enlarging Activities of UNISEC and Japanese Universities.” *Transactions Of The Japan Society For Aeronautical And Space Sciences, Space Technology Japan 7 (ists26)*, 2009.
- [3] M. E. Aydemir, M. Celebi, S. Ay, E. V. Vivas, F. Calle Bustinza and D. Phan, “Design and implementation of a rover-back CANSAT,” *Proceedings of 5th International Conference on Recent Advances in Space Technologies - RAST2011*, pp. 800-803, 2011.
- [4] M. çebebi et al., “Design and navigation control of an advanced level CANSAT,” *Proceedings of 5th International Conference on Recent Advances in Space Technologies - RAST2011*, pp. 752-757, 2011.
- [5] Fumito Uwano, Yusuke Tajima, Akinori Murata, and Keiki Takadama, “Recovery System Based on Exploration-Biased Genetic Algorithm for Stuck Rover in Planetary Exploration”, *J. Roboti. Mechatron.*, Vol.29 No.5 pp. 877-886, 2017.
- [6] Colin, Angel, and Manuel Jimenez-Lizárraga. “The CanSat technology for climate Monitoring in small regions at altitudes below 1 km,” *IAA Climate Change & Disaster Management Conference*, 2015.
- [7] T. Saito and M. Akiyama, “Development of Rover with ARLISS Requirements and the Examination of the Rate of Acceleration that Causes Damages During a Rocket Launch,” *J. Robot. Mechatron.*, Vol.31, No.6, pp. 913-925, 2019.
- [8] M. Akiyama and T. Saito, “Study on Parachute Entanglement Prevention Method Using Image Recognition in CanSat,” *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*, pp. 1629-1634, 2020.

- [9] T. Saito and M. Akiyama, "Analysis of Log Data in ARLISS 2016 of a Planetary Exploration Rover," *Bulletin of Aichi University of Technology*, Vol.15, pp.19-25, 2018.
- [10] T. Saito and M. Akiyama, "Analysis of Results in ARLISS 2016 of a Planetary Exploration Rover," *The Special Interest Group Technical Reports of IPSJ, Embedded System Symposium 2017*, pp.112-113, 2017.
- [11] T. Saito, M. Akiyama, et al., "Practical Evaluation and Functional Design of a Small Autonomous Robot in ARLISS," *Proceedings, JSME Conference on Robotics and Mechatronics : Robomec '17*, 2P2-A12, 2017.
- [12] M. Akiyama and T. Saito, "A Novel CanSat-Based Implementation of the Guidance Control Mechanism Using Goal-Image Recognition," *2020 IEEE 9th Global Conference on Consumer Electronics (GCCE)*, 2020, pp. 580-581, 2020.
- [13] M. Akiyama and T. Saito, "Study on a Method to Guide Cansat to the Goal at a Distance of 0m Using Deep Learning," *The IEICE Transactions on Information and Systems (Japanese Edition)*, Vol. J104-D, No.7, pp. 540-550, 2021.
- [14] Liu, Wei, et al. "Ssd: Single shot multibox detector." *European conference on computer vision*. Springer, Cham, pp. 21-37, 2016.
- [15] Y. Yoshimoto and H. Tamukoh, "FPGA Implementation of a Binarized Dual Stream Convolutional Neural Network for Service Robots," *J. Robot. Mechatron.*, Vol.33, No.2, pp. 386-399, 2021.
- [16] M. Akiyama and T. Saito, "A Novel Method for Goal Recognition from 10 m Distance Using Deep Learning in CanSat," *J. Robot. Mechatron.*, Vol.33, No.6, pp. 1359-1372, 2021.

Supporting Online Materials:

- [a] UNISEC, “ARLISS A Rocket Launch for International Student Satellites,”
<http://unisec.jp/activities/arliss.html/> [Accessed Feb 20, 2023]
- [b] University Space Engineering Consortium UNISEC <http://unisec.jp/>
[Accessed Feb 20, 2023]
- [c] Tanegashima Rocket Contest Executive Committee, “Tanegashima Rocket
Contest,” <http://jaxa-rocket-contest.jp/> [Accessed Feb 20, 2023]
- [d] Constituted by UNISEC-Japan faculties (Dr. Hattori, Nakanishi, and
Sakamoto) and UNISEC Student Organization (UNISON), “CanSat
Regulation at ARLISS, July 1, 2019: ver.5,”
[http://unisec.jp/history/arliss2019/pdf/CanSatRegulation_byUNISEC_2019_v
er5.pdf/](http://unisec.jp/history/arliss2019/pdf/CanSatRegulation_byUNISEC_2019_ver5.pdf/) [Accessed Feb 20, 2023]
- [e] AeroPac Network, “AeroPac.” <http://www.aeropac.org/> [Accessed Feb 20,
2023]
- [f] “Raspberry Pi,” <https://www.raspberrypi.org/> [Accessed Feb 20, 2023]
- [g] Coral USB Accelerator
<https://coral.ai/products/accelerator/> [Accessed Feb 20, 2023]
- [h] Raspberry Pi Foundation
<https://www.raspberrypi.org/> [Accessed Feb 20, 2023]

第 2 章 研究背景・研究目的

2-1 CanSat とは

CanSat(Can Satellites)とは小型模擬宇宙探査機であり，現在 CanSat 競技として様々なコンペティションが開催され注目を集めている．[1] CanSat は宇宙探査を想定し，ロケットや気球などに格納し打ち上げを行い，空中から放出し，その後地表面に着地し，宇宙ミッションを行う．ロケット打ち上げから地表面の着地までには機体に大きな衝撃が加わり，その衝撃に耐えられる耐衝撃性のある機体性能が求められる．また，ロケット格納可能寸法や重量のサイズ要件があり，宇宙探査を想定しロケット打ち上げ後人の手で修理することができないため完全自律制御であるレギュレーションの下で競技が行われる．自律制御であるため各シーケンスにおいてどんな状況下でも全ての動作を確実に行う必要があり，ロケット打ち上げ時の衝撃などで途中故障が生じ動作不能になればその時点で競技終了となり，従来の宇宙開発同様に確実に動作を行う究極の信頼性が求められる．競技ではあらかじめ指定された目標位置まで自律制御でどれだけ接近することができるか距離を競うカムバック部門がある．カムバック目標位置にはパイロンが設置され，そのパイロンからの接近距離で記録が決定される．カムバック制御手法には様々な接近制御手法が存在し，その制御手法においては様々な研究が行われている．

2-2 CanSat の起源

2-2-1 CanSat と ARLISS の誕生

1998 年 11 月ハワイで開かれた大学宇宙システムシンポジウムの USSS (University Space System Symposium) でスタンフォード大学 宇宙開発研究所の Bob Twiggs 教授が缶サイズの衛星の構想提案をきっかけに、CanSat が誕生した。

最初 350mL 缶サイズの超小型人工衛星として複数の大学が作ったものを USSS で運用デモンストレーションを行う実験コンセプトの構想があったが、人工衛星の打ち上げ機会の確保が困難であった。次に挙げられた実験コンセプトとして、高空 (サブオービタル) の打ち上げ実験の ARLISS (A Rocket Launch for International Student Satellites) [a]を年 1 回行う構想が提案された。ARLISS ではアメリカのアマチュアロケットグループの AEROPAC の協力により AEROPAC が作ったハイパワーモデルロケットを利用し、CanSat をロケット内に格納し、高度 12,000 フィート (約 3.6km) までロケット打ち上げ後空中に CanSat を放出し、地表面着地までの落下中に様々なミッションを行う実験が実際に開催された。ARLISS により衛星環境に近い状況での実験を行うことができる貴重な機会が設けられた。

2-2-2 ARLISS の始まり

1999 年 9 月 11 日に最初の ARLISS がアメリカのネバダ州北部ブラックロック砂漠で開催された。日本の大学では東京大学と東京工業大学がそれぞれ 3 機 CanSat を開発し、アメリカからはアリゾナ州立大学とケネディ中学校が参加した。CanSat はロケットから放出後、減速機構パラシュートを展開し 15-20 分かけ地上に着地するが、その間に実際の衛星ミッションに似た様々なミッションが行われた。ARLISS は毎年 1 度開催され、参加国や参加大学は年々増加していき、2012 年には日本から 12 大学と 1 高校、韓国から 1 大学、アメリカから 2 大学が参加し、総勢 100 名を超える学生が ARLISS に参加した。ARLISS には実験の他に技術交流会が存在し、国内外の様々な学生との意見交換の場としても貴重な機会となっている。

2-2-3 カムバック部門の誕生

2001年東京大学 中須賀真一教授の提案により、カムバック・コンペ（Comeback Competition）が誕生した。カムバック・コンペではロケットから放出された CanSat があらかじめ指定された目標地点まで自律制御により接近制御を行い、目標地点までどれだけ接近することができたかその距離を競う部門である。カムバック・コンペでは最も接近距離が近い CanSat が優勝となる。

ロケットから放出された CanSat は GNSS 位置情報を用いて現在位置の情報を得て、あらかじめ設定した目標位置との差分から目標位置への誘導制御を行う。放出された時点からパラフォイルの端をサーボモータで引っ張ることで、空中から目標位置まで接近制御を行うような、空中制御を用いる CanSat をフライバックと呼ぶ。フライバックにはパラフォイルの空中制御以外に 2022 年現在では固定翼機や動力飛行などの空中接近制御による手法も存在する。

ロケット放出後、空中制御しない場合には、パラフォイルで落下中に上空の風に流され、風の状態から目標位置から約 1-10km 離れて地表面へ着地する。着地後は地表面を走行することでカムバックを行う CanSat をランバックと呼ぶ。ランバックでは地表面走行時に凹凸などの走行困難な地面でも走行できる走破性が求められる。ランバック機は 2 輪型 CanSat を中心に、4 輪駆動形やクローラー型など様々な走行手法が存在する。

カムバック・コンペでは従来の CanSat 打ち上げ実験と比較して様々な実践効果を得られる。カムバック機に求められるシステムは、オンボードコンピュータ、メモリ、通信機、センサ・アクチュエータ及び構造体など、実際の衛星と同様のバスシステムが搭載され、実際の衛星運用と同様に全てのシステムが正常に動作した場合のみ目標の結果を得ることができる。競技ではロケットに格納後は実際の宇宙環境と同じように、風や地面状態など予測不可能で定性的な環境下で、機体やシステムに異常が発生しても全ての動作を完全自律制御でする必要があり、実際の宇宙開発同様の究極の信頼性が求められる。また、宇宙環境では正常に動作しなくなった衛星に対して、原因追求やメンテナンスを行うことが行うことが不可能であるが、競技終了後に回

収めることで問題点や改良点を明確化できる。

また、コンペ形式であることから各大学で性能を競い合い、モチベーションが高まる利点がある。

2-2-4 能代宇宙イベントの誕生

ARLISS をきっかけに様々な CanSat コンテストが実施され始めた。2005 年から秋田県能代市で開催される能代宇宙イベントでは日本最大規模のアマチュア宇宙イベントとして、CanSat 競技やハイブリットロケットやモデルロケットなどのロケット競技が行われた。能代宇宙イベントでは CanSat 投下方法はハイパワーモデルロケットからの放出ではなく、高度 50-20m からの気球またはドローンによる投下手法を用いている。

能代宇宙イベントの大会運営は NPO 法人 UNISEC (University Space Engineering Consortium) が運営をしており、企業からの協賛も多く受けている大きなイベントである。ARLISS も同様に UNISEC の運営下で開催されている。

2-2-5 種子島ロケットコンテストの誕生

能代宇宙イベントの他にも、いくつか CanSat コンテストが誕生した。2005 年 3 月に九州大学 大学院工学研究院 航空宇宙工学部門 麻生茂 教授の提案と当時の JAXA 種子島宇宙センター次長松山為昌氏の協力の下で種子島ロケットコンテスト実行委員会が立ち上げられ、種子島ロケットコンテストが開催された。[i]

種子島ロケットコンテストは鹿児島県南種子町にある JAXA 種子島宇宙センターで、能代宇宙イベント同様に CanSat 競技及びロケット競技が行われる。CanSat 投下方法は高度 50-20m から気球かクレーンからの投下となる。JAXA 種子島宇宙センターで開催されるため、大会期間内には JAXA 職員からの基調講演などがあり、JAXA との交流があることが魅力の 1 つである。

運営は UNISEC の関与は無く、種子島ロケットコンテスト実行委員会として宇宙工学専門の各大学の教員や南種子町の主催により開催されている。

2-3 レギュレーション

2-3-1 サイズ制限

CanSat はロケット格納可能である必要があり，サイズ制限が存在する．レギュレーションで提示されるサイズは CanSat 格納筒のサイズを示しており，規定内のサイズ内であることだけでなく，ロケット格納する筒のサイズ内に格納可能で CanSat 放出時自重で落下できることが条件である．また格納筒から CanSat が放出された後に格納筒内に何も残らないようにしなければならない．

CanSat サイズは Open Class と 350ml Class が存在する．ARLISS 及び能代宇宙イベントでは Open Class のサイズは直径 146 mm×高さ 240 mm で重量は 1050g であり，350ml Class では直径 66mm×高さ 240mm で 350g である．種子島ロケットコンテストではサイズは 1 つに統一されており，直径 154mm×高さ 300mm で重量 1050g となっており，ARLISS 及び能代宇宙イベントより規定サイズは大きくなっている．

2-3-2 衝撃荷重の耐久

ARLISS では AEROPAC のハイパワーモデルロケットのロケット衝撃荷重に耐えられる必要がある．ロケットの衝撃荷重に耐えられず CanSat 部品などが放出時に部品がバラバラに落下するような事があれば人体に落下する可能性があり危険であるため，CanSat はロケット衝撃荷重に耐え競技を行う必要がある．ARLISS のレギュレーションではロケット衝撃荷重に対し実験推奨値があり，荷重方向：垂直，準静的荷重 10G，正弦波振動 30-2000Hz で 15G または同等のランダム振動としており，ロケットと CanSat の分離衝撃には 40G としている．ロケット衝撃荷重に耐えられるか公式実験推奨値や各団体の過去の打ち上げサンプリングデータからのデータ根拠値から，準静的荷重試験，振動試験，分離衝撃試験を行い，参加大学の宇宙工学などの選考する各責任教員に提出する事前審査に合格する必要がある．

2-2-3 減速機構での落下

CanSat は約 1kg の物体が自由落下する事で人体直撃し事故を起こすことを避けるため、減速機構で減速させ落下させる必要がある。減速機構での減速は CanSat 全大会で義務付けられている。能代宇宙イベントに関しては落下速度に規約があり、減速機構で減速させかつ 5m/s 以上の落下速度で落下させる必要がある。5m/s 以下の落下速度では落下中風に流されフィールド外の走行困難な場所に落下した場合には、CanSat の再投下が認められないが、5m/s 以上の落下速度で落下させた場合には風が強く環境によりフィールド外へ着地したと判定され、再投下が認められる。また、ARLISS 及び能代宇宙イベントでは減速機構を用いて CanSat が減速することを審査書にて事前試験することで証明し、審査に合格する必要がある。

2-2-4 ロスト対策

ARLISS ではロケットで 4000m まで打ち上げを行い、その後減速機構で減速しながら落下をする。落下から着地まで風の影響で CanSat は大きく流され、風の強さや風向き次第で目標位置から約 1-10km 離れて落下する。広範囲な落下位置であるため CanSat がどこに落下したか特定するのが困難で機体が発見できずロストしてしまう場合がある。そのため CanSat がどの位置に落下したか特定ができる必要があり、長距離での通信で位置情報の特定やパラシュートの色が空との色の補色であるなど落下中の CanSat の特定が行える必要がある。ARLISS ではロスト対策として CanSat の位置特定の試験を行い審査に合格する必要がある。能代宇宙イベントではフライバックや 100m 以上の投下の CanSat は会場近辺の山や海に飛行しロストする可能性があるため、ロスト対策の審査が必須になるが、ランバックの場合は任意となっている。種子島ロケットコンテストでは機体や減速機構などに空と同色の色は用いてはいけないとされている。

2-3-5 通信条件

CanSat 全大会においてロスト対策やリアルタイムでの CanSat 制御状態の確認などのダウンリンク通信により、ロケット打ち上げ信号通信や気球キャリア開閉の通信に干渉しないことが求められる。FCC 認証かつ 100mW 以下の機器以外はロケット打ち上げ終了まで無線周波数を OFF にする必要がある。また他チームとの同じ周波数でも通信干渉せずチャンネル変更ができることが必要とされる。

2-3-6 自律制御

CanSat は宇宙空間で運用する前提であるため、競技では自律制御であることが必要とされる。また宇宙空間では故障しても人の手で修理することができないため、CanSat においても競技中にどんな状態であってもメンテナンスや遠隔操作などを行うことはなく、動作不可になった場合にはその地点で競技終了となる。また、カムバック競技においては落下した地点で距離の記録となると落下したその時の運での競技となってしまう場合があることを考慮し、競技終了後に日時や GNSS 位置情報などでモータ制御を行ったかどうか証明するための制御履歴を提出することが義務付けられており、制御履歴が認められた場合のみ大会記録となる。

2-3-7 審査書の通過

ARLISS 及び能代宇宙イベントでは CanSat のレギュレーション全てを満たし、CanSat の投下実験として危険がないかを証明する安全審査書を運営に提出し、合格する必要がある。審査書に通過することができなければ、CanSat のレギュレーションや安全要件を満たしていないため、原則実験参加は認められない。審査書はレギュレーションに沿ってサイズ、落下中の減速、ロケット荷重の耐久、ロスト対策、通信条件の遵守、自律制御の証明ができる制御履歴の提出ができることを記載する。またキャリア収納から競技終了後に制御履歴提出までの本番同様の試験を End to End として行う。また、実際の宇宙開発で用いられるプロジェクトツールである、ミッションステートメントやサクセスクライテリアを設定し、競技後の結果はその目標に

どの程度達成できたかを評価を行う。

また種子島ロケットコンテストでは競技参加チームに制限があり、制限チーム数以上の応募があった場合には、CanSatの詳細設計を記載した設計計画書を提出し、種子島ロケットコンテスト実行委員会の教員の審査により、選抜を行う。

2-3-8 表彰

2-3-8-1 ARLISS 表彰

ARLISS の表彰では Overall Winner(総合優勝)、Accuracy Award(Comeback 部門優勝)、UNISEC Award(教員評価賞)、Technical System Award(技術評価賞)、Best Mission Award(ミッション評価賞)が存在する。Overall Winner は総合優勝で Comeback での距離や、制御アルゴリズム、独創性などを評価する。評価方法はチーム間投票で新規性や技術難易度、結果の達成度などの様々な項目を段階評価し得点が高いチームが選ばれる。Accuracy Award は Comeback 部門での目標位置への距離が1番近いチームが選ばれる。Comeback 部門において制御履歴が認められた場合のみ表彰対象記録となる。遠隔操作を行う CanSat は対象外となる。同等の距離である場合は技術難易度や到達時間の速さなどで順位を評価する。UNISEC Award は ARLISS に参加している UNISEC 所属教員による評価から投票で選ばれる。主に専門的に技術を考慮した教員の評価から賞が送られる。Technical System Award は技術の評価を行う賞で、システムの技術的新規性やシステムの技術的難易度によって、チーム間投票で技術評価項目の高いチームが選ばれる。Best Mission Award はミッションの評価賞で、ミッションの優位性、ミッションの実現性、ミッションの独自性で評価される。チーム間投票によりミッション評価項目の高いチームが選ばれる。それぞれの賞には表彰状と表彰盾が送られる。各賞に1, 2, 3位まで順位がつくことがあるが、2, 3位は表彰状のみの贈呈となる。

2-3-8-2 能代宇宙イベント表彰

能代宇宙イベントではランバック部門 1～3 位，フライバック部門 1～3 位，ミッション部門能代 CanSat 大賞・企業評価賞，団体貢献度賞が存在する．ランバックは地表面を走行する CanSat が制御履歴を提出し記録と認められたチームの中で目標位置への接近距離が最も近いチームから順番に 1～3 位の表彰状が送られる．フライバックは空中制御による CanSat が制御履歴を提出し記録と認められたチームの中で目標位置への接近距離が最も近いチームから順番に 1～3 位の表彰状が送られる．ミッション部門での能代 CanSat 大賞は各提出書類や技術交流会時のプレゼンや実験結果からチーム投票で選ばれる．また，企業評価賞では各協賛企業が技術点や新規性を評価し贈呈される．企業賞では企業からの副賞として技術機器などの贈呈がある場合もある．

2-3-8-3 種子島ロケットコンテスト表彰

種子島ロケットコンテストではロケットコンテスト大賞，CanSat ペイロード部門優勝・準優勝，その他審査員特別賞が存在する．ロケットコンテスト大賞は総合優勝であり，CanSat 部門とロケット部門で優勝した中から，次回以降の参加者が目指すのにふさわしい先進的なチームに賞が与えられる．表彰状とトロフィーが与えられる．CanSat ペイロード部門では自律制御である制御履歴を提出し認められた記録の中で目標位置からもっとも接近距離が近いチームが優勝し，その次に近いチームに準優勝となる．優勝チームには表彰状とトロフィーが与えられ，準優勝チームには表彰状が与えられる．審査員特別賞には種子島ロケットコンテスト審査員が評価を行い，各協賛企業からの副賞を贈呈する賞で，例年賞の位置付けは協賛企業により異なる．2019 年の例では種子島宇宙センター所長賞としての奨励賞や，IHI エアロスペース賞としてのベストプレゼン賞（ロケット部門），宇宙技術開発賞としてのチーム賞や，川崎重工業賞としてのベストデザイン賞や，コスモテック賞としてのベストプレゼン賞（CanSat 部門）や，日本宇宙フォーラム賞としての敢闘賞や，種子島観光協会賞としてのユーモア賞や，日本エア・リキード賞としての安全賞や，三菱重工

業賞としてのプロダクト賞や、中菱エンジニアリング賞としてのユニーク賞や、日本航空宇宙学会西部支部賞としてのアカデミック賞や、アークシステムズ賞としての情報技術賞や、有人宇宙システム賞としてのチャレンジ賞や、種子島宇宙センターロケット打ち上げ 50 周年記念特別賞などがある。審査員特別賞には表彰状の協賛企業や種子島ロケットコンテスト実行委員会による副賞が贈呈される。

2-3-9 その他

その他大会ごとにレギュレーションが 1 部異なる項目もいくつか存在する。ARLISS では、CanSat をロケット格納後取り出してはいけない規定がある。格納後ロケット打ち上げ準備にて 1 時間以上経ったか、ロケットトラブルが生じた場合以外の CanSat の取り出しを行うことは認められずその時間を考慮した制御シーケンスやバッテリーを搭載する必要がある。また、ARLISS ではフライバック等の機体において翼やローターを持つ重さ 0.55 ポンドから 55 ポンド (0.249kg から 24.9kg) の CanSat は、FAA の規則により無人航空機システム (UAS) -模型飛行機-とみなされ、飛行条件を FAA 規則に準拠する必要がある。

能代宇宙イベントや種子島ロケットコンテストには円滑に競技進行をするため 15 分の競技時間制限が存在し、時間内にゴールをしなければならない。カムバック部門では時間終了時の地点を記録として扱う。また、種子島ロケットコンテストでは制御履歴の提出は競技終了後 50 分以内である必要があり、制御履歴提出できない場合には記録に 30m 距離を追加した結果として扱う。

2-4 事前審査

2-4-1 安全審査と設計計画書

宇宙開発では実際にロケットに搭載する人工衛星や探査機等は宇宙空間で本来目指しているミッションがどのような条件下であっても動作可能か徹底的に事前に検証試験を行い，JAXA などから打ち上げ搭載許可を得る安全審査がある．安全審査は1つ1つの搭載部品や機能をあらゆる環境下での動作チェックを約100回行い，宇宙空間で何が起こっても必ず動作ができるまで精度を高めることが求められる．

ARLISS 及び能代宇宙イベントには CanSat を投下する際に危険が無いかを証明する安全審査書の提出が義務付けられる．安全審査書を期限内提出し合格を貰わない限り大会競技に参加することは原則できない．サイズ規定を始め，ロケット打ち上げ衝撃荷重に機体が耐えられるか，落下が減速機構を用いて行えるかを主に審査を行う．

種子島ロケットコンテストでは安全審査書の提出は求められていないが，設計計画書の提出が義務付けられている．設計計画書はミッション内容や機体やシステムの詳細設計を p.6 以内で記述し，その他試験の実施は要求されない．種子島ロケットコンテストでは参加チーム制限により50チーム以内まででの競技が行われるが，応募が50チームを超える場合は設計計画書から種子島ロケットコンテスト実行委員会の宇宙工学専門の大学教員による審査により参加チームが選定される．

2-4-2 ミッションステートメントとサクセスクライテリア

CanSat において何を実現したいかを明確にするため，ミッション定義をする必要がある．そのためにミッションの目標を背景・根拠とともに明確に述べたミッションステートメントやミッションステートメントに対して1つ及び複数のミッション項目・システム構成が存在しそれらの成功段階を詳細かつ定量的に述べたサクセスクライテリアを記載する必要がある．これらは実際の宇宙開発でも取り入れられているミッション定義方法である．[j]

ミッションステートメントに対して一つ及び複数のミッション項目・システム構成が存在し、それらの成功段階を詳細かつ定量的に述べたものがサクセスクライテリアとなる。サクセスのレベルはミニマムサクセス、フルサクセス、エクストラサクセスに分類される。ミニマムサクセスは挑戦的なミッション目標を設定するため、独立した複数ミッション機器を搭載する場合等では、その一部の機能喪失が生じた場合でもクリアできる最低限の目標として設定することができるためのサクセスレベルを設定する。フルサクセスは予定していた要求を満たし、計画通りの成果を得るため何を達成したいか明確にサクセス定義を設定する。エクストラサクセスはフルサクセスを達成した上で、さらにそれを上回る成果を得ることであり、そのサクセスレベルを設定する。ただし単に目標を設定するだけではなく、上回る成果を得るために克服すべき課題とその対策を明らかにする必要がある。フルサクセスの設定は必須であるが、ミニマムサクセスとエクストラサクセスは任意となっている。

2-4-3 ミッションシーケンス

ミッションシーケンスではキャリア収納からミッション達成までの流れを図と文章で簡潔に示して説明する。図式にすることでミッションの流れをわかりやすく表記する。

2-4-4 システム要求項目と試験項目設定

システム開発を行う上で実施すべき業務内容を整理し文書化するためのシステム要求定義を行うために、システム要求項目を設定し、開発過程において何を行うべきか整理を行う。CanSat の場合にはレギュレーションを満たした CanSat ができているかを中心に項目を設定し、達成したいミッションのための項目はミッション要求項目として追記で設定する。

システム要求項目及びミッション要求項目からその項目を満たす試験を設定する。システム要求項目及びミッション要求項目の検証を行う検証項目名を記載し、対応する要求番号を記載する。また実施予定日や実施日を記載する。

2-4-5 設計詳細

設計詳細として CanSat の機構や電子回路システム，アルゴリズムなどを説明する．機構には機体の寸法と重量や，外観図として実際の様々な方向から撮影した写真や CAD データなどにより構造の詳細を記載する．また減速機構の設計や動作も合わせて記載する．また電子回路システムとしてブロックダイアグラムを記載し，電子回路システムの概要を記載する．また，搭載電子部品について名称や型番を明記した表を記載する．アルゴリズムとしてフローチャートを図式し，システムの構造を詳細に記載する．

2-4-6 質量試験

質量試験はレギュレーションの質量以内に CanSat が設計されているか確認を行い，質量測定時の証拠動画や画像からその質量の証明を行う．レギュレーションの質量は 1,050g 以内であり，その質量内であることを証明するため質量計にパラシュートを含む CanSat を載せ，計量しその結果の数値を撮影することで質量内であることを証明する．質量の計測は競技の前にも行われ，その際も質量以内でないと競技を行うことができない．質量試験ではロケットに格納する状態の CanSat の質量を質量計で計測し，質量内であるか確認を行う．

2-4-7 機体収納・放出試験

機体収納・放出試験では規定の内径 146mm，高さ 240mm の収納筒であるキャリアに，CanSat が収納することができ，その後自重で落下し放出ができることを，実際の規定内の筒に収納し放出ができている様子を撮影し動画提出を行うことで，規定内寸法のキャリアでの競技が問題なくできることを証明する．

2-4-8 準静的荷重試験

準静的荷重試験では ARLISS のロケット打ち上げ時に垂直方向にかかる静荷重に CanSat が耐えられるか確認を行う試験である。準静的荷重によりロケット内で CanSat が故障し部品がバラバラの状態で放出されれば、人に落下することを考慮すると危険であるため、ロケット準静荷重衝撃に耐えられることを証明できなければロケットに CanSat を搭載することはできない。ロケット荷重では ARLISS レギュレーションではロケット荷重のかかる推奨値として 10G の試験条件で行うことが参考値として記載されている。また実験値としては各団体がロケット打ち上げ時の実際の測定値や実験での測定値から試験条件の荷重値で実験を行うこともできる。

2-4-9 振動試験

振動試験は ARLISS のロケット打ち上げ時の振動に CanSat が耐えられるかどうか確認を行う試験である。ロケット振動によりロケット内で CanSat が故障し部品がバラバラの状態で放出されれば、人に落下することを考慮すると危険であるため、ロケット振動衝撃に耐えられることを証明できなければロケットに CanSat を搭載することはできない。ロケット振動では ARLISS レギュレーションではロケット振動のかかる推奨試験条件として正弦波振動 30-2000Hz で 15G または同等のランダム振動の試験条件で行うことが参考値として記載されている。

2-4-10 分離衝撃試験

分離衝撃試験は ARLISS のロケットと CanSat が分離し減速機構のパラシュートが開傘したときの衝撃に CanSat が耐えられるかの試験である。ロケットとの分離後のパラシュート開傘衝撃で CanSat が故障し部品がバラバラの状態で放出されれば、人に落下することを考慮すると危険であるため、準静荷重に耐えられることを証明できなければロケットに CanSat を搭載することはできない。分離衝撃として ARLISS レギュレーションでは 40G の衝撃が推奨値として記載されている。また実験値としては各団体がパラシュート開傘時の実際の測定値から試験条件の荷重値で実験を行うこともできる。

2-4-11 落下試験

能代宇宙イベントでは CanSat がキャリア放出後に減速機構で落下する際に強風で競技フィールド外へ CanSat が落下し、ランバックやミッション競技が困難になる場合がある。競技フィールド外へ落下したときは再度競技として CanSat 投下する権利がある。しかしパラシュートの落下速度が遅い場合には何度もやり直しとなってしまうため、レギュレーションで 5m/s 以上の減速機構の終端速度で落下する設計であることが記載されている。実際にパラシュートにて終端速度で降下している速度が 5m/s 以上であるか、落下中のセンサデータにより測定する試験を行う。また、落下している映像を撮影し動画提出を行う。

2-4-12 GNSS データダウンリンク試験

ARLISS ではロケットで高度 4000m まで打ち上げ後、CanSat は分離しパラシュートなどの減速機構で落下する。4000m からの落下では落下中の上空の風の状態により着地地点がどこになるか予測ができず、CanSat を検索することが困難になり、CanSat が見つからずロストしてしまう場合がある。ロストを防止するため長距離から CanSat の居場所を把握できる通信機能などを搭載しロスト対策をする必要がある。通信機器などで長距離から CanSat の居場所を特定できるかどうか疑似環境で試験を行い、特定できた試験のデータなどを提出する。

2-4-13 通信機電源 OFF/ON 試験

ARLISS では打ち上げロケットの無線通信と CanSat の無線通信が電波干渉の無いよう、ロケットから放出されるまで CanSat 無線機の電波は FCC 認証かつ 100mW 以下の機器以外は OFF にできるようにしておく必要がある。無線電波が干渉しないかどうか、擬似的な環境で無線機の試験を行い、動画や通信データなどの提出を行う。

2-4-14 通信周波数 ch 変更試験

CanSat 競技では各チームの CanSat の無線通信が電波干渉しないよう同じ周波数の電波無線を使用するチームと競技が重なる場合には周波数変更やチャンネル変更ができる仕様しておく必要がある。通信周波数 ch が変更でき、他同じ周波数帯の無線機と干渉せず競技が行なえることを変更している画面の提示や動画撮影により証明する。

2-4-15 着地衝撃試験

CanSat では落下後に地表面に着地した衝撃で CanSat が故障することが多い。CanSat にはロケット荷重や分離衝撃などの様々な衝撃が加わるが、着地衝撃はその中でも最も衝撃が大きい。着地衝撃試験は能代宇宙イベントの審査書では必須試験としている。実際に減速機構が終端速度に達する高度から落下させた時の衝撃や、終端速度に達する衝撃を計算で求めその高さから自由落下させた衝撃を与え、その後 CanSat の動作に問題がないか試験を行う。

2-4-16 走行性能確認試験

CanSat はランバックにおいて走行時に轍や大きな草により走行ができなくなる場合がある。走行性能確認試験ではランバック走行時の走破性能があるか確認を行う試験である。能代宇宙イベントでの審査書では必須試験項目となっている。能代宇宙イベントなどの草地などを擬似した路面での走行テストを行い、走行可能かどうか試験を行う。

2-4-17 End to End

End to End は競技の最初から最後まで通しでの動作実験することで各種シーケンスに問題がないかを確認する試験である。充足を確認した設計の機体によって、ロケットへの装填から打ち上げ後の回収までを模擬した End-to-end 試験を実施できており、今後、安全性に関わる大幅な設計変更はないようにする。CanSat の動作を本番と同じ手順で行う。各シーケンスを自律で行えることを確認できる動画を提出する。

2-4-18 制御履歴レポート試験

カムバック競技において落下した地点が偶然ゴール近くであり，動作できずに記録となり，落下した地点で競技結果となることを防ぐため，落下後に GNSS 位置情報データによるモータの制御値を制御履歴として制御履歴担当審査教員へ提出し制御が認められることで，カムバック部門での距離の記録が認められる．

制御履歴レポート試験では規定の制御履歴 End to End 試験のデータから作成し，大会終了後の大会報告書で制御履歴の提出できることを確認する．

2-4-19 その他ミッション要求試験

ミッション要求試験では大会側が求めた安全審査項目と別に，各団体が定めたミッションや目的に応じて必要な試験を追加し，試験を行う．

2-4-20 ガントチャート

CanSat 審査書では CanSat の開発計画においてガントチャートにて詳細に計画を立て，その計画に問題がないか審査を行う．

2-5 競技部門

2-5-1 カムバック部門

2-5-1-1 競技内容

カムバック部門は CanSat において予め指定された目標位置へ CanSat が自律制御でどれだけ接近できるかの距離を競う。安全審査などを通過した CanSat が、規定サイズの収納筒に格納され、ロケットまたは気球で高高度まで上空に打ち上げ落下させる。落下後に減速機構で CanSat を減速させ、減速機構との離脱を行った後ランバックまたはフライバック方法にて目標位置であるゴールまで制御誘導を行い、ゴールまでどれだけ近づけたかの距離を競う。その後偶然落下した場所で距離の記録とならないために、時系列や GNSS 位置情報からのモータ制御値の算出とその説明を行い、その走行軌跡を図式化した制御履歴を提出する必要がある。制御履歴が専門の大学教員により整合性があるか確認後に認定が許可されると距離が記録となる。ランバックとフライバックは別々で競技を行うこともあれば、カムバック部門として統合して行う場合もある。

2-5-1-2 ランバック部門

カムバックにおいてランバックは地表面を走行するカムバック機を競う。ランバックは規定サイズ内で地面の凸凹や轍に対して走破できる必要がある。また地表面着地時には着地衝撃に耐え、その後減速機構との離脱を行った後、地表面を走行しゴールまで誘導制御を行い、その接近距離の近さを競う。その後制御履歴を提出し制御を行ったか確認後に記録が認定される。

ランバック機体種類は様々あり、2 輪タイプが主流であるが、中には 4 輪駆動や 6 輪駆動、クローラー型など様々な走行タイプが存在する。フライバックと比較しランバックタイプの機体のチームが多い。

2-5-1-3 フライバック部門

カムバックにおいてフライバックは飛行タイプの接近手法を行うカムバック機である。フライバックは1番最初に地表面に接触した着地地点の位置を記録とする場合や、一旦地面に到着後再び飛行を行いゴール判定を行った地点を記録とする場合など大会などにより記録定義は異なる。飛行手法でのゴールまでの誘導制御を行い、その接近距離の近さを競う。その後制御履歴を提出し制御を行ったか確認後に記録が認定される。

フライバック機体の種類は様々あり、動力飛行や固定翼、パラフォイルでの誘導接近手法など様々存在する。大会によってはドローンの規制など関係で動力飛行タイプを禁止している場合もある。

2-5-2 ミッション部門

ミッション部門では参加チームごとに設定したミッションを行うタイプでのCanSatであり、カムバック部門のようなゴールまでの接近距離での評価は問わない。ミッション部門でも自律制御である必要があり、安全審査を通す必要がある。ミッションの評価は技術や結果をプレゼン発表など通じて参加チームの投票などで競技の順位を決定する。

ミッション機体は様々な種類があり、土壌採取や物資運搬などチームにより多くのCanSatタイプが存在する。

2-5-3 遠隔操作部門

遠隔操作部門は2022年種子島ロケットコンテストより新設された部門であり、従来CanSatは自律制御が原則であったが、実際の宇宙探査機は地上局からの遠隔操作を行うことから、CanSatにおいてもテレメトリなどの遠隔操作での制御でカムバックなどを行う部門である。宇宙空間で地上局からの遠隔操作を想定するため、遠隔操作者はコンテスト会場外からの相互通信情報からアップリンクなどを行い、直接の目視でない状況での遠隔制御である必要がある。2022年はコロナ禍によりオンライン開催となったが今後新部門として期待が高まっている。

2-6 大会

2-6-1 ARLISS

2-6-1-1 ARLISS とは

ARLISS とは A Rocket Launch for International Student Satellites の略式でアメリカネバダ州のブラックロック砂漠にて開催される CanSat の国際大会である。NPO 法人 UNISEC 運営下により、アメリカ現地のアマチュアロケット団体 AeroPAC の協力のもと AeroPAC のロケットで打ち上げ 4000m 高度で CanSat を放出し競技を行う。大会前には安全審査書の合格が義務付けられており、大会後は報告書を提出する。大会は 6 日間あり、初日及び最終日には設営やプレゼンにより発表会や結果報告会、技術交流会がある。また 4 日間には競技が行われ競技 2 回まで大会記録として扱うことができる。ブラックロック砂漠は人が近くに生活していないロケット打ち上げなどは安全に行うことができる環境であるが、その参加には危険も伴うため安全に大会を行うことが重要視されている。

2-6-1-2 大会スケジュール

1 日目はブラックロック砂漠から 1 番近い小さい街である Garlach に集合し、技術交流会を行う。その後ブラックロック砂漠に車にて全団体と拠点まで移動し、会場設営及びカムバック部門のゴール場所を決定し各団体で GNSS 位置情報の取得を行う。

2 日目-5 日目の 4 日間は CanSat 競技日で、この 4 日間でロケット打ち上げ 2 回までが競技としての記録となり、各団体で打ち上げるタイミングは自由である。しかし現地での調整を行った後打ち上げる団体が多く、その場合最終日に打ち上げが集中してしまうため、競技日 1 日目及び 2 日目に 1 回は打ち上げを行わなければならない制約がある。タイミングは自由であるがブラックロック砂漠は風の強弱の天候変化が激しく、強風であれば打ち上げを一旦中止する場合もあるため、風の状態で打ち上げタイミングを決めることが重要である。ブラックロック砂漠には危険防止のため日没後進入不可となっており、日中のみの競技となる。

6 日目は最終日で競技は行われず、午前中に朝食 MTG と呼ばれている英語での結果報告会や表彰を行う。AeroPAC と共に朝食を取りながら結果報告会で各団体が CanSat のアピールや結果報告をプレゼンにて発表を行い、そのプレゼンから団体ごとに技術点や新規性など様々な項目で得点をつけその集計結果から表彰を行う。午前中で ARLISS は終了するが、その後レンタカーなど返却後最寄りの空港まで向かうのには時間がかかるため最終日に航空券を取るの控えることを推奨している。

2-6-1-2 AeroPAC ロケット打ち上げ協力

アメリカのロケット愛好家団体 AeroPAC のロケットの提供のもとで ARLISS が行われる。ロケットは同一の仕様書で作られており、ロケット全長は約 2.4m、直径約 156mm、重量 11kg である。エンジンはエアロテック社の M1419W で日本では入手不可能なクラスのエンジンである。エンジン直径が 98mm、平均推力が 1,419N である。上空でエンジン部とそれ以外の 2 つに分離し、2 つにパラシュートが開かれる。パラシュートによりノーズコーンが下を向き、ペイロード部内に格納されていた CanSat が下向きに放出される。CanSat とのパラシュートとの絡まりを防止するため 2 段階で CanSat との分離を行う。

AeroPAC にはロケット燃料代のみ各団体が引き渡し競技を行うため、CanSat 参加団体はお礼として日本のお土産を贈呈する慣習がある。

2-6-1-3 審査書と報告書の提出

ARLISS にはロケット格納可能なサイズ要件や、ロケットの衝撃荷重で故障しないか、無線通信がロケットや他団体との CanSat と干渉を受けないか、制御履歴の提出が可能かなどレギュレーションを満たした CanSat であるか事前に安全審査書を運営に提出し、合格をする必要がある。審査員は参加団体の責任教員であり、その多くは宇宙工学等を専門とした大学教員によって専門的な観点からの審査が行われる。また審査に合

格できなかった場合にはロケット打ち上げは認められない。

また、ARLISS 終了後には報告書として、大会の結果や CanSat の設計仕様を詳細に示した書類を提出する必要がある。CanSat は実際の宇宙開発と違い事後回収が可能であるため、結果が構想通りでなかった場合には FTA 分析などを用いて原因追求を行う。

2-6-1-4 競技環境

ARLISS はブラックロック砂漠で開催されるが、ブラックロックは平らな土地で約 160km の広さがありロケット打ち上げにおいて安全に実施することができる。地表面は平らな砂漠土壌であり、乾燥している地表面となっている。地表面には車が通ったタイヤ跡などで轍ができている場合があり、轍の大きさは CanSat をスタックさせてしまうものもあり、ランバックの走行においては轍にスタックしない走破性が必要となる。またスコールなどの影響で地表面が乾燥しておらずドロドロになっている時もある。そこからの乾燥によって大きく地割れを起こしている時もあり、その状態の地面でも走行できるようにタイヤなど設計する必要がある。

また、ブラックロック砂漠では風の強弱が大きく、無風の状態のように風の弱いときもあれば、風速 5m/s を超える台風のような強風や、ストームが起こり砂嵐で危険な状態な強さになるときなど様々である。強風時には危険のためロケット打ち上げを一時休止する場合もある。競技では風の強さが打ち上げ一時休止する手前のような強さであると CanSat は大きく流され、過去の事例ではゴールから 10km 程度の地点で着地し、そこからランバックを行うチームもあった。またほぼ無風の状態である場合には 1km 前後での着地でランバックができる場合もあり、走行距離が大きければ大きいほど大きな轍に遭遇する可能性は高くなりバッテリーの消費や機体の消耗も大きくなり故障しやすくなるため、風の状態を考慮し競技のタイミングを決定することが重要である。

2-6-1-5 大会の注意事項など

ARLISS にはその他注意しておくべき事項が様々存在する。アメリカ渡来の飛行機での CanSat 持ち込みには、リポバッテリー運搬は防火袋を用意したり、CanSat を危険な機器と判断されないために CanSat 競技での参加に必要であると説明文書の空港対応書類などを用意する必要がある。また日本とは違うアメリカでの防犯意識を持っておくことも重要である。また砂漠でもし命の危機で救急病院に搬送する事案が起きた場合にはドクターヘリを呼ぶ必要があり、その金額は 1000 万円にもなることから海外保険に加入する必要がある。またブラックロック砂漠には携帯電話などの電波は全く届かず、広大で砂漠での目印などもなく遭難しやすい危険な環境であるため、定期的にトランシーバーなどの無線機で運営と頻繁に連絡を取り合い、遭難を防止する取り組みも行っている。また遭難防止用に遭難しても 1 晩車で待機することができる水分を車に搭載しておくことが周知されている。

またブラックロック砂漠は人里離れた砂漠地帯であり、一番近い街ではスーパーマーケットなどはない小さな街である Gerlach から拠点まで車で約 30 分、その次に近い街 Fernley からは車で約 2 時間程度あり、競技会場に車で向かう時間も多く必要である。

2-6-2 能代宇宙イベント

2-6-2-1 能代宇宙イベントとは

能代宇宙イベントは秋田県能代市で毎年 8 月に行われる日本最大規模のロケット打上及び自律ロボット制御のアマチュア大会である。ロケット部門と CanSat 部門の競技の開催及び一般公開日として競技会場で様々なイベントを行う。NPO 法人 UNISEC 運営及び秋田コンソーシアムや協賛企業下により開催される。CanSat 競技は ARLISS のようなロケットではなく、気球またはドローンによる投下を行う。大会前には安全審査書の合格が義務付けられており、大会後は報告書を提出する。大会は 6 日間あり、初日は設営、2-4 日目の 3 日間は競技日であり競技 2 回まで大会記録として扱うことができる。また競技期間には技術交流会や懇親会を行う。5 日目は一般公開日で、6 日目は撤収作業である。2020 年度や 2021 年度のコロナ禍では開催期間の短縮が行われている。日本開催の大会ではあるが 8 月の開催や開催場所から熱中症対策や安全対策などを徹底している。

2-6-2-2 スケジュール

1 日目はフィールド整備及び設営を行う。能代宇宙イベント CanSat 競技場所は大きな雑草が生い茂りそのフィールドの草刈り作業や開催作業場所などの設営を行う。作業は参加チーム全団体で行われるが、競技フィールドが大きいため 1 日をかけ行われる。

2 日目-4 日目は CanSat 競技が行われる。3 日間の中で予め運営が決定した投下順に 2 回投下を行う。時間が余っている場合には記録にはならないが 3 回目投下ができる場合もある。雨天の場合に一旦投下中止になる場合もある。投下順が厳密に決まっているため投下時刻に集合できない場合には失格となる。また例年競技後の夜に技術交流会や懇親会、場合によっては BBQ の交流会などが行われる。技術交流会では技術の意見交換が行われ、ミッション部門のプレゼン及び投票が行われる。懇親会では役員の挨拶や、表彰が行われ、技術交流会のような参加チーム同士で意見交換をしたりする。BBQ では交流会の延長のような各団体と意見交換ができる。

5 日目は一般公開日と呼ばれるイベント日であり、ステージの催しや表彰、体験型コンテンツとして 1 部の各団体でテント内で宇宙工学関連の工作体験教室を開いたり、地元の中学生のロケット打ち上げイベントなどがある。食事ができる露店もあり、イベント参加団体以外にも一般の地域の住民も参加している。

6 日目の最終日の午前中は撤収作業があり、設営テント等を撤収する。午前中に終了し、そこから大会終了として解散となる。

2-6-2-3 審査書と報告書の提出

ARLISS と同様に能代宇宙イベントでは事前に安全審査書を提出し合格する必要がある。審査書は運営の学生または過去参加したことがある OB に依頼した審査員で審査を行う。格納可能なサイズ要件や、無線通信が他団体の CanSat と干渉を受けないか、制御履歴の提出が可能かなどレギュレーションを満たした CanSat であるか事前に安全審査書を運営に提出し、合格をする必要がある。ARLISS のようにロケットでの打ち上げは行わないためロケット荷重の耐久性は求められないが、ロケット衝撃荷重の耐久試験を行う団体には賞への審査が有利になることもある。また審査に合格できなかった場合には投下は認められない。

また、ARLISS 同様に能代宇宙イベント終了後には報告書として、大会の結果や CanSat の設計仕様を詳細に示した書類を提出する必要がある。CanSat は実際の宇宙開発と違い事後回収が可能であるため、結果が構想通りでなかった場合には FTA 分析などを用いて原因追求を行う。

2-6-2-4 競技環境

能代宇宙イベント CanSat 部門は秋田県能代市浅内第三鉱さい堆積場で行われる。会場は大きな草が生い茂っており大会初日に 1 日かけて草刈りを行うが、草の凸凹が大きく CanSat のサイズとしては走破性に特化したタイヤなど機体に何か工夫をしなければ、ランバック機においての走行は困難である。また雨天時においては地表面はドロドロになり、大きな水たまりが存在しより走行が困難になる。また ARLISS ほど風の強弱は無

いが、風の強いときが存在し、落下時にフィールド外へ着地することを避けるため、レギュレーションでは減速機構の落下速度は 5m/s 以上としている。また競技は高度約 20-100m の範囲で気球またはドローンによる投下を行う。

2-6-3 種子島ロケットコンテスト

2-6-3-1 種子島ロケットコンテストとは

種子島ロケットコンテストは毎年 3 月頃に鹿児島県種子島の種子島宇宙センターで行われ、ロケット部門と CanSat 部門の競技の開催及び JAXA 職員などの講演会を行う。種子島や大学機関などの種子島ロケットコンテスト実行委員会下により開催される。CanSat 競技は ARLISS のようなロケットではなく、気球またはクレーンによる投下を行う。CanSat 競技には 50 チームまでの参加制限があり上限を超えた応募があった場合には、大会前には設計計画書を提出し、その評価から 50 チームが選ばれる。大会は 4 日間あり、初日は任意ではあるが事前に大会受付と機体審査を行う。2 日目は午前中は 2 日目から参加する団体の大会受付と機体審査があり、午後からは開会式と各団体プレゼンによる技術発表会がある。3 日目には競技が行われるが、種子島ロケットコンテストは CanSat 投下回数については 1 回のみである。3 日後の夜には技術発表会と表彰式があり、4 日目には workshop として JAXA 職員や宇宙工学の専門家による講演が行われる。

2-6-3-2 スケジュール

1-2 日目では、1 日目の参加は任意であるが、午後に事前受付と事前機体審査がある。これらは 2 日目午前にも行われるが、混雑防止として 1 日早く到着した団体は初日に済ませておくことができる。事前受付は団体名を申告し配布資料などを受け取る、機体審査は主にサイズ規定内の CanSat であるか実寸による測定を行う。2 日目は午前に事前受付と事前機体審査がある。午後からは技術発表会があり、各団体の CanSat の設計をプレゼン発表する。技術発表の内容は賞などの審査の評価対象となる。

雨天の場合にスケジュールが最終日の **workshop** と変更になる場合もある。投下順が厳密に決まっているため投下時刻に集合できない場合には失格となり、競技では 15 分以内に終える必要がある。

3 日目は競技日である。1 日間で予め運営が決定した投下順に 1 回のみ投下を行う。雨天の場合に一旦投下中止になる場合もある。投下順が厳密に決まっているため投下時刻に集合できない場合には失格となる。夜には技術交流会と表彰があり、技術交流会では各団体と意見交換ができ、その後表彰式が行われる。

4 日目は **workshop** として、JAXA 職員や宇宙工学の専門家による講演会が行われる。また、任意であるが、希望者には JAXA 施設見学ツアーや、模型ロケット制作教室が行われる。競技日が雨天である場合には、最終日を競技日とすることもある。

2-6-3-3 設計計画書の提出

ARLISS や能代宇宙イベントのように事前審査書にてレギュレーションに沿った **CanSat** であるか試験による審査はないが、参加チーム制限の 50 チームを超えた場合には設計計画書の評価から種子島ロケットコンテスト実行委員会内の宇宙工学などが専門である大学教員による審査による評価で 50 チームの選定が行われる。設計計画書は 6p 内で **CanSat** の特徴や設計の計画スケジュールを記載しその技術難易度やスケジュールの妥当性を評価する。

2-6-3-4 競技環境

種子島ロケットコンテスト鹿児島県種子島の種子島宇宙センターの広場で行われる。地表面の状態は芝生が生えており、能代宇宙イベントの状態と比べて凸凹の少ない状態であり、かつレギュレーション上はサイズ要件が能代宇宙イベントより大きく設計できることから、ランバックによる走行としては比較的走破しやすい環境となっている。風の強弱は **ARLISS** ほどではないが、強風により競技を一時中断する場合もある。また、強風によりフィールド場外に落下した場合には再投下が認められている。

2-6-4 その他の大会

その他 CanSat の大会はいくつか存在する。2020 年と 2021 年度にはコロナ禍により海外渡来の ARLISS 実施を中止し、その代替大会である ACTS(あさぎり CanSat 投下試験)を実施した。ACTS でのレギュレーションなどは ARLISS と同じ要件であり、実施会場はあさぎりフードパークドローン飛行場の広場で行われ、CanSat 投下方法は気球によって行われる。実施日程は競技日 2 日間であるため全チーム投下ができるよう制限時間は 15 分となっている。

また伊豆大島共同実験も 3 月頃に伊豆大島にてロケット部門と CanSat 部門が並行して行われる。2011 年 6 月から毎年開催されており、伊豆大島三原山噴火口近くの裏砂漠と呼ばれる場所で地表面は火山灰で覆われている。

またコスモパーク加太は 2009 年 2 月から実施されており、関西空港の埋立地造成のための土壌採掘跡地である平らな土地が広がっている場所で実施される。

2-7 研究目的

2-7-1 ARLISS2016 の GNSS 精度による到達距離

CanSat はレギュレーションに様々な制約条件があり自律制御であるため、カムバック部門において目標位置へゴールするチームは極めて少なく、2016 年度以前の CanSat 競技において全て参加チームがゴールができない大会も珍しくなかった。そのためゴールができれば優勝記録と言って間違いない位置づけであった。ARLISS2016 では当時我々の愛知工科大学 STELA チームが GNSS 位置情報での接近制御にて 5.81m ゴールを達成した[17]。しかしその後電気通信大学 Cake's チームは 4.8m まで接近し、東京大学中須賀研究室 Team Grenouille チームは 3.76m まで接近した。GNSS 位置情報は数 m の誤差が存在し、3 チームは同じ GNSS 位置情報でのゴール判定であるが数 m の誤差での接近距離の差が出た結果となった[18]。記録として順位は東京大学中須賀研究室 Team Grenouille 1 位、電気通信大学 Cake's 2 位、愛知工科大学 STELA 3 位の結果となった。順位表

を Table 2-7-1 に示す.

GNSS 誤差について, 我々は GNSS の測位誤差が CanSat に搭載している無線装置 XBee が発する電波により大きくなることを実験により明らかにし [19], GNSS のみのゴール判定では数 m の誤差が生じてしまうことを確認した. [20]その後カムバック部門の競技において GNSS 位置情報誤差を超えた 0m 地点でのゴールを目指す必要がある.

Table 2-7-1 ARLISS2016 Result

順位	記録距離	チーム名
1 位	3.76m	東京大学中須賀研究室 Team Grenouille
2 位	4.8m	電気通信大学 Cake's
3 位	5.81m	愛知工科大学 STELA

2-7-2 研究目的

従来 CanSat は目的地へ走行する性能を備えるため、まずはロケットの衝撃やパラシュート開傘衝撃、着地衝撃における耐久性のある機構の設計の確立を目指し、着地後は小型ながら轍を走破する走行性能や長時間の走行に機体の耐久性を備えた設計の確立を行い、ARLISS2016 において 3 チームがゴールできる技術成熟が成された。これからより惑星探査ローバとして目的地の到達の精度を高めるためには、GNSS 位置情報の誤差を超え、実際の宇宙開発でも多く利用される画像認識手法による目的地の物体の認識及び誘導制御手法の確立や、より惑星探査機能を備えた CanSat の設計を行う必要がある。

しかし CanSat のような小型精密機器において従来の耐久性を維持しながら、エッジデバイスである小型マイコンにより画像認識などの機能を搭載するには、どの手法であれば限られたリソース内で動作可能であるか、計算時間はローバの制御を行う上で十分短いか、目的地にある物体を認識して誘導する制御が可能か、どの程度の精度の認識率で認識が可能であるか調査を行う必要がある。これらの目的において調査を行い、実際に CanSat 競技において実証実験を行った結果から、最適な手法の研究を行った。

【参考文献】

- [17] 斎藤卓也,秋山実穂,前田一朗,宮里一慶,山川啓介,田宮直,"小型自律移動ロボットの設計と ARLISS による実証評価," 日本機械学会ロボティクス・メカトロニクス講演会 in 福島 2017, 2P2-A12, May 2017.
- [18] 斎藤卓也, 秋山実穂, “惑星探査ローバの ARLISS2016 における動作ログの詳細分析,” 愛知工科大学紀要第 15 巻, pp.19-25, March 2018.
- [19] 斎藤卓也,秋山実穂,"惑星探査ローバにおける XBee の GNSS 測位精度影響の研究,"情報処理学会組込シンポジウム ESS2017, pp.114-225, Aug. 2017.
- [20] Miho Akiyama, Takuya Saito, "Influence of Radio Waves Generated by XBee Module on GNSS Positioning Performance," IEEE International Conference on Consumer Electronics – Taiwan (IEEE 2020 ICCE-TW), Taoyuan, Taiwan Sep. 2020.

Supporting Online Materials :

- [i] <https://et.kyushu-u.ac.jp/index.php/report/essay02/> [Accessed Feb 20, 2023]
- [j] <https://ssl.tksc.jaxa.jp/isas-rev/koubo-support-pages/kogata2019/document/BDB-08012D.pdf> [Accessed Feb 20, 2023]

第 3 章 再現実験による安全審査と設計評価

3-1 安全審査と設計計画書

宇宙開発では宇宙空間で運用する衛星や探査機は故障などの不具合が生じればミッションは達成できなくなり、修理も一切できない環境であるため、宇宙空間での未知な環境でも何が起こっても目標の動作が行うことができる性能が求められる。そのためロケットで打ち上げを行う前に地球上で宇宙環境を模擬した再現実験を行い、何度動作を行っても確実に運用できることを証明する安全審査を設けることにより確実にミッションが成功するようにしている。安全審査には厳しい規定が多く存在し、例えば 1 つ 1 つの部品の動作を約 100 回動作させ不具合が生じないことを証明することを始め、安全審査に合格しない場合にはロケットの搭載を許可されない。

ARLISS 及び能代宇宙イベントには CanSat を投下する際に危険が無いかを証明する安全審査書の提出が義務付けられる。安全審査書を期限内提出し合格を貰わない限り大会競技に参加することは原則できない。サイズ規定を始め、ロケット打ち上げ衝撃荷重に機体が耐えられるか、落下が減速機構を用いて行えるかを主に審査を行う。

種子島ロケットコンテストでは安全審査書の提出は求められていないが、設計計画書の提出が義務付けられている。設計計画書はミッション内容や機体やシステムの詳細設計を p.6 以内で記述し、その他試験の実施は要求されない。種子島ロケットコンテストでは参加チーム制限により 50 チーム以内までの競技が行われるが、応募が 50 チームを超える場合は設計計画書から種子島ロケットコンテスト実行委員会の宇宙工学専門の大学教員による審査により参加チームが選定される。

本章では ARLISS における安全審査を元に CanSat の再現実験を行った結果を述べる。また CanSat の詳細設計を示す。安全審査は 2021 年度の実験となっており、本論文の第 7 章における CanSat モデルによる実験内容となっている。

3-2 ミッションステートメントとサクセスクライテリア

3-2-1 ミッションステートメント

CanSat において何を実現したいかを明確にするため、ミッション定義をする必要がある。そのためにミッションの目標を背景・根拠とともに明確に述べたミッションステートメントやミッションステートメントに対して1つ及び複数のミッション項目・システム構成が存在しそれらの成功段階を詳細かつ定量的に述べたサクセスクライテリアを記載する必要がある。これらは実際の宇宙開発でも取り入れられているミッション定義方法である。[j]

3-2-2 ミッションステートメントの例

2021年度 ACTS(ARLISS 代替大会)での SuperNOVA チームの実際のミッションステートメントを Fig. 3-2-1 に示す。[k]

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

Fig. 3-2-1 Mission Statement

ミッションステートメントは初頭文書に簡潔に記載する。その後ミッションの意義と目的について詳細に記載する。SuperNOVA の場合にはミッション背景を始めに記載した。その後ミッションの詳細を記載した。

ミッション背景について、CanSat の comeback 部門において、0m 距離まで誘導する手法では目的地に設置されたパイロンを画像認識する手法が多く利用される。CanSat に搭載されるマイクロコンピュータでも処理可能であり近距離時の形状変化に対応できる手法としてパイロンの色を認識する画像認識で 0m ゴール誘導する手法がある。色を用いた画像認識では直射日光の当たる部分や影の部分、時間経過による天候の変化による明るさの変化で色が大きく変化することで、認識が困難になる問題があった。

日照条件による色変化に対応するため、様々な明るさの画像をビックデータとして学習する Deep Learning の Image Classification を用いた 0m ゴールの手法では、色変化に対応でき色を用いた画像認識よりも認識率が向上することがわかった。

また Image Classification による手法では、撮影した画像の分類はできても画像中どの位置にあるかは特定ができない。そこで、我々は画像全体を ROI で横方向に 3 分割、5 分割、それぞれ Image Classification させることにより、左右どこにあるかを判定して 0m ゴールの制御をしていた。3 分割でゴールを発見している例を Fig. 3-2-2 に示す。



Fig. 3-2-2 Three-part image classification to determine left/right

しかし 5 分割まででは約 10m 前後の長距離の認識率が低く、CanSat が強風で流された際にゴールの認識ができないことがわかった。画像分類を用いたゴール認識にて、ROI の分割数を増加させる事により、10m の長距離から認識に成功した。しかし画像分類における ROI を増加させることによる長距離認識では ROI の分割数に応じて認識回数が増加するため、認識時間がかかり、ACTS2020 での競技において実験時間 15 分ギリギリでゴールへの誘導となった。そのため、今後はより制御の速度を上げられる手法を検討する必要がある。

Object Detection による 0m ゴールについて、Object Detection では 1 枚の画像から物体を複数検知でき、画像中どこにあるかがわかる。1 枚の画像でゴールの他にも人や気球なども同時に検知でき、それぞれ物体がどの位置にあるのかも座標情報で取得できる。ゴール認識だけでなく、複数の他の物体を検知できるため、ゴールと同じ赤色の服を着た人や赤い気球が同じ画像内に写ってもゴールと識別することができ、認識精度の向上ができる。Image

Classification よりも 1 回の処理時間は長くなるが、従来はゴールを探すのに ROI を 3 分割、5 分割や、さらに上下移動などを行っていたが、その制御を行う必要がなくなるため、特に CanSat のチルト方向の傾きに強くなると考えられる。問題は、Object Detection は従来の Image Classification よりも処理が複雑になり、計算量やメモリ使用量が増大するため、Raspberry Pi Zero で問題なく実装可能かである。現在我々がテストしている環境では、Raspberry Pi Zero でも Mobile Net SSD v2 モデルでは数秒間で認識可能であり、さらに Coral 等のアクセラレータを用いると、1 秒以下で認識可能であるため、実装可能であることを確認した。

3-2-2 サクセスクライテリア

ミッションステートメントに対して一つ及び複数のミッション項目・システム構成が存在し、それらの成功段階を詳細かつ定量的に述べたものがサクセスクライテリアとなる。サクセスのレベルはミニマムサクセス、フルサクセス、エクストラサクセスに分類される。ミニマムサクセスは挑戦的なミッション目標を設定するため、独立した複数ミッション機器を搭載する場合等では、その一部の機能喪失が生じた場合でもクリアできる最低限の目標として設定することができるためのサクセスレベルを設定する。フルサクセスは予定していた要求を満たし、計画通りの成果を得るため何を達成したいか明確にサクセス定義を設定する。エクストラサクセスはフルサクセスを達成した上で、さらにそれを上回る成果を得ることであり、そのサクセスレベルを設定する。ただし単に目標を設定するだけでなく、上回る成果を得るために克服すべき課題とその対策を明らかにする必要がある。フルサクセスの設定は必須であるが、ミニマムサクセスとエクストラサクセスは任意となっている。

3-2-3 サクセスクライテリアの例

2021 年度 ACTS(ARLISS 代替大会)での SuperNOVA チームの実際のサクセスクライテリアを Table 3-2-1 に示す。サクセスクライテリアがミニマムサクセスとフルサクセスで設定した。

Table 3-2-1 Success Criteria

ミニマムサクセス	CanSat が放出落下後に破損なく着地し、パラシュートから分離し、GNSS 位置情報からモータ回転数制御により 10m 以上走行する。
フルサクセス	ゴールから 5m 以内まで接近し、その後 Object Detection でゴールへ 0m まで近づきゴール判定をする。

3-3 ミッションシーケンス

3-3-1 ミッションシーケンス

ミッションシーケンスではキャリア収納からミッション達成までの流れを図と文章で簡潔に示して説明する。図式にすることでミッションの流れをわかりやすく表記する。

3-3-2 ミッションシーケンスの例

2021 年度 ACTS(ARLISS 代替大会)での SuperNOVA チームの実際のミッションシーケンスを 1-11 の番号の順序にした。またミッションシーケンス図を Fig. 3-3-2 に示す。

1. CanSat を気球のキャリアに収納する。
2. 気球のキャリアから CanSat 本体を放出する。
3. CanSat のパラシュートを開き、落下速度を減速させる。
4. 機体が破損無しで地上に着陸する。
5. CanSat 保護ケースを本体から切り離し、パラシュートと分離する。
6. GNSS 位置情報をもとに、モータ回転数制御でゴールに向かって走行する。
7. 轍にスタックしたと判定した時、轍脱出動作を行い、轍からの脱出を行う。
8. GNSS 情報によりゴール 10m 以内になったら動作速度を落とし、ゴールを目指して走行する。
9. 5m 以内になったら、カメラ画像を Object Detection により、ゴールを物体として検知する。(ミッション)
10. カメラ画像によりゴールを発見したら、ゴールに向かって走行する。
11. カメラ画像からゴールに十分近づいたらゴール判定を行い、停止する。

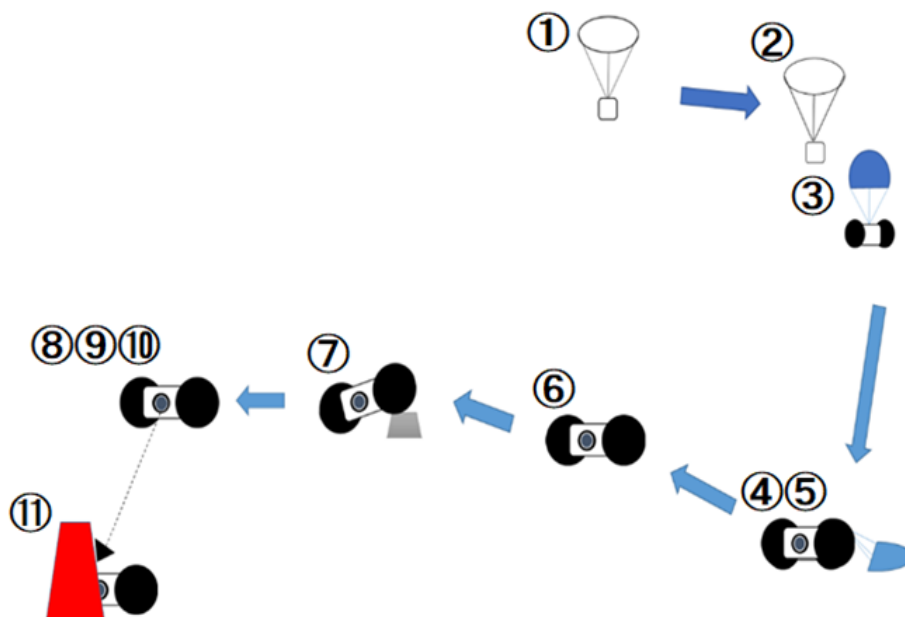


Fig. 3-3-2 Mission Sequences

3-4 システム要求項目と試験項目設定

3-4-1 システム要求項目

システム開発を行う上で実施すべき業務内容を整理し文書化するためのシステム要求定義を行うために、システム要求項目を設定し、開発過程において何を行うべきか整理を行う。CanSat の場合にはレギュレーションを満たした CanSat ができているかを中心に項目を設定し、達成したいミッションのための項目はミッション要求項目として追記で設定する。

3-4-2 システム要求項目の例

2021 年度 ACTS での SuperNOVA の実際のシステム要求項目を Table 3-4-1 に示す。またミッション要求項目を Table 3-4-2 に示す。システム要求はレギュレーションを満たした CanSat か、予測される衝撃を加えても故障がないかなど試験を行い、ミッション要求はミッションの動作を確実にこなせるか、またシステム要求以外での予測される衝撃耐久が可能か試験を行う。

Table 3-4-1 System Requirement Items

要求番号	自己審査項目
	ACTS <u>安全基準</u>
R1	<u>質量と容積</u> がレギュレーションを満たすことが確認できている
R2	ロスト対策を実施しており，有効性が試験で確認できている（例：地上局にダウンリンクする場合，ACTS で十分な通信距離が実現できるだろうと推測できる根拠が明確に示されていること.）
R3	地表近くで危険な速度で落下させないための <u>減速機構</u> を有し，その性能が試験で確認できている
R4	打ち上げ時の <u>準静的荷重</u> によって，安全基準を充足するための機能が損なわれないことが試験で確認できている
R5	打ち上げ時の <u>振動荷重</u> によって，安全基準を充足するための機能が損なわれないことが試験で確認できている
R6	分離時の <u>衝撃荷重</u> によって，安全基準を充足するための機能が損なわれないことが試験で確認できている
R7	打ち上げ時の <u>無線機の電源 OFF</u> の規定を遵守できることが確認できている（FCC 認証かつ 100mW 以下の機器は OFF しなくて良い。また，スマートフォンを用いる場合は FCC 認証かつソフトウェアまたはハードウェアスイッチで off にできること（2017 年追加））
R8	無線のチャンネル調整に応じる意思があり，また実際に調整ができることを確認できている
R9	R1-R8 の充足を確認した設計の機体によって，ロケットへの装填から打ち上げ後の回収までを模擬した End-to-end 試験を実施できている，今後，安全性に関わる大幅な設計変更はない
R10	CanSat の収納・投下準備が 5 分以内でできている
	<u>カムバックコンペティションルール</u> の充足
R11	ミッション時に人間が介入しない <u>自律制御</u> を実施することが確認できている（注：2014 年のレギュレーション改定以降，地上局設備に計算機能を持たせてアップリンクしても良い）
R12	ミッション後，規定された <u>制御履歴レポート</u> を <u>運営者へ提出</u> する準備ができている（以下の根拠の項に制御履歴レポートの例を添付すること。ダミーデータを使用しても良い）

Table 3-4-2 Mission Requirement Items

要求 番号	自己審査項目（ミッション実現要求項目）
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	ゴール地点に設置されたコーンを検知し、10m以上GNSS制御走行した後、Object Detection画像認識で0mゴールすることができることを試験で確認できている。

3-4-3 試験項目設定

システム要求項目及びミッション要求項目からその項目を満たす試験を設定する。システム要求項目及びミッション要求項目の検証を行う検証項目名を記載し、対応する要求番号を記載する。また実施予定日や実施日を記載する。

3-4-4 試験項目設定の例

2021 年度 ACTS(ARLISS 代替大会)での SuperNOVA チームの実際の試験項目設定を Table 3-4-4 に示す。対応する自己審査項目の要求番号に合わせ、検証試験項目を設定した。

Table 3-4-4 Test Item Setting

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

3-5 設計詳細

3-5-1 CanSat 機構

システム仕様として CanSat の機構について詳細説明を行う。CanSat の全体の外観を Fig. 3-5-1 に示す。機体は 1 枚の 3D プリントで作成した板に左右モータと上下バッテリーを取り付け、モータの裏側は電子回路を搭載している。

3D プリントでの制作で利用した TPU 樹脂は 3D プリントで利用できる樹脂の中でもゴムのような柔らかさを持つ性質がある。また 3D プリントでの設定が切削機を利用するよりも細かくカスタマイズできる素材であり、着地時に衝撃で少し曲がるような柔らかさの設計することにより、衝撃耐久し衝撃吸収を行う仕様となっている。またタイヤには柔らかい素材の天然ゴムを利用することで、より衝撃吸収ができる使用となっている。CanSat を上から見た図を Fig. 3-5-2、CanSat を下から見た図を Fig. 3-5-3 に示す。

減速機構は機体にパラシュートを付属した保護ケースを取り付ける。結束バンドを電熱線で焼き切ることによって展開する。パラシュート生地は頑丈なリップストップナイロンを利用し、紐には金属ワイヤーと金属メジャーを利用することで着地時に CanSat とパラシュートを遠ざけ絡まり防止ができる構造となっている。CanSat のパラシュート離脱機構と紐を直接繋げずに、保護ケースに紐を取り付けることにより、分離衝撃で離脱機構が故障することを防いだ。パラシュートの保護ケース収納時写真を Fig. 3-5-4 に示す。また、実際にパラシュートを展開した図を Fig. 3-5-5 に示す。

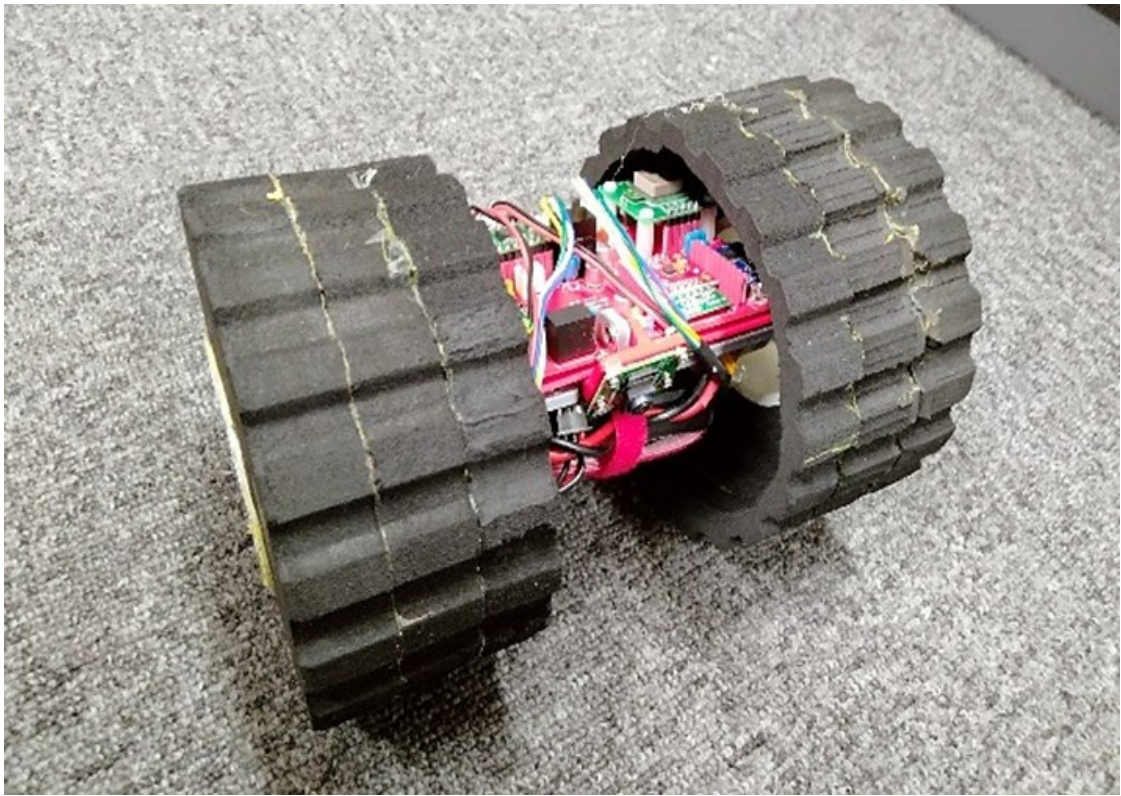


Fig. 3-5-1 CanSat

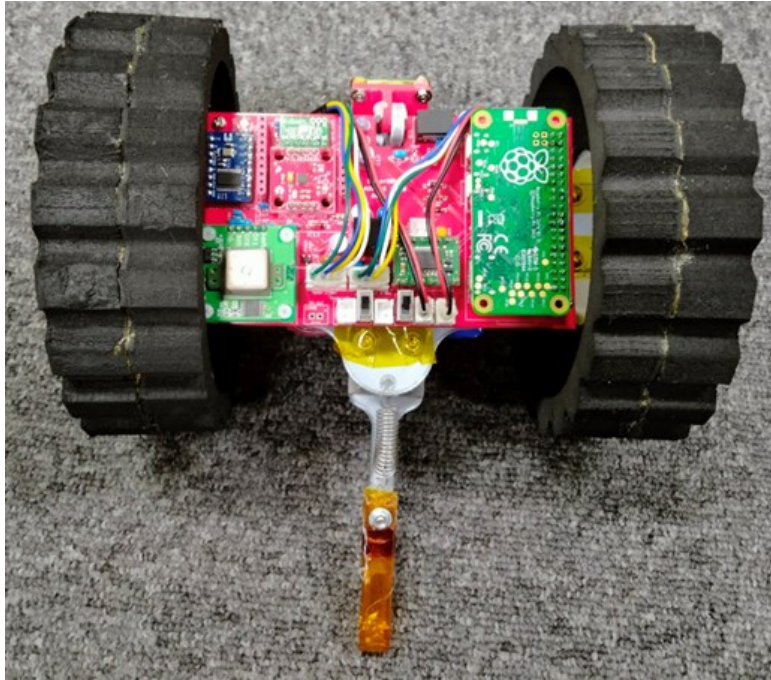


Fig. 3-5-2 Top view of CanSat

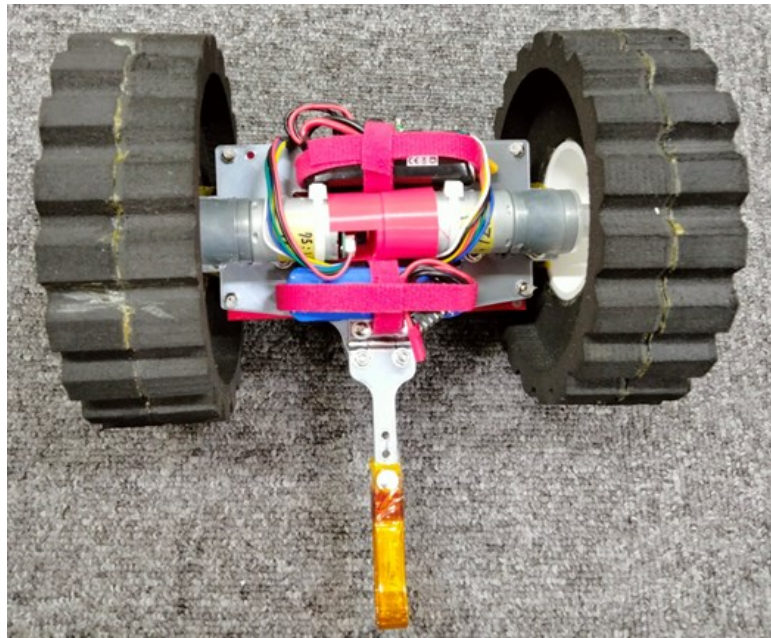


Fig. 3-5-3 View of CanSat from the bottom

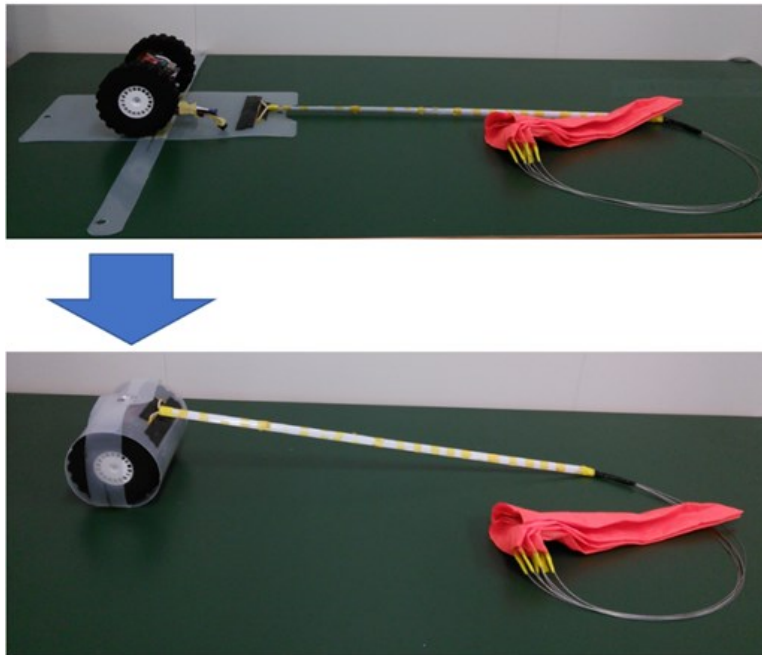


Fig. 3-5-4 CanSat in protective case

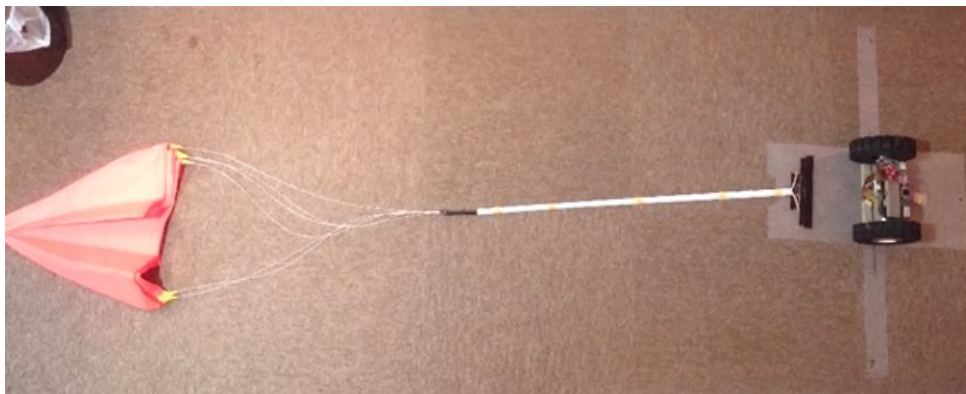


Fig. 3-5-5 Parachute

3-5-2 システム図とアルゴリズム

CanSat 電子回路における各電子部品の仕様説明やシステム図を記載する。また電子部品などの搭載計器の仕様一覧を名称や型番、購入先を詳細に記述する。

マイコン、モータについて、Raspberry Pi zero W をマイコンに利用し、Coral Accelerator を Deep Learning の計算に利用し、Michibiki Compatible GNSS で位置情報を取得し現在位置と目標位置の角度を計算し、TB6612 の Dual Motor Driver に目標位置への旋回 PWM 制御値を出力し、モータに出力する。

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

長距離通信無線について、また、気圧センサが故障時には Photo IC Diode で放出されたかどうか明るさを MCP3208 A/D Converter で A/D 変換し測定しているまた、長距離へ落下した際に IM920 wireless Communication を用いて位置情報を送信し、現在位置特定に利用している。

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

Fig. 3-5-6 にシステム図を示す。また、実際に使用している電子部品の一覧を Table 3-5-1 に示す。

Table 3-5-1 Components

分類	名称・型番	入手先・参考情報等	URL
GNSS	AE-GYSFDMAXB	秋月電子	[l]
マイコン	Raspberry Pi zero W	SWITCHSCIENCE	[m]
アクセラレータ	Coral USB Accelerator	SWITCHSCIENCE	[n]
200G 加速度センサ	H3LIS331DL	SWITCHSCIENCE	[o]
16G 加速度センサ	LSM9DS1	SWITCHSCIENCE	[p]
I2C シリアル変換	SC16IS750	SWITCHSCIENCE	[q]
A/D コンバータ	MCP3208	秋月電子	[r]
モータドライバ	TB6612	秋月電子	[s]
5V3 端子レギュレータ	V78-1000	秋月電子	[t]
3.3V3 端子レギュレータ	48033	秋月電子	[u]
通信装置	IM920	秋月電子	[v]
気圧センサ	AE-BME280	秋月電子	[w]
カメラ	Raspicam V2	SWITCHSCIENCE	[x]
モータ	POLOLU ギアードモータ 75:1	POLOLU	[y]
バッテリー（モータ用）	Hyperion 550mAh 3S 11.1V	Hyperion	[z]
バッテリー（Computer 用）	Hyperion 550mAh 2S 7.4V	Hyperion	[aa]

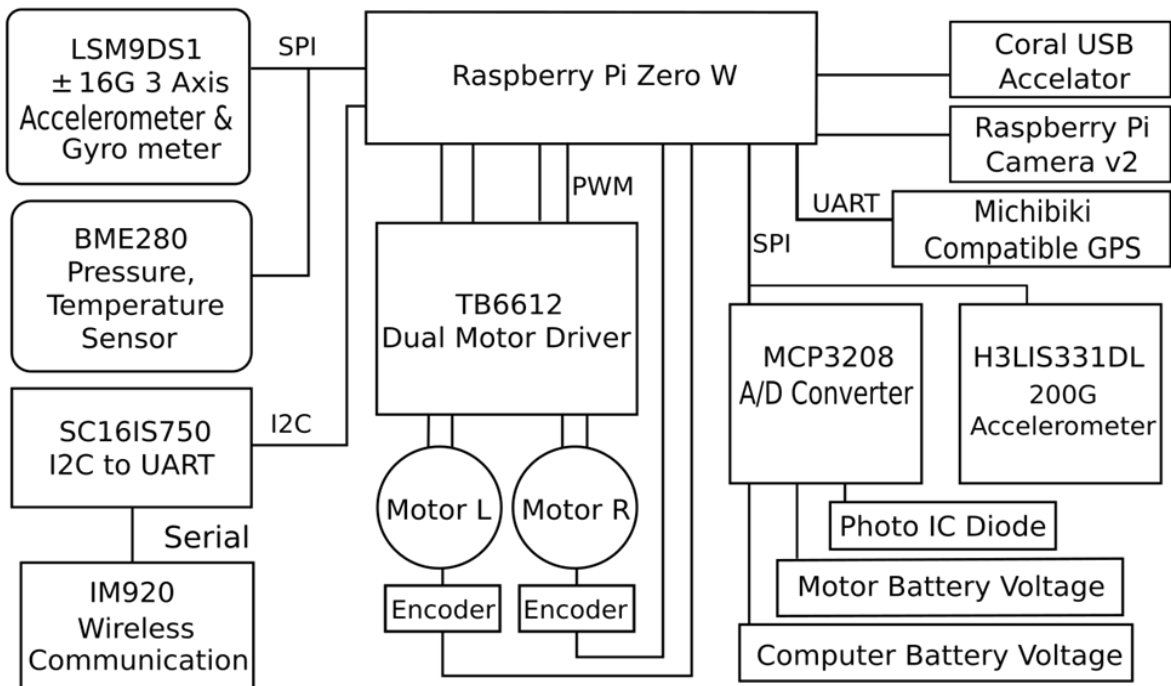


Fig. 3-5-6 System diagram

また、アルゴリズムについて詳細に記載し、フローチャートを図式し説明を行う。

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

走行制御～ゴールについて、GNSS 位置情報から走行制御を開始し、パラシュートが走行方向にあれば旋回制御を行う。スタックした場合はスタック処理のバック動作や回転動作から脱出動作を行う。ゴール 10m 以内まで近づくと減速しゴールに近づきやすい走行を行う。ゴール 4m 以内まで近づくと Object Detection(Deep Learning)でゴールを発見するまで旋回走行する。ゴールを発見するとゴールの方向へ走行し、ゴール 0m 地点と Deep Learning で判定すると停止する。フローチャートを Fig. 3-5-7 に示す。

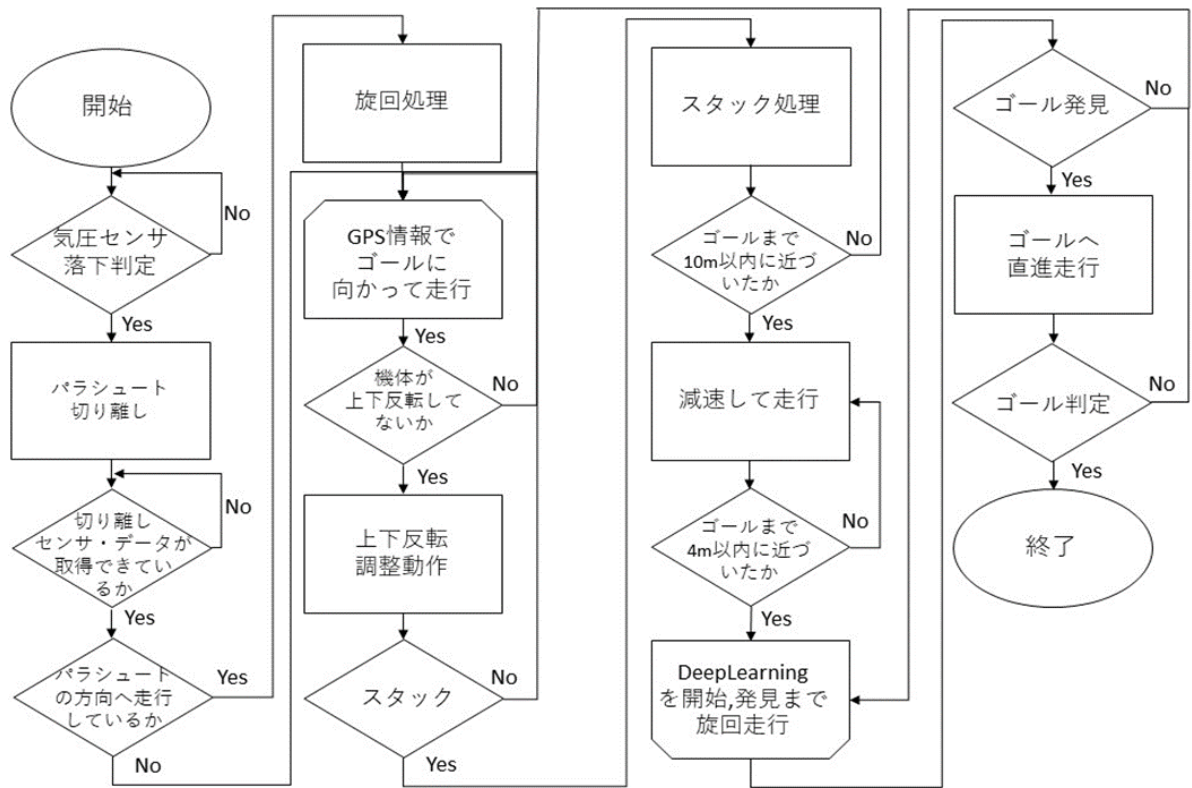


Fig. 3-5-7 Flow Chart

3-6 質量試験

3-6-1 質量試験内容

質量試験はレギュレーションの質量以内に CanSat が設計されているか確認を行い，質量測定時の証拠動画や画像からその質量の証明を行う．レギュレーションの質量は 1,050g 以内であり，その質量内であることを証明するため質量計にパラシュートを含む CanSat を載せ，計量しその結果の数値を撮影することで質量内であることを証明する．質量の計測は競技の前にも行われ，その際も質量以内でないと競技を行うことができない．

3-6-2 質量試験結果

質量計に載せた CanSat を Fig. 3-6-1 に示す．CanSat は質量計の数値では 832g を示しており，規定の 1,050g 以内であることを確認した．



Fig. 3-6-1 Mass measurement result

3-7 機体収納・放出試験

3-7-1 機体収納・放出試験内容

機体収納・放出試験では規定の内径 146mm、高さ 240mm の収納筒であるキャリアに、CanSat が収納することができ、その後自重で落下し放出ができることを、実際の規定内の筒に収納し放出ができている様子を撮影し動画提出を行うことで、規定内寸法のキャリアでの競技が問題なくできることを証明する。

実験ではキャリアには実際に ARLISS でロケットの打ち上げを行う AeroPac から購入した収納筒を使用した。キャリアの寸法奥行きを Fig. 3-7-1、外径を Fig. 3-7-2 に示す。キャリアが規定の奥行き 240mm 外径 146mm であることが確認できる。Fig. 3-7-1 では目盛りは奥行き 245mm を指しているが、定規の余白が 5mm あるため、キャリアの奥行きは 240mm である。

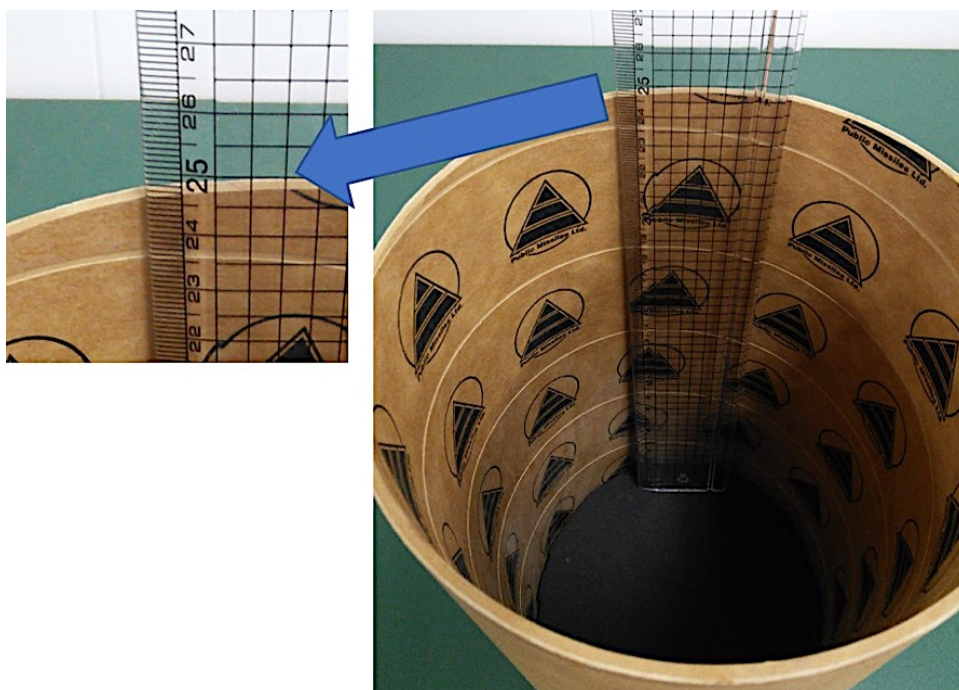


Fig. 3-7-1 Depth dimensions of carrier case

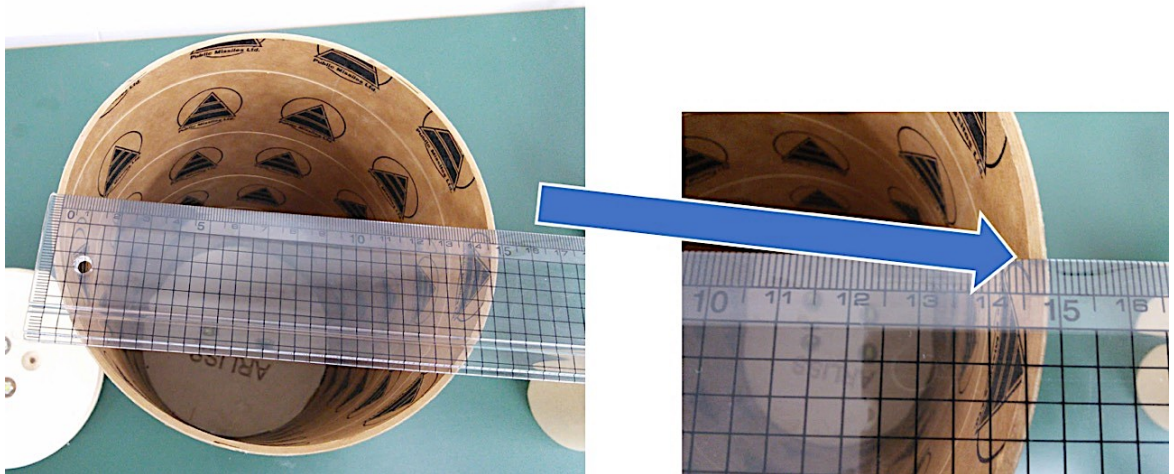


Fig. 3-7-2 Outer diameter dimensions of carrier case

3-7-2 機体収納・放出試験結果

規定の寸法のキャリアへ CanSat を収納できることが確認できた。実験動画にてキャリアへ CanSat を収納した図を Fig. 3-7-3 に示す。

また、収納後に自重で放出できることが確認できた。実験動画にてキャリアから自重で放出した図を Fig. 3-7-4 に示す。

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



Fig. 3-7-3 CanSat stowed in the carrier

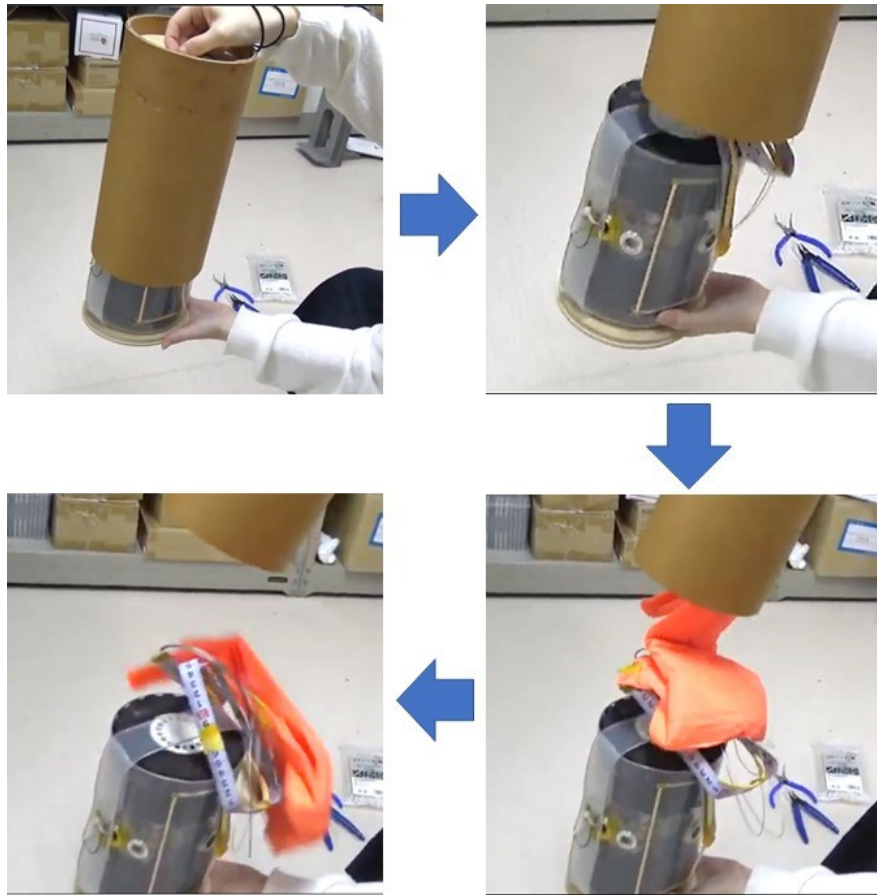


Fig. 3-7-4 Figure released under its own weight from the carrier

3-8 準静的荷重試験

3-8-1 準静的荷重試験内容

準静的荷重試験では ARLISS のロケット打ち上げ時に垂直方向にかかる静荷重に CanSat が耐えられるか確認を行う試験である。準静的荷重によりロケット内で CanSat が故障し部品がバラバラの状態で放出されれば、人に落下することを考慮すると危険であるため、ロケット準静荷重衝撃に耐えられることを証明できなければロケットに CanSat を搭載することはできない。ロケット荷重では ARLISS レギュレーションではロケット荷重のかかる推奨値として 10G の試験条件で行うことが参考値として記載されている。また実験値としては各団体がロケット打ち上げ時の実際の測定値や実

験での測定値から試験条件の荷重値で実験を行うこともできる。

準静荷重試験ではレギュレーション推奨値かつ、実際の ARLISS 打ち上げ時のデータを参考に 10G での準静的荷重で試験を行った。実際に荷重がかかっている ARLISS2019 グラフを、ARLISS 1 回目準静的荷重を Fig. 3-8-1 に、ARLISS 2 回目準静的荷重を Fig. 3-8-2 に、ARLISS 3 回目準静的荷重を Fig. 3-8-3 に示す。ARLISS2019 3 回分の打ち上げにて実際にロケットの荷重がかかっている準静的荷重の値と時間はそれぞれ、1 回目最大約 9.78G で 7s、2 回目最大約 8.93G で 8s、3 回目最大約 8.32G で 8s だった。準静的荷重の値はレギュレーション値 10G を下回っており、時間は 10s 以内であることから、実験では 10G の準静的荷重を 10s 与えた。

予めセンサや動力系が正常に動作することを確認し、その後 CanSat に搭載し、長いロープを取り付けたトートバッグに入れ、その紐を持ちトートバッグを振り回し回転運動の遠心力で大会の推奨値である 10G の加速度を与える。実際に荷重を加えている実験写真を Fig. 3-8-4 に示す。

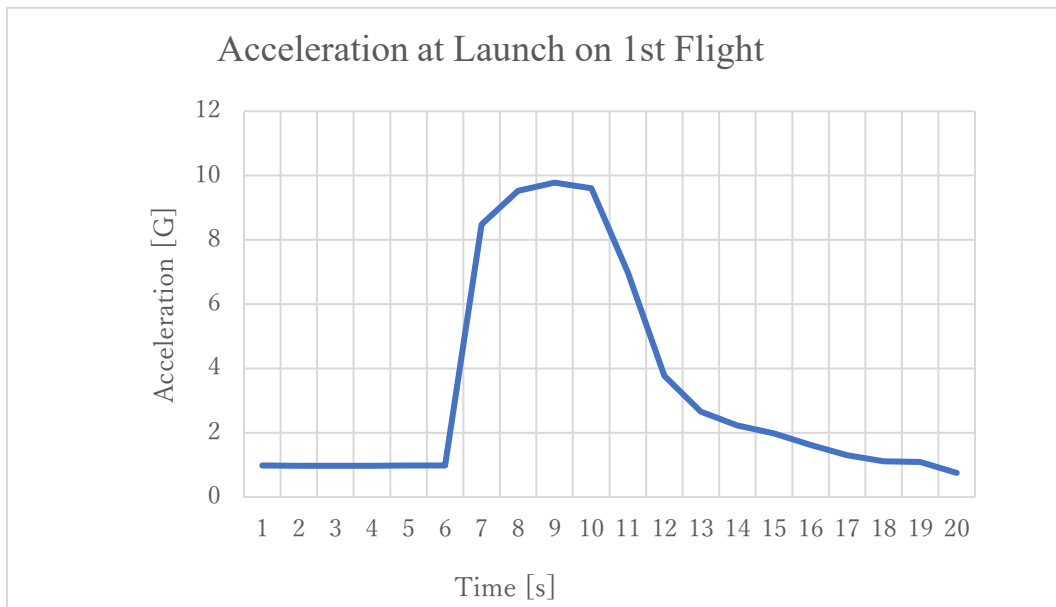


Fig. 3-8-1 1st quasi-static load

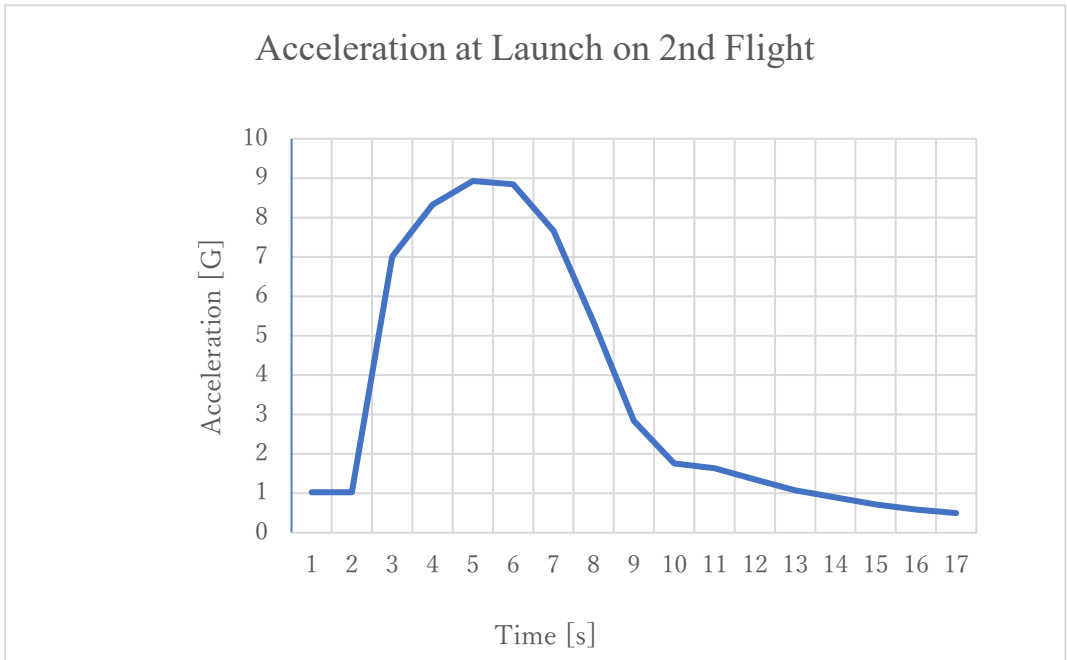


Fig. 3-8-2 2nd quasi-static load

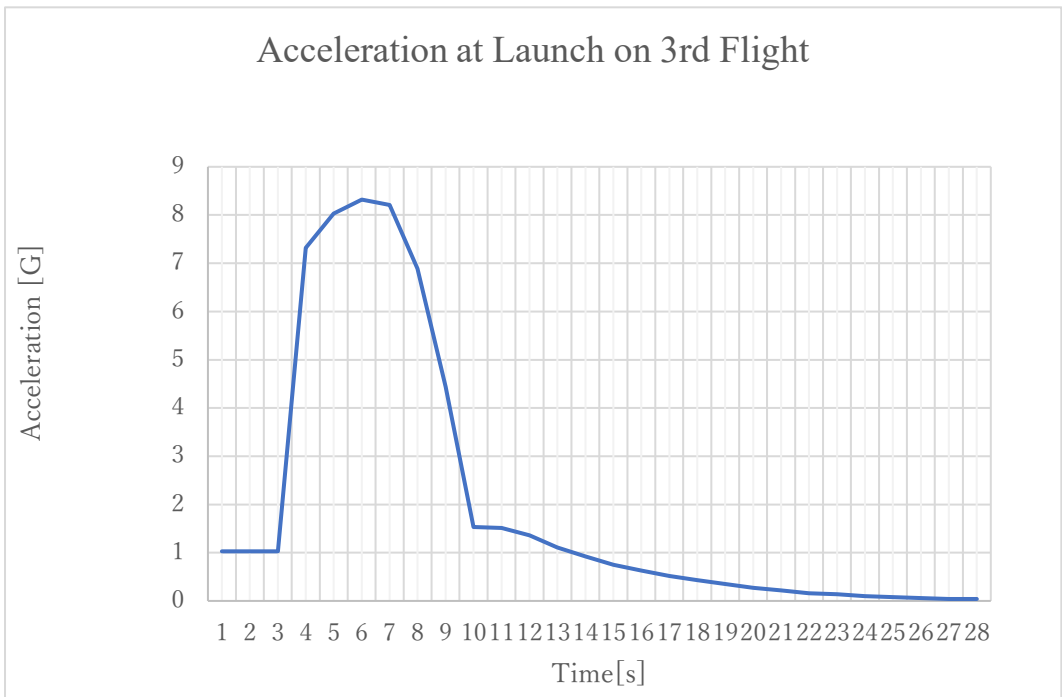


Fig. 3-8-3 3rd quasi-static load

3-8-2 準静的荷重試験結果

10G を 10 秒以上 3 回繰り返しかけた後に CanSat に搭載するセンサ(光・加速度・ジャイロ・温度・気圧・GNSS)は正常に動作し，その後問題なくパラシュート離脱動作を行い，正常に走行することを確認した．実際に 10G を 10 秒以上観測している加速度データログのグラフを Fig. 3-8-4 に示す．静荷重試験の様子は動画に撮り，異常の有無の確認を記録した結果からも，機体の異常は見られなかった． [ac]

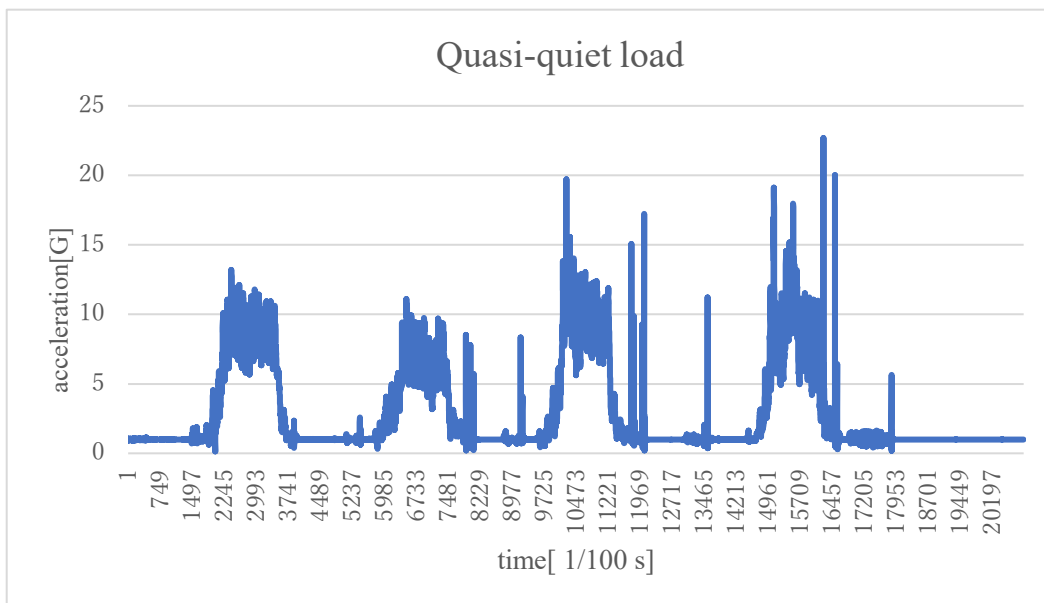


Fig. 3-8-4 Acceleration sensor value

(vertical axis: applied acceleration G, horizontal axis: time 100/1s)

3-9 振動試験

3-9-1 振動試験内容

振動試験は ARLISS のロケット打ち上げ時の振動に CanSat が耐えられるかどうか確認を行う試験である。ロケット振動によりロケット内で CanSat が故障し部品がバラバラの状態で放出されれば、人に落下することを考慮すると危険であるため、ロケット振動衝撃に耐えられることを証明できなければロケットに CanSat を搭載することはできない。ロケット振動では ARLISS レギュレーションではロケット振動のかかる推奨試験条件として正弦波振動 30-2000Hz で 15G または同等のランダム振動の試験条件で行うことが参考値として記載されている。

振動試験ではロケット打ち上げ時にかかる CanSat への振動条件として、レギュレーションで推奨されているランダム振動をフラットパターンで 30Hz~2000Hz の 15G で、ロケット打ち上げ時間約 30 秒より誤差を考慮し多く設定した 1 分間の振動を設定し、CanSat を振動させた後、電子回路の動作、パラシュート離脱機構、モータ、機体に問題がないか確認する。振動機は IMV 社 日本高度信頼性評価試験センターの振動機を利用した。実際の振動機写真を Fig. 3-9-1、振動条件設定の PC 画面を撮影した写真を Fig. 3-9-2 に示す。

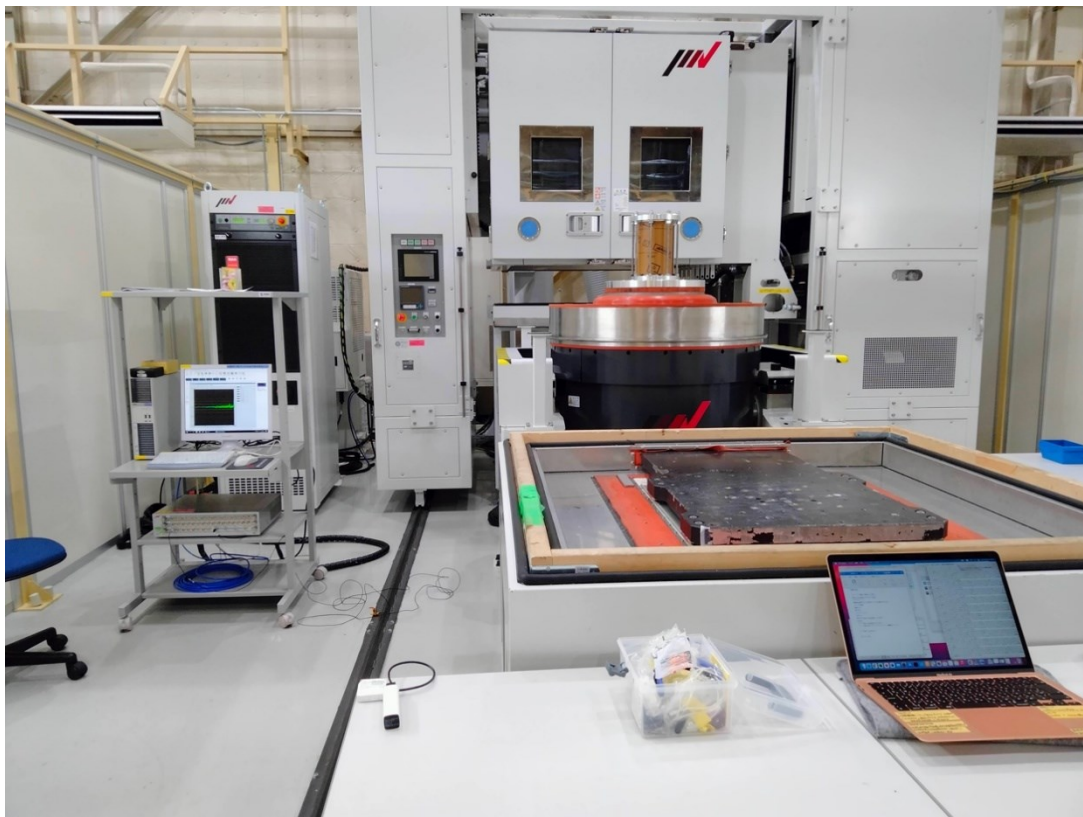


Fig. 3-9-1 Actual vibrating machine (CanSat is stored in a brown cylinder)

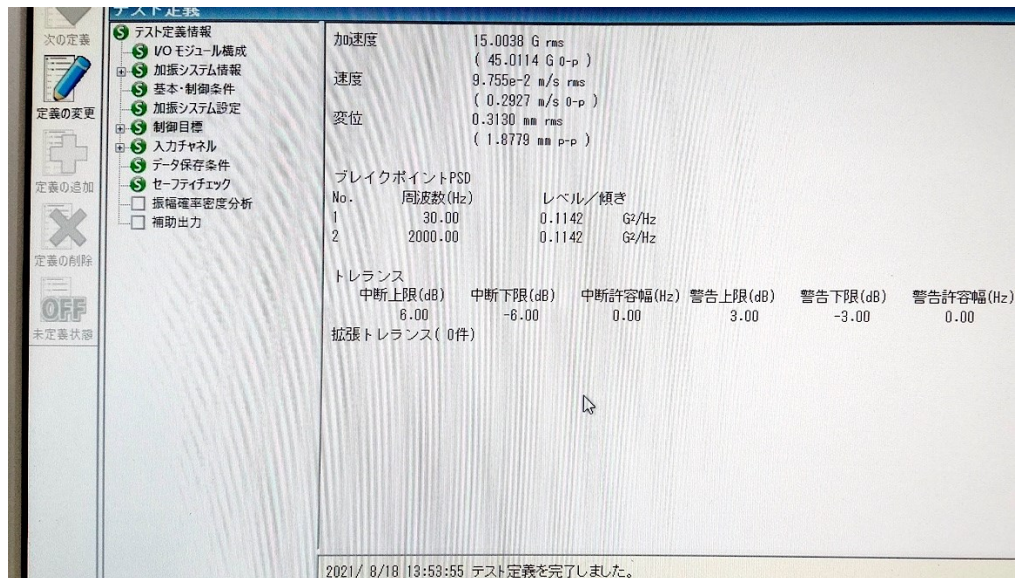


Fig. 3-9-2 Vibration Condition Setting Screen

3-9-2 振動試験結果

CanSat にランダム振動をフラットパターンで 30Hz~2000Hz の 15G の振動を与えた結果、電子回路動作、パラシュート離脱機構、モータ動作、機体の問題はなかった。また、CanSat に搭載した加速度センサの振動時データを Fig. 3-9-3 に示す。また、IMV 社提供データである実際に振動を与えた振動機からの制御及び出力データを示す。出力のレベルを Fig. 3-9-4、目標の G を Fig. 3-9-5、制御応答の G を Fig. 3-9-6、制御の m/s^2 を Fig. 3-9-7 に示す。また、実際に振動試験を行った様子を撮影した動画を下記 URL に示す。[ad]

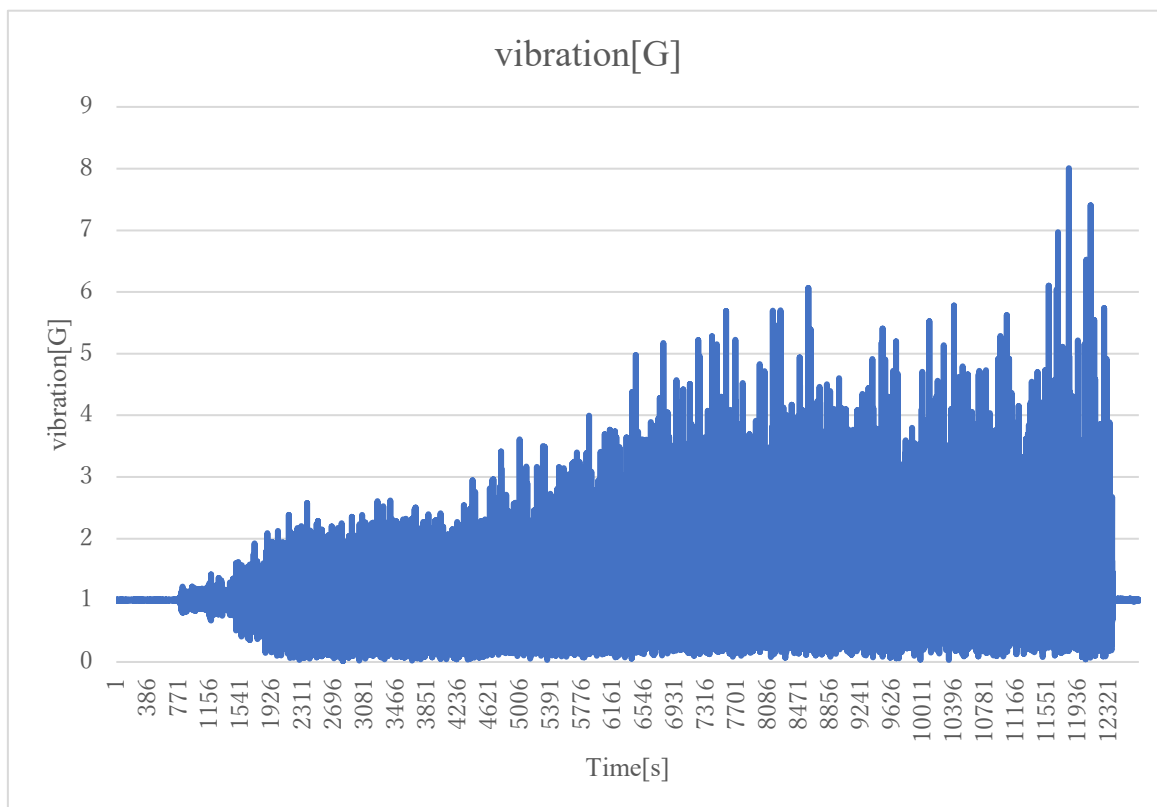


Fig. 3-9-3 Acceleration sensor data

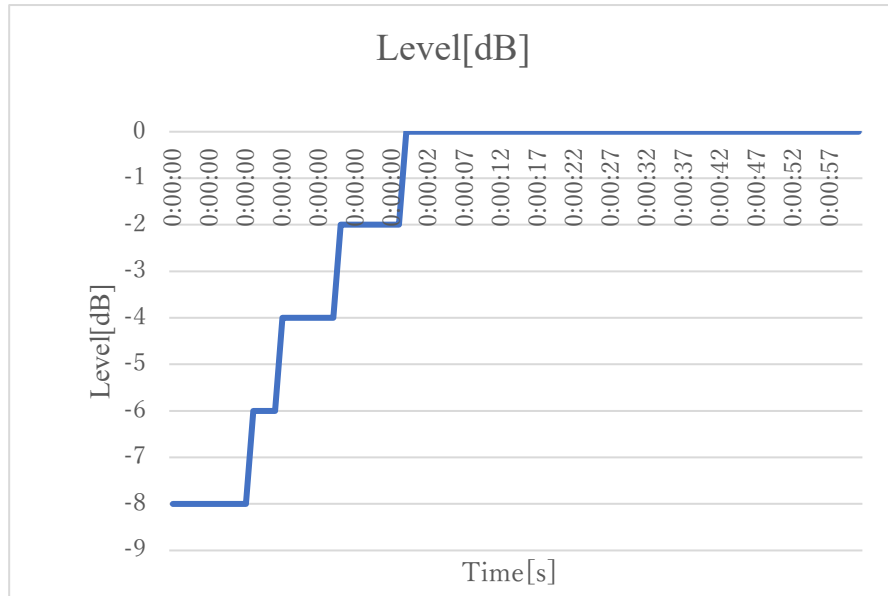


Fig. 3-9-4 Output Level

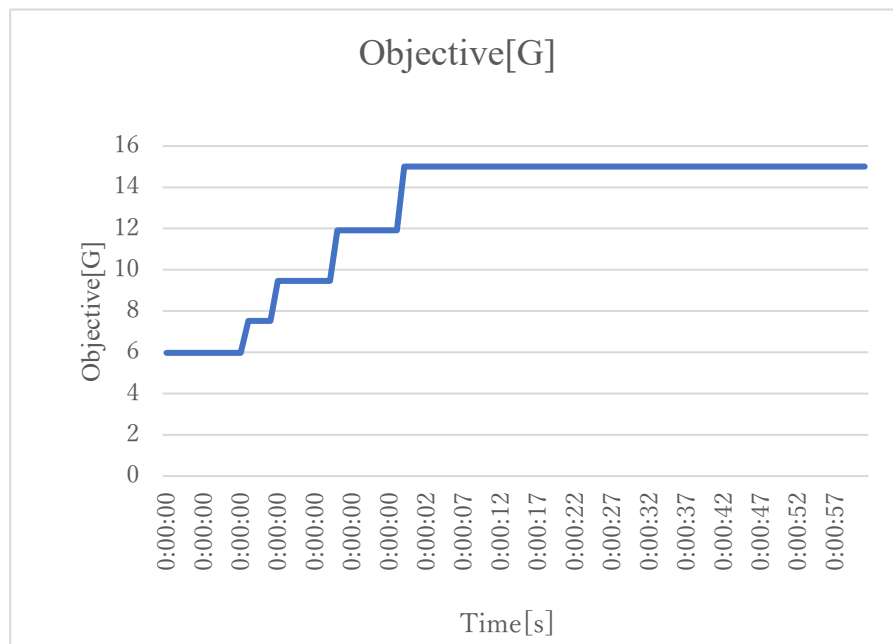


Fig. 3-9-5 Objective G

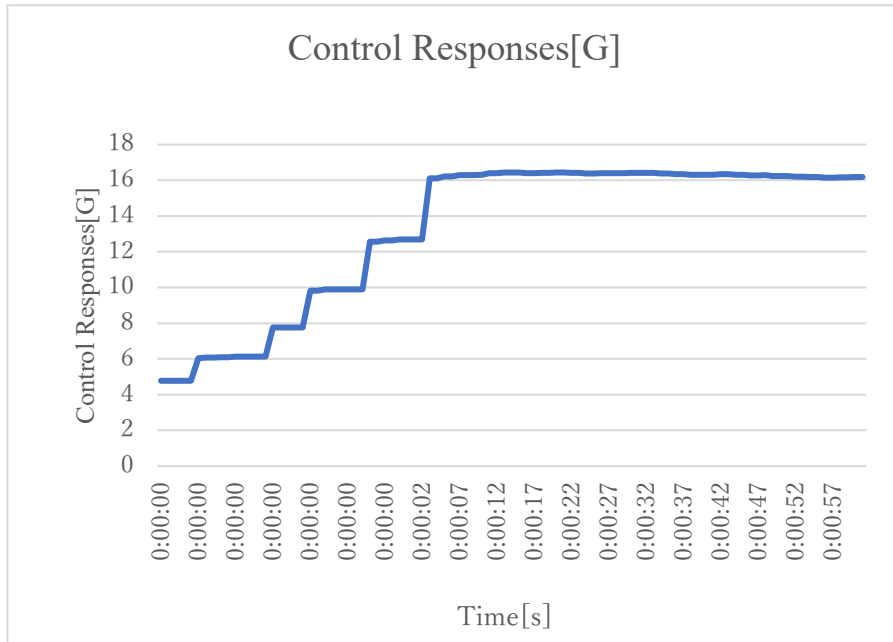


Fig.3-9-6 Control ResponsesG

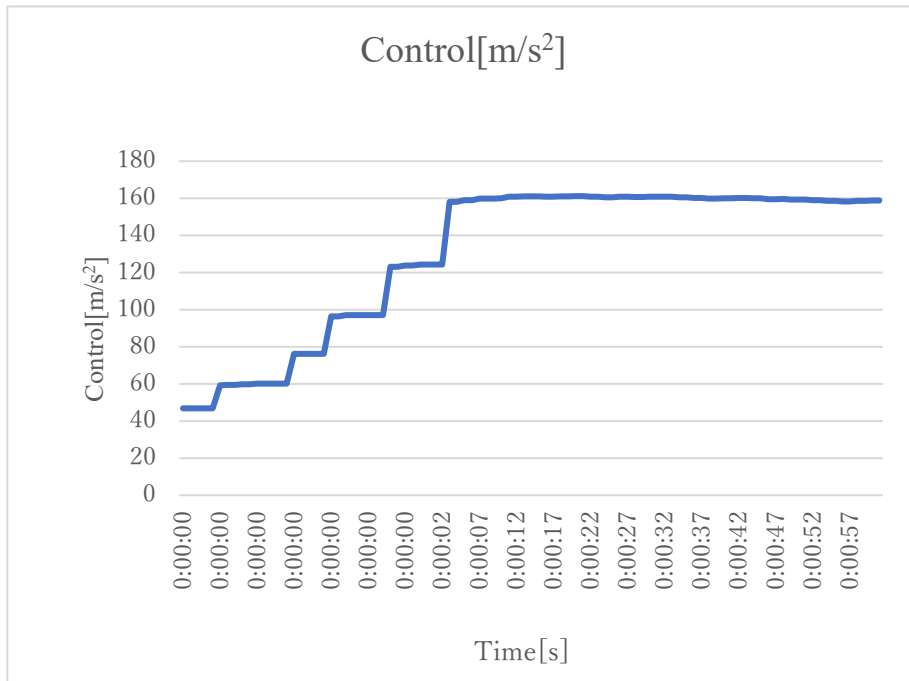


Fig. 3-9-7 Control m/s²

3-10 分離衝撃試験

3-10-1 分離衝撃試験内容

分離衝撃試験は ARLISS のロケットと CanSat が分離し減速機構のパラシュートが開傘したときの衝撃に CanSat が耐えられるかの試験である。ロケットとの分離後のパラシュート開傘衝撃で CanSat が故障し部品がバラバラの状態で放出されれば、人に落下することを考慮すると危険であるため、準静荷重に耐えられることを証明できなければロケットに CanSat を搭載することはできない。分離衝撃として ARLISS レギュレーションでは 40G の衝撃が推奨値として記載されている。また実験値としては各団体がパラシュート開傘時の実際の測定値から試験条件の荷重値で実験を行うこともできる。

分離衝撃試験では CanSat 機体がロケットから放出されパラシュートが開傘する際の瞬間的な分離衝撃の大きさを ARLISS2019 の打ち上げ 3 回分データから決定した。ARLISS2019 の放出時加速度データを、1 回目分離衝撃を Fig.3-10-1、2 回目分離衝撃を Fig. 3-10-2、3 回目分離衝撃を Fig. 3-10-3 に示す。ARLISS2019 での分離衝撃はそれぞれ、1 回目分離衝撃約 23.48 G、2 回目分離衝撃約 25.41 G、3 回目分離衝撃約 21.67 G であった。3 回の分離衝撃では 30G を下回っていることから、分離衝撃は誤差を考慮し最大 30G 相当と想定し、評価試験を実施した。

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

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

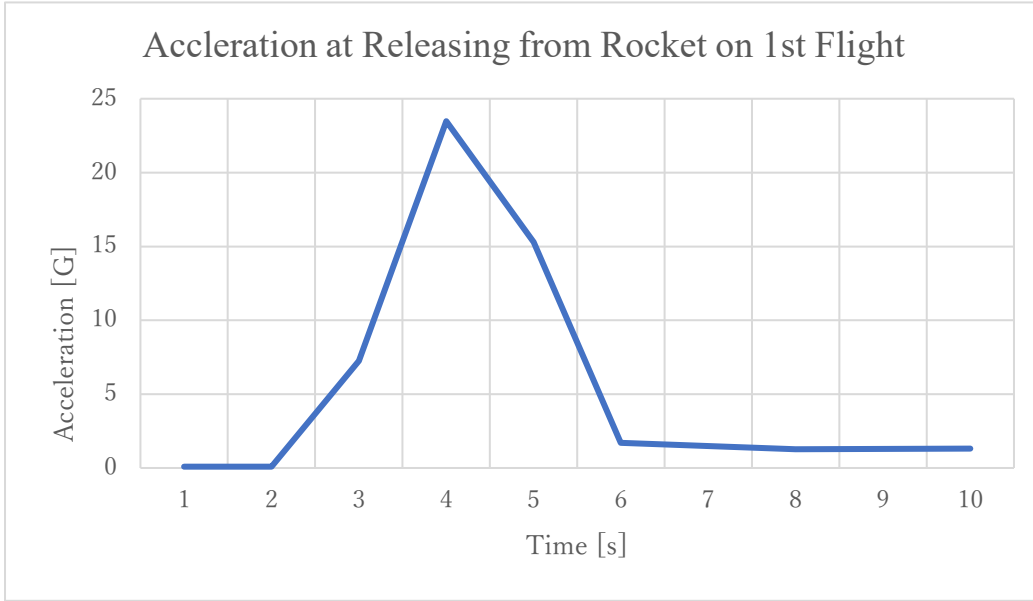


Fig. 3-10-1 Separate impact 1st

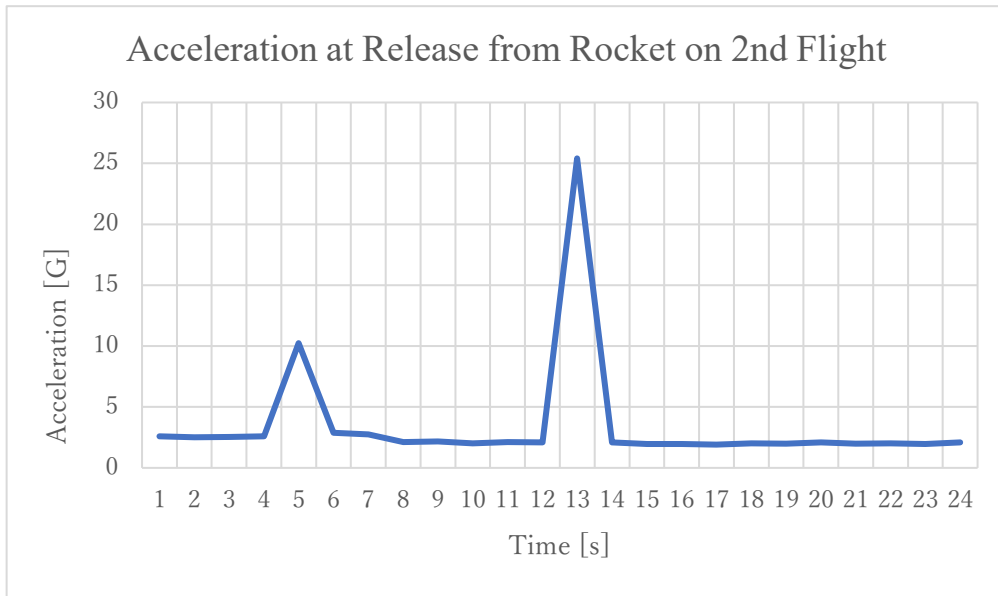


Fig. 3-10-2 Separate impact 2nd

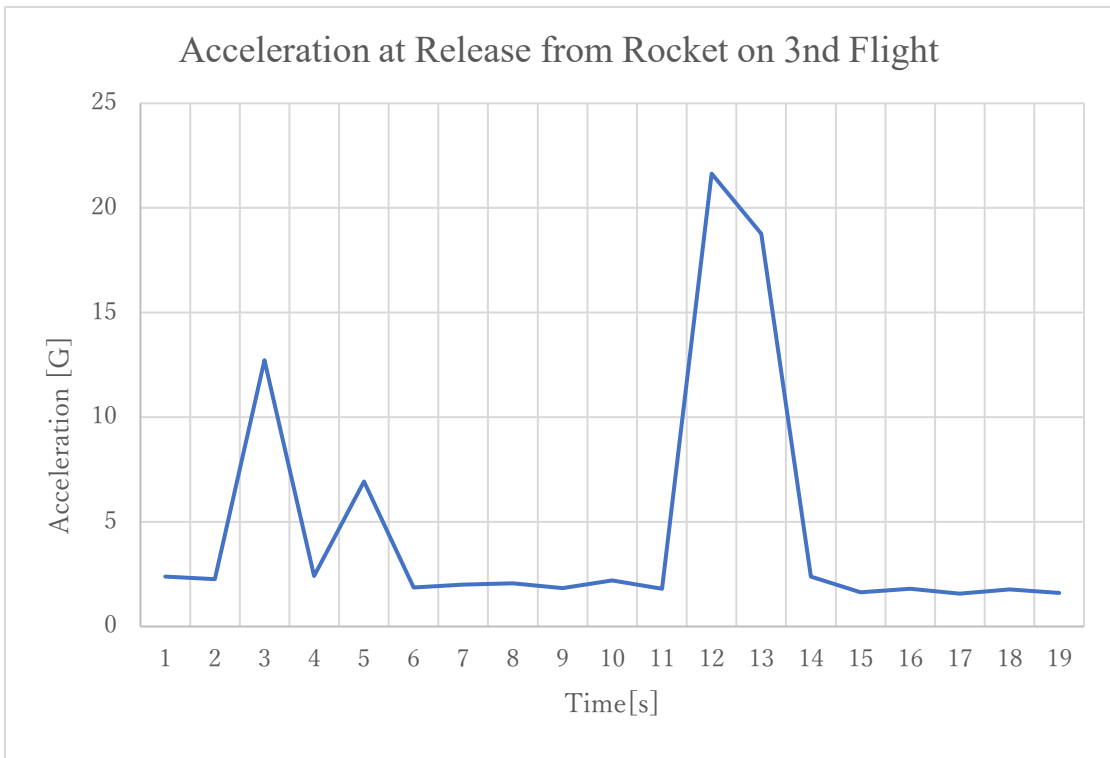


Fig. 3-10-3 Separate impact 3rd

3-10-2 分離衝撃試験結果

分離衝撃を3回与え、約30Gの加速度がかかるようにした。実際にかかった加速度のログデータを Fig. 3-10-4 に示した。30G以上の数値を観測している部分が分離衝撃を与えている部分となっている。23.63G, 33.9G, 17.13G, 32.9G, 42.58G の分離衝撃を与えた。30G以上を観測した衝撃は33.9G, 32.9G, 42.58G で観測できた。30G以上の衝撃を3回以上与えた後、パラシュートの接合部、CanSat 保護カバーとの接合部に損傷は見られなかった。また機体・センサ等には損傷は見られず、その後パラシュート焼き切り機構とモータ駆動も正常に動作した。CanSat に搭載されているセンサの値や走行にも問題がないことが確認できた。試験の様子は動画に撮り、異常の有無の確認を記録した結果からも、機体の異常は見られなかった。[ae]

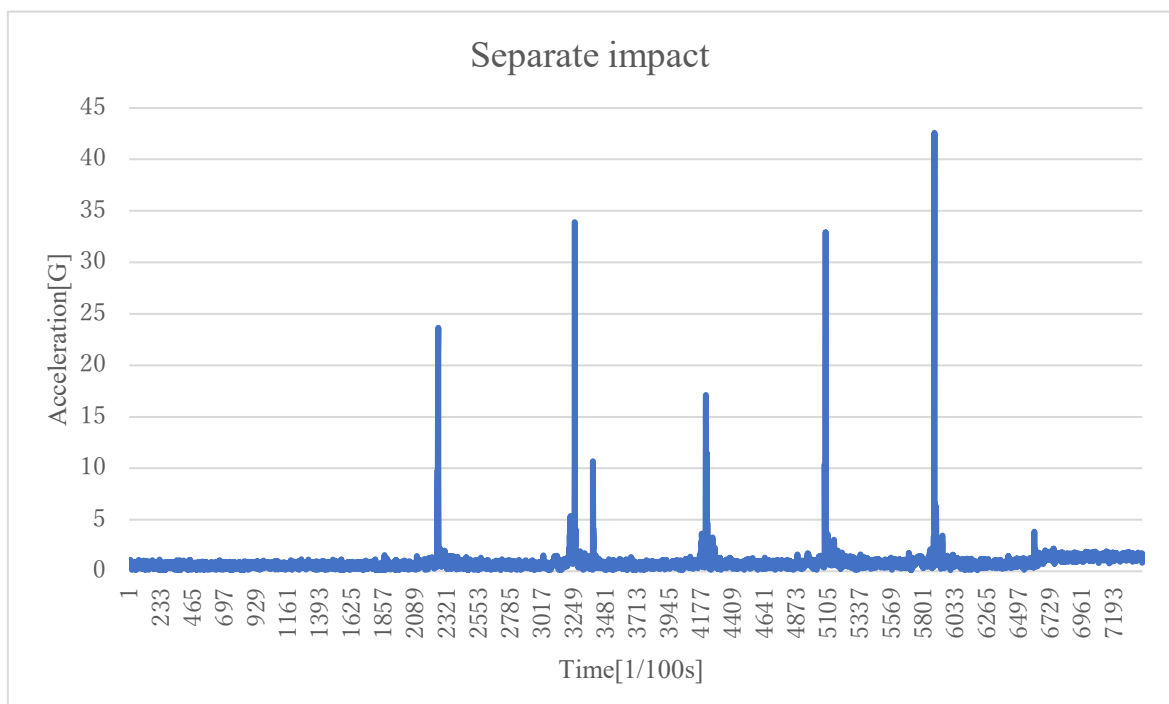


Fig. 3-10-4 Separation impact values

(Vertical axis: acceleration G, Horizontal axis: time 1s)

3-11 落下試験

3-11-1 落下試験内容

能代宇宙イベントでは CanSat がキャリア放出後に減速機構で落下する際に強風で競技フィールド外へ CanSat が落下し、ランバックやミッション競技が困難になる場合がある。競技フィールド外へ落下したときは再度競技として CanSat 投下する権利がある。しかしパラシュートの落下速度が遅い場合には何度もやり直しとなってしまうため、レギュレーションで 5m/s 以上の減速機構の終端速度で落下する設計であることが記載されている。

落下実験では実際に減速機構パラシュートを付けた状態で落下させ終端速度が 5m/s 以上であるか、CanSat に搭載したセンサデータや実際に落下させた実験を映像撮影しデータ根拠から証明する必要がある。

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

3-11-2 落下試験結果

実験結果を Table. 3-11-1 にまとめた。1 回目はグラフ中 3s で 16.713m と 4s で 10.155m の間 1s 間の落下高度は 6.558m であり、 $6.558\text{m}/1\text{s} = 6.558\text{m/s}$ 。2 回目はグラフ中 3s で 17.148m と 4s で 11.612m の間 1s 間の落下高度は 5.986m であり、 $5.986\text{m}/1\text{s} = 5.986\text{m/s}$ 。3 回目はグラフ中 4s で 17.703m と 5s で 11.369m の間 1s 間の落下高度は 6.361m であり、 $6.361\text{m}/1\text{s} = 6.361\text{m/s}$ となり、3 回 5m/s 以上の終端速度であることが確認できた。

Table 3-11-1 Parachute Drop Test Results

実験回数	終端速度	5m/s以上か ○ or ×	実験データグラフ	動画URL																						
1回目	6.558 m/s	○	<p style="text-align: center;">1回目</p> <table border="1"> <caption>Altitude vs Time for 1st Trial</caption> <thead> <tr> <th>Time [s]</th> <th>Altitude [m]</th> </tr> </thead> <tbody> <tr><td>1</td><td>17</td></tr> <tr><td>2</td><td>17</td></tr> <tr><td>3</td><td>17</td></tr> <tr><td>4</td><td>10</td></tr> <tr><td>5</td><td>7</td></tr> <tr><td>6</td><td>8</td></tr> <tr><td>7</td><td>8</td></tr> <tr><td>8</td><td>8</td></tr> </tbody> </table>	Time [s]	Altitude [m]	1	17	2	17	3	17	4	10	5	7	6	8	7	8	8	8	[af]				
Time [s]	Altitude [m]																									
1	17																									
2	17																									
3	17																									
4	10																									
5	7																									
6	8																									
7	8																									
8	8																									
2回目	5.986 m/s	○	<p style="text-align: center;">2回目</p> <table border="1"> <caption>Altitude vs Time for 2nd Trial</caption> <thead> <tr> <th>Time [s]</th> <th>Altitude [m]</th> </tr> </thead> <tbody> <tr><td>1</td><td>18</td></tr> <tr><td>2</td><td>17</td></tr> <tr><td>3</td><td>17</td></tr> <tr><td>4</td><td>12</td></tr> <tr><td>5</td><td>8</td></tr> <tr><td>6</td><td>8</td></tr> <tr><td>7</td><td>8</td></tr> <tr><td>8</td><td>8</td></tr> </tbody> </table>	Time [s]	Altitude [m]	1	18	2	17	3	17	4	12	5	8	6	8	7	8	8	8	[ag]				
Time [s]	Altitude [m]																									
1	18																									
2	17																									
3	17																									
4	12																									
5	8																									
6	8																									
7	8																									
8	8																									
3回目	6.361 m/s	○	<p style="text-align: center;">3回目</p> <table border="1"> <caption>Altitude vs Time for 3rd Trial</caption> <thead> <tr> <th>Time [s]</th> <th>Altitude [m]</th> </tr> </thead> <tbody> <tr><td>1</td><td>19</td></tr> <tr><td>2</td><td>18</td></tr> <tr><td>3</td><td>18</td></tr> <tr><td>4</td><td>18</td></tr> <tr><td>5</td><td>12</td></tr> <tr><td>6</td><td>9</td></tr> <tr><td>7</td><td>8</td></tr> <tr><td>8</td><td>9</td></tr> <tr><td>9</td><td>9</td></tr> <tr><td>10</td><td>9</td></tr> </tbody> </table>	Time [s]	Altitude [m]	1	19	2	18	3	18	4	18	5	12	6	9	7	8	8	9	9	9	10	9	[ah]
Time [s]	Altitude [m]																									
1	19																									
2	18																									
3	18																									
4	18																									
5	12																									
6	9																									
7	8																									
8	9																									
9	9																									
10	9																									

3-12GNSS データダウンリンク試験

3-12-1GNSS データダウンリンク試験内容

ARLISS ではロケットで高度 4000m まで打ち上げ後、CanSat は分離しパラシュートなどの減速機構で落下する。4000m からの落下では落下中の上空の風の状態により着地地点がどこになるか予測ができず、CanSat を検索することが困難になり、CanSat が見つからずロストしてしまう場合がある。ロストを防止するため長距離から CanSat の居場所を把握できる通信機能などを搭載しロスト対策をする必要がある。

ロスト対策として GNSS 位置情報データを長距離から無線通信で受信することで、CanSat の場所を追跡しロストを防止できないか実験を行った。ロケットは地上から高度 4000m 上空まで打ち上げられるため、長距離無線通信ができる目標距離として 4 km 以上を基準に実験を行った。

interplun IM920（使用周波数：920MHz 帯）を使用し、長距離での無線の受信を行い、CanSat の位置する方向の特定が可能か確認した。Interplun IM920c(送信機)からの電波を Interplun IM920(受信機)に受信し、受信データを接続した PC に送信し Teraterm にて表示することができれば成功とする。今回の実験では、送信データは経過時間だが、本番では位置座標を送信することにより、ロスト対策を行う。

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

実験に使用した受信機 Interplun IM920 の写真を Fig. 3-12-1、送信機 Interplun IM920c を搭載した CanSat の写真を Fig. 3-12-2、実際に使用した送信機と受信機を PC につなげた状態の写真を Fig. 3-12-3 に示す。また、実際に距離測定を行った位置座標を示す。距離の指定場所は約 4.0km、緯度 34.79416666 経度 137.13055555～緯度 34.76916666 経度 137.16861111 の距離である。



Fig. 3-12-1 Receiver Interplun IM920

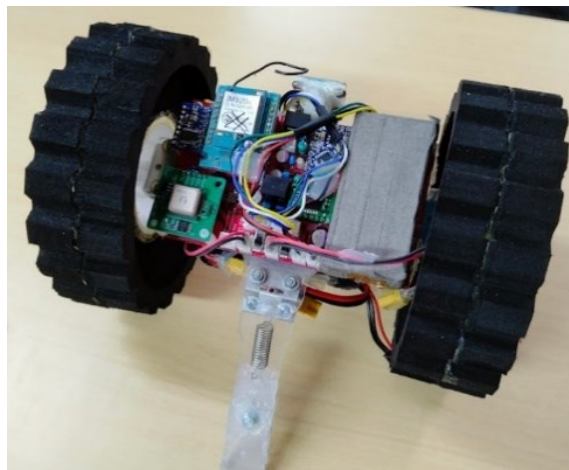


Fig. 3-12-2 Transmitter Interplun IM920c on CanSat

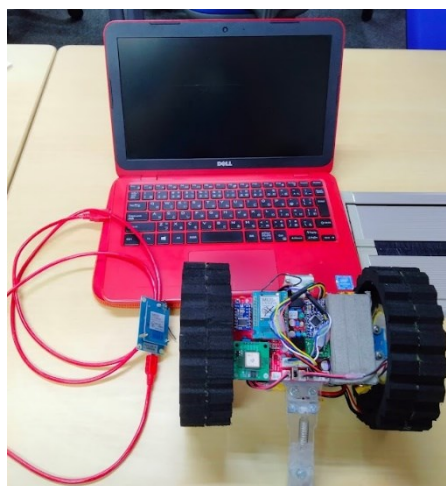


Fig. 3-12-3 Receiver (left) and transmitter (right)

3-12-2 GNSS データダウンリンク試験結果

4km の無線受信実験で、CanSat を組み込んだ送信機位置を【緯度 34.79416666 経度 137.13055555】に配置し、受信機を【緯度 34.76888888 経度 137.16861111】に配置し無線通信を試したところ、無線通信が成功した。

送信機位置写真を Fig. 3-12-4、送信機位置の緯度経度を Fig. 3-12-5、受信機位置写真を Fig. 3-12-6、受信機位置の緯度経度を Fig. 3-12-7、受信～送信位置の Google MAP を Fig. 3-12-8、実際に取得したデータを Fig. 3-12-9 示す。送信機から受信した GNSS データをリアルタイムで表示されている動画を撮影した。[ai]実験を行った結果、約 4km の長距離通信が可能であることが確認できた。



Fig. 3-12-4 Photo of transmitter location

緯度	34.79416666			⇒	緯度経度をGoogleMapで確認				
経度	137.13055555								
	↓	↑							
緯度	34	度	47	分	39	秒			
経度	137	度	7	分	50	秒			

Fig. 3-12-5 Latitude and longitude conversion of transmitter position
(conversion from GNSS logger hour, minute, second)



Fig. 3-12-6 Photo of receiver location

緯度	34.76888888	⇒	緯度経度をGoogleMapで確認				
経度	137.16861111						
<div style="display: flex; gap: 10px;"> ↓ ↑ </div>							
緯度	34	度	46	分	8	秒	
経度	137	度	10	分	7	秒	

Fig. 3-12-7 Latitude and longitude conversion of receiver position
(from GNSS logger hour, minute, second)

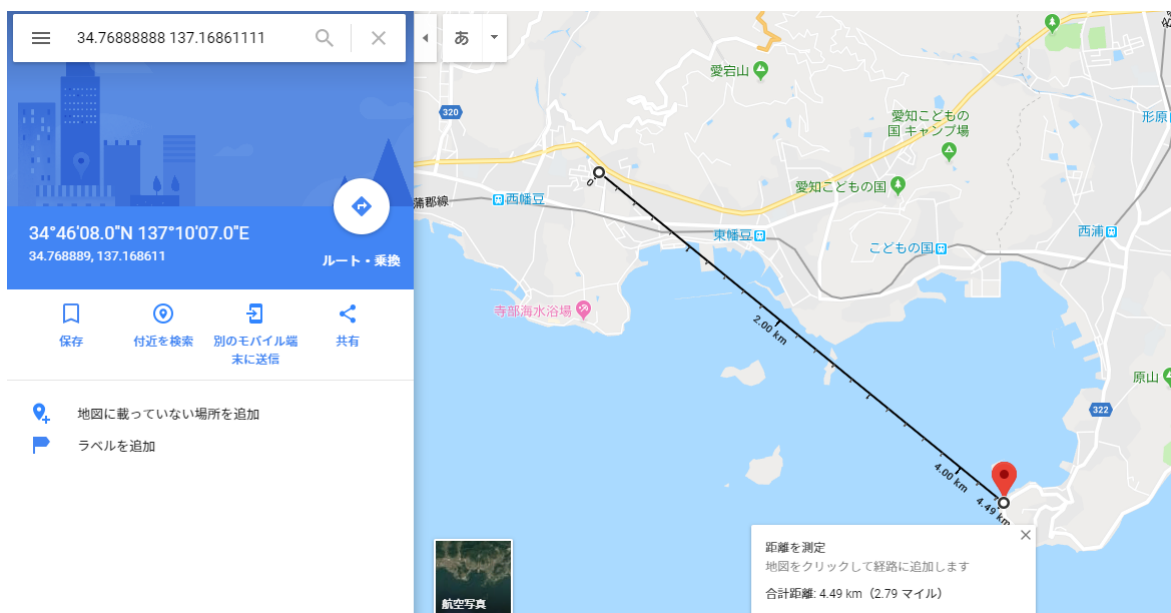


Fig. 3-12-8 Distance measurement in GoogleMap between 4 km

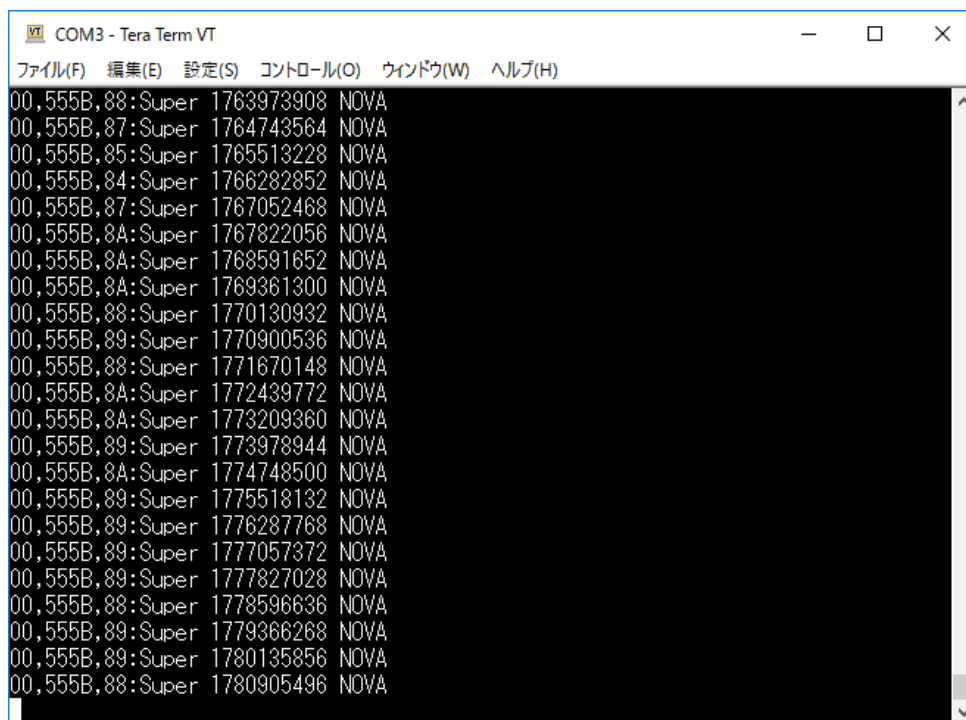


Fig. 3-12-9 Acquired data

3-13 通信機電源 OFF/ON 試験

3-13-1 通信機電源 OFF/ON 試験内容

ARLISS では打ち上げロケットの無線通信と CanSat の無線通信が電波干渉の無いよう、ロケットから放出されるまで CanSat 無線機の電波は FCC 認証かつ 100mW 以下の機器以外は OFF にできるようにしておく必要がある。

使用する通信機 interplun IM920 は、既定の電波強度以下であるが FCC 認証でないため、気圧センサ値から求めた高度値を用いて通信機の電源供給を遮断する方式とした。プログラムにて CanSat 電源を入れてから 20 秒後、小さい高度を検知している間はスリープモード(DSRX)に通信電波が OFF 状態になり、大きい高度を数秒間検知するとロケットが打ち上げられたと判断しその後 CanSat 電源入れた時の高度を数秒間検知するとロケットから放出され着地したと判断し、ジャイロセンサ値の変化が小さくなると地面に落ち停止している状態と判断し(ENRX)に入るプログラムを作成した。

実験では 4000m ほどのロケットの高度を観測する実験が不可能であるため、愛知工科大学 7 号館の建物の高さをロケットの高さと仮定した環境で行った。大学のエレベータを利用し、CanSat 電源 ON 時 1F にて高度を検知し、通信電波が OFF となっていることを確認し、その後 9F へ上昇しロケットが上昇した状態と仮定し、高度値が初期値から 15m 以上に上昇していることを 10 秒間検知したあと、1F へ降下し高度が初期値に近い値として 15m 以下を 10 秒間検知したあと、ジャイロセンサの変化が小さくなると、地面に落ち停止している状態と判断し通信電波が ON になることを確認した。また、通信条件の手順を以下の①②③④に示す。

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

また、実験機器として、PCに無線ログと、CanSatへ有線で接続した時のプログラムのリアルタイムログを並列に表示させ、更に電波観測機としてスペクトラムアナライザで電波を測定し、電波のON/OFFを目視で確認できるようにした。実際の実験機器を Fig. 3-13-1 に示す。



Fig. 3-13-1 Experimental equipment for visual confirmation of radio wave ON/OFF

3-13-2 通信機電源 OFF/ON 試験結果

実際の結果を、通信機 OFF、設定した高い高度を観測したか、高い高度から落下し終え通信機が ON にしたか結果をまとめた。建物 1F で CanSat 電源 ON し 20 秒後、無線通信ログが停止し、スペクトラムアナライザが 920MHz 帯の電波を観測しなくなり、無線通信が OFF になった。実際の実験で PC に無線ログと、CanSat へ有線で接続した時のプログラムのリアルタイムログと、スペクトルアナライザの表示写真を Fig. 3-13-2 に示す。

建物 9F に上昇後、Raspberry Pi Log より、高度 31m を観測し、上昇したことを観測した。実際の高度観測リアルタイムログの写真を Fig. 3-13-3 に示す。

1F へ降下し、CanSat を地面に置き停止させ、高度が降下したことを観測し、ジャイロセンサ値の変化が小さくなったことから、地面に着地したと判断し、無線電波が ON になった。実際の実験で PC に無線ログと、CanSat へ有線で接続した時のプログラムのリアルタイムログと、スペクトルアナライザの表示写真を Fig. 3-13-4 に示す。

また、この試験の様子を動画にまとめた。[aj] 試験により、通信機電波がロケット格納時に OFF、ロケットから CanSat が放出される際に ON となることを確認できた。

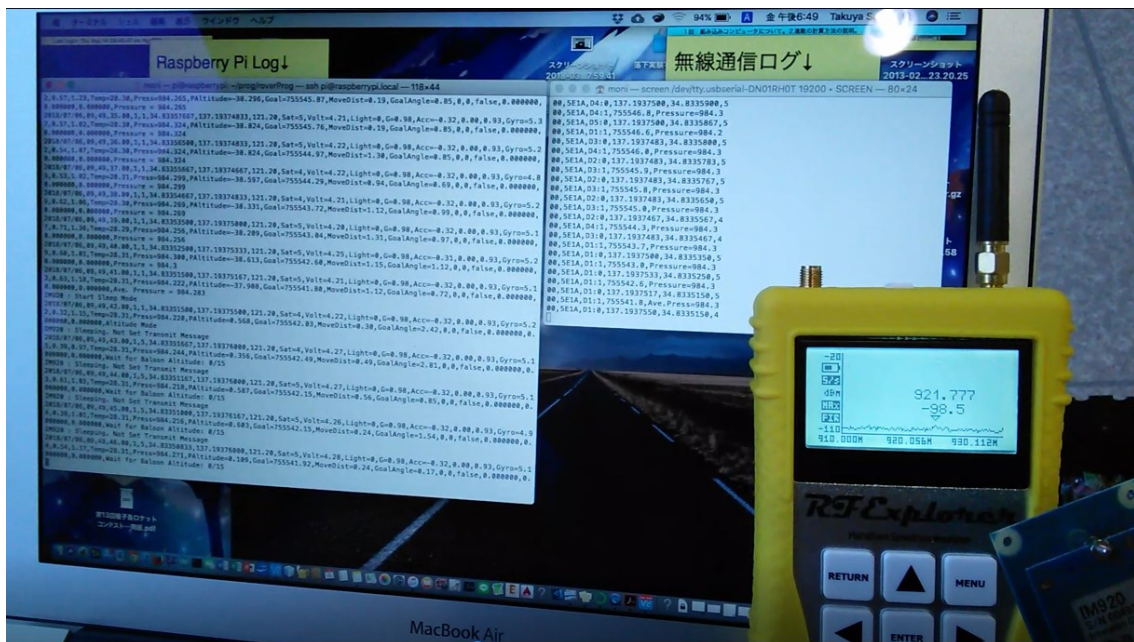


Fig. 3-13-2 Power OFF confirmation photo

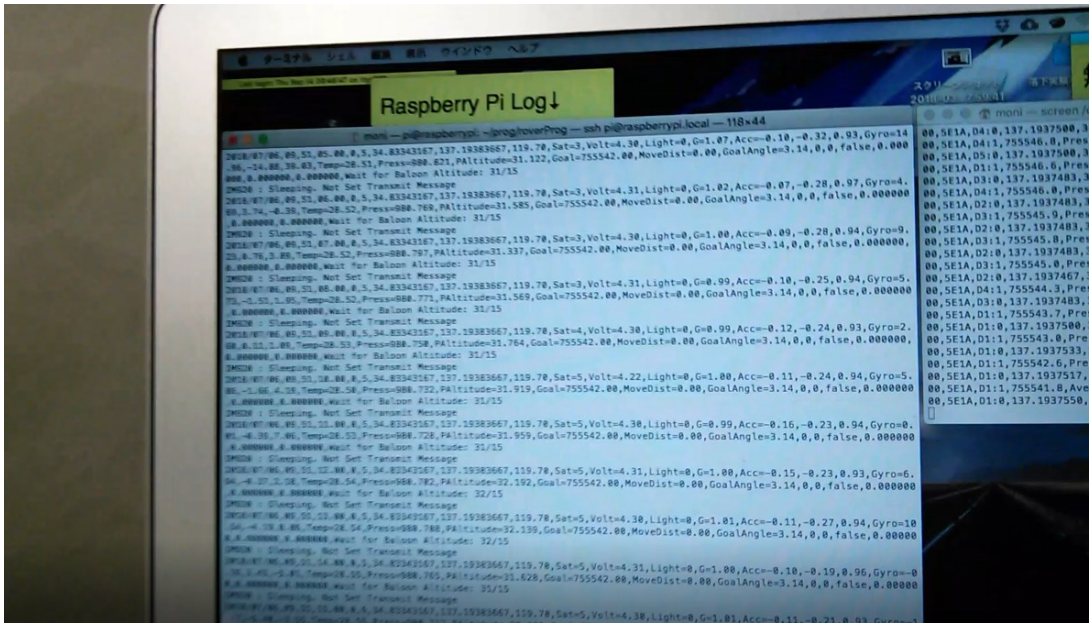


Fig. 3-13-3 High altitude observations



Fig. 3-13-4 Photo of radio ON confirmation after altitude descent

3-14 通信周波数 ch 変更試験

3-14-1 通信周波数 ch 変更試験内容

CanSat 競技では各チームの CanSat の無線通信が電波干渉しないよう同じ周波数の電波無線を使用するチームと競技が重なる場合には周波数変更やチャンネル変更ができる仕様にしておく必要がある。

使用する予定の無線 interplan IM920 をプログラムにより、状況に応じて使用する通信機を切り替える措置を行った。プログラム上でチャンネル変換を行い、データを受信する Tera Term 上で stch (チャンネル数)を入力後データが受信される仕様となっている。これらの機能を用いて、無線チャンネル変更ができることを確認する。

3-14-2 通信周波数 ch 変更試験結果

実際に利用している無線設定プログラムの説明と、そのプログラムのチャンネル変更を実験の結果を記述する。M920 と接続された Raspberry Pi zero W メインプログラム上で `sc16is750.setChannel();` の () 内の数字をチャンネル数に変更しコンパイルすることでチャンネルが変更される。実験では 5 チャンネルから 7 チャンネルに設定変更を行った。実際のプログラムを Fig. 3-14-1 に示す。IM920 が接続された PC の TeraTerm 上で STCH 07 とコンパイルすると、チャンネルが 7 になる。実際のデータログを Fig3-14-2 である。無線 (interplan IM920) のチャンネル変更については、プログラムによりチャンネルを変更し実際に通信できることを確認した。また、この試験の様子を動画にまとめた。[ak]

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

Fig. 3-14-1 Channel change program

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

Fig. 3-14-2 Data log of channel changes in the experiment

3-15 着地衝撃試験

3-15-1 着地衝撃試験内容

CanSat では落下後に地表面に着地した衝撃で CanSat が故障することが多い。CanSat にはロケット荷重や分離衝撃などの様々な衝撃が加わるが、着地衝撃はその中でも最も衝撃が大きい。着地衝撃試験は能代宇宙イベントの審査書では必須試験としている。

パラシュートの終端速度から自由落下したときと衝撃が同じになる高さを求め、その高さから自由落下させたときに CanSat に故障がないか確認する。終端速度から自由落下の高さを(3-15)に求めた。 $v^2 =$ パラシュート終端速度 5m/s , $a =$ 重力加速度 9.8g とした。

$$\text{終端速度(パラシュート投下試験参照)} = \frac{v^2 - v_0^2}{2a} = \frac{5^2 - 0}{2 \times 9.8} = 1.275 \text{ m} \quad (3-15)$$

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

3-15-2 着地衝撃試験結果

着地衝撃を3回与えた。実際にかかった加速度のログデータを Fig. 3-15-2 に示す。観測した着地衝撃は1回目 19.23G、2回目 33.42G、3回目 19.74G だった。また、電子回路の GNSS やセンサ類の値、パラシュート離脱機構、モータに異常がないか、CanSat 機体が破損していないかの確認を行い、故障無く CanSat に問題がないことを確認した。着地衝撃試験の動画に撮り、Youtube にて確認できるようにした。実際の試験の様子を下記 URL に示す。[a1]

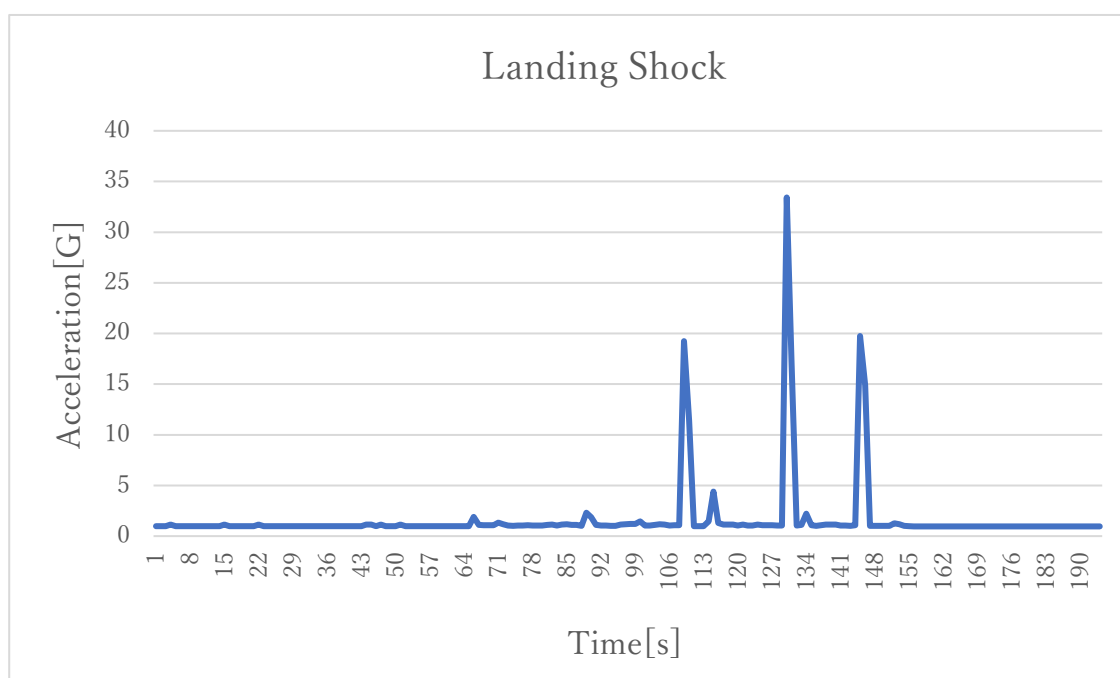


Fig. 3-15-2 Numerical graph of landing shock
(vertical axis: acceleration G applied, horizontal axis: time s)

3-16 走行性能確認試験

3-16-1 走行性能確認試験内容

CanSat はランバックにおいて走行時に轍や大きな草により走行ができなくなる場合がある．走行性能確認試験ではランバック走行時の走破性能があるか確認を行う試験である．能代宇宙イベントでの審査書では必須試験項目となっている．

能代宇宙イベントの草の状態に似たフィールドで CanSat を走行させ、CanSat がスタックしないか確認した．昨年能代宇宙イベント競技にて茎が太く大きな草にスタックし走行不能となった．そのため競技フィールドに似た大きな草が生えている土地で CanSat を走行させ、今回のミッションであるタイヤの形状を草がスタックしにくいように変更することで、茎が太く大きな草にスタックしても離脱ができることを確認した．実験での茎が太く大きな草への画像の例を、Fig. 3-16-1 に示す．



Fig. 3-16-1 Image of the experiment

3-16-2 走行性能確認試験結果

実験結果を表にまとめた。段差の走破を調査した表を Table 3-16-1, 草の走破を調査した表を Table 3-16-2 に示す。段差による走破性能調査は 1 回目では段差約 45° ほどの草を乗り越えた。2 回目では段差約 30° の草を乗り越えた。3 回目では段差約 30° の草を斜めから乗り越えた。

草の走破性能調査では 1 回目は約 25cm の長さ細い草を距離約 2m 走破した。2 回目では約 10cm 長さの太い草を距離約 10m 走破した。3 回目では約 25cm 長さの太い草を距離約 5m 走破した。4 回目では約 20cm 長さの太い草を距離約 7m 走破した。

Table 3-16-2 Running performance test results (difference in level)








実験回数	走破	草の状態	結果詳細	動画 URL
1 回目	○	段差 約 45° 	段差約 45° ほどの草を乗り越えた。	[aq]
2 回目	○	段差 約 30° 	段差約 30° ほどの草を乗り越えた。	[ar]
3 回目	○	段差 約 30° (斜めから) 	段差約 30° ほどの草を斜めから乗り越えた。	[as]

Table 3-16-1 Running performance test results (grass)

実験回数	走破	草の状態	結果詳細	動画URL
1回目	○	長さ約 25cm 細め 	約 25cm の長さ細い草を距離約 2m 走破した。	[am]
2回目	○	長さ約 10cm 太め 	約 10cm 長さの太い草を距離約 10m 走破した。	[an]
3回目	○	長さ約 20cm 太め 	約 25cm 長さの太い草を距離約 5m 走破した。	[ao]
4回目	○	長さ約 20cm 太め 	約 20cm 長さの太い草を距離約 7m 走破した。	[ap]

3-17End to End

3-17-1End to End 内容

End to End は競技の最初から最後まで通しでの動作実験することで各種シーケンスに問題がないかを確認する試験である。充足を確認した設計の機体によって、ロケットへの装填から打ち上げ後の回収までを模擬した End to end 試験を実施できており、今後、安全性に関わる大幅な設計変更はないようにする。CanSat の動作を本番と同じ手順で行う。各シーケンスを自律で行えることを確認できる動画を提出する。End to End は 1-7 の手順で行われる。

- 1 キャリアから CanSat が自重で放出
- 2 サンセンサにより放出判定（本番では高度判定）
- 3 着地衝撃
- 4 パラシュートからの離脱
- 5 目的地まで GNSS 位置情報で自律走行
- 6 Object Detection ゴール接近&判定
- 7 ログデータ取り出し

1-7 までの一連の流れを通してできるか確認する。

3-17-2 End to End 結果

EndtoEnd の実験結果を Table 3-17-1 に示す。EndtoEnd の実験が問題なく行われているか成功または失敗によって評価を行い、EndtoEnd 試験の詳細をまとめた。EndtoEnd が正常に行われた様子を動画で撮影した。また、制御履歴を制御履歴レポート試験にて記載した。

Table 3-17-1 End to End results

成功 Or 失敗	詳細	URL
成功	キャリア放出後、サンセンサの値がキャリア内から外へ放出したと判定後、パラシュートと分離し、GNSS 情報からゴール近くまでへ走行した。ゴール約 5m 付近から Object Detection でのゴール検知を開始、ゴールがカメラに映るまで旋回し、0m まで接近後ゴール判定した。その後 Object Detection ログ画像の取り出しと、走行軌跡を出力するプログラムを動作させ EndtoEnd の走行奇跡ログデータを取り出すことができた。	[at]

3-18 制御履歴レポート試験

3-18-1 制御履歴レポート試験内容

カムバック競技において落下した地点が偶然ゴール近くであり、動作できずに記録となり、落下した地点で競技結果となることを防ぐため、落下後に GNSS 位置情報データによるモータの制御値を制御履歴として制御履歴担当審査教員へ提出し制御が認められることで、カムバック部門での距離の記録が認められる。

制御履歴レポート試験では規定の制御履歴 End to End 試験のデータから作成し、大会終了後の大会報告書で制御履歴の提出できることを確認する。

3-18-2 制御履歴レポート試験結果

CanSat のプログラムを起動した地点をスタート地点として、スタート地点からゴール地点へ走行した時の GNSS データやセンサ値や制御値をサンプリング 1s と 0.01s でマイコンの SD カードに保存する。ゴール判定後、ログデータファイルを読み取り、テキストデータを Excel ファイルでまとめ見やすい形にして表示した。また、走行軌跡が見やすくなるように、ステアリングを色で変化させた走行軌跡を表示する。実際に End to End で行ったデータを用いて、制御履歴作成を行った。

制御履歴提出に色付き及び矢印付きの走行軌跡図を作成した図を Fig. 3-18-1 に示す。走行軌跡図では、右旋回へのステアリング最大値 0.20(黄色)、左旋回へのステアリング最大値 -0.20(紫)として緯度経度から走行軌跡を色で示した。また、スタート地点、ターゲット地点、ストップ地点を矢印で示し、その地点の緯度経度を示した。

制御履歴 text 生データを、Excel ファイルで見やすくまとめた。基本的な制御出力系データを確認できる。Excel ファイルはドライブ URL から確認できる。また制御項目の 1 部を Fig. 3-18-2 に示す。制御項目は、時系列データ A~D 列、制御出力系データ AF~AS 列、GNSS 位置情報値 I,J 列に記される。また、GNSS による制御方法のアルゴリズムを Fig. 3-18-3 に示す。

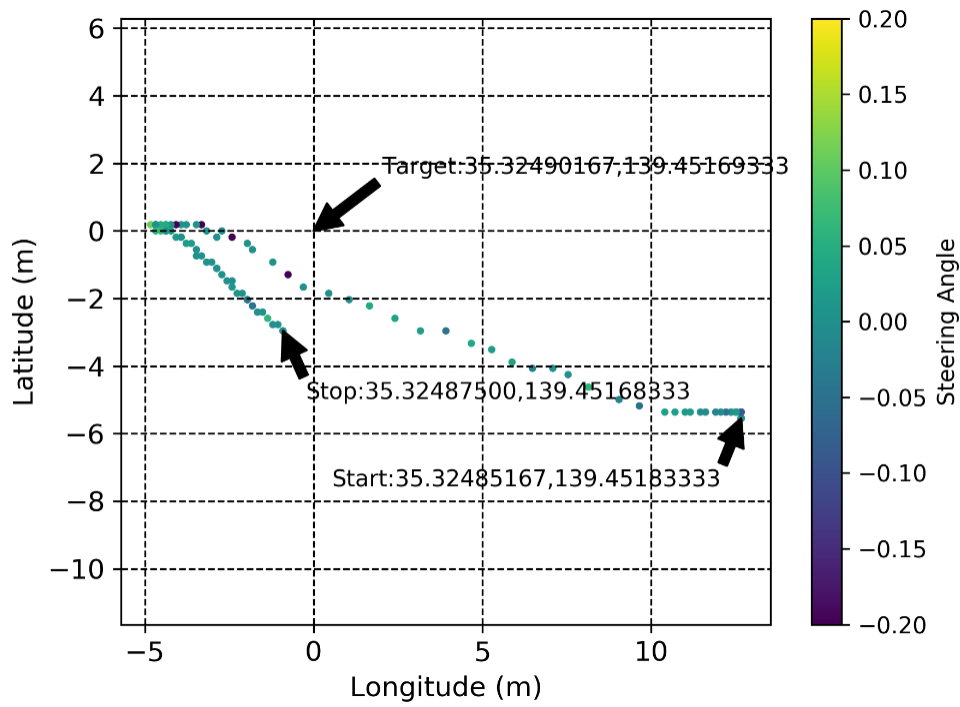


Fig. 3-18-1 End to End Running Path

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

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

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

Fig. 3-18-2 Control items summarized in Excel

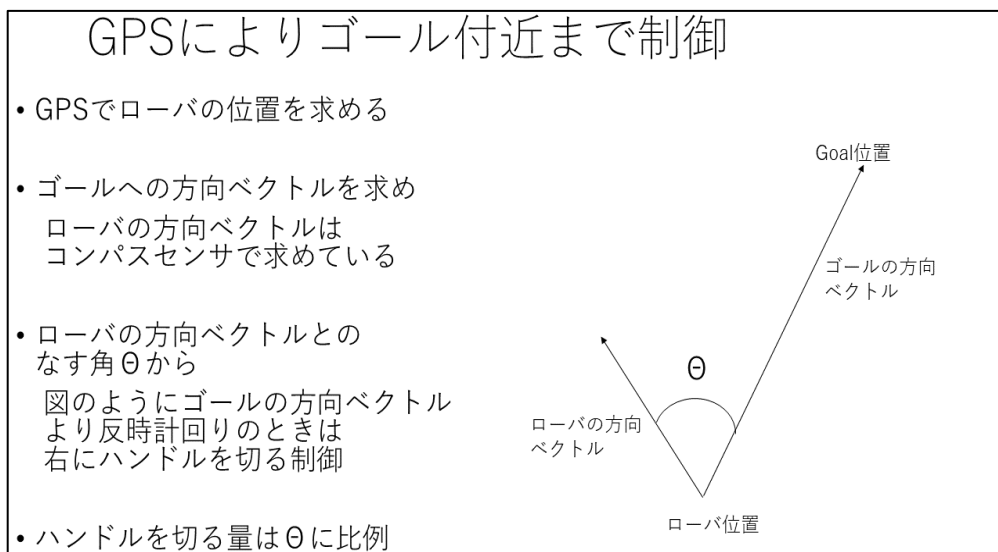


Fig. 3-18-3 Control Method


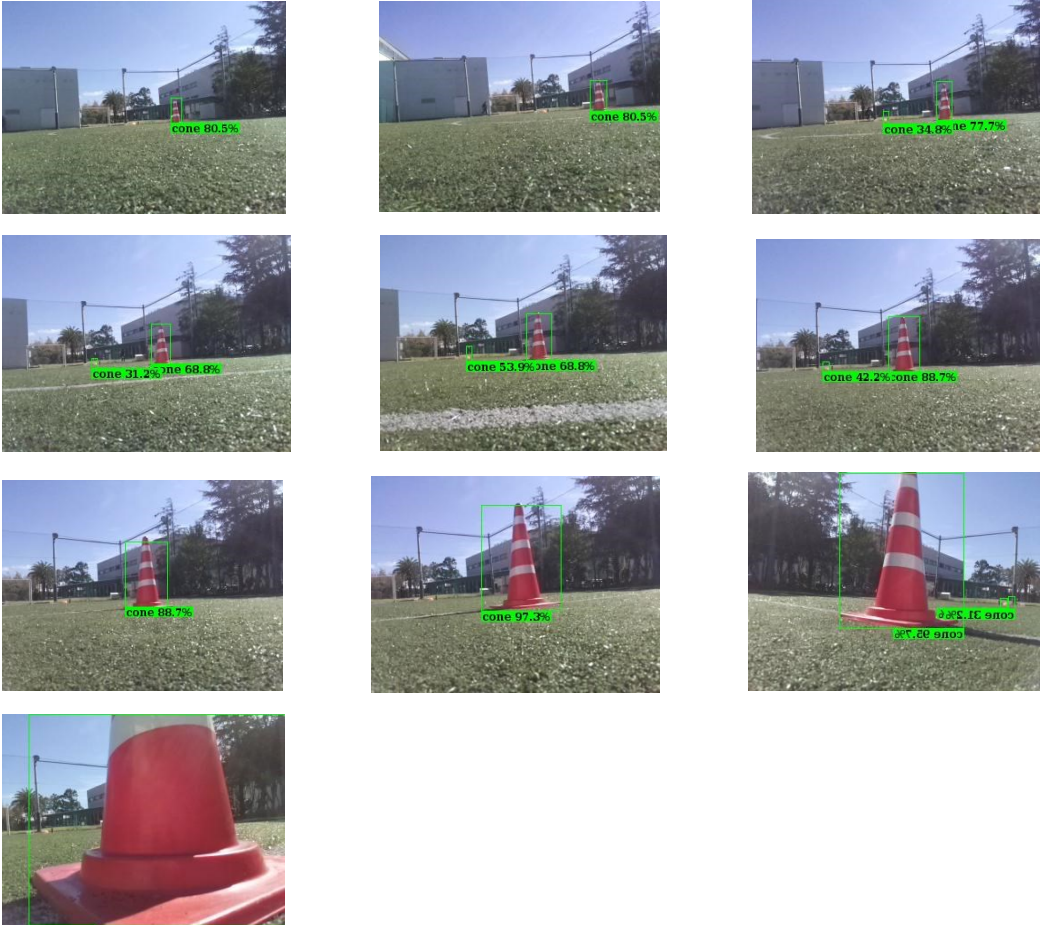
また、制御出力系データの項目の説明を示す。AH列CanSat進行方向とゴールとの角度radは1秒前と現在のCanSatの一からCanSatの進行方向とゴールとの角度をラジアン単位で求め、制御に反映している。AI列モータ出力スピード(-255~+255)はモータに加える出力を-255~+255(右~左)で与えている。AJ列スピード分解能はGNSSを1秒ごとで制御しているが、新しい速度にする際のノイズ対策として、指定時間(0.1秒)の線形変化でモータの出力を変換している。(電圧の変化を少量ずつにする事で、ノイズ対策)AL列ステアリング補正(TRUE or FALSE)は制御値からCanSatにステアリング補正を与えるかどうかを判定し、TRUEかFALSEで表している。AC列ステアリング角度(-1~+1)(-1:左,+1:右)は左旋回最大-1,右旋回最大+1として、-1~+1でステアリング角度を表している。AM列修正舵(ステアリングI制御)はモータ損傷時を考慮し、制御が片方のステアリング角度が著しく大きい時、もう片方のステアリング角度を大きくするようI制御を与えている。AN列実際のステアリング角度(-1~+1)はAC列ステアリング角度をAD列修正舵に適応させた、実際にモータへ出力するステアリング角度を表している。AO列右モータ目標回転数(-255~255)は回転数制御による右モータステアリング目標回転数である。AP列左モータ目標回転数(-255~255)は回転数制御による左モータステアリング実際の測定結果の回転数。

AQ 列右モータ目標回転数(-255~255)は回転数制御による右モータステアリング目標回転数である。AR 列左モータ目標回転数(-255~255)は回転数制御による左モータステアリング実際の測定結果の回転数である。

また、End to End での Object Detection の画像ログを Table 3-18-1 に示す。また、実際の制御ログデータの End to End 制御履歴 text 生データ、Excel ファイル、走行軌跡、ゴール画像をドライブ URL にまとめた。

[au]

Table 3-18-1 Object Detection

Object Detection での 0m ゴールまでの画像ログ & 詳細		
<p><u>ゴール発見まで 4 回旋回</u></p> 		
<p><u>発見後, 10 回の旋回 & 直進でゴール判定</u></p> 		

3-19 その他ミッション要求試験

3-19-1 ゴール検知試験内容

ゴール検知試験はミッション要求を満たすか確認を行う試験である。目標位置から約 10m 以上離れた地点から、GNSS で数 m まで接近後、目標カラーコーンを Object Detection による画像認識で接近し 0m ゴールを行う。下記に Object Detection の仕様詳細等を説明する。

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

昨年時に利用していた Image Classification は Raspberry Pi Zero W で可能な限り短時間で認識できるよう Tensorflow から Tensorflow Lite に変換すると、1 回ゴール認識に約 12 秒程度かかった。そこで Deep Learning 用計算アクセラレータの Coral USB Accelerator を Raspberry Pi Zero W に接続し、高速化した。これにより、1 回ゴール認識は約 12 秒から約 0.3 秒でできるようになっている。Object Detection 使用詳細を 1-3 に示す。また、実際のアノテーションの例を Fig. 3-19-1 に示す。

1.モデル：SSD MobileNet V1

2.入力サイズ：300×300pixel,RGB3 チャンネル

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

走行制御方法は、Raspberry Pi Camera v2 から撮影された画像から、Object Detection で認識結果が Cone を 50% 以上で見つけるとゴールが見つかったと判定し、認識範囲の重心座標の方へハンドルを切ってゴールへ接近走行している。また、複数ゴールが見つかった場合はコーンの可能性が 1 番高いものだけを採用している。ゴールが見つからない場合は左旋回を繰り返す。ゴールが見つかったときの例の画像を Fig. 3-19-2 に示す。ゴール判定では、横幅が 250 ピクセル以上のときゴール判定をしている。ゴール判定をした例の画

像を Fig. 3-19-3 に示す。

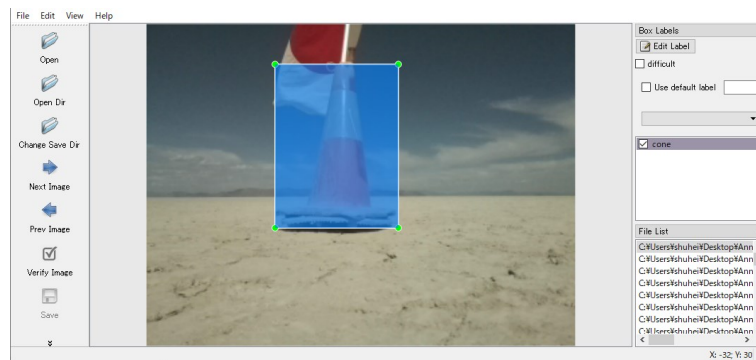


Fig. 3-19-1 Annotations



Fig. 3-19-2 Finding Goal in Object Detection



Fig. 3-19-3 Goal Judgment in Object Detection

3-19-2 ゴール検知試験結果

ゴール検知試験結果を 1 回目 Table 3-19-1, 2 回目を Table 3-19-2, 3 回目を Table 3-19-3 に示す. ゴール検知を行うことで 0m ゴールができた場合○と表に記載する. また, 実際の撮影写真と詳細を表に記載した. また, ゴール検知試験画像ログ, 制御ログをまとめたドライブ URL に示した. [av]

Table 3-19-1 Goal detection test results 1st

0m goal	1回目 実際の撮影写真&詳細	動画 URL
○	<p>ゴール発見まで 5 回巡回</p> 	
	<p>発見後 5 回の巡回 & 直進しゴール判定</p> 	[aw]

Table 3-19-2 Goal detection test results 2nd


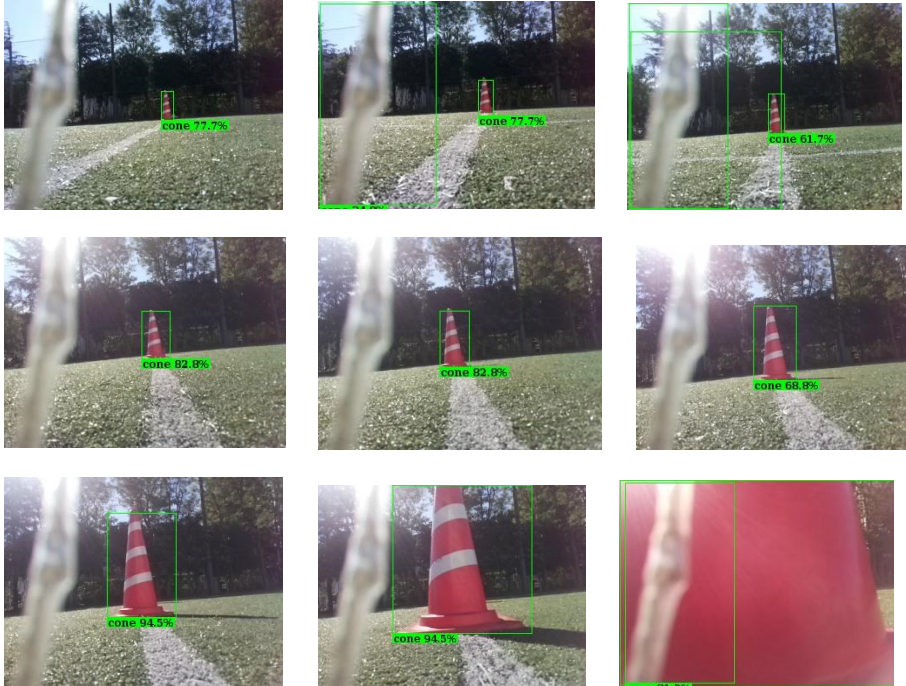
0m goal	2回目 実際の撮影写真&詳細	動画 URL
	<p>ゴール発見まで 6 回巡回</p> 	
○	<p>発見後, 9 回の巡回 & 直進でゴール判定</p> 	[ax]

Table 3-19-3 Goal detection test results 3rd

0m goal	3回目 実際の撮影写真&詳細	動画 URL
	<p>ゴール発見まで 4 回巡回</p> 	
○	<p>発見後, 10 回の巡回 & 直進でゴール判定</p> 	[ay]

3-20 ガントチャート

CanSat 審査書では CanSat の開発計画においてガントチャートにて詳細に計画を立て、その計画に問題がないか審査を行う。ガントチャートを Fig. 3-20 に示す。

全体的に祝日と土日を休みとして扱い、本来かかる時間の倍程度の日程で余裕を持ったスケジュールになっている。メンバーが複数人いないために担当ごとの項目と、大学院のため試験のスケジュールは無い。全体的には 6 月にプロトタイプ作成&テスト、7 月に EndtoEnd 以外の審査試験、8 月に EndtoEnd&最終調整とリカバリーを行う。

6 月中は CanSat のプロトタイプを作成し、問題点などテストし確認し、改良を重ねていく。第 1 週は従来より改良する点を考案し部品や材料の発注をする。第 2 週から 4 週にテスト CanSat 作成及び問題点など確認し改良を重ねる。第 5 週はリカバリーができるようできるだけスケジュールを空け、テストを重ねる

7 月中は予備審査に向け、EndtoEnd 以外の試験をできる限り終える。(各 2 試験 1 週間) 第 1 週は質量試験、キャリア収納試験。第 2 週は準静荷重試験、分離衝撃試験。第 3 週は落下試験、走行性能確認試験。第 4 週は振動試験。

(NSE 予備審査書提出) 第 5 週はミッションのゴール検知試験。(ACTS 予備審査書提出)

8 月中は EndtoEnd や本番への最終調整し、予定に余裕を持ち全体的なリカバリー可能にする。第 1 週～第 2 週 EndtoEnd, 制御履歴試験を行う。(NSE 本審査書提出) 第 2 週までに 6 月から始めたミッションのシステムを確立する。第 3 週で試験の結果を踏まえた本番用の CanSat とスペアを製作する。(ACTS 本審査書提出)。第 4 週～第 5 週にリカバリー及び最終調整や準備を行う。

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

Fig. 3-20 Gantt chart

【参考文献】

Supporting Online Materials :

- [k] http://unisec.jp/history/acts/2021/ACTS2021_SuperNOVA.pdf[Accessed Feb 20, 2023]
- [l]http://akizukidenshi.com/download/ds/akizuki/AE-GNSS_manual_r1.06_s.pdf[Accessed Feb 20, 2023]
- [m] <https://www.raspberrypi.org/products/raspberry-pi-zero-w/>[Accessed Feb 20, 2023]
- [n] <https://www.switch-science.com/catalog/5817/>[Accessed Feb 20, 2023]
- [o] <https://cdn.sparkfun.com/assets/c/6/5/8/d/en.DM00053090.pdf>[Accessed Feb 20, 2023]
- [p]https://cdn.sparkfun.com/assets/learn_tutorials/3/7/3/LSM9DS1_Datasheet.pdf[Accessed Feb 20, 2023]
- [q] <https://www.switch-science.com/catalog/2310/>[Accessed Feb 20, 2023]
- [r] <http://akizukidenshi.com/download/MCP3208.pdf>[Accessed Feb 20, 2023]
- [s]http://akizukidenshi.com/download/ds/Toshiba/TB6612FNG_datasheet_ja_20141001.pdf[Accessed Feb 20, 2023]
- [t]<https://www.marutsu.co.jp/contents/shop/marutsu/ds/v78xx-1000.pdf>[Accessed Feb 20, 2023]
- [u] <http://akizukidenshi.com/download/ta48033s.pdf>[Accessed Feb 20, 2023]
- [v]<https://www.interplan.co.jp/solution/wireless/im920/im920c.php>[Accessed Feb 20, 2023]
- [w]http://akizukidenshi.com/download/ds/bosch/BST-BME280_DS001-10.pdf[Accessed Feb 20, 2023]
- [x] <https://www.switch-science.com/catalog/2713/>[Accessed Feb 20, 2023]
- [y] <https://www.pololu.com/product/4867>[Accessed Feb 20, 2023]
- [z] <https://robin.jp/SHOP/HP-G550-0550S3.html>[Accessed Feb 20, 2023]
- [aa] <https://robin.jp/SHOP/HP-G550-0550S2.html>[Accessed Feb 20, 2023]

[ab] <https://youtu.be/8c5nkpluJB8>[Accessed Feb 20, 2023]

[ac] <https://youtu.be/0zAlQskb808>[Accessed Feb 20, 2023]

[ad] <https://youtu.be/AZFDzesqxwI>[Accessed Feb 20, 2023]

[ae] <https://youtu.be/mRiDGR8nIJ0>[Accessed Feb 20, 2023]

[af] <https://youtu.be/5953Tbg-3go>[Accessed Feb 20, 2023]

[ag] <https://youtu.be/t4yRBvbcNMk>[Accessed Feb 20, 2023]

[ah] <https://youtu.be/cxqcJjokrwA>[Accessed Feb 20, 2023]

[ai] <https://youtu.be/yN6jo3j9JEs>[Accessed Feb 20, 2023]

[aj] <https://youtu.be/i2LnoMklAzI>[Accessed Feb 20, 2023]

[ak] <https://youtu.be/Kq6UdYgvaNI>[Accessed Feb 20, 2023]

[al] https://youtu.be/LO44Cxx_eVY[Accessed Feb 20, 2023]

[am] https://youtu.be/cjU6iamw_pE[Accessed Feb 20, 2023]

[an] <https://youtu.be/6DkAx0p8TXU>[Accessed Feb 20, 2023]

[ao] <https://youtu.be/BUgSj4pkEzE>[Accessed Feb 20, 2023]

[ap] <https://youtu.be/8EpsYX0whW0>[Accessed Feb 20, 2023]

[aq] <https://youtu.be/6y91e2jU-ws>[Accessed Feb 20, 2023]

[ar] <https://youtu.be/iCkg-aBB724>[Accessed Feb 20, 2023]

[as] https://youtu.be/_uD-dRfjmSM[Accessed Feb 20, 2023]

[at] <https://youtu.be/C22BAId8d8g>[Accessed Feb 20, 2023]

[au] <https://drive.google.com/drive/folders/1KQbfpw0UfTRibQ5sM4mzXXoUbdICZASD?usp=sharing>[Accessed Feb 20, 2023]

[av] <https://drive.google.com/drive/folders/1X6ckDw7wWwkSn3dRpMSOs3WlkO0DerYV?usp=sharing>[Accessed Feb 20, 2023]

[aw] <https://youtu.be/Wzyks52hrss>[Accessed Feb 20, 2023]

[ax] <https://youtu.be/IE6JOsJyuCE>[Accessed Feb 20, 2023]

[ay] <https://youtu.be/nokhP7sH7gI>[Accessed Feb 20, 2023]

第 4 章 GNSS 測位精度を改善するゴール誘導制御手法

4-1 GNSS による誤差の問題点

CanSat では目標位置へ接近する制御手法として GNSS 位置情報を用いて誘導制御を行う。ARLISS2016 では 3 チームが GNSS 位置情報によるゴールを達成した。しかしゴールからの距離で競技記録となるカムバック競技において、同じゴール判定制御であったとしても、GNSS の測位誤差において約 1m ずつ差がついてのゴール距離記録となった。このような GNSS 測位誤差により、ゴールまで誘導制御を行うときゴールから数 m 範囲でのズレが生じることから、これ以上の精度でゴールへの接近制御を行う場合には、GNSS 測位精度を高めたり、誤差範囲以内まで GNSS で接近後に新たな接近制御手法でより接近させる必要がある。カムバック部門ではゴールまでの距離を競技記録として評価するため、ゴールに接触した状態でゴール判定をする 0m ゴールを行うことで GNSS 測位誤差の問題を解決できないか検討を行った。

4-2 通信機器における GNSS 電波干渉調査

ロスト対策やリアルタイムログの確認のため無線通信機を搭載する。ARLISS2016 での CanSat に搭載している通信装置 XBee[az]から出力される電波により、GNSS の測位精度に影響があるのではないかと仮定し、その影響について実験により調べた。その結果、XBee から出力される電波が GNSS の測位精度に明らかに影響を与えることが分かった。さらに、実験の結果 Arduino の計算精度自体が測位精度にも影響を与えていることが分かった。これら GNSS 測位精度を低下させる要因について実験により明らかにし、組込機器における対策方法について述べる。

4-3 XBee による GNSS 測位精度影響実験

XBee により出力される電波が GNSS の測位精度に影響を与えているのかを調査するため、XBee による GNSS 測位精度影響調査実験を行った。本実験では同一構造のローバ 2 台を用いてデータの取得を行った。実験に使用した 2 台のローバを Fig.4-3-1 に示す。XBee には XBee-PRO ZB S2B の Wire アンテナタイプを使用した。GNSS には Adafruit[bc] Ultimate GNSS Breakout v3 を用いた。また、これらローバに使用している実験機材を Table 4-3-1 に示す。制御用マイコンは Genuino 101[bb]を用いた。実験環境は、2017 年 6 月 13 日、愛知工科大学グラウンド、快晴の天気で行った。実験方法は、時間が 1 回あたり 10 分、回数は 6 回（ローバの設置場所を入れ替えて 3 回ずつ）、2 台のローバ間距離は約 20m 離して行った。実際の実験の様相を Fig. 4-3-2 に示す。XBee 2 台を約 20m の距離で離して置き、10 分間 GNSS のデータ取得を行った。片方のローバには XBee は搭載しておらず、もう片方には XBee を搭載し、ログデータを出力している。XBee を搭載したローバの近くで、ノート PC に受信用 XBee を接続し、実際にログデータを取得している。受信用 XBee は受信専用で、送信は行っていない。またノート PC の Wi-Fi や Bluetooth も OFF にし、さらに携帯電話も機内モードに設定し、電波を出力する機器からの影響を排除した。送信用及び受信用 XBee の各種設定はデフォルト設定を用いた。GNSS 本体には RTC やセルフ・エフェメリスのためにバッテリーバックアップする機能が搭載されているが、バッテリーを搭載せず実験を行った。Table 4-3-1 に示した通り、実験は 1 回あたり 10 分間 GNSS データの取得を行い、それを 6 回行った。6 回のうち最初に 3 回 XBee ありとなしのローバを約 20m 離して置いてデータを取得し、次に XBee ありとなしのローバの位置を入れ替えて同様に 3 回データを取得した。これはローバの位置による GNSS 測位の影響を排除するためのものである。実験では 2 台のローバの電源を同時に入れ、GNSS 測位データを取得した。

Table 4-3-1 Experimental Equipment

Microcontroller	Intel Genuino 101
XBee 送信機	XBee-PRO ZB S2B (Wire Ant)
XBee 受信機	XBee-PRO ZB S2B (RPSMA)
GNSS	Adafruit Ultimate GNSS Breakout v3

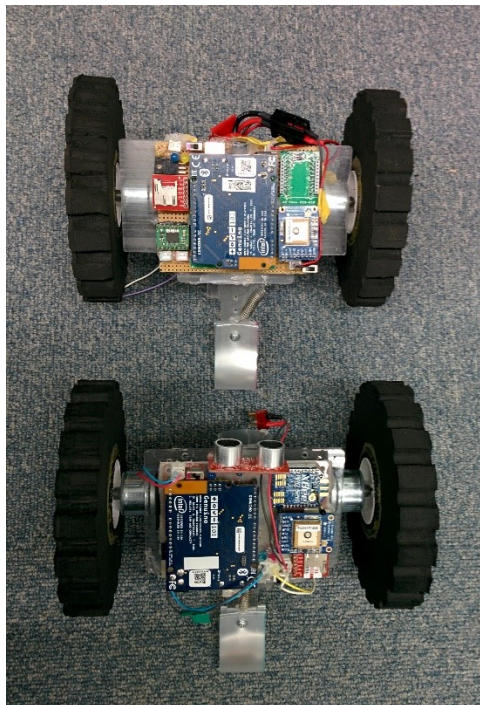


Fig.4-3-1 Rover for Using
Experiment

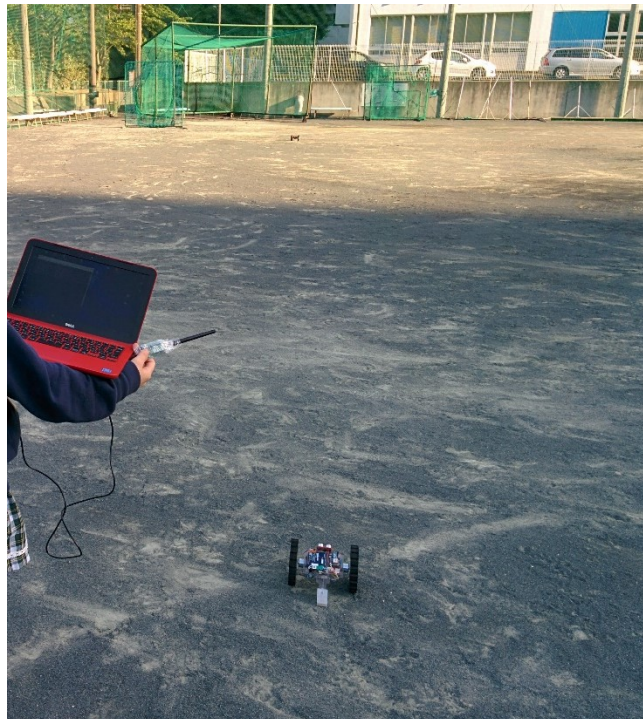


Fig. 4-3-2 Experimental Situation

4-4 GNSS 測位精度影響実験

Fig. 4-4-1 及び Fig. 4-4-2 にローバの電源を入れてからの GNSS の衛星測位数の変化を示す。XBee を搭載していないローバのデータである Fig. 4-4-1 では 100 秒以内に GNSS 衛星測位数が 10~11 個となっている。それに対して XBee を搭載しているローバのデータである Fig. 4-4-2 では、Fig. 4-4-1 と比べ、明らかに GNSS 衛星測位数の伸びが悪いことがわかる。100 秒を過ぎても最大 7~8 個しか衛星を捕捉できておらず、Fig. 4-4-1 よりもゆっくりと捕捉衛星数が上昇している。さらに、最大捕捉数も 8~10 個と、XBee なしと比べて低下している。次に GNSS 測位位置の変化をプロットした結果を Fig. 4-4-3 及び Fig. 4-4-4 に示す。Fig. 4-4-1 及び Fig. 4-4-2 より、衛星捕捉数が最初の 300 秒程度は安定していないため、300 秒から 600 秒の 6 回の実験の測位位置データをプロットした。緯度・経度の中心値は、それぞれのデータの中央値を使用した。XBee を搭載していないローバの GNSS 測位位置プロットデータである Fig. 4-4-3 では、緯度方向には中央値から -3m~0m の範囲内の誤差であり、経度方向では -1m~+2m の範囲内に誤差が収まっている。さらに、Frequency では、それぞれ 0m のデータが 1500 個近くあり、それ以外のデータは数十個しかなく、非常に高い精度で測位出来ていることが分かる。それに対して XBee を搭載しているローバの GNSS 測位位置プロットデータである Fig. 4-4-4 では、緯度方向に ±3m、経度方向に -5~+3m の広い範囲に位置情報が分布している。さらに、Frequency は XBee のない Fig. 4-4-3 よりもなだらかであり、GNSS 測位位置情報が Fig. 4-4-3 よりも大きな割合でブレていることが分かる。以上より、明らかに XBee から送出される電波は、GNSS の測位精度に影響を与えていることが分かった。

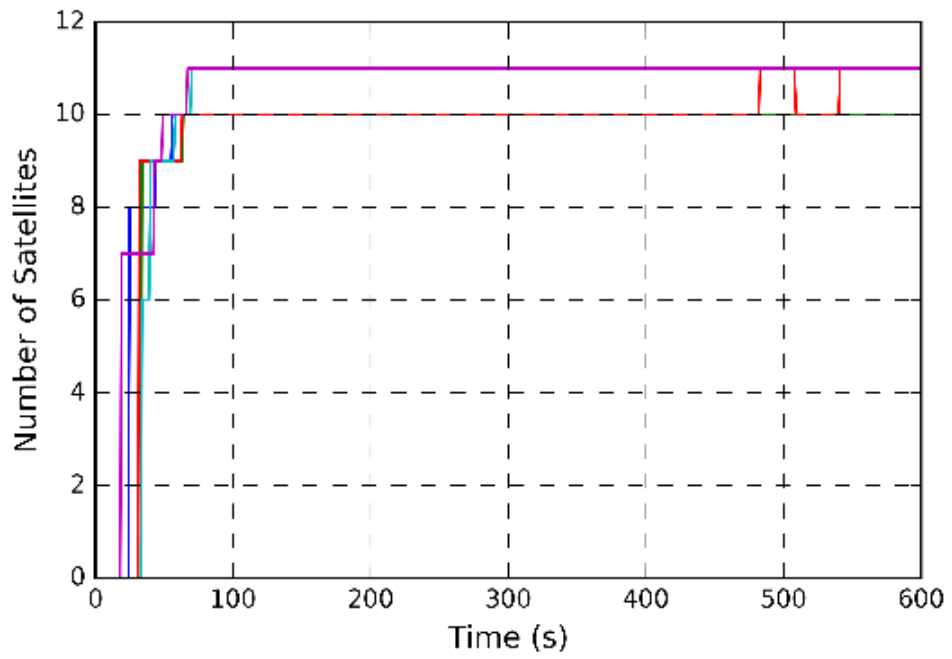


Fig. 4-4-1 Number of Satellites without XBee

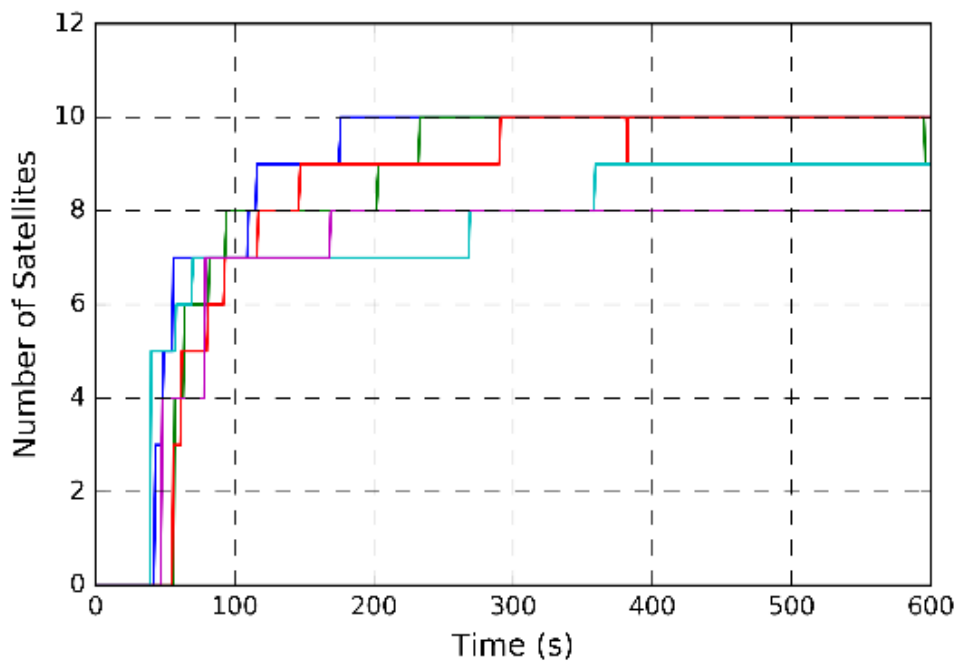


Fig. 4-4-2 Number of Satellites with XBee

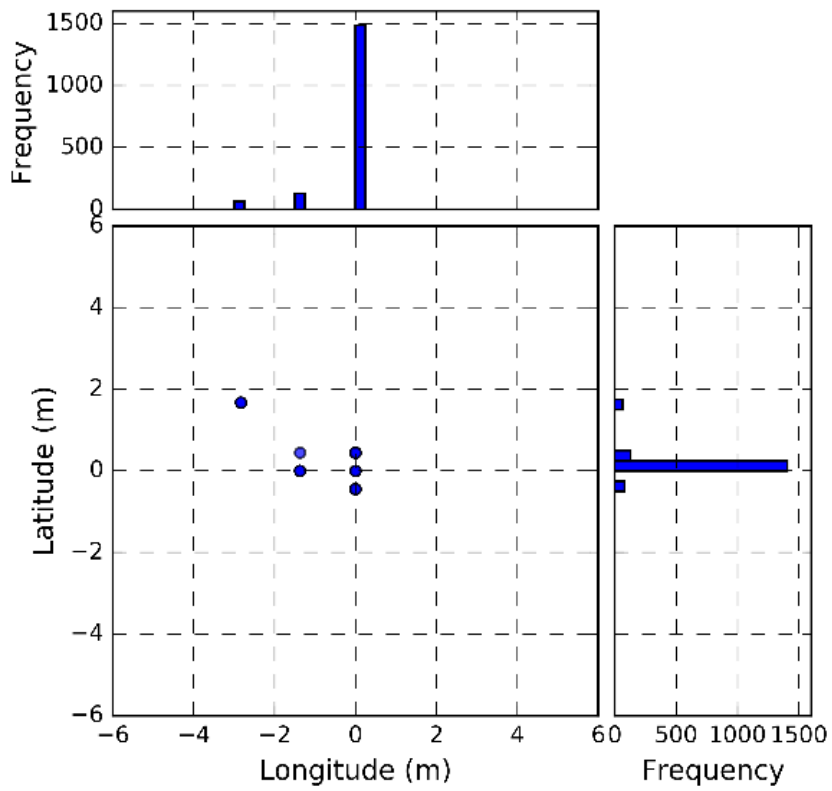


Fig. 4-4-3 GNSS Location Data without XBee

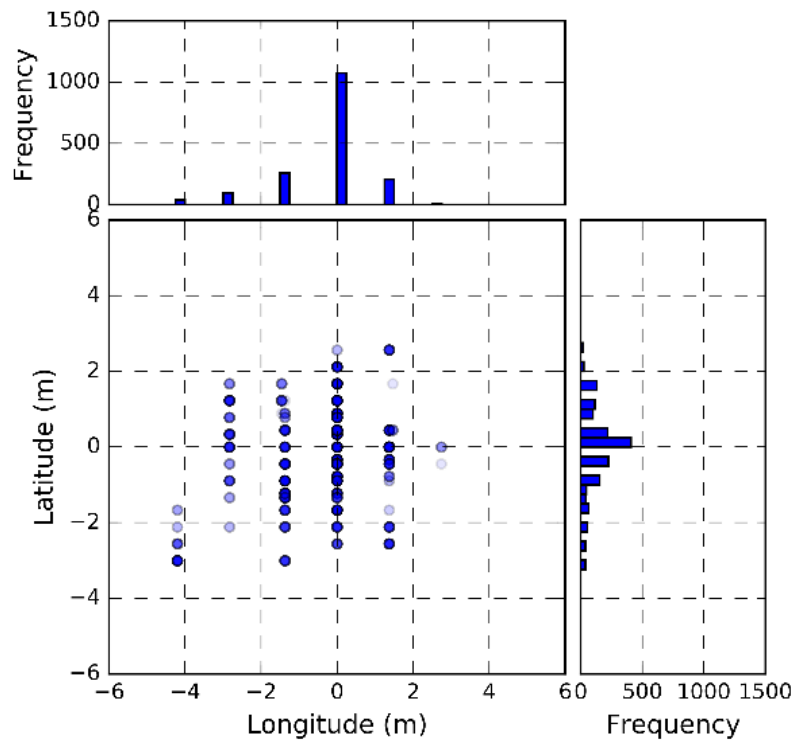


Fig. 4-4-4 GNSS Location Data with XBee

4-5 浮動小数点数精度による測位位置分解能

Fig. 4-4-3 及び Fig. 4-4-4 では GNSS の測位位置をプロットしているが、緯度方向と経度方向でプロットされている点の間隔が違っている。緯度方向は 2m で 5 個弱プロットがあるため、40cm 程度の分解能があるが、経度方向は 1.4m 程度の分解能しかない。この緯度・経度方向の測位分解能の違いについて調べた。GNSS の NMEA0184 フォーマットによる緯度・経度の分解能は 18.5 cm 以下であり、経度は緯度が 0 度から離れるほど、さらに分解能が高くなる。このことより、Fig. 4-4-3 及び Fig. 4-4-4 の経度の分解能が緯度よりも悪い原因は NMEA0184 フォーマットではないことが分かる。さらに、緯度の分解能も約 18.5 cm であるはずが、実際に取得したデータの分解能は約 40cm になっており、緯度の分解能も NMEA0184 フォーマットの分解能よりも悪い。以上より、この分解能の問題は Arduino の浮動小数点数演算の精度が原因ではないかと考えた。我々は NMEA0184 の解析に Adafruit Ultimate GNSS ライブラリを用いた。GNSS 解析ライブラリでは緯度、経度の値は float 変数を用いて計算している。Arduino Uno の float 変数は 32bit の浮動小数点数であるため、有効桁数は約 7 桁である。愛知工科大学の座標は北緯 34.78 度、東経 137.17 度である。整数部分が緯度では 2 桁、経度で 3 桁であるため、緯度と比較して経度では小数点以下の有効桁数が 1 桁少なくなる。このため緯度・経度を 32 bit float 変数で表現した場合、緯度の分解能は 0.00001 度以下 0.000001 度以上、経度の分解能は 0.0001 度以下 0.00001 度以上になる。このことから、緯度の誤差は 0.111 m 以上であり、経度の誤差は 0.911 m 以上である。よって、Arduino Uno では緯度方向で約 0.91m の誤差があることが分かった。

4-6 電波干渉による GNSS 測位精度

ローバに搭載している XBee が出力する電波が GNSS の測位精度の影響について実験により調べた。その結果、XBee が出力する電波は GNSS の測位衛星数及び測位精度に影響を与えることが分かった。さらに計算上 GNSS の緯度・経度は度(degree)単位に変換し、それにより計算しているが、Arduino Uno の浮動小数点数変数である 32bit の float 型変数では GNSS から出力される NMEA0184 を変換するには精度が不足しており、実際の計算結果から、緯度方向で約 40 cm, 経度方向では約 1.4 m の分解能しか示せないことが分かった。この精度ではローバの走行制御及びゴールに少しでも近づけるためには障害になる。GNSS の測位精度を向上させるため Xbee ではない電波干渉しない通信機の搭載をし、Arduino のような浮動小数点の問題のない Raspberry Pi などのマイコンで動作させることにより GNSS 測位精度を従来より向上させる。[21][22][23][24]

4-7 カムバックにおける 0m ゴール手法

カムバック部門では ARLISS2016 以前には GNSS 位置情報によるゴールを達成するチームは少なく、GNSS を用いたゴールができればカムバック競技では優勝記録となっていた。しかし ARLISS2016 以降では GNSS 測位誤差を考慮し GNSS 位置情報で接近後、より正確にゴールまで接近しゴールに接触した状態でゴール判定を行うことができる 0m ゴールを目指すチームが出てきた。ARLISS2016 以降にはカムバック部門の競技において 0m ゴールすることで優勝記録とする傾向となった。

0m ゴールを行うためには GNSS 位置情報のみでは GNSS 測位誤差の影響から極めて困難であり、ゴールへ GNSS 測位誤差の距離まで誘導制御後、ゴールのカラーコーンを画像認識することで 0m 距離まで誘導制御できないか検討を行った。CanSat においてゴールには赤色のカラーコーンが設置されるが、その際の赤色を画像認識する手法において、GNSS 測位誤差の問題を解決し、0m ゴールが可能であるか検討を行った。[25][26][27][28]

4-8 ゴール画像認識手法の検討

ゴール地点には赤色のパイロンが置かれており，それをカメラ画像認識する必要がある．また，サイズ及び重量の制約上，我々の CanSat に搭載可能な最高性能コンピュータは Raspberry Pi Zero である．画像認識手法は，このコンピュータでリアルタイムにゴールを認識できる計算量である必要がある．そこで我々はどの画像認識手法が有効であるか検討を行った．画像認識方法には，色範囲のバイナリ画像で認識する方法（以下，カラーバイナリと呼ぶ），テンプレートマッチング，SIFT/SURF，ディープラーニング画像分類がある．CanSat におけるゴール画像認識手法の定性的評価を Table 4-8-1 に示す．Table 4-8-1 より，Deep Learning は計算時間が長いため使用が困難である．CanSat のサイズ制約において搭載可能な最高性能コンピュータは Raspberry Pi Zero であるため，コンピュータの性能が低く Deep Learning を動作させながら他誘導動作の制御やセンシングのサンプリングの並列処理をすることが困難である．Template Matching や SIFT, SURF は計算時間的には問題ないが，ゴールに近づいた時，ゴールの一部分しかカメラ画像に映らず，その状態で認識することが難しい．特にゴール付近では画像全部がゴールポストの赤色のみになり，そうなるに認識できず，ゴール判定できない問題がある．このことから，我々は画像認識手法として Color Binary を用いることにした．さらに Color Binary では色範囲を指定する色空間として RGB, YCrCb, HSV の 3 種類がある．それぞれの色空間における定性的評価を Table 4-8-2 に示す．RGB は色と明るさを 3 次元で表しているため，明るさと色温度の変化に対応しづらいため，本用途には不向きである．YCrCb 及び HSV は明るさを Y, H パラメータで表現し，色を Cr, Cb または S, V の 2 次元で表現するため本用途に向いている．実験の結果 YCrCb の方が，天候変化による環境光の変化に強く，調整しやすかったため，我々は色空間には YCrCb を選択した．

Table 4-8-1 Qualitative evaluation of image recognition methods

	計算時間	精度	ゴール部分認識	距離
Color Binary	短い	高い	できる	わかる
Template Machining	短い	低い	できない	近いと分からない
SIFT, SURT	中	中	できない	近いと分からない
Deep Learning	長い	高い	できることもある	工夫により分かる

Table 4-8-2 Qualitative evaluation of color space

	色範囲の指定しやすさ	明るさの影響
RGB	3次元のためしにくい	大きい
YCrCb	色範囲は Cr,Cb の 2次元なのでしやすい	小さい
HSV	色は 1次元,S 入れて 2次元なのでしやすい	小さい

4-9 画像認識手法

Raspberry Pi カメラで解像度 160×120 ピクセルで画像を撮影し、それを RGB から YCrCb に変換する。次に Y, Cr, Cb の値がゴールコーンの赤色の範囲内かどうかの 2 値画像を生成する。2 値画像の重心座標によりゴールの存在する左右の方向を求め、ピクセル数によりゴールまでの距離を求める。

実験の結果、カメラがゴールまでの距離が 5m では、ゴールの 2 値画像のピクセル数は約 0.1%であった。同様に 4 m では 0.25%、10cm では 35%であった。この結果から、ゴールに置かれている赤色のパイロンの一部分しか認識できない場合を考慮し、2 値画像により認識された赤色が 0.08%以上のときに、ゴールを認識したと判断するようにした。

4-10 画像認識による CanSat の誘導制御手法

ゴールまで距離が遠い状態では GNSS 位置情報を用いて CanSat をゴール地点付近まで誘導する。GNSS 位置情報がゴールまで 1.4m 以内になったとき、画像認識によりゴールが認識可能なため、画像認識ゴール誘導へと切り替える。カメラでゴールが認識できないときは、CanSat は左に約 30 度旋回する。ゴールを認識した場合、認識した 2 値画像の重心座標をゴール方向とする。そして、その方向にステアリングを切りながら 2 秒間進み、停止する。この作業をゴール認識ピクセル数が 35%以上になるまで繰り返す。35%以上になったとき、CanSat はゴールまで距離が 0m の地点に来たと判定し停止する。ゴールがカメラで認識できなくなった時、CanSat は約 30 度左旋回して、再度カメラでゴールを認識する。カメラで再びゴールを認識できるまでこの作業を繰り返す。ただし、CanSat がゴールの見つからない場所にいる可能性があるため 7 回繰り返してゴールが見つからない場合、GNSS 誘導制御に切り替えて 40 秒間走行する。走行中にカメラがゴールを認識したら走行を停止し、上記ゴール誘導制御に移行する。40 秒間走行してもゴールが見つからなかった場合も、CanSat は走行を停止し、ゴール誘導制御に移行する。

4-11 種子島ロケットコンテストにおける評価

種子島ロケットコンテスト 2018 で実際にゴール画像誘導制御した結果について述べる。CanSat は GNSS 誘導制御により，落下地点から，ゴールまで距離が約 4m の地点まで走行し，停止した。そして CanSat はカメラ画像認識によるゴール地点への誘導制御に切り替えた。そして CanSat は Fig. 4-11-1 に示す画像でゴールを認識した。画像のシアン色がゴールとして認識した部分を表している。この時 CanSat はゴールからの距離が約 4m の地点であり，ゴールの赤色を認識したピクセル数は約 0.09% だった。ゴールは画像の中央に位置しているため，CanSat は 2 秒間直進して停止した。これにより，CanSat はゴール地点からの距離が約 2m の地点まで近づいた。そしてカメラにより Fig. 4-11-2 の画像を取得し，ゴールの赤色認識ピクセル数は約 0.8% となった。CanSat は画面右側にゴールを認識したので右旋回し，2 秒間前進して停止した。これにより CanSat はゴールから約 50cm の距離まで接近した。そしてカメラで Fig. 4-11-3 の画像を取得し，ゴールの赤色認識ピクセル数は約 23% になった。ゴールが画面左側に認識されたため，CanSat は左へ旋回し，2 秒間走行した。これにより CanSat はゴール地点までほぼ 0m の地点まで近づいた。そしてカメラで Fig. 4-11-4 の画像を取得し，ゴールの赤色認識ピクセル数は約 72% となり，ゴール判定して CanSat は停止した。このようにして，種子島ロケットコンテスト 2018 にて実際にゴールまでの距離 0m まで CanSat の誘導制御に成功し，コンテストで優勝した。実際に 0m ゴールした CanSat を Fig. 4-11-5，種子島ロケットコンテスト 2018 にて優勝した表彰を Fig. 4-11-6 に示す。



Fig. 4-11-1 Goal Image at 4m

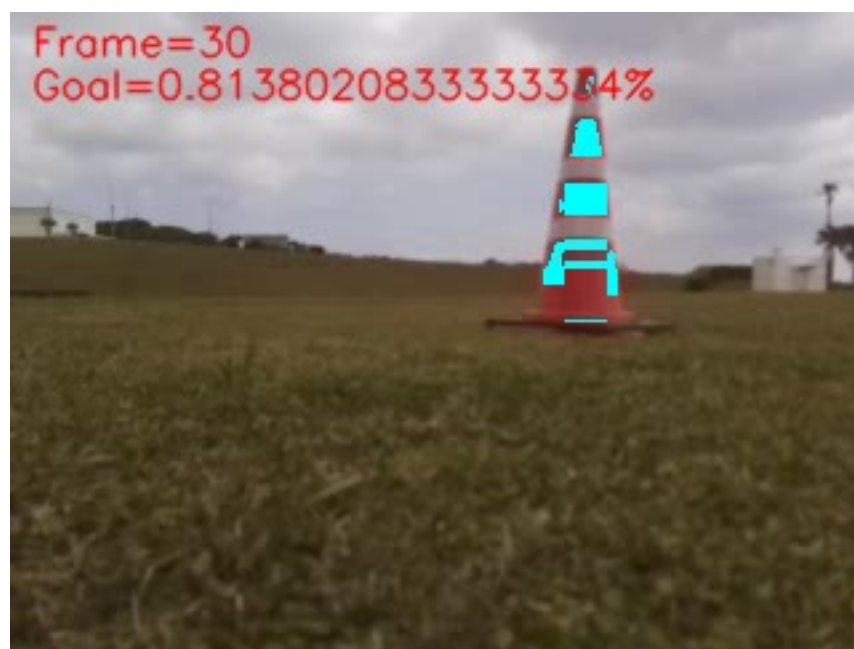


Fig. 4-11-2 Goal Image at 1~2m



Fig. 4-11-3 Goal Image at 50cm



Fig. 4-11-4 Goal Image at 0m



Fig. 4-11-5 Tanegashima Rocket Contest 0m goal



Fig. 4-11-6 Tanegashima Rocket Contest 2018 Championship Trophy

4-12 結論

我々は CanSat 競技における GNSS の測位誤差問題を解決するために、GNSS 測位精度を電波干渉する通信機器を特定し、マイクロコンピュータの Float 精度により GNSS 位置情報の誤差が大きくなる問題を特定した。またカメラ画像によるゴール認識及び誘導制御手法の提案を行い、コンテストで評価実験を行った。種子島ロケットコンテスト 2018 における実証実験により、0m ゴールを実現したことから、本手法が有効であることを実証した。しかし本手法では晴れ、曇り、朝、昼、夕方等の環境光の変化の対応が難しく、そして 5m 以上離れると認識できなくなる問題点がある。今後、Deep Learning 等の新しい画像認識手法で、計算時間とバランスを保ちながら、これら問題が解決する方法について、さらに検討を行う必要がある。

【参考文献】

- [21]秋山実穂,斎藤卓也,"惑星探査ローバの ARLISS2017 における動作ログの分析,"電子情報通信学会総合大会 2018, B-2-19,東京電機大学, Mar. 2018.
- [22]斎藤卓也,秋山実穂,"小型自律移動ロボットの設計と ARLISS2017 による実証評価,"ロボティクス・メカトロニクス講演会'18 講演論文集, 2A2-C07, 北九州国際コンベンションゾーン, Jun. 2018.
- [23]斎藤卓也,秋山実穂,"惑星探査ローバの機体設計及び ARLISS2017 における動作ログの詳細分析,"愛知工科大学紀要第 16 巻, pp. 17-26, Mar. 2019.
- [24]Takuya Saito, Miho Akiyama, " Development of Rover with ARLISS Requirements and the Examination of the Rate of Acceleration that Causes Damages During a Rocket Launch, " Journal of Robotics and Mechatronics, Vol. 31, No. 6, pp. 913-925, Dec. 2019
- [25]秋山実穂,斎藤卓也,"種子島ロケットコンテスト 2018 でゼロメートルゴールを達成した CanSat のログデータの分析", 情報処理学会第 48 回組込みシステム研究発表会, 論文番号 8, 東海大学, Jun. 2018.
- [26]斎藤卓也,秋山実穂,"種子島ロケットコンテストで優勝した惑星探査ローバの開発", 日本機械学会ロボティクス・メカトロニクス講演会'19, Jun. 2019.
- [27]秋山実穂,斎藤卓也,"0m ゴール誘導を実現する CanSat 向け画像認識及び制御手法の研究", 2019 年電子情報通信学会総合大会 ISS 特別企画「学生ポスターセッション」, Mar. 2019.
- [28]Miho Akiyama, Takuya Saito," A Novel CanSat-Based Implementation of the Guidance Control Mechanism Using Goal-Image Recognition," 2020 IEEE 9th Global Conference on Consumer Electronics (GCCE 2020), Oct. 2020.

Supporting Online Materials:

- [az] Digi International Inc, "Digi XBee Ecosystem – Everything you need to explore and create", <https://www.digi.com/xbee> [Accessed Feb 20, 2023]
- [ba] 種子島ロケットコンテスト実行委員会,"種子島ロケットコンテスト", <http://jaxa-rocket-contest.jp/> [Accessed Feb 20, 2023]
- [bb] Arduino, "Getting Started with the Arduino/Genuino 101", <https://www.arduino.cc/en/Guide/Arduino101> [Accessed Feb 20, 2023]
- [bc] Adafruit Industries, "Adafruit Industries, Unique & fun DIY electronics and kits", <https://www.adafruit.com/> [Accessed Feb 20, 2023]

第 5 章 Deep Learning 画像分類を用いたゴール認識制御

5-1 色の画像認識による日照条件の色変化

ARLISS2016 にて 3 チームが GNSS 位置情報でのゴール接近制御にてゴールの距離が約 1m 差の記録となり，GNSS 測位誤差で順位が決定した．より接近を行うためにはゴールのカラーコーンの赤色を画像認識する誘導制御手法を加え，GNSS 測位誤差を克服する手法により，種子島ロケットコンテスト 2018 では 0m ゴールを達成した．

しかし色を用いた画像認識には晴れや曇り，日陰や直射日光により，カラーコーンの色が変わり認識が困難になる問題があった．より正確なゴールへの誘導制御をする場合には，色を用いた画像認識の日照条件による色変化に対応した新たな認識手法の必要性が求められる．

5-2 従来の誘導制御手法

5-2-1 GNSS による誘導制御

従来ゴールまで誘導するための制御では GNSS 位置情報を用いて，ゴールの位置情報と CanSat の位置情報の角度差から進行方向を左右タイヤで調整し，ゴールへ向かう走行制御であった．制御周期は 1 秒となっており，GNSS 位置情報は 1Hz で取得し，CanSat の 1 秒前の位置情報と現在の位置から走行方向を計算し，走行ステアリング制御を行う．ステアリング制御は P 制御で行っており，ゴールから GNSS 誤差範囲内である距離数 m ほどまで誘導制御を行う．

5-2-2 色によるゴール画像認識手法

GNSS 位置情報の誤差範囲内まで誘導制御を行った後，色を用いた画像認識により 0m 地点まで誘導制御を行う．CanSat に搭載されているカメラで正面の写真を撮り，ゴール地点に設置されている赤色のパイロンの色を認識する．色を用いた画像認識では，赤色を抜き出すために，RGB 画像データを YCrCb に変換し，Y, Cr, Cb それぞれの範囲を指定し，2 地下画像を求めることにより，ゴールの赤色部分の認識を行っている．RGB による赤色のデー

タ範囲の指定の場合には、R,G,Bの3次元で色範囲を指定する必要があるため、原点と白点と特定の色とを通過する空間上の曲線とその周辺を指定する必要があり、計算や色指定が複雑になる。YCrCbではCrとCbの2次元で色範囲を指定可能となり、Yを含めても四角柱内部かどうかの指定で良く、明るさの影響を受けにくく簡単に特定の色範囲を指定可能であるためである。実際に色を用いた画像認識でのゴールへの誘導走行制御で、種子島ロケットコンテスト2018にてCanSat競技を行った結果、目標位置0m地点までの誘導に成功した。[24][25]。種子島ロケットコンテスト2018にてゴール画像認識を行った結果をFig.5-2-2-1～5-2-2-4に示す。図中のシアンで示した部分がゴールの赤色を認識した部分である。GNSSによりゴールまでの距離約5mまで接近し、カメラにゴールが写るまでCanSatを旋回させる。Fig.5-2-2-1では最初約4m地点で0.087%赤色を認識した。赤色の重心座標が中央であり、その後2秒間直進走行した。その後Fig.5-2-2-2で約1-2m地点にて0.81%でやや右側にゴールを認識し、右にステアリングを切り2秒走行。その後Fig.5-2-2-3で約50cm地点にて23.08%で左側に発見し2秒間左走行後、Fig.5-2-2-4にて0m地点で約72.61%認識し、ゴール判定である35%以上超えたため、ゴール判定として停止し0mゴールを達成した。種子島ロケットコンテスト2018にて0mの記録によりコンテストで優勝した。



Fig.5-2-2-1 Goal Image at 4m



Fig.5-2-2-2 Goal Image at 1~2m



Fig.5-2-2-3 Goal Image at 50cm



Fig.5-2-2-4 Goal Image at 0m

5-2-3 色によるゴール画像認識の問題点

種子島ロケットコンテスト 2018 にて 0m ゴール誘導に成功したが、色を用いた画像認識にはいくつか問題点があった。能代宇宙イベント 2018 でのゴールの色の認識に Y, Cr, Cb で調整し設定したパラメータ値で、異なる時間帯で認識した 2 つの画像を Fig. 5-2-3-1 に示す。左側の画像では曇天時において、ゴールを認識しつつも周りの草も誤認識していることがわかる。また右側では晴天時に日光に照らされ大きく色が変わり、ゴールをほとんど認識していないことがわかる。さらに Fig.5-2-2-1~5-2-2-4 での種子島ロケットコンテスト 2018 での色を用いたゴール認識画像でも、パイロンの赤色を全ての範囲で認識できていないことがわかる。色を用いた画像認識では、時間変化による日照条件の変化や角度によって光の反射や影の部分で大きく色が異なり、認識が困難になる問題があった。より確実なゴール認識を行うためには、このような日照条件などにおける色の変化に対応できるような手法を取り入れる必要があった。



Fig.5-2-3-1 Goal recognition problems by color

5-3 提案手法

5-3-1 Deep Learning によるゴール認識

従来のゴールの色を画像認識する手法では、日照条件などの光の変化により色が大きく変化し、ゴール認識が困難になる問題があった。この問題を解決する新たな手法として、あらゆる日照条件や光の順光や逆光のゴールの画像を全てビックデータでゴールと学習させる Deep Learning の画像分類手法を用いる手法を提案した。Deep Learning の画像認識では多数有用性が示されており、例えば VGG16 モデル[29]等を用いれば、非常に高い認識精度での画像分類が実現可能であることが分かっている。しかし VGG16 をはじめた複雑なモデルである Deep Learning での画像認識においては計算量が必要であり、Raspberry Pi などのマイクロコンピュータで動作する CanSat に適応するには長時間の認識時間やメモリ不足などで動作が困難である。Çalik らは、組み込みシステム向けの Convolutional Neural Network (CNN)を用いた CIFAR-10 画像分類手法について提案しており 85.9%の高い認識率得ているが、メモリは 2GB しか使っていないと述べている[30]。CanSat には Raspberry Pi Zero を利用しているが、メモリは 512MB しか搭載されておらず、メモリ使用量を減らしたモデルであっても、メモリ不足により動作が困難である。

そこで Keras の CIFAR-10 サンプルプログラムで使用されている小さいサイズのモデルで動作ができないか、検討を行った。Raspberry pi Zero で動作可能なメモリサイズであり、実用的な認識時間で計算可能であり、従来の色を用いた画像認識よりも認識精度が高い手法であるか調査した。

5-3-2 使用する Deep Learning モデル構造

本手法で使用する Deep Learning モデルの構造を Fig. 5-3-2-1 に示す。このモデルは Keras ライブラリ [bd] の CIFAR-10[be]用サンプルプログラム (cifar10_cnn.py) [bf]に用いられているモデルをベースに、今回の用途に合わせて出力層を 10 から 3 へと変更したものである。このモデルは Yann LeCun らの LeNet[31]モデルを CIFAR-10 に適応し、2 つの 2 次元 CNN をそれぞれさらにもう一つ 2 次元 CNN を重ねることにより、認識精度を向上させたものである。モデルの層構造の図式を Fig. 5-3-2-2 に示す。

入力画像は解像度 32×32 ピクセルの 3ch(RGB カラー)画像であり、最初 32×32 解像度の 2 次元 CNN(Conv2D)で 32 回の畳み込みを 2 重で行う。その後その後 Max Pooling により解像度を 16×16 に減少させ、ドロップアウト率 25%で次の 16×16 解像度の 2 次元 CNN で 64 回畳み込みを 2 重で行う。その後 Max Pooling により解像度を 8×8 に減少させ、ドロップアウト率 25%で、その 4096 個の出力を flatten によりベクトルに変換し、ドロップアウト率 50%で 512 ニューロンの全結合層へと入力する。最後に出力層に 3 ニューロンの全結合層を用いて、3 種類の画像分類を行う。

このモデルは VGG16 等の一般的な Deep Learning Image Classification モデルと比較し、解像度が非常に荒く単純な構造である。しかし CanSat におけるゴール認識では複雑な形状の物体ではなく、画像分類の種類もゴールが写っているか、ゴール判定が行えるか、ゴールが写っていないかの 3 種類で誘導制御ができるため、単純なモデルであっても従来よりも高い精度で認識が可能であるのではないかと考えた。

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 32, 32, 32)	896
conv2d_2 (Conv2D)	(None, 32, 32, 32)	9248
max_pooling2d_1 (MaxPooling2D)	(None, 16, 16, 32)	0
dropout_1 (Dropout)	(None, 16, 16, 32)	0
conv2d_3 (Conv2D)	(None, 16, 16, 64)	18496
conv2d_4 (Conv2D)	(None, 16, 16, 64)	36928
max_pooling2d_2 (MaxPooling2D)	(None, 8, 8, 64)	0
dropout_2 (Dropout)	(None, 8, 8, 64)	0
flatten_1 (Flatten)	(None, 4096)	0
dense_1 (Dense)	(None, 512)	2097664
dropout_3 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 3)	1539
Total params: 2,164,771		
Trainable params: 2,164,771		
Non-trainable params: 0		

Fig. 5-3-2-1 Structure of deep learning model

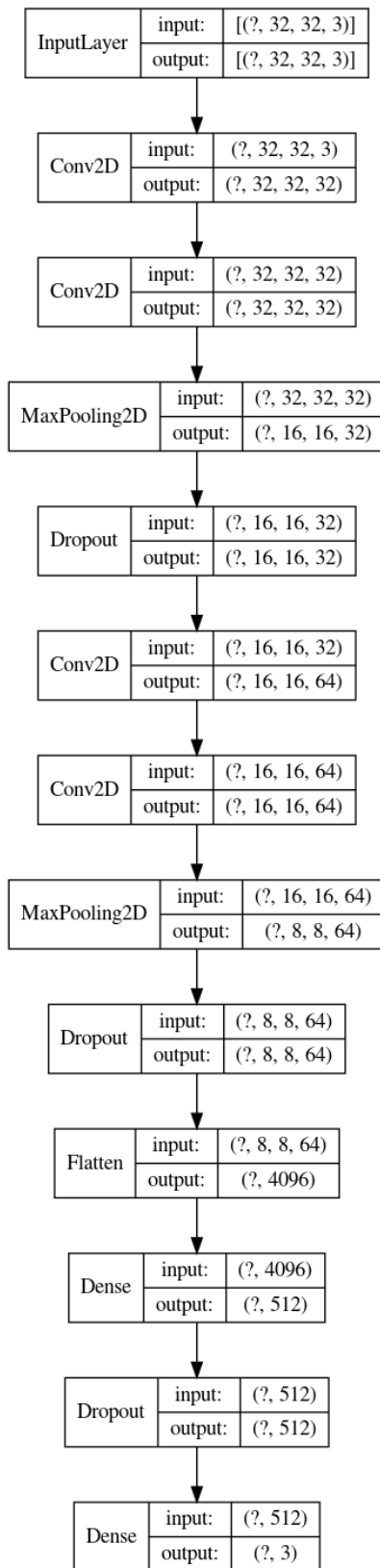


Fig. 5-3-2-2 Model layers

5-3-3 Deep Learning プログラム構造

Raspberry Pi Zero W の OS には Raspbian 9.4 Stretch を用いた。また Deep Learning のプログラミングには、Python 3.5.3 を使用し、Deep Learning ライブラリには Keras 2.2.4 を用いた。Keras のバックエンドには TensorFlow 1.8.0 を用いた。Keras は Python にしか対応しておらず、Keras を使う場合は必ず Python を使用する必要がある。一方、CanSat 制御用メインプログラムは高速化のため C++ 言語であり、処理速度の問題から Python で置き換えることはできない。このため、メインの制御用プログラムと Deep Learning 処理はプログラミング言語が異なるため、別プロセスとして動作させる。メイン制御用 C++ プロセスと Deep Learning 用 Python プロセスとの 2 言語のプロセス間通信には ZeroMQ[32] を用いた。Python の Deep Learning 画像認識のプロセスは ZeroMQ サーバとして動作させ、メインの C++ プログラムから画像データが送られてきたら Deep Learning 認識処理を行い、認識結果を返す構造とした。

5-3-4 ゴール画像の学習

Deep Learning 画像分類によりモデルにゴール画像を学習させる種類として、ゴール画像、ゴール判定画像、その他の画像の 3 パターンで学習を行った。Fig.5-3-5-1 にゴール画像としてゴールが認識できる画像を示す。あらゆる日照条件や距離から撮影したゴール画像を約 1,500 枚学習させた。Fig.5-3-5-2 にゴール判定としてゴールの 0m 地点に到達したと判定できる画像として約 240 枚学習させた。Fig.5-3-5-3 にその他の画像として、ゴールが写っていない画像を約 1,400 枚学習させた。



Fig.5-3-4-1 Goal recognition image



Fig.5-3-4-2 Goal judgement image



Fig.5-3-4-3 No goal image

5-3-5 ゴールの位置認識手法

CanSat をゴールまで誘導走行させるには、画像に写っているゴールの位置を認識し、左右走行方向を調整しながら誘導していく必要がある。Deep Learning Image Classification は画像全体を分類する手法であるため、ゴールが左右どちらにあるかを認識することができない。そこで画像の横方向に左、中央、右へ Region of Interest (ROI) により分割し、左右走行方向を判断する手法を用いた。最初画像全体をゴール画像、ゴール判定画像、ゴール無し画像として 3 パターンに分類し、ゴール判定であれば停止する。Fig.5-3-5-1 にゴール判定を行った画像を示す、その後ゴール判定でなければ、ROI により横方向を 3 分割し左右のゴール位置を認識し、発見すればその方向へ走行し誘導走行を行う。Deep Learning 画像分類では入力画像の解像度が 32×32 ピクセルであり、ROI も 32×32 ピクセルに解像度をリサイズしている。3 分割で発見できない場合、より長距離に小さくゴールが写っている可能性があるため、さらに 5 分割に分割数を増加させ認識する。ゴールまでの距離 8m で認識された 5 分割時の認識結果例を Fig.5-3-5-3 に示す。5 分割で認識できなかった場合はゴールが写っていないと判断し、ゴールが発見できるまで巡回走行を行う。また ROI を 3 分割及び 5 分割する場合には、ROI が 1/4 ほど重なるように設定し、各 ROI の重なり部分にゴールがある場合にも認識できるようになっている。



Fig.5-3-5-1 goal judgement



Fig.5-3-5-2 Goal position 3 divisions (distance to goal: 3 m)



Fig.5-3-5-3 Goal position 5 divisions (distance to goal 8 m)



Fig.5-3-5-4 No goal

5-4 評価

5-4-1 学習結果の評価

Deep Learning 画像分類による学習結果を Fig.5-4-1-1 及び Fig.5-4-1-2 に示す。学習回数は 15 epoch であり、これ以上の学習回数では過学習となる傾向となったため、15 epoch の学習回数とした。Fig.5-4-1-1 は認識精度であり、accuracy は学習に使用した画像での認識率であり、val_acc は学習に使用していない画像での認識率を示している。Fig.5-4-1-2 は損失を示しており、loss は学習に使用した画像での損失となっており、val_loss は学習に使用していない画像での損失を示している。認識精度は 15 epoch の学習後、学習に使用した画像による accuracy の認識率は約 97%であり、学習に使用していない画像による val_acc の認識率は約 99%となり、非常に高い認識率を示した。

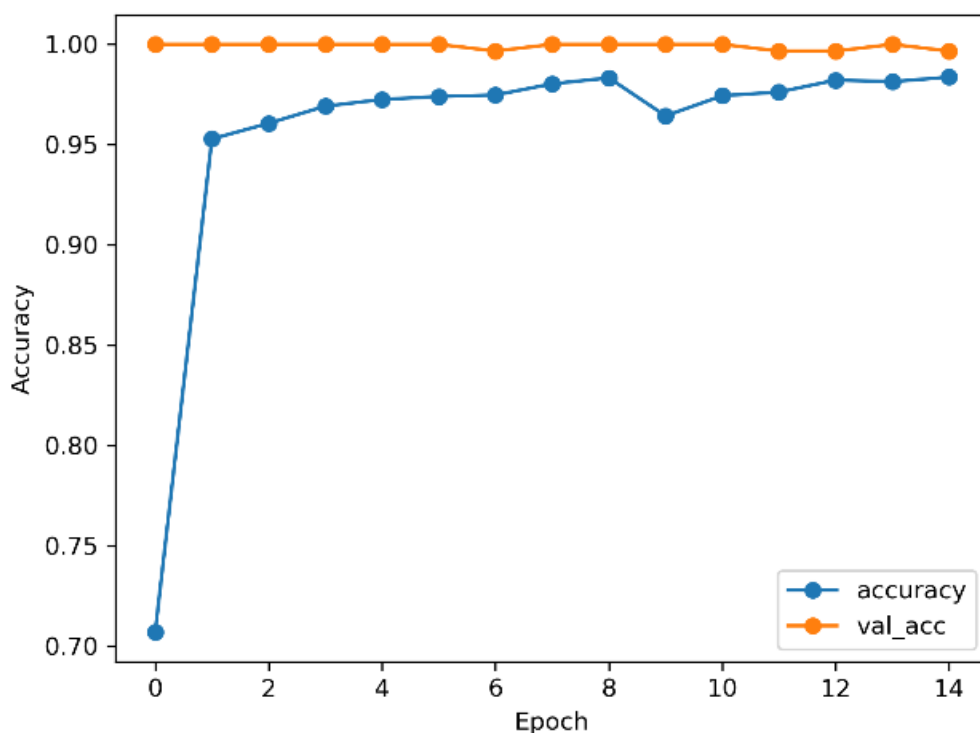


Fig.5-4-1-1 Model Training Results (accuracy)

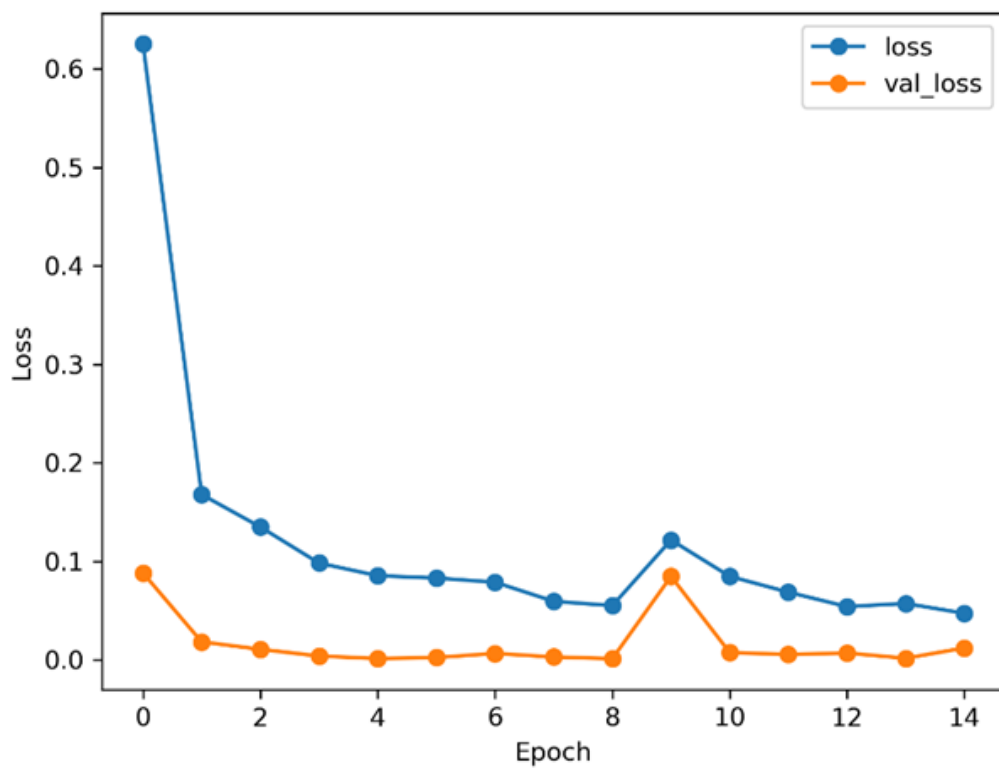


Fig.5-4-1-2 Model Training Results (loss)

5-4-2 Deep Learning による認識計算時間

Deep Learning 画像分類による認識での計算時間の測定を行った。Fig.5-4-2-1 は画像全体を 1 回認識した時の計算時間のヒストグラムである。1 回の認識では約 0.28s で計算できることを確認した。Fig.5-4-2-2 は ROI により 3 分割まで認識したときの計算時間のヒストグラムであり、3 分割をする際には最初に全体画像の認識後に 3 分割の認識を行うため、合計 4 回の Deep Learning 画像分類を行っている。3 分割における計算時間では約 1.2s で認識ができることを確認した。Fig.5-4-2-3 では ROI を 5 分割で認識したときの計算時間のヒストグラムであり、3 分割後に 5 分割を行うため合計 9 分割の Deep Learning 画像分類を行っている。計算結果はばらつきがあるが、全体は約 2.7s 以下で認識が可能であることが確認できた。本手法において最大でも 2.7s で認識ができることが確認できた。

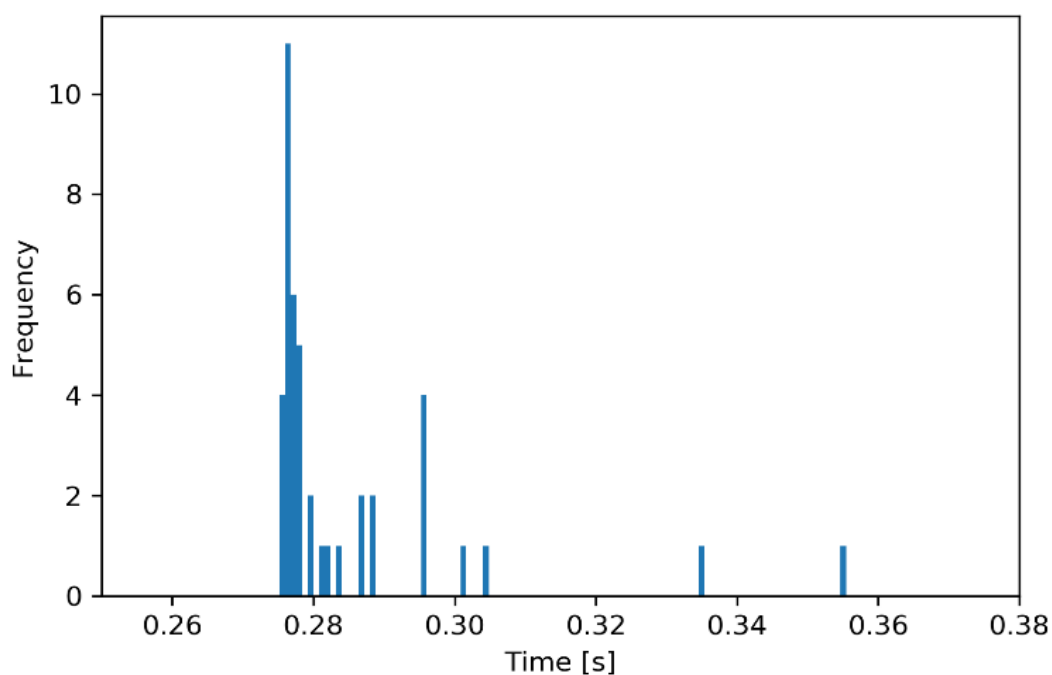


Fig.5-4-2-1 Computation time for one recognition

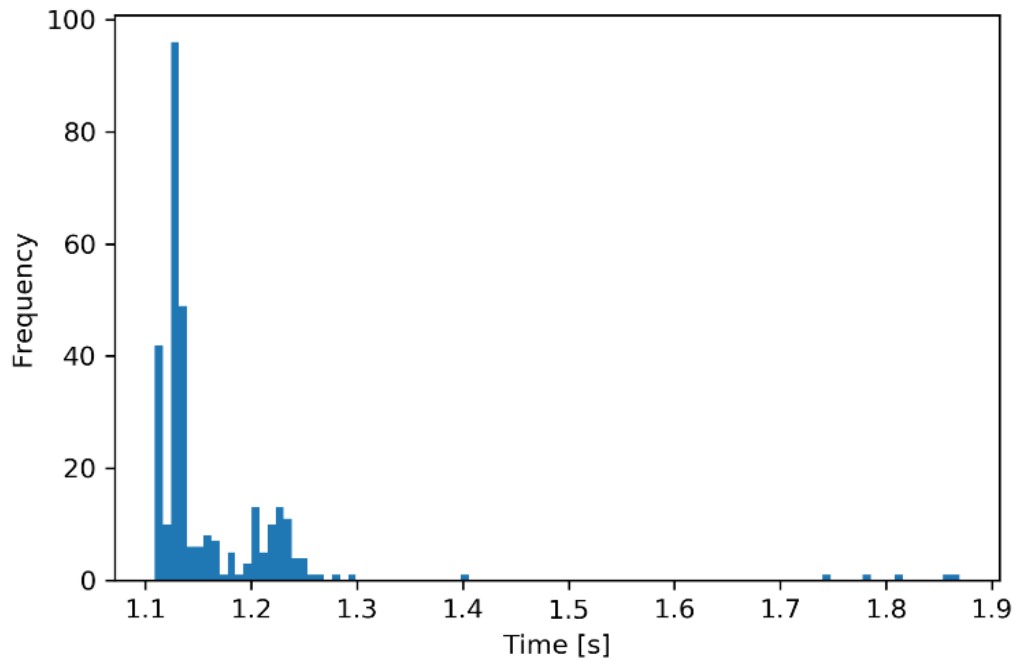


Fig.5-4-2-2 Computation time for 3 division recognition

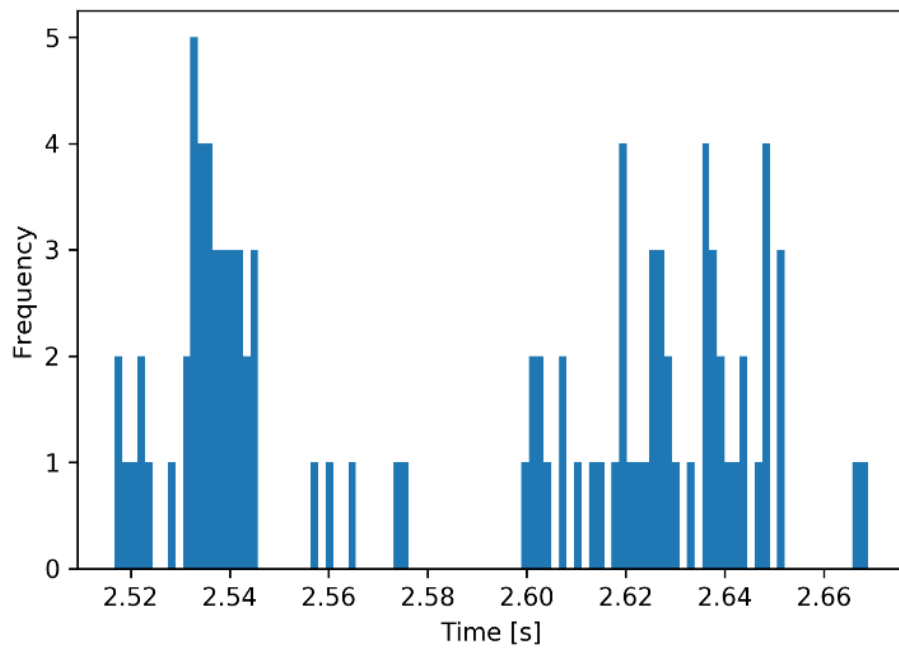


Fig.5-4-2-3 Computation time for 5 division recognition

5-4-3 メモリ使用量

メモリ使用量について、CanSat 搭載のマイクロコンピュータは Raspberry Pi Zero であるが、搭載メモリは 512MB であり、このメモリ内で動作可能であるか調査を行った。カメラの Raspberry Pi Camera v2 を使用する際メモリに最低 128MB を設定する必要がある、実際の使用可能なメモリは 384MB になる。Deep Learning 画像分類には通常多くのメモリが必要となるが、このメモリ容量において動作可能であるか調査を行う。Deep Learning サーバを動作させ、ゴール認識中のメモリ使用量を ps コマンドにより調べた。その結果、RSS 値は約 130MB であった。また、使用可能なメモリ量 384MB 内で、OS と Deep Learning サーバが使用した状態で、残りの使用可能メモリ量を調べるため free コマンドで調べた結果、available 値は 122MB であった。そのため Raspbian OS 上で CIFAR-10 モデルの Deep Learning サーバ使用時には 122MB のメモリが使用可能であることが確認できた。CanSat のメインプログラムの使用量は ps コマンドの RSS 値で約 25MB であり、122MB 内であることから、Raspberry Pi Zero の 512MB 内で CanSat のメインプログラムの動作などを含めても、Deep Learning 画像分類のメモリ使用量に問題なく動作できることを確認した。

5-4-4 ゴール認識率評価

Deep Learning 画像分類と従来手法ゴールの認識率の比較を行った。ゴールからの距離 0-10m を 1m ずつ、各距離でゴールを 45 度ずつ回転し、各地点でゴールが画像左, 中央, 右に写るように撮影を行った画像データを用いて、ゴールからの距離ごとにおける認識率を調べた。また画像データは直射日光が当たっている晴天時に撮影した画像と、日陰で撮影された曇天時に撮影された画像データの 2 種類ごとに認識率の比較を行った。ゴール画像データの撮影条件を Table 5-4-4 に示す。画像データは実際に CanSat を用いて Raspberry Pi Camera V2 で撮影を行った。通常撮影された画像データは解像度が高くなるが、実際の CanSat ではカメラから画像取得後解像度 320×240 pixels にリサイズしており、同様に画像データ読み込み後 320×240 pixels にリサイズし認識を行う。直射日光時におけるゴールの認識率を Fig.5-4-4-1 に、日陰時におけるゴールの認識率を Fig.5-4-4-2 に示す。それぞれの凡例で Deep Learning 画像分類における認識率と、従来の色を用いた CanSat 競技で実際に使用した様々な色範囲の指定値とで認識率の比較を行った。従来手法の凡例では種子島ロケットコンテスト 2018, 能代宇宙イベント 2017 の 1 回目, 2 回目, 及び ARLISS 2017 の各コンテスト本番で使用した YCrCb の調整値で認識させた認識率を示している。直射日光時におけるゴール認識率では、Deep Learning 画像分類では 0-6m 地点まで 100% の認識率を示し、8m 地点においては 90% 以上の認識率と、従来手法と比較し高い認識率であることが分かった。また長距離の 9m 地点では約 55%, 10m 地点では約 50% と認識率は低下するが、従来手法よりも高い精度での認識率となっている。従来手法と比較し、全ての距離において Deep Learning 画像分類での認識は最も高い認識率を示すことが確認できた。日陰時におけるゴールの認識率では 6m 地点から認識率は低下するが、7m 地点で約 90% の認識率であり、8m 地点で 50% の認識率となっており、従来手法では 8m 地点で全てのパラメータにおいて 0% であるのに対して、日陰状態におけるゴール認識率も従来より高くなることが確認できた。提案手法である Deep Learning 画像分類では日照条件が直射日光状態や日陰状態であっても、従来の色を用いた画像認識より高い認識率を示すことが確認できた。

Table 5-4-4 Goal image shooting conditions

Place	Nishiura coast	Ground at Aichi University of Technology
Weather	Sunny	Sunny
Lighting conditions	Direct sunlight	Shade of the building
Time	11:00 – 16:00	16:00 – 18:00
Date	August 26, 2020	August 13, 2018
Resolution	1280 x 960 pixels	640 x 480 pixels
Camera	Raspberry Pi Camera V2	
Goal pylon	Red color standard size with white line	
Image shooting distance	0m, 0.5m, 1m, 2m, 3m, 4m, 5m, 6m, 7m, 8m, 9m, 10m	
Image shooting direction	8 directions by shifting by 45 degrees	
Number of images	Each angle goal image position left, middle, right 3 images in total 24 images. 288 images at all distances.	

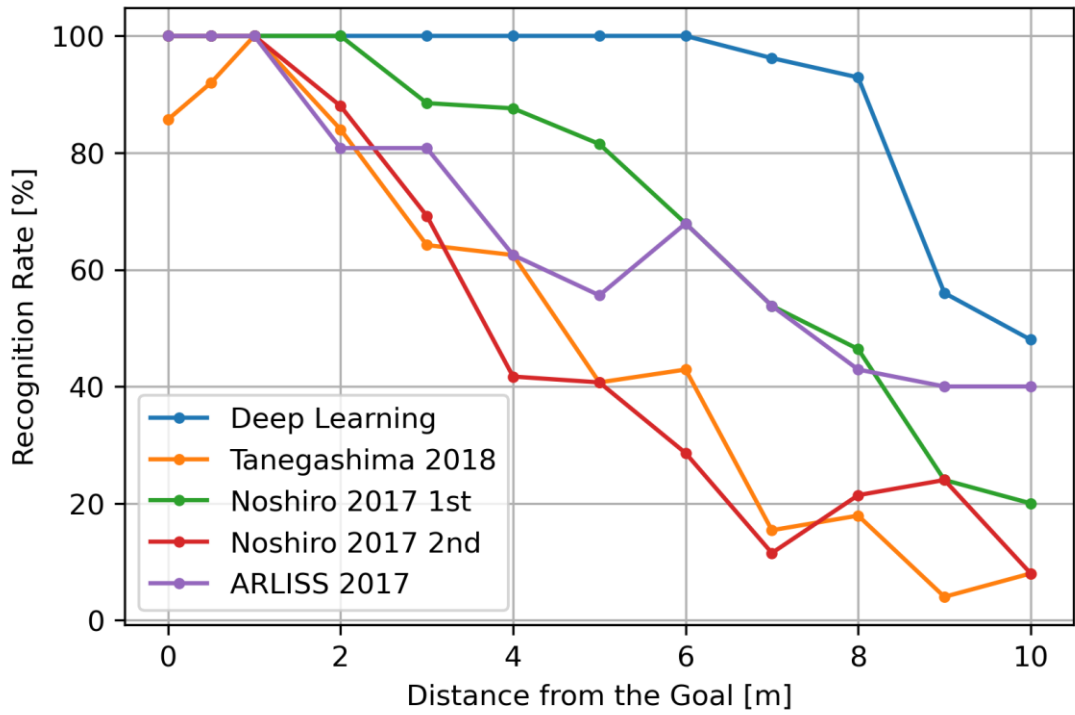


Fig.5-4-4-1 Recognition rate at a distance from the goal (on a sunny day)

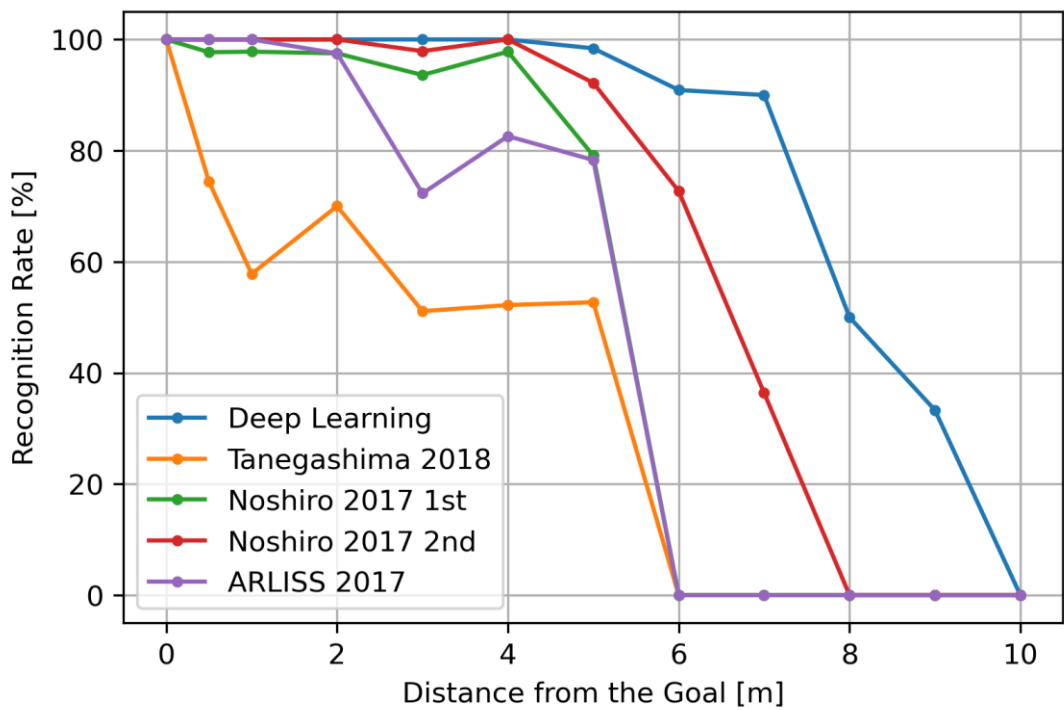


Fig.5-4-4-2 Recognition rate at a distance from the goal (on a cloudy day)

5-4-5 ゴール判定認識率評価

CanSat を 0m 地点まで誘導後、カメラに写るゴールは画像全体に写り、ゴールが画像全体に写る場合には確実にゴール判定し停止する必要がある。このようなゴール判定画像における認識率を従来手法と比較したグラフを、直射日光時を Fig.5-4-5-1 に、日陰時を Fig.5-4-5-2 に示す。

0m 地点では確実にゴール判定する必要がある、理想は 0m 地点では認識率 100%となる必要がある。直射日光時と日陰時において、Deep Learning 画像分類での認識率は 100%となった。従来手法では能代宇宙イベント 2017 の 2 回目ではほぼ 100%を示しているが、その他のパラメータでは認識率は低下しており、ARLISS2017 においては 40%と低い認識率となっている。認識率 40%では 0m 地点まで誘導しても、ゴール判定することができず停止できないことを示している。

ゴールからの距離 0.5m 地点では、この位置でゴール判定を行うと 0m 地点より手前で停止してしまうため、認識率は 0%であることが理想である。従来手法では全体的に約 50-90%でゴール判定の認識率となっており、ゴールの手前でゴール判定により停止してしまう可能性が高い。Deep Learning 画像分類においては、直射日光時と日陰時共にゴール判定認識率は約 20%にとどまっております。従来手法と比較しゴール手前で停止してしまう可能性が低いことがわかった。また距離 0.5m 地点で ARLISS 2017 及び Tanegashima 2018 では Deep Learning 画像分類と同等の認識率となっているが、0m 地点ではそれぞれ約 40%、65%と低くゴール判定の認識率が低くなっており、0m 地点で確実に停止することができない可能性がある。

0m 地点と 0.5m 地点の認識率の比較から、従来手法よりも Deep Learning 画像分類のゴール判定の認識率がより確実に判定ができることが確認できた。

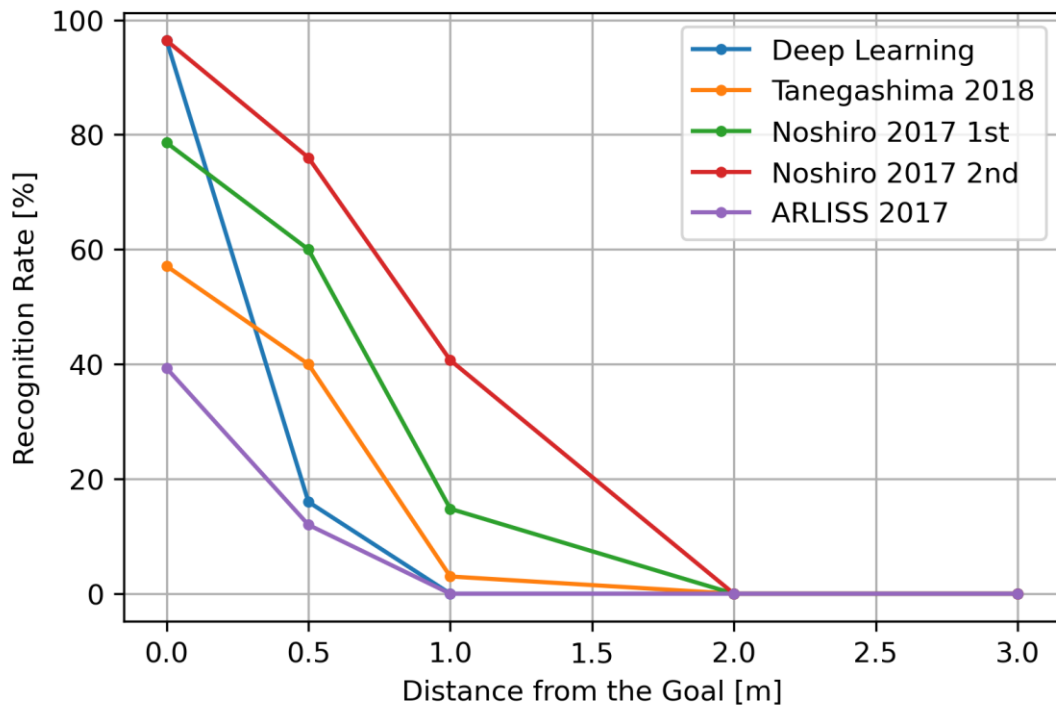


Fig.5-4-5-1 Goal judgment recognition rate at a distance from the goal
(on a sunny day)

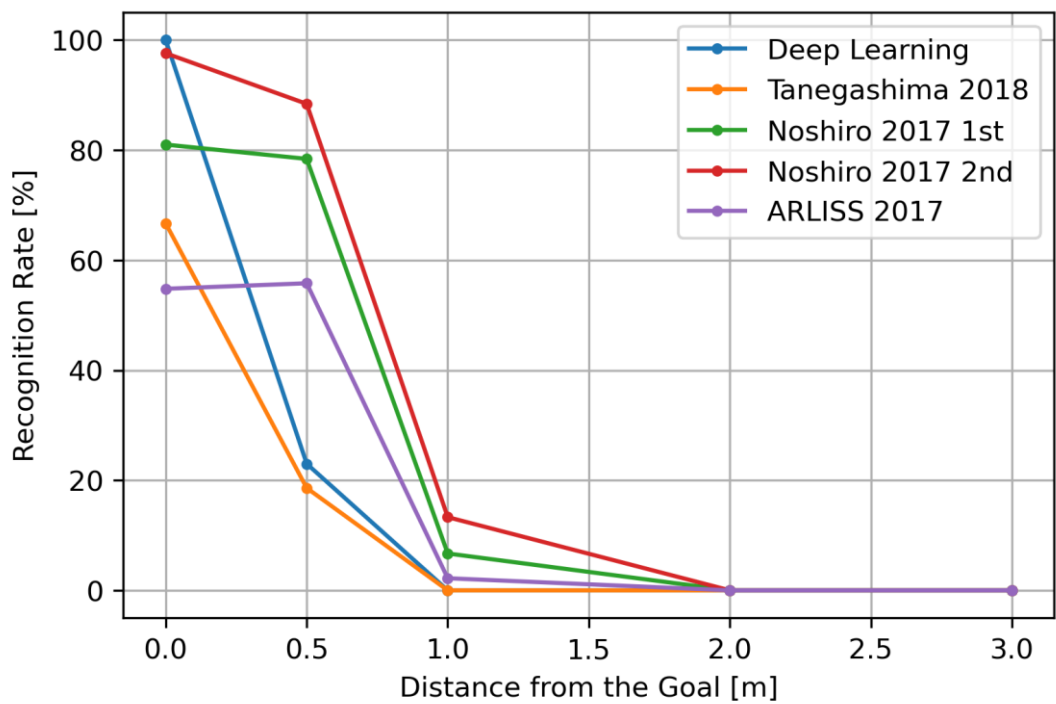


Fig.5-4-5-2 Goal judgment recognition rate at a distance from the goal
(on a cloudy day)

5-4-6 ARLISS 2019 における実証評価実験

本手法を採用した CanSat を実際に ARLISS2019 CanSat 競技において、距離 0m まで誘導可能であるか実証評価実験を行った。ARLISS2019 では日本から 18 チーム、海外から 13 チームの計 31 チームが参加し、我々はチーム名 SuperNOVA チームとして参加した。ARLISS ではロケットの打ち上げによる競技を各チーム 2 回行う。3 回以降も打ち上げは可能であるが、3 回目は競技の記録として認定されず、のデータ取得などの実験としての打ち上げとなり参考記録となる。我々は ARLISS2019 において 3 回の打ち上げを行った。ARLISS 2019 の結果[bg]の到達距離をまとめた表を Table 5-4-6 に示す。

1 回目はゴールから約 3.7km 離れた地点に落下し、GNSS 位置情報からゴール付近数 m まで走行した。ゴールには遠方から目視で発見するため、大きな旗が垂れ下がっているが、ゴールへ誘導走行時その旗を巻き込み 0.05m での停止となった。1 回目の結果の写真を Fig. 5-4-6-1 に示す。2 回目はゴールから 970m の地点に落下し、GNSS 位置情報からゴール付近数 m まで誘導後、Deep Learning 画像分類で 0m まで誘導に成功し、0m ゴールを達成した。2 回目の結果の写真を Fig. 5-4-6-2 に示す。3 回目はゴールから約 1.7km 地点に落下し GNSS 位置情報でゴール付近数 m まで誘導後、Deep Learning 画像分類で 0m まで誘導に成功し、0m ゴールを達成した。3 回目の結果の写真を Fig. 5-4-6-2 に示す。到達距離 0m の記録により、SuperNOVA は (a)-(d)の Award を受賞した。

- (a) Accuracy Award 1st
- (b) Best Mission Award 1st
- (c) Technical System Award 2nd
- (d) Overall Winner 1st

Accuracy Award は CanSat が目標位置であるゴールまでどれだけ到達できたかを競っており、0m 記録であるため、優勝記録となった。Best Mission Award では CanSat におけるミッションの難易度や新規性、達成度などを評価し、CanSat において初めて Deep Learning 画像分類を適応し 0m 記録を複数達成

したことから、優勝成績を得た。Technical System Award では CanSat における技術レベルを評価し、技術を重点においた新規性や難易度、達成度を評価する。こちらも Deep Learning 画像分類の精度や、複数回の 0m 記録が評価され準優勝の成績を得た。Overall Winner では全体優勝の順位となっており、Accuracy Award や Best Mission Award, Technical System Award の評価から、全体優勝の成績を得た。優勝したトロフィー及び表彰を Fig.5-4-6-4 に示す。本手法により、ARLISS における CanSat 競技においてゴールまでの距離 0m の誘導制御を実現し、本手法の有効性を確認した。

Table 5-4-6 ARLISS 2019 Result

Number of Launch	1 st	2 nd	3 rd
Dropped Position	3.7km	970m	1.7km
Result	0.05m	0m	0m



Fig. 5-4-6-1 1st goal



Fig. 5-4-6-2 2nd goal



Fig. 5-4-6-3 3rd goal



Fig.5-4-6-4 ARLISS2019 Overall Winner 1st, Best Mission 1st, Accuracy 1st, Technical System 2nd

5-5 結論

ARLISS などの CanSat における Comeback Competition で、目標位置まで 0m 距離に誘導制御を行う手法では、GNSS 位置情報では位置情報誤差があり困難であるため、従来は色を用いた画像認識で目標位置に設置されたパイロンであるゴールを認識することにより 0m 距離までの誘導を行っていた。従来手法の色を用いた画像認識ではゴールが日照条件や時間変化による明るさの変化が大きく、認識が困難になる問題があった。提案手法として Deep Learning 画像分類で様々な日照条件の画像を学習させることで、従来手法よりも認識率の向上ができないか調査を行った。しかし CanSat ではサイズ制約により小型組み込みコンピュータである Raspberry Pi Zero を利用しており、色を用いた画像認識では動作が可能であったが、メモリの小ささから複雑なモデルは動作することができない。そこで単純なモデルである CIFAR-10 用モデルを用いて、動作可能なメモリサイズであるか、認識時間は問題がないか、従来手法よりも単純なモデルであっても認識率の向上ができるか調査を行った。その結果、Deep Learning 画像分類と CanSat 制御に必要なメモリ使用量を測定し十分に動作可能であることを確認し、認識時間は 2.66s 以内にとどまることがわかり、従来と比較しゴール認識率は向上することが確認できた。また認識率は各距離ずつ直射日光時と日陰時の日照条件でどちらの条件においても認識率の向上が確認でき、ゴール判定を行う 0m 地点付近においても従来と比較し理想的な認識率であることが確認できた。提案手法を適応した CanSat で ARLISS2019 の CanSat 競技に参加し実証評価実験を行った結果、実際に 1 回目は 0.05m, 2 回目は 0m, 3 回目は 0m までの誘導を実現し Award において 3 部門で優勝, 1 部門準優勝の成績を得られ、本手法の有効性を確認した。[33][34][35]

【参考文献】

- [29] Simonyan, Karen, and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556 2014.
- [30] Çalik, Rasim Caner, and M. Fatih Demirci. "Cifar-10 image classification with convolutional neural networks for embedded systems," 2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA). IEEE, 2018.
- [31] Y. LeCun, Y. Bengio, "Convolutional Networks for Images, Speech, and Time-Series," The Handbook of Brain Theory and Neural Networks, vol. 3361, 1995
- [32] Sústrik, Martin. "ZeroMQ," Introduction Amy Brown and Greg Wilson 2015.
- [33] 秋山実穂, 斎藤卓也, "コンテスト用惑星探査ローバにおけるディープラーニングを用いたゴール認識の研究", 情報処理学会第 81 回全国大会, Mar. 2019.
- [34] Miho Akiyama, Takuya Saito, "Goal Recognition Using Deep Learning in a Planetary Exploration Rover Developed for a Contest," IEEE International Conference on Consumer Electronics - Taiwan (IEEE 2020 ICCE-TW), Sep. 2020.
- [35] 秋山実穂, 斎藤卓也, " Deep Learning を用いた CanSat をゴールまで距離 0m へ誘導する手法の研究, "電子情報通信学会論文誌 D J104-D(7), pp.540-551, July 2021.

Supporting Online Materials:

- [bd] "Keras Simple. Flexible. Powerful.," <https://keras.io/> [Accessed Feb 20, 2023]
- [be] Alex Krizhevsky, "The CIFAR-10 dataset," <https://www.cs.toronto.edu/~kriz/cifar.html> [Accessed Feb 20, 2023]
- [bf] "Keras: Deep Learning for humans," <https://github.com/keras-team/keras> [Accessed Feb 20, 2023]
- [bg] Ryosuke Koyanagi, "ARLISS 2019," <http://unisec.jp/archives/3474#more-3474> [Accessed Feb 20, 2023]

第 6 章 10m 長距離からの Deep Learning を用いた認識手法

6-1 長距離における Deep Learning を利用したゴール画像認識

Deep Learning 画像分類を用いたゴール認識は，天候による照明条件の変化に強く，無調整で高精度にゴールを発見できることから，CanSat においても非常に有効である．我々は Deep Learning 画像分類を用いてゴールを認識させる手法を提案し，そして ARLISS 2019 にて 1 回目 0.05 m，2 回目 0 m，3 回目 0 m を達成し，Accuracy Award 1 位，Best Mission Award 1 位，Technical System Award 2 位を獲得し，Overall Award で 1 位となり総合優勝を獲得した[35]．このときの CanSat のゴールの様子を Fig. 6-1 に示す．



Fig. 6-1 CanSat reached zero distance to the goal

我々は Deep Learning を用いて画像分類を行っているため，そのままではゴールの位置がカメラ画像のどこにあるのかがわからない．そこで水平方向に画像全体の部分領域である Region of Interest (ROI) を 3 分割，5 分割として複数回画像認識を行うことにより，ゴールの位置を求めている．この手法では，ゴールから約 6 m までが問題なく認識できる限界であり，それを超えると急速に認識率が低下し，ゴールを見つけられなくなっていく．特に GNSS の測位誤差に加えて強風により CanSat が流される等，いくつかの問題が重なると，ゴ

ール地点から最悪の場合 10m 近く離れた地点からゴールを認識する必要があることがある。しかしそのような場合はゴールの認識ができず、ゴール付近をさまよい続けることになる。

そこで、我々は従来の ROI 5 分割ではゴールが見つからない時に、さらに ROI 分割数を増やして認識を行うことにした。ROI 分割数が増えると ROI の上下範囲も狭くなるため、5 分割以上ではさらに ROI を上下方向に 3 段階に変化させて認識させることにし、この手法の有効性について評価実験をおこなった。また、ROI 分割数を増やすことにより計算量が増大し、計算時間が増えるため、認識率との関係から適切な分割数及び上下移動のあり、なしについて実験により明らかにした。さらに提案手法を実証評価するため、コロナ禍のため中止された ARLISS 2020 の代替大会として開催されたあさぎり CanSat 投下試験(ACTS)に出場し、約 10 m の距離から提案手法の画像認識によりゴールまでのゼロ距離までの誘導に 3 回連続で成功し、提案手法の有効性を確認した。

Deep Learning を用いた画像認識手法には、提案手法のように Deep Learning 画像分類を用いて、その ROI を変化させることにより、ゴールの位置を認識させる手法の他に、Object Detection を用いる手法がある。我々は軽量 Object Detection として代表的な SSD MobileNet V1 [36]を用い、提案手法とゴールからの距離における認識率の比較を行った。その結果提案手法の方が SSD MobileNet V1 よりもゴールからの距離が遠い場合に認識率が高く、また計算時間も短いことを確認した。

一般的に Object Detection は計算量が多いため、組込み機器に用いる場合は、Y. Yoshimoto らのように処理の重い計算の一部を FPGA に行わせ高速化する[37]ことが多いが、専用のハードウェアを用意する必要がある問題がある。我々は CanSat のようにサイズ制約が非常に厳しい環境において、Linux が動作する組込みシステムとして最低性能クラスである Raspberry Pi Zero を用い、単純なモデルを用いて ROI 分割を工夫することにより、CPU の計算能力のみで SSD MobileNet V1 よりも、ゴールから遠距離において高い認識率かつ高速で認識する手法を提案する。

本章では，サイズや重量等制約条件が厳しい CanSat において，計算量の多い Deep Learning 画像認識を適用し，従来手法では困難であったゴールまで 10m の距離からゴール認識を可能にする手法について述べる．

6-2 第 5 章による従来手法

6-2-1 CanSat における Deep Learning 画像認識

現在 CanSat においてゴールまでゼロ距離まで誘導を可能とするために，いくつかのチームで採用されているカメラ画像からゴールの赤色を認識して誘導制御する手法は，晴れや曇りなど，天候の変化による照明条件の変化に弱く，どのような照明条件でもゴールを認識できるように調整すると，ゴールではない似た色の枯れ草等をゴールと誤認識する問題がある．これら問題点を解決するために，我々は Deep Learning を用いたゴール画像認識を用いることにより，色のみで認識する従来手法と比べて，照明条件の変化に強く，より確実にゴールを認識できることを確認している．実際にこの手法で ARLISS 2019 において複数回距離 0 m のゴールを唯一達成し，Overall Winner で 1 位となり，この手法の有効性を確認した．

6-2-2 Deep Learning モデル

一般的に画像分類に用いられている Deep Learning のモデルである VGG16[16]などは、非常に大きなメモリや計算量を必要とするため、我々の CanSat のように Raspberry Pi Zero [bh]等の小型組込みコンピュータでは動作させることができない。Çalik らは、組込みシステム向けの Convolutional Neural Network (CNN)を用いた CIFAR-10 画像分類手法について提案しており 85.9%の高い認識率得ているが、メモリ使用量が 2GB もあり[17]、全メモリが 512MB しかない Raspberry Pi Zero では、これですら動作不可能である。

そこで我々はほぼ最小サイズの Deep Learning 画像分類モデルを用いている。使用している Deep Learning のモデル構造を Fig. 6-2-2 に示す。このモデル構造は Yann LeCun らの LeNet モデル[18]を CIFAR-10 [be]に適応し、2つの 2次元 CNN をそれぞれさらにもう一つ 2次元 CNN を重ねることにより、認識率を向上したものをベースに、本用途に合わせて出力層を 10 から 3 へと変更したものである。また、少しでも認識率を向上させるため、4つ目の 2次元畳み込み層(L4_Conv2D_4)をパディングなしから、元の入力と同じサイズの出力になるようにパディングありに変更している。入力画像は CIFAR-10 と同様の解像度 32x32 ピクセル、RGB フルカラー画像であり、認識結果はゴールが写っている画像、ゴール判定可能な画像、及びゴールが写っていない画像の 3種類とした。Deep Learning の実装には Keras を使い、バックエンドには Tensorflow を用いた。

この単純なモデルで、従来手法の色によるゴール認識手法よりも高い認識率が出せるのか、また天候の変化による明るさや色温度の変化に強いのが重要である。このことについて、我々は過去の研究において、従来手法である色を用いたゴール認識手法よりも、提案モデルを用いた Deep Learning 画像分類の方が、天候変化による明るさや色温度の変化に強く、ゴールからの距離が 0m から 10m まで、全ての場合で従来手法よりも認識率が高いことを明らかにしている[13]。

本研究では従来手法と同じモデルを使用し，ROI 分割についてより詳細に調べることにより，従来よりも長距離であるゴールから 10 m 離れた地点からでも高い認識率を得られる手法について述べる．

Layer (type)	Output Shape	Param #
L1_Conv2D_1 (Conv2D)	(None, 32, 32, 32)	896
L2_Conv2D_2 (Conv2D)	(None, 32, 32, 32)	9248
MaxPooling_1 (MaxPooling2D)	(None, 16, 16, 32)	0
Dropout_1 (Dropout)	(None, 16, 16, 32)	0
L3_Conv2D_3 (Conv2D)	(None, 16, 16, 64)	18496
L4_Conv2D_4 (Conv2D)	(None, 16, 16, 64)	36928
MaxPooling_2 (MaxPooling2D)	(None, 8, 8, 64)	0
Dropout_2 (Dropout)	(None, 8, 8, 64)	0
Flatten (Flatten)	(None, 4096)	0
L5_Dense_1 (Dense)	(None, 512)	2097664
Dropout_3 (Dropout)	(None, 512)	0
L6_Dense_2 (Dense)	(None, 3)	1539
Total params: 2,164,771		
Trainable params: 2,164,771		
Non-trainable params: 0		

Fig. 6-2-2 Structure of deep learning model

6-2-3 Deep Learning モデルの学習

Deep Learning モデルの学習のために、我々はゴールが写っている画像、ゴール判定可能なほどゴールに近い画像、及びゴールが写っていない画像の3種類の画像をそれぞれ 1,703 枚、277 枚、及び 1,441 枚の合計 3,421 枚を用いて訓練を行った。それぞれの画像例を Fig. 6-2-3-1 に示す。(a)のゴールが写っている画像が通常のゴール認識に用いるためのものである。(b)のゴール判定画像は、CanSat がゴールにほぼゼロ距離まで接近し、ゴール判定を行い、動作を停止しても良いと判定するための画像である。(c)はゴールが写っていないときの画像である。訓練回数は 15 EPOCH である。訓練結果の認識精度を Fig. 6-2-3-2 に示す。図の Accuracy は訓練に用いた画像データによる精度であり、val_acc は訓練に用いていない画像による精度である。この図から、過学習は起こっておらず、また認識率は約 98%と高い認識率を実現していることが分かる。



(a) Goal recognition



(b) Goal judgement



(c) No goal

Fig. 6-2-3-1 Examples of images used for training

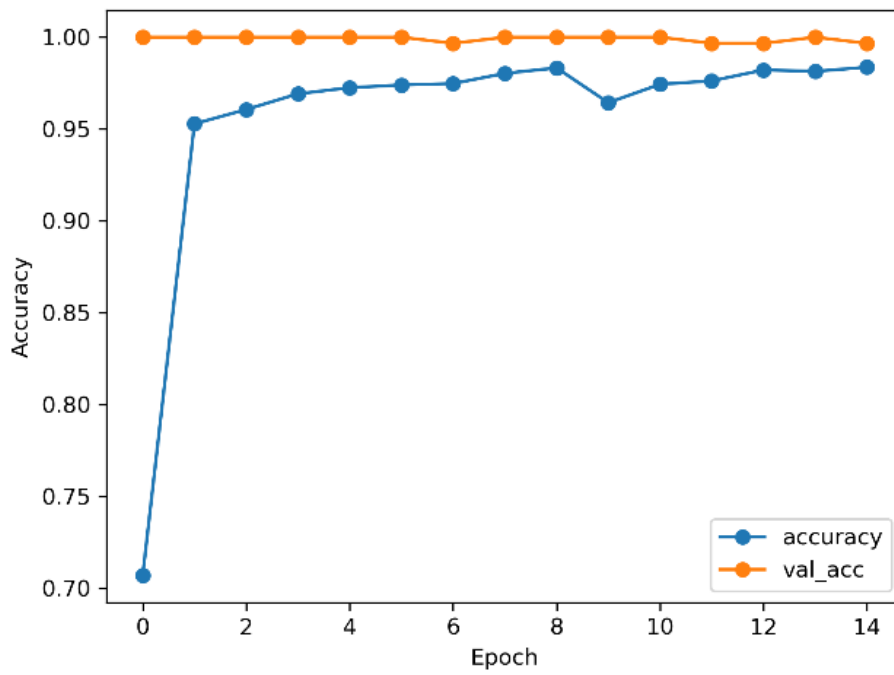


Fig. 6-2-3-2 Model Training Results (Accuracy)

6-2-4 ゴール位置認識手法

我々が用いている Deep Learning のモデルは画像分類であるため、ゴールが画像のどこに写っているのかはわからない。CanSat をゴールまでゼロ距離まで誘導制御するためには、カメラで捉えたゴール画像の左側，中央，右側のどこにゴールが写っているのかを判定し，そちらへステアリングを切る必要がある。そこで，我々は水平方向に ROI を 1 分割，3 分割，5 分割として，それぞれ Deep Learning により画像分類を行うことにより，画像の左右のどの位置にゴールがあるのかを認識している。このアルゴリズムの詳細は以下の通りである。

最初に 1 分割（全画像）で Deep Learning による画像分類を行う。これにより，近距離にゴールがある，ゴール判定可能，及びゴールが写っていない，の 3 通りに分類される。1 分割は，ゴール判定可能かどうかを調べるために用いている。また，どの場合であっても続いて 3 分割での認識を行っている。これは，例えばゴール判定であってもゴールが画像の中央にあるのか，左右にあるのかにより，CanSat の最終的なステアリング位置を決めるためである。また，ゴールなしの場合も全画像では認識不可能なくらいゴールから遠い可能性がある。このことから，続いて水平方向に ROI を 3 分割にして認識を行う。

3 分割での認識例を Fig. 6-2-4-1 に示す。ROI を水平方向に 3 分割し，それぞれ画像分類を行っている。3 分割全ての結果がゴールなしである場合はゴールなしとしている。ここままでゴールが見つかった場合は，ゴール認識処理は終了する。

ここままでゴールが見つからない場合，ゴールが遠くにあり認識できない可能性があるため，さらに水平方向に ROI を 5 分割して，それぞれ画像分類を行う。5 分割での認識結果例を Fig. 6-2-4-2 に示す。3 分割と同様に 5 分割した位置それぞれで Deep Learning ゴール分類を行い，ゴール認識率が最も高い ROI をゴールの位置としている。ここままでゴールが見つからなかった場合には，ゴールなしと判定している。



(a) Left

(b) Center

(c) Right

Fig. 6-2-4-1 Recognition examples in 3 divisions



(a) Left

(b) Center Left

(c) Center

(d) Center Right

(e) Right

Fig. 6-2-4-2 Recognition Examples in 5 divisions

6-2-5 ROI 分割手法

ROI 分割手法について述べる．画像全体を ROI で 2 分割した例を Fig. 6-2-5 に示す．全画像は水平方向に ROI 1 と ROI 2 の 2 つの ROI で分割されている．2 つの ROI の隣り合った部分は overlap で示すようにオーバーラップできるようにしている．オーバーラップ割合は全く重なっていない場合が overlap=0%，ROI の幅の半分まで隣の ROI 同士で重なっている場合が最大とし，その場合を overlap=50 としている．ROI width は水平方向の分割数と overlap から求めている．ROI height は ROI width と同じにしており，ROI は正方形となっている．

ROI は垂直方向に ROI raising amount パラメータ指定で移動できるようになっており，ROI raising amount = 0 なら Center of full image height と同じになり，ROI raising amount=100 なら ROI の四角形の上の辺が Full image の上端と同じになる位置になり，ROI raising amount=-100 なら，ROI の四角形の下辺が Full image の下辺と同じになる位置になるようにしている．

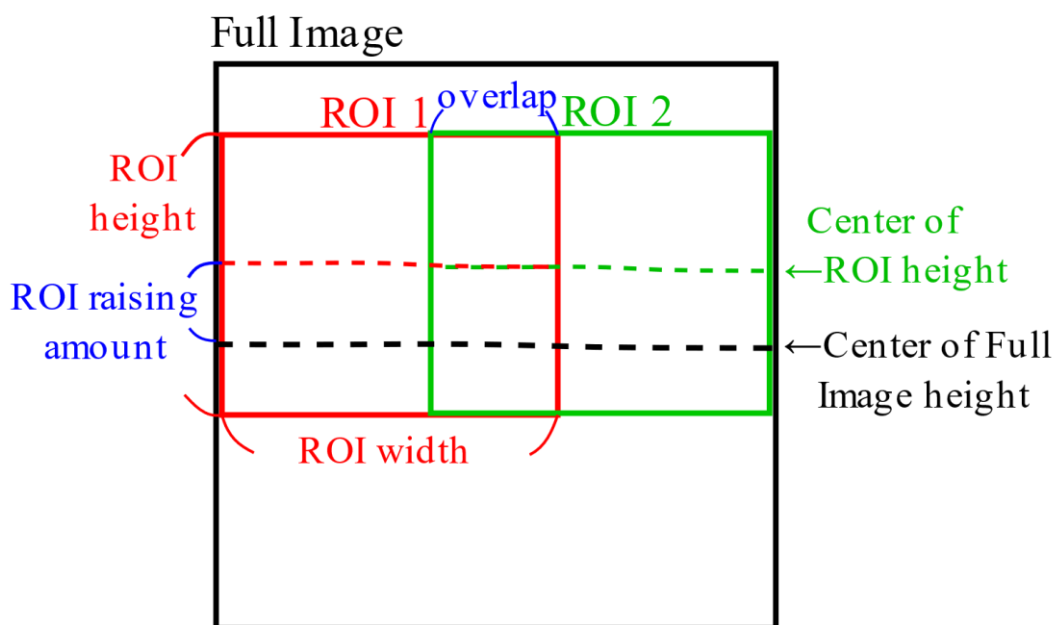


Fig. 6-2-5 ROI 2 divisions

6-2-6 ROI 計算方法

ROIの計算方法について述べる。画像を水平方向にn分割（nは2以上の整数）する場合，その分割位置を固定して，各ROIの幅をoverlap量に応じて増やす手法の場合，両端のROI widthは，その間にあるROI widthより小さくなってしまいう問題がある。そこで我々は全てのROI widthを同一にするため，(6-2-1)(6-2-2)(6-2-3)(6-2-4)の手法で計算している。

$$ROI\ width = image\ width \times \frac{1}{n} \quad (6-2-1)$$

まずoverlap=0でn分割した場合を考える。この場合Fig. 6-2-6-1のように，ROI 1からROI nまでが均等にならび，それぞれ四角形の下端に番号を振ると0からnまでになる。この場合のROI widthは(6-2-1)式で求められる。

$$ROI\ width = image\ width \times \frac{2}{n+1} \quad (6-2-2)$$

次に最大にオーバーラップしているoverlap=50の場合を考える。このときはFig. 6-2-6-2に示すように，隣り合ったROIはROI widthの50%まで重なり合っている状態になっている。1つのROI widthの丁度半分の位置で長方形が重なる形となり，この場合もFig. 6-8のように全ROIの縦線の位置に左から順に0から番号を振ると，右端はn+1になる。この場合，ROI widthの半分はimage width / n+1なので，ROI widthは式(6-2-2)で求められる。

$$ROI\ multiply = \frac{overlap}{50} \times \left(\frac{2}{n+1} - \frac{1}{n} \right) + \frac{1}{n} \quad (6-2-3)$$

このことからoverlap=0のとき $\frac{1}{n}$ ，overlap=50のとき $\frac{2}{n+1}$ とimage widthとの積がROI widthとなるので，overlapを指定してimage widthに掛ける値ROI multiplyは式(6-2-3)で表すことができる。

$$ROI\ width = image\ width \times ROI\ multiply \quad (6-2-4)$$

以上よりROI widthは式(6-2-4)により求められる。

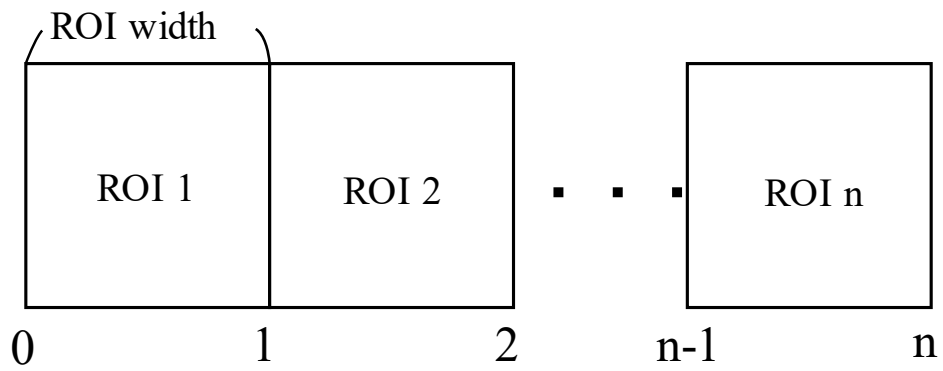


Fig. 6-2-6-1 ROI n divisions

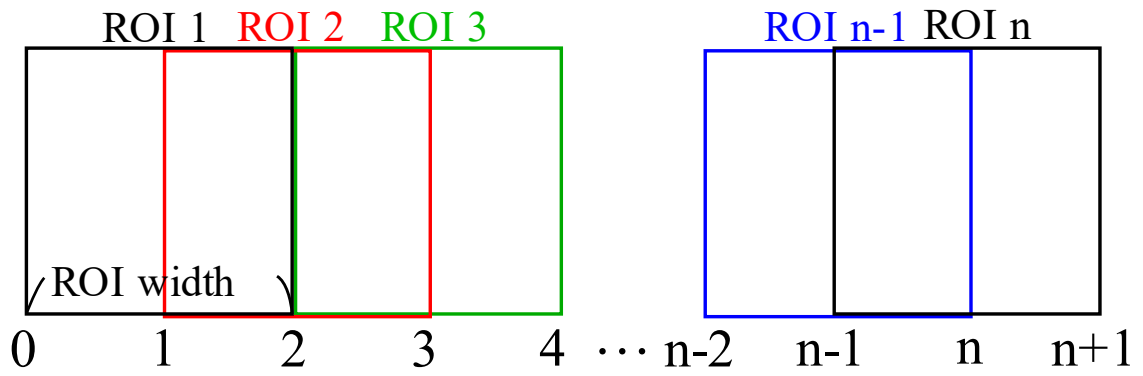


Fig. 6-2-6-2 ROI n divisions with 50 percent overlap

6-2-7 画像認識によるゴール誘導制御手法

Deep Learning を用いた画像認識では、計算に時間がかかるため、リアルタイムで走行しながらゴールへとそのまま走行していくことは困難である。そのため、GNSS によりゴール付近へと近づき、カメラ画像認識誘導に切り替わると、その時点で CanSat は停止する。そしてカメラ画像からゴール認識を行う。ゴールが画像の左右どの位置で認識されたかで、ステアリングをそちらの方向へ切り、2 秒間ゆっくりその方向へ走行して再び停止し、再度カメラ画像からゴールを探す。この動作を繰り返してゴールへと向かって行く。CanSat の方向がゴールの方向へ向いておらず、カメラでゴールが認識できない時には、左に最大にステアリングを切り 2 秒間走行し停止する。これにより約 30 度 CanSat は左側へ回転する。これを繰り返し、周囲 360 度どの方向にゴールがあっても見つけられるようにしている。ゴール認識動作を 20 回繰り返してもゴールを見つけれないときは、何らかの理由によりゴールが発見できない位置にいる可能性を考慮して、再び 40 秒間 GNSS 誘導制御でゴールへ向けて走行を行い、再度その位置から、ゴールを探す動作を行う。

6-3 提案手法

6-3-1 従来手法の問題点

最近複数の CanSat チームで採用されはじめたゴールの赤色を認識する手法と比較して、Deep Learning を用いた従来手法は照明条件の変化や、薄暗かったり、直射日光の強い日差しだったりなどでも高い認識率を示し、様々な問題点を解決している。実際に ARLISS 2019 では 3 回中 2 回はゴールまでゼロ距離まで誘導することに成功し、残り 1 回も 0.05 m まで近づくことに成功した。

しかし、この 0.05 m ゴールの時には、GNSS でゴールまで約 2 m 程度まで近づき停止したが、その後強風により CanSat が約 8 m の距離まで流されてしまった。この時には、既に GNSS による誘導制御はやめていたため、画像認識によりゴールに近づく必要があるが、従来手法ではゴールから 8 m も離れると認識することが難しい。20 回以上ゴールを画像認識で探しても見つからない時には、再び GNSS で 40 秒間ゴール付近まで走行し、その地点で再びゴールを探す動作を繰り返し行うが、このときには GNSS の測位誤差が大きくなっており、再び停止する位置がやはり約 6~9 m 離れた地点であったため、何度繰り返してもなかなかゴール認識することができず、結局 30 分ほどしてようやくゴールを認識して 0.05 m まで近づくことに成功した。

最終的にゴールを発見できたが、この動作には問題がある。実際に大学のグラウンド等でゴール誘導実験を行うと、悪天候時に GNSS の測位衛星数が少なくなると、測位誤差が大きくなり、6 m 以上離れた位置で止まることがある。そうなると同様にゴールを見つけるのが難しくなる問題点がある。ARLISS は時間制限がないが、能代宇宙イベントや種子島ロケットコンテストには 15 分間の制限時間があるため、このように GNSS の測位誤差が大きくなった場合には、従来手法ではゼロ距離ゴールを実現できない可能性がある。そこで、我々の GNSS 測位誤差の研究結果[19]から、GNSS の測位誤差は最大でも 10 m 以内であるため、ゴールまでの距離が 10 m の地点からでもゴールを認識することができれば、どんな条件でもほぼ確実にゼロ距離ゴールが可能であると考えられる。

6-3-2 水平 ROI 分割数の増加手法

我々の従来手法の研究により，ゴールまで距離が 10 m での認識率は，晴天時には約 50%であり，曇天時には 0%であることが分かっている．また，8 m の地点であっても曇天時には 50%程度の認識率しかないため，照明条件が必ずしも良くない場面では，従来手法では約 7 m までが限界であり，それ以上の距離では認識困難であることが分かっている[13]．そこで，より遠距離からでもゴールを認識できるようにするために，5 分割でも見つからなかったときには，さらに ROI を水平方向に 6 分割から 9 分割することにより ROI を小さくすることにより，より遠距離からのゴール認識が可能になるのではないかと考えた．

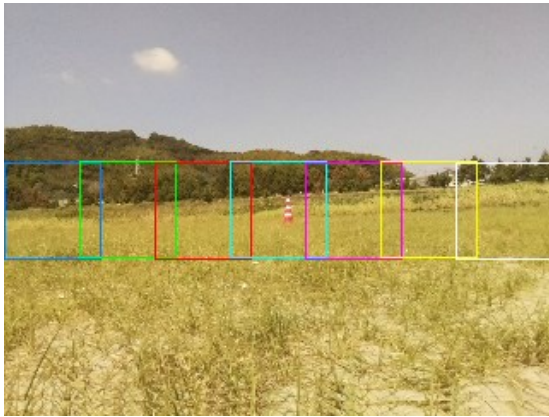
6-3-3 上下 ROI 3 段階移動手法

ROI の分割数を増やしていくと，水平方向だけではなく，垂直方向の ROI のサイズも小さくなるため，ゴールの画像中の上下の位置が重要になってくる．CanSat のフィールドは，凹凸のある砂漠や草地で走行するため，CanSat のピッチ角度は変化してしまい，その結果，画像内でのゴールの位置は上下することになる．このため，たとえ水平方向への分割数を増やしても，それだけでゴールを見つけられるとは限らない．そこで，5 分割時以上では，ROI を基準位置より上下に移動させて，さらに認識を行うことにより，CanSat のピッチ角度変化問題を吸収できるのではないかと考えた．

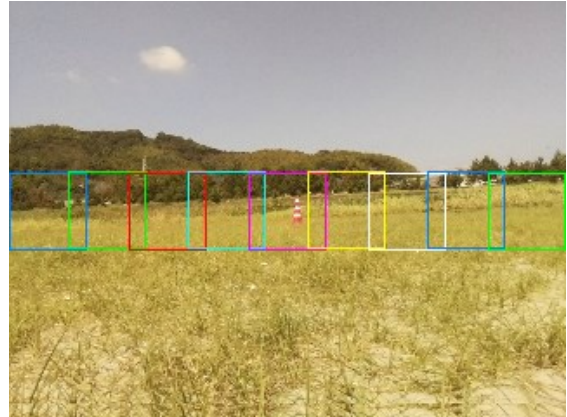
6-4 評価実験

提案手法の有効性を評価するために，ゴールからの距離を 0 m から 10 m まで変化させ，ゴールの撮影角度を 45 度ずつ回転しながら，それぞれゴールが画像の左，中央，右になるように変化させて 1 周分撮影した画像データを用いて，各分割数におけるゴールからの距離と認識率との関係を求めた．ゴール画像は，Table 6-4-1-1 に示す通り，晴天時と曇天時の条件で撮影した 2 種類の画像を用いた．ゴール画像の撮影には我々の開発した CanSat を用いた．ゴール撮影用カメラは Raspberry Pi Camera V2 を用いた．撮影解像度は直射日光画像が 1280x960 pixels であり，日陰画像が 640x480 pixels である．実際の CanSat プログラムでは 320x240 pixels で撮影しているため，これら画像データを 320x240 pixels にリサイズして認識データの取得を行った．評価実験に使用したコンピュータやソフトウェア環境を Table 6-4-1-2 に示す．

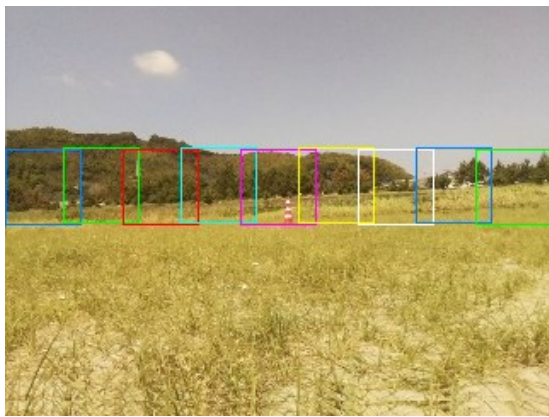
実験ではオーバーラップは 15%，上下移動量は 15% に設定した．ゴールのパイロンから 10 m 離れた画像を，提案手法で 7 分割や 9 分割したときの実際の分割 ROI を Fig. 6-4-1-1 に示す．(a) は 7 分割，(b) は 9 分割，(c) は 9 分割で上に 15% 移動，及び (d) は 9 分割で下に 15% 移動したときの ROI を示している．実際のフィールドは Fig. 6-4-1-1 のように傾いていたり，CanSat 自体のピッチ変化により画像中を上下にゴール位置が移動することになる．そのため (c) や (d) のように ROI を上下に移動できるようにしている．また Fig. 6-4-1-2 は Fig. 6-4-1-1 (b) の左から 5 つ目の ROI により取り出された画像データを 32x32 pixels にリサイズしたものである．この画像が実際に Deep Learning Image Classification に入力される．



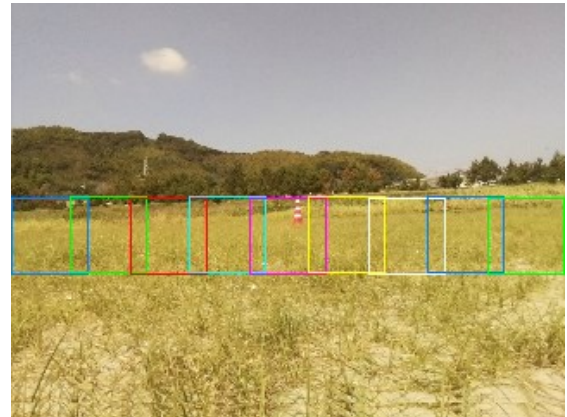
(a) 7 divisions



(b) 9 divisions



(c) 9 divisions 15% up



(d) 9 divisions 15% down

Fig. 6-4-1-1 ROI image at 10m distance from the goal



Fig. 6-4-1-2 9 divisions ROI Image

Table 6-4-1-1 Goal image shooting conditions

Place	Nishiura coast	Ground at Aichi University of Technology
Lighting conditions	Direct sunlight (equivalent to a sunny day)	Shade of the building (equivalent to a cloudy day)
Time	11:00 – 16:00	16:00 – 18:00
Date	August 26, 2020	August 13, 2018
Resolution	1280 x 960 pixels	640 x 480 pixels
Camera	Raspberry Pi Camera V2	
Goal pylon	Red color standard size with white line Size: Height about 70 cm / Width about 39 cm	
Image shooting distance	0m, 0.5m, 1m, 2m, 3m, 4m, 5m, 6m, 7m, 8m, 9m, 10m	
Image shooting direction	8 directions by shifting by 45 degrees	
Number of images	Each angle goal image position left, middle, right 3 images in total 24 images. 288 images at all distances.	

Table 6-4-1-2 Experimental environment

Computer	Raspberry Pi Zero WH
OS	Raspbian GNU/Linux 9.13 (stretch)
Python	Version 3.5.3
Tensorflow	Version 1.11.0
Keras	Version 2.1.6-tf

6-4-1 ROI6 分割から 9 分割の認識率

従来手法の 5 分割と 6 分割から 9 分割までのゴールからの距離における認識率の変化を測定した結果を Fig. 6-4-1-3 及び Fig. 6-4-1-4 に示す。Fig. 6-4-1-3 は晴天時の認識率であり、Fig. 6-4-1-4 は曇天時の認識率である。それぞれ従来手法の 5 分割時と提案手法の 6 分割から 6 分割時における、ゴールからの距離と認識率との変化を示している。どちらも 5 分割と 6 分割のラインは全く同じ結果であったため重なっており、6 分割の結果は 5 分割と全く同一であった。晴天時に 10 m 地点において 5 分割及び 6 分割で 48%の認識率が、7 分割では 76%に大きく向上しており、8 分割では 88%、9 分割では 92%と、分割数が大きくなるに従って向上している。曇天時も同様に、10 m 地点において 5 分割及び 6 分割で 0%の認識率が、7 分割は 45%、8 分割及び 9 分割では 64%と向上している。

どちらの場合も 6 分割は 5 分割と比べて全く認識率が向上しておらず、7 分割では大きく認識率が向上することから、5 分割よりも分割数を増やす場合は最低でも 7 分割を用いる必要がある。

以上より、ゴールからの距離が 10 m では、従来手法の 5 分割では認識率が不足しており、発見が難しいが、提案手法の 7 分割から 9 分割では、10 m であっても晴天時に 76%以上、曇天時でも 45%以上まで認識率が向上することが明らかになった。

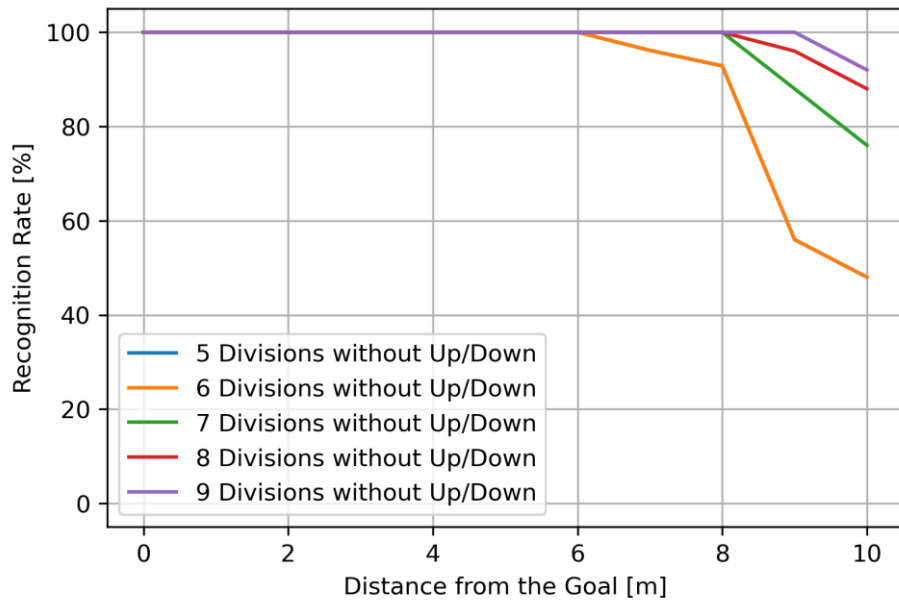


Fig. 6-4-1-3 Recognition rate at a distance from the goal without up and down (on a sunny day)

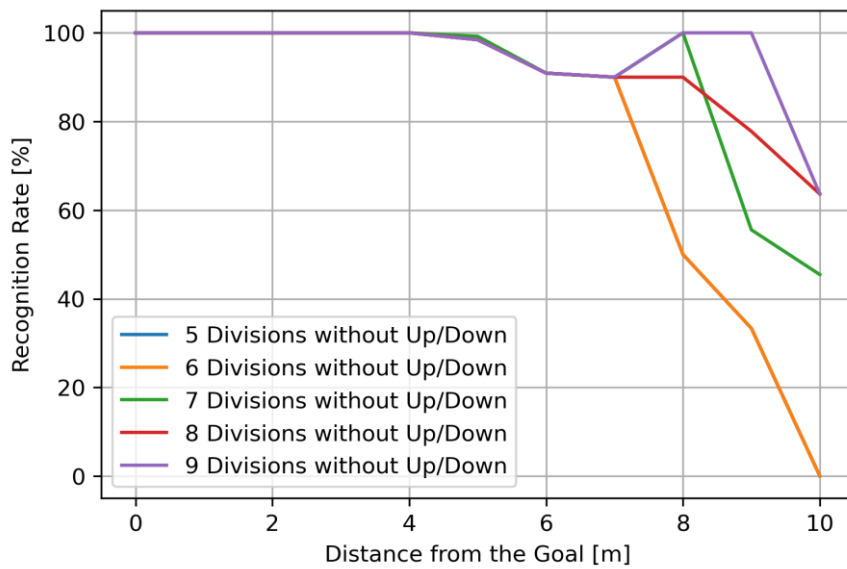


Fig. 6-4-1-4 Recognition rate at a distance from the goal without up and down (on a cloudy day)

6-4-2 上下移動あり ROI 7 分割時の認識率

次に、6 分割以上では ROI を上下へ移動させて認識させた結果について述べる。CanSat ではきれいな水平面ではなく、土や草により凹凸のある路面を走行するため、停止してカメラで前方を撮影したときに CanSat のピッチは多少前後に傾くため、カメラで捉えた画像は上下方向に少し動いた画像になることが多い。そのため ROI を水平方向へ 6 分割以上すると、ROI の垂直方向も狭くなるため、ゴールが ROI から上下方向へ外れてしまい、そのため認識できない可能性がある。この問題を解決するために、6 分割以上では、ROI を上下方向に移動させて 3 段階の認識も加えた場合、認識率がどれほど向上するかを実験により調べた結果を示す。

Fig. 6-4-2-1 は晴天時、Fig. 6-4-2-2 は曇天時の距離と認識率を、6 分割から 9 分割においては上下に ROI を移動させて調べた結果である。ROI 上下移動量は 15%とした。晴天時では、上下移動のない Fig. 6-4-1-3 と上下移動のある Fig. 6-4-2-1 を比較すると、6 分割では全く同一であり、上下移動させても認識率は向上していない。10 m 地点において 7 分割では 76%から 84%に向上し、8 分割では 88%のまま変化せず、9 分割では 92%から 96%へと向上した。

曇天時では、上下移動のない Fig. 6-4-1-4 と上下移動のある Fig. 6-4-2-2 とで比較すると、6 分割では認識率の向上がなかった。10 m 地点において 7 分割では 45%から 64%へと向上し、8 分割及び 9 分割では 64%から 73%へと向上した。

以上より、上下移動による認識率の向上は 6 分割では全くみられなかったが、7 分割から 9 分割では 4%から 19%の向上がみられた。このことから、ROI 上下移動は 7 分割以上では認識率の向上がみられるが、向上する認識率は余り大きくはなかった。

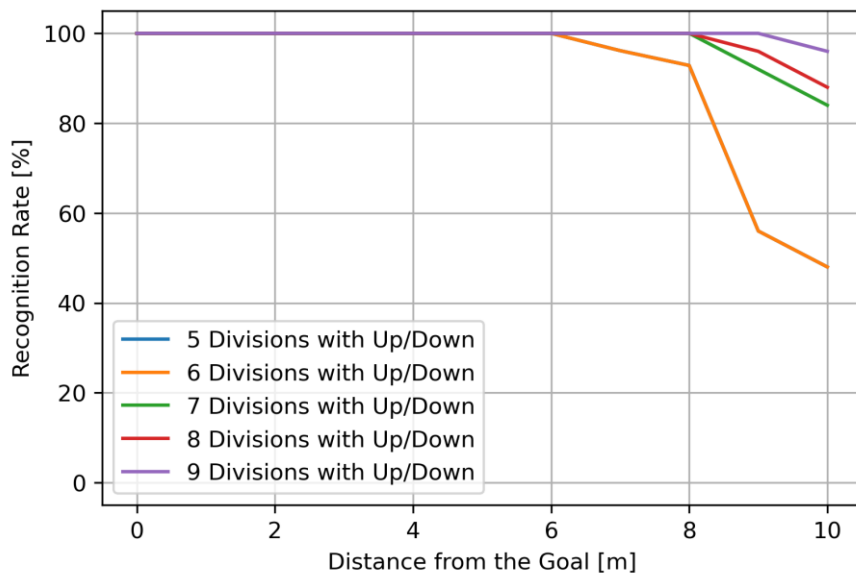


Fig. 6-4-2-1 Recognition rate at a distance from the goal with up and down (on a sunny day)

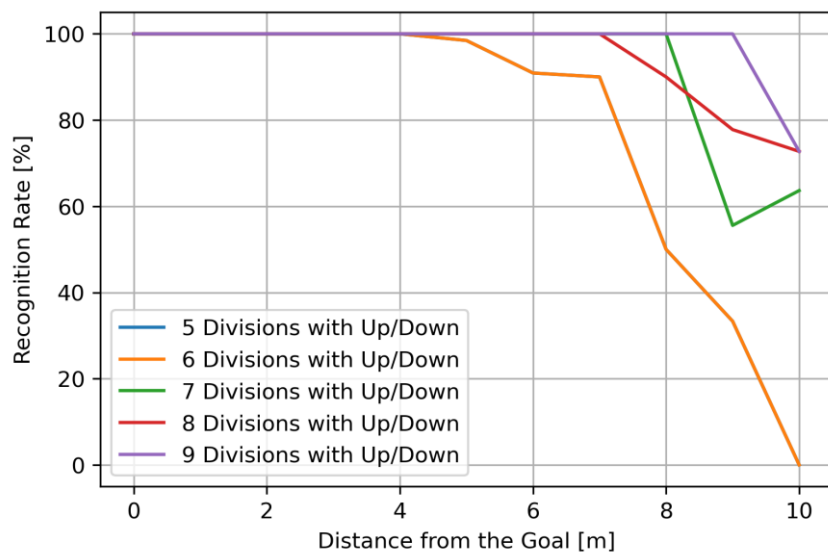


Fig. 6-4-2-2 Recognition rate at a distance from the goal with up/down (on a cloudy day)

6-4-3 ROI 分割数とゴールからの距離との関係

次に、ゴールからの距離と ROI 分割数との関係について述べる。Fig. 6-4-3-1 及び Fig. 6-4-3-2 に、ゴールからの距離と認識した ROI 分割数との関係について示す。Goal Judgement は全画面（1 分割）でゴール判定画像と認識された割合であり、Full Screen は 1 分割（全画面）でゴール認識した割合である。3, 5, 6, 7, 8, 9 divisions はそれぞれ 3 分割から 9 分割でゴールを認識した割合を示している。

Fig. 6-4-3-1 の晴天時及び Fig. 6-4-3-2 の曇天時のどちらもほぼ同じ傾向を示している。Goal Judgement を見ると、どちらもゴールまでゼロ距離では 100%、1 m では 0%となっている。これは理想は 0 m のとき以外は認識率が 0%になるのが理想であるため、ゴール判定認識率がうまく機能していることが分かる。ゴールから 0.5 m から約 3 m までは Full Screen (1 分割) でゴールを認識している。このため Full Screen でゴールを発見した場合には、ゴールまでの距離が 3 m 以内である可能性が高い。ゴールからの距離が 4 m ~ 6 m では 3 分割での認識率が高くなっているため、3 分割でゴールを認識した場合は、ゴールまでの距離が 4 m ~ 6 m である可能性が高い。同様に約 6 m ~ 8 m では 5 分割の認識率が高いため、5 分割では 6 m ~ 8 m である可能性が高い。8 m 以上では 7 分割から 9 分割での認識率も高くなっているため、7 分割から 9 分割で発見した場合は、ゴールまでの距離は 8 m 以上である可能性が高い。

このように、ゴールを発見した ROI 分割数により、ゴールまでのおおよその距離が分かるため、これによりゴールを発見した ROI 分割数と画像の左右の位置とステアリングを回す角度との関係を、それぞれの距離から適切に設定することが可能である。

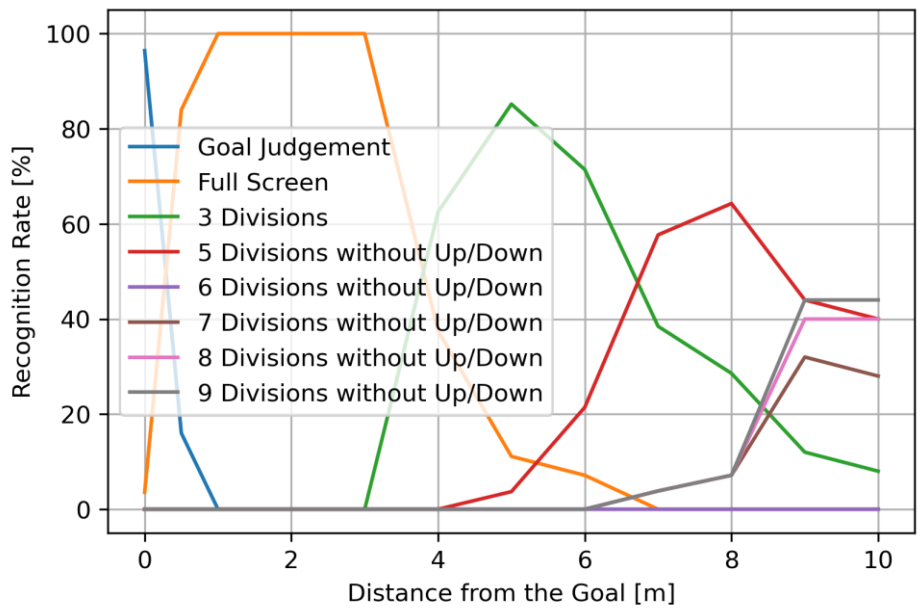


Fig. 6-4-3-1 Number of ROI segments and recognition rate (in direct sunlight)

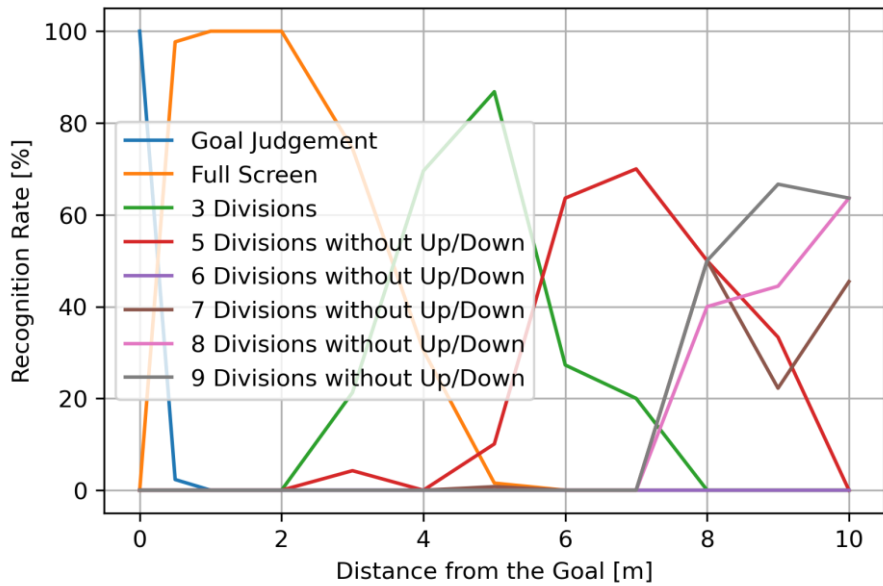


Fig. 6-4-3-2 Number of ROI segments and recognition rate (on a cloudy)

6-4-4 計算時間

Deep Learning を用いたゴール認識に要する計算時間について述べる．計算時間の測定結果を Fig. 6-4-4 に示す．横軸の Division type は計算に用いた画像の分割数を示しており，1 であれば全画像のみ，3 であれば，全画像の計算をした後に 3 分割の計算をしたとき，7UD は全画像の後，3 分割と 5 分割の計算をし，さらに 7 分割で ROI を上下にずらして計算したときの合計計算時間である．棒グラフは平均計算時間，黒色の縦棒は標準偏差を示している．

Division type が 1 では，平均計算時間が約 0.32 s であることから，Deep Learning による 1 回の画像分類にかかる計算時間は約 0.32 s ということになる．また，その標準偏差が約 0.08 s であることから，計算時間のバラつきはほとんどないことが分かる．実際には最低でも 3 分割までは計算するアルゴリズムであるため，1 回あたり最低必要な計算時間は Division type が 3 の約 1.32 s である．従来手法では，ゴールが見つからないときは 5 分割まで計算していたので，Number of divisions が 1+3+5 の約 2.94 s の計算時間が 1 回あたり最大でかかっている．さらに提案手法である 7, 8 及び 9 分割も計算したものがそれぞれ Division type が 7, 8 及び 9 であり，この場合の平均計算時間はそれぞれ約 5.51 s，約 5.57 s 及び約 6.26 s である．従来手法と比べて計算時間が約 3 秒間増加している．

従来手法の 5 分割に加えて 7, 8 及び 9 分割の上下移動したものがそれぞれ Division type が 7UD, 8UD 及び 9UD であり，その場合の平均計算時間は約 10.5 s，約 10.9 s 及び約 12.4 s であった．

この計算時間増加は，ARLISS のように時間制限が設けられていない CanSat コンテストであれば特に問題はない．しかし能代宇宙イベントや種子島ロケットコンテストのように 15 分間の制限時間があるコンテストでは，少しでも計算時間は短いほうが良い．

このようなことから，ARLISS のように時間がそれほど重要ではない場合は 9 分割に ROI 上下移動を行うものを用いても良いが，制限時間がある場合には 7 分割または 7 分割に ROI 上下移動を行い，CanSat のピッチ変化に対応したものが良いと考えられる．

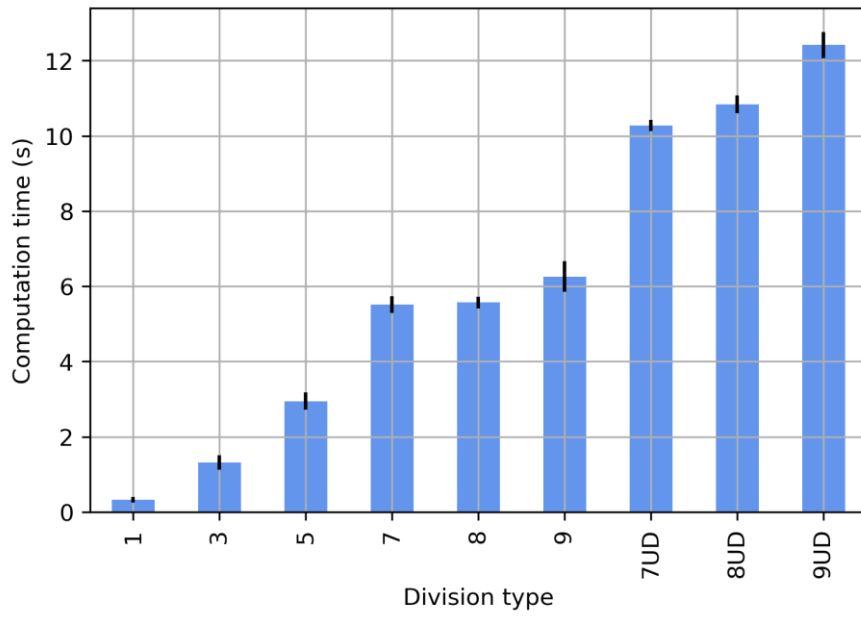


Fig. 6-4-4 Computation time

6-4-5 通信時間

我々の CanSat のメインプログラムは高速化のため C++でプログラムしている． Deep Learning 画像認識のプログラムは Keras ライブラリ [38]を用いているため Python を用いてプログラムしている． このため， メインプログラムと Deep Learning の Python プログラムを別スレッドとして同時に動作させている． Deep Learning プログラム側はサーバとして動作しており， メインプログラムから画像データをプロセス間通信により送信すると， その認識結果をメインプログラムに返してくる仕組みになっている．

プロセス間通信には ZeroMQ [21]を使用した． 認識させる画像データは 32x32 ピクセル， RGB 各 8bit であるため， 1 回の Deep Learning 認識に約 3kB のデータを送信している． その他制御や結果受信に約 10B のデータを送受信している． この通信にかかる平均時間の測定結果を Fig. 6-4-5 に示す． 1 回の画像認識に必要な通信時間は約 7.6 ms であり， 全体の認識時間約 0.33 s のうち約 2.3%である． また， Division type が 9UD では最大 36 回通信を行うが， その時の通信時間は約 0.28 s であり， 全体の認識時間約 12.3 s のうち， 約 2.3%である． このように， 最大でも約 279 ms であり， 全体に認識時間約 12.7 s に対する通信時間は約 2.2 %であるため， ZeroMQ によるプロセス間通信時間は十分に短い時間であることが明らかになった．

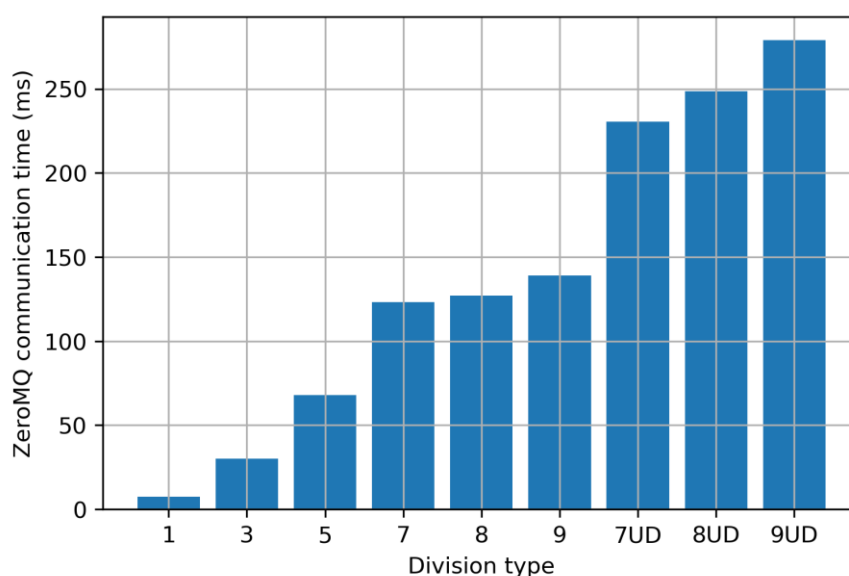


Fig. 6-4-5 ZeroMQ communication time

6-5 Object Detection との比較

6-5-1 認識率の比較

Deep Learning による画像認識手法として、画像のどの位置に目的物体が存在するかを認識させるには、Object Detection を用いる方が簡単である。そこで軽量な Object Detection モデルとして代表的な SSD MobileNet V1 を用いてゴールからの距離と認識率との関係を調べた。

SSD MobileNet V1 の学習データは提案手法のモデルの訓練に用いた画像と同じものを同じ枚数使用し、同じ条件になるようにした。ゴールが写っている画像が 1,703 枚であり、ゴール判定可能画像が 277 枚の合計 1,980 枚を訓練に使用した。全ての画像データに LabelImg [bi]によりパイロンにアノテーションを行った。使用したモデルは COCO データセットで 90 objects を認識するように訓練済みのものを使用した。我々はこのモデルの先頭部分の CNN10 層をフリーズし、それ以降の層を訓練する転移学習を行った。訓練ステップ数は 500 ステップである。訓練には Tensorflow v1.14 を用い、訓練結果は高速化のために Tensorflow Lite 形式に変換した。SSD MobileNet V1 の入力画像解像度は 300x300 pixels, RGB3 チャンネルフルカラー画像データである。

認識結果を Fig. 6-5-1 に示す。晴天時も曇天時も 5m 辺りから認識率が低下しはじめ、7m で約 50%まで認識率が低下し、10m では晴天時は 12%、曇天時は 0%まで認識率が低下している。これは我々の提案手法による認識率を示した Fig. 6-4-1-3, Fig. 6-4-1-4, Fig.6-4-2-1 及び Fig. 6-4-2-2 と比較して、従来手法である 5 分割までの認識率よりもさらに低い。このため、長距離で小さい物体の認識に関しては、SSD MobileNet V1 よりも我々の提案手法の方が高いことが明らかになった。

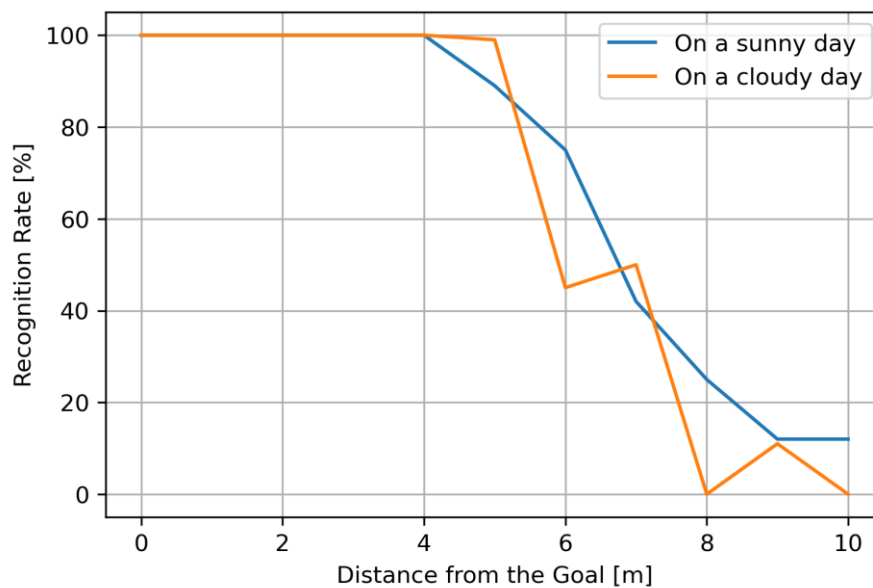


Fig. 6-5-1 Recognition rate of SSD MobileNet V1

6-5-2 計算時間の比較

SSD MobileNet V1 の計算時間を測定した結果、平均約 12.3 s であり、標準偏差は 0.11 であった。この計算時間は、提案手法の 9 分割上下移動ありの約 12.4 s とほぼ同じである。提案手法ではゴールが見つかった場合、そこで計算をやめるので、例えば 3 分割でゴールが見つければ約 1.32 s で、5 分割であれば約 2.94 s で計算が終わる。9 分割上下移動ありの場合の最長計算時間が約 12.4 s である。それに対して SSD MobileNet V1 では、ゴールまでの距離に関係なく、必ず約 12.3 s の計算時間が必要となる。

このようなことから、提案手法で最長の計算時間がかかった場合であっても、SSD MobileNet V1 とほぼ同じであり、我々の提案手法はゴールに近い場合はもっと短時間で計算できるため、総合的に計算時間は短いと考えられる。

6-6 実証評価実験

提案手法の実際の CanSat コンテストでの有効性を評価するため、あさぎり CanSat 投下試験 2020 (ACTS 2020)に参加した。2020 年は COVID-19 感染問題のため ARLISS が開催中止となり、その代替大会として ACTS 2020 が開催された。会場は静岡県富士宮市のあさぎりフードパークであり、CanSat は気球を用いて上空約 30~50 m まで上昇させ、そこから投下する。落下地点からゴール地点までの距離は約 50~60 m であり、なだらかな斜面の草地を走行する。ゴール地点には赤色のパイロンが置かれており、制限時間は気球から投下開始から 15 分間である。この競技内容は能代宇宙イベントとほぼ同じものである。競技では CanSat の投下は 2 回までとなっているが、公式記録にはならないものの 3 回目も投下可能であったため、我々はデータ取得のため 3 回投下を行った。

我々は提案手法の有効性を確認するため、3 回の投下のうち 2 回目と 3 回目では、ゴールまでの距離が 10 m 以内になったところで GNSS 誘導制御をやめ、そこから提案手法を用いた Deep Learning ゴール誘導制御のみでゴールまでゼロ距離まで誘導可能かを調べた。また、ゴール画像認識には提案手法のうち 7 分割で上下移動ありのものを用いた。これは本コンテストでは制限時間 15 分間の指定があるため、提案手法の分割数としては最も小さい 7 分割を選ぶことにより短時間で計算を行いたかったこと、及び提案手法の中では最も認識率の低い 7 分割でも 10m からの認識が本番でも可能かを調べるためである。また実際のフィールドは路面が凸凹しているため、CanSat はピッチ変化が大きくなるため、ROI 上下移動ありを用いた。

6-6-1 投下 1 回目

CanSat 投下 1 回目の走行軌跡を Fig. 6-6-1 に示す. CanSat は高度約 33 m から投下され, ゴール地点から GNSS 測定結果で 60.84 m 離れた地点へと落下し, そこからゴールへ向けて GNSS で走行していった. 図の軌跡の色は, 各地点におけるステアリング角度を表しており, 0.20 が右へ最大に, -0.20 が左へ最大にステアリングを操作したことを示している. GNSS 測位によりゴールまでの距離が 1.33 m の地点で GNSS 制御はやめて, カメラ画像の Deep Learning を用いた画像認識によるゴール誘導制御へと切り替えている. 結果はゴール地点までゼロ距離まで誘導制御に成功した. このときの GNSS 測位誤差は 0.8 m であった. 競技開始からゴール判定までの所要時間は 5 分 55 秒であった.

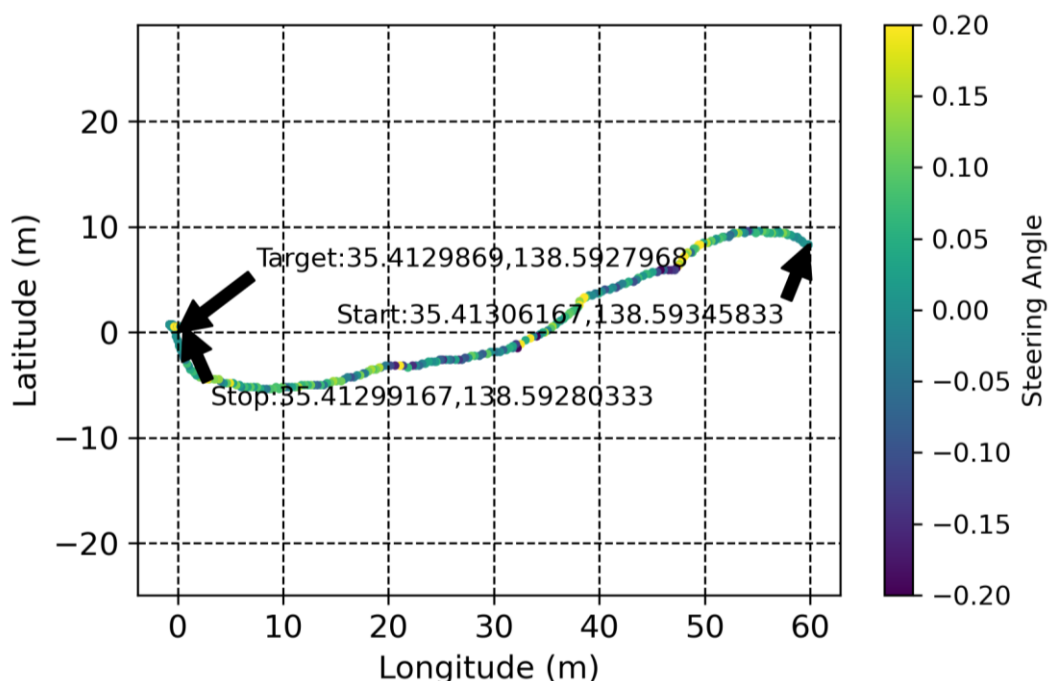


Fig. 6-6-1 Running locus in 1st experiment

6-6-2 投下 2 回目

CanSat 投下 2 回目の走行軌跡を Fig. 6-6-2 に示す. CanSat は高度約 40 m から投下され, ゴール地点から GNSS 測定結果で約 58 m 離れた地点へと落下し, そこからゴールへ向けて GNSS で走行していった.

2 回目は提案手法による Deep Learning ゴール認識のみでゴール地点までゼロ距離まで誘導可能か調べるために, ゴールまでの距離が 10 m になったところで GNSS 誘導制御をやめた. その結果, GNSS 測位でゴールから 9.84 m 離れた地点でローバは停止し, そこから画像認識による誘導制御へと切り替えた. 結果, ゴールまでゼロ距離まで誘導制御に成功した. このときの GNSS 測位誤差は 2.5 m であった. 競技開始からゴール判定までの所要時間は 7 分 17 秒であった.

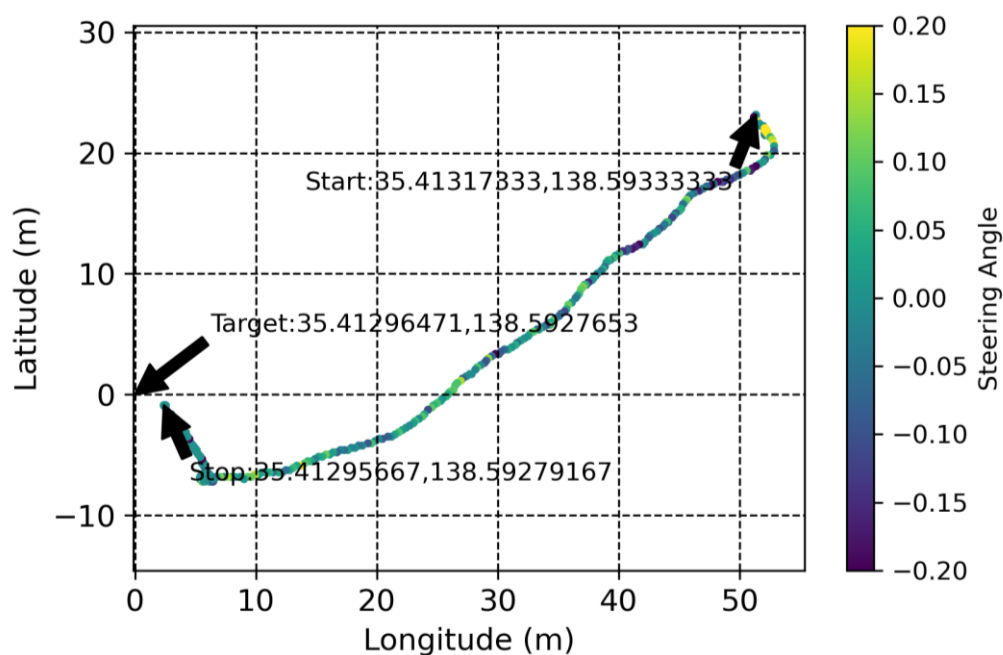


Fig. 6-6-2 Running locus in 2nd experiment

6-6-3 投下 3 回目

CanSat 投下 3 回目の走行軌跡を Fig. 6-6-3-1 に示す。CanSat は高度約 24 m から投下され、ゴール地点から GNSS 測定結果で約 58 m 離れた地点へと落下し、そこからゴールへ向けて GNSS で走行していった。

3 回目もゴールまでの距離が 10 m 以内になったところで GNSS 誘導制御をやめているため、GNSS 測位でゴールから 9.77 m 離れた地点でローバは停止し、そこからカメラ画像の Deep Learning を用いたゴール認識による誘導制御へと切り替えた。ゴールへ向けて Deep Learning で誘導していったが、ゴール手前約 30 cm のところで 15 分間のタイムアウトとなったが、3 回目は公式記録ではないため、そのまま競技を続行した。結局競技開始後 17 分 11 秒でゴール判定したため、2 分 11 秒オーバーしたが、ゴールまでゼロ距離誘導制御に成功した。このときの GNSS 測位誤差は 3.01 m であった。3 回目は公式記録にならないため、データ取得を優先し、動画を撮影しながら動作させたため、その分 CPU 負荷が増加しており、通常よりも計算時間がかかっていたことが時間制限をオーバーした主な原因と考えられる。

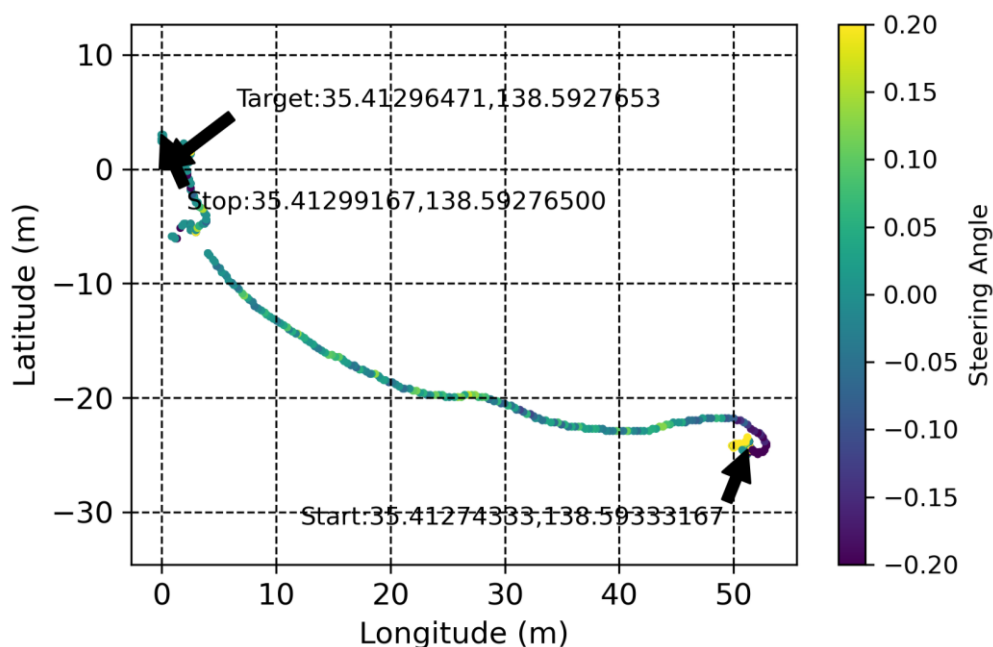


Fig. 6-6-3-1 Running locus in 3rd experiment

このとき、ゴールから約 10 m の位置でカメラが捉えた実際の画像を Fig. 6-6-3-2 に示す。Fig. 6-6-3-2(a)ではほぼ水平ではあるが、ゴールは少し画像の中央よりも上にある。また Fig. 6-6-3-2 (b)では画像全体が傾いている。これは草地であるため、CanSat 本体がピッチ、ロール方向に傾くためである。このような時に ROI の上下移動が必要になる。



(a) Recognized in the center



(b) Recognized in the right

Fig. 6-6-3-2 Example of goal recognition at a distance of 10m from the goal in ACTS 2020

6-6-4 実証実験結果の考察

1 回目は通常通り，GNSS 測位結果がゴールまでの距離が 1.33 m のところで GNSS 誘導制御をやめ，カメラ画像による Deep Learning ゴール認識誘導制御に切り替え，ゴールまでゼロ距離ゴールを達成した。

2 回目及び 3 回目は提案手法の有効性を確認するため，ゴールまでの距離が 10 m 以内になったところで GNSS 誘導制御をやめ，提案手法によるカメラ画像によるゴール誘導制御を行った。2 回目 3 回目ともにゴールから約 10 m 離れた地点からカメラ画像でゴール認識に成功し，最終的にゴールまでゼロ距離まで誘導に成功した。

このことから提案手法を用いることにより，GNSS の測位誤差等の理由により，CanSat がゴールから 10 m 離れた地点で停止したとしても，そこからゴールを画像認識で見つけ，ゴールまでゼロ距離まで誘導制御可能であることを確認した。

公式記録ではない 3 回目は 15 分の時間制限を 2 分 11 秒オーバーしてしまい，制限時間内にはゼロ距離ゴールが達成できなかったが，これは通常は用いない動画撮影を CanSat で行ったのが主な原因である。しかし Deep Learning によるゴール認識の計算時間が長いために制限時間をオーバーしたこともまた事実である。そのため，計算時間は可能な限り短い方が良い。今回は division type として 7 分割上下移動あり (7UD) を用いたが，8UD や 9UD ではなくてもゴールから 10 m 離れた地点からゼロ距離まで誘導できたため，制限時間のあるコンテストでは division type は 7UD が良いと考えられる。

また，最終的にゼロ距離ゴールしたときの GNSS 測位情報は 1 回目 0.8 m，2 回目 2.5 m，3 回目 3.01 m であった。このため，いずれも GNSS のみではゼロ距離ゴールは不可能であり，特に 2 回目及び 3 回目は約 2.5 m から 3 m の地点までしか誘導できなかったことになる。3 回の CanSat 投下のうち，最初の 2 回までが競技記録となり，ACTS 2020 参加チームのうち，我々は 2 回ともゼロ距離ゴールを達成した唯一のチームであったため，カムバック部門で 1 位を獲得し，カムバック賞を受賞した。ACTS 2020 での 0m ゴールの図を Fig. 6-6-4-1 に，優勝表彰を Fig. 6-6-4-2 に示す。



Fig. 6-6-4-1 0m goal in ACTS2020

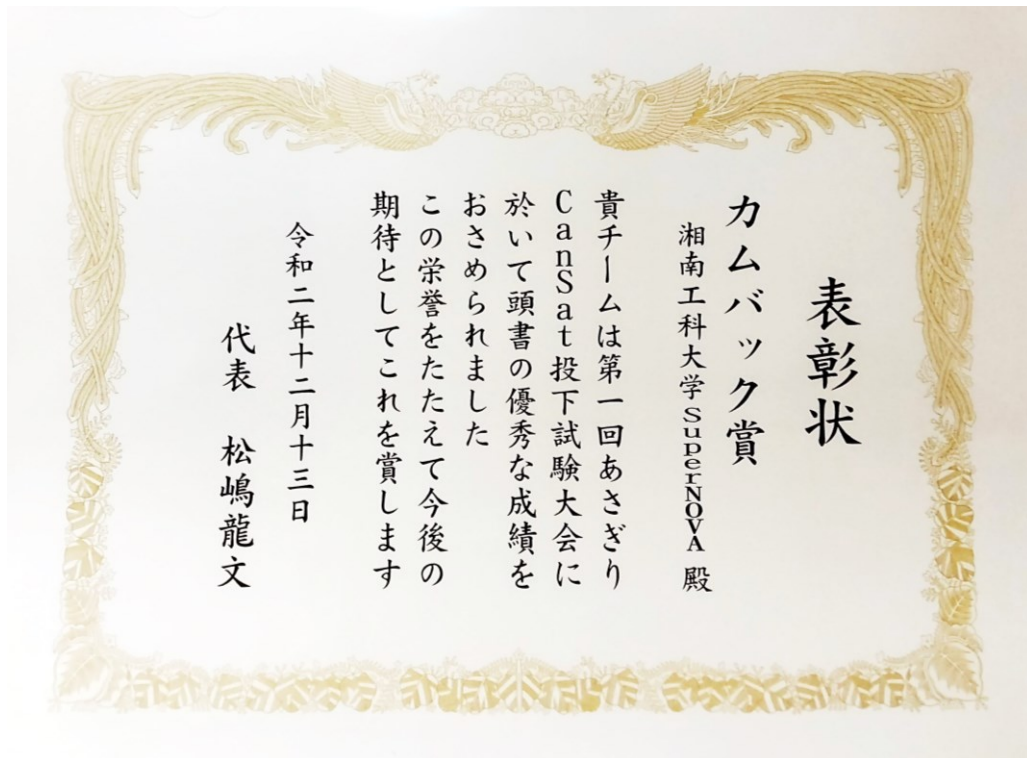


Fig. 6-6-4-2 ACTS2020 Comeback Award

6-7 結論

我々は ARLISS 2019 で CanSat チームとして初めてゴール認識に Deep Learning を適用し、3 回のロケット打ち上げで 3 回全てゴールまでゼロ距離誘導を成功し、Overall Winner 1 位となり優勝した。しかし、同時にゴール地点から 6~7 m 程度離れるとゴールを認識できないことがあり、その場合はゴールを見つけるまでに数十分もの時間がかかることがあり、この問題を解決する必要があることが分かった。そこで、従来は ROI を水平方向へ 5 分割まで行っていた認識アルゴリズムに、さらに 6 分割から 9 分割まで分割数を増やすことに加え、さらに CanSat のピッチ変化に対応するために ROI を上下に移動させて認識させる手法を提案した。

このように ROI 分割数を増やせば遠距離での認識率が向上することは推測できるが、実際に我々の目標とする 10 m 離れた位置からの認識率がどうなるのか、また ROI の上下移動はどれほど認識率が向上するのかを実験により調べた。その結果、6 分割では従来の 5 分割と比べて全く認識率は向上しなかったが、7 分割以上では 10 m での認識率が向上し、分割数を大きくすればするほど認識率が高くなることを確認した。7 分割で上下移動あり場合、10 m 離れていても直射日光時には従来手法では 48% だった認識率が 84% まで向上し、日陰時には従来手法では 0% だった認識率が 64% まで認識率が向上した。このように提案手法はゴールから 10 m の地点での認識率向上に有効であることを示した。

ROI 分割数を増やすと計算時間が増えるため、実際の計算時間を測定した。その結果、従来手法では約 2.94 s の認識時間が、提案手法の 7 分割のみ上下移動なしで約 5.51 s であり、7 分割上下移動ありでは約 10.5 s であることが明らかになった。最も計算時間の長い 9 分割上下移動ありの場合、約 12.4 s であった。

Deep Learning により画像から特定の物体の位置を検出する場合、我々のように Image Classification の ROI を移動させる手法の他に、一般的には Object Detection が用いられる。軽量の Object Detection モデルとして SSD MobileNet V1 と提案手法とで認識率の比較を行った。その結果、ゴールから 6m 以上の距離では、SSD MobileNet V1 よりも提案手法の方が認識率が高く、有効な手

法であることを確認した。また計算時間も SSD MobileNet V1 の計算時間約 12.3 s であり、提案手法で最長の計算時間である 9 分割上下移動ありの約 12.4 s と同等であった。提案手法はゴールに近ければ近いほど計算時間は短くなるが、SSD MobileNet V1 は常に一定の計算時間がかかるため、計算時間に関しても提案手法の方が有効であることを示した。

これら実験による認識率向上や計算時間が実際の CanSat コンテストでも有効であるか実証評価するため ACTS 2020 に参加した。アルゴリズムは提案手法の中で最も低い認識率である 7 分割上下移動あり (7UD) を用いた。その結果 3 回実験を行い、3 回ともゼロ距離ゴールを達成した。そのうち 2 回はゴールから約 10 m 離れた地点で GNSS 誘導制御をやめ、提案手法によるカメラ画像認識のみで誘導制御を行ったが、約 10 m の地点から実際にゴール認識に成功し、2 回ともゼロ距離ゴールを達成した。このことから競技でも誘導制御距離で 1 位となり、カムバック賞を受賞した。これにより実証実験においても提案手法は有効であることを示した。 [39][40]

【参考文献】

- [36] Liu, Wei, et al. "Ssd: Single shot multibox detector." European conference on computer vision. Springer, Cham, pp. 21-37, 2016.
- [37] Y. Yoshimoto and H. Tamukoh, "FPGA Implementation of a Binarized Dual Stream Convolutional Neural Network for Service Robots," J. Robot. Mechatron., Vol.33, No.2, pp. 386-399, 2021.
- [38] Ketkar, Nikhil. "Introduction to keras." Deep learning with Python. Apress, Berkeley, CA, pp. 97-111, 2017.
- [39] Miho Akiyama, Takuya Saito, " Proposal of Long-Distance Goal Recognition Method Using Deep Learning in CanSat," IEEE International Conference on Consumer Electronics – Taiwan (IEEE 2021 ICCE-TW), Sep. 2021.
- [40] Miho Akiyama, Takuya Saito, "A Novel Method for Goal Recognition from 10 m Distance Using Deep Learning in CanSat, " Journal of Robotics and Mechatronics 33(6), pp.1359-1372, Dec. 2021.

Supporting Online Materials:

- [bh] "Raspberry Pi," <https://www.raspberrypi.org/> [Accessed Feb 20, 2023]
- [bi] Tzutalin , "LabelImg," <https://github.com/tzutalin/labelImg/> [Accessed Feb 20, 2023]

第 7 章 Deep Learning を用いた高速目標認識手法の提案

7-1 長距離における Image Classification ゴール認識の速度

CanSat において Deep Learning 画像認識を行う条件下では、CanSat はサイズや重量制限が厳しいため、搭載できるコンピュータの種類は限られている。我々は Raspberry Pi Zero W を使用しており、サイズの的にこれ以上大きなコンピュータの搭載は困難である。Raspberry Pi Zero は性能が低いため、Deep Learning に使用できるモデルも小さなモデルに限られる。そのため、我々は可能な限り小さなサイズの Image Classification モデルを用いた。

Image Classification モデルであるため、画像のどの位置にゴールがあるかは分からないため、画像を水平方向に複数分割した Region of Interest (ROI) を用意し、それぞれ Image Classification 処理を行うことにより、画像の左右どの位置にゴールがあるのかを判定している。ゴールが見つからないと ROI を増やしていくため、1 回のゴール認識に、最大で 40 回ゴール認識処理を行うことになる。このため、1 回の認識で 10 秒以上の計算時間がかかることになる [7]。

実際の CanSat では、GPS でゴール付近まで近づいた後、停止してカメラ画像の Deep Learning 画像認識を行い、ゴールが見つからない場合は少し左へ旋回して停止し、再びカメラ画像の認識を繰り返し行うという動作を行う。ゴールが見つかった場合、そちらの方向へステアリングを向け、数秒走行したのち停止して、上記を繰り返し、画像認識結果がゴール判定可能な画像となったと判定したところでゴール判定を行い、動作を停止する、という制御を行っている。

このため、Deep Learning によりゴール認識は誤認識が少なく、高い精度でゴールの認識が可能であり、より確実にゴールまで 0m まで誘導可能になったが、認識時間がかかるため、ゴールにたどり着くまでに時間がかかるのが欠点であった。特に日本で行われている CanSat 大会には 15 分間の制限時間が設けられているため、計算時間によりこの制限時間をオーバーしてしまうことがある問題があった。

そこで、我々は従来よりも圧倒的に計算量が大きいため採用が困難であった、Object Detection モデルである MobileNet SSD V1 及び V2 を用いながら、Coral USB Accelerator[bj]を Raspberry Pi Zero[bi]と連携させることで従来手法を超え

る高い認識率を実現しながら、約 0.3 秒で認識させることに成功した。また提案手法の実証評価のため能代宇宙イベント 2021 及びあさぎり CanSat 投下試験 (ACTS)2021 に参加し、どちらも最短時間記録でゴールまでの距離 0m を達成し、優勝した。

本章では Coral USB Accelerator を Raspberry Pi Zero と連携させることにより、高速でかつ高い精度でのゴール認識を可能にし、実証評価でも高い成果を示した手法について述べる。

7-2 では Deep Learning を用いたゴール認識の従来手法について述べる。7-3 では提案手法について述べる。7-4 では提案手法の有効性について、認識率及び計算時間について様々な評価を行った結果について述べる。7-5 では能代宇宙イベント及びあさぎり CanSat 投下試験による提案手法の実証評価について述べる。7-6 で結論を述べる。

7-2 第 6 章による従来手法

7-2-1 Image Classification モデル

ここでは Image Classification モデルを使用したゴール認識の従来手法について述べる。CanSat に搭載可能なコンピュータはサイズ及び重量の制約から限られており、我々は Raspberry Pi Zero W を用いている。このコンピュータは Linux が動作する組み込みコンピュータとしてはほぼ最小サイズのものであるが、性能も最低レベルである。このため、VGG16 等の[8]一般的な Deep Learning モデルを使用すると、認識に時間がかかり過ぎてしまったり、メモリ容量をオーバーしてしまったりすることがあるため、色を用いたゴール認識手法よりも高い認識精度が出せるモデルの中で、可能な限り小さなモデルを使用する必要がある。以上の検討の結果、我々は Fig. 7-2-1-1 のモデルを使用することにした。このモデルの構造は Yann LeCum らの LeNet モデル[9]を CIFAR-10[]向けに変更したものである。2D convolutional layer の padding 処理については、実験の結果、padding する方が良いため、全ての 2D convolutional layer で padding 処理を行った[41]。このモデルの入力画像解像度は 32x32pixels, RGB 3ch フルカラー画像である。

Image Classification モデルの学習には 3 通りの画像を用いた。Fig. 7-2-1-2 (a)に示す通常のゴール画像 1703 枚，Fig. 7-2-1-2 (b)に示すゴール判定可能な画像 277 枚，及び Fig. 7-2-1-2 (c)に示すゴールが写っていない画像 1441 枚を用いて学習を行った。モデルのプログラミングには Keras を使い，バックエンドには Tensor Flow も使用した。学習は 15 EPOCH 行った。学習結果を Fig. 7-2-1-3 に示す。accuracy は学習に使用したデータの認識率であり，val_acc は学習に使用していないデータの認識率である。Accuracy は約 97%，val_acc は約 99%の認識率となっている。

Layer (type)	Output Shape	Param #
L1_Conv2D_1 (Conv2D)	(None, 32, 32, 32)	896
L2_Conv2D_2 (Conv2D)	(None, 32, 32, 32)	9248
MaxPooling_1 (MaxPooling2D)	(None, 16, 16, 32)	0
Dropout_1 (Dropout)	(None, 16, 16, 32)	0
L3_Conv2D_3 (Conv2D)	(None, 16, 16, 64)	18496
L4_Conv2D_4 (Conv2D)	(None, 16, 16, 64)	36928
MaxPooling_2 (MaxPooling2D)	(None, 8, 8, 64)	0
Dropout_2 (Dropout)	(None, 8, 8, 64)	0
Flatten (Flatten)	(None, 4096)	0
L5_Dense_1 (Dense)	(None, 512)	2097664
Dropout_3 (Dropout)	(None, 512)	0
L6_Dense_2 (Dense)	(None, 3)	1539
Total params: 2,164,771		
Trainable params: 2,164,771		
Non-trainable params: 0		

Fig. 7-2-1-1 Structure of deep learning model



(a) Goal recognition



(b) Goal judgement



(c) No goal

Fig. 7-2-1-2 Examples of images used for training

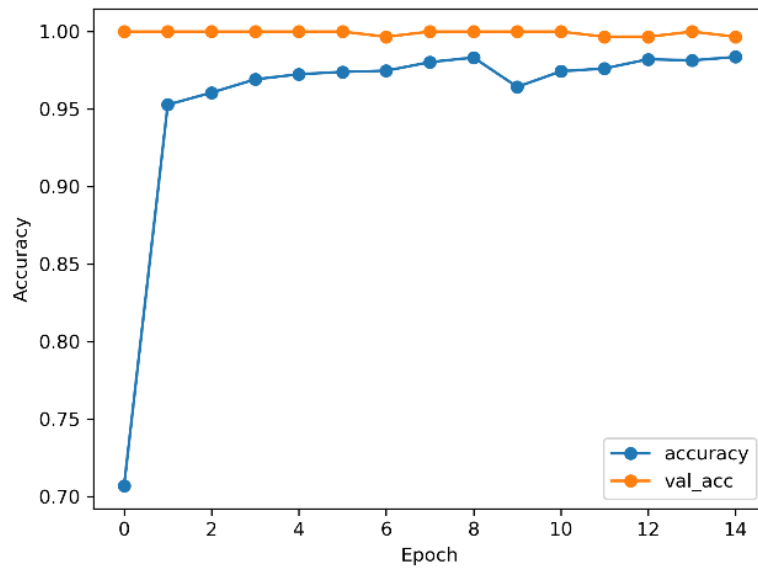


Fig. 7-2-1-3 Model Training Results (Accuracy)

7-2-2 仕様の整合性の維持

ここで用いているモデルは Image Classification モデルであるため、Fig. 7-2-1-2 に示したどの種類の画像なのかは分かるが、ゴールが画像の左右どの位置にあるのかが分からない。これが分からないとゴールの方向へと CanSat を誘導することができない。

そこで我々は Fig. 7-2-2-1 のように画像を水平方向に 3 分割、5 分割、7 分割した ROI を用い、それぞれの ROI に対して Image Classification 処理を行うことにより、画像の左右どの方向にゴールがあるのかを判別させる手法を用いている。

最初にゴール判定可能な状態かを調べるために、ROI 分割なしの全画像で認識処理を行う。次にゴールに近いときにゴールが左右どの位置にあるのかを調べるために Fig. 7-2-2-1 に示した通り、水平方向に画像を 3 分割した ROI それぞれで認識処理を行う。ゴールまでの距離が遠き場合は ROI 3 分割では見つからないため、ゴールが見つからないときは、さらに水平方向に ROI を 5 分割して認識処理を行う。それでも見つからないときには、さらに ROI を 7 分割にして同様に認識処理を繰り返す。5 分割以上では ROI が垂直方向に小さくなるため、ROI を垂直方向にも 3 段階で移動させて認識を行う。このため、7 分割まですべて認識処理を行うと、1 回のゴール認識で最大で 40 回のゴール認識処理を行うことになる。この場合、計算時間は Fig. 7-2-2-2 に示す通り、10 秒以上かかる。このゴール認識時間は、日本で行われている CanSat コンテストの制限時間 15 分からは、ほぼギリギリの計算時間であり、我々は実際に 0m ゴールを達成しながらも、15 分間を少しオーバーしてしまったこともある。このようなことから、より短時間で高い精度でゴール認識が可能な手法が切望されていた。



(a) Left

(b) Center

(c) Right

Fig. 7-2-2-1 Recognition examples in 3 divisions

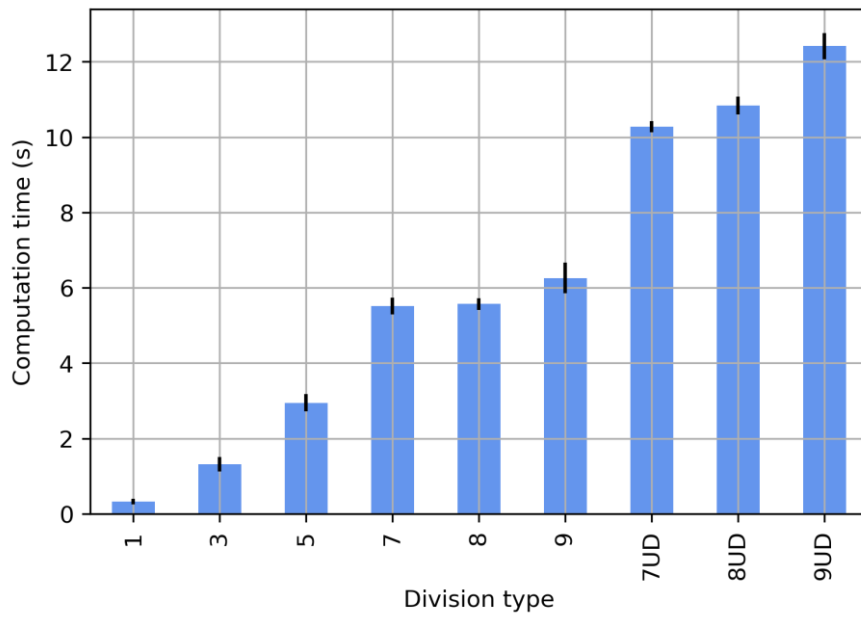


Fig. 7-2-2-2 Computation time

7-3 提案手法

7-3-1 エッジデバイスでの機械学習アクセラレータ

従来手法では、認識時間を極力短くするため、小さな Image Classification モデルを用いていた。しかし、CanSat の制御には画像の左右どの位置にゴールがあるのかが分かる必要があるため、Object Detection モデルを用いて、ゴールが画像のどの位置に、どの大きさで存在するのかが分かる方が良い。しかし Object Detection モデルは Image Classification モデルよりもずっと複雑であるため、Raspberry Pi Zero では認識に時間がかかりすぎる問題があった。

そこで、我々は小型の Deep Learning 画像認識アクセラレータである Coral USB Accelerator を用いることにより、これら問題を解決できるのではないかと考えた。Coral USB Accelerator はコンピュータに USB で接続することにより、機械学習を高速化するアクセラレータであり、小型軽量であるため、エッジデバイス向けに開発された機械学習用アクセラレータである。Coral USB Accelerator はニューラルネットワークに特化した行列プロセッサである Google Edge TPU の構造が採用されており、乗算が行われるたびにその結果が次の乗算アキュムレータに渡される構造であり、行列乗算処理中にメモリアクセスの必要がなく高速な処理が可能となる。サイズは 65 mm×30 mm でありながら、性能は 1W あたり 2TOPS の動作が可能である。

Raspberry Pi Zero にこの Coral USB Accelerator を接続することにより、Object Detection がどれくらいの計算時間で可能になるのか、またどれくらいのゴール認識精度が出せるのかについて調査した。

7-3-2 Object Detection 学習

Coral USB Accelerator では、object detection モデルを自分で用意したオリジナル画像で再訓練するための方法が用意されている。訓練用環境をまとめた Docker が用意されており、そのソースコードを解読して自分のオリジナル画像で訓練させることができるようになっている。訓練用画像には画像内のどこにどういった物体があるかを指定するアノテーション作業が必要になる。このため、我々は訓練用に用意した画像約 5000 枚をすべて手作業でアノテーションを行った。アノテーション作業には LabelImg を用いた。当初ゴールのみをアノテーションして認識させたところ、人間もゴールと誤認識することが多かったため、人間もアノテーションに加え、最終的にその他認識できると便利なものを加えて、Table 7-3-2 に示した 5 種類の物体についてアノテーション作業を行った。goal cone は CanSat においてゴール地点に置かれるトラフィックコーンである。Person は人間であり、将来的には人間を避けることにも使用可能である。balloon は日本の CanSat 大会で CanSat を上空 50 ~ 100m に上昇させるために用いられる気球である。この気球が赤色の場合があるが、その場合ゴールだけを学習させていると、地上に戻ってきた気球をゴールと誤認識して、気球の方向へと誘導するのを防ぐために、気球は別のものだと認識させるために学習させている。parachute は上空から地上へ CanSat を落下させるときに減速装置として使用しているものである。地上へ落下後、CanSat が走行開始した後に、パラシュートの方向へと走行していき、CanSat 本体がパラシュートに絡まり走行不能に陥ることがある。それを防止するために、走行開始後にはパラシュートを画像認識して、見つかった場合最大限にパラシュートの見つかった方角とは逆に旋回してパラシュートへの絡まりを防止している。現在は色によりパラシュートを認識しているが、将来的には object detection でパラシュートの認識が可能かを調べるために学習させている。

最後に blue cone は、ゴールに置かれるトラフィックコーンの青色版が、あさぎり CanSat コンテスト等一部の CanSat コンテストでは、フィールドの外形部分に置かれている。青色のコーンを別のものとして学習させていないと、形状からゴールのコーンと誤認識してしまう可能性があるため、青色コーン

も訓練させている。

Docker の訓練用モデルには， object detection として SSD[11] MobileNet V1[42]及び SSD MobileNet V2[43]の 2 種類が用意されている． SSD MobileNet V1 は SSD のベースネットワークの VGG16 をモバイル端末向けに軽量化した MobileNet V1 に置き換えたモデルである． SSD MobileNet V2 は SSD MobileNet V1 と比較し同等の精度もしくはより高い精度で，より高効率なモバイルモデルを構築したモデルである．これらモデルは入力サイズが 300x300x3 であり， COCO データセットで 90 objects を認識できるように訓練済みである．本研究ではこれら 2 つのモデルそれぞれで同様に訓練させた．

また，訓練にはモデル whole layer を訓練する手法と，モデルの last few layers のみを再訓練する手法が選べるが，本研究では last few layers のみを再訓練する手法を用いた．訓練ステップ数は 1000 ステップとした．

Table 7-3-2 Trained annotation type

object name	description
goal cone	CanSat のゴール地点に置かれるトラフィックコーン
person	人間
balloon	日本の CanSat 大会でロケットの代わりに使われる気球
parachute	CanSat 落下時に用いられるパラシュート
blue cone	一部大会でフィールド外形に置かれる青色のコーン

7-3-3 コンパイル手法

訓練済みのモデルは、最初に TensorFlow Lite 用にコンパイルする必要がある。このコンパイルされたモデルを使うと、Coral USB Accelerator を使わず、通常の TensorFlow Lite として CPU で実行される。

Coral USB Accelerator で使用するには、さらに Edge TPU Compiler にてコンパイルする必要がある。Raspberry Pi Zero では、初期バージョンのライブラリしか用意されていないため、コンパイル時には、Edge TPU runtime compiler version に 1.0 を指定してコンパイルした。

Coral USB Accelerator は Raspberry Pi Zero には正式には対応していないが、初期バージョンである Runtime version 10 のアンオフィシャルライブラリがあるため、このライブラリを使用した。

7-4 評価

7-4-1 認識率の評価

提案手法を用いたゴールからの距離における認識率を調べた。実験に使用した画像データを Table 7-4-1 に示す。CanSat ではゴール地点にトラフィックコーンが置かれるため、市販の一般的な赤色のトラフィックコーンを用いて撮影を行った。CanSat では、トラフィックコーンがそのまま使用される場合と、白線付きのトラフィックコーンが使用される場合があるため、白線なしと白線ありの2種類の画像を用意した。トラフィックコーンの撮影は、ゴールからの距離がほぼ 0m, 0.5m 及び 1m から 15m まで 1m 感覚で撮影を行った。撮影には実際に CanSat で使用されている Raspberry Pi Camera v2 を使用した。撮影時の天候は晴れで、CanSat コンテストが行われる天候の良い状態と似た状態を選んだ。

Table 7-4-1 Goal image shooting conditions.

Place	Nishiura coast
Lighting conditions	Direct sunlight
Resolution	1280 × 960 pixels
Camera	Raspberry Pi Camera v2
Goal pylon	Red color standard size. Size: hight about 70cm / width about 39 cm
Image shooting distance	0m, 0.5m, 1m, 2m, 3m, 4m, 5m, 6m, 7m, 8m, 9m, 10m, 11m, 12m, 13m, 14m, 15m
Image shooting direction	8 directions by shifting by 45 degrees
Number of images	Each angle of goal image position: left, middle, and right, 3 images in total 24 images. 288 images at all distances.

7-4-2 トラフィックコーンに白線がないときの認識率

提案手法を用いて、トラフィックコーンに白線がないときの認識率結果を Fig. 7-4-2-1 及び Fig. 7-4-2-2 に示す。Fig. 7-4-2-1 はモデルに SSD MobileNet V1 を用いた認識率であり、Fig. 7-4-2-2 は SSD MobileNet V2 を用いた認識率である。認識率は、認識させる物体を四角で囲ったアノテーション領域と、実際に認識された四角の領域の重なり部分である、Intersection over Union (IoU)を変化させて認識率を調べた。

$\text{IoU} > 0$ は実際の画像のゴール領域と、認識結果の領域が 1 ピクセル以上重なっているときの認識率であり、 $\text{IoU} \geq 0.5$ は面積が 50%以上重なっているときの認識率を示している。

本研究で対象とする CanSat がカメラ画像からゴールコーンを認識する場合、正確にゴールが画像の中のどこにどの位置にあるかは、5m 以上距離が離れているときには、あまり関係がなく、1m 以内の場合は、ゴール直前のため、確実にハンドルをゴールの方向へ切ったり、ゴール判定を行って停止したりする必要があるため、IoU が高いものでも認識率が高い必要がある。

Fig.7-4-2-1 及び 7-4-2-2 のどちらもほぼ同じ結果が得られているが、どちらも $\text{IoU} \geq 0.25$ では、ゴールからの距離が 10m まで、80%以上の認識率を示している。ゴールからの距離が 10m を超えてくると SSD MobileNet V1 の方が V2 よりも認識率が高い傾向がある。ゴールからの距離が 15m 地点では、V1 が約 60%の認識率であるのに対して、V2 では約 40%である。

逆に、距離が近い場合は、IoU が大きい条件でも V2 の方が認識率が高くなる傾向がある。 $\text{IoU} \geq 0.50$ では、0m~4m まで V2 は認識率が 100%なのに対して、V1 では 90%~95%程度に低下している。 $\text{IoU} \geq 0.75$ でも同様の傾向は見られるため、V2 の方が V1 よりも、より正確にゴールコーンの画像内での場所を認識できていると考えられる。しかしその差は僅かであるため、CanSat の実際の制御には余り影響はないと考えられる。

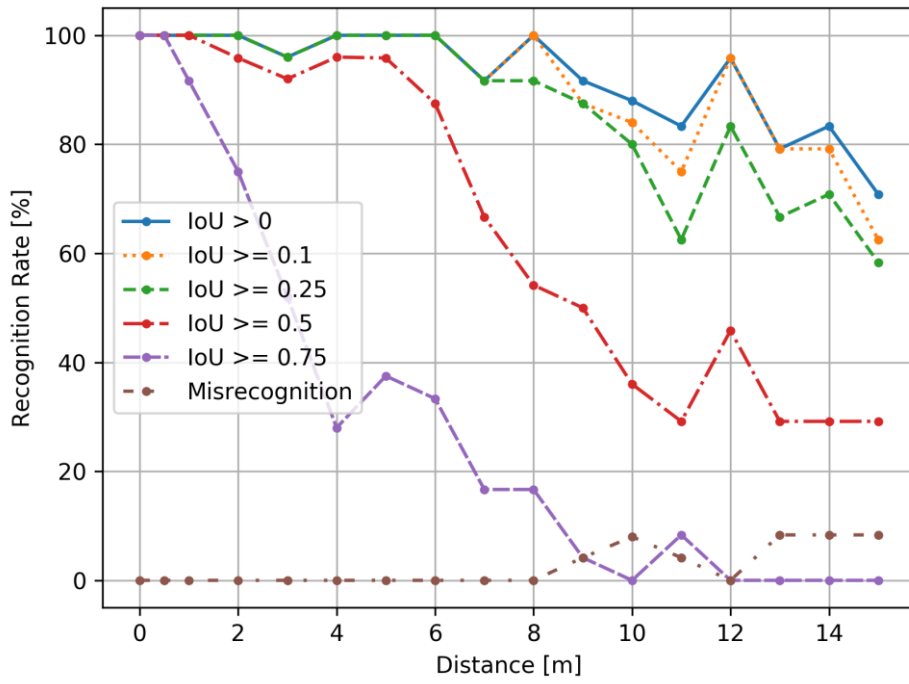


Fig. 7-4-2-1 Recognition rate of SSD MobileNet V1 without

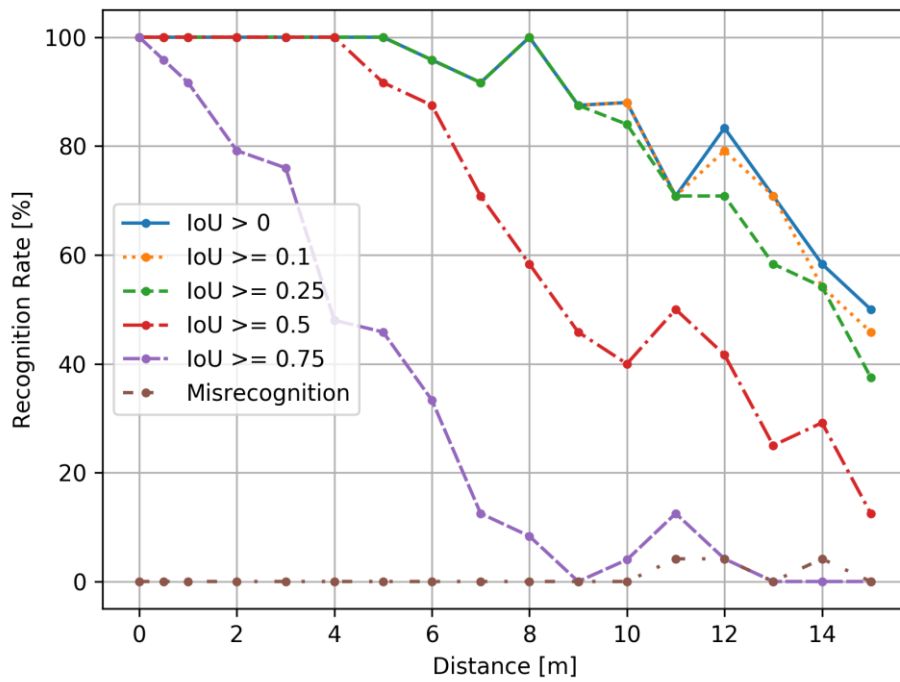


Fig. 7-4-2-2 Recognition rate of SSD MobileNet V2 without
the white lines

7-4-3 トラフィックコーンに白線があるときの認識率

ゴール地点に置かれたトラフィックコーンに白線がある場合の SSD MobileNet V1 による認識率を Fig. 7-4-3-1 及び SSD MobileNet V2 による認識率を Fig. 7-4-3-2 に示す. どちらの場合でも, 白線なしの場合と大きな違いは見られないが, SSD MobileNet V1 では, 白線ありの方が, 10m 以上の長距離における認識率が高くなっている. SSD MobileNet V2 の場合は V1 ほどの差はみられないが, やはり 10m 以上の長距離での認識率は, 僅かに高くなっている.

以上より, SSD MobileNet V1 及び V2 を使用した CanSat のゴール認識においては, トラフィックコーンに白線がある場合もない場合も距離に応じた認識率は, ほぼ同様の傾向を示し, に大きな差は見られないが, SSD MobileNet V1 では 10m 以上において白線がある方が, 認識率が高くなる傾向があることが分かった.

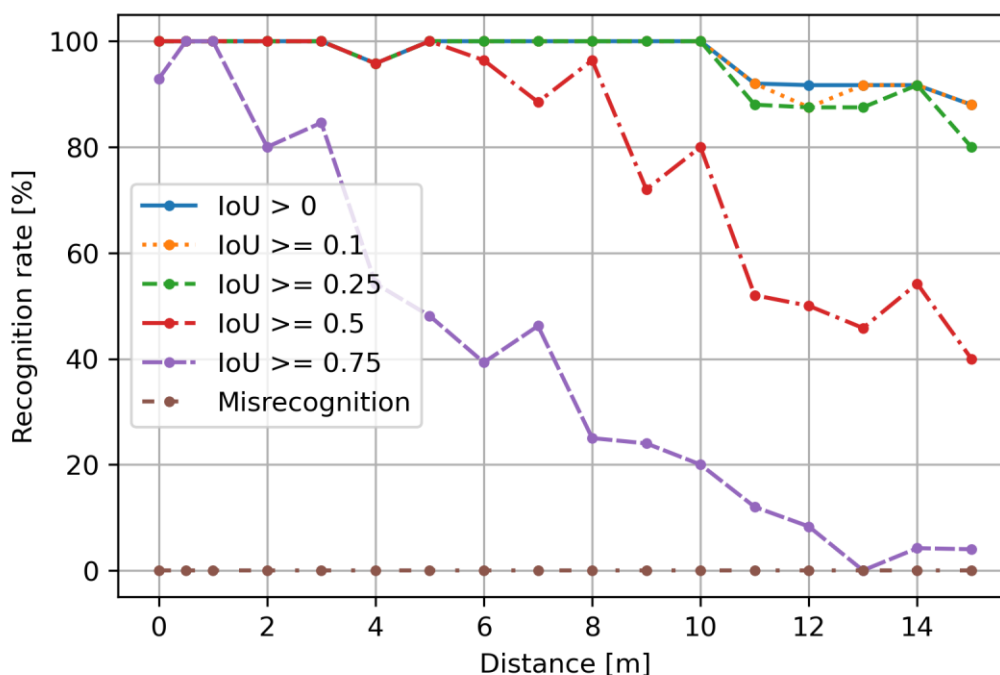


Fig. 7-4-3-1 Recognition rate of the SSD MobileNet V1

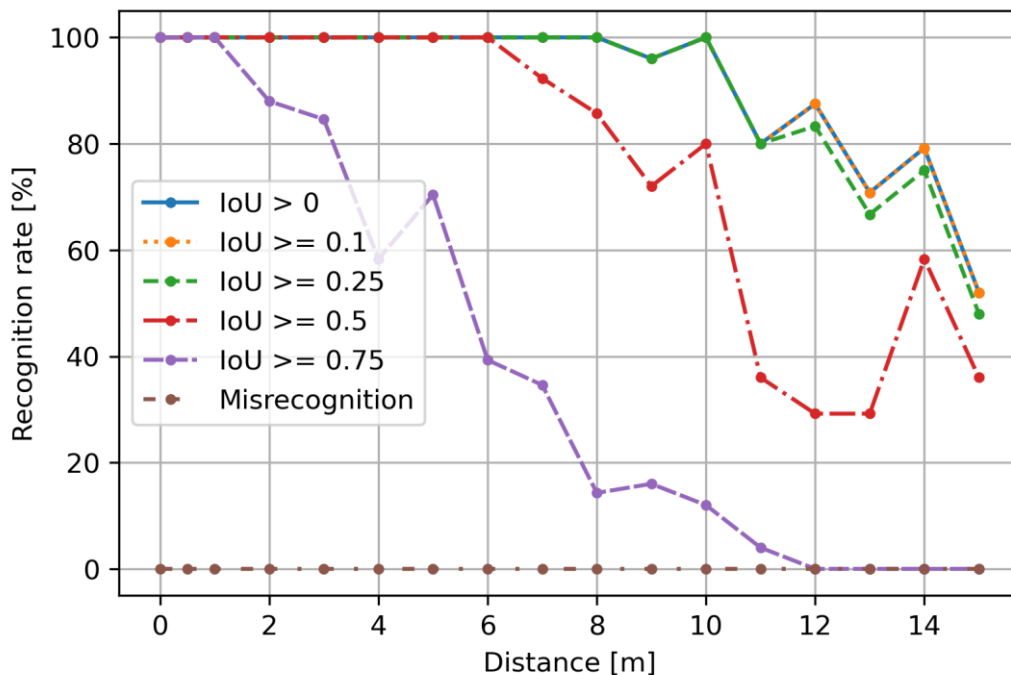


Fig. 7-4-3-2 Recognition rate of the SSD MobileNet V2

7-4-4 従来手法との認識率の比較

提案手法の認識率を，従来手法を比較した．比較は，ゴールに置かれたトラフィックコーンに白線がない場合と白線がある場合の両方で比較した．白線がない場合の結果を Fig. 7-4-4-1 に，白線がある場合の結果を Fig. 7-4-4-2 に示す．提案手法の認識率は，従来手法の認識率の算出法に合わせるため，IoU > 0 のものを用いた．Conventional 5 division は ARLISS 2019 で使用した手法であり，Conventional 7 division with up and down は ACTS 2020 で使用し，我々が従来手法を長距離での認識率を向上させた手法である．

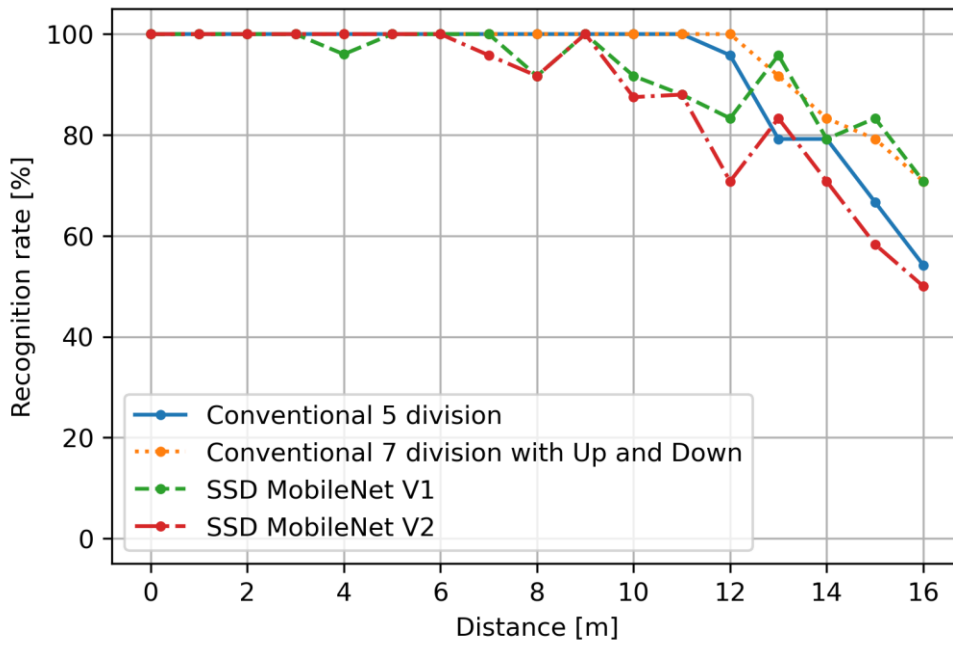


Fig. 7-4-4-1 Comparison of the recognition rate with

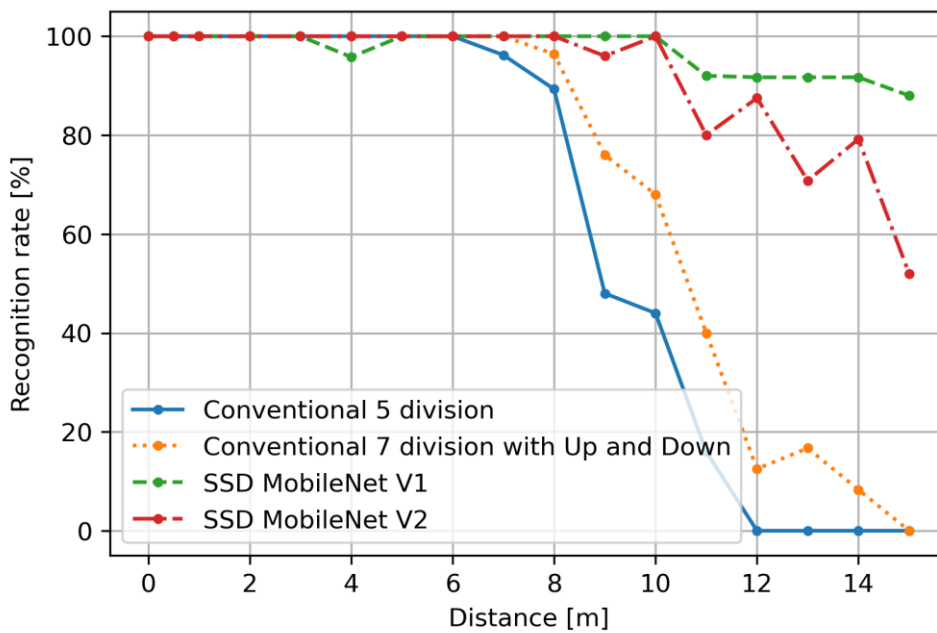


Fig. 7-4-4-2 Comparison of recognition rate with conventional method with white lines on the cone.

7-4-4-1 白線ありの場合の従来手法との比較結果

トラフィックコーンに白線がない場合の認識率比較結果について述べる。Fig. 7-4-3-1 に示した通り，SSD MobileNet V1 は，この Conventional 7 division with up and down に匹敵する認識率を示している。10～15m では多少認識率の低下が見られる。

一方 SSD MobileNet V2 は V1 よりも長距離になるに従って認識率が低下しており，ARLISS 2019 に使用した Conventional 5 division の手法とほぼ同等もしくは若干劣る認識率となっている。

結論として，SSD MobileNet V1 の長距離における認識率は，従来手法で長距離での認識率を向上させた手法とほぼ同等の認識率が得られることが分かった。

7-4-4-2 白線ありの場合の従来手法との比較結果

トラフィックコーンに白線がある場合の認識率の比較結果について述べる。Fig. 7-4-4-2 に示した通り，白線なしの場合と比較して大きな違いが見られた。白線ありの場合，従来手法では，10m あたりから大きく認識率が低下していく，12m 以上では非常に低い認識となっている。しかし，提案手法である SSD MobileNet V1 及び V2 とともに，白線なしのときとほぼ同様の高い認識率を維持している。このため，ゴールのトラフィックコーンに白線がある場合には，従来手法では，10m 以上の長距離において大きく認識率が低下してしまうが，提案手法では白線の有無と認識率とは余り関係がなく，特に長距離において高い人しい率を示すことが分かった。

7-4-5 認識の計算時間

Object Detection の計算に必要な計算時間について述べる。Coral USB Accelerator を使用して SSD MobileNet V1 及び V2 にてゴールの認識に要した時間を Fig. 7-4-5-1 に示す。V1 及び V2 は、使用した SSD MobileNet のバージョンを示しており、その後に記載されている default と max は、Edge TPU runtime インストール時に指定した Coral USB Accelerator の動作周波数を示しており、default が default frequency 設定であり、max が maximum frequency 設定である。太いバーグラフは平均の値であり、縦棒は標準偏差である。Fig. 7-4-5-1 によると、SSD MobileNet v1 では、動作周波数設定が default でも max でも、計算時間は約 0.197 秒であり、ほとんど差が見られなかった。V2 では default の計算時間は約 0.2 秒であり、max での計算時間は約 0.195 秒であったため、max frequency 設定の方が少し計算時間の短縮が見られた。SSD MobileNet V1 でも V2 でも、また動作周波数設定が default でも max でも、計算時間は約 0.195~0.2 秒であり、大きな差は見られなかった。

7-4-6 アクセラレータを使わないときの計算時間

Coral USB Accelerator を使用せずに、Raspberry Pi Zero W の CPU により計算した場合の計算時間を Fig. 7-4-5-2 に示す。SSD MobileNet v1 のときに約 12 秒、V2 のときに約 8.8 秒の計算時間がかかっている。

SSD MobileNet V1 の場合、アクセラレータを使用すると、約 0.197 秒であるため、アクセラレータにより約 61 倍高速になっており、同様に V2 の場合はアクセラレータを使用すると 0.2 秒であるため、約 44 倍高速になることが分かった。

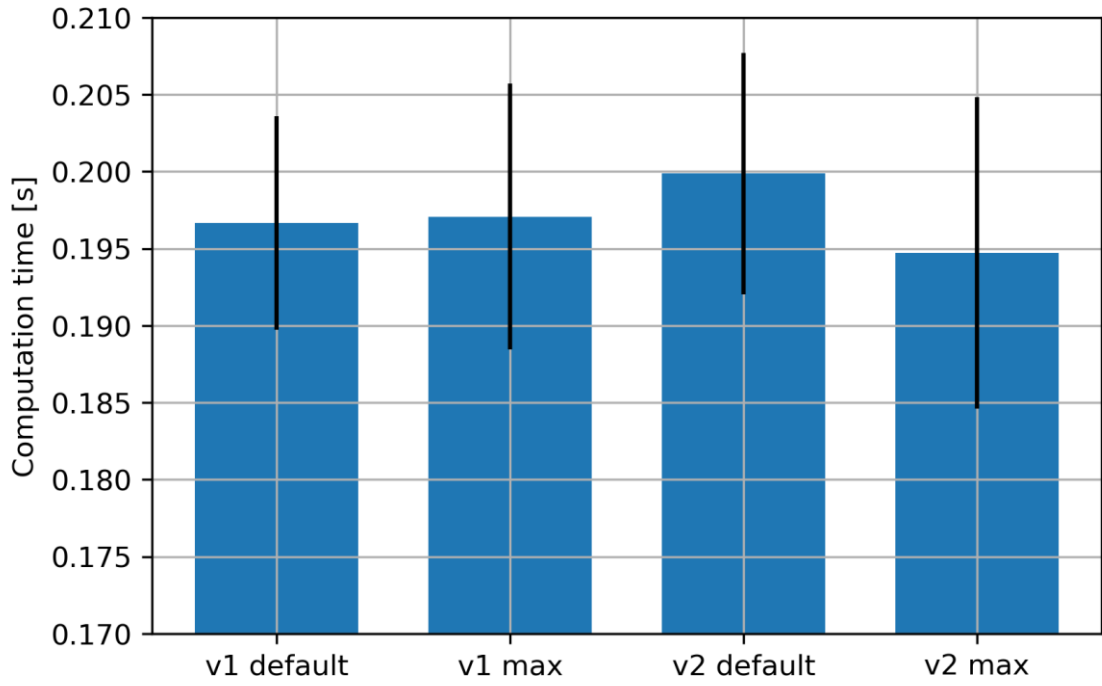


Fig. 7-4-5-1 Computation time with Coral USB Accelerator

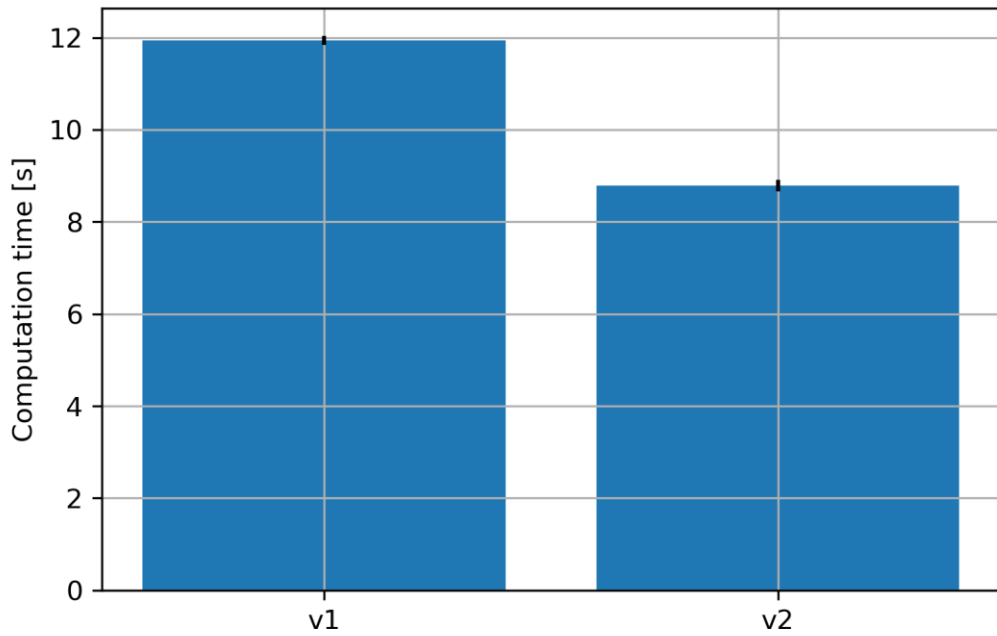


Fig. 7-4-5-2 Computation time without Coral USB Accelerator

7-4-7 従来手法との認識時間の比較

Coral USB Accelerator を用いた SSD MobileNet V1 の認識時間は、Fig. 7-12 より約 0.197 秒であり、V2 の認識時間は約 0.2 秒である。従来手法の認識時間は、Fig. 7-5 より、Conventional 7 divisions with up and down で約 10.5 秒、Conventional 5 divisions で約 2.94 秒である。このことから、提案手法では ARLISS 2019 で使用した Conventional 5 divisions と比較して SSD MobileNet V1, V2 とともに約 15 倍高速化しており、ACTS 2020 で使用した Conventional 7 division with up and down と比較して SSD MobileNet V1, V2 とともに約 53 倍高速化している。それでいながら、従来手法と比較して、認識率が低下する長距離において、ほぼ同等の認識率を有している。

7-5 実証評価

本研究は CanSat コンテストにおいて、より確実に、早くゴール地点まで CanSat を誘導可能とすることを目標としている。そのため、実際に提案手法を用いた CanSat でコンテストに参加し、実証評価を行った。実証評価のため、我々は能代宇宙イベント 2021 とあさぎり CanSat 投下試験(ACTS 2021)の2つの大会に参加した。

7-5-1 能代宇宙イベント 2021

能代宇宙イベント 2021 は 2021 年 11 月 5 日～7 日に開催された。我々は CanSat のカムバック部門に参加した。競技は 2 回行われるが、1 回目はプログラムのバグにより、提案手法による誘導制御が働いていなかったが、結果的にはゴール付近のコンクリートブロックに乗り上げ転倒して停止し、13cm の記録となった。

プログラムを修正した 2 回目は、落下地点から走行開始後、すぐに提案手法によりゴールを発見し、走行の大半を SSD MobileNet V1 による誘導制御によりゴールへと走行を行った。2 回目の走行の様子を Fig. 7-5-1-1 に示す。Fig. 7-5-1-1 (a)のように、CanSat は SSD MobileNet V1 によりゴールを認識し、リアルタイムでステアリングをゴールの方向へと修正して、ゴールへ向けて走

行していく．そして Fig. 7-5-1-1 (b)のようにゴールにくっついたところでゴール判定を行い停止し，ゴールまでの距離 0m を達成した．これにより，ランバック部門で優勝した．

このときの CanSat が実際にゴールを認識した結果を Fig. 7-5-1-2 に示す．Fig. 7-5-1-2 (a)は，ゴールから数 m 離れた地点での認識結果であり，(b)はゴール直前の認識結果である．本コンテストでは，ゴールに置かれたコーンに白線がないタイプが用いられており，問題なく認識できていることがわかる．また，CanSat が地上へと落下した地点からゴール地点までの走行軌跡を Fig. 7-5-1-3 に示す．落下後数秒間は GPS で走行するが，すぐにカメラにより SSD MobileNet V1 がゴールを発見し，画像認識によるゴールへの誘導となった．実際に走行軌跡でも，ゴール方向へ向かってほぼ確実に走行している様子が分かる．またこの軌跡のゴール地点(Target)は GPS により測定した地点であるため，実際にゴール地点まで距離 0m まで行っているが，Fig. 7-5-1-3 では 2m ほどゴールから離れていることになっている．これは GPS の測位誤差によるものである．このため，GPS のみで誘導した場合，ゴール地点から約 2m 離れた地点で停止したことになり，ゴールまでの距離 0m の達成はできなかったことになる．走行開始後，0m ゴールするまでの時間は約 1 分 1 秒であった．またパラシュートから投下後，0m ゴールするまでの時間は 2 分 3 秒であった．競技時間は 15 分間であるが，今回は 2 分 3 秒で 0m ゴールを成功した．0m の記録により能代宇宙イベント 2021 にて優勝した．能代宇宙イベント 2021 優勝の賞を Fig. 7-5-1-4 に示す．



(a) Running towards the goal



(b) Zero-meter mark achieved.

Fig. 7-5-1-1 A scene from Noshiro Space Event 2021



(a) Recognition result a few meters from the goal.



(b) Recognition result just in front of the goal.

Fig. 7-5-1-2 Goal recognition results in Noshiro Space Event 2021.

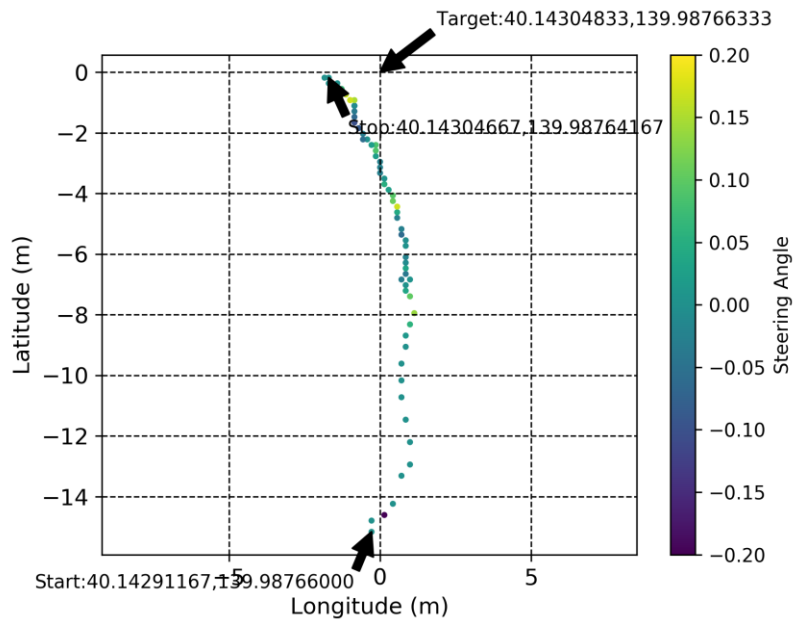


Fig. 7-5-1-3 Running path of the CanSat in NSE 2021

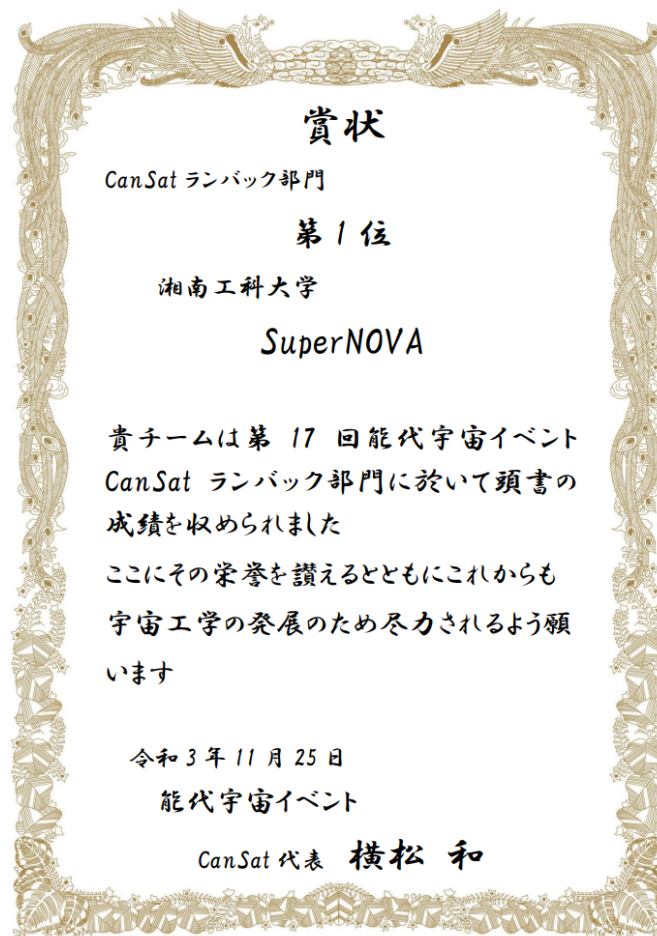


Fig. 7-5-1-4 Comeback Award in NSE 2021

7-5-2 あさぎり CanSat 投下試験(ACTS)2021

あさぎり CanSat 投下試験(ACTS)2021 は、2021 年 11 月 20 日～21 日に、静岡県富士見市あさぎりフードパークにて開催された。本大会は例年アメリカネバダ州ブラックロック砂漠で開催されている ARLISS がコロナ禍により開催できないために、開催された代替大会である。本大会では気球を使い、上空約 50～100m まで CanSat を上昇し、そこから放出してパラシュートにより地上へと落下させる。そこからゴール地点まで CanSat が走行していく。

本大会も競技が 2 回行われる。我々はタイヤの試験も兼ねていたため、1 回目は従来タイヤ、2 回目は 3D プリンタで TPU 樹脂を用いて作成した新タイヤで参加した。1 回目は従来タイヤでは草が生い茂る部分で走行困難であり、最終的にスタックにより走行不能となった。2 回目は改良型の新タイヤで実験したところ、全く問題なく走行できた。このときの走行の様子を Fig. 7-5-2-1 に示す。Fig. 7-5-2-1 (a)はゴール数 m の地点で SSD MobileNet V1 がゴールを認識し、誘導制御しているところである。Fig. 7-5-2-1 (b)は 0m ゴールを達成したところである。

このときの、CanSat のカメラで捉えた画像の SSD MobileNet V1 によるゴール認識結果を Fig. 7-5-2-2 に示す。Fig. 7-5-2-2 (a)はゴールから数 m 離れた地点の認識結果、(b)はゴール直前の認識結果である。ACTS ではゴールのコーンに白線があるタイプが用いられている。白線があるタイプであっても、問題なくゴールの認識ができていることが分かる。

本大会の走行軌跡を Fig. 7-5-2-3 に示す。Start が走行開始地点であり、Stop が停止した地点である。走行開始後 10 秒間はパラシュートの絡まりを防止するため、直進走行をしている。その方向がゴールとは違う方向だったため、途中からゴール方向へと旋回し、ゴール方向へと向きを変えている。その後はゴール付近約 10m まで GPS で誘導制御した後、ゴールから約 10m 地点で CanSat のカメラ画像から SSD MobileNet V1 がゴールを発見して、画像認識による誘導制御を行っている。最終的にゴールのコーンまで距離 0m 地点で停止し、0m ゴールを達成した。Fig. 7-5-2-3 では、Stop 地点は Target 地点とはやはり 3m ほどずれているが、これは GPS の測位誤差によるものである。能代宇宙イベント 2021 と同様に、画像認識による誘導制御がなければ、

0m ゴールは達成できなかったことになる。

走行開始してから 0m ゴールするまでの時間は 2 分 5 秒であった。落下開始してからゴールするまでの時間は 3 分 5 秒であった。競技時間は 15 分間であるが、今回は 3 分 5 秒で 0m ゴールすることに成功した。カムバック部門で 1 位を受賞し、さらに総合優勝も受賞した。Fig. 7-5-2-4 に ACTS2021 での優勝の賞を示す。



(a) Running towards the goal



(b) Zero-distance goal achieved

Fig. 7-5-2-1 A scene from ACTS 2021



(a) Recognition result a few meters from the goal



(b) Recognition result just in front of the goal.

Fig. 7-5-2-2 Goal recognition results in ACTS 2021

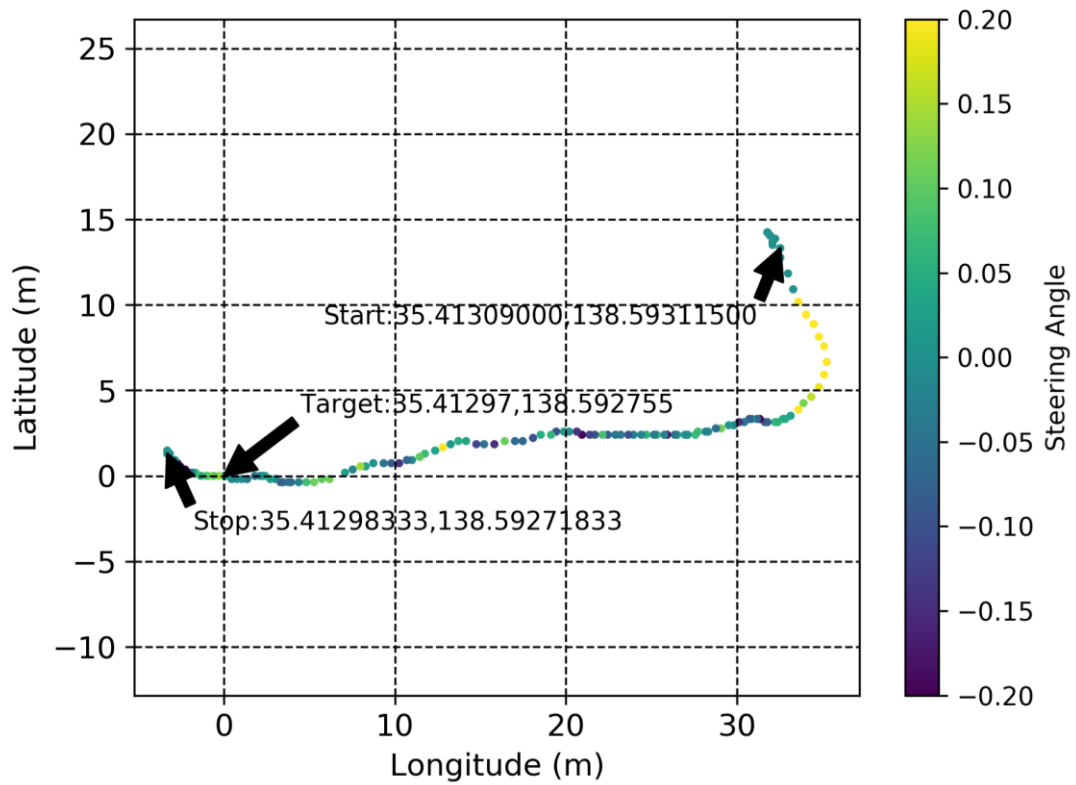


Fig. 7-5-2-3 Running path of the CanSat in ACTS 2021

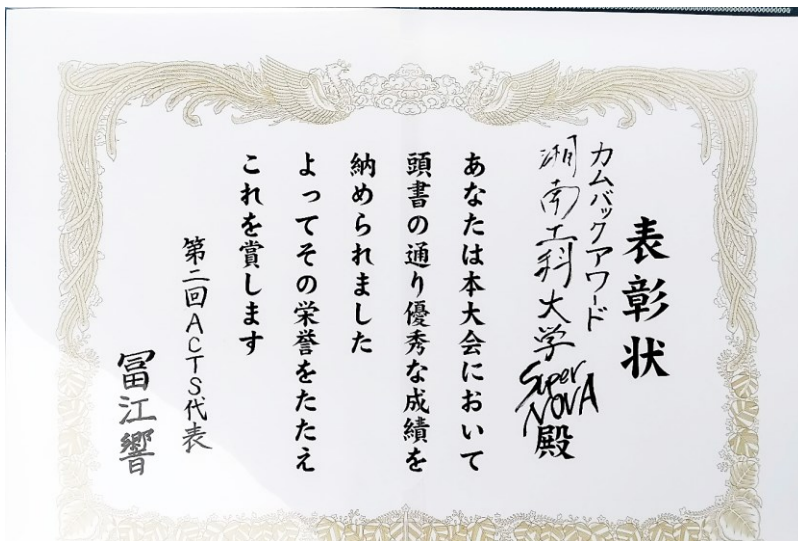
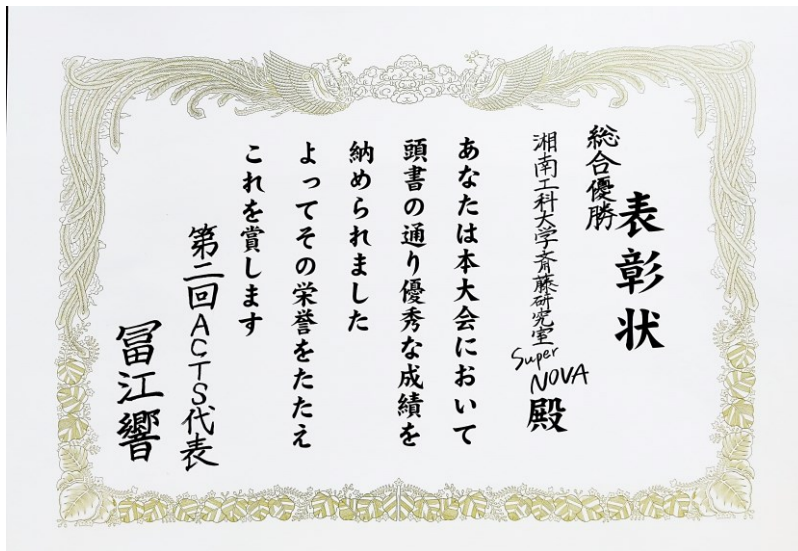


Fig. 7-5-2-4 Comeback Award & Over All Winner

7-6 結論

我々は CanSat でゴールまで確実に距離 0m まで誘導制御するために, ARLISS 2019 で初めて Deep Learning によるゴールコーンの画像認識を行い, 距離 0m のゴールを達成して優勝した. しかし CPU により Deep Learning 処理を行っていたため, Object Detection ではなく, Image Classification タイプの非常に単純なモデルを使ってゴール認識をしていたが, ゴールの認識に一回あたり 10 秒以上の認識時間がかかっていた. そのため, CanSat を停止させてカメラで写真を撮り, Deep Learning で認識を行い, 見つからなければ左旋回して再度認識を行い, 見つかった場合は 3 秒間ゴールの方向へステアリングを回して走行後停止し, 再度認識を行う, という手順を繰り返していた. このため, GPS でゴール近辺数 m に到着後, ゴールまで 0m まで誘導するのに時間がかかってしまう問題点があった. 特に日本で行われている能代宇宙イベントやあさぎり CanSat 投下試験, 種子島ロケットコンテスト等は 15 分間の制限時間があるため, この制限時間ギリギリまで使って 0m ゴールをすることが多く, ACTS 2020 の 3 回目では 15 分間を数秒間オーバーしてしまうこともあった.

この問題を解決するため, 画像認識を高速化するために Coral USB Accelerator を用いた. このアクセラレータでは Image Classification ではなく, Object Detection モデルが使用可能なため, SSD MobileNet V1 及び V2 によるゴールからの距離に応じた認識率を実験により明らかにした. その結果, 従来手法と比べて, ゴールコーンに白線がない場合でほぼ同等の性能が, また白線がある場合には, 従来手法よりも 10m 以上の長距離において高い認識率を有することを明らかにした. また SSD MobileNet V1 と V2 の認識率の比較を行った結果, V1 の方が長距離において高い認識率を有することが分かった. さらに計算時間は約 0.2 秒であることが分かり, CanSat の制御周期である 1 秒よりもずっと短い時間であるため, CanSat を停止せずに走行させたままゴールを認識して誘導制御することが可能になった.

この手法を用いて我々は実証評価のため能代宇宙イベント 2021 及びあさぎり CanSat 投下試験(ACTS)2021 に参加した. その結果, 両方の大会においても 0m ゴールを達成し, 両方の競技で優勝した. また, 投下からゴールまでの時間は, それぞれ 2 分 3 秒及び 3 分 5 秒であり, 従来の 15 分間ギリギリまでかか

っていた 0m までの誘導時間と比較して、大幅に短縮することに成功した。
[44][45]

【参考文献】

- [41] T. Saito and M. Akiyama, "A Study of Image Recognition Rate with and without Padding of 2D Convolutional Layer for Deep Learning in CanSat," 2021 IEEE 10th Global Conference on Consumer Electronics (GCCE), 2021, pp. 816-817, doi: 10.1109/GCCE53005.2021.9621978.
- [42] HOWARD, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint arXiv:1704.04861, 2017.
- [43] SANDLER, Mark, et al., "Mobilenetv2: Inverted residuals and linear bottlenecks," Proceedings of the IEEE conference on computer vision and pattern recognition. 2018. p. 4510-4520.
- [44] Miho Akiyama, Hiroshi Ninomiya, Takuya Saito " Initial Study of the Effectiveness by Using Coral USB Accelerator to Accelerate Goal Recognition on CanSat, " The 20th International Conference on Embedded Systems, Cyber-physical Systems, & Applications (ESCS'22), (In printing), Jul, 2022.
- [45] Miho Akiyama, Hiroshi Ninomiya, Takuya Saito "Method to Achieve High Speed and High Recognition Rate of Goal from Long Distance for CanSat, "Journal of Robotics and Mechatronics, Vol.35 No.1, pp. 194-205, Feb. 2023.

Supporting Online Materials:

- [bj] Coral USB Accelerator <https://coral.ai/products/accelerator/> [Accessed Feb 20, 2023]

第 8 章 結論

CanSat コンテストでは，大学で開発された小型惑星探査機をロケット等で上空へ打ち上げ，上空でロケットから放出され，パラシュートで地表面へと落下させる．落下した CanSat は，自律制御により予め定められている目的地へと向けて走行する．最終的にどれだけ目的地へ近づくことができたかを競い合っている．CanSat 大会は世界中で多数開催されているが，その中でも最も歴史があり，最も大規模に開催されている世界大会は A Rocket Launch for International Student Satellites (ARLISS) である．

ARLISS では当初は大学で開発された CanSat はロケットの打ち上げや上空での放出，地上への落下の大きな衝撃に耐えられず，破損して動作不能になるチームが多く，走行自体が困難な状況であった．また，ARLISS ではゴールから数 km～十 km 以上離れた地点に落下するため，ゴール地点付近までの走行において，地表の轍等でスタックして走行不能になったりするチームが多かった．我々が ARLISS 2017 で測定した結果では，ロケットから放出時に 35G，地上への落下時に 100G 以上の衝撃が CanSat に加わることが分かった．

しかし ARLISS2016 では 1 位はゴールまで 3.76m，2 位が 4.8m，3 位は 5.81m まで近づき，目標位置へ誘導できる CanSat の技術成熟が成された．この 1 位から 3 位までのゴールまでの距離の差は GNSS の測位誤差によるものである．実際に我々のチームはゴールまでの距離が 5.81m であったが，GNSS のログデータでは 0.4m まで近づいたことになっていた．

以降，CanSat コンテストでは GNSS の測位精度を超えて，いかにしてゴールまで近づくことができるか，またより惑星探査の機能を搭載した高度なミッションを行いその性能を競う大会となった．本研究の目的として，従来 GNSS 位置情報による誘導制御であったが，より目的位置へ誘導し，惑星探査を実現する手法として物体を認識する画像認識技術の確立を行った．

第 3 章では実際の宇宙開発で採用されているロケット搭載要件などを満たした設計であるか再現実験により証明する安全審査をもとに CanSat の設計評価を行い，安全審査要件を満たした CanSat の設計仕様を詳細に示した．

第 4 章では搭載している通信機が発する電波と GNSS の測位精度との関係に関する研究を行った．XBee を搭載した CanSat と搭載していない CanSat を 2 台

用意し、同時に GNSS の測位実験を行った。その結果、XBee を搭載している CanSat では、搭載していない CanSat と比べて、測位衛星数が低下し、さらに測位までの時間が長くなることを明らかにした。この実験により、搭載通信機が発する電波が GNSS の測位精度に影響があることを明らかにした。

GNSS には数 m の測位誤差が存在するため、GNSS のみではゴール地点まで数 m までしか誘導することができない。そこで GNSS 測位精度を超え、CanSat がゴール付近へ到達後ゴールをカメラで撮影し画像認識することでゴールまでの距離 0m まで誘導制御手法について研究を行った。

CanSat サイズ要件によりマイクロコンピュータには Raspberry Pi Zero W を搭載している。Raspberry Pi Zero は計算能力が低いため、複雑な計算を短時間で行うことができない。そのため、なるべく単純な計算でゴールを認識することが要求される。GNSS で可能な限りゴール地点に近づいた後、CanSat に搭載したカメラで周囲を撮影し、ゴールに置かれているパイロンの赤色を認識して、その赤色の方へ誘導して行き、カメラ画像の赤色が十分な大きさになった時点でゴール判定を行う手法を用いて、ゴールまでの距離 0m まで誘導する手法を提案した。

ゴールの色の画像認識をするため Raspberry Pi カメラで解像度 160×120 ピクセルで画像を撮影する。画像から特定の色部分を認識する場合、RGB データのままでは、例えば同じ色であっても明るさが変化した場合、RGB の範囲は 3 次元空間上の楕円体の内部を示すことになる。これを指定したり計算するのは困難であるため、我々は色と明るさを分離し、色を 2 次元で表すことができる YCrCb に変換し、それぞれのパラメータの範囲をゴールの赤色に合わせて指定することにより、簡単かつ高精度にゴールを認識できるようにした。RGB から YCrCb に変換する次に Y, Cr, Cb の値がゴールコーンの赤色の範囲内かどうかの 2 値画像を生成する。2 値画像の重心座標によりゴールの存在する左右の方向を求めピクセル数によりゴールまでの距離を求めた。

本手法で種子島ロケットコンテスト 2018 で実証実験を行った結果、CanSat はゴールからの距離が約 4m の地点で認識を開始し、ゴールの赤色を認識したピクセル数は約 0.09% であった。その後 2m の地点で約 0.8% となり、その後約 50cm で約 23% になった。CanSat はゴール地点までほぼ 0m の地点で約 72%

となり、ゴール判定して CanSat は停止した。種子島ロケットコンテスト 2018 にて 0m ゴールを達成して優勝し、本手法の有効性を示した。

第 5 章では色を用いた画像認識において、日照条件の問題を解決するため Deep Learning Image Classification 手法での Tensorflow Keras を用いた可能な限り小さなモデルにおいてのゴール認識率や認識速度を調査し従来手法より誘導制御に適していることについて述べた。

色を用いた画像認識手法では計算量が少ないため、非力な組み込み用コンピュータにとって適した手法ではあるが、いくつか問題点があった。なるべく明るさに左右されずゴールの色範囲を指定するために、我々は YCrCb 色空間を用いている。しかし、実際にはゴールに直射日光が当たって真っ白になっている状態や、逆光でゴールが黒っぽくなったり、さらに雲により暗くなるなど、様々な照明条件の変化により、ゴールの色範囲は刻々と変化するため、コンテスト前に調整した色範囲では本番でゴールを見つけられないことが多々生じる。様々な環境光の変化に対応するため、色の範囲を広げると、ゴールではない枯れ草等をゴールと誤認識してしまい、全く違う方向へと誘導されたりする問題がある。これら問題を解決するために、我々は CanSat では初めてゴールの認識に Deep Learning による画像認識を用いた。

CanSat はサイズや重量制限が厳しいため、搭載できるコンピュータの種類は限られている。我々は Raspberry Pi Zero W を使用しており、サイズ制約によりこれ以上大きなコンピュータの搭載は困難である。Raspberry Pi Zero W は性能が低いため、Deep Learning に使用できるモデルも小さなモデルに限られる。そのため、我々は可能な限り小さなサイズの Image Classification モデルを用いた。

学習画像には約 3000 枚の各距離から撮影したゴール画像やゴールにぶつかりゴール判定ができる画像やゴールの写っていないその他画像を用意して学習を行った。その結果、学習結果では約 97%以上の高い認識精度を実現し、計算時間においても最大 2.7 秒であった。また色を用いた画像認識を各大会の色範囲などの設定と提案手法 Deep Learning での様々な日照条件下のゴール画像の認識率を比較した結果、全ての距離において Deep Learning が最も高い認識率を示した。この結果により、従来の色のみでゴールを認識する手法よりも誤認識が少なく高い認識精度を実現した。

この手法の有効性を確認するため、ARLISS 2019に参加し、実証評価を行った。その結果1回目0.05m、2回目及び3回目に0mの記録を達成し、Mission・Accuracy・Over All Winnerの3 Awardにて優勝し、本手法の有効性を示した。

第6章ではImage Classification手法における10m以上の長距離での認識率が大幅に低下する問題があり、従来制御でゴール方向を決定するために用いていた水平方向のRegion of Interest (ROI)分割数を、従来の5分割よりも増やすことによる、長距離での認識率の向上について述べた。

Deep Learningを用いて画像分類を行う場合、そのままではゴールの位置がカメラ画像のどこにあるのかがわからない。そこで水平方向に画像全体の部分領域であるROIを3分割、5分割として複数回画像認識を行うことにより、ゴールの位置を求めている。この手法では、ゴールから約6mまでが問題なく認識できる限界であり、その距離を超えると急速に認識率が低下し、ゴールを見つけられなくなっていく。特にGNSS測位誤差に加え画像認識誘導中に強風により約10m付近まで流され、長距離からのゴール誘導が困難になる場合がある。

そこで、我々は従来のROI5分割ではゴールが見つからない時に、さらにROI分割数を増やして認識を行う手法を提案した。ROI分割数が増えるとROIの上下範囲も狭くなるため、5分割以上ではさらにROIを上下方向に3段階に変化させて認識させることにし、この手法の有効性について評価実験を行った。また、ROI分割数を増やすことにより計算量が増大し、計算時間が増えるため、認識率との関係から適切な分割数及び上下移動のあり、なしについて実験により明らかにした。

水平方向の分割数を増加させ認識率を比較した結果、従来の5分割と比較し、晴天時画像に10m地点において5分割及び6分割で48%の認識率が、7分割では76%に大きく向上しており、8分割では88%、9分割では92%と、分割数が大きくなるに従って向上した。曇天時画像も同様に、10m地点において5分割及び6分割で0%の認識率が、7分割は45%、8分割及び9分割では64%と向上した。

また、ピッチ角度によりゴールが上下に移動することによる認識率低下を防ぐため、ROIを上下に3段階に動かす手法において分割数を増加させ認識率を比較した結果、ROI晴天時では、上下移動のない認識率と上下移動のある認識

率を比較すると、6分割では全く同一であり、上下移動させても認識率は向上していない。10 m 地点において7分割では76%から84%に向上し、8分割では88%のまま変化せず、9分割では92%から96%へと向上した。

曇天時では、上下移動のない認識率と上下移動のある認識率を比較すると、6分割では認識率の向上がなかった。10 m 地点において7分割では45%から64%へと向上し、8分割及び9分割では64%から73%へと向上した。

さらに計算時間の測定を行った結果、従来手法では、ゴールが見つからないときは5分割まで計算していたので、Number of divisions が1+3+5の約2.94 sの計算時間が1回あたり最大でかかっている。提案手法である7,8及び9分割も計算した場合の平均計算時間はそれぞれ約5.51 s, 約5.57 s及び約6.26 sである。従来手法と比べて計算時間が約3秒間増加している。

従来手法の5分割に加えて7,8及び9分割の上下移動したものがそれぞれDivision type が7UD, 8UD及び9UDであり、その場合の平均計算時間は約10.5 s, 約10.9 s及び約12.4 sであった。

ARLISSのように時間制限のない場合は認識率の高い9分割にROI上下移動を用いても良いが、15分などの制限時間がある大会の場合には7分割または7分割にROI上下移動を行い、CanSat ピッチ変化に対応したものが適している。

本手法を実装したCanSatによりACTS 2020に参加し、実証実験を行った結果、ゴール0m地点へ誘導制御に成功し、優勝し、本手法の有効性を確認した。

第7章ではImage Classificationにゴールの認識に時間が最大約15sほどかかり、CanSat競技には15分間という時間制限があるため、このように認識時間が長いと競技時間内にゴールまで距離0mまで誘導制御することが困難になる問題があった。この問題を解決するためにCoral TPU Acceleratorを用いてSSD MobileNet V1でゴールを認識する手法について調査した結果について述べた。

従来手法ではImage Classificationモデルであるため、画像のどの位置にゴールがあるかは分からないため、画像を水平方向に複数分割したROIで、Image Classification処理を行うことにより、画像の左右どの位置にゴールがあるのかを判定している。ゴールが見つからないとROIを増やしていくため、1回のゴール認識に、最大で40回ゴール認識処理を行うことになる。このため、1回の認識で10秒以上の計算時間がかかることになる。

実際の CanSat 誘導制御動作では GNSS でゴール付近へ誘導後、停止し Image Classification を行いゴールが発見できればその方向へ誘導し、できなければ左へ角度を少しずつ変え旋回させ発見するまでその動作を繰り返すため、長距離からの誘導走行には時間がかかり、15 分間の制限時間内にゴール誘導ができない問題があった。

そこで、我々は従来よりも圧倒的に計算量が大きいため採用が困難であった、Object Detection モデルである MobileNet SSD V1 及び V2 を用いながら、Coral USB Accelerator を Raspberry Pi Zero と連携させることで従来手法よりも長距離における認識時間短縮と認識率向上ができないか調査を行った。

画像認識を高速化するために Coral USB Accelerator を用いた。このアクセラレータでは Image Classification ではなく、Object Detection モデルが使用可能なため、SSD MobileNet V1 及び V2 によるゴールからの距離に応じた認識率を実験により明らかにした。その結果、従来手法と較べて、ゴールコーンに白線がない場合でほぼ同等の性能がり、また白線がある場合には、従来手法よりも 10m 以上の長距離において高い認識率を有することを明らかにした。また SSD MobileNet V1 と V2 の認識率の比較を行った結果、V1 の方が長距離において高い認識率を有することが分かった。さらに計算時間は約 0.2 秒であることが分かり、CanSat の制御周期である 1 秒よりもずっと短い時間であるため、CanSat を停止せずに走行させたままゴール認識して誘導制御することが可能になった。

この手法を用いて我々は実証評価のため能代宇宙イベント 2021 及び ACTS2021 に参加し、両方の大会においても 0m ゴールを達成し、両方の競技で優勝した。また、投下からゴールまでの時間は、それぞれ 2 分 3 秒及び 3 分 5 秒であり、従来の 15 分間ギリギリまでかかっていた誘導制御時間と比較して、大幅に短縮することに成功し、本手法の有効性を示した。

以上、本研究では CanSat におけるゴール認識の最適手法について、エッジデバイスである Raspberry Pi Zero W の性能と計算時間との比較により、最も適している手法について調査を行った。ゴールの認識手法について提案手法の有効性を確認できたが、ゴールまで誘導制御を行う前段階である地上走行において、ランバックでは轍や草の障害物で走行不能になり、ゴールまで走行できない問題があり、その走行時における信頼性のある走行制御の手法についても今後検

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

CanSat における **Deep Learning** を用いた誘導制御手法を確立したことにより、惑星探査ローバの設計要件を満たした構造内に搭載可能であるエッジデバイスの **Raspberry Pi Zero** での画像認識誘導制御において、軽量なモデルを用いてどの程度の精度で制御が可能であるかを明らかにした。惑星探査ローバの画像認識の制御手法を確立し、CanSat における探査性能の向上を可能にした。

【研究業績】

○ 論文

【原著論文】

査読付き

・ Miho Akiyama, Hiroshi Ninomiya, Takuya Saito "Method to Achieve High Speed and High Recognition Rate of Goal from Long Distance for CanSat, "Journal of Robotics and Mechatronics, Vol.35 No.1, pp. 194-205, Feb. 2023.

(第 7 章)

・ Miho Akiyama, Takuya Saito, "A Novel Method for Goal Recognition from 10 m Distance Using Deep Learning in CanSat, " Journal of Robotics and Mechatronics 33(6), pp.1359-1372, Dec. 2021.

(第 6 章)

・ 秋山実穂, 斎藤卓也, " Deep Learning を用いた CanSat をゴールまで距離 0m へ誘導する手法の研究, "電子情報通信学会論文誌 D J104-D(7), pp.540-551, July 2021.

(第 5 章)

・ Takuya Saito, Miho Akiyama, " Development of Rover with ARLISS Requirements and the Examination of the Rate of Acceleration that Causes Damages During a Rocket Launch, " Journal of Robotics and Mechatronics, Vol. 31, No. 6, pp. 913-925, Dec. 2019

・ 斎藤卓也, 秋山実穂, "惑星探査ローバの機体設計及び ARLISS2017 における動作ログの詳細分析,"愛知工科大学紀要第 16 巻, pp. 17-26, Mar. 2019.

・ 斎藤卓也,秋山実穂,"惑星探査ローバの ARLISS2016 における動作ログの詳細分析,"愛知工科大学紀要第 15 巻, pp.19-25, Mar. 2018.

○ 講演・口頭発表等

【査読付国際会議】

・ Miho Akiyama, Hiroshi Ninomiya, Takuya Saito " Initial Study of the Effectiveness by Using Coral USB Accelerator to Accelerate Goal Recognition on CanSat, " The 20th International Conference on Embedded Systems, Cyber-physical Systems, & Applications (ESCS'22), (In printing), Jul. 2022.

(第 7 章)

・ Takuya Saito, Miho Akiyama, " A Study of Image Recognition Rate with and without Padding of 2D Convolutional Layer for Deep Learning in CanSat," 2021 IEEE 10th Global Conference on Consumer Electronics (GCCE) ,pp.881-882, Oct. 2021.

・ Miho Akiyama, Takuya Saito, " Proposal of Long-Distance Goal Recognition Method Using Deep Learning in CanSat," IEEE International Conference on Consumer Electronics – Taiwan (IEEE 2021 ICCE-TW), Sep. 2021.

(第 6 章)

・ Miho Akiyama, Takuya Saito, "Study on Parachute Entanglement Prevention Method Using Image Recognition in CanSat," The 2020 International Conference on Computational Science and Computational Intelligence (CSCI), Dec. 2020.

・ Miho Akiyama, Takuya Saito," A Novel CanSat-Based Implementation of the Guidance Control Mechanism Using Goal-Image Recognition," 2020 IEEE 9th Global Conference on Consumer Electronics (GCCE 2020), Oct. 2020.

(第 4 章)

・ Miho Akiyama, Takuya Saito, "Goal Recognition Using Deep Learning in a Planetary Exploration Rover Developed for a Contest," IEEE International Conference on Consumer Electronics – Taiwan (IEEE 2020 ICCE-TW), Sep. 2020.

(第 5 章)

・ Miho Akiyama, Takuya Saito, " Influence of Radio Waves Generated by XBee Module on GPS Positioning Performance," IEEE International Conference on Consumer Electronics – Taiwan (IEEE 2020 ICCE-TW), Sep. 2020.

(第 4 章)

【国内会議】

・ 秋山実穂, "CanSat 競技完全制覇までの道のり", 第 18 回情報科学技術フォーラムコンペの覇者, 講演 (3), Sep. 2019

・ 斎藤 卓也, 秋山 実穂, "種子島ロケットコンテストで優勝した惑星探査ローバの開発", 日本機械学会ロボティクス・メカトロニクス講演会'19, Jun. 2019.

・ 秋山実穂, 斎藤卓也, "0m ゴール誘導を実現する CanSat 向け画像認識及び制御手法の研究", 2019 年電子情報通信学会総合大会 ISS 特別企画「学生ポスターセッション」, Mar. 2019.

・ 秋山実穂, 斎藤卓也, "コンテスト用惑星探査ローバにおけるディープラーニングを用いたゴール認識の研究", 情報処理学会第 81 回全国大会, Mar. 2019.

・ 秋山実穂, 斎藤卓也, "種子島ロケットコンテスト 2018 でゼロメートルゴールを達成した CanSat のログデータの分析", 情報処理学会第 48 回組込みシステム研究発表会, 論文番号 8, 東海大学, Jun. 2018.

- ・ 斎藤卓也, 秋山実穂, "小型自律移動ロボットの設計と ARLISS2017 による実証評価,"ロボティクス・メカトロニクス講演会'18 講演論文集, 2A2-C07,北九州国際コンベンションゾーン, Jun. 2018.
- ・ 秋山実穂, 斎藤卓也, "惑星探査ローバの ARLISS2017 における動作ログの分析,"電子情報通信学会総合大会 2018, B-2-19,東京電機大学, Mar. 2018.
- ・ 斎藤卓也,秋山実穂, "惑星探査ローバにおける XBee の GPS 測位精度影響の研究,"情報処理学会組込シンポジウム ESS2017, pp.114-225, Aug. 2017.
- ・ 斎藤卓也,秋山実穂, "惑星探査ローバの ARLISS2016 における動作ログの分析,"情報処理学会組込シンポジウム ESS2017, pp.112-113, Aug. 2017.
- ・ 斎藤卓也,秋山実穂,前田一朗,宮里一慶,山川啓介,田宮直,"小型自律移動ロボットの設計と ARLISS による実証評価,"日本機械学会ロボティクス・メカトロニクス講演会 in 福島 2017, 2P2-A12, May 2017.

【謝辞】

本論文の執筆にあたり、斎藤卓也准教授には、愛知工科大学工学部電子制御・ロボット工学科の学部生の時より競技会での引率、研究の着想から、調査、論文執筆まで多くのご指導をいただきました。心から感謝申し上げます。

湘南工科大学電気情報専攻博士後期課程より主指導教員である二宮洋教授には多くのご支援をいただきました。お礼申し上げます。

博士論文の審査において、渡辺重佳教授、三浦康之教授、堀越力教授には本論文の作成にあたり、副査として助言を賜りました。感謝申し上げます。