

# 卒業研究進捗報告

B4 西本 慎吾

2020 年 11 月 17 日

## Abstract

To improve the reliability of nano-satellites, it is essential to fix satellite failures which are generated in design phase and manufacturing phase. Nevertheless, fault analysis depends on human ability or experiences, and the process of identifying failures causes requires considerable human resources and time. This research proposes a new approach to support identifying failure causes of satellite based on the model of signal, electricity and physical interaction between subsystem or components in satellites. The effectiveness of the approach is proved by applying it to a simple satellite model and showing the process to identify failure cause based on the approach.

超小型衛星の信頼性向上のためには、地上試験によって設計や製造過程での不良を事前に発見し、不具合の改修、対策を十分に行うことが重要である。一方で、不具合分析が個人の知識や経験に大きく依存するため、経験が浅いエンジニアや衛星に関する知識の乏しいエンジニアが故障原因の特定を十分に行うことは困難である。

本研究では、電気・信号・物理量の流れで表現したコンポーネント間の接続関係のモデル化、テレメトリ及びコマンドがコンポーネント間を伝達する経路のモデル化を行う。また、これらのモデルを用いて故障原因を特定する方法（確認すべきテレメトリ、打つべきコマンド）の提示を行い、簡易的な衛星モデルを用いて有効性を検証する。

## 1 はじめに

現在、コマンドとテレメトリをベースにして行う衛星の不具合分析を支援する手法を検討している。以下の章では、コンポーネント間の情報の流れを信号や電気、物理現象の流れで表現した接続関係のモデル化、コマンドとテレメトリの衛星内部での情報伝搬の流れのモデル化、そのモデルを用いた不具合分析について説明し、今後の方針を示す。

## 2 研究背景

### 2.1 超小型衛星の信頼性の低さ

超小型衛星の開発が大学や小企業の中で盛んになってきている。これまでは教育目的が主であったが、商用利用や革新的なミッションへの応用も増えてきている<sup>1)</sup>。一方で現状の超小型衛星は中・大型衛星と比較して軌道上での不具合の確率は高く、2002 から 2016 の間に打ち上がった 270 の Cubesat のうち、139 のミッションが失敗している<sup>1)</sup>。

これらの不具合は、大学衛星が宇宙環境での使用を保証されていない民生部品を使用すること多いため、軌道上での部品の故障によって発生すると考えられてきた。しかし、実際には多くが設計や製造過程に起因する不具合であることが知られている<sup>2)</sup>。軌道上での不具合の根本原因に対する調査 (Figure 1) では、民生

部品の品質の不確定性が原因であったものはわずか 17 % であり、それ以外の多くが設計や、地上試験の不足に起因するものである<sup>2)</sup>。

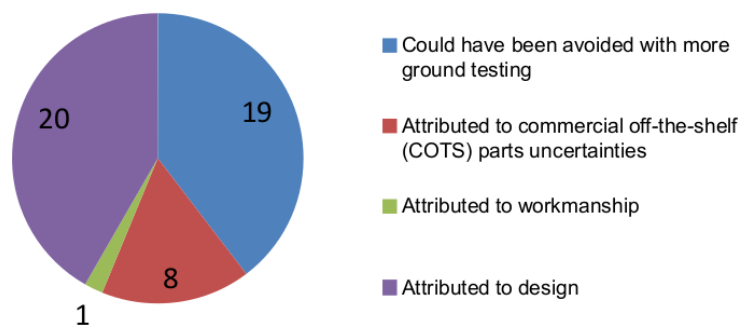


Figure 1: 故障原因に関するインタビュー結果<sup>2)</sup>

また、大学衛星が商用利用や革新的なミッションに挑戦するためには、超小型衛星のメリットであるコストの低さを十分に確保しながら、ほどよい信頼性を実現することが、重要であると考えられている<sup>3)</sup>。故障に設計や製造の不良が含まれていることを考えると、超小型衛星の「ほどよい信頼性」の評価を行うためには、従来用いられてきた各コンポーネントごとの信頼度の組み合わせでは不十分である。そこで、設計・製造・運用における信頼度を加味した評価手法が提案されている<sup>3)</sup>。式 (1) が示すように、この評価手法では製造時の信頼性も重要な要素であると捉えられている。

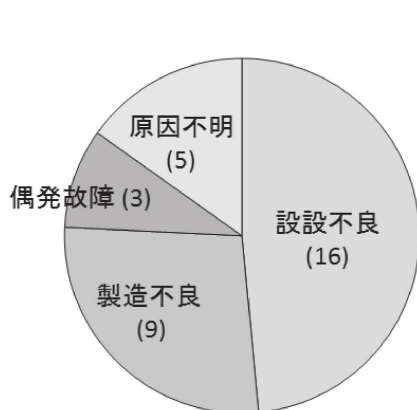
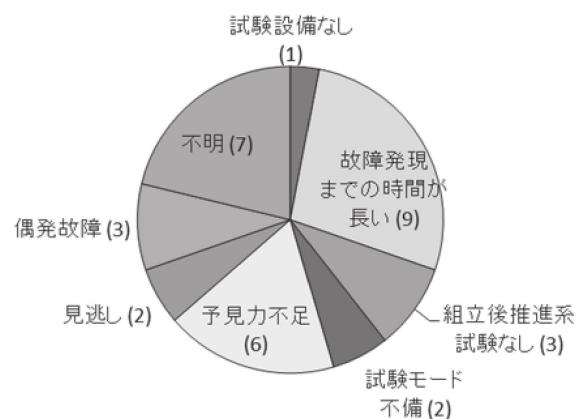
$$R_{sat} = R_{des} \times R_{fab} \times R_{comp} \times R_{op} \quad (1)$$

$R_{sat}$	衛星の真の信頼度
$R_{des}$	設計における信頼度
$R_{fab}$	製造における信頼度
$R_{comp}$	衛星の信頼度 (従来の信頼度)
$R_{op}$	運用における信頼度

## 2.2 地上試験における問題

以上で示したように、不具合の多くが設計、製造などに起因しているという問題がある。一方で、これは超小型衛星開発のみに限られたことではなく、中・大型衛星においても大きな問題となっている。軌道上故障データを分析した結果<sup>4)</sup>(Figure 2)によると、軌道上で偶発的に発生した故障はわずか 11 % であり、それ以外は設計、製造などの開発活動に起因するものであることがわかっている。

また、軌道上で発生した不具合が地上試験で発現しなかった、または発見できなかった原因が以下の Figure 3 のように知られている。試験設備の不足によるものや、故障発見までの時間が長く試験で発見することが現実的で無いものに関しては、コストとリソースの面から試験による対策では限界がある。一方で、試験モードの不備や、発現していたのにもかかわらず発見できなかった不具合に関しては試験に対する習熟度が不足していること、不具合・リスクの分析が不十分であることが推測される<sup>4)</sup>。

Figure 2: 軌道上故障の原因類型の分布<sup>4)</sup>Figure 3: 軌道上故障の要因を地上で発見できなかった原因類型の分布<sup>4)</sup>

### 2.3 不具合分析の難しさ

以上のように、衛星の不具合及びリスク分析を、地上試験で十分に行うことができていないという現状がある。

以下に人による不具合原因分析の大まかな流れを示す。

1. 不具合が起きた際の衛星の状態を保存する。
2. テレメトリから考えられる故障原因の候補を洗い出す。
3. それらの故障の中でテレメトリから分かる情報を元に候補を棄却していく。
4. 更に切り分けが必要な場合はコマンドを送って、それに影響を受けるテレメトリによって判断する。
5. 判断できない場合は、コンポーネントを取り出し直接確認を行う。

分析が不十分になっている原因の一つとして、2の故障原因の候補の洗い出しを網羅的に行うことの難しさがある。組み上げ状態の衛星から得られる情報はテレメトリのみである。この際、衛星の内部状態を理解し、テレメトリから現在の衛星の状態を想像することができなければ、十分に不具合の原因の候補を考えることはできない。本研究室の過去プロジェクト (PRISM) を対象にした研究では、事前に想定していた故障モードの粒度は、山口ら<sup>5)</sup> がモデルを用いて洗い出したものと比較して、不十分であるという結果も出ている。このように、人による故障モードの洗い出しは思いつきによるものなので、考えが及んでいないことが多い。また、分析が不十分になっている原因の一つとして、3, 4の故障原因の切り分け作業の難しさもある。超小型衛星は内部状態が複雑に絡み合っており、一つの不具合に対して非常に多くの故障候補が洗い出されることが想像できる。そのため、多くの故障候補の中から切り分けを行い、最終的な故障を特定するという作業は多くの知識と労力を必要とする作業である。

## 3 本研究での目的

以上を踏まえると、不具合発生時に故障候補を洗い出し、その切り分けを行って原因を特定していく過程に、高い知識と経験が必要になっていることが、衛星の不具合やリスクの分析が不十分になっている原因の一つであると言える。

そこで本研究では、経験が浅く、衛星に関する知識の乏しいエンジニアであっても不具合分析を実施できるような不具合分析支援の手法を提案する。この手法は下記の 3 つの要素で構成されている。

- 衛星内部のサブシステムや要素間の信号・電気・物理的相互作用のモデル化
- コマンド及びテレメトリによって情報が流れる経路のモデル化
- 不具合原因の特定を行うために必要なコマンド及びテレメトリの探索

モデル化に関しては、今回は専

まずは、簡易的な衛星のモデルに対してテストケースを考え、手法の有効性を検証し、有効であることが十分に示すことができれば、本研究室における衛星開発で部分的に実践し有効性を検証することを考えている。

## 4 先行研究調査

上述のように、不具合原因の洗い出しが網羅的にできていないこと、仮説の検証過程が専門家等の知識依存になっていることが不具合分析が不十分になっている原因の一つであった。これらの課題に対して、古くから故障診断システムの研究が盛んに行われており、特にモデルベースで行う手法に関して

モデルを元に不具合診断を行うシステムの研究は古くから行われており、<sup>1)</sup> 正常時のコンポーネント間のつながりモデルに基づく不具合分析手法として GDE が広く知られている。

以下に比較結果を示す。

Table 1: 不具合分析手法の比較

手法	故障網羅性	手法の目的	モデル複雑度
GDE	×	故障仮説生成	低
GDE+ <sup>6)</sup>		故障仮説生成	中
網状故障解析 <sup>5)</sup>		異常モード洗い出し	高
故障オントロジー <sup>7)</sup>	○	故障仮説生成	高
本手法		故障箇所切り分け	中

故障原因の切り分けをモデルのみを用いて行うことは（論文参照する）、非現実的であり、実機の状態をフィードバックしながら対話的に切り分けを行っていくシステムのほうが現実的である。

詳細なモデルを用いて、不具合原因である機器の故障の原因などを探索可能にしたものなどがあるが、網羅的な故障仮説の洗い出しを行うことがメインであり、実際の検証段階を組み込んだものは少ない。

また、故障診断のコンテキストによってどこまで掘り下げるべきか使い分けるべきである<sup>8)</sup> ことを書く。今回の手法で扱える範囲はどこまで、試験時に役立てることができるのか、運用時だとどうなのかを示す。

故障診断の全ての過程を対象の制約モデルを用いて行うことは、対象とするモデルの粒度に非常に高い忠実度が求められるため、複雑に物理現象が絡み合う衛星では難しい。むしろ、人間を対話的にサポートすることによってシステムに求められるモデル化のコストを下げつつ、不具合分析（故障診断）の過程を体系化し、経験の少ないエンジニアの支援ができると考えられる。

本手法では、衛星へのアクションとしてコマンドを用いながら、故障仮説の洗い出しと検証を同時に行うことによって、用意すべきモデルに求められる粒度のレベルを下げている。

自分の手法の新規性を示す。

## 5 現在の状況

以下では、現在検討しているモデル化の手法に関して述べたあと、ケーススタディを用いて本手法による具体的な不具合分析の流れを説明する。

## 6 提案手法

以下の Figure 4 に提案手法の不具合分析を行うフローを示す。図中において本研究の対象は、黄色に示しているところであり、不具合発生時のコマンド送信情報とテレメトリ情報が与えられてから、それらを通る経路の探索を行い、その経路内に存在する故障箇所を検証するために必要な確認事項の洗い出しを行うことである。

### 6.1 不具合分析アルゴリズム

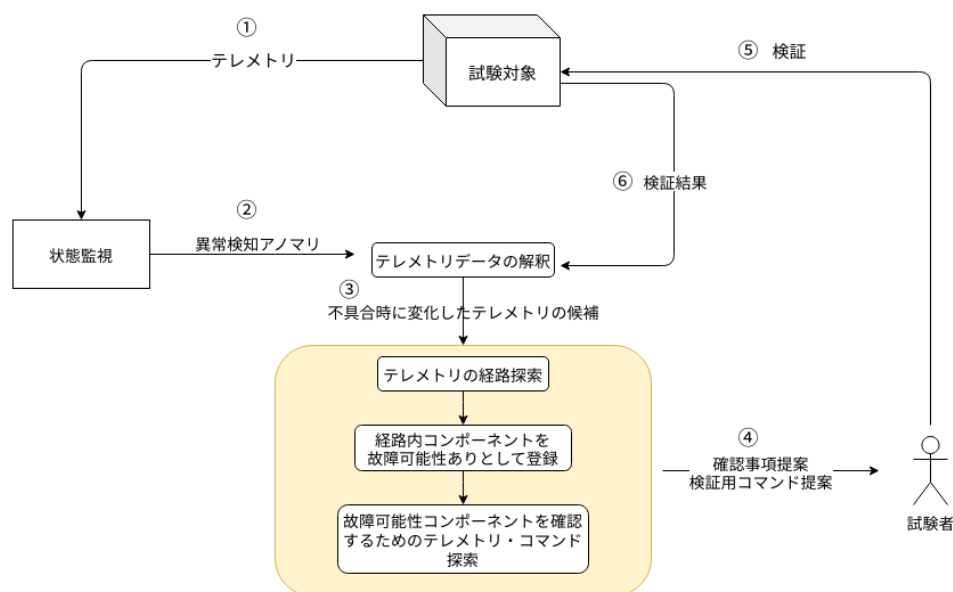


Figure 4: 不具合分析の流れ

本手法の不具合分析の流れは以下である。

1. 不具合検知のきっかけとなったテレメトリの候補を与える。
2. そのテレメトリに影響を与えるコマンドを送信してから、地上局がテレメトリを受信するまでの一連の経路を取得する。
3. 得られた経路内にあるコンポーネントを「故障候補」として登録する（故障仮説の生成）
4. 他のテレメトリを確認することによって、棄却できる「故障候補」を棄却する。
5. コマンドを送って得られるテレメトリ情報によって、故障候補を棄却し、故障箇所の切り分けができるコマンドの探索を行う。

(a)

6. 以上で得られたコマンドに関して，人が切り分けを行う際の判断の指標と共に提示する．
7. システムが提示した指標を元に人が打つコマンドを選択し，仮説の検証を行う．
8. コマンドによって返ってきた，テレメトリを確認し正常かどうかのフィードバックを行う．

ここで，以上の手法で利用する情報の粒度としてはコマンドとテレメトリのみである．よって，軌道上での運用中でも同じような情報を用いて異常テレメトリから検証を行う際に支援が可能である．

## 6.2 事前定義モデル

以上で述べた，不具合分析を行うために必要になるモデルに関して，具体的なテストケースをベースにして説明を行う．

### 6.2.1 対象とするテストケース

今回，以下の Figure 5 のような簡易衛星モデルを考え，モデル化の手法について述べる．

また，矢印の色が情報の方向性を表しており赤がコマンドによる情報の伝達過程，青がテレメトリによる情報の伝達過程である．また，矢印の種類が情報として伝わる物を表しており，それぞれ以下のようにになっている．

- Signal：電気信号
- Radio wave：電波
- Power：電源
- Heat：熱

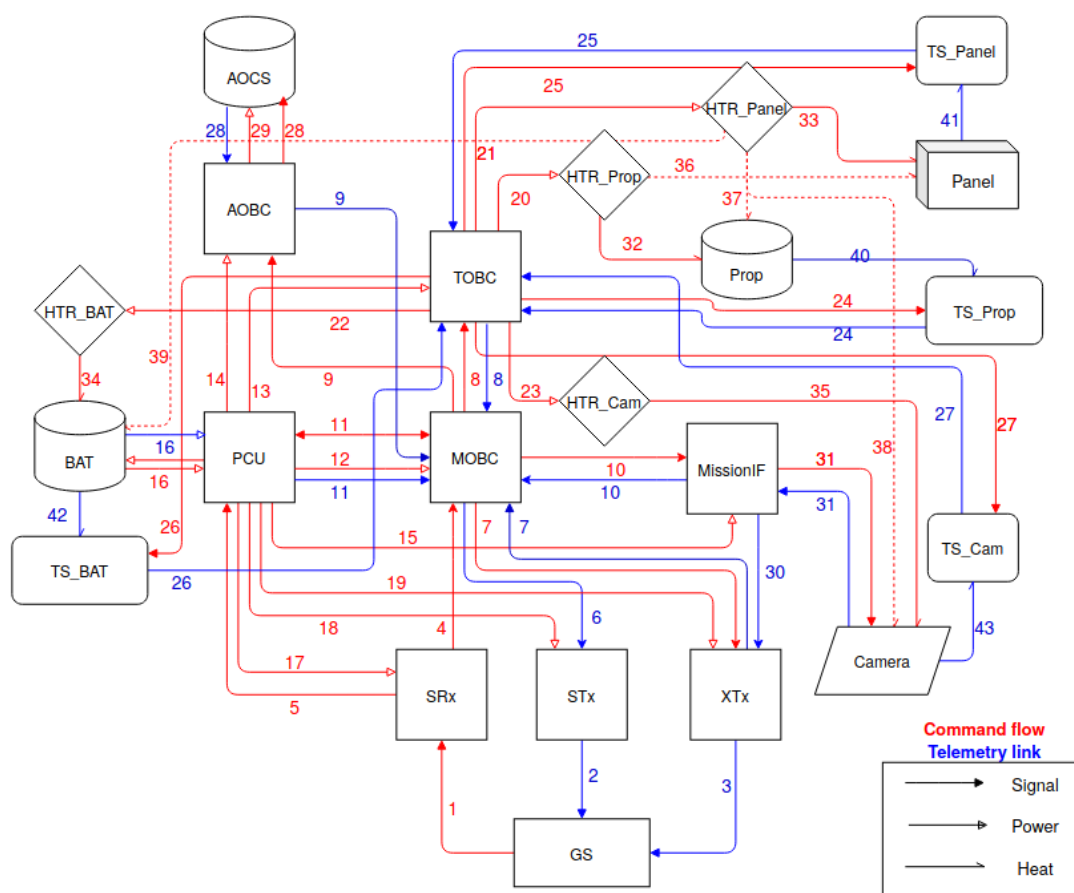


Figure 5: 簡易衛星モデル

### 6.2.2 各コンポーネント間の接続関係モデル

山口ら<sup>5)</sup>は人工衛星デバイスオンロジーとして、衛星内部でのコンポーネント間のつながりを表現するために「ポート」と「導管」という概念を定義している。この手法を参考にし、以下の Table 2 のようにリンクの定義を行い、Table 3 のように各コンポーネントの属性としてリンクを定義した。

リンクが持つ情報としては、リンク名、接続コンポーネント、ID、伝達物となっており、ID が各リンク固有の識別子としてリンクを参照する際に使用される。また、実際にコンポーネント間を接続している実態（配線やコネクタなど）を表現しているのではなく、接続関係を概念的に表現したものにすぎない。

Table 2: リンク定義

ポート名	接続コンボ	ID	伝達物
GS-MOBC	[GS, MOBC]	1	Radio waves
GS-PCU	[GS, PCU]	2	Radio waves
MOBC-TOBC	[MOBC, TOBC]	3	Signal
MOBC-PCU	[MOBC, PCU]	4.1	Signal
		4.2	Power
PCU-TOBC	[PCU, TOBC]	5	Power
TOBC-Heater_p	[TOBC, Heater_p]	6	Power
TOBC-Heater_sys	[TOBC, Heater_sys]	7	Power
TOBC-TempSensor_p	[TOBC, TempSensor_p]	8	Signal
TOBC-TempSensor_sys	[TOBC, TempSensor_sys]	9	Signal
Heater_p-Prop	[Heater_p, Prop]	10	Heat
Heater_p-Sys	[Heater_p, Sys]	11	Heat
Heater_sys-Prop	[Heater_sys, Prop]	12	Heat
Heater_sys-Sys	[Heater_sys, Sys]	13	Heat
TempSensor_p-Prop	[TempSensor_p, Prop]	14	Heat
TempSensor_sys-Sys	[TempSensor_sys, Sys]	15	Heat

次に，コンポーネントの定義を行う．以下の Table 3 では，衛星システム全体として使用されているコンポーネントのリストを作成し，各コンポーネントの属性としてコマンドリンクとテレメトリリンクを，上で定義したリンクの ID を用いて定義している．ここで，コマンドリンクというのはコマンドによる情報の伝達で使用されるリンクであり，テレメトリリンクというのはテレメトリによる情報の伝達で使用されるリンクである．この時，属性として持つリンクはそのコンポーネントが出力元となる場合としている．

Table 3: コンポーネント定義

コンポーネント	コマンドポートID	テレメトリポートID
GS	1, 2	
MOBC	3, 4.1	1
PCU	4.1, 4.2, 5	4.1
TOBC	6, 7	3
Heater_p	10, 11	
Heater_sys	12, 13	
Temp_p		8
Temp_sys		9
Prop		14
Sys		15



以上の情報によって、衛星内部でコンポーネント全体がどのように接続しているかを定義することが可能になる。

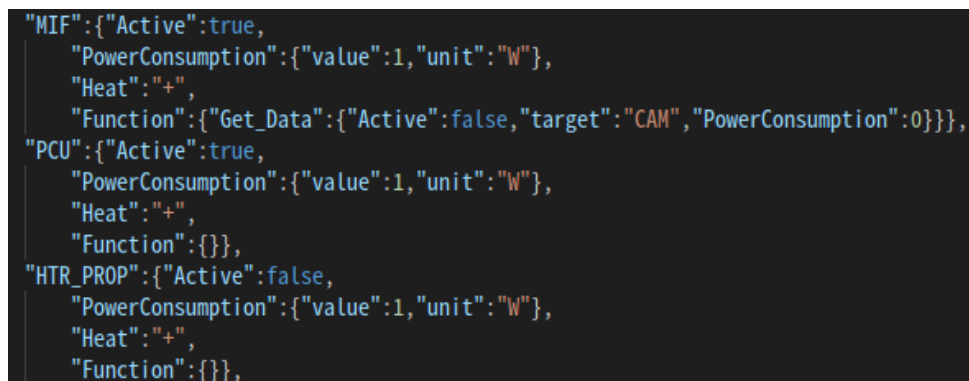
また、テレメトリに含める状態量として、以下の ような各コンポーネントの状態を定義する。

Code 1: コンポーネント初期状態例

```

1  {...
2  "AOBC":{"Active":true,
3          "PowerConsumption":{"value":2,"unit":"W"},
4          "Heat":"+ ",
5          "Function":{}}},
6  "MIF":{"Active":true,
7          "PowerConsumption":{"value":1,"unit":"W"},
8          "Heat":"+ ",
9          "Function":{"Get_Data":{"Active":false,"target":"CAM","PowerConsumption":0}}},
10 "PCU":{"Active":true,
11         "PowerConsumption":{"value":1,"unit":"W"},
12         "Heat":"+ ",
13         "Function":{}}},
14 ...}

```



```

"MIF":{"Active":true,
  "PowerConsumption":{"value":1,"unit":"W"},
  "Heat":"+ ",
  "Function":{"Get_Data":{"Active":false,"target":"CAM","PowerConsumption":0}}},
"PCU":{"Active":true,
  "PowerConsumption":{"value":1,"unit":"W"},
  "Heat":"+ ",
  "Function":{}}},
"HTR_PROP":{"Active":false,
  "PowerConsumption":{"value":1,"unit":"W"},
  "Heat":"+ ",
  "Function":{}}},

```

Figure 6: コンポーネント初期状態例

本研究では簡単のため状態量として扱うものは、各コンポーネントの電源状態、それに伴う電力消費、姿勢変化及び、熱の発生としている。また、電源 ON/OFF 状態以外にも機能を持つコンポーネントは Function という概念を追加し、

### 6.2.3 コマンド・テレメトリの情報がコンポーネント間を伝わる経路のモデル

次に、コマンド及びテレメトリがコンポーネント間を伝わる経路を探索するために必要なコマンド及びテレメトリの定義に関して述べる。まず、今回の衛星モデルにおいて使用できるテレメトリ及びコマンドを以下の Table 4, 5 に示す。

Table 4: 使用テレメトリ

ID	テレメトリ	経路
1	MOBC カウンター	MOBC→GS
2	MOBC 電流値	PCU→MOBC→GS
3	TOBC カウンター	TOBC→MOBC→GS
4	TOBC 電流値	PCU→MOBC→GS
5	推進系温度	Prop→TempSensor_p→TOBC→MOBC→GS
6	システム温度	Sys→TempSensor_sys→TOBC→MOBC→GS
7	推進系ヒータ電流値	TOBC→MOBC→GS
8	システムヒータ電流値	TOBC→MOBC→GS
9	MOBC コマンドカウンター	MOBC→GS
10	TOBC コマンドカウンター	TOBC→MOBC→GS

Table 5: 使用コマンド

ID	コマンド	経路	影響テレメトリ ID
1	MOBC 電源 ON	GS→PCU→MOBC	1,2,9
2	MOBC コマンドカウンターアップ	GS→MOBC	9
3	TOBC コマンドカウンターアップ	GS→MOBC→TOBC	10
4	推進系ヒータ ON	GS→MOBC→PCU→TOBC →Heater_p→(Prop,Sys)	5,6,7
5	システムヒータ ON	GS→MOBC→PCU→TOBC →Heater_sys→(Prop,Sys)	5,6,8
6	TOBC 電源 ON	GS→PCU→TOBC	3,4,10

今回、Table 4 に示すテレメトリの経路及び、Table 5 に示す経路と影響テレメトリ ID に関しては事前に定義したものを使用した。

また、コマンドに対するテレメトリの応答を確認して不具合原因を絞り込んでいくためにはコマンドがテレメトリ情報に変化をもたらす必要がある。

実際の衛星ではコンポーネント数やコマンド・テレメトリの数が非常に多く、モデルが複雑化するため人によるモデル生成では作業量を考えると非現実的である。そのため、将来的にはこれらを必要最低限の情報から生成する手法に関しても検討していく。

### 6.3 コマンド評価指標

次に、上記のアルゴリズムによって切り分けを行う際、人間がコマンドを選択するための指標に関して説明する。本研究の当初の目的は地上試験においての不具合分析支援であったが、本手法は軌道上での運用時でも活用できると考えられる。そのため、指標として運用時に有用となるものも提案し、使用状況に合わせて評価指標を切り替えることのできるフレームワークであることを示す。

不具合分析を行う際、衛星の安全を確保しながら確実に切り分けを行えることが重要である。そのためには、不具合発生時の状態をなるべく変えないようにしながら徐々に切り分けを行っていくことが必要である。そこで、コマンドを打つことによって発生する衛星内部状態の変化の大きさを定量的に示し、そのコマンドの「負の効果」を表現することにする。

各コマンドを打つことによる状態量の変化として、以下のものが有用であると考えた。

- 変化する状態量の数
- コマンドを打つ前のバッテリー残量と、コマンドを打つことによって発生する消費電力
- 姿勢変化を起こすか否か

また、運用時には可視時間が限られていて、そのパス中に不具合原因を究明しなければいけないような時間制約がある場合がある．その際には、少ないコマンド数で効率的に原因特定が行えることが望ましい．そのため、以下のようにコマンドの「正の効果」を定義した．

コマンドの効果として重要なのは、コマンドを送ることによって確認できる故障候補がいくつ存在するかという点である．これは絞り込むことのできる能力を表していると言える．

各コマンドによって情報が伝達し、テレメトリとして地上局に返ってくる経路によって、確認する対象のリンクまでに通る経路が異なる．その各経路に存在するコンポーネントをつなぐリンクが正常である確率を  $P(l = \text{normal})$ 、異常である確率を  $P(l = \text{abnormal})$  として与える．あるリンク  $l_i$  を確認するためには、リンク  $l_i$  が接続されているコンポーネントまでの経路が正常であることが必要である．このことから、「リンク  $l_i$  を確認することができる確率」がそれぞれの経路によって定まる．このことを以下の Figure 7 に示す例を用いて示す．

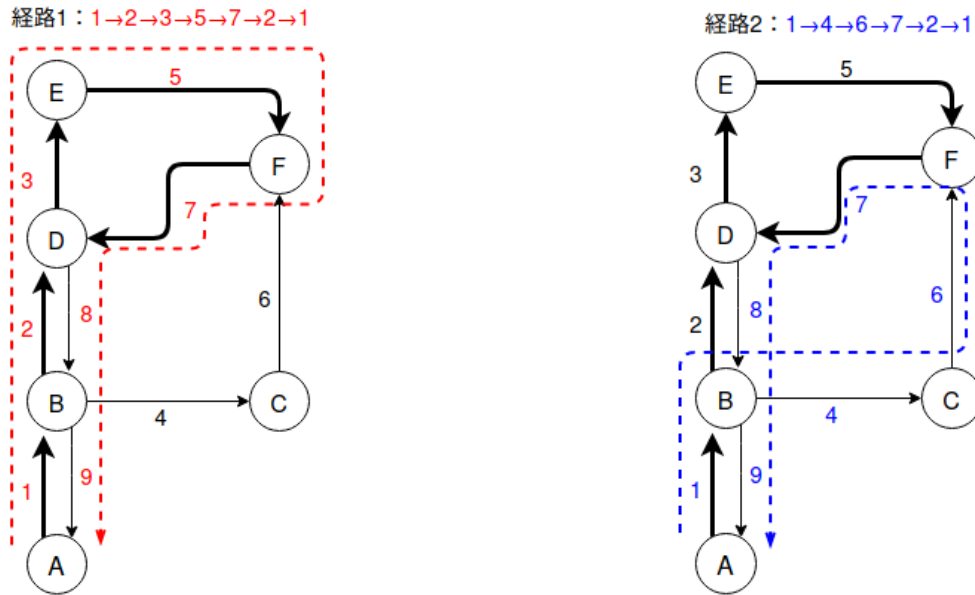


Figure 7: 故障候補とそれを確認するための情報伝達経路の例

以下では簡単のため、 $P(l_i = \text{normal}) = P(l_i = \text{abnormal}) = 0.5$  であるとし、太矢印になっている箇所が故障候補である．故障候補以外は正常であるとし、 $P(l_i = \text{normal}) = 1$  である．この時、リンク 7 を確認するためにはノード A からノード F までの経路 (1,2,3,5) と、ノード D からノード A までの経路 (8,9) が正常である必要がある．よって、経路 1 でリンク 7 を確認することができる確率は

$$P(l_7 | \text{経路 1}) = \prod_{i \in \text{経路 1}, i \neq 7} P(l_i = \text{normal}) \quad (2)$$

$$= \left(\frac{1}{2}\right)^4 \quad (3)$$

であることが分かる．同様に経路 2 に関してもリンク 7 を確認することができる確率は

$$P(l_7 | \text{経路 2}) = \prod_{i \in \text{経路 2}, i \neq 7} P(l_i = \text{normal}) \quad (4)$$

$$= \frac{1}{2} \quad (5)$$

となる．

これを元に考えると，コマンドが通る経路（R とする）によって確認できる故障可能性リンク数の期待値  $E(R)$  は以下のように求まる．

$$E(R) = \sum_{i \in R} P(l_i | R) = \sum_{i \in R} \prod_{j \in R, j \neq i} P(l_j = \text{normal}) \quad (6)$$

これが，あるコマンドとそれによって影響を受けるテレメトリとで形成される経路を伝達する情報によって確認することのできるリンクの数を示している．つまり，コマンドが不具合原因を絞り込むことのできる能力を表していると言える．Figure 7 の経路 1( $R_1$  とする) と経路 2( $R_2$  とする) に関して計算してみると

$$E(R_1) = \quad (7)$$

$$E(R_2) = \quad (8)$$

理想的には通る経路のすべてを確認できることを考えると，通る経路の数と期待値の差が大きなコマンドは不確実性が高いということになる．

上で述べたように，切り分けを行っていくためには確実性が重要である．そこで，確実性を示す指標として以下のものを提案する．

$$= \quad (9)$$

$$= \quad (10)$$

今回，故障候補の洗い出しは研究対象ではないため，簡単に不具合検知のきっかけとなったテレメトリが通る経路としており，それ以外の経路は正常であるという仮定をする．

まず，コマンドを打つことによってどれだけの情報が得られるかという点から，そのコマンドを打つことによる情報量を考えた．

その指標として，コマンドの能力を情報量（エントロピー）の定義から考えた．無理にエントロピーを使うのは間違っている気がする．情報量という考え方が妥当ではないか？確率分布になっていないことを考えると．

コマンド  $j$  による切り分けの効果の指標を以下に定義する．

コマンド  $j$  が確認できるリンクの集合を  $\mathbb{L}_j$  とする．この時，集合に含まれるリンクは  $l_i \in \mathbb{L}_j$  とする．また，「 $\mathbb{L}_j$  を確認する」という事象を  $\Omega$  とすると，各リンク  $l_i$  を確認するという事象  $E_i \in \Omega$  とする，

ここで，各リンクの状態を確認する方法として，本手法ではコマンドによるアクセスのみを考えているので「リンク  $l_i$  を確認する」という事象と「リンク  $l_i$  を確認できるコマンドを選択する」という事象の確率は等価である．よって「リンク  $l_i$  を確認する」という事象の確率は式 (11) で与えられる．この時，事象 ( $E_i$ ) が起こった時に受け取る（選択）情報量  $I_i$  は式 (12) で定義される．

$$P(E_i) = \frac{\text{Number of command which can verify Link } i}{\text{All command number}} \quad (11)$$

$$I(E_i) = -\log P_i \quad (12)$$

$$(13)$$

確率  $P(E_i)$  は確率分布であるという説明が必要  
 よって事象  $\Omega$  による平均情報量は

$$H(P) = - \sum_{E_i \in \Omega} P(E_i) \log P(E_i) \quad (14)$$

となる．定性的な意味としては，コマンドを打つことによってどれだけの情報が得られる可能性があるかを示している．

コマンドが絞り込むことのできる能力という点はどうする？

以上の指標を踏まえた上でどのように提示し人間の判断を支援するのかを考える．

地上試験時は，不具合原因の特定を時間を掛けて十分に行いたいので，衛星の安全を担保しながら確実な箇所から切り分けを行うことが重要である．不具合分析を行う過程で，内部状態が大きく変化してしまうと切り分けたい故障箇所がわからなくなってしまう可能性があるため，確実に切り分けを行っていくためには，コマンドによって変化する状態量が少ないことが望ましい．よって，地上試験では「波及効果の大きさ」が小さなコマンドを用いて切り分けを行っていくことが望ましい．

一方で，運用中は可視時間中に不具合原因の特定を行わなければならないなどの時間制約の厳しい条件下での分析が必要になることもある．そのような時には，リスクを大きく取りつつ効果の大きなコマンドを選択する必要がある．

また，試験では考慮しない電力による制約も，コマンドを選択する際の一つの指標であると言える．

このように，コマンドによって不具合特定を行っていく際には，評価指標を使い分けることができるようなシステムである必要がある．

このようなアルゴリズムで不具合分析を行うためには，以下のようなモデルが必要となることがわかった．

- 各コンポーネント間の接続関係
- コマンド・テレメトリがコンポーネント間を伝わる経路
- コマンドに影響を受けるテレメトリの関係

#### 6.4 各故障を切り分けていくための確認事項の探索

次に，故障候補の中から切り分けを行っていくためのアルゴリズムに関して検討状況を述べる．不具合が発生している状態で予期せぬ二次故障を起こさないために，探索順序としては，衛星の状態を変えずに確認できるものを優先的に探索することが望ましい．

そこで，不具合発生時に取得しているテレメトリデータの中から故障原因特定に役に立つテレメトリ情報が存在するなら，そのテレメトリを確認事項として提案する．

また，衛星の状態変化を行わずに故障原因特定のために得られる情報がなくなれば，次ステップとしてコマンドを打って得られる情報から切り分けを行っていくことになる．このとき，対象となるコマンドが通る経路の中で，地上局 (Ground Station, GS) の近くの経路から順に，検証を行うためのコマンドの探索を行う．

#### 6.5 問題設定

最後に，不具合分析の具体的な流れをみるために，以下のような故障を考え，不具合分析を行っていく．まず，MOBC と TOBC からテレメトリは全て降ろされているものと仮定する．また，故障は「推進系ヒータの接着不良」であるとする．その時，テレメトリを通して確認できるのは

- 「推進系ヒータ ON」コマンドを送ったのに「推進系温度」テレメトリが変化しない．

という事象である．よって不具合検知は，この事象によって行われる．

### 6.5.1 不具合分析の例

この不具合において、「推進系ヒータ ON」コマンドを送信して推進系ヒータに熱が伝わるまでの経路と、推進系温度計が温度を読み取り「推進系温度」テレメトリとして地上局に伝わるまでの経路の中に故障箇所があると考えることができる．この経路を以下の Figure 8 に太矢印で示している．

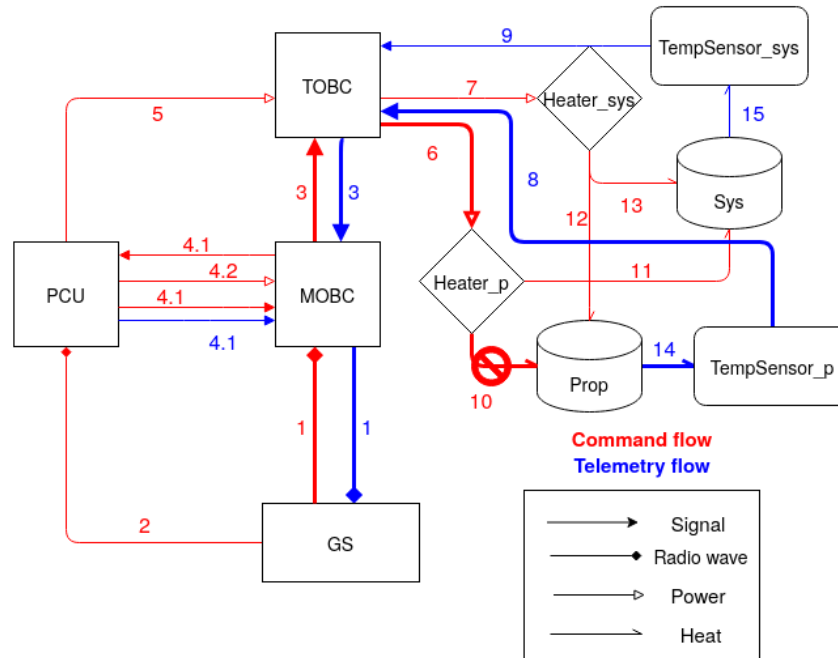


Figure 8: 故障箇所と不具合検知に関連するコマンドとテレメトリの経路

また、この経路内にあるコンポーネントに電源が入っているかどうかを確認するためには、そのコンポーネントに電源を供給するための経路が正常に作動しているかどうかを確認する必要がある．そこで

- PCU→MOBC(リンク ID:4.2) , PCU→TOBC(リンク ID:5)

も検証を行う対象として考える．以上より、検証すべき経路は下記ようになる．

- コンポーネント : GS , MOBC , TOBC , Heater\_p , Prop , TempSensor\_p
- コマンドリンク : GS-MOBC(1) , MOBC-TOBC(3) , MOBC-PCU(4.2) , PCU-TOBC(5) , TOBC-Heater\_p(6) , Heater\_p-Prop(10)
- テレメトリリンク:GS-MOBC(1) , MOBC-TOBC(3) , TOBC-TempSensor\_p(8) , TempSensor\_p-Prop(14)

以下、提案手法によるアルゴリズムによって不具合分析を行っていく．

まず、問題設定より MOBC 及び TOBC のテレメトリは全てダウンリンクされている状態にあるので、

- テレメトリリンク:TOBC→MOBC(3) , MOBC→GS(1)
- コマンドリンク : PCU→MOBC(4.2) , PCU→TOBC(5)

は問題ないことが確認できるため、故障可能性はなくなる。問題設定ではあるが、実際に「MOBC 及び TOBC のテレメトリが全てダウンリンクされている」ことを確認するためには、「MOBC カウンター」「TOBC カウンター」を確認することが必要である。

また不具合発生時、推進系ヒータが正常に作動しており、システム温度計からテレメトリを下ろす経路に問題がなければ「システム温度」が上昇しているはずである。よって、テレメトリの確認によって検証できる経路として、コマンドリンク「TOBC-Heater\_p(6)」がある。以上より、不具合発生時から状態変化させずに確認すべき事項として以下が挙げられる。

Table 6: コマンドなしでの確認事項

確認テレメトリ	確認対象ポート	正常時のテレメトリ変化
MOBCカウンター	MOBC-GS(1), PCU-MOBC(4.2)	TOBCカウンターが時間変化する
TOBCカウンター	TOBC-MOBC(3), PCU-TOBC(5)	TOBCカウンターが時間変化する
システム温度	TOBC-Heater_p(6)	推進系ヒータONコマンド後温度が上昇する

衛星の状態を変化させることなく、テレメトリを確認するだけで検証できる箇所はこれ以上存在しないので、次にコマンドによる検証を行う。

まず、コマンドパスとして GS に近い箇所から順に確認を行う必要がある。MOBC へのコマンドが通っているかを確認するためには、「MOBC コマンドカウンターアップ」コマンドを送って、「MOBC コマンドカウンター」が変化していることを確認できれば良い。同様に考えると、以下の様に確認事項を洗い出すことができる。

Table 7: コマンド送信による確認事項

確認用コマンド	確認対象ポート	正常時のテレメトリ変化
MOBCコマンドカウンターアップ	GS-MOBC(1)	MOBCコマンドカウンターが上昇する
TOBCコマンドカウンターアップ	MOBC-TOBC(3)	TOBCコマンドカウンターが上昇する
システムヒータON	TempSensor_p-TOBC(8), Prop-TempSensor(14)	推進系温度が上昇する。

以上の項目を確認した際、今回想定した故障モード（推進系ヒータの接着不良）では期待されるテレメトリデータの変化が起こるので、Table 7 の確認対象パスの中にある故障可能性箇所は棄却され、故障可能性リンクとして残るのは

- コマンドリンク：Heater\_p-Prop(10)

となる。

この経路上で考えうる故障モードと照らし合わせると、この切り分けによって残る故障モードは

- 推進系ヒータの故障
- 推進系ヒータの接着不良

となる。この時「システム温度」の上昇によって「推進系ヒータの故障」の可能性は棄却できるため、最終的に「推進系ヒータの接着不良」が残り、実際の故障を棄却すること無く、絞り込みができていえる。

## 7 今後の方針

今後は、上のような不具合分析の流れをモデルを元に行うことができるように、簡易的なものを作成し、検証していく。

また、生成された確認事項を実行するに当たって、衛星の状況によってはコマンドを打つことで衛星が危険に晒されることも考えられる。最終的には、コマンドが安全であるかどうかという判断基準も試験を行う人に対して明示できると良いと考えている。

また、熱や運動のような支配方程式が微分方程式で定義されており、単純なつながりだけでは表現できないような現象に関しては、この手法では簡易的に表現することしかできていない。そのため、この手法を物理現象が複雑に絡んだ不具合分析に適用することは難しい。将来的には、これらの支配方程式を解くシミュレータを組み込むことによって複雑な物理現象も扱えるように手法の改良が必要である。

## References

- 1) M Langer and J Boumeester. Reliability of CubeSats Statistical Data, Developers' Beliefs and the Way Forward. *Proceedings of 30th Annual AIAA/USU Conference on Small Satellites*, pp. 1–12, 2016.
- 2) Catherine C Venturini. Improving Mission Success of CubeSats. Technical report, 2017.
- 3) Seiko SHIRASAKA, Kanenori ISHIBASHI, and Shinichi NAKASUKA. F4 Study on Reasonably Reliable Systems Engineering for nano-Satellite. *The Proceedings of the Space Engineering Conference*, Vol. 2010.19, No. 0, pp. 1–4, jan 2011.
- 4) Hirobumi SAITO. Secondary Analysis on On-Orbit Failures of Satellite. *JOURNAL OF THE JAPAN SOCIETY FOR AERONAUTICAL AND SPACE SCIENCES*, Vol. 59, No. 690, pp. 190–196, 2011.
- 5) Kota Yamaguchi and Hori Koichi. Fault Network Analysis of Artificial Satellite Using Ontology. pp. 1–4, 2014.
- 6) Peter Struss and Oskar Dressier. "Physical Negation" - Integrating Fault Models into the General Diagnostic Engine. Vol. 89, pp. 1318–1323, 1989.
- 7) 來村徳信, 西原稔人, 植田正彦, 池田満, 小堀聡, 角所収, 溝口理一郎. 故障オントロジーの考察に基づく故障診断方式：網羅的故障仮説生成. PhD thesis, sep 1999.
- 8) Kitamura Yoshinobu and Riichiro Mizoguchi. An Ontology of Faults - Articulation and Organization -. pp. 1–10, 1998.