卒業研究報告

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 figuring out possible failures and identifying fault location requires considerable human resources and time. This research proposes a new approach to support identifying failure causes of satellite by showing the verify methodology and the performance index which is searched based on the model of signal, electrical and physical interaction between subsystem or components and models of signal transmission routes in a satellite. The effectiveness of the approach is proved by applying it to a simple satellite model and comparing with results or process of identifying failure cause without 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異常事象から故障候補の洗い出し及び，故障箇所の特定を行うことは困難である．

本研究では，コンポーネント間の接続関係モデル，情報伝達の経路モデルを用いて衛星の故障候補の検証方法（確認事項，打つべきコマンド）を人間の判断を支援する指標と共に提示することで，不具合分析を支援する手法を提案する．また，簡易的な衛星モデルを用いて不具合分析を実践し，手法を用いない場合との比較によっ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用いた衛星の故障箇所の特定を支援する手法を検討している．
以下の章では，まず研究背景として地上試験におけるリスク分析の不十分さ及び，不具合原因仮説の検証を支援する研究が十分に行われていないことを述べ，次にそれを踏まえた研究目的に関して述べる．また，提案手法の章では，不具合分析のアルゴリズム，使用するモデルに関して述べ，そのモデルを用いた仮説検証のためのコマンド及び確認事項の探索方法，人がコマンド選択をする際に必要な評価指標に関して説明する．最後に，簡易衛星モデルを用いた実践例を示し，今後の方針について述べる．

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 [1)]．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 [1)]．

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も多いため，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 [2)]．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 [2)]．

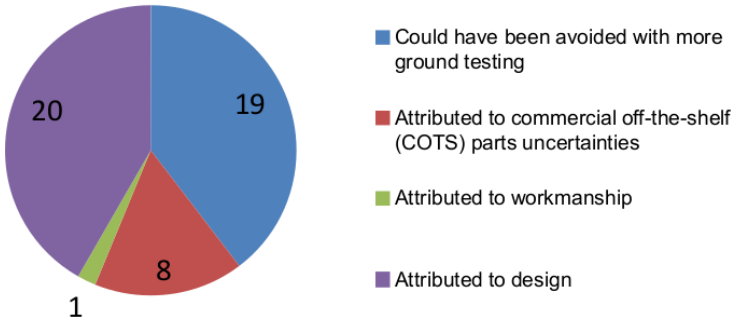

Figure 1: 故障原因に関するインタビュー結果 [2)]

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 [3)]．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 [3)]．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．

$$R_{sat} = R_{des} \times R_{fab} \times R_{comp} \times R_{op} \tag{1}$$

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 [4)](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 [4)]．

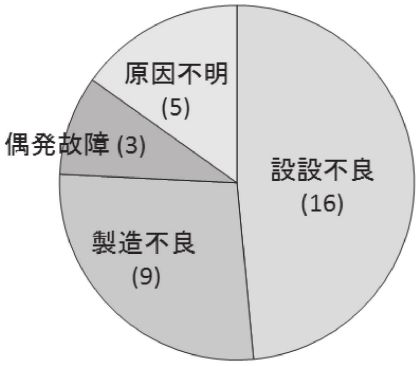

Figure 2: 軌道上故障の原因類型の分布 [4)]

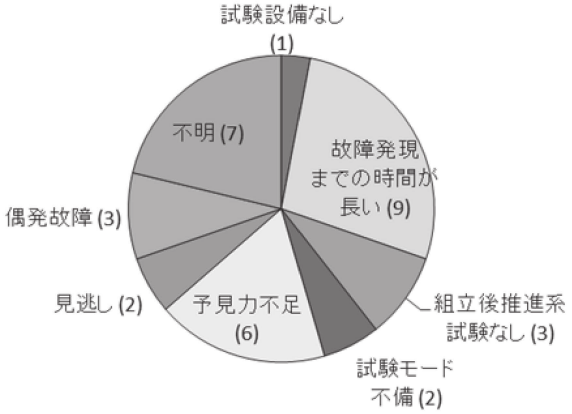

Figure 3: 軌道上故障の要因を地上で発見できなかった原因類型の分布 [4)]

2.3　不具合原因特定の難しさ

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．
その原因を具体的に示すため，以下に人間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し記録に残す．

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り，それにに対するテレメトリの挙動によって判断するという作業を繰り返す．

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．

上の流れを元に分析が不十分になっている原因を考える．まず，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，十分に不具合原因の候補を洗い出すことはできない．
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 [5)] 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もう一つの原因として，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．また，実ミッションで使用するコマンドとテレメトリは膨大な数であるため，その中から切り分けを行うための情報を選択し，仮説の検証を行う作業は無駄やヒューマンエラーを生むきっかけとなる．検証作業の際，未熟な運用者が不具合原因特定のために誤ったコマンドを送信してしまうと，衛星の生存を脅かす可能性がある．このため，不具合原因特定を行う際にはそのコマンドが ”安全 ”なのかという点も非常に重要となる．

2.4　不具合分析関連研究

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，コマンドとテレメトリを用いて原因特定を行う過程が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，古くから不具合分析システムの研究が盛んに行われている．以下の Table 1 に，モデルベースで機械などを対象にした不具合分析，故障診断を行う手法に関してまとめた．

Table 1: 不具合分析手法の比較

手法	故障網羅性	手法の目的
GDE	低	故障仮説生成
GDE+[6)]	中	故障仮説生成
網状故障解析 [5)]	中	異常モード洗い出し
故障オントロジー [7)]	高	故障仮説生成
本手法	中	故障箇所特定

故障仮説生成の研究に関しては，機器の正常時のモデルだけでなく，故障時モデルを組み込んだもの [6)] や，オントロジーを用いてプロセスのつながりまでモデル化したもの [5)]，異常伝播事象までモデル化して階層的な推論を行うもの [7)] などがあり，網羅的に故障候補を洗い出すために広く取り組まれている．一方で，來村ら [7)] が効率の良い検証方法に関しては今後の課題として言及しているように，故障仮説の検証に取り組んだものは少ない．

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，その中から原因を特定していく過程に，高い知識と経験が必要であることが，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推察される．また，故障候補の網羅的な洗い出しに関しては広く取り組まれている一方で，仮説の検証作業の支援に関して取り組んだ研究は行われていない．

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，不具合事象から故障箇所の特定を行えるような不具合分析支援手法の提案を目的とする．また，以下では不具合発生から故障箇所の特定を行う過程を「不具合分析」と表現している．

具体的には，本手法はコマンドとテレメトリをベースにして行う不具合分析を対象にしており，不具合発生時に故障箇所を特定するために，確認すべきテレメトリ，打つべきコマンドを選択肢として提示することで，人間の判断の支援を行う．

また，不具合原因特定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，不具合分析の過程を体系化することで，経験の少ないエンジニアの支援ができると考えられる．そのため，システムが提示した選択肢を用いて人間が実機での検証を行い，その結果をシステムにフィードバックすることで，対話的に故障箇所の特定を行っていく構成になっている．

以上の機能を満たすために，本手法は下記の 3 つの要素で構成されている．

- 衛星内部のコンポーネント間接続関係モデル及び，情報伝達経路モデル
- 故障箇所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
- 人間の判断を支援するコマンドの評価指標の提案

4　提案手法

以下の Figure 4 に，本手法を用いて不具合分析を行う流れを示す．本研究の対象は，Figure 4 で色付けしているところであり，不具合発生時の異常テレメトリ情報が与えられてから，故障仮説の生成，その仮説を検証するための「コマンド及び確認事項」を探索し，人間に提案を行うところまでである．

4.1　不具合分析アルゴリズム

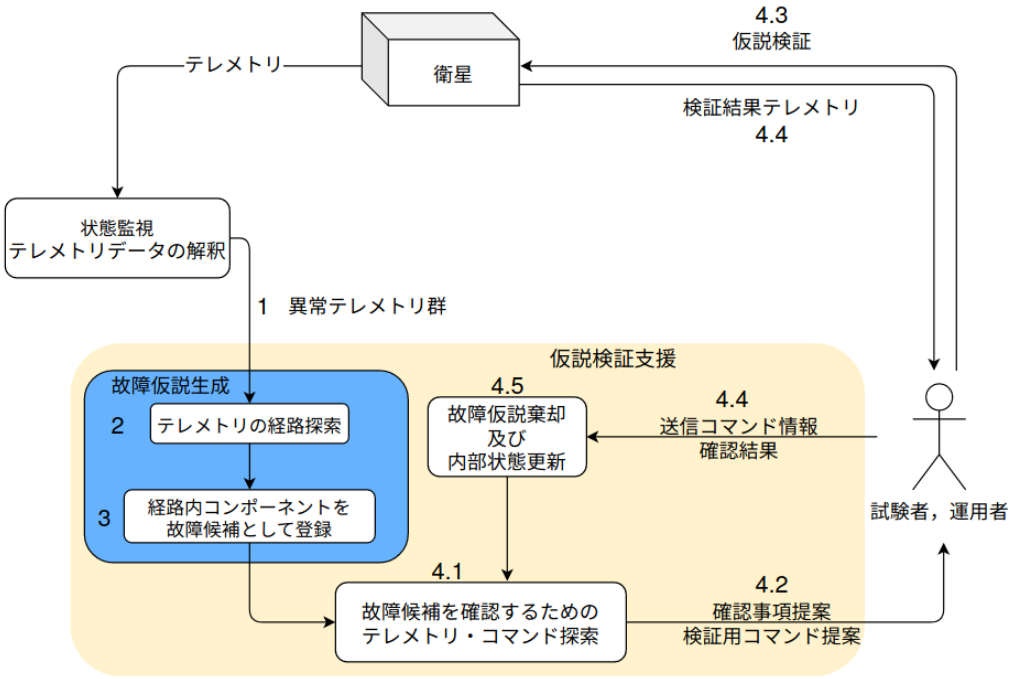

Figure 4: 不具合分析の流れ

本手法を用いた不具合分析の流れは以下である．

1 異常検知のきっかけとなったテレメトリ群を与える．

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 打つコマンドが無くなるか，不具合原因の特定ができるまで以下を繰り返す．

- 4.1 故障候補を確認するためのテレメトリ及びコマンド探索
- 4.2 上で得られたコマンド及び確認事項を，人間の判断を支援する指標と共に提示する．
- 4.3 システムが提示した情報を元に人が打つコマンドを選択し，仮説の検証を行う．
- 4.4 送信コマンドに対する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．
- 4.5 人間からのフィードバックに応じて故障仮説の棄却及び，モデルが持つ状態の更新を行う．

故障候補を確認するためのテレメトリ・コマンド探索の流れの詳細に関しては後ほど言及する．

4.2　事前定義モデル

次に，以上で述べたアルゴリズムで不具合分析を行うために必要なモデルに関して，具体的なテストケースをベースにして説明する．

4.2.1　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対象にしてモデルの定義及び不具合分析手法の実践を行う．
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，青がテレメトリによる情報の伝達である．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なっている．

- Signal：電気信号
- Power：電源
- Heat：熱

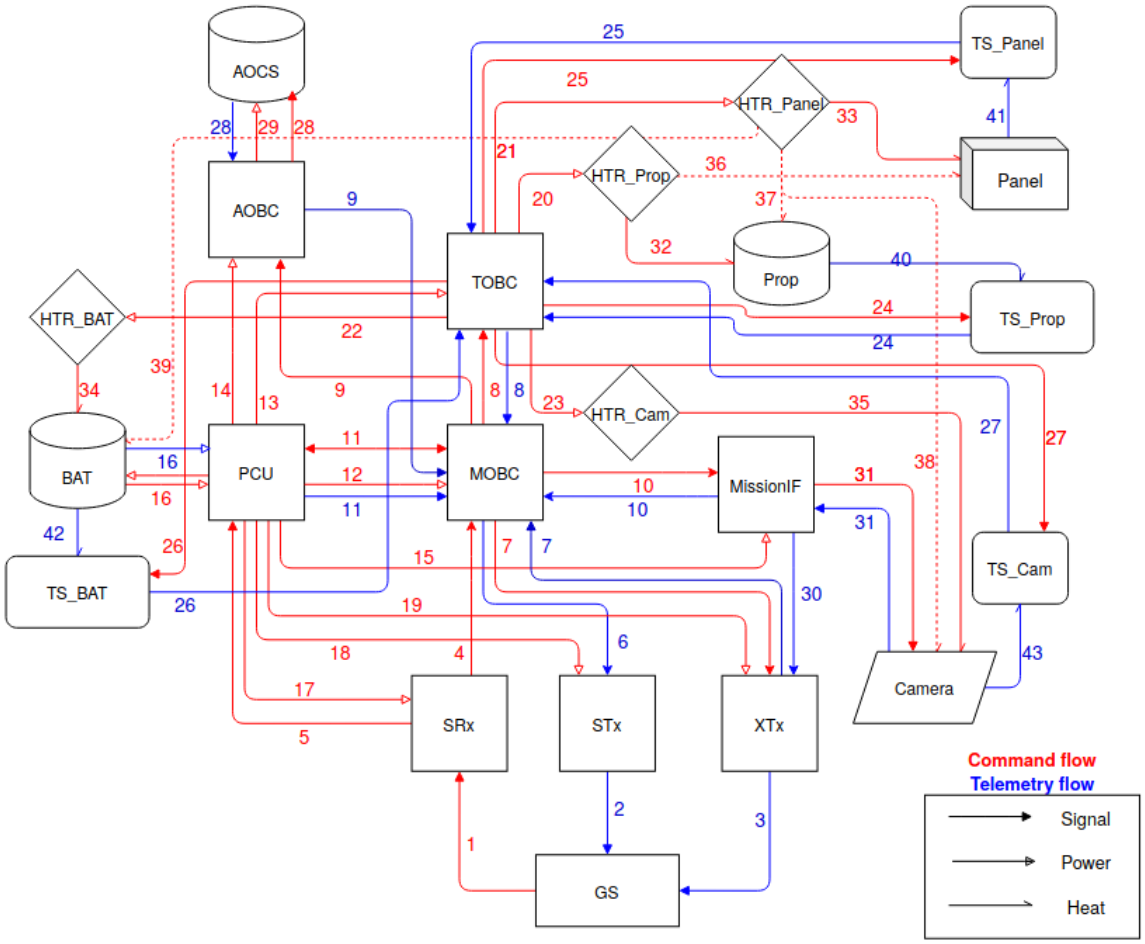

Figure 5: 簡易衛星モデル

4.2.2　各コンポーネント間の接続関係モデル

来村ら [8] は拡張デバイスオントロジーとして，機器を構成する装置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．このオントロジーを用いて，山口ら [5] は人工衛星デバイスオントロジーを構築している．これらを参考にし，以下の Table 2 のように接続関係を「リンク」として定義した．
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，そのリンクが正常に情報伝達を行う確率となっており，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: リンク定義例

ID	Link_name	Compo1	Compo2	Medium	Probability
17	PCU-SRx	PCU	SRx	Power	0.5
18	PCU-STx	PCU	STx	Power	0.5
19	PCU-XTx	PCU	XTx	Power	0.5
20	TOBC-HTR_PROP	TOBC	HTR_PROP	Power	0.5
21	TOBC-HTR_PANEL	TOBC	HTR_PANEL	Power	0.5
22	TOBC-HTR_BAT	TOBC	HTR_BAT	Power	0.5
23	TOBC-HTR_CAM	TOBC	HTR_CAM	Power	0.5
24	TOBC-TS_PROP	TOBC	TS_PROP	Signal	0.5
25	TOBC-TS_PANEL	TOBC	TS_PANEL	Signal	0.5
26	TOBC-TS_BAT	TOBC	TS_BAT	Signal	0.5
27	TOBC-TS_CAM	TOBC	TS_CAM	Signal	0.5
28	AOBC-AOCS	AOBC	AOCS	Signal	0.5
29	AOBC-AOCS	AOBC	AOCS	Power	0.5
30	MIF-XTx	MIF	XTx	Signal	0.5
31	MIF-CAM	MIF	CAM	Signal	0.5
32	HTR_PROP-PROP	HTR_PROP	PROP	Heat	0.5
33	HTR_PANEL-PANEL	HTR_PANEL	PANEL	Heat	0.5
34	HTR_BAT-BAT	HTR_BAT	BAT	Heat	0.5
35	HTR_CAM-CAM	HTR_CAM	CAM	Heat	0.5
36	HTR_PROP-PANEL	HTR_PROP	PANEL	Heat	0.5
37	HTR_PANEL-PROP	HTR_PANEL	PROP	Heat	0.5
38	HTR_PANEL-CAM	HTR_PANEL	CAM	Heat	0.5
39	HTR_PANEL-BAT	HTR_PANEL	BAT	Heat	0.5

次に，コンポーネントの定義を行う．以下の Table 3 では，衛星システム全体で使用されているコンポーネントのリストを作成し，各コンポーネントが接続している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，コンポーネントが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例

Component	Com_linkID	Tel_linkID
GS	1	
MOBC	7,8,9,10,11	6
PCU	11,12,13,14,15,16,17,18,19	11
TOBC	20,21,22,23,24,25,26,27	8
AOBC	28,29	
MIF	31	30
XTx		3,7
STx		2
SRx	4,5	
HTR_PROP	32,36	
HTR_PANEL	33,37,38,39	
HTR_BAT	34	
HTR_CAM	35	
TS_PROP		24
TS_PANEL		25
TS_BAT		26
TS_CAM		27
PROP		40
PANEL		41
BAT		42
CAM		31,43
AOCS		28

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，各コンポーネントの状態を以下の Figure 6 のように定義する．本研究では，簡単のため扱う状態は，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その機能を Function で定義しており，機能の動作状態がコマンドによって操作される構成となっている．初期状態を Figure 6 のようなファイル形式で与え，その後の状態の更新は人間が選択したコマンドが持つ機能情報に基づいて行うようにしている．

```
"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

4.2.3　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

今回の衛星モデルにおけるテレメトリ及びコマンドを以下の Table 4，5 に定義した．
まず，本手法で用いるテレメトリの情報は，ID，テレメトリの名前，テレメトリが変化するためのトリガー，テレメトリの情報が衛星内部及び地上局まで伝わる経路である．今回は簡単のため，状態が変化するためのトリガー (TransitionTrigger) として，時間とコマンドのみを考えており，姿勢変化や軌道条件に依存した状態変化は考えないことにする．また，経路は通るリンクの ID を用いて表現している．

Table 4: 使用テレメトリ

ID	TelemetryName	TransitionTrigger	path				
1	MOBC_Counter	Time	6	2			
2	TOBC_Counter	Time	8	6	2		
3	AOBC_Counter	Time	9	6	2		
4	MIF_Counter	Time	10	6	2		
5	MOBC_COM_Counter	Command	6	2			
6	TOBC_COM_Counter	Command	8	6	2		
7	AOBC_COM_Counter	Command	9	6	2		
8	MIF_COM_Counter	Command	10	6	2		
9	MOBC_Current	Command	11	6	2		
10	TOBC_Current	Command	11	6	2		
11	AOBC_Current	Command	11	6	2		
12	MIF_Current	Command	11	6	2		
13	SRx_Current	Command	11	6	2		
14	STx_Current	Command	11	6	2		
15	XTx_Current	Command	11	6	2		
16	PANEL_Temp	Command	41	25	8	6	2
17	PROP_Temp	Command	40	24	8	6	2
18	CAM_Temp	Command	43	27	8	6	2
19	BAT_Temp	Command	42	26	8	6	2
20	HTR_PANEL_Current	Command	8	6	2		
21	HTR_PROP_Current	Command	8	6	2		
22	HTR_CAM_Current	Command	8	6	2		
23	HTR_BAT_Current	Command	8	6	2		
24	BAT_Power	Command	16	11	6	2	
25	AOCS_Current	Command	28	9	6	2	
26	RW_RotateSpeed	Command	28	9	6	2	
27	M_DATA	Command	31	30	3		

また，コマンドの情報として ID，コマンドの名前，コマンドによって影響を受けるテレメトリの ID，コマンドの種別，コマンドによって情報が伝達する経路を与えている．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，コマンドが持つ機能によって，いくつかの種別に分類することができる．JAXA[9] は，衛星と衛星搭載機器の機能をモデル化し，機能情報の再利用性を高めることを目的とした手法を提案している．今回，その手法の中の一部を採用しコマンドの種別を 2 種類 (ACTION, GET) 定義した．また，各コマンドが持つ機能に関する情報を以下の Figure 7 のように定義している．これによって，各コマンドが上記で定義したコンポーネントが持つ機能を操作するという関係性を表現可能になる．

Table 5: 使用コマンド

ID	CommandName	impact_Tel_ID	type	path					
1	MOBC_ON	5,9	ACTION	1	5	12			
2	TOBC_ON	6,10	ACTION	1	5	13			
3	AOBC_ON	7,11	ACTION	1	5	14			
4	MIF_ON	8,12	ACTION	1	5	15			
5	MOBC_OFF	5,9	ACTION	1	5	12			
6	TOBC_OFF	6,10	ACTION	1	5	13			
7	AOBC_OFF	7,11	ACTION	1	5	14			
8	MIF_OFF	8,12	ACTION	1	5	15			
9	MOBC_NOOP	5	ACTION	1	4				
10	TOBC_NOOP	6	ACTION	1	4	8			
11	AOBC_NOOP	7	ACTION	1	4	9			
12	MIF_NOOP	8	ACTION	1	4	10			
13	HTR_PANEL_ON	5,6,10,16,20	ACTION	1	4	11	13	21	33,37,38,39
14	HTR_PROP_ON	5,6,10,17,21	ACTION	1	4	11	13	20	36
15	HTR_CAM_ON	5,6,10,18,22	ACTION	1	4	11	13	23	35
16	HTR_BAT_ON	5,6,10,19,23	ACTION	1	4	11	13	22	34
17	HTR_PANEL_OFF	5,6,10,16,20	ACTION	1	4	11	13	21	33,37,38,40
18	HTR_PROP_OFF	5,6,10,17,21	ACTION	1	4	11	13	20	33
19	HTR_CAM_OFF	5,6,10,18,22	ACTION	1	4	11	13	23	32
20	HTR_BAT_OFF	5,6,10,19,23	ACTION	1	4	11	13	22	31
21	RW_ON	5,7,11,25	ACTION	1	4	11	14	29	
22	RW_OFF	5,7,11,25	ACTION	1	4	11	14	29	
23	RW_START	5,7,11,26	ACTION	1	4	9	28	29	
24	RW_STOP	5,7,11,26	ACTION	1	4	9	28	29	
25	M_DATA_DOWN	5,8,27	GET	1	4	10	31		
26	GET_PANEL_TEMP	5,6,16	GET	1	4	8	25		
27	GET_PROP_TEMP	5,6,17	GET	1	4	8	24		
28	GET_CAM_TEMP	5,6,18	GET	1	4	8	27		
29	GET_BAT_TEMP	5,6,19	GET	1	4	8	26		
30	TAKE_PICTURE	5,8,27	ACTION	1	10	31			


```
"RW_START":{"type":"ACTION",
        "Active":true,
        "target":[{"Component":"AOCS",
            "Function":["RW_SPIN"]}]},
"RW_STOP":{"type":"ACTION",
        "Active":false,
        "target":[{"Component":"AOCS",
            "Function":["RW_SPIN"]}]},
"M_DATA_DOWN":{"type":"GET",
        "Active":true,
        "target":[{"Component":"CAM",
            "Function":["Get_Data"],
            "value":["Mission_Data"]}]},
```


Figure 7: コマンドの機能モデル

4.3　コマンド評価指標

次に，上記のアルゴリズムによって故障候補の切り分けを行う際，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．不具合分析を行う際，衛星の安全を確保しながら正確な故障箇所の特定を行うことが，地上での不具合改修に必要である．そのため，コマンドが衛星にとって安全であることが重要である．
また，本研究の当初の目的は地上試験における不具合分析支援であったが，提案手法はコマンドとテレメトリの粒度で得られる情報を用いて不具合分析を行っているため，軌道上での運用時にも活用できると考えられる．運用時には地上試験時とは異なり，不具合改修のための時間制約が発生することがあるため，地上試験時とは異なる指標が必要となる．そのため以下では，地上試験時と運用時の両方に関してコマンドを選択する上で必要な指標としてコマンドによる衛星生存性への副作用を示す指標とコマンドの故障候補の切り分け能力を示す指標を提案し，システムの使用状況に合わせてそれらの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．

4.3.1　コマンドによる衛星生存性への副作用

打つコマンドが安全であるかという点は，衛星の状態に依存するが，不具合発生時には衛星の状態把握が十分に行えていない状況であるため，網羅的にリスクを考慮した安全性を評価するのは困難である．そこで，以下では簡単に電力と姿勢の制約を元に，コマンドの危険性を定量化するための指標を示す．
まず，運用時には発電量と各コンポーネントの電力消費状態に応じて電力の制約が発生する．バッテリ残量が残り少ない状態で大きな電力を消費するコンポーネントの電源を ON にするといったことは，衛星の生存を脅かす危険な動作であると言える．コマンドを選択する際に電力に関する制約を明示的に示すことは，未熟な運用者が誤ったコマンドを打つことを防ぐために効果的であると考えられる．そのため，コマンドの副作用を示す一つの指標として「バッテリ残量」と「コマンドを打つことによって発生する消費電力」を示すことにする．ここでは簡単のため，バッテリ残量は電源が ON になっている機器の消費電力のみから計算することとし，姿勢の変化や日照条件に応じた充電量の変化は考慮していない．
次に，姿勢の制約による指標に関して述べる．今回のモデルでは姿勢が変化することによる各状態量への影響は考慮していないが，軌道上で姿勢が変化すると日照条件や入放熱量など，様々な波及効果が考えられ，衛星の状態が大きく変化する．そのため本手法では姿勢変化を起こすことは，衛星の生存にとってリスクの大きな動作であると考え,「姿勢変化を起こすか否か」を二つ目の指標として提示する．

最後に，コマンドによる波及効果の大きさを示す指標に関して述べる．上述したように，不具合発生時は状態に対する不確定性が大きいため，状態を大きく変化させるようなコマンドは危険であると言える．そこで，コマンドによって発生する衛星内部状態の変化の大きさを「コマンドを打つことで変化するテレメトリの数」を用いて定量的に示す．これは，事前にコマンドの定義によって定められているため，コマンドを選択した時点で一意に決まる．この情報を示すことで，コマンドが引き起こす衛星内部の状態変化の大きさを人間に対して認識させることが可能である．
以上で述べたコマンドの副作用を示す 3 つの指標を以下に再掲する．

- コマンドを打つ前のバッテリ残量と，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
- コマンドを打つことで変化するテレメトリの数

4.3.2　コマンドの故障候補切り分け能力

運用時には可視時間が限られており，その可視パス中に不具合原因を特定しなければなら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不具合分析が行えることが重要である．
効率的な不具合分析を行うためには，一度に確認できる故障候補の数が多いことが望ましい．コマンドによ

る検証を行う際，検証結果が正常テレメトリであれば，そのコマンドとテレメトリで形成される経路内にある故障候補は正常であると言えるため，故障候補の切り分けを行うことができる．一方で，選択したコマンドによる検証結果が異常テレメトリであった場合，伝達する情報が経路内のどこで異常になったかが分からなければ，その経路内に存在する故障候補の切り分けを行うことはできない．そのため，経路内に多くの故障候補が存在する場合でも，切り分けの能力が高いとは言えない．故障候補の切り分け能力を考えるためには，検証結果が正常，異常に関係なく経路内にある故障候補をどれだけ確認できるかが重要になる．そのため以下では，故障候補にあるリンクが正常に情報を伝達できる確率を用いて，コマンドが確認できるリンクの数を見積もる．

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 8 に示す例を用いて示す．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．

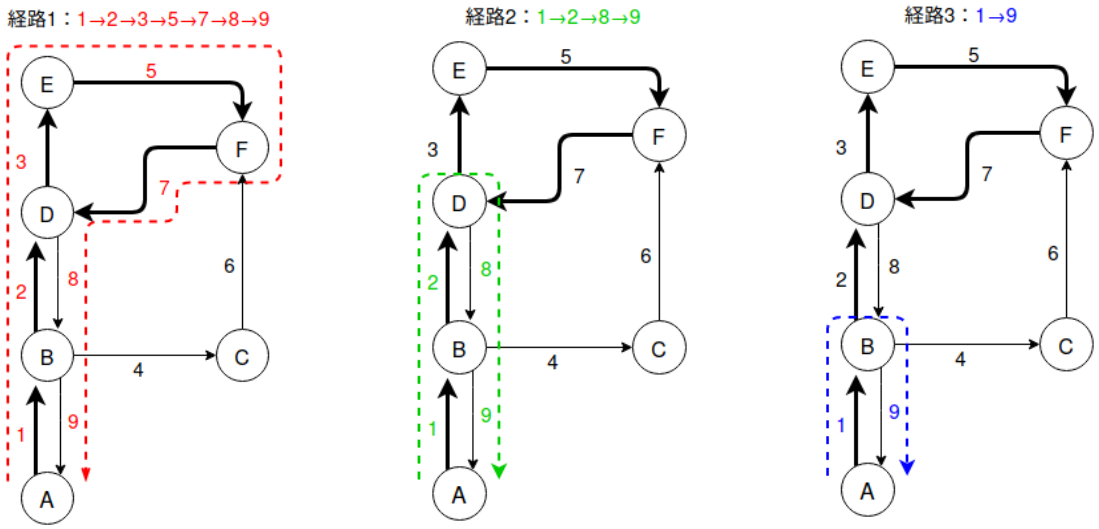

Figure 8: 故障候補とそれを確認するための情報伝達経路の例

Figure 8 では，あるコマンド C_1 によって影響を受けるテレメトリが 3 つ存在する場合を示している．各テレメトリとコマンド C_1 が形成する経路は異なり，それぞれ経路 1，2，3 としている．
この時，それぞれの経路に関してリンク 1 を確認することができる確率を考えることにする．まず，経路 1 でリンク 1 の確認をするためにはノード B からノード D までの経路 (2,3,5,7) が正常である必要がある．ここで，経路を表す記号を R，経路内にある故障候補リンクの集合を $\mathbb{F}$ とすると，経路 1 を通る情報でリンク 1 を確認することができる確率は

$$P(l_1|\mathrm{R}_1) = \prod_{i\in\mathbb{F}_1, i\neq 1} P(l_i = \text{normal}) \tag{2}$$

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

であることが分かる．ここで，R_1 は経路 1，また

$$\mathbb{F}_1 = \{2,3,5,7\} \tag{4}$$

である．
同様に経路 2，3 に関してもリンク 1 を確認することができる確率を求めると

$$P(l_1|\mathrm{R}_2) = \prod_{i\in\mathbb{F}_2, i\neq 1} P(l_i = \text{normal}) \tag{5}$$

$$= \frac{1}{2} \tag{6}$$

$$P(l_1|\mathrm{R}_3) = \prod_{i\in\mathbb{F}_3, i\neq 1} P(l_i = \text{normal}) \tag{7}$$

$$= 1 \tag{8}$$

となる．このように，あるリンク l_i を通る経路が複数存在する場合，経路に依存してそのリンク l_i を確認できる確率（以下では確認可能性とする）が変わる．ここで，あるコマンド C_k による情報伝達経路の中で，リンク l_i を通る経路が複数存在する場合には，その経路のリンク l_i の確認可能性はそれらの最大値を取るものとする．コマンド C_k が影響を与える各テレメトリと成す経路の内，リンク l_i を含むものを $\mathbb{R}_{ki} = \{\mathrm{R}_{1_i}, \cdots, \mathrm{R}_{N_{ki}}\}$（ただし N_{ki} はリンク l_i を含む経路の数）とすると，コマンド C_k によるリンク l_i の確認可能性は

$$P(l_i|\mathrm{C}_k) = \max\{P(l_i|\mathrm{R}_{1_i}), \cdots, P(l_i|\mathrm{R}_{N_{ki}})\} \tag{9}$$

となる．
これは経路 $\mathbb{R}_{ki}$ 内にある故障可能性リンク全てに対して求めることができるので，これらの平均を取り，そのコマンドの「平均確認可能性」と定義する．平均確認可能性は，コマンド C_k が影響を与える各テレメトリと成す経路を $\mathbb{R}_k = \{\mathrm{R}_1, \cdots, \mathrm{R}_{N_k}\}$ とし，それらに含まれる故障可能性リンクの数を N_{F_k} とすると

$$P_m(\mathrm{C}_k) = \frac{1}{N_{F_k}} \sum_{i=1}^{N_{F_k}} P(l_i|\mathrm{C}_k) \tag{10}$$

とできる．これは，コマンドとテレメトリが通る経路に含まれる故障候補の内，どれだけのリンクの状態を確認できるかという指標である．つまり，この指標が高いほど経路内に存在する故障可能性リンクの多くを確認できるということになる．

また，平均確認可能性を経路 $\mathbb{R}_{ki}$ 内にある故障可能性リンクの数 N_{F_k} にかけると，コマンド C_k によって確認できるリンク数の期待値を求めることができ，

$$E(\mathrm{C}_k) = N_{F_k} P_m(\mathrm{C}_k) \tag{11}$$

$$= \sum_{i=1}^{N_{F_k}} P(l_i|\mathrm{C}_k) \tag{12}$$

となる．これを「確認可能リンク数」と定義する．

ここで，Figure 8 に示すコマンド 1 に関して平均確認可能性及び，確認可能リンク数を計算してみると

$$P_m(\mathrm{C}_1) = \frac{1}{N_{F_1}}\{P(l_1|\mathrm{C}_1) + P(l_2|\mathrm{C}_1) + P(l_3|\mathrm{C}_1) + P(l_5|\mathrm{C}_1) + P(l_7|\mathrm{C}_1)\} \tag{13}$$

$$= \frac{1}{5}\{1 + \frac{1}{2} + \left(\frac{1}{2}\right)^4 + \left(\frac{1}{2}\right)^4 + \left(\frac{1}{2}\right)^4\} \tag{14}$$

$$= 0.3375 \tag{15}$$

$$E(\mathrm{C}_1) = 1.6875 \tag{16}$$

となる．結果からわかるように，通る経路に存在する故障候補の数が必ずしも確認できるリンクの数に対応しているわけではない．故障候補にあるリンクを通ることで不確定性が蓄積されるため，全体として経路内にあるリンクを確認できる確率は小さくなる．平均確認可能性が高く，確認可能リンク数も高いものが故障候補の切り分け能力が高いコマンドであると言える．

4.3.3　評価指標の使い分け

次に，地上試験と軌道上での運用とで上述した指標の使い分けに関して説明する．

地上試験では，電源供給に関してはバッテリではなく安定化電源を用いた試験コンフィギュレーションで行うことが多い．そのため，上述した電力の制約に関しては地上試験で考慮する必要はない．また，試験時は衛星を試験台に固定して行うため，姿勢変化に関する制約も考慮する必要はないと考えられる．これらを踏まえると，地上試験で安全を重視して二次故障などを引き起こさないように切り分けを行うためには，波及効果の大きさを示す指標である「コマンドによって影響を受けるテレメトリの数」が小さなコマンドを選択すれば良い．
また，地上試験では部分的なコンポーネントを組み上げた状態による試験も行う．システムを構成するコンポーネントの種類によっては，コマンドの効果によって二次故障が発生する可能性はあまりない場合も考えられる．そのような際には,「平均確認可能性」及び「確認可能リンク数」が大きなコマンドを選択することで効率的な切り分けが行える．

一方で，運用中は先ほど述べた電力や姿勢に関する制約を考える必要がある．また，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．よって，運用時は上で示した全ての指標を考慮してコマンドの選択を行うことが望ましい．

4.4　故障候補を切り分けるためのコマンド及び確認事項の探索

4.1 で述べた不具合分析アルゴリズムにおける，故障候補の中から切り分けを行うためのコマンド及び確認事項の探索に関して，本手法における仮定とともに詳しく説明する．
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．そのため，不具合発生時に取得しているテレメトリの中から不具合原因特定に役に立つテレメトリ情報が存在するのであれば，そのテレメトリを確認事項として提案する．
その後，衛星の状態を変化させること無く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，残ったコマンドで故障候補の状態を確認できるものを探索し，上で示した指標の計算を行い提示する．

本手法では簡単のため，コマンド及びテレメトリが情報を伝達する経路において，故障候補を通るものがあれば「確認できる可能性がある」としている．一方で，各コンポーネントの状態を一切変えないコマンドや，コマンドを送ってもテレメトリに変化として現れない組み合わせは，不具合原因特定のために得られる情報がないため「確認できる可能性がない」としている．
ここであくまでも「確認できる可能性」として記述しているのは，コマンドの効果についての説明で言及したように，通る経路に故障候補があったとしても，途中の経路で情報が途切れてしまえば確認することはできないからである．

5　実践例

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．

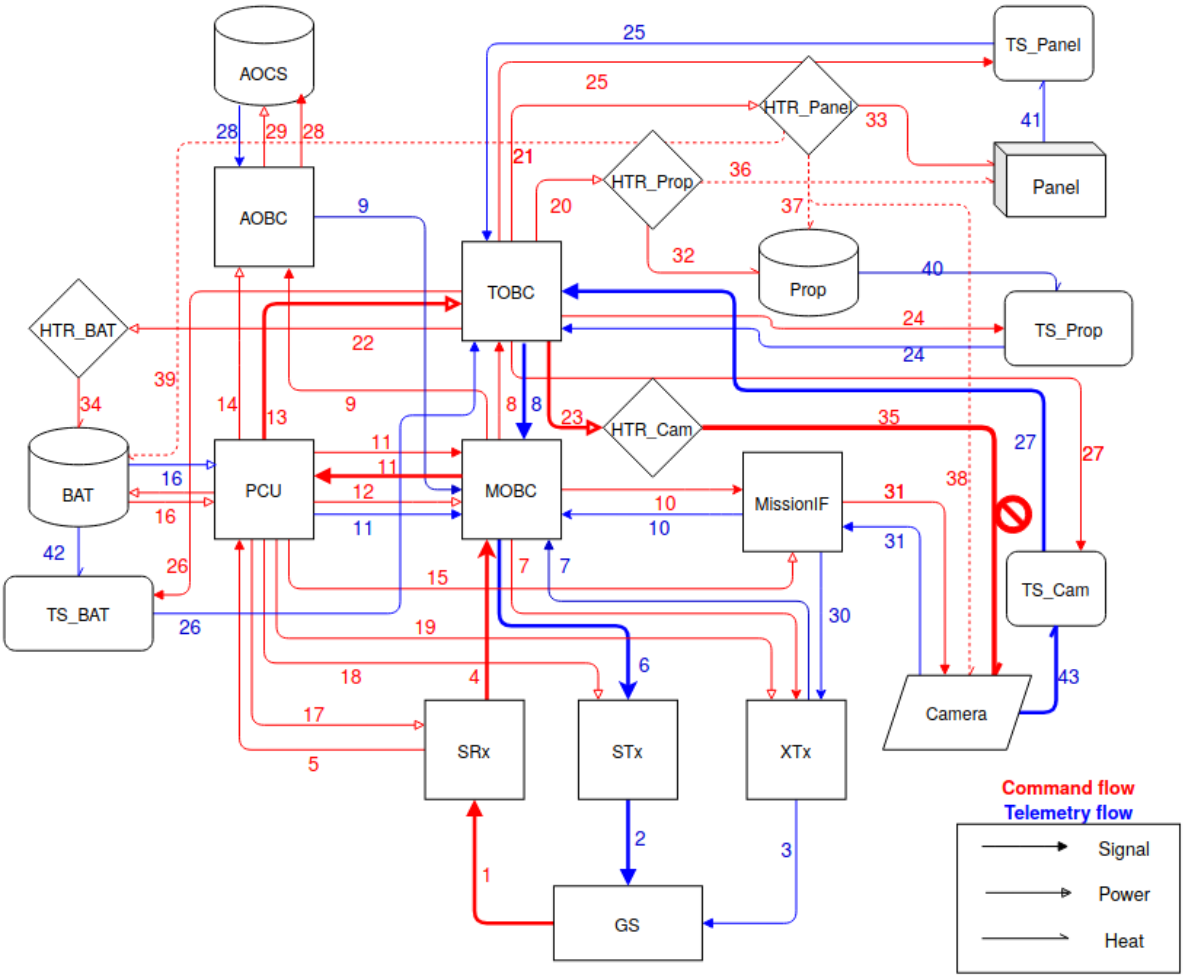

Figure 9: 故障モード (HTR_Cam ― Camera 間)

Figure 9 に示すような，カメラヒーターカメラ間接触不良が発生している場合を考える．この時，異常検知の際の不具合事象としては，

- カメラヒータ ON コマンド (ID:15) を送信したのに，カメラ温度 (ID:18) が上昇しない

という事象である．以下に，この事象に本手法を適用した例を示す．まず，Figure 10 に故障候補の決定及び，テレメトリ情報を用いた確認の段階を示す．故障候補の決定では，不具合事象を検知するきっかけとなったコマンドとテレメトリが形成する経路を探索し，targetTEL，targetCOM として提示している．
その後，時間変化するテレメトリ情報を用いて確認できる故障候補を提示し，返ってくるテレメトリが正常かどうかを入力させることで，切り分けを行っている．

```
targetTEL: [43, 27, 8.0, 6.0, 2.0]
targetCOM: [1, 4, 11, 13, 23, 35]
TELtarget: [43, 27, 8.0, 6.0, 2.0]
Telemetry 1 ( MOBC_Counter ) can verify following links
 [6, 2]

Please check MOBC_Counter
Input result(OK or NG)>>OK
TELtarget: [43, 27, 8.0]
Telemetry 2 ( TOBC_Counter ) can verify following links
 [8]

Please check TOBC_Counter
Input result(OK or NG)>>OK
```

Figure 10: テレメトリによる確認

次に，Figure 11 に示すのが，不具合発生時に送信していたコマンド情報から考えられるテレメトリの変化を用いて故障候補の確認を行う段階である．今回は，初期コマンドとしては異常検知の際に送ったコマンド (カメラヒータ ON) のみを考えている．

```
 Check telemetries which influenced by initial Command state

COMtarget: [1, 4, 11, 13, 23, 35] TELtarget: [43, 27]
Command 15 ( HTR_CAM_ON ) & Telemetry 5 ( MOBC_COM_Counter ) can verify following links
 COMlink: [4, 1] TELlink []
Command 15 ( HTR_CAM_ON ) & Telemetry 6 ( TOBC_COM_Counter ) can verify following links
 COMlink: [13, 11, 4, 1] TELlink []
Command 15 ( HTR_CAM_ON ) & Telemetry 10 ( TOBC_Current ) can verify following links
 COMlink: [11, 4, 1] TELlink []
Command 15 ( HTR_CAM_ON ) & Telemetry 22 ( HTR_CAM_Current ) can verify following links
 COMlink: [13, 11, 4, 1] TELlink []

Please check MOBC_COM_Counter
Input result(OK or NG)>>OK
COMlink: [4, 1] & TELlink: [] were verified

Please check TOBC_COM_Counter
Input result(OK or NG)>>OK
COMlink: [13, 11] & TELlink: [] were verified
```

Figure 11: 初期コマンドを用いた確認

以下の Figure 12 に示すのが，上記の流れを経て残った故障候補を確認できるコマンドを探索し，指標と共に提示した結果である．

```
COMtarget: [23, 35] TELtarget: [43, 27]
COM 13 HTR_PANEL_ON {'candidate link number': 2, 'Check link number': 1.0, 'Mean Pr
obability of check': 0.5} {'impact TEL num': 8, 'Remaining Power': 4.80000000000000
1, 'Power consume by this COM': 2}
COM 19 HTR_CAM_OFF {'candidate link number': 4, 'Check link number': 0.5, 'Mean Pro
bability of check': 0.125} {'impact TEL num': 5, 'Remaining Power': 4.8000000000000
01, 'Power consume by this COM': -1}
```

Figure 12: コマンドの選択肢表示

```
Please select Command above(input ID) >>19
Command 19 ( HTR_CAM_OFF ) & Telemetry 18 ( CAM_Temp ) can verify following links
 COMlink: [35, 23] TELlink [43, 27]

Please check CAM_Temp
Input result(OK or NG)>>NG
selected Command: [15, 19] remaining Command: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16, 17, 18, 2
0, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30]
COMtarget: [23, 35] TELtarget: [43, 27]
COM 13 HTR_PANEL_ON {'candidate link number': 2, 'Check link number': 1.0, 'Mean Probability of check': 0.
5} {'impact TEL num': 8, 'Remaining Power': 4.800000000000001, 'Power consume by this COM': 2}
Please select Command above(input ID) >>13

Please check CAM_Temp
Input result(OK or NG)>>OK
COMlink: [] & TELlink: [43, 27] were verified
selected Command: [15, 19, 13] remaining Command: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 16, 17, 18, 2
0, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30]
COMtarget: [23, 35] TELtarget: []
nothing can verify
finish
faulty COMlink: [23, 35] faulty TELlink: []
```

Figure 13: コマンドによる確認

今回用意したコマンドとテレメトリだけでは，最終的な故障箇所の特定まで行うことができなかった．設計段階において FMEA 上に定義されている故障モードに対して適用することで，実際の試験時にその故障を特定するために必要なコマンドとテレメトリが定義されているかの確認作業にも本手法は利用できると考えられる．

6　今後の方針

以上では，超小型衛星の信頼性向上の為の不具合分析支援の手法に関して示し，テストケースに対する実践例を示した．今後，いくつかの故障例を考えて実践し，本手法を用いて不具合分析を行った結果と，指標を提示せず任意でコマンドを選択した結果を比較し，本手法の有効性を検証したいと考えている．比較する際の評価軸としては

- 効率的に不具合の切り分けが行えたかどうか（打ったコマンドの数で評価）
- 安全に切り分けを行うことができたかどうか（電力，姿勢の変化によって評価）

を考えている．

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) Yoshinobu Kitamura and Riichiro Mizoguchi. *A Framework for Systematization of Functional Knowledge based on Ontological Engineering.* PhD thesis.

9) JAXA. 衛星の機能モデル (Functional Model of Spacecrafts (FMS)). Technical report, 2020.