東京大学工学部航空宇宙工学科
令和２年度学士論文

衛星内の情報伝達経路モデルに基づく不具合分析支援に関する研究

03-183005　西本 慎吾
指導教員 船瀬 龍 准教授
2020 年 11 月 30 日

第 1 章

序論

1.1 研究背景

1.1.1 超小型衛星の信頼性の低さ

近年，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)]．
大学衛星は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，このような超小型衛星で頻発している不具合は，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故障分析を通じて知られている [2)]．軌道上での不具合の根本原因に対する調査 (図 1.1) 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ことが分かっている [2)]．

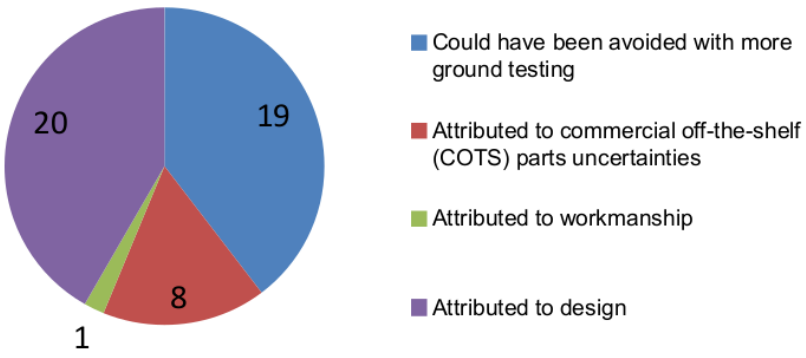

図 1.1　故障原因に関するインタビュー結果 [2)]

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，ほどほどの信頼性を実現する「ほどよし」の考え方が重要であると

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，超小型衛星のほどほどの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 (1.1) が示すように，この評価手法では設計や製造時の信頼性も重要な要素であると捉えられている．
超小型衛星のコストの低さを考慮すると，信頼性の高いコンポーネントを使用することによってそれぞれのコンポーネントの信頼性 R_{comp} を高めるより，設計や製造過程における信頼性を高めることが超小型衛星の信頼性の向上につながる．また上述したように，超小型衛星の信頼性の低さの根本原因である設計不良や地上試験の不足を改善していくことが，信頼性向上には不可欠である．

$$R_{sat} = R_{des} \times R_{fab} \times R_{comp} \times R_{op} \tag{1.1}$$

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

1.1.2 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](図 1.2) 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分かっている．
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，または発見できかった原因」が以下の図 1.3 のように知られている．試験設備の不足によって確認できなかったものや，故障発見までの時間が長く地上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，地上試験で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].

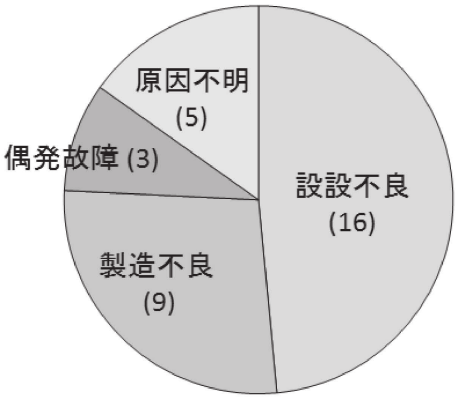

図 1.2 軌道上故障の原因類型の分布 [4)]

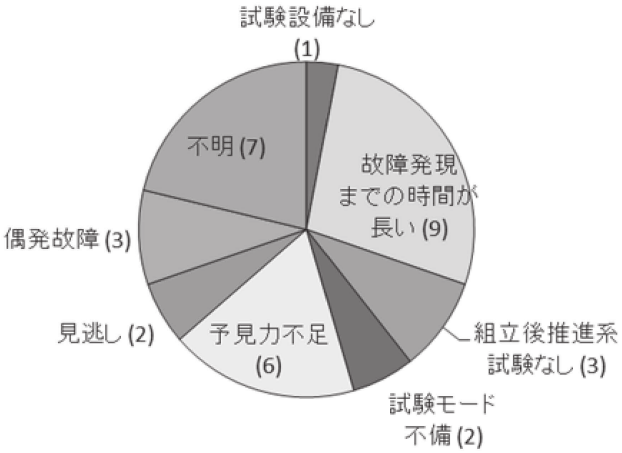

図 1.3 軌道上故障の要因を地上で発見できなかった原因類型の分布 [4)]

1.1.3 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の分析を，地上試験で十分に行えていないことが，超小型衛星の信頼性の低さの原因の一つである．
そこで，地上試験で衛星の不具合及びリスク分析が十分に行えていない原因を具体的に示すため，以下に人間による不具合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．

まず，地上試験において十分に不具合分析が行えていない原因の 1 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主に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類推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．また，衛星のように内部の機器が複雑に絡み合ったシステムでは，人間が想定していないつながりが多数存在するため，不具合事象から全ての故障可能性を洗い出すことは難しい．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)] が構築したシステムを用いて洗い出した故障モードと比較して，不足しているという結果も出ている．このように，人間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，人の知識や経験に依存し，考えが及んでいないことで見逃している故障モードが多く存在する．
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

がある．
上述したように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．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．不具合発生時は衛生の状態を十分に把握できていない状況であるため，故障仮説を検証する際，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，意図しなかった動作を起こし衛星の生存を脅かす危険性がある．このため不具合原因特定を行う際には，不具合分析に用いるコマンドが「安全」なのか，という点が非常に重要となる．

1.1.4 不具合分析に関する先行研究

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に関する研究が盛んに行われている．以下の表 1.1 に，モデルに基づいて行う機械などを対象にした不具合分析，故障診断手法に関してまとめた．

表 1.1 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	故障箇所特定支援

まず，GDE はモデルを元に行う不具合分析手法として一般的なもので機器の正常時の制約モデルを元にして故障仮説の生成を行う．GDE に対して，故障時モデルを組み込んだものが GDE+[6] と呼ばれる手法であり，入出力の観測結果から正常時との不整合を検知し，その不整合を説明するための仮説を洗い出すことを行っている．
また，山口ら [5] は，衛星内部の機器の接続関係だけでなく，衛星が起こすアクションや状態などのつながりをモデル化し想定していた機器間の接続関係からだけでは見えていなかった波及効果を洗い出すことを可能にしている．
また，來村ら [7] は故障を事象としてだけでなく，時間経過や伝搬過程を含めて捉えるために必要となる概念を故障オントロジー [8] として定義し，故障箇所に対する仮説だけでなく，より遡った故障原因に対する仮説を網羅的に生成する手法を提案している．
以上のように故障仮説生成の研究に関しては，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．また，故障候補の洗い出しを十分に行うことができたとしても，特定を行うことができなければ地上試験

によって設計や製造における不備を除くことができない.

1.2 研究概要

1.2.1 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以下の機能を満たす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.

- 故障候補を確認するためのコマンド及びテレメトリを提案する.
- コマンドの選択肢を選ぶ際の判断の指標を定量的に提示する.

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.

- 衛星内部機器の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アルゴリズム
- コマンドの安全性，及び故障候補切り分け能力を示す指標

1.2.2 本論文の構成

本論文は次のような構成となっている．第 2 章では，本研究で提案する不具合分析手法で使用する衛星のモデルと不具合分析のアルゴリズム，探索された結果を選択する際に必要となる指標に関して，安全性と故障候補の切り分け能力の観点から述べている．また，第 3 章では提案手法を実践した結果に関してまとめており，本手法によって故障箇所特定のプロセスを体系化できたこと，指標に関する考察を述べる.

第 2 章

情報伝達経路モデルに基づく対話的不具合分析手法の仕様

2.1　概要

本研究では，衛星の不具合分析において故障仮説の検証を支援するシステムを提案する．

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，人間が実機に対して打つコマンドを選択する際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衛星の制約モデルを用いて行うためには，非常に忠実度の高いモデルが必要である．衛星は複雑に物理現象が絡み合うため，物理法則に基づいた事象が各コンポーネント間を伝搬する様子を表現することはモデル化のコストが非常に大きい．むしろ，人間を対話的に支援することによってモデルに求められる忠実度のレベルを下げつつ不具合分析の過程を体系化することができ，経験の少ないエンジニアの支援ができる．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，その検証結果をシステムに反映することで，対話的に故障箇所の特定を行っていく構成となっている．

2.2　不具合分析アルゴリズム

以下の図 2.1 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，図 2.1 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．

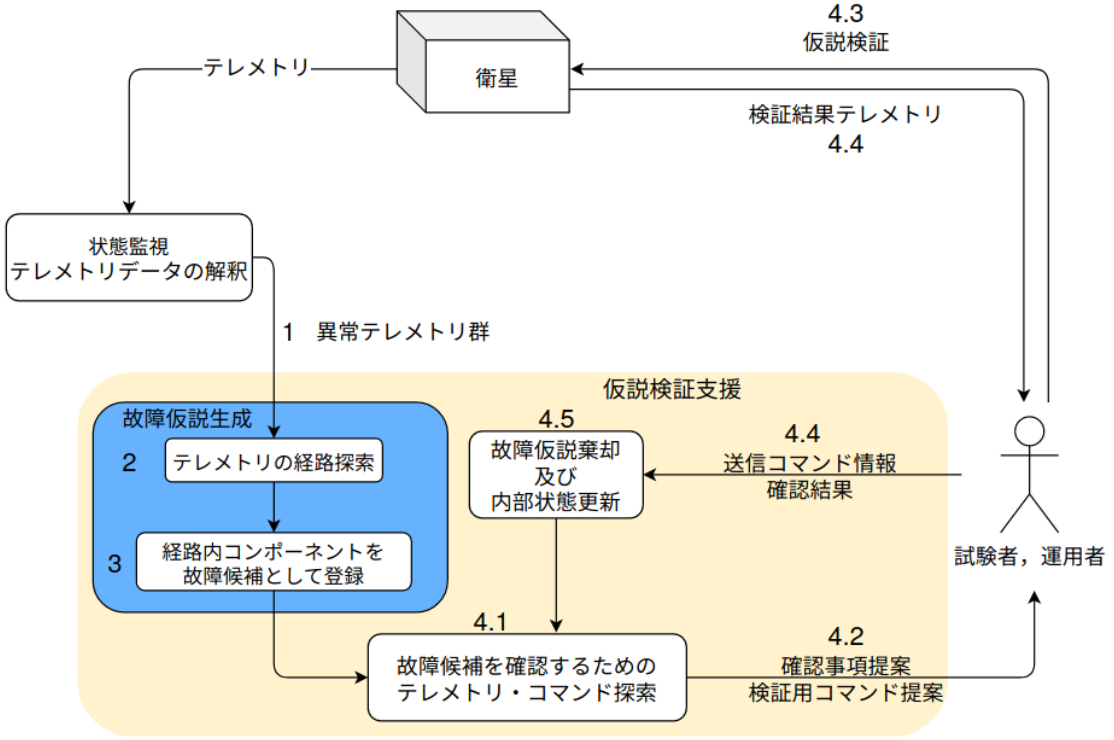

図 2.1　不具合分析の流れ

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 1 で得たテレメトリに影響を与えるコマンドを送信されてから，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．

2.2.1 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

以上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細を述べる．
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．

2.3 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．

2.3.1 対象とするテストケース

今回，以下の図 2.2 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

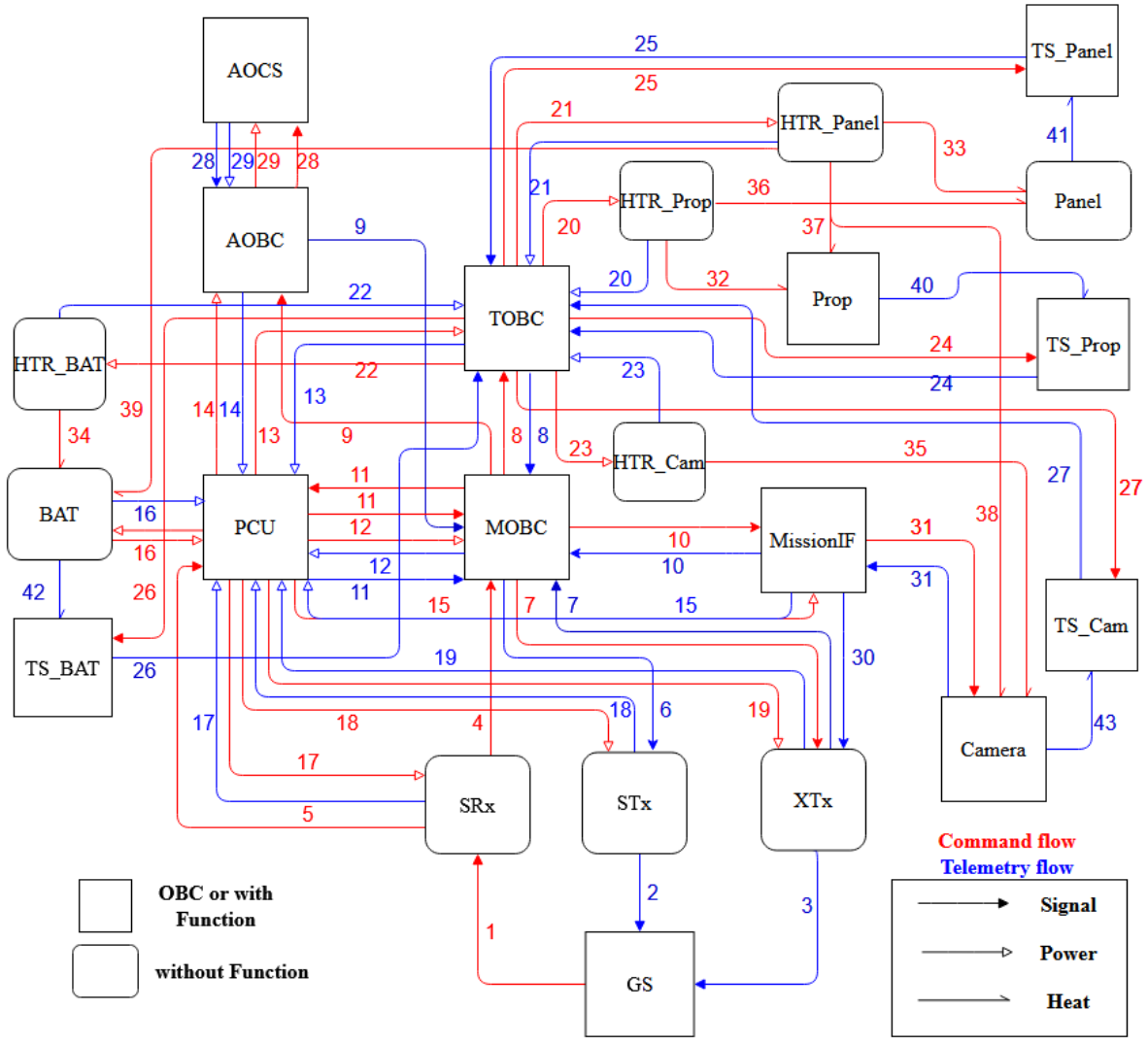

図 2.2　簡易衛星モデル

2.3.2　各コンポーネント間の接続関係モデル

来村ら [9] 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，以下の表 2.1 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．

表 2.1　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．以下の表 2.2 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．

表 2.2　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図 2.3 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．初期状態を図 2.3 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":{}},
```


図 2.3　コンポーネント初期状態例

2.3.3　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表 2.3，2.4 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．時間変化するテレメトリに関しては，コマンドによって状態変化をさせなくても変化を確認することが可能なので，不具合分析の初めのアプローチに利用可能である．

表 2.3　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	12	11	6	2	
10	TOBC_Current	Command	13	11	6	2	
11	AOBC_Current	Command	14	11	6	2	
12	MIF_Current	Command	15	11	6	2	
13	SRx_Current	Command	17	11	6	2	
14	STx_Current	Command	18	11	6	2	
15	XTx_Current	Command	19	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	21	8	6	2	
21	HTR_PROP_Current	Command	20	8	6	2	
22	HTR_CAM_Current	Command	23	8	6	2	
23	HTR_BAT_Current	Command	22	8	6	2	
24	BAT_Power	Command	16	11	6	2	
25	AOCS_Current	Command	29	9	6	2	
26	RW_RotateSpeed	Command	28	9	6	2	
27	M_DATA	Command	31	30	3		
28	CAM_Status	Command	31	10	6	2	

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，表 2.3 に示すテレメトリの経路及び，表 2.4 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[10] 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図 2.4 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．

表 2.4　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	5,6,10,16,17,18,19,20	ACTION	1	4	8	21	33,37,38,39
14	HTR_PROP_ON	5,6,10,16,17,21	ACTION	1	4	8	20	32,36
15	HTR_CAM_ON	5,6,10,18,22	ACTION	1	4	8	23	35
16	HTR_BAT_ON	5,6,10,19,23	ACTION	1	4	8	22	34
17	HTR_PANEL_OFF	5,6,10,16,17,18,19,20	ACTION	1	4	8	21	33,37,38,39
18	HTR_PROP_OFF	5,6,10,16,17,21	ACTION	1	4	8	20	32,36
19	HTR_CAM_OFF	5,6,10,18,22	ACTION	1	4	8	23	35
20	HTR_BAT_OFF	5,6,10,19,23	ACTION	1	4	8	22	34
21	AOCS_ON	5,7,11,25	ACTION	1	4	9	29	
22	AOCS_OFF	5,7,11,25	ACTION	1	4	9	29	
23	RW_START	5,7,11,26	ACTION	1	4	9	28	
24	RW_STOP	5,7,11,26	ACTION	1	4	9	28	
25	M_DATA_DOWN	5,8	GET	1	4	10	31	
26	GET_PANEL_TEMP	5,6	GET	1	4	8	25	
27	GET_PROP_TEMP	5,6	GET	1	4	8	24	
28	GET_CAM_TEMP	5,6	GET	1	4	8	27	
29	GET_BAT_TEMP	5,6	GET	1	4	8	26	
30	TAKE_PICTURE	5,8,27,28	ACTION	1	4	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"]}]},
```


図 2.4　コマンドの機能モデル

2.4　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．

また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内に故障候補が存在していれば，その故障候補を「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，情報が通る経路に故障候補が含まれていたとしても，伝達の途中で情報が途切れてしまえば，その故障候補の状態を確認することはできないためである．

2.4.1　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 にするといった行為は，衛星が生存するために必要な機能を動作させるための電力を枯渇させる可能性があるため，危険な行為であると言える．そのため，故障箇所の特定を行うためにコマンドを打つ際には，現在の衛星の電力状態を把握し，コマンドを打つことで電力不足にならないかを確認しながら行動を起こさなければならない．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，未熟なエンジニア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．そのため本手法では，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．ここでは電力による制約を簡単に表現するため，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．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．一方で，上で述べたように不具合発生時には衛星の状態に不確定な要素が多く含まれているため，意図せず姿勢変化を起こし状態を大きく変化させることは非常に危険である．本手法で用いる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，将来的に実ミッションで使用することを考えると，姿勢変化は衛星の生存にとってリスクの大きな動作であるとため，「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，状態を大きく変化させるようなコマンドを故障個所の特定のために用いることは望ましくない．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「コマンドに影響を受けるテレメトリ」と，人間からフィードバックを受けながら更新される衛星内部コンポーネントの状態から求めることが可能である．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

2.4.2 コマンドの故障候補切り分け能力

運用時には，通信可能な時間（以下，可視時間）が限られており，その時間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面が存在する．運用形態によっては，可視時間が非可視時間に比べて非常に短いこともあり，その際には一つの可視時間を逃すとミッション失敗につながるため，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．

以下では，一つのコマンドの故障候補切り分け能力を表す指標と，ある故障候補がある際にどのコマンドから検証を始めれば最終的に少ないコマンド数で終えることができるかを表す指標の 2 点に関して述べる．

コマンドが確認できるリンクの数

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．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．このことを以下の図 2.5 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．また故障候補以外は正常であるとし，正常なリンクに関しては $P(l_i = \text{normal}) = 1$ である．

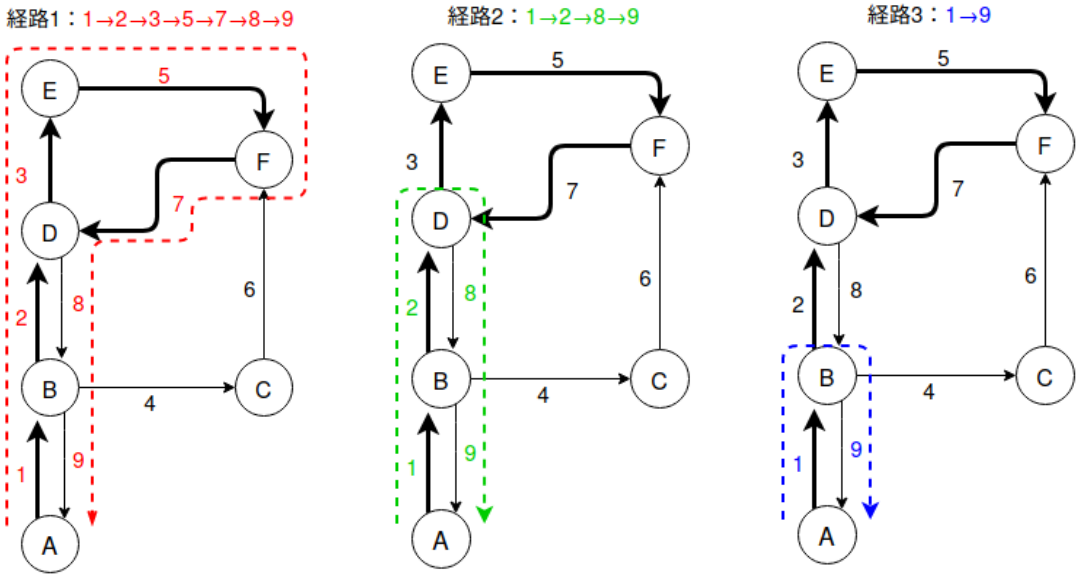

図 2.5　故障候補とそれを確認するための情報伝達経路の例

図 2.5 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

$$P(l_1|\mathrm{R}_1) = \prod_{i\in\mathbb{F}_1, i\neq 1} P(l_i = \mathrm{normal}) \tag{2.1}$$

$$= \left(\frac{1}{2}\right)^4 \tag{2.2}$$

であることが分かる．ここで，R_1 は経路 1，また

$$\mathbb{F}_1 = \{2, 3, 5, 7\} \tag{2.3}$$

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

$$P(l_1|\mathrm{R}_2) = \prod_{i\in\mathbb{F}_2, i\neq 1} P(l_i = \mathrm{normal}) \tag{2.4}$$

$$= \frac{1}{2} \tag{2.5}$$

$$P(l_1|\mathrm{R}_3) = \prod_{i\in\mathbb{F}_3, i\neq 1} P(l_i = \mathrm{normal}) \tag{2.6}$$

$$= 1 \tag{2.7}$$

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，リンク l_i に対する確認可能性が最大となる経路を用いて確認すればいいので，コマンド C_k によるリンク l_i の確認可能性はそれらうち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

$$P(l_i|\mathrm{C}_k) = \max\{P(l_i|\mathrm{R}_{1_i}), \cdots, P(l_i|\mathrm{R}_{N_{ki}})\} \tag{2.8}$$

となる．
式 (2.8) 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の集合を $\mathbb{R}_k = \{\mathrm{R}_1, \cdots, \mathrm{R}_j, \cdots, \mathrm{R}_{N_k}\}$ とし，それぞれの経路内に存在する故障可能性リンクの数を $N_{F_{kj}}$，集合 $\mathbb{R}_k$ 全体で考えた時の数を $\mathbf{N_{F_k}}$ とすると

$$P_m(\mathrm{C}_k) = \frac{1}{\mathbf{N_{F_k}}} \sum_{i=1}^{\mathbf{N_{F_k}}} P(l_i|\mathrm{C}_k) \tag{2.9}$$

と表すことができる．平均確認可能性は，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うち，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 $\mathbb{R}_k$ 内にある故障可能性リンクの数 $\mathbf{N_{F_k}}$ 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，

$$E(\mathrm{C}_k) = \mathbf{N_{F_k}} P_m(\mathrm{C}_k) \tag{2.10}$$

$$= \sum_{i=1}^{\mathbf{N_{F_k}}} P(l_i|\mathrm{C}_k) \tag{2.11}$$

となる．これを「確認可能リンク数」と定義する．

ここで，図 2.5 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

$$P_m(\mathrm{C}_1) = \frac{1}{\mathbf{N_{F_1}}}\{P(l_1|\mathrm{C}_1) + P(l_2|\mathrm{C}_1) + P(l_3|\mathrm{C}_1) + P(l_5|\mathrm{C}_1) + P(l_7|\mathrm{C}_1)\} \tag{2.12}$$

$$= \frac{1}{5}\left\{1 + \frac{1}{2} + \left(\frac{1}{2}\right)^4 + \left(\frac{1}{2}\right)^4 + \left(\frac{1}{2}\right)^4\right\} \tag{2.13}$$

$$= 0.3375 \tag{2.14}$$

$$E(\mathrm{C}_1) = 1.6875 \tag{2.15}$$

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．

故障箇所特定のためにかかるコマンドの総数

次に，コマンドを選択する順番によって，故障箇所を特定するために打つコマンドの総数に変化が現れることを示し，コマンドの総数の見積もりに関して述べる．まず，上で定義した各リンクに関する正常確率を用いることによって，各径路ごとの平均確認可能性を以下の式 (2.16) のように求めることが可能になる．これを「経路別確認可能性」と定義する．

$$P_m(\mathrm{R}_j) = \frac{1}{N_{F_{kj}}} \sum_{i=1}^{N_{F_{kj}}} P(l_i|\mathrm{R}_j) \tag{2.16}$$

あるコマンドを送った際にテレメトリを確認するときは「経路別確認可能性」が高い順番に行うことで，経路内にあるリンクの状態を確認し故障候補から棄却する，または故障箇所であると特定する可能性が高くなるため，効率的に絞り込むことが可能になる．また，各テレメトリの検証結果に応じてそれ以降に確認するテレメトリによるリンクの確認可能性が変化する．つまり，各テレメトリの結果が正常である（もしくは異常である）確率は他のテレメトリの結果に依存することになる．

まず簡単のため，他のテレメトリの結果を考慮せずにテレメトリが正常（または異常）となる確率に関して述べる．テレメトリの結果が正常であるためには，そのテレメトリが通る経路内のリンクがすべて正

常であればよいので，コマンド C_k を送った時にテレメトリ T_j が正常または異常である確率は以下の式 (2.17)，(2.18) のようになる．この時，経路の添え字とテレメトリの ID が対応している．

$$P(\mathrm{T}_j = \mathrm{normal}) = \prod_{i \in \mathbb{F}_j} P(l_i = \mathrm{normal}) \tag{2.17}$$

$$P(\mathrm{T}_j = \mathrm{abnormal}) = 1 - P(\mathrm{T}_j = \mathrm{normal}) \tag{2.18}$$

次に，上述したように「経路別確認可能性」が高い順でテレメトリを確認して行った場合に，各テレメトリの結果に応じて以降のテレメトリの正常（または異常）確率がどのように変化していくのかを示す．以下の図 2.6 では，上で示した例（図 2.5）に関して各テレメトリの結果ごとに，それ以降の結果を場合分けしたものを示している．この時，図 2.5 の例では確認可能性は経路 3，2，1 の順に高いため，その順番に従って検証を行っている．

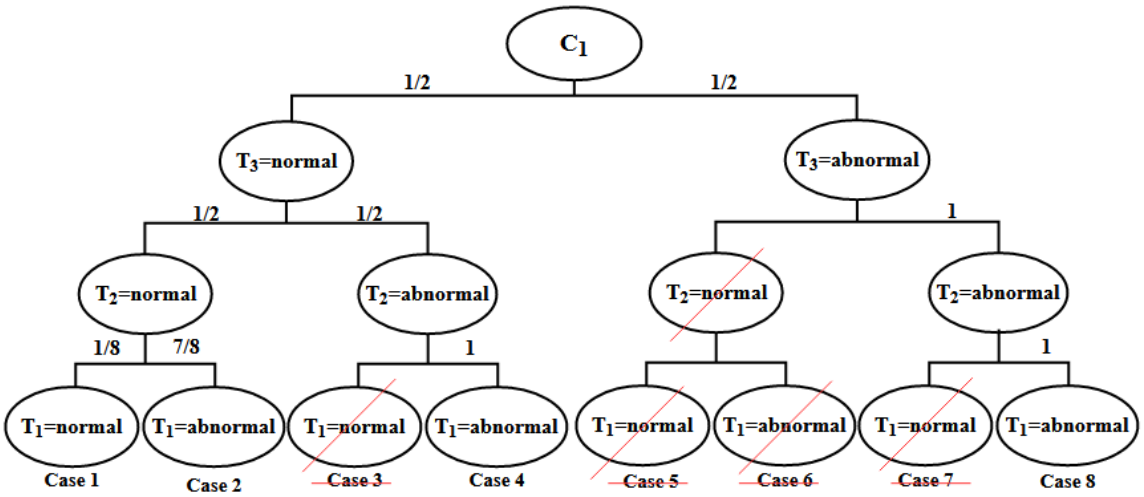

図 2.6　各テレメトリの結果による検証過程の種類

まず，コマンド 1 を送信した際にテレメトリ 3 を確認すると，図 2.5 の経路 3 が情報伝達経路であるため，その経路内に存在する故障候補はリンク 1 のみであり，そのテレメトリの結果の確率は

$$P(\mathrm{T}_3 = \mathrm{normal}) = P(l_1 = \mathrm{normal}) = \frac{1}{2} \tag{2.19}$$

$$P(\mathrm{T}_3 = \mathrm{abnormal}) = P(l_1 = \mathrm{abnormal}) = \frac{1}{2} \tag{2.20}$$

と求まる．これ以降の各テレメトリの結果の確率は T_3 の結果に依存することになる．まず，T_3 が正常である場合を考えると，これによってリンク 1 は正常であることが確認できるためリンク 1 は故障候補から除かれ，テレメトリ 2 の結果に関する確率は図 2.6 に示すように求まる．同様に，テレメトリ 1 の結果に関する確率も，それ以前に確認したテレメトリの結果によって経路内に存在する故障候補を更新した上で求めると図に示すようになる．

次に，T_3 の結果が異常であった場合を考えると，経路 3 に含まれる故障候補はリンク 1 のみであるため，リンク 1 の故障が確定する．この時，以降に確認するテレメトリ 2，1 の結果は，T_3 が正常である

場合のときと同様に，正常もしくは異常の二通りが考えられる．実際の衛星で使用されるテレメトリでは，接続関係に依存せずコンポーネントのみの状態によって決まる状態量を担うテレメトリも存在するため，経路の途中に異常箇所が存在していても正常なテレメトリが下りてくることは考えられる．しかし，このような場面を考えるためにはテレメトリに含まれる情報がどの状態量に対応しているのかを考えなければならない．これらの対応付けは，事前にモデルに組み込むことによって対応可能であると考えられる．ここでは扱いを簡単にするため，既知の故障箇所が含まれている経路を通るテレメトリは異常値となるという仮定をおく．そのため，一度テレメトリが異常値を示したものに関しては，以降のテレメトリも異常となる必要があるため図 2.6 の Case 3,5,6,7 のような場合は考慮しない．
これを踏まえて, Case 1,2,4,8 のようになる確率は以下のように求まる．以下では nomarl を n, abnormal を a と略記している．

$$
\begin{aligned}
P(\text{Case 1}) &= P(\mathrm{T}_3 = \mathrm{n} \cap \mathrm{T}_2 = \mathrm{n} \cap \mathrm{T}_1 = \mathrm{n}) \\
&= P(\mathrm{T}_3 = \mathrm{n}) P(\mathrm{T}_2 = \mathrm{n} | \mathrm{T}_3 = \mathrm{n}) P(\mathrm{T}_1 = \mathrm{n} | \mathrm{T}_2 = \mathrm{n}, \mathrm{T}_3 = \mathrm{n}) \\
&= \frac{1}{2} \times \frac{1}{2} \times \frac{1}{8} \\
&= \frac{1}{32} \qquad (2.21) \\
P(\text{Case 2}) &= P(\mathrm{T}_3 = \mathrm{n} \cap \mathrm{T}_2 = \mathrm{n} \cap \mathrm{T}_1 = \mathrm{a}) \\
&= \frac{7}{32} \qquad (2.22) \\
P(\text{Case 4}) &= P(\mathrm{T}_3 = \mathrm{n} \cap \mathrm{T}_2 = \mathrm{a} \cap \mathrm{T}_1 = \mathrm{a}) \\
&= P(\mathrm{T}_3 = \mathrm{n} \cap \mathrm{T}_2 = \mathrm{a}) \\
&= \frac{1}{4} \qquad (2.23) \\
P(\text{Case 8}) &= P(\mathrm{T}_3 = \mathrm{a} \cap \mathrm{T}_2 = \mathrm{a} \cap \mathrm{T}_1 = \mathrm{a}) \\
&= P(\mathrm{T}_3 = \mathrm{a}) \\
&= \frac{1}{2} \qquad (2.24)
\end{aligned}
$$

次に，図 2.6 のように，各テレメトリの結果によって分岐した Case それぞれに関して，検証のためのコマンド探索に入る必要がある．この時，以前のコマンド送信による検証結果によって残る故障候補が変化するため，以下では，Case 2 の場合を取り上げ，次のコマンドを選択する流れに関して述べる．
Case 2 の場合に残る故障候補は以下の図 2.7 のようになる．問題設定として，図 2.5 で示したような 3 つの情報伝達経路を持つコマンド 1 と，下図 2.7 のような経路を通るコマンド 2 を持つとする．この時コマンド 2 によって影響を受けるテレメトリは ID3 と 4 であり，それぞれが経路 3,4 に対応しているとする．

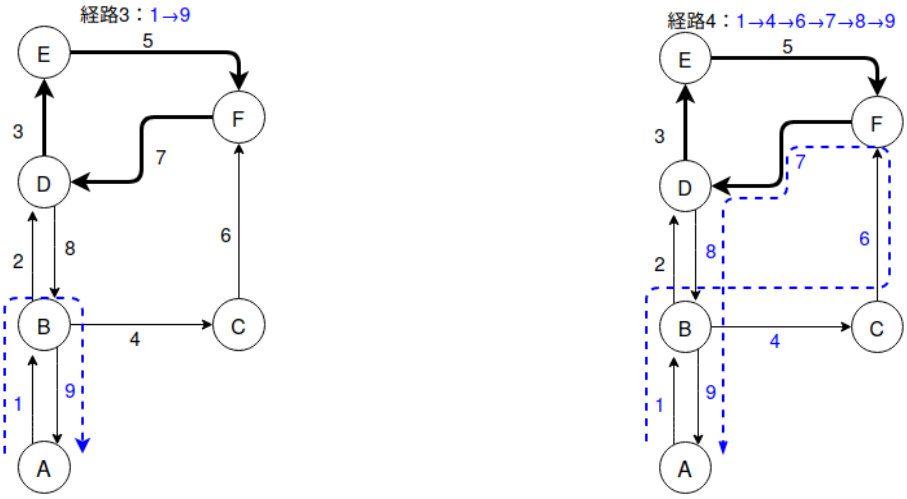

図 2.7　Case 2 の場合に残る故障候補とコマンド 2 に影響を受けるテレメトリが成す経路

Case 2 の結果になった時点で，コマンド 2 とテレメトリ 3 によって形成される経路 3 で確認できる故障候補は存在しないので，経路 4 による検証を行うことになる．経路 4 による検証結果は以下の図 2.8 のように 2 通りが考えられる．テレメトリ 4 が正常もしくは異常となる確率に関しては上述した式 (2.17), (2.18) で求められ，以下の図のようになるため，Case 2 になる確率と合わせて考えると，Case 2-1 または Case 2-2 になる確率は以下の式 (2.25)，(2.26) のように求まる．

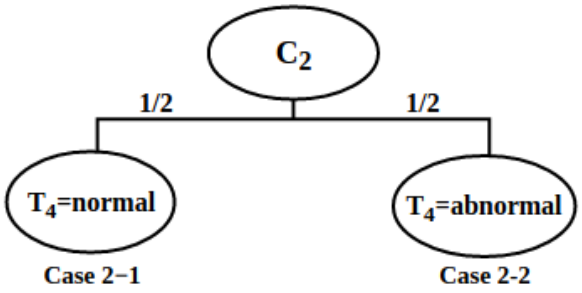

図 2.8　Case 2 の時のコマンド 2 送信時の結果

$$
\begin{aligned}
P(\text{Case 2-1}) &= P(\text{Case 2})P(\mathrm{T}_4 = \text{normal}) \\
&= \frac{7}{32} \times \frac{1}{2} = \frac{7}{64} \qquad (2.25) \\
P(\text{Case 2-2}) &= P(\text{Case 2})P(\mathrm{T}_4 = \text{abnormal}) \\
&= \frac{7}{32} \times \frac{1}{2} = \frac{7}{64} \qquad (2.26)
\end{aligned}
$$

以上のような検証のプロセスを，送信することで故障候補を切り分けられるコマンドが存在しなくなる，もしくは故障箇所を特定するまで繰り返すことで，故障候補の切り分けを行う．この流れをシステム上で先に計算し，人間がコマンドを送信する前に最終的に打つコマンドの総数を見積もることが可能である．

以下の図 2.9 に，今回説明のために使用した例 (図 2.9) における検証の全プロセスを示す．

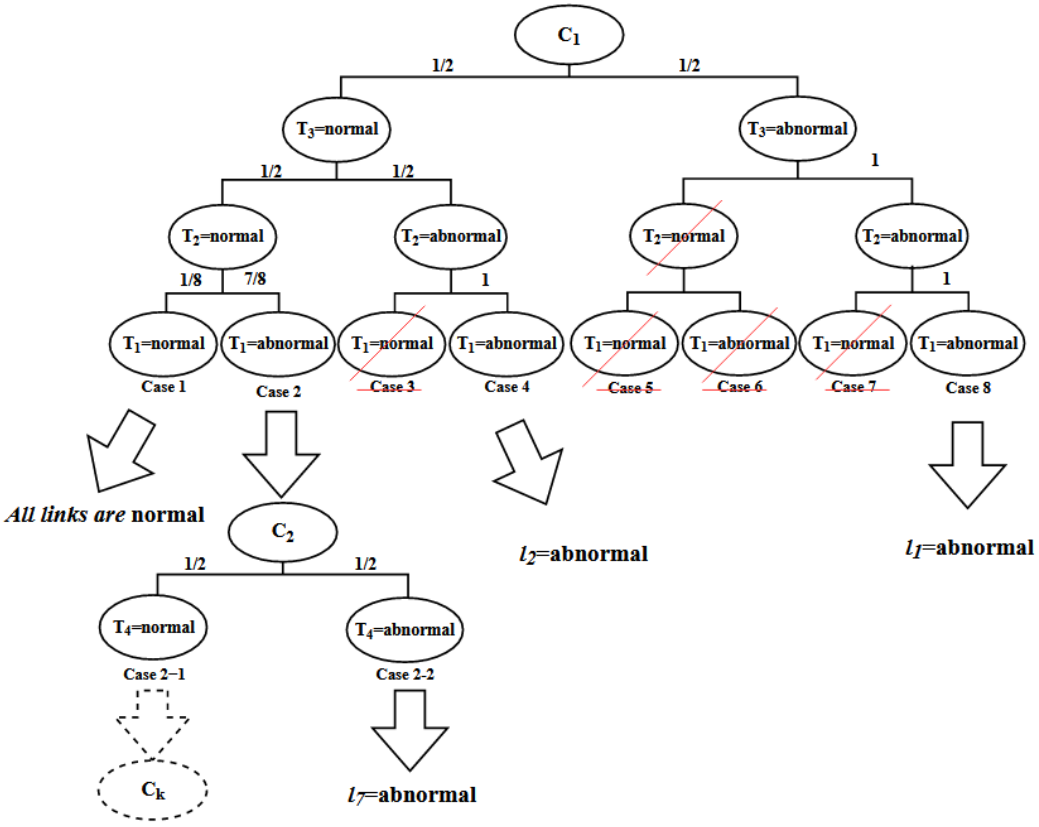

図 2.9　検証プロセスの全体像

Case 1,4,8 に関してはコマンド 1 を送信した時点で故障箇所の特定もしくは故障候補の棄却ができているので，その時点で検証は終了する．よって，検証にかかるコマンドの総数は 1 である．一方で，Case 2 の場合は，Case 2-1, 2-2 へと続くため，コマンドの総数が 2 以上となる．Case 2-2 では故障箇所がリンク 7 であると特定できたためコマンドの総数は 2 となる．また，Case 2-1 では故障候補をリンク 3 もしくは 5 に絞り込むことができたが，故障箇所の特定までは至っていない．そのため，残りの故障候補を確認できるコマンドを衛星システムが持つ場合には，次のコマンドを打つプロセスの結果によってコマンドの総数が変わる．

ここでは簡単のため衛星システムがコマンド 1 と 2 のみを持つ場合を考え，コマンドの総数の期待値を算出する．以上で各 Case になる確率は算出しているので，それを用いると検証を行う際にコマンド 1 から選択した場合のコマンドの総数は以下の式 (2.27) のようになる．ここで，$\mathbb{C}$ は検証が終了した検証結果の集合であり，今回の例では $\mathbb{C} = \{\text{Case 1,Case 2-1, Case 2-2, Case 4, Case 8}\}$ である．

$$
\begin{aligned}
N(\mathrm{C}_1) &= \sum_{\text{Case } i \in \mathbb{C}} P(\text{Case } i) N_{\text{Case } i} \\
&= \frac{1}{32} \times 1 + \frac{7}{64} \times 2 + \frac{7}{64} \times 2 + \frac{1}{4} \times 1 + \frac{1}{2} \times 1 \\
&= 1
\end{aligned}
\tag{2.27}
$$

同様にして，コマンド 2 から選択した場合も考えると，以下のように求まる．

$$N(\mathrm{C}_2) = 1.875 \tag{2.28}$$

このように，ある故障候補が残っている時に選択するコマンドの順番によってコマンドの総数の期待値が変化する．この期待値を「検証コマンド総数」と定義し，上述した指標と合わせて提示することで，どのコマンドから検証を開始することによって少ないコマンド数で検証を終えることが可能なのかを人間が直感的に認識することが可能になる．また検証プロセスにおいて，前検証結果に応じて検証コマンド総数は更新され，コマンド選択を行う際にどのコマンドを選択すれば最終的に早く検証を終えることができるのかを示すことができる．
ここで，以上で示したコマンドの故障候補切り分け能力を示す指標を以下に再掲する．

- 平均確認可能性及び確認可能リンク数
- 検証コマンド総数

以上では簡単のため，各リンクの正常確率は全て 0.5 として統一していたが，この情報は事前にモデルに組み込むことが可能であるため，実際の衛星に適した正常確率を考えることで，より効率的に故障個所の特定ができると考えられる．例えば，衛星の主要通信ラインである受信機と地上局間のリンクや，OBC 間のリンクは信頼性が高いと考え，正常確率を高く設定したり，新規実装項目に関しては信頼性が低いと考え，低い正常確率を設定したりするなどが挙げられる．

2.4.3 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，地上試験では衛星全体ではなく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多数行う．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低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，運用中は上述し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元に，リスクと効率のトレードオフを考えながらコマンドの選択を行う必要がある．

第 3 章

本手法による不具合分析の実践と評価

3.1 概要

不具合分析の具体的な流れをみるために，いくつかの事例を取り上げて実践した結果を示す．以下では，まず実際に故障箇所特定が行えた事例，特定できなかった事例を取り上げ，そこから得られた知見に関して述べる．次に，評価指標に関する考察を行うために，優先する指標を変えることによって故障特定のプロセスに変化が現れるのかどうかを検証した結果に関して述べる．最後に，本手法によって扱うことのできる故障の種類に関する限界を述べ，発展させるために必要な方針に関して議論を行う．

3.2 実践例

3.2.1 ヒータ接触不良の特定

まず，以下の図 3.1 のような故障を考え，不具合分析を行っていく．

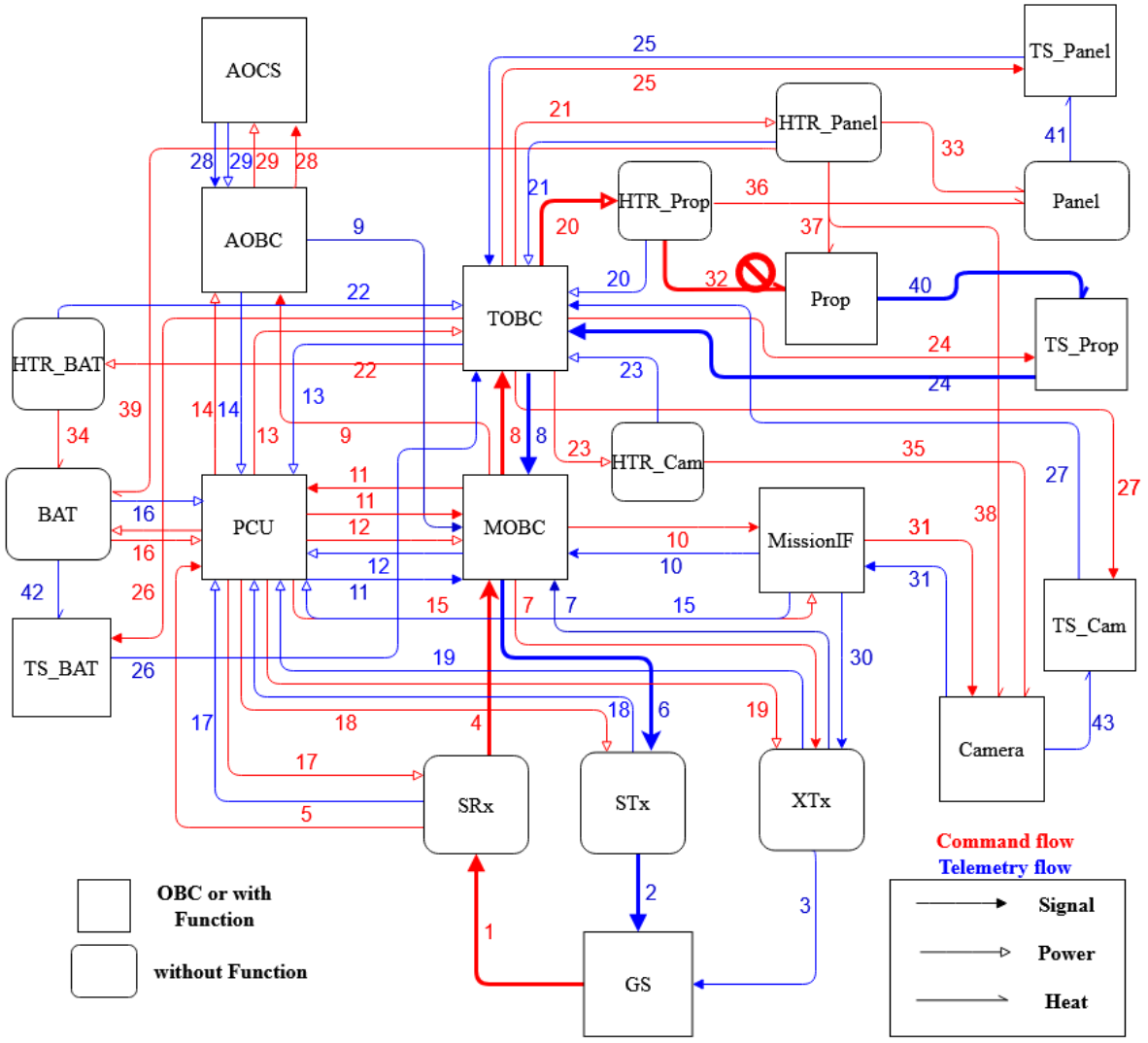

図 3.1 故障箇所：リンク 32(推進系ヒータ－推進系間) の時の故障候補

図 3.1 に示すような推進系ヒータ - 推進系間でのヒータ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，

- 推進系ヒータ ON コマンド (ID:14) を送信したのに，推進系温度 (ID:17) 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，図 3.2 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 (OK) か否 (NG) かを入力させることで，切り分けを行っている．今回の例では MOBC 及び TOBC は正常に動作しているはずなので，MOBC カウンタ及び TOBC カウンタは正常 (OK) と入力し，それを元に状態の更新を行っている．

```
targetTEL: [40, 24, 8.0, 6.0, 2.0]
targetCOM: [1, 4, 8, 20, 32]
TELtarget: [40, 24, 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
TELlink: [6, 2] were verified
TELtarget: [40, 24, 8.0]
Telemetry 2 ( TOBC_Counter ) can verify following links
 [8]

Please check TOBC_Counter
Input result(OK or NG)>>OK
TELlink: [8] were verified
```

図 3.2 テレメトリによる確認

次に，図 3.3 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 (推進系ヒータ ON) のみを考えている．確認可能性の高い経路を形成するテレメトリから順に表示され，人間に確認をさせているのが分かる．ここでの確認テレメトリに関しても，今回の例では正常であるため，そのように入力し状態の更新を行っている．ここに検証しているリンクがどれに当たるのかに関して可視化できるようにしたい

```
 Check telemetries which influenced by initial Command state

COMtarget: [1, 4, 8, 20, 32] TELtarget: [40, 24]
Command 14 ( HTR_PROP_ON ) & Telemetry 5 ( MOBC_COM_Counter ) can verify following links
 COMlink: [4, 1] TELlink []
Command 14 ( HTR_PROP_ON ) & Telemetry 6 ( TOBC_COM_Counter ) can verify following links
 COMlink: [8, 4, 1] TELlink []
Command 14 ( HTR_PROP_ON ) & Telemetry 10 ( TOBC_Current ) can verify following links
 COMlink: [8, 4, 1] TELlink []
Command 14 ( HTR_PROP_ON ) & Telemetry 16 ( PANEL_Temp ) can verify following links
 COMlink: [20, 8, 4, 1] TELlink []
Command 14 ( HTR_PROP_ON ) & Telemetry 21 ( HTR_PROP_Current ) can verify following links
 COMlink: [20, 8, 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: [8] & TELlink: [] were verified

Please check PANEL_Temp
Input result(OK or NG)>>OK
COMlink: [20] & TELlink: [] were verified
COMtarget: [32] TELtarget: [40, 24]
```

図 3.3 初期コマンドを用いた確認

最後に，以下の図 3.4 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．残った故障候補は，

- コマンドリンク 32:推進系ヒータ - 推進系間
- テレメトリリンク 40:推進系 - 推進系温度計間

- テレメトリリンク 24:推進系温度計 - TOBC 間

である．この時，探索結果として表示されたのはコマンド 13(パネルヒータ ON) と 18(推進系ヒータ OFF) であり，これらのコマンドに関する指標が図 3.4 のように示されている．
図中において Check link number が「確認可能リンク数」，Mean Probability of check が「平均確認可能性」，total_COM_number が「検証コマンド総数」を表している．またコマンドの衛星生存性への副作用を示す指標に関しては，impact TEL num が「コマンドによって影響を受けるテレメトリの数」，Remaining Power と Power consume by this COM が「コマンド送信前のバッテリ残量とコマンド送信による消費電力」，Attitude が「姿勢変化を起こすか否か」を示している．Attitude に関しては，姿勢変化を起こす場合は”Change”，起こさない場合は”Keep”と表示するようにしている．
今回の例では，推進系温度計には故障はないはずなので，パネルヒータによって推進系に伝わった熱を読み取りテレメトリに温度上昇として現れることになる．最終的に残ったリンクが 32 となり，想定した故障箇所の特定が行えた．

```
COMtarget: [32] TELtarget: [40, 24]
COM 13 HTR_PANEL_ON
        {'candidate link number': 2, 'Check link number': 1.0, 'Mean Probability of check': 0.5, 'total_COM_number': 2.0}
        {'impact TEL num': 8, 'Remaining Power': 3.8, 'Power consume by this COM': 2, 'Attitude': 'Keep'}
COM 18 HTR_PROP_OFF
        {'candidate link number': 3, 'Check link number': 0.75, 'Mean Probability of check': 0.25, 'total_COM_number': 1.875}
        {'impact TEL num': 6, 'Remaining Power': 3.8, 'Power consume by this COM': -1, 'Attitude': 'Keep'}
Please select Command above(input ID) >>13
Command 13 ( HTR_PANEL_ON ) & Telemetry 17 ( PROP_Temp ) can verify following links
 COMlink: [] TELlink [40, 24]

Please check PROP_Temp
Input result(OK or NG)>>OK
COMlink: [] & TELlink: [40, 24] were verified
selected Command: [14, 13] remaining Command: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 2
5, 26, 27, 28, 29, 30]
COMtarget: [32] TELtarget: []
COM 18 HTR_PROP_OFF
        {'candidate link number': 1, 'Check link number': 1, 'Mean Probability of check': 1.0, 'total_COM_number': 1.0}
        {'impact TEL num': 6, 'Remaining Power': 3.8, 'Power consume by this COM': -1, 'Attitude': 'Keep'}
Please select Command above(input ID) >>18
Command 18 ( HTR_PROP_OFF ) & Telemetry 17 ( PROP_Temp ) can verify following links
 COMlink: [32] TELlink []

Please check PROP_Temp
Input result(OK or NG)>>NG
selected Command: [14, 13, 18] remaining Command: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 15, 16, 17, 19, 20, 21, 22, 23, 24, 2
5, 26, 27, 28, 29, 30]
COMtarget: [32] TELtarget: []
nothing can verify
finish
faulty COMlink: [32] faulty TELlink: []
```

図 3.4　コマンドの選択肢表示及び検証過程

3.2.2 温度計故障に関する検証

次に，以下の図 3.5 に示すような温度計故障（断線）を考え検証を行った例に関して述べる．この時，異常検知の際の不具合事象としては，上の事例と同じく

- 推進系ヒータ ON コマンド (ID:14) を送信したのに，推進系温度 (ID:17) が上昇しない

という事象である．テレメトリの確認や，初期コマンド状態からの確認情報の提示の流れは先ほどの例と同様となる．

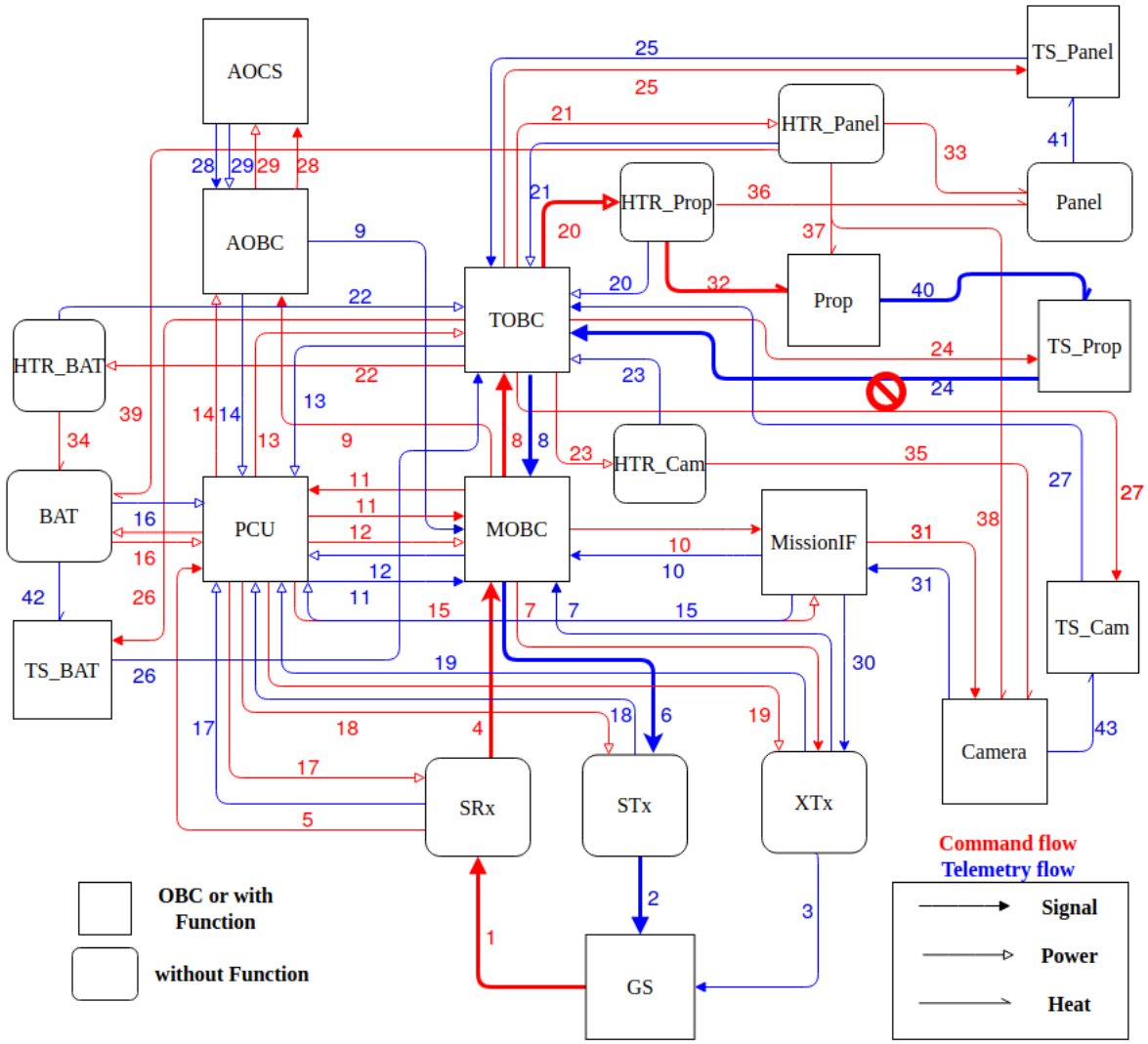

図 3.5 故障箇所：リンク 24(推進系温度計-TOBC 間) の時の故障候補

この時，システムによって洗い出された検証用のコマンドは上の例 (図 3.4) のものと同じであり，指標から判断するとコマンド 13 の方が切り分けられるリンクの数は大きい一方で全体のコマンド数の見積もりでは，コマンド 18 の方が少ないコマンド数であるという結果になっている．コマンド 13 及び 18 を

初めに選択した結果はそれぞれ以下の図 3.6，3.7 のようになった．今回の不具合は温度計の断線であるため，パネルヒータによる推進系温度の変化も推進系ヒータによる推進系温度変化も見ることはできないため，どちらも入力は異常 (NG) を与えている．最終的な結果は，どちらの過程を経ても同じになっており，一つのリンクにまで故障箇所の特定を行うことができなかった．このことから，この衛星モデルが今回扱った故障「(推進系) 温度計故障」が発生した際に故障特定を行える設計になっていないことが分かる．このように，本手法は単に故障箇所の特定を支援するだけでなく，設計の不備を洗い出すことにも利用できる．

```
Please select Command above(input ID) >>13
Command 13 ( HTR_PANEL_ON ) & Telemetry 17 ( PROP_Temp ) can verify following links
 COMlink: [] TELlink [40, 24]

Please check PROP_Temp
Input result(OK or NG)>>NG
selected Command: [14, 13] remaining Command: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 2
5, 26, 27, 28, 29, 30]
COMtarget: [32] TELtarget: [40, 24]
COM 18 HTR_PROP_OFF
        {'candidate link number': 3, 'Check link number': 0.75, 'Mean Probability of check': 0.25, 'total_COM_number': 1.0}
        {'impact TEL num': 6, 'Remaining Power': 3.8, 'Power consume by this COM': -1, 'Attitude': 'Keep'}
Please select Command above(input ID) >>13
Please select Command above(input ID) >>18
Command 18 ( HTR_PROP_OFF ) & Telemetry 17 ( PROP_Temp ) can verify following links
 COMlink: [32] TELlink [40, 24]

Please check PROP_Temp
Input result(OK or NG)>>NG
selected Command: [14, 13, 18] remaining Command: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 15, 16, 17, 19, 20, 21, 22, 23, 24, 2
5, 26, 27, 28, 29, 30]
COMtarget: [32] TELtarget: [40, 24]
nothing can verify
finish
faulty COMlink: [32] faulty TELlink: [40, 24]
```

図 3.6　コマンド 13 から選択した時の検証結果

```
Please select Command above(input ID) >>18
Command 18 ( HTR_PROP_OFF ) & Telemetry 17 ( PROP_Temp ) can verify following links
 COMlink: [32] TELlink [40, 24]

Please check PROP_Temp
Input result(OK or NG)>>NG
selected Command: [14, 18] remaining Command: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 16, 17, 19, 20, 21, 22, 23, 24, 2
5, 26, 27, 28, 29, 30]
COMtarget: [32] TELtarget: [40, 24]
COM 13 HTR_PANEL_ON
        {'candidate link number': 2, 'Check link number': 1.0, 'Mean Probability of check': 0.5, 'total_COM_number': 1.0}
        {'impact TEL num': 8, 'Remaining Power': 3.8, 'Power consume by this COM': 2, 'Attitude': 'Keep'}
Please select Command above(input ID) >>13
Command 13 ( HTR_PANEL_ON ) & Telemetry 17 ( PROP_Temp ) can verify following links
 COMlink: [] TELlink [40, 24]

Please check PROP_Temp
Input result(OK or NG)>>NG
selected Command: [14, 18, 13] remaining Command: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 15, 16, 17, 19, 20, 21, 22, 23, 24, 2
5, 26, 27, 28, 29, 30]
COMtarget: [32] TELtarget: [40, 24]
nothing can verify
finish
faulty COMlink: [32] faulty TELlink: [40, 24]
```

図 3.7　コマンド 18 から選択した時の検証結果

本研究で構築したシステム上では故障箇所の特定までを行うことができなかったが，不具合分析の過程で得た情報を用いて人間側で故障箇所の類推をすることは可能である．今回の事例では，コマンドリンク 20 が正常であることは「パネル温度」によって確認できており，パネル温度の上昇を確認することでヒータの正常が動作していることも確認できるため，推進系ヒータの故障ではないことが分かる．また，「パネルヒータ ON」によって推進系温度の変化を見ることができなかったことから，テレメトリリンク：40,24 が異常であることも類推できる．このように，本システムに従って検証を行うことによって，故障箇所の推論するために必要な情報が取得可能であることがわかる．

実ミッションでは，設計段階において FMEA(Failure Mode and Effect Analysis) などを用いて，衛星システムに起こりうる故障モードを列挙し，それらの故障モードによる影響や，発見のしやすさなどをもとに設計へのフィードバックを行う．よって，FMEA 上で洗い出された故障モードに対して本手法を適用することによって，それぞれの故障モードが発見可能な設計になっているかを確認することができる．

3.2.3　評価指標に関する考察

以下では，2 章で示したコマンドの評価指標に関して，複数のコマンド候補があった際に優先するコマンドの評価指標によって検証結果が変化するかどうかを検証し，評価指標の定性的な意味を考察した．
以下の図 3.8 のような複数の故障がある場合を考える．なお，本手法では 1 つの故障しか正確に見ることができない手法になっているが，今回は評価指標による違いを見るためにこのような問題設定を与えた．

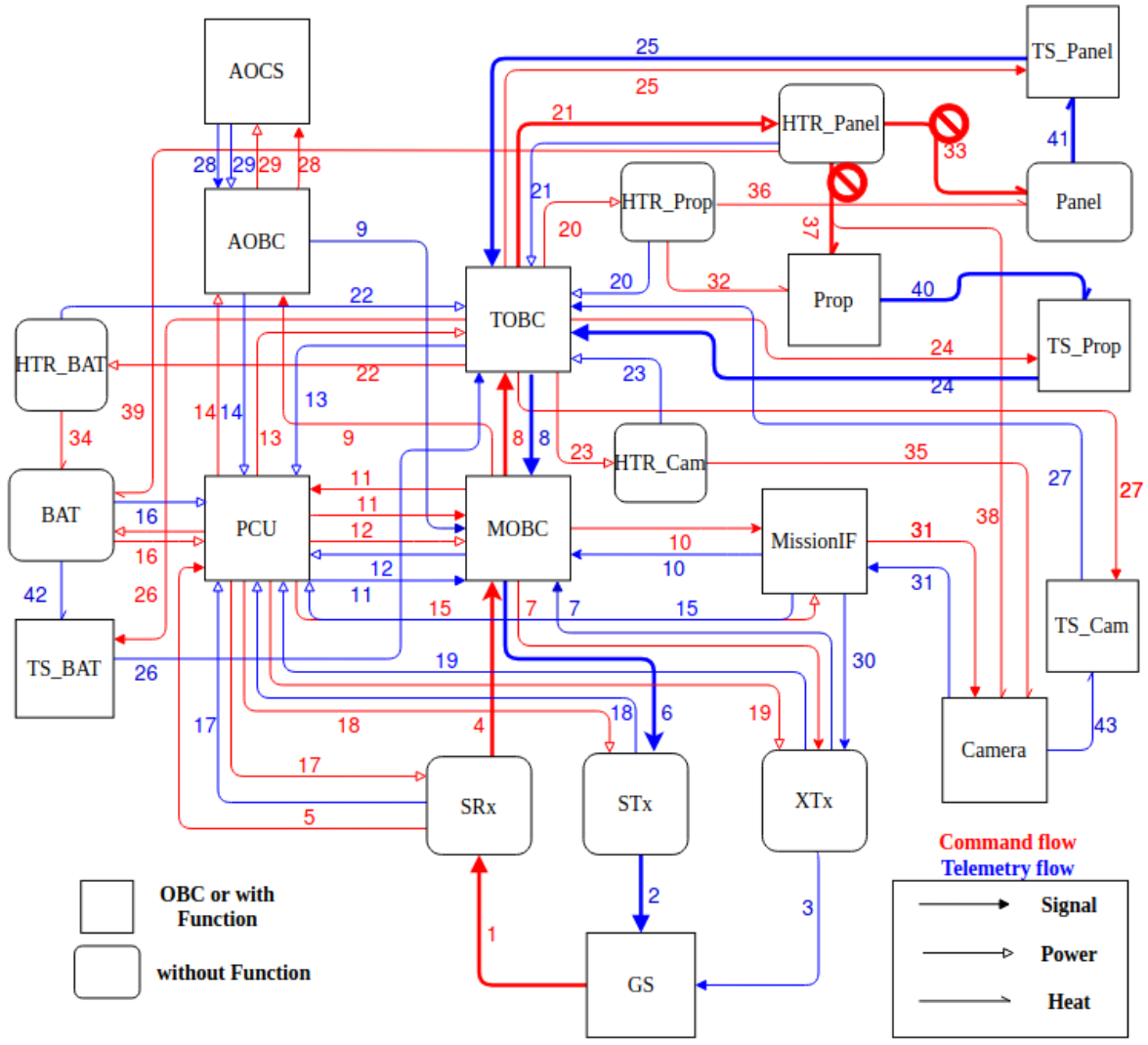

図 3.8 故障箇所：リンク 33(パネルヒータ－パネル間)，リンク 37(パネルヒータ－推進系間) の時の故障候補

まず，テレメトリによる確認事項の提示と，初期コマンドによって確認できる項目の提示に関する検証プロセスを以下の図 3.9，3.10 に示す．

```
targetTEL: [40, 24, 8.0, 6.0, 2.0, 41, 25]
targetCOM: [33, 1, 4, 8, 21, 37]
TELtarget: [40, 24, 8.0, 6.0, 2.0, 41, 25]
Telemetry 1 ( MOBC_Counter ) can verify following links
 [6, 2]

Please check MOBC_Counter
Input result(OK or NG)>>OK
TELlink: [6, 2] were verified
TELtarget: [40, 24, 8.0, 41, 25]
Telemetry 2 ( TOBC_Counter ) can verify following links
 [8]

Please check TOBC_Counter
Input result(OK or NG)>>OK
TELlink: [8] were verified
```

図 3.9 取得テレメトリによる確認事項の提示及び検証結果

```
 Check telemetries which influenced by initial Command state

COMtarget: [33, 1, 4, 8, 21, 37] TELtarget: [40, 24, 41, 25]
Command 13 ( HTR_PANEL_ON ) & Telemetry 5 ( MOBC_COM_Counter ) can verify following links
 COMlink: [4, 1] TELlink []
Command 13 ( HTR_PANEL_ON ) & Telemetry 6 ( TOBC_COM_Counter ) can verify following links
 COMlink: [8, 4, 1] TELlink []
Command 13 ( HTR_PANEL_ON ) & Telemetry 10 ( TOBC_Current ) can verify following links
 COMlink: [8, 4, 1] TELlink []
Command 13 ( HTR_PANEL_ON ) & Telemetry 18 ( CAM_Temp ) can verify following links
 COMlink: [21, 8, 4, 1] TELlink []
Command 13 ( HTR_PANEL_ON ) & Telemetry 19 ( BAT_Temp ) can verify following links
 COMlink: [21, 8, 4, 1] TELlink []
Command 13 ( HTR_PANEL_ON ) & Telemetry 20 ( HTR_PANEL_Current ) can verify following links
 COMlink: [21, 8, 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: [8] & TELlink: [] were verified

Please check CAM_Temp
Input result(OK or NG)>>OK
COMlink: [21] & TELlink: [] were verified
COMtarget: [33, 37] TELtarget: [40, 24, 41, 25]
```

図 3.10 初期コマンドを用いた確認

この結果残る故障候補として以下のようになり

-

その結果以下のようなコマンド探索結果が提示される．評価指標を比較すると，平均確認確率は検証コマンド総数はとなる．以下では，この 2 つの指標に関して，優先する指標によってどのように検証プロセスが変化するかを示す．

複数故障を考えた場合，指標の選択基準によって探索プロセスが変化する．
平均確認可能性が高いコマンドから送ると，一回のコマンドで多くの絞り込みが行えるので，時間制

約に不確定性があり，コマンドを 1 度送ることができるかどうかという状況であればワンチャンスをものにできるので良い．

一方で検証コマンド総数を指標に選んだ場合，少ないコマンド数で絞り込みを行えるが，あくまで最後まで検証を行うことを元にコマンドの数を計算しているため，結果によっては最悪値の数のコマンドを送信する必要があり，時間制約の厳しいときには十分に絞り込みを行えないまま検証作業を終えることになる．

恐らく OBC 系列が死んだ時，かつそれに関するコマンドが複数ある場合に有効になりそう．

第 4 章

結論

4.1　本研究で得られた知見

本研究ではモデルを用いてコマンドによる故障箇所特定のプロセスを体系化する手法を提案し，テストケースを用いてその有効性を検証した．

本手法を用いて最終的な故障箇所の特定を行うのは難しい．どちらかというと不具合分析過程を体系化して，それを用いたコマンドの選択をすることによって故障箇所の推論に必要な情報を集めるような働きをしていると言える．

テレメトリを発行している機器の故障の場合は，そのコンポーネントからの情報ラインに冗長系がなければかなり多くの故障候補が残ってしまう．

4.2　今後の展望

今後，接続関係の異常だけでなく，実問題に近い故障状態も扱えるようにするために，扱う状態量をより詳細にモデル化していく必要がある．また，テレメトリと状態量の対応付けを考えることによって異常状態をリンクとして表現するのではなく，各コンポーネントの機能の異常を特定できると考えている．

また本手法では，簡易的に故障候補の洗い出しを情報伝達の経路のみに絞っていたが，）で提案されているオントロジーを用いることで，細かい粒度の特定を目標に行っていく必要がある．

今回の例では，モデルの構築を手作業により行ったが，実ミッションでの適用を考慮すると手作業によるモデル構築は現実的ではない．今後ある程度の事前定義情報からモデル化を自動化することを目標にする．

リンクの正常確率として，より実機で用いているコンポーネントの信頼度に近い値を考えることで，モデルが複雑になった際により効率的な故障箇所特定を行えると考えている．

人間からのフィードバックの情報として，正常か異常かの 2 値しか与えることができていない．実際には，テレメトリの種類によって正常か異常かの基準はいくつかあり，

- テレメトリが下りてきているか否か
- パラメータの値が大きいのか小さいのか

-

などが挙げられる．テレメトリの種別という概念を導入し，人間による入力のパターンを増やすことで，故障の種類を見分けることができるようにする必要がある．

4.3 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．

また，これらの信頼性を試験結果から学習させることによって，対象とする衛星に対するモデルの再限度を高め，効率的な不具合分析を行うことが可能になる．

参考文献

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.
9) Yoshinobu Kitamura and Riichiro Mizoguchi. *A Framework for Systematization of Functional Knowledge based on Ontological Engineering.* PhD thesis.
10) JAXA. 衛星の機能モデル (Functional Model of Spacecrafts (FMS)). Technical report, 2020.