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

東京大学 工学部 航空宇宙工学科

03-183005　西本 慎吾

令和 2 年 11 月 30 日 提出

指導教員　船瀬 龍 准教授

概要

近年，大学や高専などの教育機関や，民間企業による超小型衛星の開発，およびそれを利用した事業の展開が盛んになっている一方で，超小型衛星の信頼性の低さが問題となっている．軌道上故障に関する調査の結果信頼性の低さの原因として，設計および製造過程における不良が多いことが分かっており，地上試験によって不具合の改修，対策を十分に行うことが超小型衛星の信頼性向上のために重要である．一方で，衛星のように多くの機器が複雑に絡みあったシステムでは，一つの不具合事象に対して非常に多くの故障が考えられ，不具合事象から故障箇所の特定を行うことは非常に多くの知識と経験を必要とする．そのため，衛星開発を専門としない機関などの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不具合事象から網羅的に故障候補を洗い出し，故障箇所の特定を行うことは困難である．これに対して，機器や故障状態をオントロジーなどを用いてモデル化し，不具合事象から故障仮説を網羅的に洗い出すための研究が広く行われている．一方で，故障仮説の検証段階に対して取り組んだ研究は少なく，検証過程が人の知識や経験に依存してしまっている．

そこで，本研究ではコンポーネント間の接続関係モデル，衛星内の情報伝達の経路モデルを用いて衛星の故障候補の検証手順（打つべきコマンド，確認事項）を探索し，それらを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．

本手法を用いて，簡易的な衛星モデルに対して不具合分析を実践することで，コマンドによる故障箇所の特定作業が体系化できること，設計不備の発見につながることを確認した．

目次

図目次

表目次

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 [3]．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 [3]．
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 [1]．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 [1]．

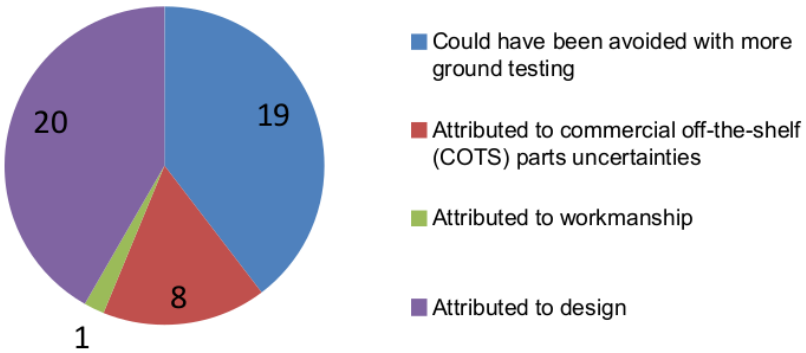

図 1.1　故障原因に関する調査結果 [1]

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 [4)]．
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 [4)]．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 [2)](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 [2)]．

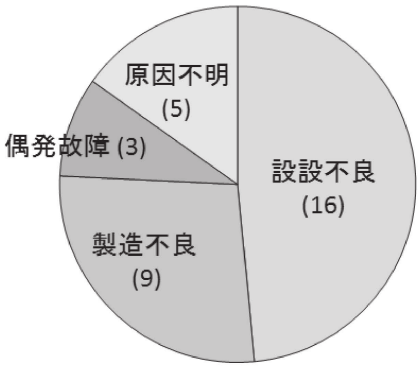

図 1.2　軌道上故障の原因類型の分布 [2)]

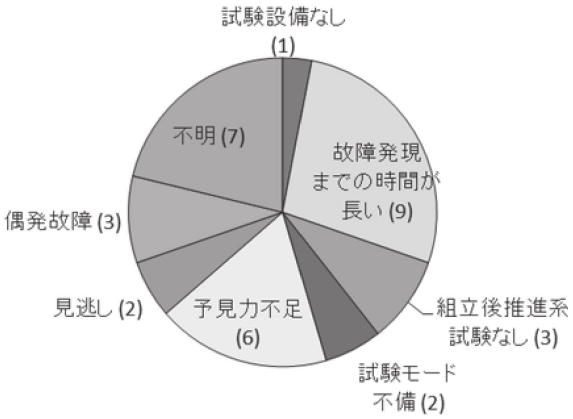

図 1.3　軌道上故障の要因を地上で発見できなかった原因類型の分布 [2)]

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に関して一例を示す．

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容易ではない．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機器間の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，事前に人によって洗い出された故障モードは，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，網羅性にかけていたという研究結果もある．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．不具合発生時は衛星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の安全性も非常に重要な点である．

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，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

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．上述したように，不具合分析は主に故障仮説の生成と故障仮説の検証作業から構成されるため，検証過程に関しても取り組む必要性は十分にある．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十分に行われていない．

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．

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，コマンドの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．また，第 4 章では本論文での結論として，まとめとして研究目的に対する本研究の成果を述べ，今後の課題を示している．

第 2 章

情報伝達経路モデルに基づく不具合分析支援手法の仕様

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．
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
- 故障仮説検証の流れ及び検証用コマンド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

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．

以下の図 2.1 に，本手法を用いた不具合分析の全体構成を示す．本手法では，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」の探索を行う．探索結果に関して後ほど示すコマンドの安全性及び故障候補切り分けの能力を表す指標を求め，それと共に人間に対して提示する．その中から人間が選択し，実際に検証を行った結果を入力させ，故障箇所の特定ができるまでそれを繰り返し行う．
また，本研究の主な目的は検証段階を支援することにあるため，異常検知の際のコマンド及びテ

レメトリが通る経路内に故障箇所が存在すると仮定しており，網羅的な故障仮説の生成は考慮しない．

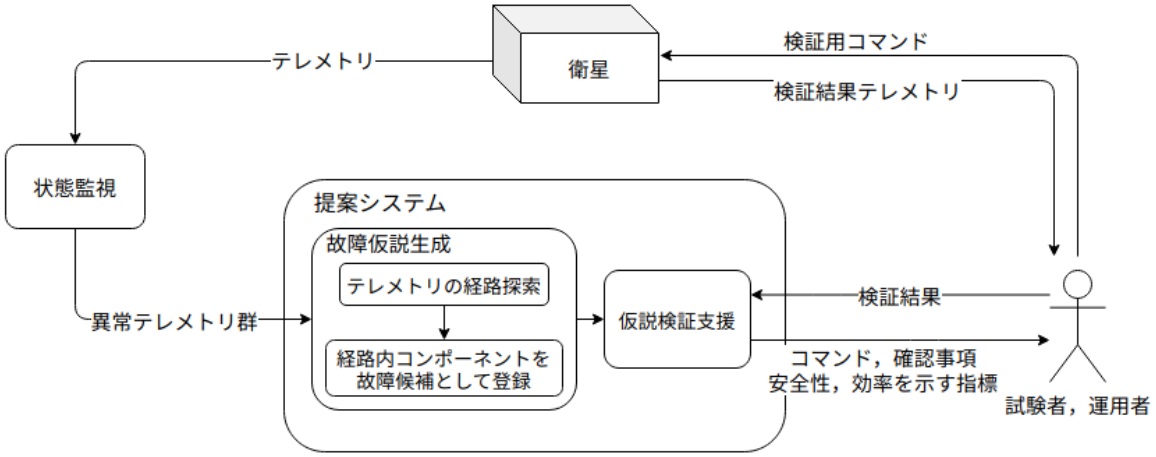

図 2.1　本手法による不具合分析の構成

以下に，上で示した不具合分析システムによる不具合分析の流れを簡単に示す．また，故障仮説の検証の流れ，故障候補を確認するためのテレメトリ・コマンド探索アルゴリズムの詳細に関しては後ほど言及する．

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が送信されてから，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（仮説検証支援）．
　a 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
　b 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．
　c 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．
　d 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．
　e 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．

2.2　モデル

次に，以上で述べた流れ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に基づいて述べる．以下で示す各モデルに関する図表は，テストケースで作成したものになっているが，モデルの内容自体はこれに限らず他の衛星モデルに対しても適用可能である．

2.2.1　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

表 2.1　簡易衛星モデル (図 2.2) における略語の意味

GS	:	地上局	MOBC	:	メイン OBC[*1]
PCU	:	電源操作コンポーネント	TOBC	:	熱系 OBC[*1]
AOBC	:	姿勢系 OBC[*1]	MIF	:	ミッション系 OBC[*1]
XTx	:	X バンド送信機	STx	:	S バンド送信機
SRx	:	S バンド受信機	HTR_PROP	:	推進系ヒータ
HTR_PANEL	:	パネルヒータ	HTR_BAT	:	バッテリヒータ
HTR_CAM	:	カメラヒータ	TS_PROP	:	推進系温度計
TS_PANEL	:	パネル温度計	TS_BAT	:	バッテリ温度計
TS_CAM	:	カメラ温度計	PROP	:	推進系コンポーネント
PANEL	:	衛星筐体パネル	BAT	:	バッテリ
CAM	:	カメラ	AOCS	:	姿勢制御コンポーネント

[*1] OBC : On Board Computer

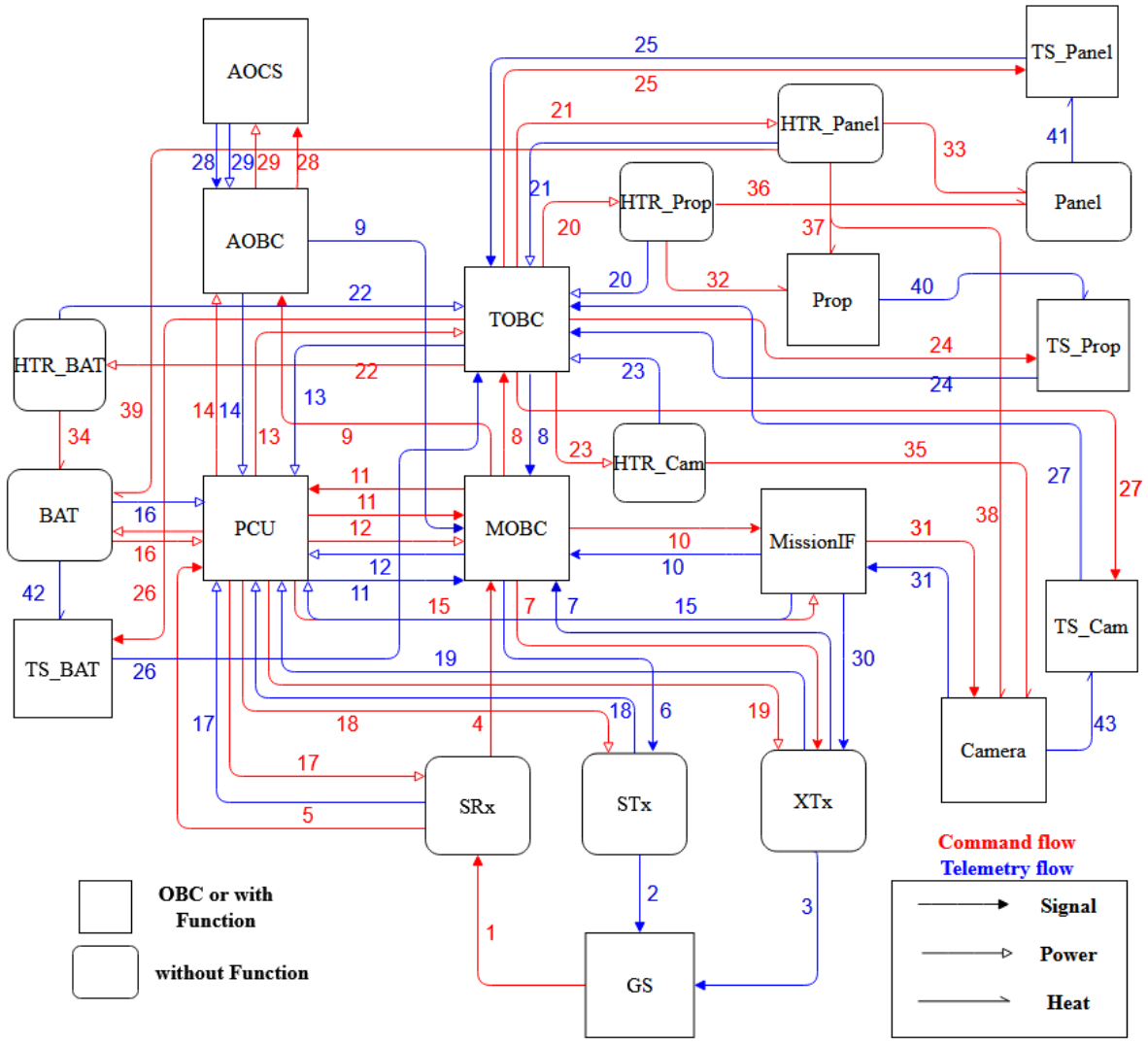

図 2.2　衛星内コンポーネントの接続関係図 (矢印の数字はリンクの ID)

2.2.2　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 2.2 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 (Medium)，そのリンクが正常に情報伝達を行う確率 (Probability) となっている．ID は各リンク固有の識別子として，リンクを参照する際に以下でも使用される．リンクが正常に情報伝達を行う確率（以下，正常確率）は後に示すコマンドの故障候補切り分け能力を示すために用いられ，本研究では簡単のため全てのリンクに関して 0.5 で統一している．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 2.2　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．以下の表 2.3 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，コマンドリンクはコマンドによる情報の伝達で使用されるリンクを意味し，テレメトリリンクはテレメトリによる情報の伝達で使用されるリンクを意味する．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おり，これによって情報の方向を決めている．
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．

表 2.3　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

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．簡単のため本研究では，扱う状態を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」として定義し，各コンポーネントが持つ機能が動作しているか否かという状態を持つ．この機能の動作状態に関しては後ほど言及する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構成となっている．


```
"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

2.2.3　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表 2.4，2.5 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 (TransitionTrigger)，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．時間によって変化するテレメトリは，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できる．そのため，不具合分析の初めのアプローチに利用可能である．

表 2.4　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	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(impact_TEL_ID)，コマンドの種別 (type)，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 2.4 に示すテレメトリの経路及び，表 2.5 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[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．コマンドの機能情報としてはコマンドの対象を「target」とし，その中に対象コンポーネント (Component)，そ

のコンポーネントが持つ機能 (Function) を与えている．また，各コマンドが持つ「Active」という属性は，そのコマンドによってコンポーネントが持つ機能が動作状態になる，もしくは電源状態が ON になるのであれば「true」，その逆なら「false」としている．例えば，電源 ON コマンドは「Active:true」であるが，電源 OFF コマンドは「Active:false」となる．後ほど言及するが，故障仮説の検証において，対象の状態を変化させるコマンドでないと故障候補の確認はできないとしている．そのため，そのコマンドがどのような状態変化を起こすコマンドなのかという情報を以上で述べたように定義した．

表 2.5　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	5,6,10,16,17,18,19,20	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	

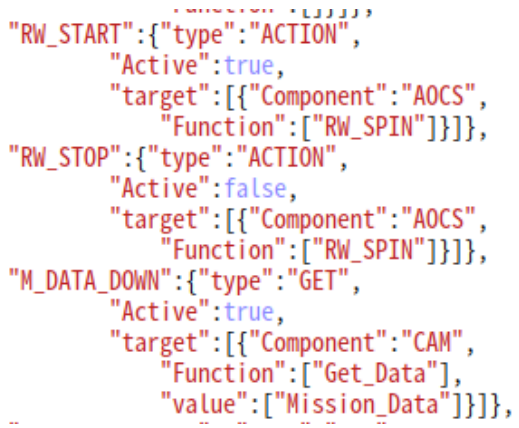

図 2.4　コマンドの機能モデル

2.3　本手法による故障仮説検証の流れ

2.3.1　故障仮説の検証アルゴリズム

次に，本手法による検証作業の流れに関して述べる．以下の図 2.5 に示すのが，本手法による故障仮説検証の流れである．
不具合が発生している状態は内部状態に関する不確定性が多い．そのため，予期せぬ二次故障を起こさないために，衛星の状態を変えずに確認できる箇所を優先的に確認することが望ましい．よって，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，そのテレメトリを用いた切り分けを行う．
その後，衛星の状態を変化させること無く故障箇所特定のために得られる情報がなくなれば，次のステップとしてコマンド送信によ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，衛星システムが持つコマンドの中から故障候補の状態を確認できるものを探索し，後ほど述べるコマンドの評価指標と共に提示する．上述したように，実行するコマンドの選択及び，検証結果の確認に関しては人間が行い，その結果を本システムにフィードバックすることで切り分けを行っていく構成になっている．

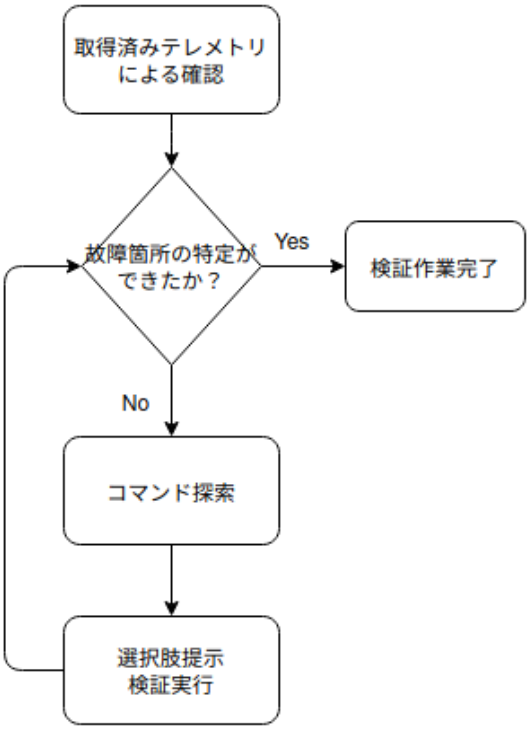

図 2.5　故障仮説検証の流れ

2.3.2　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

次に，上で示した故障仮説検証の流れにおいて，故障候補の切り分けを行うためのコマンドの探索アルゴリズムを以下の図 2.6 に示す．図に示すように，衛星システムが持つコマンドに対して確認するためのコマンドが無くなる，もしくは故障候補がなくなるまで探索を行う．本来，故障候補の確認が行えるかどうかを判断するためにはそのコマンドによって発生する衛星の状態変化を考え，その状態変化による波及効果からテレメトリの変化を洗い出す必要がある．今回は簡単のため，以上で示したように，各コマンドに対して影響を受けるテレメトリを事前に定義しておき，コマンドとそのコマンドによって影響を受けるテレメトリによって形成される経路内に故障候補があれば，その故障候補のリンクの状態を確認できる可能性があるとしている．
このとき，コマンド送信によって衛星内部に状態変化が発生しないのであれば得られる情報はないため，故障候補を通る経路であっても，そのコマンドが状態を変化させない場合は故障候補の確認ができないと仮定している．
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．

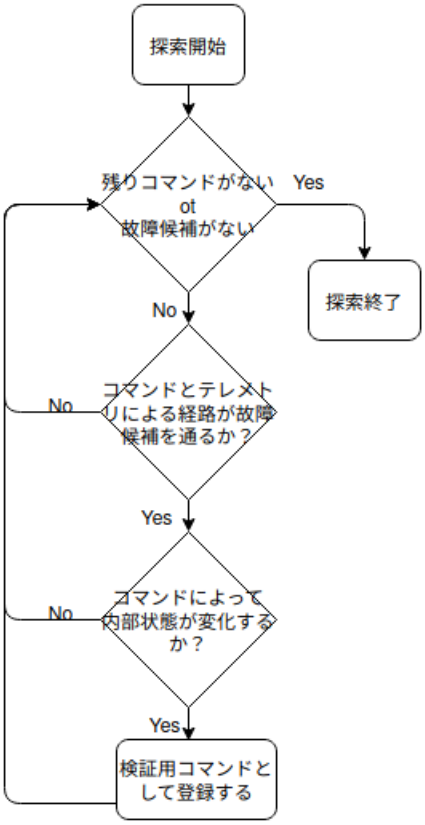

図 2.6　検証用コマンド探索アルゴリズム

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．

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．
以上を踏まえると，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，意図しない姿勢変化を起こし内部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，実在システムにおいて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箇所の特定のために用いることはリスクの大きな動作であると言える．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「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．つまり，状態変化を起こすコマンドによる影響を受けるテレメトリの数がこれにあたる．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ような，時間制約が特に厳しい中での不具合分析が考えられる．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．
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．

2.4.2.1　1 つのコマンドで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，重要な点の一つとして一度に確認できる故障候補の数が多いことが挙げられる．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 2.7 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．

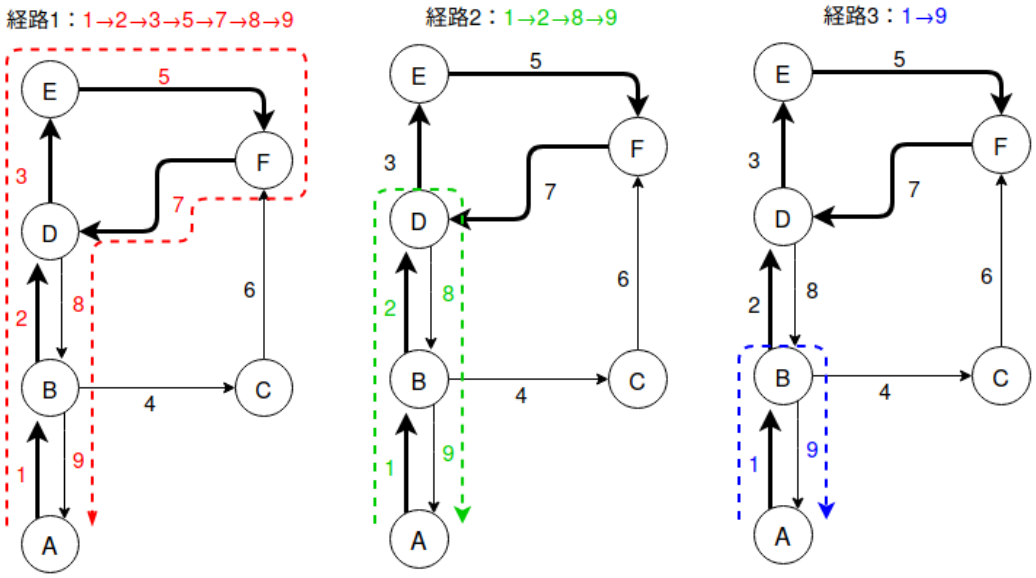

図 2.7　故障候補とそれを確認するための情報伝達経路の例

図 2.7 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

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

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

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

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

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

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

$$= 1 \tag{2.7}$$

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 2.7 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．

2.4.2.2　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でき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 2.8 では，上で示した例（図 2.7）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 2.7 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．

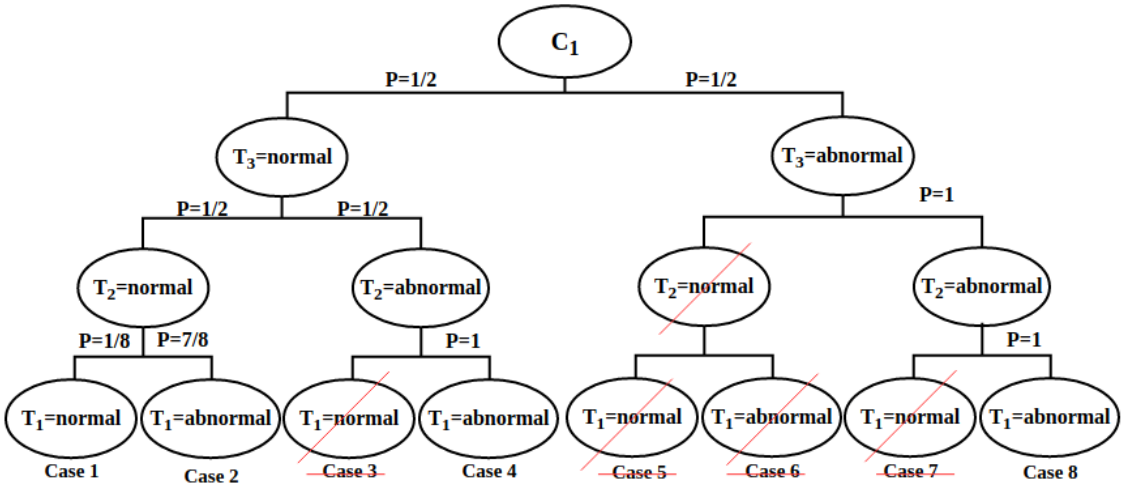

図 2.8　各テレメトリの結果による検証過程の種類

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 2.7 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 2.8 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 2.8 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 2.8 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 2.9 のようになる．問題設定として，図 2.7 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 2.9 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．

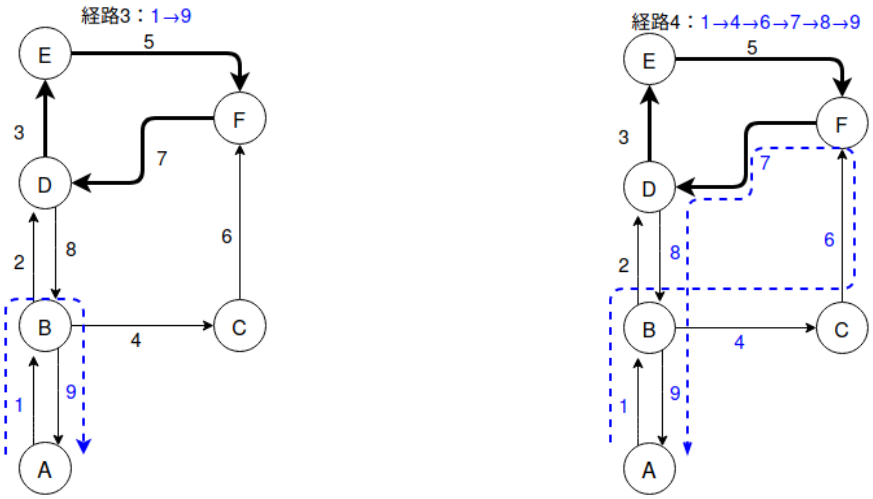

図 2.9　Case 2 の場合に残る故障候補とコマンド 2 に影響を受けるテレメトリ (3,4) が成す経路

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 2.10 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．

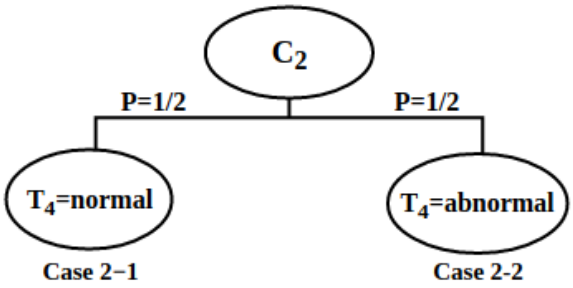

図 2.10　Case 2 の時のコマンド 2 送信時の結果

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

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．先にモデル上で定義した各リンクの正常確率を用いることで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．

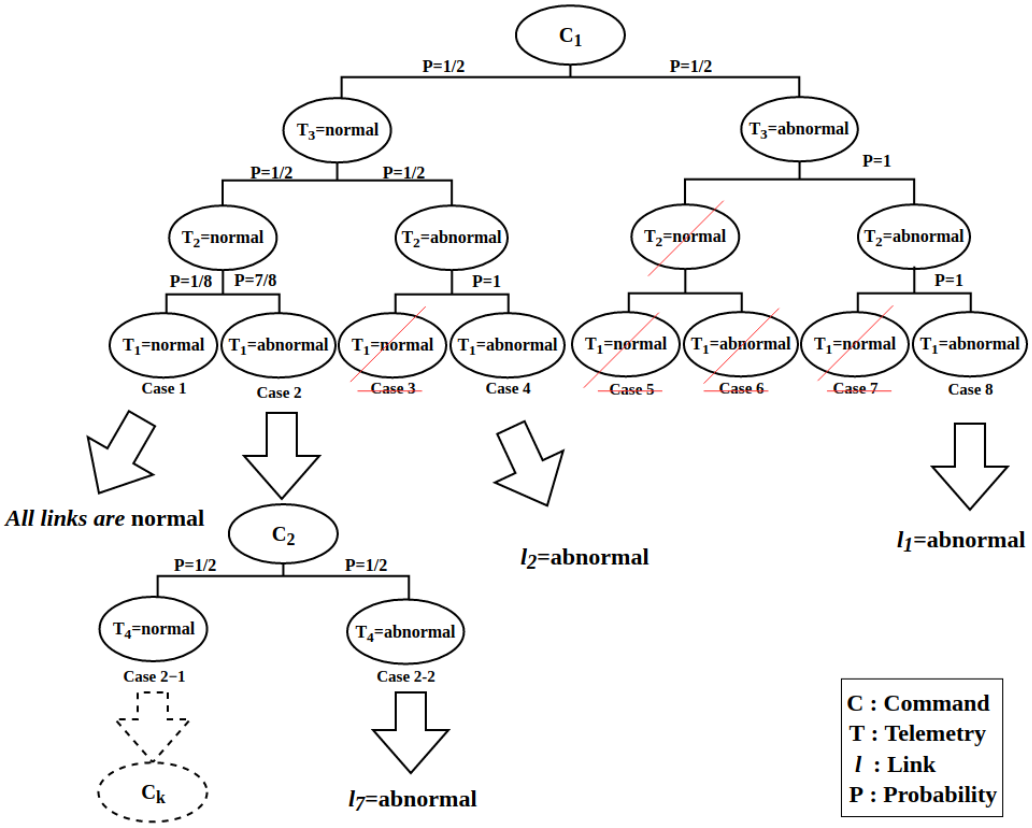

図 2.11　検証プロセスの全体像

図 2.11 に，今回説明のために使用した例における検証の全プロセスを示す．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 $\mathbb{C} = \{$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，より効率的に故障箇所の特定ができると考えられる．実際の情報を組み込む例としては，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．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「コマンド送信によって変化するテレメトリの数」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，コマンドによって引き起こされる状態変化の波及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，コマンドの故障候補切り分け能力を重視し，「平均確認可能性」及び「確認可能リンク数」が高く，「検証コマンド総数」が小さいコマンドを選択することで効率的な切り分けが行える．

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．そのような際には，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．一般に，故障候補を切り分ける能力が高いコマンドは，「コマンドを打つことで変化するテレメトリの数」も大きくなる．よって，運用時は上で示した全ての指標を元に，コマンドによって二次故障などが発生するリスクと，故障候補切り分けの効率のトレードオフを考慮し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

本性では，提案手法を用いた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を取り上げる．その結果に関して，コマンドを選択する際に優先する指標によって検証プロセスが異なることを述べ，評価指標に関する考察を行う．また，その例に対して所属する研究室の方々に実践頂いた結果と比較を行い，人間による不具合分析と本手法の違いに関して示す．
次に，本手法のみでは故障箇所の特定を上手く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．

3.2　実践例

3.2.1　故障箇所の特定ができた例（ヒータの接触不良）

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．

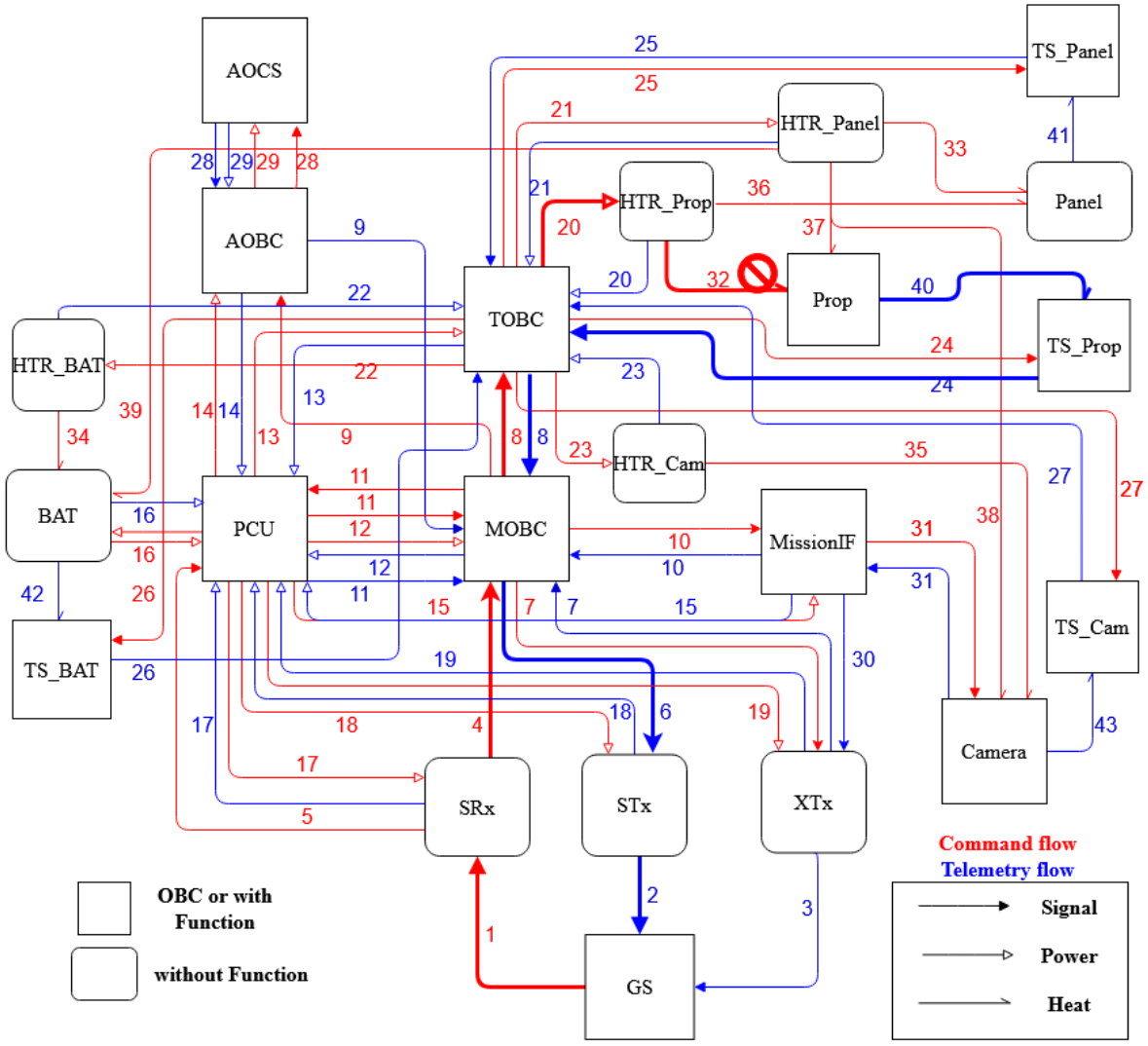

図 3.1　故障箇所：リンク 32(推進系ヒータ－推進系間) の時の故障候補

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．
ここで，本手法では前章で述べたように衛星内部コンポーネントの初期状態として，送信したコマンドによってもたらされる状態変化が起こっているものとして与えている．そのため，不具合事象の際に送っているコマンド「推進系ヒータ ON」によって，推進系ヒータは電源 ON 状態であると仮定し，以下の検証用コマンドの探索を行っていることを述べておく．

以下に，この事象に本手法を適用した例を示す．まず，図 3.2，3.3 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に関してコンソールへの出力結果と切り分けの流れを可視化したものを示す．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．

```
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

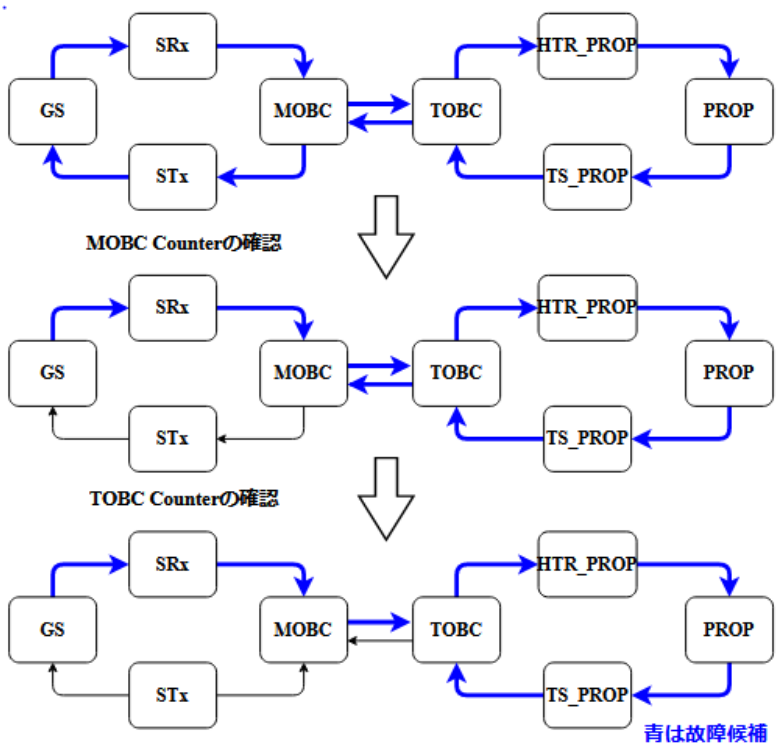

図 3.3　テレメトリの確認による故障候補切り分けの流れ

```
 Check telemetries which influenced by initial Command state

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 3.4　初期コマンドによる情報を用いた確認

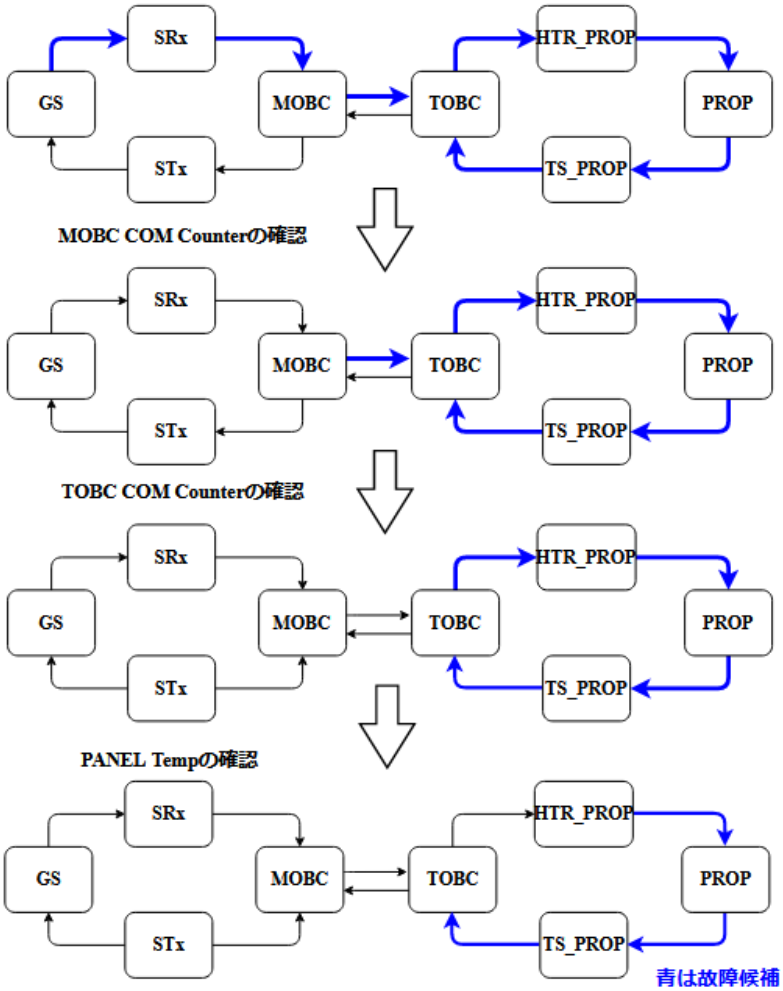

図 3.5　初期コマンドに影響を受けるテレメトリの確認による故障候補切り分けの流れ

次に，図 3.4 及び図 3.5 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．

最後に，以下の図 3.6 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 3.6 のように示されている．

図中において Pm(C) が「平均確認可能性」，E(C) が「確認可能リンク数」，N(C) 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，Remain Power が「コマンド送信前のバッテリ残量」，Power Consume 「コマンド送信による消費電力」，Impacted TEL num が「コマンドによって影響を受けるテレメトリの数」，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．

```
COM 13 HTR_PANEL_ON
        Pm(C): 0.5 , E(C): 1.0 , N(C): 2.0
        Remain Power: 3.8 , Power Consume: 2 , Impacted TEL num: 8 , Attitude: Keep
COM 18 HTR_PROP_OFF
        Pm(C): 0.25 , E(C): 0.75 , N(C): 1.875
        Remain Power: 3.8 , Power Consume: -1 , Impacted TEL num: 6 , Attitude: Keep
```

図 3.6　コマンドの探索結果の表示

以下の図 3.7，3.8 に，探索されたコマンドに関して各コマンドを選択し検証を行ったプロセスを示す．どちらのコマンドから開始しても最終的に，今回想定した故障箇所である「推進系ヒータ－推進系間 (リンク ID:32)」が故障箇所であると特定された．

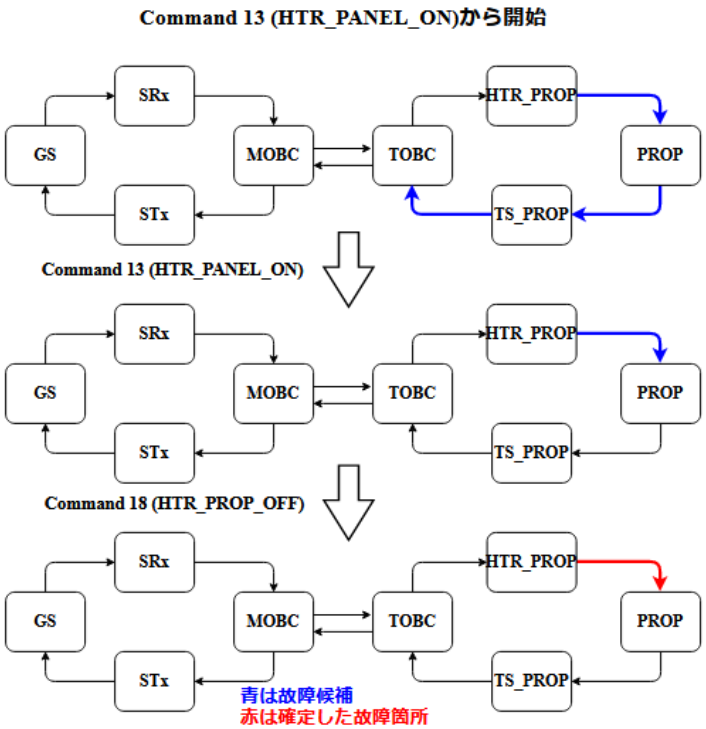

図 3.7　ヒータ接触不良時の検証プロセス（パネルヒータ ON からスタート）

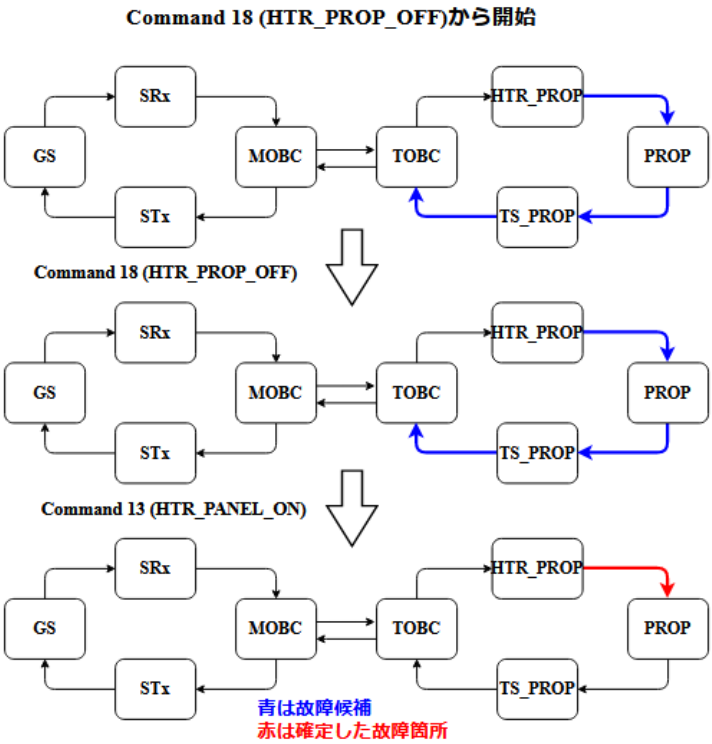

図 3.8　ヒータ接触不良時の検証プロセス（推進系ヒータ OFF からスタート）

3.2.2　コマンドの故障候補切り分け能力を示す指標に関する考察

上で示した例 (ヒータ接触不良) に関して，コマンドを選択する際に優先する評価指標によって図 3.7，3.8 のように検証プロセスの違いが生じた．この結果の違いに基づき，提示された評価指標が持つ意味を考察する．

図 3.6 にあるように提示された 2 つのコマンドを比較すると，「平均確認可能性」はパネルヒータ ON コマンド (13:HTR_PANEL_ON) が高く，「検証コマンド総数」は推進系ヒータ OFF コマンド (18:HTR_PROP_OFF) が小さくなっている．パネルヒータ ON コマンドから検証を開始した場合，初めに「推進系－推進系温度計間 (テレメトリリンク 40)」，「推進系温度計－ TOBC 間 (テレメトリリンク 24)」の正常が確認できており，1 つのコマンドによる確認で故障候補が 1 つのリンクにまで絞り込めている．最終的に，推進系ヒータ OFF コマンドを送信した際に推進系温度の変化が見られなかったことから「推進系－推進系温度計間 (テレメトリリンク 32)」の異常が確認でき，故障箇所を特定している．

一方で，推進系ヒータ OFF コマンドから検証を開始した場合，1 つ目の検証では推進系温度に変化が見られず，状態変化を確認できないため，故障候補の切り分けをすることができない．一方で，故障候補の中に確実に故障箇所が存在することを確かめることができる．次に，2 つ目のコマンド「パネルヒータ ON」で推進系温度の上昇を見ることができるため，先程のプロセスと同様の切り分けができ，故障箇所の特定ができている．

運用時，通信が不安定であり不具合分析に使える時間が明確でない時には一度のコマンドで多くの確認ができることが望ましい．そのため，図 3.6 において「パネルヒータ ON」コマンドから送信する検証プロセスが良いと言える．これを元に考えると，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可能性が高いと言える．より厳しい時間制約の際には，最後まで検証作業を行うことができるという保証はない．そのため，平均確認可能性が高いコマンドを優先的に選択して各コマンドによって得られる切り分けの効果を大きくすることが望ましい．

一方で検証コマンド総数が小さなコマンドを優先的に選択した場合，少ないコマンド数で絞り込みを行える可能性があるが，この指標はあくまで，故障箇所を特定するまで検証を行うことを前提としてコマンドの数を計算している．そのため，故障状態によっては見積もられた数以上のコマンドを送信する必要があり，時間制約が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．このことを踏まえると，検証コマンド総数は不具合分析を最後まで行うことができる保証がある時に，優先的に考えることで，全体で打つコマンドの数を少なくできる可能性があると言える．

3.2.3　本手法と人間の不具合不具合分析の違い

上で示した不具合事象「推進系ヒータ ON コマンドを送った時，推進系温度が変化しない」に関して，想定した故障状態に関する情報を与えずに不具合分析を行う際の過程（送信するコマンド，コマンドを選択する理由，確認するテレメトリ）を不具合分析の経験が豊富な本研究室の先輩方に伺った．

与えた情報としては，以下に示す通りであり，コマンドやテレメトリが通る経路や，コマンドが影響を及ぼすテレメトリに関する情報は与えなかった．

- 衛星システムのコンポーネントの接続関係図
- システムを構成するコンポーネント及びその電源状態
- コマンド及びテレメトリの定義情報（何の機能を持つコマンドなのか，テレメトリに含まれる情報）

質問内容は以下であり，コマンドによる不具合分析を始める段階からの意思決定を対象にしている．

- 不具合分析を開始する際に初めに送信するコマンドとその理由
- 初回の検証で何のテレメトリ情報を確認するか
- 上で送信した結果が正常もしくは異常であった場合に次に選択するコマンド及びその理由
- 2 回目の検証で何のテレメトリ情報を確認するか

以下の図 3.9 に，初回の検証で選択するコマンドの調査結果を示す．本手法で検証用コマンドとして提案された 2 つのコマンド「推進系ヒータ OFF(HTR_PROP_OFF)」と「パネルヒータ

ON(HTR_PANEL_ON)」を選択している人が半数以上を占めていることから，今回の問題設定では本手法による検証用コマンドの探索は人間の推論に近いと言える．また，提示された選択肢以外のコマンドを選択する人が一定数いたが，これらを選択した理由を見ると，これらのコマンドによって回答者が確認しようとしている事項は既に確認しているとして問題を設定していたため，問題での前提条件に対する認識の違いによるものであった．また，それらの回答者に関しても 2 回目のコマンド選択では，「パネルヒータ ON」を選択しており，検証用コマンドとして本手法による探索と同じような推論を行っていると言える．

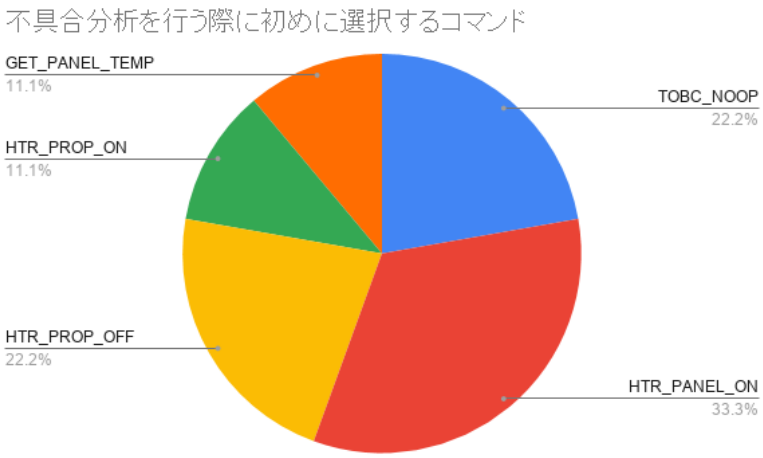

図 3.9　不具合分析時に初めに選択するコマンドの調査結果

また，このコマンドを選択した理由を見ると，「パネルヒータ ON」コマンドを送る人は積極的に故障候補の絞り込みを行う目的で選択しており，「推進系ヒータ OFF」コマンドを送る人は，故障していると考えられる推進系ヒータに通電を続けるのが危険であると考え，安全を考慮してこのコマンドを選択していた．本手法でコマンドの安全性を示す指標として提示するものは，電力と姿勢による制約からくるものと，状態をどれだけ変化させるかという点から提案を行っていた．しかし，不具合発生時に故障箇所が含まれている可能性があることを考えると，そのままの状態を保持することが危険であるという考え方もできる．そのため，安全状態に戻すという視点での指標の提案が必要であると考えている．
また，状態を不具合発生前の状態に戻すコマンド「推進系ヒータ OFF」は同時に，故障候補を検証可能なコマンドであることが分かる．人間による不具合分析ではこのことを意識せずに，故障候補切り分けのための情報を見逃してしまうことも多い．本手法を用いることで，このコマンドも検証用コマンドとして提示することで切り分けのために必要な情報を見逃すことを防ぐことにつながる．

次に，初回の検証結果が異常であった場合に次に選択するコマンドに関して以下の図 3.10 に調査結果を示す．初回に選択するコマンドが本手法で提案したものと一致していたのに対し，次に選択するコマンドは大きなばらつきが見られた．こちらに関しても，これらのコマンドを選択した理

由を見ると，故障候補の特定よりも二次故障の発生を防ぐために安全を考慮してコマンドを選択した場合や，故障候補をより網羅的に洗い出したことによる検証プロセスの違いであることがわかった．

前者に関しては，上述したように不具合状態を保持することによる危険を考えており，本手法では扱うことができていない概念である．

また後者に関しては，本手法では故障候補の洗い出しは不具合発生時のコマンドとテレメトリが通る経路のみに限定しているが，不具合分析経験が豊富な人による推論では，コマンドによる状態変化を起こすための電力供給源を考えた場合や，コンポーネントの機能単位の回路の故障を疑ったものが見られた．実際の不具合発生時では，このようにコンポーネント間の接続関係だけでない機能間のつながりや，状態変化によって発生する波及効果によって他のコンポーネントに影響を与えることも考えられるため，今後の課題としてより網羅的な故障仮説の生成に取り組む必要があると考えている．

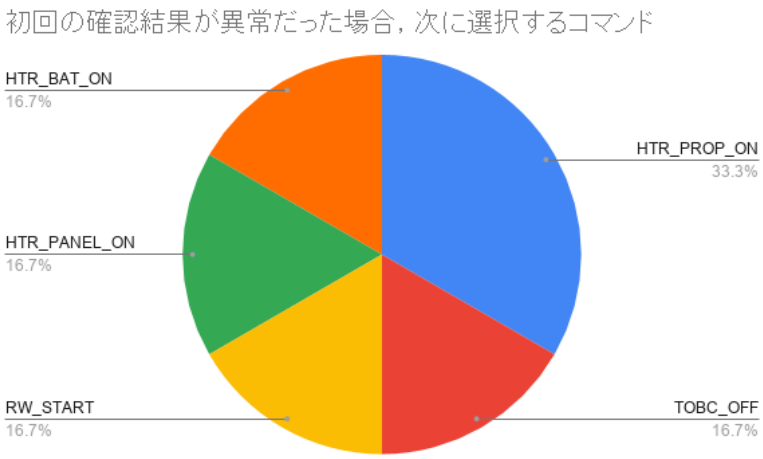

図 3.10　初回の検証で確認したテレメトリが異常であった場合

また，本手法による故障箇所特定は単一故障を前提としたアルゴリズムとなっていたが，複数故障を考慮した分析を行う人も見られた．その場合，安全なコマンドを評価するためには全ての故障候補の組み合わせの場合の数だけコマンドによる影響を考慮する必要がある．それらの組み合わせに対して波及効果までを考え，安全性を評価できるのは人間の強みであると言える．一方で，このように多くの候補を考えることは多くの知識と経験を要する作業である．経験の少ない人物が安全性の評価を十分に行うことを支援するために，複数故障を考慮した検証作業に関しても考慮する必要があることが分かった．

3.2.4　故障箇所の特定ができなかった例（温度計故障）

次に，以下の図 3.11 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．

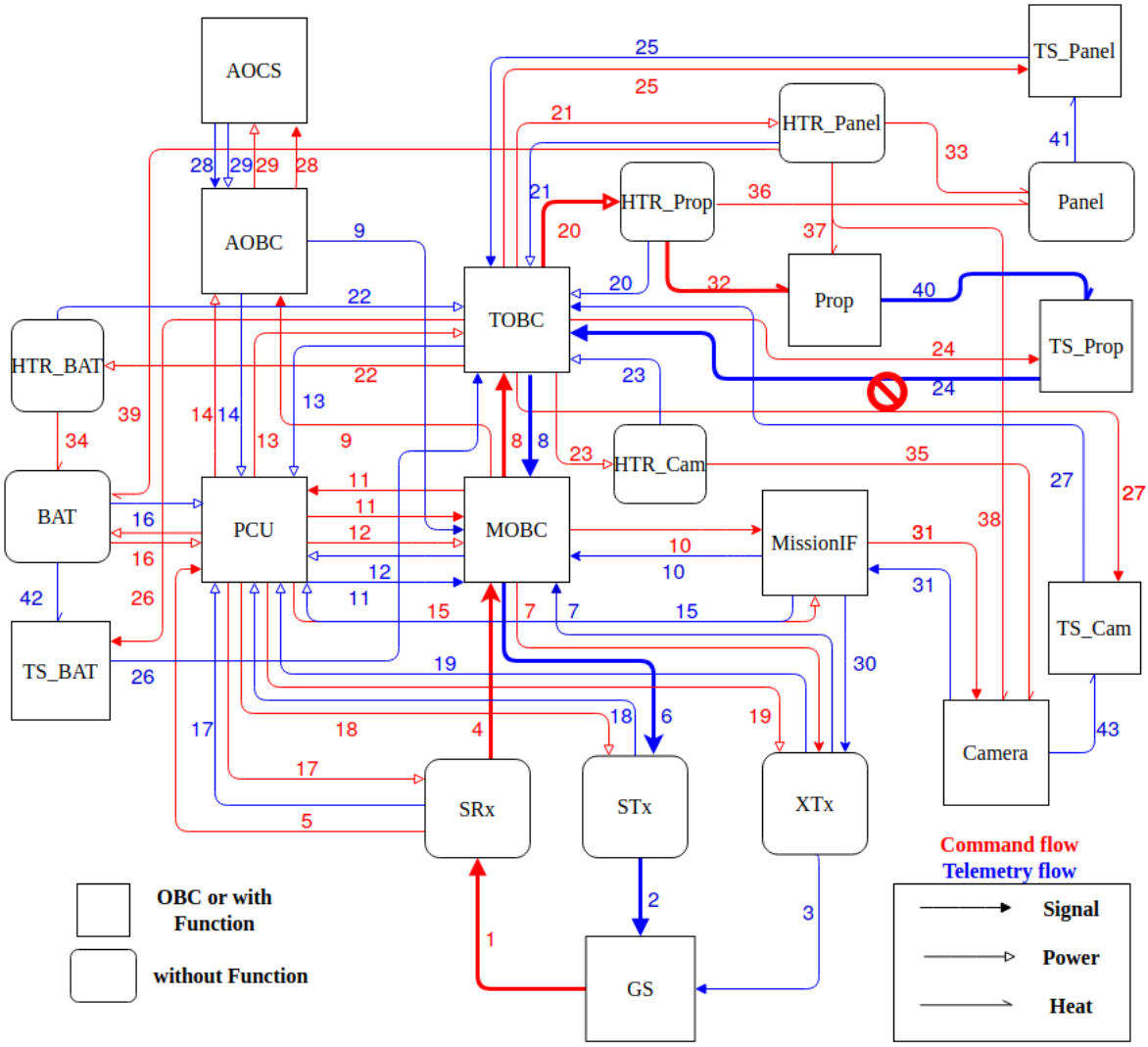

図 3.11　故障箇所：リンク 24(推進系温度計-TOBC 間) の時の故障候補

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 3.6) のものと同じであり，どちらのコマンドから検証を始めても結果は同じで以下の図 3.12 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，提案されたコマンドによって推進系温度計に変化は見られず，異常テレメトリとなる．そのため，どちらのコマンドによる検証でも切り分けを行うことができず，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，この衛星の設計では今回扱った故障「(推進系) 温度計故障」が発生した際に，本手法を用いて故障特定を行えないことが分かる．

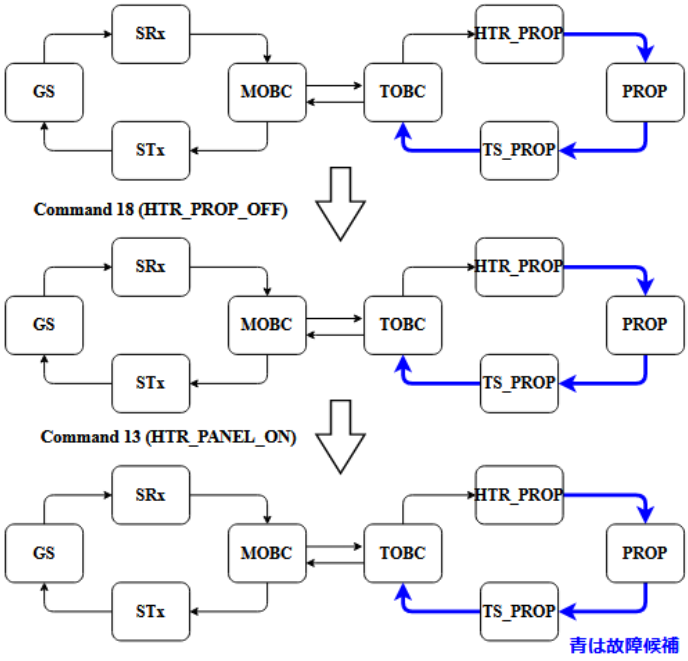

図 3.12　温度計故障時の検証の流れ

一方で，不具合分析の過程で得た情報を用いた人間の推論を組み合わせることで故障箇所を特定することは可能である．まず，今回の事例では，TOBC から推進系ヒータへの電源供給ライン（コマンドリンク 20）が正常であることは「推進系ヒータ電流値」によって確認できており，同時に「パネル温度」の上昇によってヒータが正常に作動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切り分けられる．

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，「推進系－推進系温度計間」か「推進系温度計－ TOBC 間」のいずれかに確実に故障箇所があることが推測できる．よって，温度計自体の故障か，TOBC の温度計読み取り回路のどれかが故障していると判断することが可能である．

このように，本システムのみでは故障箇所の特定が行えなかった場合においても，提示された選択肢に従って検証を行うことによって，故障箇所を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．

また，人間の推論を組み合わせても故障箇所の特定が行えなかった場合には，設計の不備を考え

ることができる．衛星の地上試験及び軌道上での運用では，基本的には本手法のようにコマンドとテレメトリでのやり取りによって人間との通信を行う．そのため，これらの情報を用いて不具合分析が行えるような，コンポーネントの接続関係，コマンドやテレメトリの設計を行う必要がある．本手法を設計段階の衛星に対して適用することによって，衛星システムとして不具合発生時の検証能力が十分であるかを確認することが可能である．

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考え，設計の正しさを確認する．この過程において，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．

第 4 章

結論

4.1　まとめ

本研究では，衛星内の情報伝達経路モデルを用いてコマンドによる故障箇所特定の過程を体系化する手法，及びコマンドの安全性と故障候補切り分け能力の大きさを示す指標を提案した．また，本手法を簡易的な衛星モデルで仮想的に与えた故障状態に対して適用し，本手法の評価を行った．実践例では，複数のコマンドの中から故障箇所特定のために適切なコマンドを探索し，そのコマンドに対して評価指標の計算を行ったものと共に提示した．提示されたコマンドを用いて検証を行うことにより，想定した故障箇所を特定できる能力があることを示した．
次に，コマンドの故障候補切り分け能力を示す指標に関して考察を行った．そこでは，時間制約がより厳しい状況での不具合分析においては「平均確認可能性」の大きなコマンドを用いることで数少ない通信の機会を利用できる可能性が高まること，不具合分析に使用できる時間が明確に分かっており，ある程度の余裕がある場合には「検証コマンド総数」が小さなコマンドを選択することで，最終的に少ない作業工程で故障箇所の特定が行える可能性があることを示した．
また，故障状態によってはシステムのみでは特定を行えなかった場合があることもわかった．そのような場合には人間の推論と組み合わせることで故障候補の絞り込みができ，提示された選択肢に従うことで推論に必要な情報を効率的に得ることができることを示した．同時に，人との推論を組み合わせても特定できない場合には設計の不備を考えられるため，設計の不備を発見することにつながることを示した．

一方で，本手法では安全性を示す指標として状態変化の小さなものが好ましいと考えていたが，人による不具合分析結果との比較によって，実際は安全を確保するために状態変化をする場合があること，本手法で見ることができる故障は主に接続関係に関するものに限定されており，コンポーネントが持つ機能の故障まで特定することはできないことなどが知見として得られた．
これらを踏まえた今後の展望を次に示す．

4.2　今後の展望

今後取り組むべき課題として以下にまとめた．

- コンポーネントの機能の接続関係を組み込んだ，より粒度の細かい故障箇所特定
- リンクの正常確率に実機の情報を組み込むことによる，検証の効率化
- 設計情報からのモデル自動生成

上記の課題に関して，今後進めていくべき具体的な取り組みを述べる．

4.2.1　機能モデルを組み込んだ粒度の細かい故障箇所特定

まず，コンポーネントの接続関係の故障だけでなく，各コンポーネントの機能の故障を扱うために，各コンポーネントが持つ機能の詳細なモデル化が必要になる．コンポーネントの機能は，階層的になっておりある機能を満たすためのサブ機能が存在するというような関係になっている．検証を行う過程に関してもまずは粗い粒度で故障箇所の特定を行い，その後故障箇所コンポーネントの持つ機能単位での故障の特定，サブ機能単位での故障の特定という風に段階的に切り分けを行っていく必要があると考えている．
また，上述した機能に対する故障を見るためにはテレメトリが持つ情報の意味に関してもシステムが扱えるようなオントロジーを定義する必要がある．現在，人間からの入力情報として，テレメトリが正常か異常かの 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，

- テレメトリの取得可否
- テレメトリに含まれるパラメータの大小
- テレメトリの変化の有無

などが考えられる．これらの情報の違いを扱うために「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．また同時に，機能に対応した故障を特定するためには，テレメトリと状態量の対応付けを考える必要がありこのモデルをどのように構築するかに関しては今後検討を進めていく必要がある．

4.2.2　対象システムの信頼性の組み込み

また，本研究では各リンクの正常確率を簡単のため 0.5 と固定して与えた．実際の衛星開発では，その機関で長年培われた技術や，実績のある機器など信頼性の高い設計項目が存在し，同時に新規実装項目や開発途中のソフトウェアなど信頼性の低い設計箇所も存在するなど，ばらつきがある．そこで，リンクの正常確率を，実際に設計・製造している衛星の各設計項目に対する信頼度に

基づいて考えることによって，より効率的な検証作業につながることが想像できる．これらの信頼度は事前知識的に組み込むことは可能であるが，今後の課題としては試験結果を元に不具合発生するコンポーネントの信頼性を下げていくなどし，学習させるシステムを構築することによって，対象とする衛星のモデルの再現度を高めることを検討したい．

4.2.3　設計情報からのモデル自動生成

また，本研究で用いた簡易的な衛星モデルは全て手作業によって記述した．実際に開発される衛星では，超小型衛星であっても膨大な数のコマンド及びテレメトリ，多くのコンポーネント間の回路などが存在し，手作業によるモデルの記述は作業コストやヒューマンエラーのリスクを考えると現実的ではない．また，実際の開発過程においては設計変更が幾度に渡って繰り返されることがほとんどである．そのため，その設計変更との整合性が取れなくなると本手法を用いた不具合分析を行うことは不可能である．これらを踏まえると，設計情報からモデルを自動生成し，設計情報の更新に応じて本手法で用いるモデルにも反映されるシステムが求められる．設計情報の中には，各コンポーネントの電気回路の接続情報や，物理的位置関係など，本手法で用いたモデルを構築するために必要な情報が含まれている．それらの文書の中からモデル生成を行う手法に関して，検討していこうと考えている．

参考文献

1) Catherine C Venturini. Improving Mission Success of CubeSats. Technical report, 2017.
2) 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.
3) 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.
4) 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.
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.

謝辞

本論文は筆者が東京大学工学部航空宇宙工学科での卒業研究の成果をまとめたものである．
指導教員の船瀬龍准教授には，ご多忙の中研究相談にのって頂き，私の研究方針が定まらない段階からも取り留めのない内容の私の話に親身に耳を傾けて下さり，終始ご指導を頂いた．ここに深謝の意を表する．

中須賀真一教授には，私とのミーティングを何度も快く受けて下さり，その度に私の研究に対する漠然としたイメージを適切な言葉で表現し，新たな気付きと多くのアイデアを頂いた．ここに深謝の意を表する．

五十里哲助教には，輪講の場や資料の添削を通じて自身の豊富な衛星開発経験の知見を元にした数々の鋭いご指摘を頂くと共に，資料作成の細部にわたりご指導を頂いた．ここに感謝の意を表する．

小畑俊裕共同研究員はご多忙の中，私とのミーティングを快く受けて下さり，私の浅薄な研究アイデアに対してご自身の知見を元に多くのご指摘とご提案をして頂いた．また，私の選択した研究テーマに興味を持って下さり，有難い応援の言葉を頂いた．深く御礼申し上げる．

横堀慎一研究員には，民間企業での衛星開発の経験を元にシステムズエンジニアリング的な取り組みや，大規模な衛星開発における課題に関する話をして下さり，研究の種となるような気付きを与えて頂いた．心より感謝申し上げる．

松本健研究員にはミーティングの場で，自身の衛星開発及び運用経験に基づき，様々な過去の不具合分析の事例を取り上げ，不具合分析における課題をご説明頂いた．また，全てのお話の中で衛星開発に関して無知な私が理解するまで丁寧にご説明頂いた．心より感謝申し上げる．

中島晋太郎共同研究員には，研究グループ発表時のコメントや個別のミーティング，高橋亮平さんとのミーティングなど数多くの場面を通して私の研究への深いご理解と丁寧なご指導を頂いた．厚く御礼を申し上げる．

高橋亮平さんには，自身の研究やプロジェクトでの立場上ご多忙にも関わらず，私の研究相談を毎週開いて下さり，進捗が生めていない中でも現状を聞き，終始ご指導頂いた．また，研究者として未熟な私に対して，文書作成や輪講での発表に対する助言を頂き，研究の方向性が定まらなかった私を正しい方向に導いて下さった．ここに深謝の意を表する．

また，3 章で述べた不具合分析に関する調査にご協力下さった中須賀・船瀬研究室の皆様，新型コロナウイルス感染症の影響により研究室での研究活動を行うことが困難な中，私に素晴らしい環

境を与えて下さった研究室関係者各位に改めて感謝する．

最後に，この 3 年間の東京での不自由ない生活を支援し続け，このような学習の機会を与えて下さった，家族に深謝する．