

第 1 章

モデルベース不具合分析手法の仕様

1.1 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、モデルに対して非常に高い忠実度が求められる。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。

1.2 不具合分析アルゴリズム

以下の図 1.1 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 1.1 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。

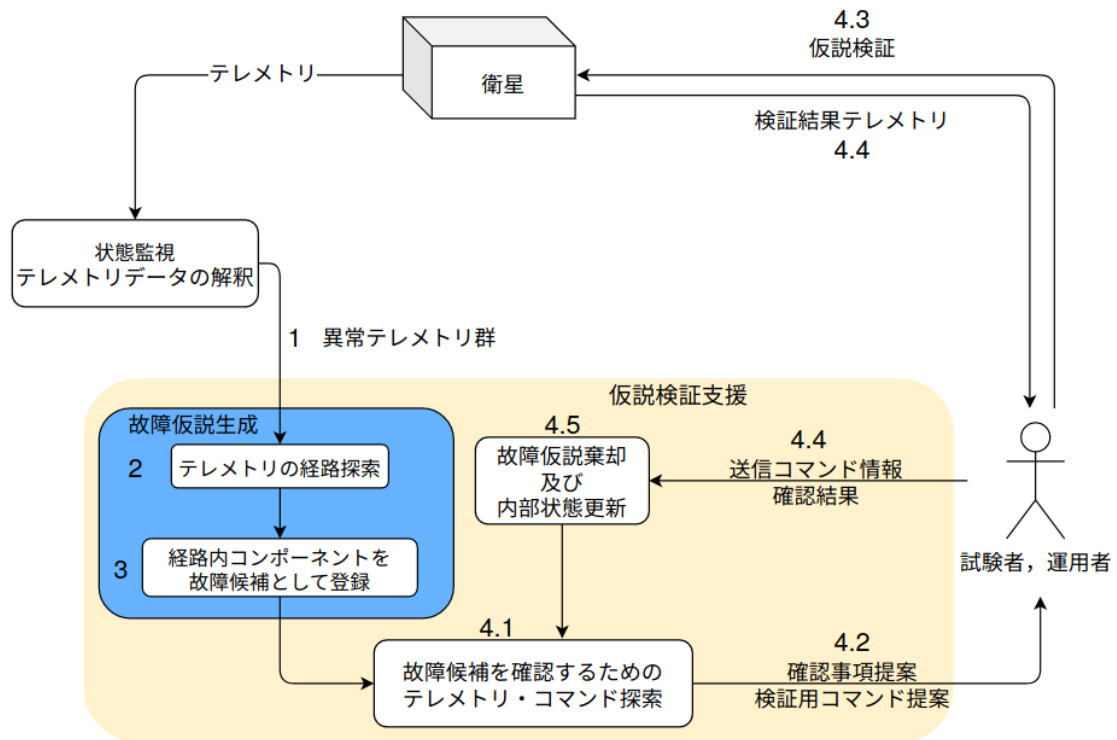


図 1.1 不具合分析の流れ

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。

1.2.1 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。

1.3 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。

1.3.1 対象とするテストケース

今回、以下の図 1.2 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

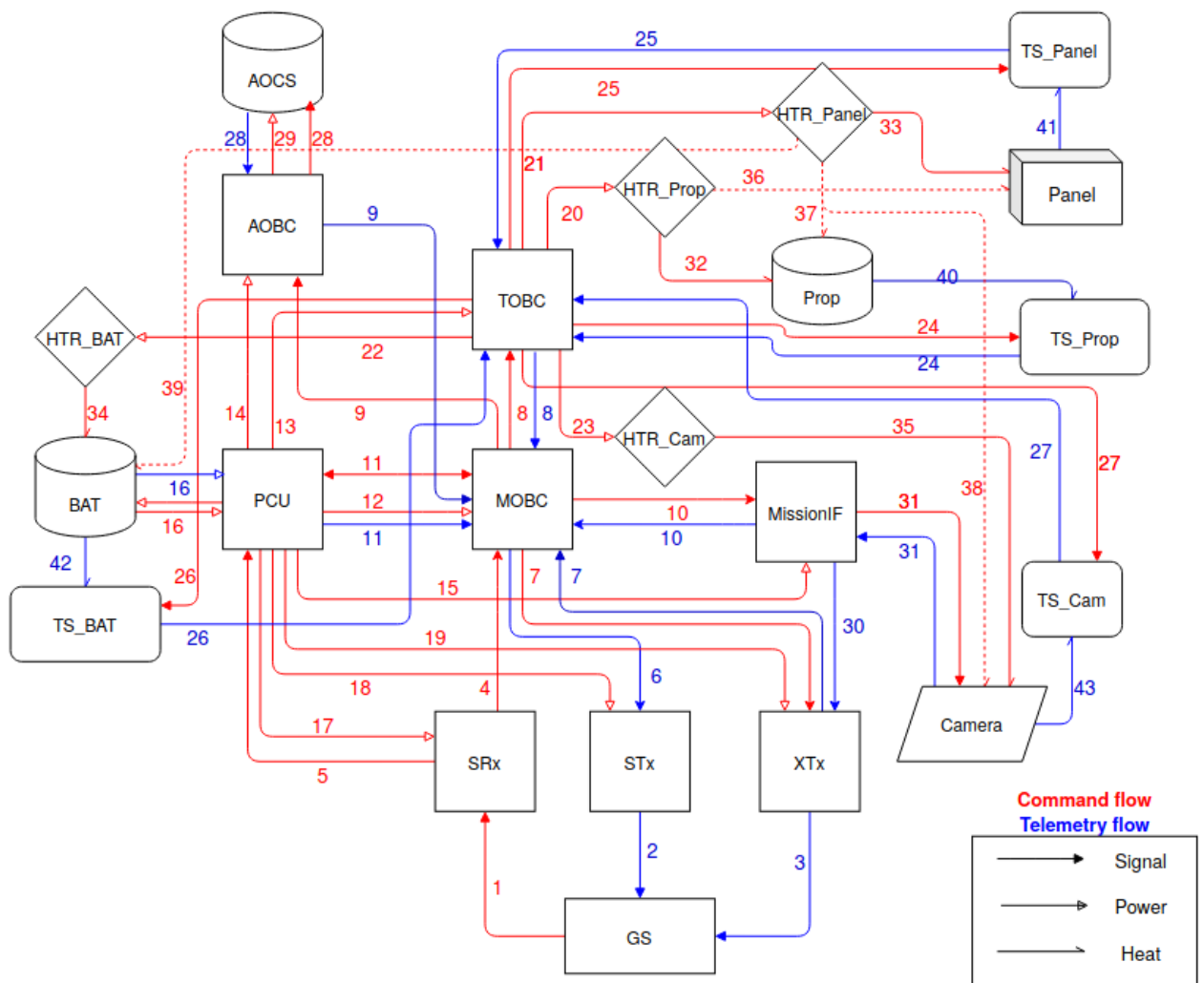


図 1.2 簡易衛星モデル

1.3.2 各コンポーネント間の接続関係モデル

来村ら [?] 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 [?] 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 1.1 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はすぎない。

表 1.1 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 Table 1.2 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 1.2 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 1.3 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 1.3 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 1.3 コンポーネント初期状態例

1.3.3 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 Table 1.3, 1.4 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 1.3 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	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 1.3 に示すテレメトリの経路及び, Table 1.4 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[?] 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 1.4 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进行操作するという関係性を表現可能になる。

表 1.4 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	5,6,10,16,20	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"]}]},
..

```

図 1.4 コマンドの機能モデル

1.4 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。

1.4.1 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。バッテリー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「バッテリー残量」と「コマンドを打つことによって発生する消費電力」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、バッテリー残量は電源が 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ー残量と、コマンドを打つことによって発生する消費電力
- 姿勢変化を起こすか否か
- コマンドを打つことで変化するテレメトリの数

1.4.2 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。

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

情報でリンク 1 を確認することができる確率は

$$P(l_1|R_1) = \prod_{i \in \mathbb{F}_1, i \neq 1} P(l_i = \text{normal}) \quad (1.1)$$

$$= \left(\frac{1}{2}\right)^4 \quad (1.2)$$

であることが分かる．ここで、 R_1 は経路 1, また

$$\mathbb{F}_1 = \{2, 3, 5, 7\} \quad (1.3)$$

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

$$P(l_1|R_2) = \prod_{i \in \mathbb{F}_2, i \neq 1} P(l_i = \text{normal}) \quad (1.4)$$

$$= \frac{1}{2} \quad (1.5)$$

$$P(l_1|R_3) = \prod_{i \in \mathbb{F}_3, i \neq 1} P(l_i = \text{normal}) \quad (1.6)$$

$$= 1 \quad (1.7)$$

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 $\mathbb{R}_{ki} = \{R_{1i}, \dots, R_{N_{ki}}\}$ （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

$$P(l_i|C_k) = \max\{P(l_i|R_{1i}), \dots, P(l_i|R_{N_{ki}})\} \quad (1.8)$$

となる．

式 (1.8) 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を $\mathbb{R}_k = \{R_1, \dots, R_{N_k}\}$ 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

$$P_m(C_k) = \frac{1}{N_{F_k}} \sum_{i=1}^{N_{F_k}} P(l_i|C_k) \quad (1.9)$$

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}_{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、

$$E(C_k) = N_{F_k} P_m(C_k) \quad (1.10)$$

$$= \sum_{i=1}^{N_{F_k}} P(l_i|C_k) \quad (1.11)$$

となる。これを「確認可能リンク数」と定義する。

ここで、図 1.5 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

$$P_m(C_1) = \frac{1}{N_{F_1}} \{P(l_1|C_1) + P(l_2|C_1) + P(l_3|C_1) + P(l_5|C_1) + P(l_7|C_1)\} \quad (1.12)$$

$$= \frac{1}{5} \left\{ 1 + \frac{1}{2} + \left(\frac{1}{2}\right)^4 + \left(\frac{1}{2}\right)^4 + \left(\frac{1}{2}\right)^4 \right\} \quad (1.13)$$

$$= 0.3375 \quad (1.14)$$

$$E(C_1) = 1.6875 \quad (1.15)$$

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して不確実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。

リンクの正常確率をもとにすると、ある経路を通るテレメトリが正常もしくは異常となる確率を算出することが可能になる。そのため、あるコマンドを選択した際に影響を受けるテレメトリに対して、検証結果を

全体としていくつのコマンドが必要になりうるのかを算出する。

以上では簡単のため、各リンクの正常確率は 0.5 として統一していたが、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検証が足りていない可能性があるため、低い正常確率を設定したりするなど、人間の判断でモデルに工夫を加えることが可能である。

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。

1.4.3 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ー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。