第 1 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

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．
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 1.1 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．

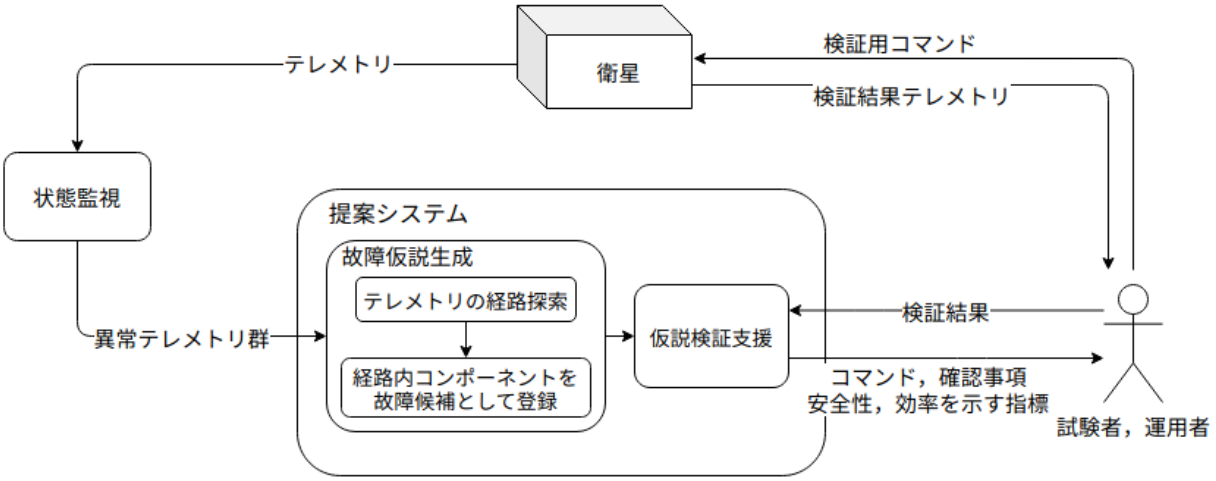

図 1.1　本手法による不具合分析の構成

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．

1.2　事前定義モデル

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

1.2.1　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

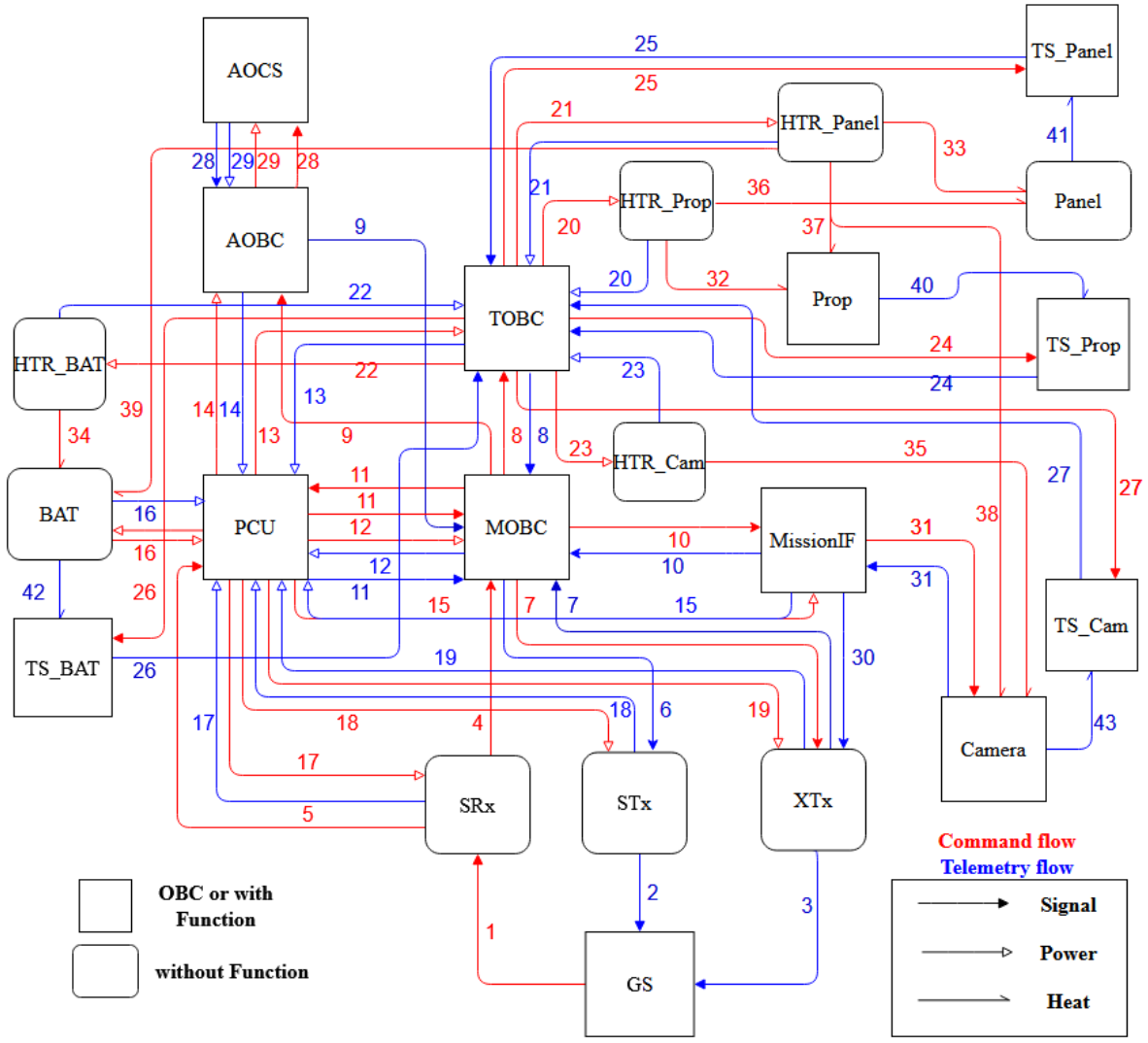

図 1.2　衛星内コンポーネントの接続関係図 (矢印の数字はリンクの ID)

1.2.2 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，以下の表 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 (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 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．以下の表 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 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

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．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 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 1.3　コンポーネント初期状態例

1.2.3　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 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 (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 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	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 1.3 に示すテレメトリの経路及び，表 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．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 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	5,6,10,16,17,18,19,20	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	

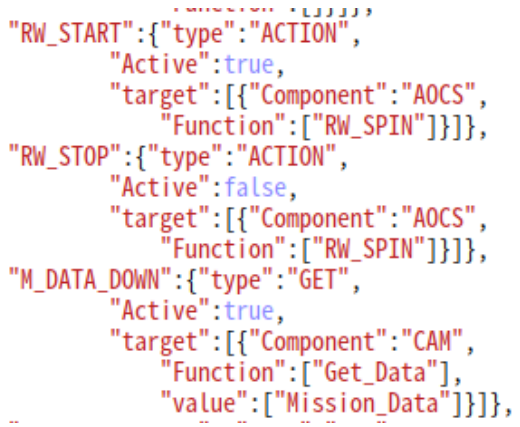

図 1.4　コマンドの機能モデル

1.3　本手法による故障仮説検証の流れ

1.3.1　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 1.5 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．

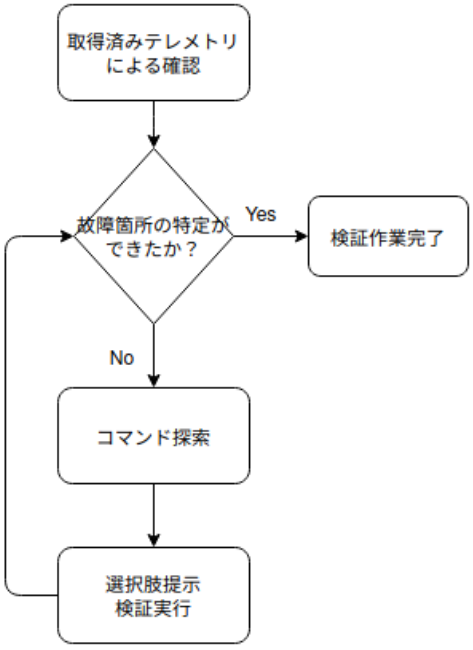

図 1.5　故障仮説検証の流れ

1.3.2　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 1.6 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．

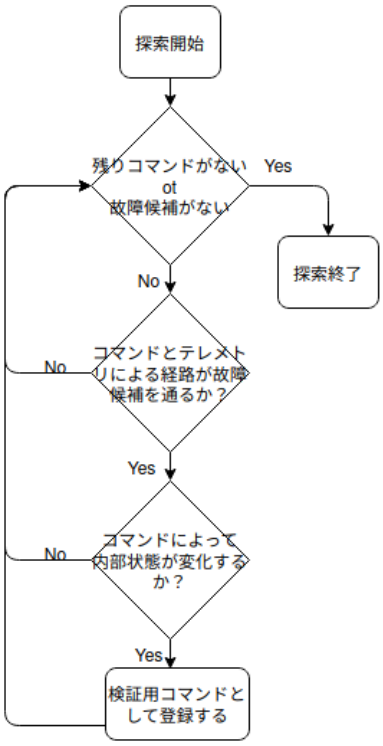

図 1.6　検証用コマンド探索アルゴリズム

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．

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

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．

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 $P(l = \mathrm{normal})$，異常である確率を $P(l = \mathrm{abnormal})$ 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 1.7 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．

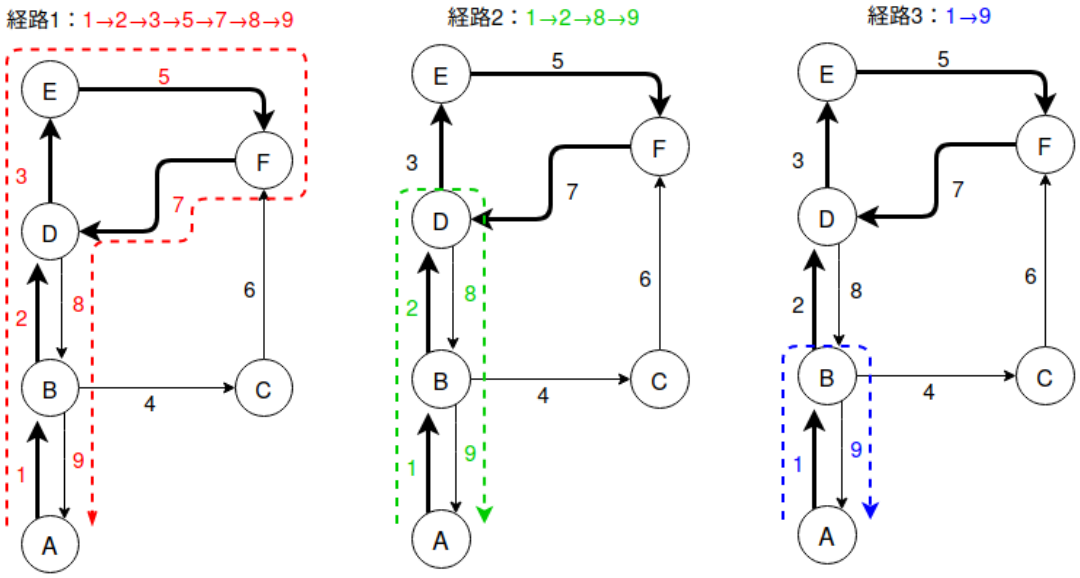

図 1.7　故障候補とそれを確認するための情報伝達経路の例

図 1.7 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

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

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

であることが分かる．ここで，R_1 は経路 1，また

$$\mathbb{F}_1 = \{2, 3, 5, 7\} \tag{1.3}$$

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

$$P(l_1|\mathrm{R}_2) = \prod_{i\in\mathbb{F}_2, i\neq 1} P(l_i = \mathrm{normal}) \tag{1.4}$$

$$= \frac{1}{2} \tag{1.5}$$

$$P(l_1|\mathrm{R}_3) = \prod_{i\in\mathbb{F}_3, i\neq 1} P(l_i = \mathrm{normal}) \tag{1.6}$$

$$= 1 \tag{1.7}$$

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 $\mathbb{R}_{ki} = \{\mathrm{R}_{1_i}, \cdots, \mathrm{R}_{N_{ki}}\}$（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

$$P(l_i|\mathrm{C}_k) = \max\{P(l_i|\mathrm{R}_{1_i}), \cdots, P(l_i|\mathrm{R}_{N_{ki}})\} \tag{1.8}$$

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

$$P_m(\mathrm{C}_k) = \frac{1}{\mathbf{N_{F_k}}} \sum_{i=1}^{\mathbf{N_{F_k}}} P(l_i|\mathrm{C}_k) \tag{1.9}$$

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 $\mathbb{R}_k$ 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，

$$E(\mathrm{C}_k) = \mathbf{N_{F_k}} P_m(\mathrm{C}_k) \tag{1.10}$$

$$= \sum_{i=1}^{\mathbf{N_{F_k}}} P(l_i|\mathrm{C}_k) \tag{1.11}$$

となる．これを「確認可能リンク数」と定義する．

ここで，図 1.7 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

$$P_m(\mathrm{C}_1) = \frac{1}{\mathbf{N_{F_1}}}\{P(l_1|\mathrm{C}_1) + P(l_2|\mathrm{C}_1) + P(l_3|\mathrm{C}_1) + P(l_5|\mathrm{C}_1) + P(l_7|\mathrm{C}_1)\} \tag{1.12}$$

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

$$= 0.3375 \tag{1.14}$$

$$E(\mathrm{C}_1) = 1.6875 \tag{1.15}$$

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 (1.16) 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．

$$P_m(\mathrm{R}_j) = \frac{1}{N_{F_{kj}}}\sum_{i=1}^{N_{F_{kj}}} P(l_i|\mathrm{R}_j) \tag{1.16}$$

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 (1.17)，(1.18) 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．

$$P(\mathrm{T}_j = \mathrm{normal}) = \prod_{i\in\mathbb{F}_j} P(l_i = \mathrm{normal}) \tag{1.17}$$

$$P(\mathrm{T}_j = \mathrm{abnormal}) = 1 - P(\mathrm{T}_j = \mathrm{normal}) \tag{1.18}$$

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 1.8 では，上で示した例（図 1.7）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 1.7 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．

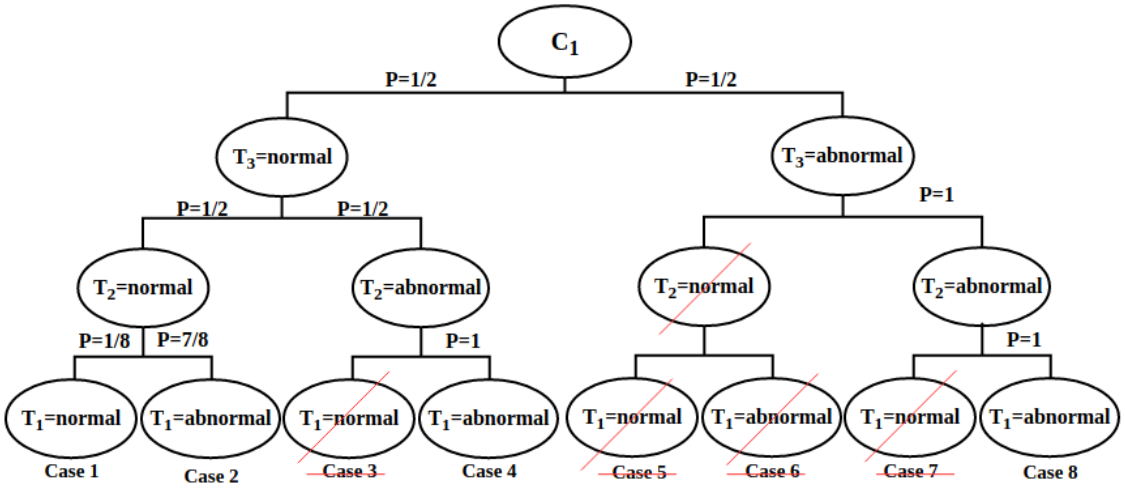

図 1.8　各テレメトリの結果による検証過程の種類

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 1.7 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

$$P(\mathrm{T}_3 = \mathrm{normal}) = P(l_1 = \mathrm{normal}) = \frac{1}{2} \tag{1.19}$$

$$P(\mathrm{T}_3 = \mathrm{abnormal}) = P(l_1 = \mathrm{abnormal}) = \frac{1}{2} \tag{1.20}$$

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 1.8 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 1.8 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．

$$
\begin{aligned}
P(\text{Case 1}) &= P(\mathrm{T}_3 = \mathrm{n} \cap \mathrm{T}_2 = \mathrm{n} \cap \mathrm{T}_1 = \mathrm{n}) \\
&= P(\mathrm{T}_3 = \mathrm{n}) P(\mathrm{T}_2 = \mathrm{n} | \mathrm{T}_3 = \mathrm{n}) P(\mathrm{T}_1 = \mathrm{n} | \mathrm{T}_2 = \mathrm{n}, \mathrm{T}_3 = \mathrm{n}) \\
&= \frac{1}{2} \times \frac{1}{2} \times \frac{1}{8} \\
&= \frac{1}{32} \qquad (1.21) \\
P(\text{Case 2}) &= P(\mathrm{T}_3 = \mathrm{n} \cap \mathrm{T}_2 = \mathrm{n} \cap \mathrm{T}_1 = \mathrm{a}) \\
&= \frac{7}{32} \qquad (1.22) \\
P(\text{Case 4}) &= P(\mathrm{T}_3 = \mathrm{n} \cap \mathrm{T}_2 = \mathrm{a} \cap \mathrm{T}_1 = \mathrm{a}) \\
&= P(\mathrm{T}_3 = \mathrm{n} \cap \mathrm{T}_2 = \mathrm{a}) \\
&= \frac{1}{4} \qquad (1.23) \\
P(\text{Case 8}) &= P(\mathrm{T}_3 = \mathrm{a} \cap \mathrm{T}_2 = \mathrm{a} \cap \mathrm{T}_1 = \mathrm{a}) \\
&= P(\mathrm{T}_3 = \mathrm{a}) \\
&= \frac{1}{2} \qquad (1.24)
\end{aligned}
$$

次に，図 1.8 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 1.9 のようになる．問題設定として，図 1.7 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 1.9 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．

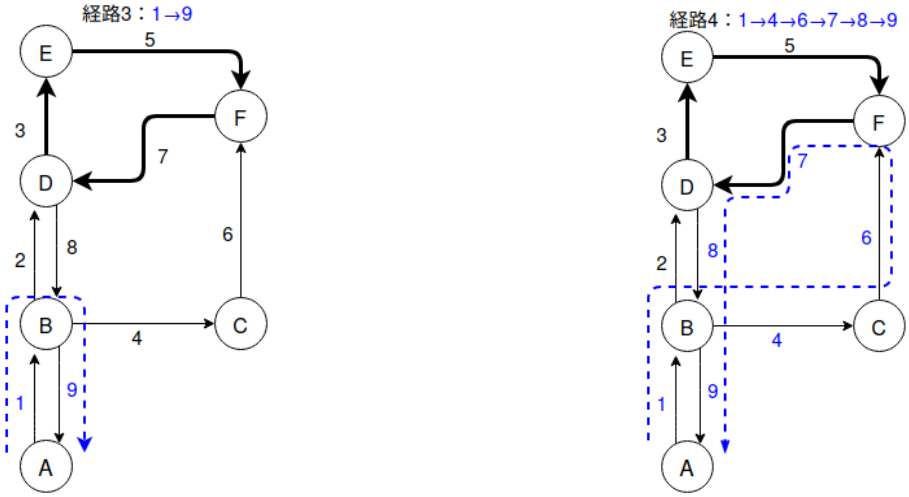

図 1.9　Case 2 の場合に残る故障候補とコマンド 2 に影響を受けるテレメトリ (3,4) が成す経路

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

1.10 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 (1.17)，(1.18) 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 (1.25)，(1.26) のように求まる．

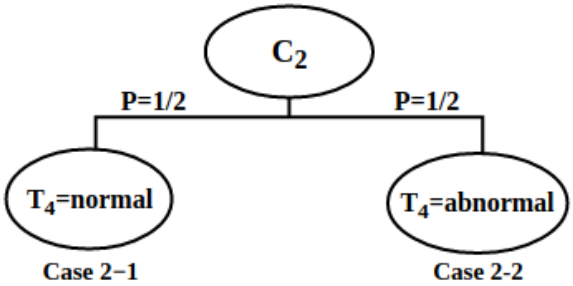

図 1.10　Case 2 の時のコマンド 2 送信時の結果

$$
\begin{aligned}
P(\text{Case 2-1}) &= P(\text{Case 2})P(\mathrm{T}_4 = \text{normal}) \\
&= \frac{7}{32} \times \frac{1}{2} = \frac{7}{64} \qquad (1.25) \\
P(\text{Case 2-2}) &= P(\text{Case 2})P(\mathrm{T}_4 = \text{abnormal}) \\
&= \frac{7}{32} \times \frac{1}{2} = \frac{7}{64} \qquad (1.26)
\end{aligned}
$$

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．

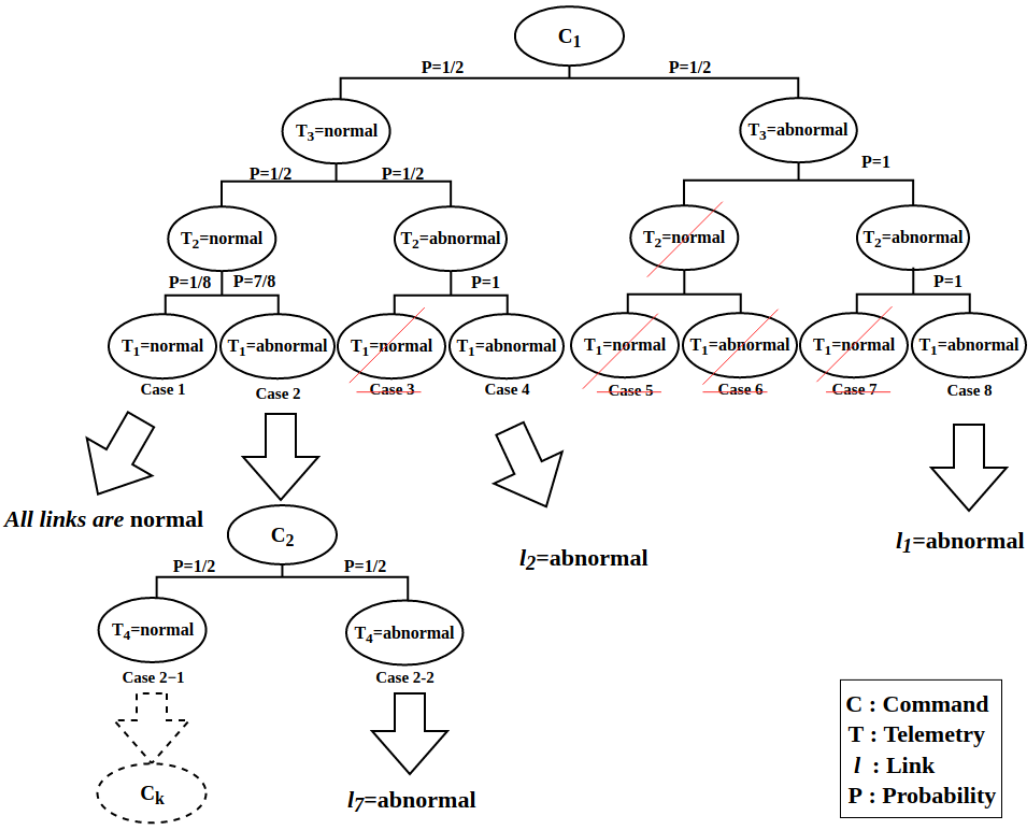

図 1.11　検証プロセスの全体像

図 1.11 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 (1.27) 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．

$$
\begin{aligned}
N(\mathrm{C}_1) &= \sum_{\text{Case } i \in \mathbb{C}} P(\text{Case } i) N_{\text{Case } i} \\
&= \frac{1}{32} \times 1 + \frac{7}{64} \times 2 + \frac{7}{64} \times 2 + \frac{1}{4} \times 1 + \frac{1}{2} \times 1 \\
&= 1
\end{aligned}
\tag{1.27}
$$

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．

$$N(\mathrm{C}_2) = 1.875 \tag{1.28}$$

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．

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.