

衛星の情報伝達経路モデルを用いた 対話的不具合分析支援手法の提案

03-183005 西本 慎吾

2020年12月01日

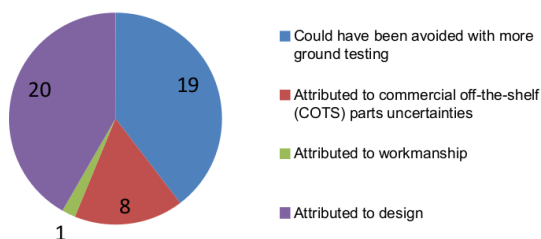
概要

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重要である。しかし、衛星のような複雑なシステムでは、内部の機器間で異常状態が波及するため、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が必要とする。そこで、本研究ではコンポーネント間の接続関係モデル、情報伝達の経路モデルを用いて衛星の故障候補の検証方法（確認事項、打つべきコマンド）を人間の判断を支援する指標と共に提示することで、不具合分析を支援する手法を提案する。本手法では、簡易的な衛星モデルに対して実践することでコマンドによる故障箇所の特定を効率的に行えること、設計の不備を発見することにつながることを確認した。

1 序論

1.1 研究背景

超小型衛星開発に大学などの参加が増加している中、信頼性の低さが問題となっている¹⁾。軌道上故障の調査の結果、衛星の故障原因の多くは設計・製造過程に起因するものである²⁾ことが分かっており、それらの多くは地上試験によって確認することができるものであるという結果が出ている³⁾。

図1 超小型衛星の故障原因に関する調査結果²⁾

1.2 問題提起

以上より、地上試験での不具合分析が不十分になっていることが、超小型衛星の信頼性の低さの原因の一つである。地上試験での不具合分析を十分に行うためには以下の2点の作業に高い知識と経験が必要とされる。

- 故障候補の網羅的洗い出し
- 故障候補の切り分け作業

まず、故障候補を洗い出す作業はFTAを用いて不具合事象から考えられる故障モードを網羅的に洗い出すことを行うが、衛星は内部機器の物理的相互作用が複雑であるため、人による思い付きでは網羅的に行うことは難しい。また、切り分け作業は衛星から得られる情報を元に衛星の安全を確保しながら行う必要があり、不具合事象から衛星の状態を十分に想像できなければ安全な切り分け作業を行うことができない。これに対して、下表1に示すように、故障候補の洗い出しを網羅的に行う研究が盛んにおこなわれている。一方で、不具合分析の一つの大きな課題である検証過程に関して取り組んだものは少ない。

表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+ ⁴⁾	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 ⁶⁾	高	故障仮説生成
本手法	中	故障箇所特定支援

1.3 本研究の目的

以上より、次の機能を持った不具合事象から故障候補の切り分けまでの作業を体系化し、不具合分析経験の少ない人を支援する手法を提案する。

- 異常テレメトリから故障候補を生成する。
- 故障候補を確認するためのコマンドおよびテレメトリを提案する。
- 上の探索結果に関して優先度を人間に提示する。

上記の機能を実現するために、本種研究では以下の3点の構築を目的とする。

- 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
- 人間の判断を支援するためのコマンドの評価指標

2 モデルベース不具合分析手法の仕様

2.1 不具合分析アルゴリズム

本手法による不具合分析の流れを下図2に示す。本研究の対象は色付けているところであり、人間との対話的に故障箇所の特定を行う。これにより実機の情報をシステムにフィードバックしながら故障箇所を絞り込むことができるようになる。

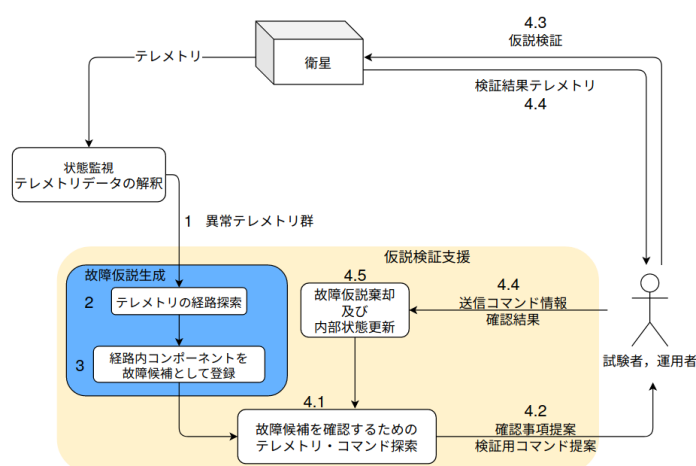


図2 本手法による不具合分析の流れ

2.2 モデル

上述したアルゴリズムで故障箇所特定を行うために使用するモデルに関して以下に示す。

2.2.1 コンポーネント間接続関係モデル

來村ら⁷⁾は拡張デバイスオンロジーとして、機器間の接続関係を「ポート」と「導管」という概念を用いて表現している。これを元に、コンポーネントの接続関係を表す「リンク」(表2)を定義した。各リンクには正常確率を属性として持ち、これによって以降に述べるコマンドの評価を行うことを可能にしている。また、表3のように各コンポーネントがリンクを属性として持ち、コマンドの情報伝達で使用するリンクとテレメトリの情報伝達で使用するリンクを区別している。

表2 リンク定義

ID	Link_name	Compo1	Compo2	Medium	Probability
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

表3 コンポーネント定義

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	

2.2.2 情報伝達経路モデル

以下の表4,5のようにコマンドおよびテレメトリを定義した。それぞれ情報伝達の経路を上述のリンクによって表現している。また、コマンドには属性として影響を与えるテレメトリ及びコマンドの種別を与えており、これによってコマンドを送信した際に得られる情報を類推している。

また、テレメトリのモデルでは、テレメトリが変化するトリガの種類を指定しており、これによって故障箇所特定に必要な情報取得のために取る行動を決めることができる。ここでは簡単のため、軌道運動などによる状態変化は考慮せず、時間とコマンドだけによる状態遷移を考えている。

表4 コマンドモデル

ID	CommandName	impact_TEL_ID	type	path					
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		

表5 テレメトリモデル

ID	TelemetryName	TransitionTrigger	path					
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	

コマンドおよびテレメトリの機能モデルでも状態量の更新に関して説明するために必要。

2.3 評価指標の提案

本手法の対象は地上試験における支援であるが、不具合分析に利用する情報の粒度がコマンドとテレメトリのみであるため、軌道上不具合発生時の故障箇所特定にも利用可能である。そこで、以下では地上試験及び軌道上での運用時の両方で重要となる指標を提案し、本手法が両状況で使い分け可能なフレームワークであることを示す。

2.3.1 衛星の生存への副作用

まず、生存への副作用を示す指標として、以下の3点を与える。

- コマンドを打つ前の電力状態と、コマンドを打つことによって発生する電力消費量
- 姿勢変化を起こすか否か
- コマンド送信によって変化するテレメトリの数

前者2点の電力と姿勢による制約から来る指標は運用時に特有のものであり、コマンドを打つことで衛星の安全を脅かすことがないように危険な動作を明示的に示すことで、未熟な運用者による誤ったコマンド送信を防ぐ目的がある。また、不具合発生時は衛星の状態に対する把握が不十分であるため、衛星の状態を大きく変化させるコマンドは危険であるといえる。そのため、コマンドに

よって発生する状態変化の大きさを定量的に示す指標として3点目の指標を与えている。

2.3.2 故障候補切り分け能力

運用時は可視時間が限られており、その時間内に不具合の改修を行わなければミッションが失敗するような、時間制約を考慮した不具合分析を行う場面が考えられる。その際には、少ないコマンド数で効率的に故障個所の特定を行えることが望ましい。

まず、一つのコマンドで切り分けられる故障候補の数を表す指標に関して述べる。以下の図3に示すような故障候補(太矢印)がある場合を考える。あるリンク(l_i)の状態を確認するためにはその経路(R_j)内にある他のリンクが正常である必要がある。よって、

$$P(l_i|R_j) = \prod_{m \in \mathbb{F}_j, i \neq m} P(l_m = \text{normal}) \quad (1)$$

の確率でリンク l_i を確認できる。ここで \mathbb{F}_j は R_j 内の故障候補リンクの集合、 $P(l_m = \text{normal})$ は上述した各リンクの正常確率を表している。 $P(l_i|R_j)$ はコマンドが形成する各径路すべてに対して求まるのでそれらの最大値を取りコマンド(C_k)による l_i の確認可能性は

$$P(l_i|C_k) = \max\{P(l_i|R_j)\} \quad (2)$$

となる。

また、これが各リンクに対して求められるのでそれらの平均を取り、「平均確認可能性($P_m(C_k)$)」及び C_k によって確認可能なリンクの数を表す「確認可能リンク数($E(C_k)$)」が以下のように求まる。ここで、 \mathbf{N}_{F_k} はコマンドが形成する経路内にある故障候補の数を表す。

$$P_m(C_k) = \frac{1}{\mathbf{N}_{F_k}} \sum_{i=1}^{\mathbf{N}_{F_k}} P(l_i|C_k) \quad (3)$$

$$E(C_k) = \mathbf{N}_{F_k} P_m(C_k) = \sum_{i=1}^{\mathbf{N}_{F_k}} P(l_i|C_k) \quad (4)$$

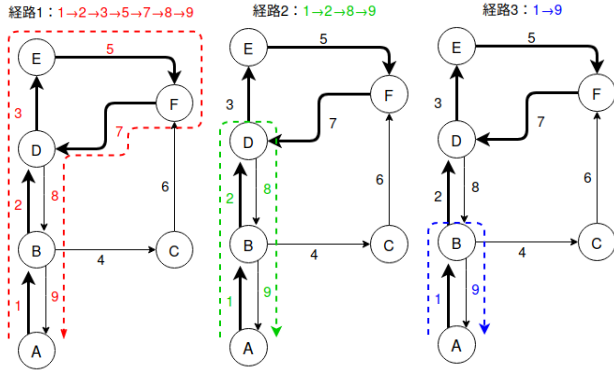


図3 故障候補とそれを確認するための情報伝達経路の例

式(3),(4)がどちらも高いコマンドを選択することで、一つのコマンドでより多くの故障候補を絞り込むことが可能である。

次に、あるコマンドから検証を開始した時に、最終的に故障箇所の特定を行うまでにかかるコマンドの総数に関して述べる。以下の図4に示すように、あるコマンドによる検証を考えると各テレメトリの結果によって残る故障候補が変化する。この時、故障候補が残っている場合にはそれに応じたコマンドの探索を行う必要がある。各テレメトリが正常値を示すか否かは各リンクごとの正常確率を用いて以下の式(5),(6)のように算出でき、これを元に図4の各Caseへの確率が求まる。ここで、 T_j はIDが j のテレメトリを表しており、経路の添え字と対応している。

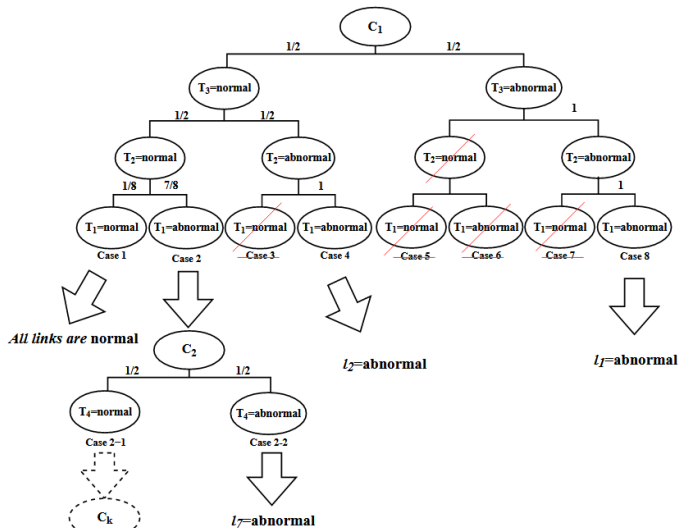


図4 検証プロセスの全体像

$$P(T_j = \text{normal}) = \prod_{i \in \mathbb{F}_j} P(l_i = \text{normal}) \quad (5)$$

$$P(T_j = \text{abnormal}) = 1 - P(T_j = \text{normal}) \quad (6)$$

各Caseに応じて、最終的故障箇所を切り分けるまでのコマンドの総数が異なるので、それによって以下の式(7)のように期待値が求められ、

$$N(C_k) = \sum_{\text{Case } i \in \mathbb{C}} P(\text{Case } i) N_{\text{Case } i} \quad (7)$$

これを「検証コマンド総数」と定義する。ここで、 \mathbb{C} は検証が終了した結果の各場合(Case)の集合である。検証コマンド総数が少ないコマンドを選択することによって、全体的にかかるコマンドの数の期待値が小さい検証プロセスを選択することができる。

2.3.3 評価指標の使い分け

以下の表6に以上で示した指標をどのように使い分けるかに関して示す。地上試験時は電力と姿勢による制約はほぼ考えなくてよい。効率性を重視したい場合は $P_m(C_k)$, $E(C_k)$ が高く、 $N(C_k)$ が小さいコマンドを選択すれば良い。

表6 地上試験と運用時での指標の優先度(3段階, 降順, 赤: 安全重視, 黒: 効率重視)

	電力	姿勢	影響 TEL 数	$P_m(C_k)$	$E(C_k)$	$N(C_k)$
地上試験	-	-	2,1	1,3	1,3	1,3
運用時	3,1	3,1	2,1	1,2	1,2	1,3

3 提案手法による実践と評価

3.1 対象問題設定と実践結果

テストケースに用いる衛星モデルを以下に示す。

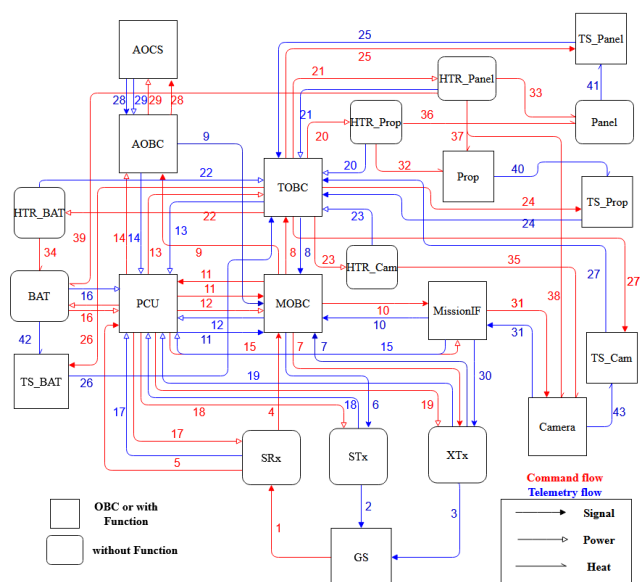


図5 簡易衛星モデル

実践例での対象故障。複数の事例を確認して、どうだったかという結果も欲しい

実践例として上手く特定ができるものと、できないものを示す。

- ヒータの接触不良に関して
- コンポーネントの故障などに関して (そもそものテレメトリが発行されていない場合など)

3.2

4 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。どちらかというの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、¹⁾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。

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) 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.
- 4) Peter Struss and Oskar Dressier. "Physical Negation" - Integrating Fault Models into the General Diagnostic Engine. Vol. 89, pp. 1318–1323, 1989.
- 5) Kota Yamaguchi and Hori Koichi. Fault Network Analysis of Artificial Satellite Using Ontology. pp. 1–4, 2014.
- 6) 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.

7) Yoshinobu Kitamura and Riichiro Mizoguchi.

A Framework for Systematization of Functional Knowledge based on Ontological Engineering. PhD thesis.