

第 1 章

本手法による不具合分析の実践と評価

1.1 概要

不具合分析の具体的な流れをみるために、いくつかの事例を取り上げて実践した結果を示す。以下では、まず実際に故障箇所特定が行えた事例を取り上げる。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を変えること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。次に、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。最後に、本手法によって扱うことのできる故障の種類に関する限界を述べ、発展させるために必要な方針に関して議論を行う。

1.2 実践例

1.2.1 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 1.1 のような故障を考え、不具合分析を行っていく。

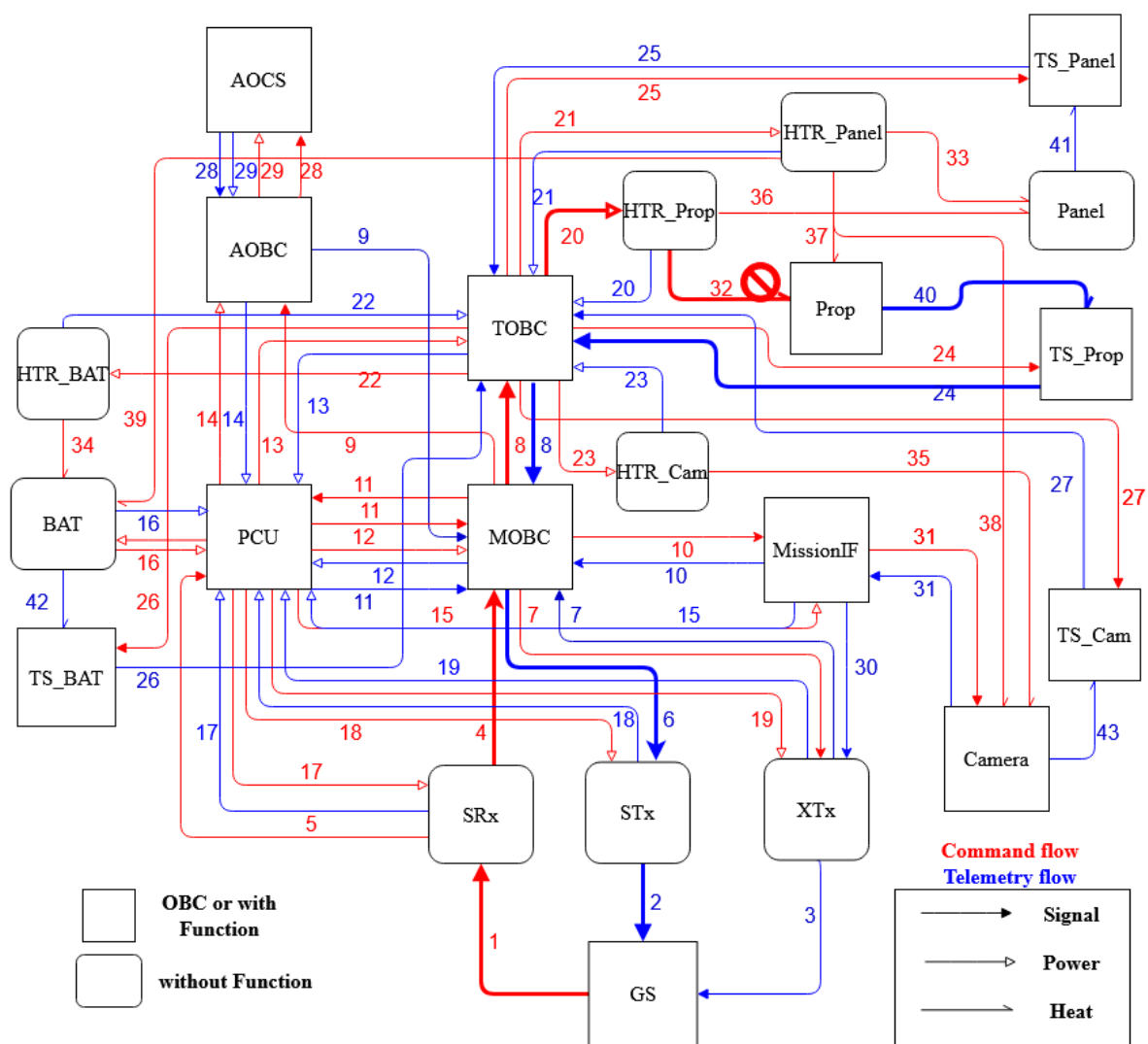


図 1.1 故障箇所：リンク 32(推進系ヒータ - 推進系間) の時の故障候補

図 1.1 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 1.2 に故障候補の決定及び，テレメトリ情報を用いた確認の段階を示す．故障候補の決定では，不具合事象を検知するきっかけとなったコマンドとテレメトリが形成する経路を探索し，targetTEL，targetCOM として提示している．

その後、時間変化するテレメトリ情報を用いて確認できる故障候補を提示し、返ってくるテレメトリが正常 (OK) か否 (NG) かを入力させることで、切り分けを行っている。今回の例では MOBC 及び TOBC は正常に動作しているはずなので、MOBC カウンタ及び TOBC カウンタは正常 (OK) と入力し、それを元に状態の更新を行っている。

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

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

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

図 1.2 テレメトリによる確認

次に、図 1.3 に示すのが、不具合発生時に送信していたコマンド情報から考えられるテレメトリの変化を用いて故障候補の確認を行う段階である。今回は、初期コマンドとしては異常検知の際に送ったコマンド (推進系ヒータ ON) のみを考えている。確認可能性の高い経路を形成するテレメトリから順に表示され、人間に確認をさせているのが分かる。ここでの確認テレメトリに関しても、今回の例では正常であるため、そのように入力し状態の更新を行っている。

```
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 1.3 初期コマンドを用いた確認

最後に、以下の図 1.4 に示すのが、上記の流れを経て残った故障候補を確認できるコマンドを探索し、

指標と共に提示した結果である。残った故障候補は、

- コマンドリンク 32:推進系ヒータ 推進系間
- テレメトリリンク 40:推進系 推進系温度計間
- テレメトリリンク 24:推進系温度計 TOBC 間

である。この時、探索結果として表示されたのはコマンド 13(パネルヒータ ON) と 18(推進系ヒータ OFF) であり、これらのコマンドに関する指標が図 1.4 のように示されている。

図中において $P_m(C)$ 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ー残量」、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 1.4 コマンドの選択肢表示及び検証過程

以下の図 1.5 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。

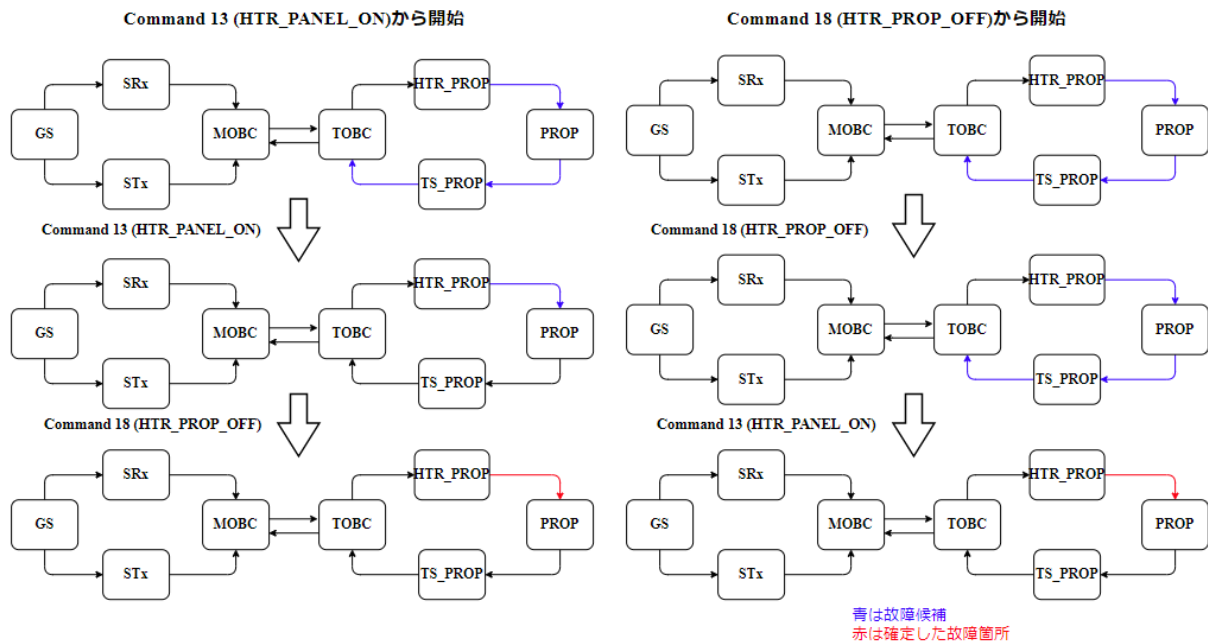


図 1.5 ヒータ接触不良時の各検証プロセスの流れ

1.2.2 評価指標に関する考察

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 1.5 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 1.4 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認することができる．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と，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 1.4 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。

1.2.3 温度計故障に関する検証

次に、以下の図 1.6 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。

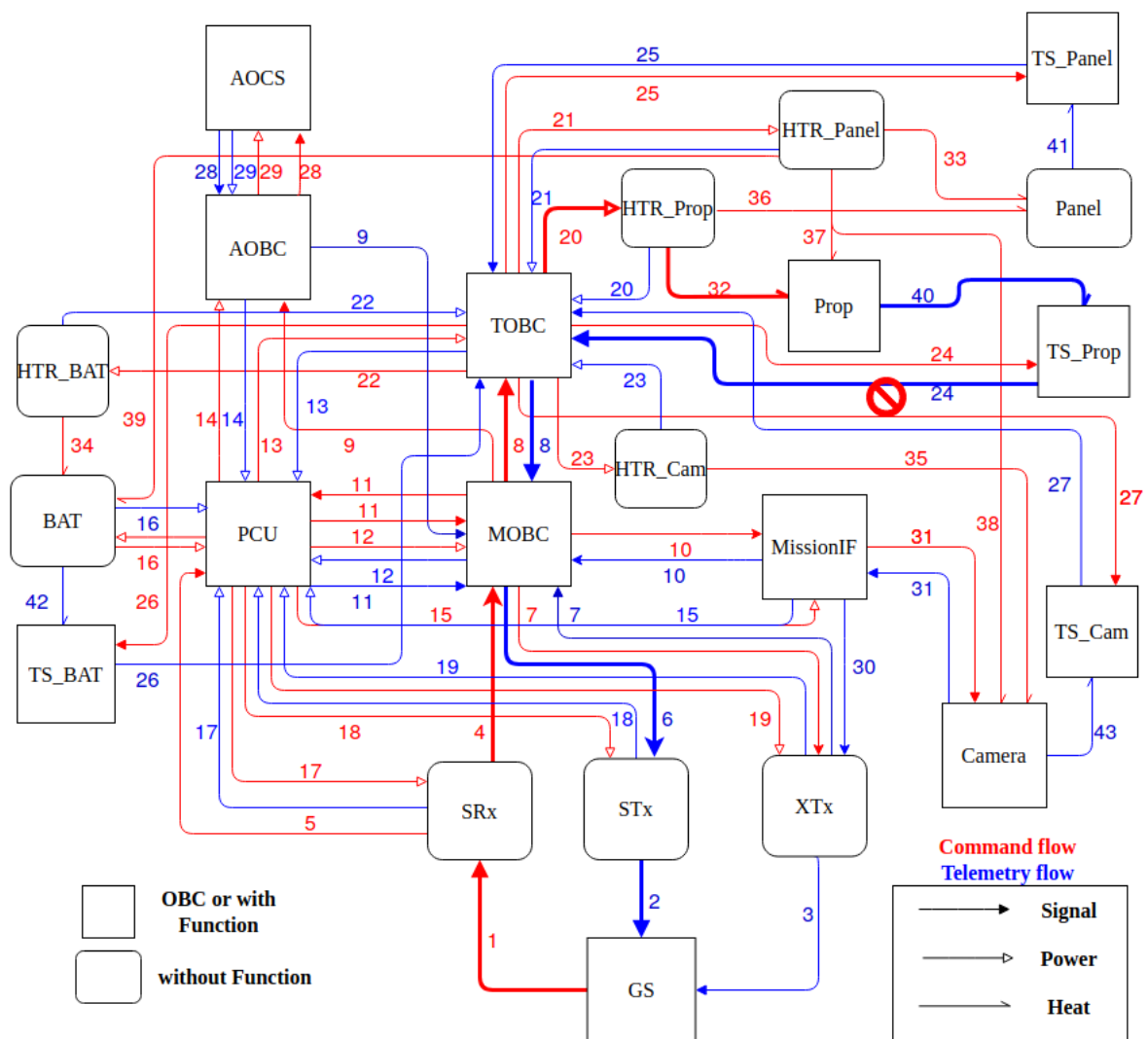


図 1.6 故障箇所：リンク 24(推進系温度計-TOBC 間) の時の故障候補

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 1.4) のものと同じであり、指標から判断するとコマンド 13 の方が切り分けられるリンクの数は大きい一方で全体のコマンド数の見積もりでは、コマンド 18 の方が少ないコマンド数であるという結果になっている。コマンド 13 及び 18 を初めに選択した結果はそれぞれ以下の図??、?? 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いないため、どちらも入力に異常 (NG) を与えている。最終的な結果は、どちらの過程を経ても同じになっており、一つのリンクにまで故障箇所の特定を行うことができなかった。このことから、この衛星モデルが今回扱った故障「(推進系) 温度計故障」が発生した際に故障特定を行える設計になっていないことが分かる。このように、本手法は単に故障箇所の特定を支援するだけでなく、設計の不備を洗い出すことにも利用できる。

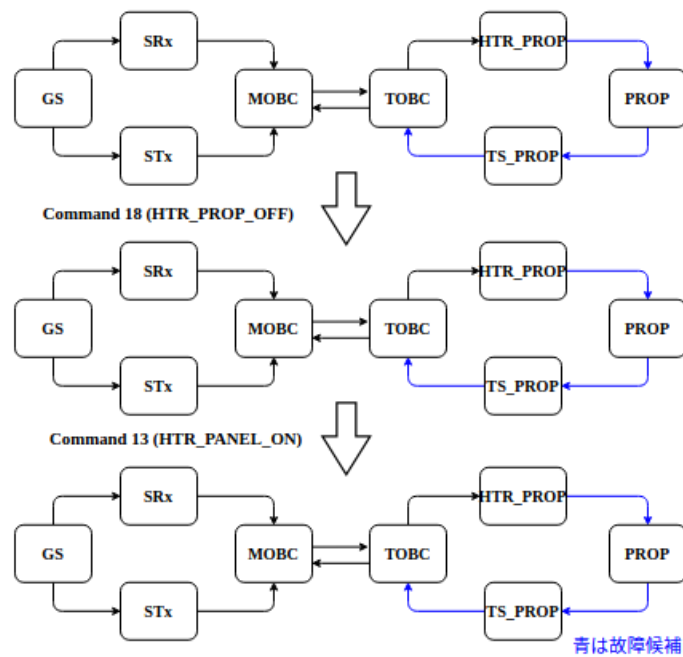


図 1.7 温度計故障時の検証プロセスの流れ

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

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