第３３回衛星設計コンテスト

ジュニア概要書

応募区分　　ジュニアの部

１．作品情報・応募者情報

作品名 超小型衛星におけるビット反転の対策
副題 Raspberry Pi Pico を 2 台用いた「ハイブリッド型 TMR」
学校名 長崎県立長崎西高等学校

２．ミッションの概要

放射線の影響により軌道上で発生するビット反転を監視し修正する仕組みを作成する。既存のミッション回路にマイコンを 1 つ加えるだけで実装可能な「ハイブリッド型 TMR」であれば、リソースが限られる超小型衛星においても、多数決方式や相互監視方式によりファイルエラーやソフトウェアエラーへの対策が可能となる。これらの対策により、将来的には超小型衛星の誤作動やエラーを起こす可能性を低くすることが目的である。

３.目的と意義

(a)目的

放射線によって起こるビット反転は人工衛星の誤作動や故障につながる。規模の大きい人工衛星開発では、放射線に強い機器を使うなどしてビット反転の対策を行っている。しかし、超小型衛星開発においては、主に民生品を用いる機会が多いにもかかわらず、ミッション期間が短いことと、放射線の影響を受けにくい低軌道でのミッションが多いこともあり、本格的なビット反転の対策はなされていないのが現状である。近年、超小型衛星の利用は増加しており、ミッションの高度化や長期化に伴い、今後ビット反転による誤作動や故障も増加すると考えられ、超小型衛星においても、厳しい宇宙環境の中で安定して処理・作動できるプログラムと機能の開発が求められる。そこで、超小型衛星内でビット反転が起こった時に訂正を行い、超小型衛星を正常な状態に戻すことができる仕組みの開発に取り組んだ。

(b)重要性・技術的意義等

安価なマイコンである Raspberry Pi Pico は軌道上での利用実績もあり、ミッション遂行のためのメイン CPU として利用されている。そこで、既存のミッション回路にマイコンを 1 つ加えるだけで、三重モジュール冗長（TMR:Triple Modular Redundancy）によるファイルエラー監視と、相互監視によるソフトウェアエラー監視が可能な仕組みを試作した。

４．アイデアの概要

現在用いられている TMR の手法は、3 台のハードウェアにそれぞれファイルを保存して比較する方法（ハード型 TMR：図 1）と、1 台のハードウェアに 3 つのファイルを保存して比較する方法（ソフト型 TMR：図 2）の二種類があるが、ハードウェアを 2 台用い、片方にはファイルを 1 つ、もう片方にはファイルを 2 つ保存して比較する新たな TMR を考案した。これを「ハイブリッド型 TMR」（図 3）と呼ぶことにする。ハイブリッド型の利点は 1 台のハードウェアの追加で容易に TMR の機能を持たせることが可能なことである。さらに、測定機器を持つ衛星本体はシールドせず、追加ハードウェアのみをシールドすることで全体の放射線耐性を強化できる。そこで、3 つの TMR の型に対して具体的な確率計算を行い、ファイルが破損した場合に TMR によってエラー修正が正しく実行できる確率を比較した。衛星本体のハードウェアを HW1、TMR を実装するために追加するハードウェアを HW2、HW3、TMR の実行ソフトウェアをそれぞれ SW1、SW2、SW3 とし、各 SW が破損の監視対象とするファイルを F1、F2、F3 とする。

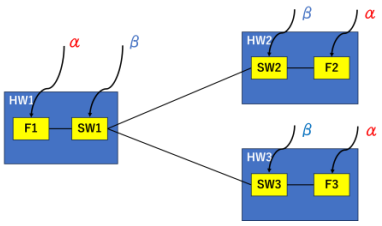


図 1：ハード型 TMR

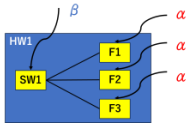


図 2：ソフト型 TMR

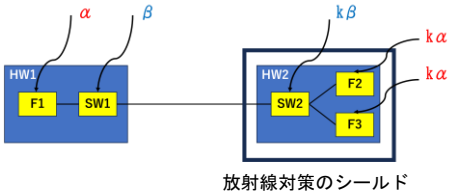


図 3：ハイブリッド型 TMR

■システム構成別の訂正成功確率

ファイルが破損する確率を α、ソフトウェアが破損する確率を β とする。TMR によって破損を修正するには、ファイルとソフトウェアの両方が正常である組が 2 組以上必要である。ハイブリット型では、ファイルが 2 つ搭載されたハードウェアをシールドで囲み、放射線対策を強化するものとする。その場合の破損確率は、シールドの係数を k（$0<k\leqq1$）とおき、それぞれ $k\alpha$、$k\beta$ と仮定して計算を行ったところ、ファイル破損を修正できる確率 P は次のようになった。（計算過程は補足説明資料に掲載）

・ハード型 TMR　　$P = 3(1-\alpha)^2(1-\beta)^2 - 2(1-\alpha)^3(1-\beta)^3$

・ソフト型 TMR　　$P = (1-\beta)[3(1-\alpha)^2\alpha + (1-\alpha)^3]$

・ハイブリッド型 TMR　$P = (1-k\beta)[(1-k\alpha)^2 + 2(1-\alpha)(1-\beta)(1-k\alpha)k\alpha]$

$10^{-10}\leqq\alpha\leqq10^{-1}$、$10^{-10}\leqq\beta\leqq10^{-1}$ の範囲で独立して変化させて、訂正成功率が最も高かった型を一覧にした（表 1）。同時に、k の値の変化による分布の変化を確認したところ、k=1（シールドなし）では、すべての組み合わせに対してハード型が有利となったが、k=0.1 まで小さくすると、α=0.01 以上のすべての組み合わせにおいて、ハイブリッド型が最も有利となった。以上のことから、シールドの係数 k 次第では、ハイブリッド型がハード型よりも有利になることがわかった。

表 1：k=0.1 において訂正成功率が最も高い型

k= 0.1		ソフトウェア故障率 β									
		1.00E-01	1.00E-02	1.00E-03	1.00E-04	1.00E-05	1.00E-06	1.00E-07	1.00E-08	1.00E-09	1.00E-10
ファイル故障率 α	1.00E-01	ハイブリッド型	ハイブリッド型	ハイブリッド型	ハイブリッド型	ハイブリッド型	ハイブリッド型	ハイブリッド型	ハイブリッド型	ハイブリッド型	ハイブリッド型
	1.00E-02	ハイブリッド型	ハイブリッド型	ハイブリッド型	ハイブリッド型	ハイブリッド型	ハイブリッド型	ハイブリッド型	ハイブリッド型	ハイブリッド型	ハイブリッド型
	1.00E-03	ハード型	ハード型	ハード型	ハード型	ハイブリッド型	ハイブリッド型	ハイブリッド型	ハイブリッド型	ハイブリッド型	ハイブリッド型
	1.00E-04	ハード型	ハード型	ハード型	ハード型	ハード型	ハード型	ハイブリッド型	ハイブリッド型	ハイブリッド型	ハイブリッド型
	1.00E-05	ハード型	ハード型	ハード型	ハード型	ハード型	ハード型	ハード型	ハード型	ハイブリッド型	ハイブリッド型
	1.00E-06	ハード型	ハード型	ハード型	ハード型	ハード型	ハード型	ハード型	ハード型	ハード型	ハード型
	1.00E-07	ハード型	ハード型	ハード型	ハード型	ハード型	ハード型	ハード型	ハード型	ハード型	ハード型
	1.00E-08	ハード型	ハード型	ハード型	ハード型	ハード型	ハード型	ハード型	ハード型	ハード型	ハード型
	1.00E-09	ハード型	ハード型	ハード型	ハード型	ハード型	ハード型	ハード型	ハード型	ハード型	ハード型
	1.00E-10	ハード型	ハード型	ハード型	ハード型	ハード型	ハード型	ハード型	ハード型	ハード型	ハード型

そこで本研究では、ハイブリッド型 TMR におけるファイルエラーおよびソフトウェアエラーの修正方法について検討を行い、具体的な実装方法を提案する。後述の通り、ファイルエラーの修正には TMR を、ソフトウェアエラーの修正にはマイコンの相互監視の手法をそれぞれ用いた。

■TMR によるファイルエラーの修正手順

Rasberry Pi Pico 2 台を HW1 と HW2 として、TMR を行う仕組みを作成した。HW1、HW2 間のデータ通信には UART を使用した。また、F1、F2、F3 の状態にはそれぞれに青、緑、赤の 3 色の LED を割り当て、ファイルの状態を視覚化した。正常であれば青、1 が足されるエラーが発生した場合は緑、2 が足されるエラーの場合は赤が点灯する。これによって TMR が行われる一連の様子を視覚化できるようになった。エラー修正の流れは次の通りである。

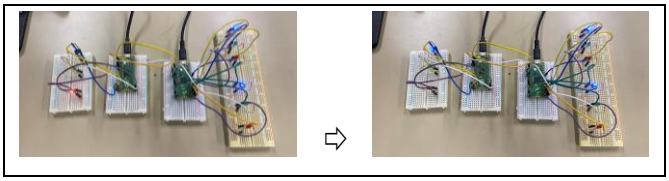


図 4：HW1 の F1 に生じたファイルエラーを TMR により修正

① HW1 がランダムに整数を生成→②整数を 3 つのファイル F1、F2、F3 として保存→③ファイルに 50%の確率でシングルイベント（SEU：Single Event Upset）を模擬したエラーが発生し、1 または 2 が整数に足される→④HW1 と HW2 はファイルに対して同一の演算処理（2 倍する）を実施→⑤HW1 が HW2 に演算結果を問い合わせる→⑥HW1 が 3 つの演算結果を比較→⑦3 秒待機→⑧エラーがあればファイルを修正

TMR で 3 つの演算結果を比較する際は以下の 3 パターンが考えられ、その後の処理は次の通りである。

[1]3 つの演算結果が全て同値の場合は全てのファイルが正常だと判断して何もしない。

[2]2 つの演算結果が同値で 1 つだけ異なる場合は異なる値のファイルにエラーが発生したと判断し、そのファイルの数値を多数派にあわせて修正する。

[3]3 つの演算結果が全て異なる場合は、TMR 処理ができず何もしない。

図 4 は F1 にエラーが発生した場合の TMR 処理の様子である。F1 がエラー（LED が赤）、F2、F3 はともに正常（LED が青）の状態から 3 秒後に F1 の LED が青に点灯している。F2、F3 のエラーについても同様の処理が成されたことから、1 つのファイルに発生したエラーは確実に修正することができた。

■相互監視によるソフトウェアエラーの修正手順

多数決を行う処理自体がビット反転を起こした場合はファイルの修正ができず、システムが予期しない動作をするので速やかにソフトウェアを修正しなければならない。そのためには、マイコンを電源リセットして、プログラムを再びメモリに読み直すことで対処できる。そこで、2 台のマイコンが相互監視して、相手の異常を検知したときに相手を電源リセットする仕組みを作成した。この仕組みでは 2 つのマイコンが互いにハートビート（正常に動作していることを示す信号）を送りあい、ハートビートが送られてこなかったら相手のマイコンで SEU によるソフトウェアエラーが発生したと判断し、相手のマイコンをリセットする。プログラムでは具体的に次のような動作を行う。

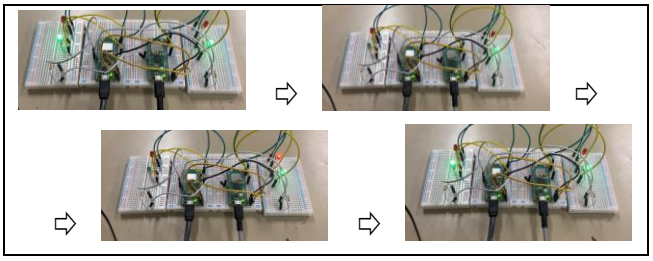


図 5：SW1 に生じた破損をリセットにより修正

①HW1 が UART 通信を用いてハートビートを 5 秒間隔で HW2 へ送信し、HW1 は緑色 LED を点灯→②HW1 には 20 秒間隔で 10 分の 1 の確率で SEU を模擬したエラーが発生し、ハートビートを HW2 へ送信しなくなる→③HW1 は緑色 LED を消灯させてプログラムのメインループから抜け出して永久に待機→④HW1 からのハートビートが 10 秒経っても送られてこない場合、HW2 は赤色 LED を 1 秒点灯→⑤HW2 は HW1 の RUN ピンの電圧を LOW（0V）にして HW1 をリセット→⑥リセットされた HW1 は緑色 LED を点灯してハートビートの送信を再開

このプログラムは HW2 でも同様に実行しており、HW2 から HW1 へハートビートを送信して常に相互監視を行っている。TMR のソフトウェア自体でエラーが生じる等が原因でマイコンがハングアップした場合には、監視役であるもう一方のマイコンが相手をリセットし、エラーから復帰できる。この仕組みであれば 2 つのマイコンで同時にエラーが発生しない限り、互いをリセットすることができるためエラーの修正が行われ続けて正常な動作を維持することができ、宇宙で生き抜くことができる。

■まとめ

以上のように、衛星本体にマイコンを 1 つ追加するだけで、ファイルエラーとソフトウェアエラーの両方に対応できる SEU 対策が実施できる。私達の取組が TMR の可能性の拡大に少しでもつながり、超小型人工衛星におけるビット反転によるエラーがなくなることで今後も超小型人工衛星が活躍して人間の明るい未来につながることを願っている。

５．得られる成果

- マイコンを 2 つ用いた TMR と相互監視が可能であることが確認できた。超小型人工衛星の重量が大きく増加することなく放射線対策ができる。
- 単体の SEU であれば、エラーの発生がファイルでもソフトウェア自体でも対応可能である。
- マイコンを既存の回路に 1 つ後付けすることで SEU 対策が可能となる。

６．主張したい独創性または社会的な効果

- マイコンを使用することで、コストを抑えつつビット反転による衛星の誤作動や故障のリスクを減らすことができる。
- ハイブリッド型という超小型衛星に最適な新たな TMR の手法を用いることで SEU による故障の可能性を最大限小さくすることができる。
- 元のミッション回路に対する変更は、マイコン 1 つを後付けするだけなので、人工衛星の重量も大きく増加せず、ミッションを邪魔することもなく放射線対策を行うことができる。

以上

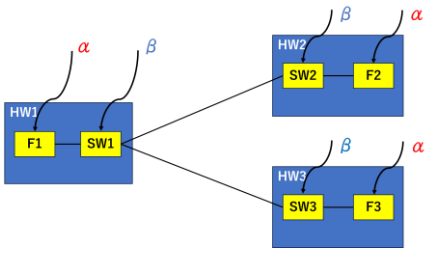
第３３回衛星設計コンテスト 補足説明資料

長崎県立長崎西高等学校 超小型衛星におけるビット反転の対策

システム構成別の訂正成功確率の計算過程について

■ハード型 TMR の訂正成功確率の計算過程

ハードウェア１～３(HW1, HW2, HW3)の中に，ソフトウェア１～３(SW1, SW2, SW3)があり，ファイル１～３(F1, F2, F3)をそれぞれ操作できる。HW1 が HW2 と HW3 に問い合わせて TMR によりファイル破損を訂正するためには，ソフトウェアとプログラムが両方とも破損していない組が2組以上なければならない。ファイル破損の確率を α，ソフトウェア破損の確率を β として，TMR によりファイル破損を訂正できる確率を求めた。



1. まず 1 台が「ファイルもプログラムも正常」である確率

- ファイル破損確率: α
- プログラム破損確率: β
- 2つの事象が独立と仮定すると，どちらも正常である確率pは $p = (1-\alpha)(1-\beta)$

2. 3 台のうち少なくとも 2 台が正常である確率

エラー訂正が機能できる条件は

- 2 台が正常，1 台が異常
- 3 台すべて正常

のどちらかである。

(a) 2 台が正常・1 台が異常

- 正常 2 台・異常 1 台になる並びは 3 通り(どの 1 台が異常かの選び方)。
- 確率は $3 \times p^2 \times (1-p)$

(b) 3 台すべて正常

- 確率は p^3

3. 全体の訂正成功確率P

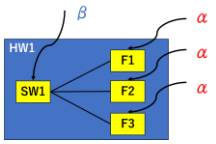
全体の訂正成功確率 P は(a), (b)の和で求まり，$P = 3p^2(1-p) + p^3 = 3p^2 - 2p^3$

ここで $p = (1-\alpha)(1-\beta)$ を代入してまとめると

$$\boxed{P = 3(1-\alpha)^2(1-\beta)^2 - 2(1-\alpha)^3(1-\beta)^3}$$

■ソフト型 TMR の訂正成功確率の計算過程

ハードウェア1(HW1)の中に，ソフトウェア1(SW1)があり，ファイル 1(F1)，ファイル2(F2)，ファイル3(F3)を操作できる。TMR によりファイル破損を訂正するためには，ファイルとソフトウェアが両方とも破損していない組が2組以上なければならない。F1，F2，F3 の破損の確率をα，SW1 の破損の確率をβとして，TMR によりファイル破損を訂正できる確率を求めた。



1. 状況整理

- 単一ハードウェア HW1 上に共通のソフトウェア SW1(破損確率 β)があり，ファイル F1，F2，F3(各破損確率 α)を操作する。
- エラー訂正ができるためには，少なくとも 2 個のファイルが正常かつ SW1 が壊れていないことが必要。
- SW1 が壊れているとどのファイルも使えなくなる。

2. 確率計算

ファイルが正常である確率を $q = 1 - \alpha$ とおく。

SW1 が正常である確率は $1 - \beta$ 。

SW1 が正常の条件下で「少なくとも 2 個のファイルが正常」である確率は，2 個正常の場合 と 3 個正常の場合の和になるため $3q^2(1-q) + q^3$

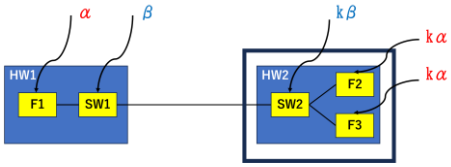
3. 全体の訂正成功確率P

したがって全体の訂正成功確率 P は

$$\boxed{P = (1-\beta)[3(1-\alpha)^2\alpha + (1-\alpha)^3]}$$

■ハイブリッド型 TMR の訂正成功確率の計算過程

ハードウェア1(HW1)の中に，ソフトウェア1(SW1)とファイル1(F1)が格納されている。ハードウェア 2(HW2)の中には，ソフトウェア 2(SW2)があり，ファイル2(F2)とファイル3(F3)を操作できる。HW1 が HW2 に問い合わせて TMR によりファイル破損を訂正するためには，ファイルとソフトウェアが両方とも破損していない組が2組以上なければならない。F1 の破損の確率を α，SW1 の破損の確率を β とする。また，シールドの係数を $k\,(0 < k \leqq 1)$ とおき，F2, F3 の破損の確率を $k\alpha$，SW2 の破損の確率を $k\beta$ として，TMR によりファイル破損を訂正できる確率を求めた。



1. 状況整理

- HW1 にはファイル F1 とソフト SW1 が入っており，両方が壊れていない確率 p は
$$p = (1-\alpha)(1-\beta)$$
- HW2 にはソフト SW2 が入っていて，ファイル F2 と F3 を操作できる。
  - SW2 が壊れていない確率は $(1-k\beta)$
  - F2 や F3 が壊れていない確率はそれぞれ $(1-k\alpha)$
- F2 と F3 は独立に壊れるが，SW2 が壊れると F2 と F3 のどちらも操作できなくなる。

HW1 が TMR で訂正できるためには，ファイルとソフトウェアの両方が正常な組み合わせが 少なくとも 2 つ 存在する必要がある。

2. 条件分け

(a) SW2 が壊れている場合

- 確率 $k\beta$
- このとき HW2 上の F2 と F3 はどちらも使えないため，正常な組は HW1 だけ。
- 2 つ以上の正常な組が存在できないため，エラーの訂正が不可能。

(b) SW2 が正常な場合

- 確率 $(1-k\beta)$
- HW1 が正常な確率 a
- F2 が正常な確率 $(1-k\alpha)$
- F3 が正常な確率 $(1-k\alpha)$
- これらは独立。

このとき「ファイルが少なくとも 2 つ正常」になる確率 q は，
$$q = (1-k\alpha)^2 + 2\,p\,(1-k\alpha)[1-(1-k\alpha)]$$
- 第1項は，F2 と F3 が両方正常な場合
- 第2項は，「HW1 と F2」または「HW1 と F3」が正常だが残りは正常でなくてもよい場合。

3. 全体の訂正成功確率 P

全体の訂正成功確率 P は $P = (1-k\beta)q = (1-k\beta)[(1-k\alpha)^2 + 2p(1-k\alpha)k\alpha]$

ここで $p = (1-\alpha)(1-\beta)$ を代入して

$$\boxed{P = (1-k\beta)[(1-k\alpha)^2 + 2(1-\alpha)(1-\beta)(1-k\alpha)k\alpha]}$$