

発表種別：short 発表

## 複数エージェントによる大規模構造体の故障診断

東京工業大学 服部聖彦、高玉圭樹

東京工業大学 総合理工学研究科 知能システム科学専攻

〒226-8502 神奈川県 横浜市 緑区 長津田町 4259

Tel:(045)924-5204

[hattori@cas.dis.titech.ac.jp](mailto:hattori@cas.dis.titech.ac.jp)

[keiki@dis.titech.ac.jp](mailto:keiki@dis.titech.ac.jp)

<http://www.cas.dis.titech.ac.jp/>

### 1. はじめに

近い将来、宇宙空間では国際協カステーションのような人が常時滞在可能なサイズのステーションや太陽発電衛星[1]等の大規模構造体が複数作られる可能性がある。宇宙空間では、漂う隕石やチリ、軌道上に多数存在するデブリ、強烈な放射線によりそれらの構造体が損傷を受ける確率が高く、絶えずメンテナンスが必要になると考えられる。大規模構造体において、故障の発生から機能の回復までの流れを分割すると次に示す6つのフェーズに分けられる。(1)故障の発生、(2)故障箇所の発見、(3)故障状況の確認、(4)修復方法の検討、(5)修復、(6)修復結果の確認。しかしながらこれらの作業をすべて人手で行うことは難しく、何らかの代替手段が必要になると考えられる。本研究では上で示したフェーズのうち、(2)故障箇所の発見、(3)状況の確認の2フェーズの作業を構造体の各パーツをエージェントと見立て、複数エージェント間の相互作用により行う方法について述べる。

### 2. 大規模構造体における故障診断の難しさ

宇宙空間における大規模構造体の故障診断では下記のような様々な問題が考えられる。

・環境から受ける制約

- (1) 人間の活動時間、空間の制約
- (2) 人間の同時活動可能人数の制約
- (3) 即時対応の難しさ

そこで、これらの問題を解決するために構造体の各モジュールにセンサを付け、そのセンサ値をモニタすることによって故障を診断することが望ましいと考える。しかしながら、構造体が大規模になると以下のような問題が起きると考えられる。

・規模から受ける制約

- (1) コンピュータの処理能力制約 (Single Event Upset)[2]
- (2) 情報の時間遅延

### 3. 各モジュールの相互作用により故障診断

上記で述べた規模から受ける制約を回避するために、各センサ値を中央に集めて集中処理をするのではなく、各モジュールごとに分散して処理し、処理結果のみを中央に送信することが望ましいと考える。その理由を以下に示す。

- (1) 分散処理することによって、低性能のCPUでも処理が可能である。
- (2) モジュール単位で処理をするので、時間遅延がさほど問題にならない。

- (3) モジュール単位で処理し、結果のみを中央に送るので、情報量を削減できるとともに、伝送エラーを最少にできる。

#### 4. シミュレーションルール

2次元の格子状の構造体を想定し、各構造体の接合部分であるノードモジュールをエージェントと見立て、以下のルールに従ってシミュレーションを行った。

- (1) 各エージェントは隣接しているエージェントと一定間隔で通信、センサ値を取得。
- (2) デブリの接触により、状況(センサ値)が変化。
- (3) 隣接するエージェントに対して現在取得した値と前回所得した値と比較。その差がセンサの誤差範囲内ならば正常に動作中と判定。
- (4) 比較した差が誤差範囲外ならば、機能はしているが、故障中と判定。
- (5) 値を全く取得できない場合は、そのエージェントが全く機能していないと判定。

#### 5. 結果

上記で述べたルールのもとにシミュレーションした結果を以下に示す。

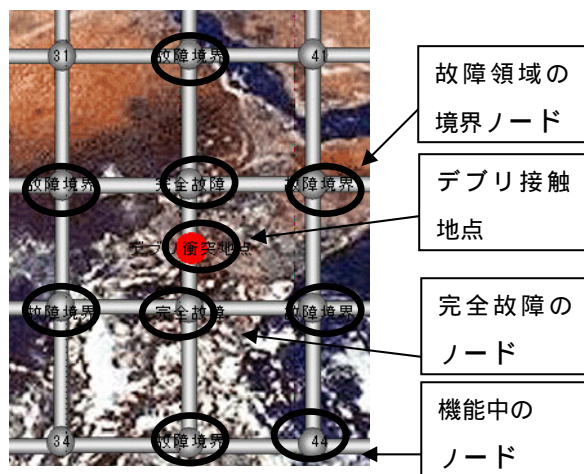


fig. 1 全く機能していないノード(完全故障)が存在する場合

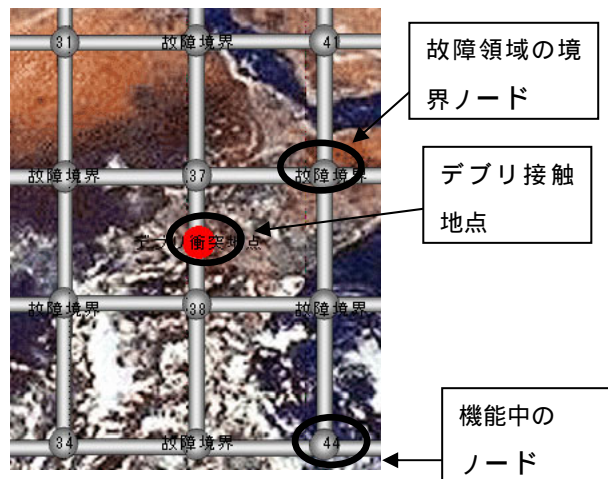


fig. 2 故障中のノードが存在する場合

#### 6. まとめ

シミュレーションの結果、全く機能していないノードモジュールのみが存在する時の領域推定はうまく行えるが、故障中のノードモジュールが存在する時は領域の推定がうまく行えなかった。これは、故障が起きたノードのセンサ値の出力として正常値と異常値とが時間ごとに確率的に出るようなルールにした結果、今回取得した値と前回取得した値の差が時間によって誤差範囲内の場合と、範囲外の場合があるため、故障の認識が安定してできなかったからだと考える。今後は故障の判定方法を変更するとともに、より現実的なモデルになるように修正を行う。

#### 参考文献

- [1]<http://giken.tksc.nasda.go.jp/seika/gaiyou/H13/16files/>
- [2]<http://www.index.isas.ac.jp/System/ComputerSoftware.htm>