

## IoT システムにおけるセンサエラー時戦略を考慮した 時間制約の評価手法

武田雅大<sup>†1</sup>  
早稲田大学

高橋竜一<sup>†2</sup>  
茨城大学

深澤良彰<sup>†3</sup>  
早稲田大学

### 1. はじめに

近年, IoT(Internet of Things)が身近なものとなっている。IoT システムは, リアルタイムにセンサから情報を取得し, 得た情報を元にシステムを制御している。IoT システムの導入により, 利用者の利便性の向上, 事業の効率化やコスト削減, また新しいビジネスの創出など様々な利点がある。このような IoT システムの恩恵を最大限受けるためには, リアルタイム性を確保することが重要である。そのために, 設計段階から時間制約を意識し, 評価することが重要である。

しかし, センサは一定の確率で異常値を取得してしまう。そこで, 一つのセンサが異常値を取得した場合には, そのセンサを再起動させることや, 別のセンサに切り替えるといったことでリカバリーすることが必要になってくる。このように, センサにエラーが起こった際に, どのようにリカバリーするかという戦略を予め立てておくことが, リアルタイム性を確保する上で必要である。

そこで本研究では, 複数のセンサで多重化されたシステムの時間制約充足を, センサエラーが起きた際の戦略を考慮した上で, 評価する手法を提案する。

### 2. 先行研究

高橋の研究[1]では, センサを複数組み合わせるようなIoT システムをモデル化した際に, 故障やエラー発生が想定される, 多重化された IoT システムをモデリングする方法と, 記述されたシステムに対して, 設計段階においてシステムの機能が達成される確率を, 確率的モデル検査ツール PRISM を使用し検証する手法を提案している。IoT システムを評価する方法として3つのサイクルステップを考える。

- Software Product Line(SPL)をもとにセンサー構成木を提案する
- センサー構成木を PRISM モデルへ自動変換する
- センサーの動作環境を考慮した上で機能の達成確率を評価する

しかし, この手法ではシステムにおける時間制約を評価することはできない。したがって, 本研究では時間概念を導入するために拡張を行った。

### 3. 提案

本研究では, 先行研究から時間概念を導入するために, 以下の拡張を行った。

- センサ構成木に対し, それぞれの機能が並列的に実行されるのか逐次的に実行されるのかを区別できるようにする。
- センサが情報の取得をする際にエラーが発生した場合, エラー時の動作として, 再取得及び再起動の動作定義と, それに伴う時間コストの情報をモデルに導入する。
- 特定のセンサが故障した場合に, 同種の代用可能なセンサへの切り替え動作を定義する。

一つ目は, ある高次の機能を達成するために, 複数の低次の機能を達成することが必要な場合, その実行方法(並列実行または逐次実行)によって, 全体の必要とする時間が異なる。モデル上にそれらの実行方法の区別をつけられるようにするとともに, 並列実行では低次機能の中で最大の実行時間を, 逐次実行では低次機能の実行時間の和を, 高次機能の達成の所要時間として評価する。また, センサ構成木上では図1のように表記する。

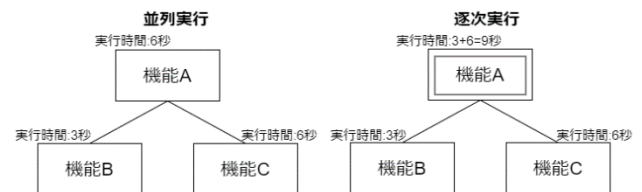


図1 並列実行(左)と逐次実行(右)の表記

Fig.1 Description of parallel (left) and sequential(right) execution

二つ目は, 特定のセンサが動作不良を起こした場合, 情報の再取得に関する戦略を選択できるようにする。具体的にはセンサをそのままに再度情報取得を試み, それでも正常に動作しない場合, 再起動を行う。当然再起動をすれば実行時間が余計に必要な。再取得及び再起動に関して, それぞれ何回を上限として動作するのかを定義可能にする。

三つ目は, 特定のセンサが故障して動かなくなった場合, 別のセンサに切り替えられるようにする。ただし, 切り替えるには別のセンサがあることが条件になる。また, あるセンサが故障した際にどのセンサを使用して代用するのかということ, あらかじめ決めておく必要がある。

さらに, 一つのセンサが故障した際に複数のセンサでリ

カバリーする場合もある。このような場合は複数のセンサをセンサ郡とみなして記述する。以下にセンサ構成木での記述の例を挙げる。

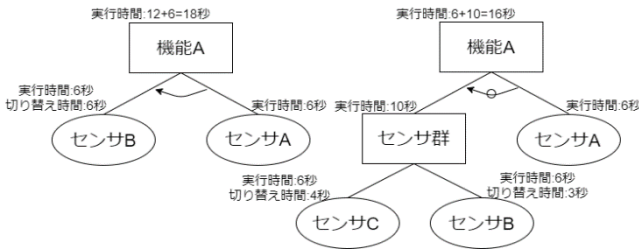


図2 切り替えの表記  
Fig.2 Description of switch

図2の左図では、センサAが故障した際、センサBに切り替えることを表記している。切り替えを行う際は、センサの実行時間だけでなく、切り替えにかかる時間も考慮する。右図では、センサAが故障した際、センサBとCの両方を使用してリカバリーすることを記述している。また、切り替えする際に、BとCは元々電源を入れていたため、切り替え時間が左図よりも少なくなっている。センサを切り替える際、一から立ち上げを行うのか、元々電源を入れているのかを、矢印途中の円の有無で区別するように記述する。

4. 評価

ケーススタディとして、あるロボットが特定の場所に移動することを想定する。今回はその中の人物感知に焦点を当てて考える。

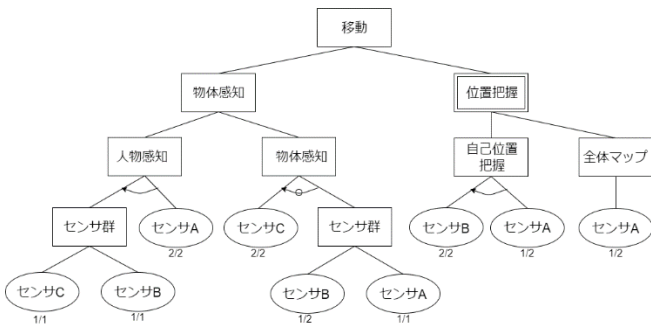


図3 移動のセンサ構成木  
Fig.3 Sensor configuration tree of movement

人物感知のセンサーは A,B,C の三種類あるとする。メインではAのセンサ二つで稼働しているが、一つ故障してしまった場合はBとCのセンサ両方を使用してリカバリーすることにする。つまり、人物感知という機能を達成するため

- Aのセンサ二つが両方成功する
- Aのセンサ一つとBとCの両方のセンサが成功するという二種類の達成方法がある。以下に各センサの情報を表に示す。

表1 センサ情報

Table 1 Information of sensors

センサ名(個数)	検知確率	取得時間	再起動時間	切り替え時間
A(2)	0.8	4	7	14
B(1)	0.7	4	8	10
C(1)	0.9	5	8	20

そして、これらのセンサに対して

- 再取得は四回まで試みる
  - 二連続で失敗した際は再起動する
- といった戦略を立てる。

さらに、人物感知が15秒以内に95%以上の確率で達成しなければならないとする。この制約を検証するために、人物感知に15秒以上要するという違反式をLTL式で記述し、PRISMモデル検査器に入力すると以下のような結果が得られる。

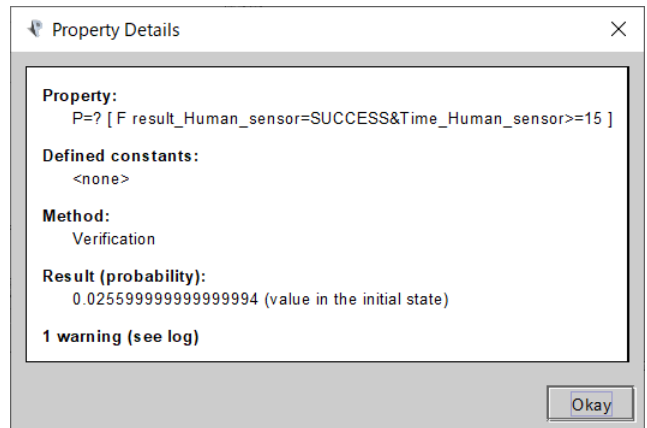


図4 人間感知の評価

Fig.4 Evaluation of human detection

得られた結果から、人物感知が15秒以上かけて達成される確率が約2.5%であると分かった。そのため、人物感知が15秒以内に達成される確率は約97.5%であると分かる。このように、設計段階で時間制約を守られていることが検証でき、設計段階で時間制約を評価することができるようになる。

5. おわりに

今回、複数のセンサで多重化されたシステムの時間制約充足を、センサエラーが起きた際の戦略を考慮した上で、評価する手法を提案した。今後の課題は、システム全体の中でどこがボトルネックになっているのかを検知することで、修正が必要な個所をわかりやすくすることが挙げられる。

参考文献

[1] Ryuichi Takahashi, "Evaluation the Redundancy of the IoT System Based on Individual Sensing Probability", IEICE TRANS. INF. & SYST., VOL. E103D, NO. 8, pp. 1783-1793, 2020.