

# 防災情報の流通ネットワークの検証を目的とした CyReal センサシステムの開発

## A CyReal Approach to Sensor System Development for Preliminary Verification of Disaster Information Network

廣井慧・小比賀亮仁<sup>(1)</sup>・篠田陽一<sup>(1)</sup>

Kei HIROI, Akihito KOHIGA<sup>(1)</sup> and Yoichi SHINODA<sup>(1)</sup>

(1) 北陸先端科学技術大学院大学

(1) Japan Advanced Institute of Science and Technology, Japan

### Synopsis

IoT devices are expected to enable inexpensive and easy measurement and collection of a wide range of environmental information, especially in the field of disaster prevention. Whereas, preliminary verification is difficult, because of a functional design that assumes their distributed deployment, the cost of development itself. Therefore, we develop a sensor system emulator with CyReal concept, which integrates the virtual device with the actual device, and federated with other simulators. This paper presents a prototype of our sensor system, its feasible performance, and discusses a design for flexible integration, in order to support the development, update, debugging, and operation of sensor systems as a distribution network for disaster prevention information.

**キーワード:** センサネットワーク, 防災情報通信, エミュレータ, IoTデバイス

**Keywords:** sensor network, disaster information network, emulator, IoT device

## 1. はじめに

センサを搭載したIoT (Internet of Things) デバイスおよびその関連技術は、広範囲の環境情報の安価かつ容易な計測、収集が可能になると期待される。防災分野においては河川観測や傾斜計測などに導入されつつあり、これまで状態把握が困難であった地点でのデータ収集に役立つと考えられる。災害対応のための従来の環境観測は、特に監視が必要な少数の拠点に、設置されることが一般的であり、専用のネットワークを用いた強固な監視体制ネットワークが利用されてきた。

近年のIoT, 通信技術の発展により、導入・運用のコストが大幅に削減でき、これまで監視の行き届か

なかつた地点まで、より密で大規模な観測体制が容易に適用できるようになってきたものの、大量のセンサデバイスの設置が可能となることで新たな問題が生じる可能性がある。IoTデバイスは市街地や山間部などで使用されることが想定されており、大量のデバイスを設置できることに、観測網としての利点がある。ところが、大量のデバイスを広範囲に設置することで、その扱いは煩雑になり設置後の再配置を伴う改修が困難になる。特にデバイス同士の関係性が重要となるデータ計測では、分散配置を想定した機能設計や活用方法の検討が重要となるが、大量のデバイスが必要であるために、必要数の実機を用意した事前の機能検証が難しく、開発自体や再設置などの改修のコストが高くなる。

本稿で述べる事前の機能検証とは、設置前にセンサネットワーク全体の動作を検証し、ボトルネックを見つけ、その機能設計に基づいて解決策を検討することを指す。センサ機器を設置する前に事前検証を行うことで、センサ機器の固有の特性や複数のセンサ機器からのデータの関係性、地形や通信インフラによるデータの伝送特性などを明らかにし、その結果を利用目的に応じた計測設計に役立てることができる。しかしそのためには、多数の機器が必要となるため、必要な数の実機を用意して事前に機能を検証することは難しく、開発自体や移設などの改善にかかるコストが増大する。こうした事前検証を可能にするため、各種センサエミュレータが開発されているものの、仮想デバイスを用いたうでのネットワーク性能やデバイス性能など一部の機能の検証に留まっている。

本稿では、防災情報の収集・運用の研究・開発・運用を支援し、特定の設置環境や運用形態を想定した事前検証を可能にするセンサエミュレータシステムを開発する。センサエミュレータとは、センサデバイスに関わるコンピュータやその機能を模倣する仮想化技術であり、一般的には、仮想センサデバイスを用いた機能検証に用いられる。本稿では、防災情報の流通ネットワークとして、センサシステムの開発、更新、デバック、運用を支援する目的で、仮想機と実機が交換可能かつ他シミュレータと連携可能なエミュレータを開発する。

大規模なセンサデバイスを想定した機能検証を可能にするため、センサデバイスのうち、ノードとセンサを分離しそれぞれに対して仮想機、実機の接続機構を取り入れる。仮想機だけでなく実機との接続、検証を可能とすることで、仮想機だけでは実証できない機能や通信環境を想定できるほか、クラウドとの接続も容易になり、実際の運用環境を前提とした運用コストの削減や資源の多重利用が可能となる。他シミュレータとの連携は、IoT連携基盤を用いたデータ交換によって連携可能とする。連携により、センサシステム内に実運用を想定したデータを組み込み、さらにデータ収集を前提とした災害対応など運用形態まで含めた事前検証が可能となる。

## 2. 関連研究

センサエミュレータは研究者や開発者、利用者がセンサネットワークのデザインを検討するツールであり、センサネットワークに関連する様々な技術(通信プロトコル、コンピュータプロセス、アプリケーションソフトウェア)に対して機能検証を目的として多くのエミュレータが開発されてきた。特に近年

では、IoTの普及に伴うセンサ台数の増加によって、大規模な数のセンサを扱う必要が出たために、数多くのセンサ仮想化技術が開発されている。SenaAS (Sarfraz et al., 2010) は、コネクテッド・オブジェクトに対してイベント処理をサポートするIoT仮想化フレームワークであり、基盤となるIoTクラウドのセマンティックオーバーレイによるオントロジデザインや、セマンティックルールによるポリシベースのサービスアクセス機構を提供する。Bose et al. (2015) は、様々な分野のアプリケーションを開発するために、センサを仮想化したクラウド上で、センサレベルでのリソース抽象化を行った。これは、クラウド環境でのセンサ仮想化のデザインに活用される。SenseWrap (Evensen et al., 2009) は、物理的なセンサの代わりに仮想的なセンサを提供するミドルウェアであり、アプリケーションがセンサに固有の情報を処理することなく使用できる通信インタフェースを提供する。

センサネットワークに欠かせない技術である、ワイヤレスネットワークも、エミュレーションの対象となる。TOSSIM (Levis et al., 2003) は、ネットワークの相互作用をシミュレートすることに重点を置いており、数千のノードに拡張しながらネットワークの挙動を捉える高いスケーラビリティを特徴としている。COOJA (Osterlind et al., 2006) は、クロスレベルシミュレーションを目的としたセンサネットワークエミュレータで、センサノードプラットフォーム、オペレーティングシステム、ソフトウェア、無線伝送モデルなど、システムの多くのレベルで同時にシミュレーションできる。無線センサネットワークを前提にしたエミュレータは、EmStar (Girod et al., 2007), Avrora (Titzer et al., 2005), J-Sim (Sobeih et al., 2006) など数多くが研究開発されている。最近では、LPWA通信の方式のひとつであるLong Range (LoRa)の台頭など、低消費電力の広域ネットワークの発展を受けて、LoRa Coverage Emulator (Homssi et al., 2021)が登場した。このLoRaエミュレータは、送信機と受信機で構成され、LoRaネットワークデザインフレームワークに基づいて、信頼性の高いネットワークカバレッジを推定する。

ハードウェア、ソフトウェアレベルでは、ノード用のソフトウェアを開発するためのエミュレータFreemote Emulator (Maret et al., 2008)が挙げられる。これは、物理層、データリンク層、ルーティング層、アプリケーション層でのシステム・アーキテクチャを開発者に提供する。ATEMU (Polley et al., 2004) は、異なるアプリケーション/ハードウェアプラットフォーム上で動作することができ、異種センサーネットワークをシミュレートできる。これは、プロセ

ッサ、無線インターフェース、タイマー、LED、その他のデバイスをエミュレートが対象となる。

Kasprowicz et al. (2007) は、デバイスとしてCCDセンサに着目し、組み立て後のテストやデバックの高速化、効率化を狙ったハードウェアエミュレータを開発した。また、エミュレータの研究も、軽量かつ小型のIoTシステムを重視した、コモディティ化が着目されている。Brady et al. (2017) は、QEMUシステムエミュレータを用いてIoT環境用のエミュレータを開発し、エミュレートされたRaspberry Piデバイスを相互に接続したテストベッドを構築している。

以上に挙げたようなエミュレータを利用することで、ハードウェア、ソフトウェア、ネットワークを統合的にエミュレートすることが可能となった。しかし、IoTの発展とともに、電力供給や異種のセンサデバイス、ネットワークデバイスを使いこなす必要が出てきている。これまでのエミュレータは、コンピュータのある機能に特化したエミュレータであり、幾つかのエミュレータを利用することで個々の技術についての機能検証は可能である。しかし、アプリケーション・サービスの利用を業務とする実際の運用者にとって重要となる、電力供給、通信状況まで踏まえたシステム全体としての動作を検証することは難しい。また、利用者にとっては、地理空間や業務の効率化を考慮した分散設置など活用を鑑みた検証を考える際、これまでのエミュレータではこれらの要件が検討されていない。そこで本稿は、このような要求を満たすセンサシステムの開発に向けて、「センサシステムの要件を実システムとの互換性があること」、「多数のセンサを扱うことができること」と定義し、CyRealの概念を取り入れたエミュレータのプロトタイプを開発する。

### 3. CyRealによるセンサシステムの設計

#### 3.1 概要

本稿で開発したセンサシステムは、大規模センサシステムの開発、更新、デバック、運用を支援する目的で、CyRealのコンセプトに基づき、仮想機と実機が交換可能かつ他シミュレータと連携可能な設計とした。CyRealは、CyberとRealの混在として、デジタルツインの概念の中間的な役割を果たす概念であり、コンピュータシステム、人、環境などの実物と仮想物を柔軟に入れ替え、統合する。Fig. 1の左側の構成は、すべてのサブシステムがシミュレータによって構成され、そのサブシステムが連携基盤を介し

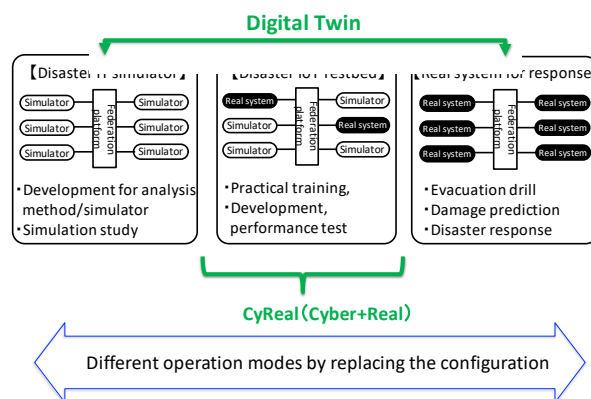


Fig. 1 CyReal Approach for Sensor System

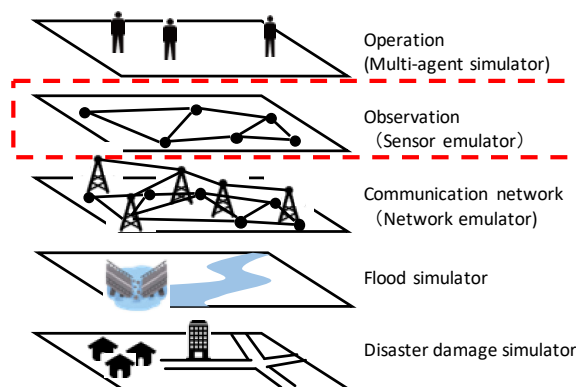


Fig. 2 Federation with Simulators for External Environment

て接続されている構成となる。この構成中の様々なシミュレーションやシステムは、すべて実物に置き換えることを可能とする。Fig. 1の右側の構成は、人や自然現象、実際の防災情報システムなど、サブシステムがすべて実在し、実際に運用されている状況を示している。この二つは、デジタルツインの関係にあり、デジタルツインの概念をさらに拡張して、サブシステムの中で実システムとシミュレータが混在できるようにセンサシステムを設計した (Fig. 1中央の構成)。

#### 3.2 CyRealセンサシステムの特徴

本稿のセンサシステムは、CyRealのコンセプトに沿って構成される。センサは、Fig. 1のサブシステムとして接続されており、実在のものや仮想のものに柔軟に置き換えることができる。このCyReal化により、サブシステムが様々な形で動作し、置き換えるシステムに応じてシステム全体が多様な目的で使用されることを期待して、このような構成とした。

システム全体がシミュレータで構成されている場合は、防災ITシミュレータとして機能する。例えば、仮想のセンサシステムと新たな解析技術やシミュレータを接続することで、すべてのシミュレータを過去の災害のデータに基づいて動かすことができる。データの収集が困難であるため、これまで行うことが難しかった解析技術やシミュレータの性能評価について、容易に実施可能な評価環境を提供し、開発を促進できる見込みがある。また、センサシステムを実物に置き換えると、そのセンサシステムの研究開発や性能評価に用いることのできるテストベッドとして動作する。

この構成を採用することで、本論文の要件である、検証できるデバイス、シミュレータ、システムの制限をなくすことが可能になる。センサデバイスだけでなく、地理空間やオペレーションに関するシミュレーションを接続し、実際の運用上での機能と効果検証を行うことも目的としている。

### 3.3 CyRealセンサシステムの活用目的

このセンサシステムは、単なるセンサの仮想化ではなく、実際にセンサがデータを利用することを想定して設計している。従来のセンサ仮想化では、センシングに関わるデータや機能のシミュレートを目的としていた。本稿のセンサシステムは、データやデバイスを模擬するだけでなく、センサデバイスの設置環境をインスタンスとして組み込めるよう開発している。

本センサシステムが見込んでいる活用を以下にあげる。大規模なセンサ配置を構成すると、いくつかのセンサでデータ欠損が生じる。データだけを模擬するシステムでは、このようなデータ欠損の扱いを表現することは難しかった。本センサシステムでは、センサデバイス自体に加え、無線環境やセンサが設置されている場所の状況といった外部環境も取り入れてデータを生成する。そのため、外部環境の影響によるデータの揺らぎを表現することが可能となる。

さらに、オペレーションに関するシミュレータや自然現象のシミュレータなど外部のシミュレータと協調動作したデータの多様性も表現できる (Fig. 2)。これが、実機と仮想ノード、センサを置き換え可能とし、統合するエミュレータの利点となる。このようなセンサシステムはオペレーションを検証するうえで多くの用途が見込まれるが、同時に構築にあたっては多くの課題がある。本稿では、このコンセプト

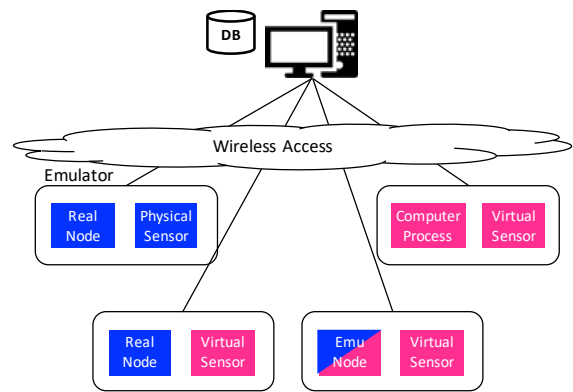


Fig. 3 Structure of CyReal Sensor System

トに従ってセンサシステムを設計し、仮想センサを大規模に動作させたときの性能を調べるとともに、実装に向けた課題を整理する。

### 3.4 CyRealセンサシステムの設計

CyRealの構想に基づき、仮想デバイスだけでなく実物のデバイスとの接続、検証ができるセンサシステムを開発する。センサエミュレータは、ネットワーク性能やデバイス性能、コンピュータの処理性能などセンサに関するシステムの事前検証を可能にする仮想化技術の一種であり、従来は仮想センサデバイスを使った機能検証などに用いられてきた。これまでの既存技術、先行研究は仮想センサデバイスのみに対応しているため、実際の運用環境を前提とした検証が行いにくいという問題があった。近年のIoTデバイス・関連技術では、センサの広範囲な分散配置や、安価かつ容易な計測、収集が可能になると期待されている。しかし、実際には、分散配置を想定した機能設計、必要数の実機を用意した事前の機能検証は難しく、利用する側の判断に任されてしまうという欠点がある。

そこで、防災情報の収集、流通に関する研究開発や運用を支援する目的で、CyRealセンサシステムの開発を試みた。Fig. 3中の、Sensorは計測デバイスを指しており、Nodeは計測データの処理や通信を行うコンピュータを示す。ここでは、Sensor、Nodeともに実機、仮想機と入れ替え可能な設計とした。さらに、他シミュレータと連携可能とし、過去の災害のデータをもとに停電やデバイスの故障、ネットワークの断絶など実際の状況を鑑みた検証まで拡張することを目指している。

## 4. センサシステムの実装と動作試験

### 4.1 概要

Fig. 3の設計に基づき、本センサシステムのプロトタイプを実装した。本センサシステムの実装には2つの特徴があげられる。1点目は、既存のIT技術を活用していることで、商用のIoTクラウドを活用して、サーバを自前で一切用意せずデータの集約、処理を可能にする。メンテナンス、維持管理する必要がなくなり、導入と運用に関するコストを最小化できる。

2点目がAPI統一であり、仕様が公開されているクラウドを使うことで、既にある機能を活用でき、オープンシステムとして他者がこの仕組みを活用することができる。既にある機能を活用することでシステムとしての開発も簡略化でき、ストレージや出力先のアプリケーションを容易に変更可能となる。

### 4.2 実装

Fig. 4に本稿で実装したセンサエミュレータの概念実証の構成を示す。Real Nodeからは、Sigfoxを使って計測データを集約し、AWSクラウドへ集約した。Real Nodeのハードウェアとして、Arduino Unoを使用しており、Arduino UnoからSigfox Una shieldを介してこれらのデータをAWS API gatewayに送信する。

本稿で使用したハードウェアはArduino Unoに限らず、他のデバイスから得られるデータを接続する可換性も備えている。可換性を示すために、もうひとつのReal NodeとしてRaspberry piを使用した。Raspberry piのデータはVirtual Sensorのデータとして構成し、この概念実証では、気象情報APIである、Open Weather Mapからデータを取得した。取得にはNode-Redを利用しFig. 4のようなフローでAWS API gatewayにデータを送信する。

AWSはServerless serviceとして利用しており、AWS API gatewayに送信された、Physical/Virtual Sensorのデータは、AWS Lambdaを介してSlack webhook APIに送信される。Fig. 4は、出力先の例としてSlackに送信され、記録されたデータを示している。

さらに、Virtual Nodeとして、コンピュータプロセスとVirtual Sensorの組み合わせを作成した。Virtual Sensorによって生成したセンサデータはComputer Processを介して、AWSに送信される。この機能を利用すれば、インターネット上に公開されたオープンな気象情報をセンサデータとして取り入れることが可能であるうえ、降雨や氾濫予測のシミュレーションで生成されたセンサデータをシステム全体のデータとして利用することが可能となる。この構成をもって、Real/Virtual Node、Physical/Virtual Sensorが混在するCyRealの環境を実装した。

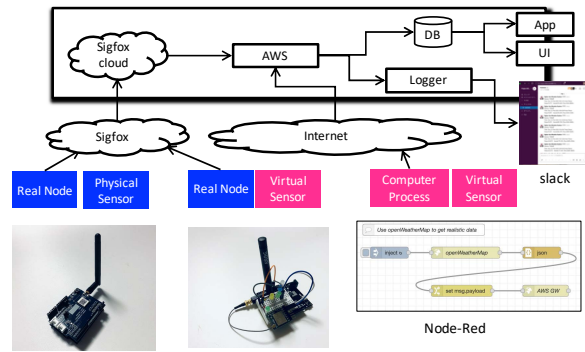


Fig. 4 Prototype Configuration

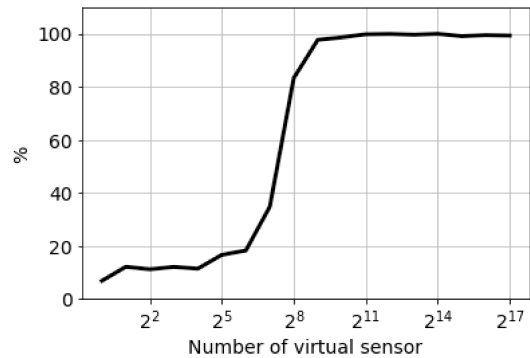


Fig. 5 Performance of Sensor System (CPU Usage)

### 4.3 動作試験

開発したCyRealセンサシステムのプロトタイプがエミュレータとして機能することを示すため動作試験を行う。はじめに、仮想センサ大規模設置時の動作性能を調べ、得られた結果から、本センサシステムで検証できるデバイス、シミュレータ、システムの制限と、実際のオペレーションを踏まえた検討を行う。

#### (1) 実験環境

性能評価には、Pythonで実装したコンピュータプロセスと仮想センサを使用した。近年の観測機器の設置状況から数千台規模を想定し、この仮想センサを2<sup>n</sup>台(n=0~17)の規模で構成し、MQTTにてデータを送信した。実験では、エミュレーションのためにすべての仮想センサは1秒おきに100回データ送信を行う。

はじめに、一般的な性能のPCでこのCyRealセンサシステムをエミュレートさせた際の、プロセッサの使用率を算出する。これは、本センサシステムがどの程度大規模な仮想センサに対応できるかを調べることを目的としている。次に、本稿での要件が実装されたかを考察し、さらに今後の実用化に向けた検

討を行う。プロセスおよび仮想センサは、MacOS 10.15.7, 64ビット, 2.4 GHzのIntel Core i9 (4コア/8スレッド), 64 GBのRAMを搭載したコンピュータで実行した。

## (2) 結果と考察

仮想センサ大規模設置時のパフォーマンスについて、得られた結果をFig. 5に示す。この図は仮想センサの台数を $2^n$ とし、 $n=0\sim 17$ のCPU使用率の最大値を表す。センサ台数が $2^0$ 台のとき、CPU使用率は6.8%で、送信にかかる時間は約90秒となった。センサ台数を $2^9$ 台としたとき、CPU使用率は97.64%に達した。この際の送信にかかる処理時間は119秒となった。その後、 $n=10\sim 17$ では、98.58～99.95%となり、ほぼ100%のCPU使用率となった。また処理時間は、 $2^{11}$ 台までは120秒程度だったが、台数を増やすにつれ徐々に処理時間が増大し、 $2^{12}$ のとき212秒となった。さらに、 $2^{14}$ のとき1,071秒、 $2^{16}$ のとき2,528秒であり、台数の増加とともに処理時間は増大した。

このようなセンサシステムの災害時利用を考えたとき、事前検証とリアルタイムでデータの検証が必要なシナリオとの2つのパターンが考えられる。事前検証においては、時間の制約がないため $n=14$ 以上の処理速度でも実用可能であるが、リアルタイムでの利用として災害発生中にセンサデータの欠損やそれに関わる対応を検証する際には $n=12$ 程度の台数での使用が妥当であると考えられる。このパフォーマンスは利用するコンピュータによって異なるが、高性能なコンピュータを利用することの難しいケースにおいて、この結果は活用のシナリオを検討するのに役立つ。

装置やシミュレータ、システムの限界や、実運用に基づく議論として、センサシステムのエミュレータ機能要件として、「実システムとの互換性があること」「多数のセンサを扱えること」の2点がある。本実験では、一般的な利用環境で大規模な数のセンサをエミュレーションできる可能性とその想定される台数を示した。既に運用されている実センサデバイスとの親和性については、Fig. 4に示す構成でプロトタイプを実装したことにより果たしている。この構成では、Real Node/Physical Sensor, Real Node/Virtual Sensor, Computer Process/Virtual Sensorの3種類のサブシステムを統合させた。

要件である統合は達成したため、今後はセンサデバイスやコミュニケーションポートの種類をより拡張させていく。実センサに関して今回はいくつかのReal NodeとComputer Processしか接続していないが、Nodeをエミュレーションすることによって、より多様な種類のデバイスをテストすることが可能になる。

次に重要となるのがCyRealの程度とソフトウェア透過性となる。CyRealの程度について、本稿では仮想センサを組み込んでいるが、仮想の度合い、CyRealの統合に関する度合いを検討する必要がある。センサがもつ、ハードウェア、アルゴリズム、データ、そしてその活用（データ欠損やヘテロジニアスなデータの統合）など機能ごとに仮想化させ、ノード、センサ単位でなく、より機能に特化した仮想化とその統合を行う。これにより、大規模なIoT環境において、より現実的なエミュレーションが行えると期待できる。

ソフトウェア透過性について、本センサシステムは大規模なIoT環境を想定しているが、ノードやデバイス、システムの機種に依存せず、共通的に利用できるソフトウェアを検討することでこのような大規模な数のセンサにも対応しやすいエミュレータとなる。さらに、今後は具体的なユースケースを想定した他シミュレータを構築する。例えば、災害現象を再現するシミュレータや避難などの行動を再現するエージェントシミュレータ、ネットワークエミュレータと他の様々なセンサシステムエミュレータなどを通じて、目的であるセンサ最適設置や開発支援などの機能を作り込んでいく。

## 5. 結論

本稿では、防災情報の収集・運用に関する研究・開発・運用を支援し、特定の設置環境・運用形態を想定した事前検証を可能にすることを目的として、大規模なIoT環境の構築や適切な事前検証を可能にするセンサシステムのエミュレート環境を開発、設計した。本センサシステムは、機能設計を前提としてセンサネットワーク全体の動作を設置前に試験し、ボトルネックを見つけ解消への手立てを検討するための事前検証を目的としている。センサデバイス固有の特性によるデータや複数センサデバイス間のデータの関係性、地形や通信インフラによるデータの伝送特性などを明らかにし、活用を鑑みた検証を想定している。CyRealのコンセプトをベースにして、実システムと仮想システムを統合したセンサシステムのプロトタイプを構築した。プロトタイプの実験結果は、一般的な利用環境での大規模化の目安となる台数を示した。これは現在、災害対応に活用されているセンサ台数から鑑みると十分な台数のセンサを本センサシステム上で動作させることが可能となる。実験結果から、本センサシステムにおいて、Physical/VirtualのSensor/Nodeが統合可能なこと、一般的な計算環境においてのバーチャルセンサのパフォーマンスを示した。

今後は、CyRealの程度として、ハードウェア、アルゴリズム、データといったいくつかのレイヤで、Real/Virtualの選択的利用を可能とするシステムへの拡張を目指す。このことで、データ欠損のようなより実態に沿った検証をはじめとして、機能、活用の両面からIoTシステムの事前検証を行うことが可能となる。今後、本CyRealセンサシステムを利用して機能設計に役立てるためには、通信状況や電力消費、設置環境など様々な条件を機能追加し、そのうえでこれらの機能もPhysical/Virtualを統合させる必要がある。通信状況だけを考えても、多様なインフラが利用できるため、これらの機能をセンサデバイス上に作り込んでいくことは非常に複雑な構成を必要とする。そのため、センサデバイスとは別のレイヤに通信や電力のCyRealを可能とするシミュレーションレイヤを作り、これらのシミュレーション同士を連携させることで、この問題を解決する。さらに、他のシミュレータとMQTTベースの連携により、センサ設置の地理的特性や活用における観点から分散設置に関する検証環境を構築していく。

## 謝 辞

本研究は、JSTさきがけJPMJPR2036の支援を受けたものである。

## 参考文献

- Abrishambaf, O., Faria, P., Vale, Z. (2020): Laboratory Emulation of Energy Scheduling in an Agriculture System. In 2020 IEEE/PES Transmission and Distribution Conference and Exposition (T&D), pp. 1-5.
- Boschert, S., Roland, R. (2016): Digital Twin—The Simulation Aspect, Mechatronic futures, Springer, pp. 59-74.
- Bose, S., Gupta, A., Adhikary, S., Mukherjee, N. (2015): Towards a Sensor-cloud Infrastructure with Sensor Virtualization, In the Second Workshop on Mobile Sensing, Computing and Communication, pp. 25-30.
- Brady, S., Hava, A., Perry, P., Murphy, J., Magoni, D., Portillo-Dominguez A. O. (2017): Towards an Emulated IoT Test Environment for Anomaly Detection using NEMU, In 2017 Global Internet of Things Summit (GIoTS), pp. 1-6.
- Debnath, H., Gehani, N., Ding, X., Curtmola, R., Borcea, C. (2018): Sentio: Distributed Sensor Virtualization for Mobile Apps, In 2018 IEEE International Conference on Pervasive Computing and Communications (PerCom), pp. 1-9.
- Deda, S., Eder, A., Mhetre, V., Kuchling, A., Greul, R., Koenig, O. (2020): Designing a Battery Emulator/Tester from Scratch to Prototyping to Automated Testing within a HIL Digital Twin Environment, In International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management, pp. 1-8.
- Evensen, P., Hein, M. (2009): SenseWrap: A Service Oriented Middleware with Sensor Virtualization and Self-configuration, In 2009 IEEE International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), pp. 261-266.
- Evensen, P., Hein, M. (2009): Sensor Virtualization with Self-configuration and Flexible Interactions, In the 3rd ACM International Workshop on Context-Awareness for Self-Managing Systems, pp. 31-38.
- Girod, L., Ramanathan, N., Elson, J., Stathopoulos, T., Lukac, M., Estrin, D. (2007): Emstar: A Software Environment for Developing and Deploying Heterogeneous Sensor-actuator Networks, In ACM Transactions on Sensor Networks (TOSN), 3(3), pp. 1-34.
- Homssi Al B., Dakic, k., Maselli, S., Wolf, H., Kandeepan, S., Al-Hourani, A. (2021): IoT Network Design Using Open-Source LoRa Coverage Emulator, IEEE Access, No.9, pp. 53636-53646.
- Kasprowicz, G., Mankiewicz, L., Pozniak, K. T., Romaniuk, R. S., Stankiewicz, S., Wrochna, G. (2007): Hardware Emulator of the High-resolution CCD Sensor for the Pi of the Sky Experiment, In Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2007, Vol.6937, pp. 693-708.
- Levis, P., Lee, N., Welsh, M., Culler, D. (2003): Tossim: Accurate and Scalable Simulation of Entire Tinyos Applications, In Computer Communications and Networks, International Conference on Embedded networked sensor systems, pp. 126-137.
- Maret, T., Kummer, R., Kropf, P., Wagen, J. F. (2008): Freemote Emulator: A Lightweight and

- Visual Java Emulator for WSN, In International Conference on Wired/Wireless Internet Communications, pp. 92-103.
- Osterlind, F., Dunkels, A., Eriksson, J., Finne, N., Voigt, T. (2006): Cross-level Sensor Network Simulation with Cooja. In 31st IEEE conference on local computer networks, pp. 641-648.
- Polley, J., Blazakis, D., McGee, J., Rusk, D., Baras, J. (2004): Atemu: a Fine-grained Sensor Network Simulator, In Sensor and Ad Hoc Communications and Networks, pp. 145-152.
- Sarfraz, A., Chowdhury, MR. M., Noll, J. (2010): Senaas: An Event-driven Sensor Virtualization Approach for Internet of Things Cloud, In 2010 IEEE International Conference on Networked Embedded Systems for Enterprise Applications, pp. 1-6.
- Sobeih, A., Hou, J. C., Kung, L., Li, N., Zhang, H., Chen, W., Tyan, H., Lim, H. (2006): J-Sim: A Simulation and Emulation Environment for Wireless Sensor Networks, IEEE Wireless Communications, 13(4), pp. 104-119.
- Titzer, B., Lee, D., Palsberg, J. (2005): Avrora: Scalable Sensor Network Simulation with Precise Timing, In IEEE Fourth International Conference on Information Processing in Sensor Networks (IPSN'05), pp. 477-482.
- Wang, Z., Liu, M., Zhang, S., Qiu, M. (2014): Sensor Virtualization for Underwater Event Detection, Journal of Systems Architecture, 60.8, pp. 619-629.
- Ye, Z., Hu, F., Zhang, L., Chu, Z., O'Neill, Z. (2020): A Low-Cost Experimental Testbed for Energy-Saving HVAC Control Based on Human Behavior Monitoring, International Journal of Cyber-Physical Systems (IJCPS), 2(1), pp. 33-55.

(論文受理日 : 2021年8月31日)