

# センサネットワークとビッグデータ解析を用いた応用技術開発

(組込みシステム技術と無線ネットワーク技術の応用)

基盤技術部 部長 藤本 和貴  
産業技術総合研究所 前田 龍太郎  
産業技術総合研究所 藤本 淳  
産業技術総合研究所 鈴木 章夫

本研究では、複数のセンサを相互に接続して計測を行うセンサネットワーク技術、組込み技術を用いたデータ収集システムを開発するとともに、収集したデータの解析を行うビッグデータ解析システムを用いた応用技術開発を行い、県内企業への技術移転を行うことを目的とした。

平成27年度はセンサネットワークの基礎となるセンサユニットとデータの収集を行うサーバユニットを試作し、クラウド上のサーバへの通信試験を行った。<sup>[1]</sup>

平成28年度はセンサネットワークの拡張、データ収集方法の改良、サーバシステムの改造、クラウド上のサーバへの通信試験を行った。

## 1. 緒言

組込み技術は、産業用・家庭用の電子制御・計測を行う機器の多くに用いられている技術であり、生産機械・工場、環境・エネルギー、スマートコミュニティ、医療・介護・見守り、防災など、応用分野が広く、地場産業が新産業へ進出するための基盤技術である。汎用コンピュータと異なり、ハードウェアとソフトウェアが一体となっており、その種類が非常に多いことが特徴である。

センサネットワーク技術は、計測を行うために多くのセンサを配置し、センサ相互に接続してデータの計測・収集を行う技術であり、各センサにおいてデータを計測するとともにセンサ間で自律的にネットワークを構成し、ネットワークの変更や障害に対応して通信環境を維持できることを特徴としている。製造現場や環境・エネルギー分野など広い分野での計測・制御における用途が見込まれて、HEMS (Home Energy Management System)、BEMS (Building Energy Management System)、SEMS (Small Office Energy Management System: 小規模EMS)、スマートグリッド、環境モニタリング、等に用いられる。その中でも特にM2M (Machine-to-Machine) 技術は、コンピュータネットワークに繋がれた機械同士が人間を介在せずに相互に直接情報交換し、自動的に最適な計測・制御が行われるシステムである。

ビッグデータ解析技術は近年急速に注目されている技術である。ビッグデータについては定量的な定義は

ないが、従来のデータベース管理ツールでは処理が困難なほど巨大なデータの集積を指すことが多く、収集・選択・保存・検索・解析・可視化が困難なデータを対象とする。予め有意性を限定してデータを収集・解析・検証していたこれまでの手法から、全件のデータを取得して、傾向・相関を把握することが可能になる。用途例としては、電力センサの計測値から機器の稼働状況の把握・異常の検知を行う、温度・湿度・照度・土壤水分等の環境データから農作物の育成監視・予測を行う、形状・振動等のデータから建築物・構造物の異常検知を行う (インフラマネジメント) 等が想定される。

ビッグデータ解析のためには、容量・処理能力等に考慮したハードウェア・ソフトウェアの構成が必要となり、ハードウェアとしては分散処理システム、クラウド・コンピューティングが、ソフトウェアとしては専用のオープンソースソフトウェア (Hadoop、MapReduce 等) が用いられることが多い。収集した多量のデータ間の関連付けを行い、データの可視化による、異常の予兆検知等に用いる。

いずれの技術も県内企業が関心を持っており、開発の要望があるものであり、県内の地場産業が新産業へ進出するための基盤技術となるものである。<sup>[2]</sup>

そこで、本研究では、センサネットワーク技術と組込み技術を用いてデータ収集システムを開発し、収集した大量のデータをクラウド上に送信・保存するシステムを構築することを目的とした。

ビッグデータ解析を応用した計測・解析システムの試作開発を行って技術知見を蓄積し、県内企業への技術移転を行うとともに、実用化のために解決すべき課題であるセンサ・通信の信頼性向上、耐環境性の向上、速度の向上、長期間運用のための駆動電源の確保、新規計測手法の開発等に関する技術開発を行うことを目的とした。

具体的には以下の項目について開発を行い、システムの試作を行った。いずれも IoT (Internet of Things) の基盤技術である。

● センサネットワークを介したデータ収集

計測機能と通信機能を持つセンサデバイスを用い、データの計測・収集を行うセンサネットワークシステムを試作した。計測対象としては温度・湿度・照度・電流・傾斜等を想定し、対象によりセンサを選定するが、本研究ではセンサネットワークの構築とクラウドサーバ上への格納技術の構築を主目的とすることから主に温度センサを用いて計測試験を行った。

開発にあたっては、信頼性と拡張性を考慮してオープンシステム、オープンソースを活用した。

● 収集したデータの蓄積・解析を行うシステムの構築

センサネットワークを介して収集したデータを組み込み用のワンボードコンピュータを用いて収集・解析し、必要なデータを選定してクラウド上のデータベースへ転送するサーバユニットを試作した。

## 2. 実験

平成 27 年度は、センサネットワークの基礎となるセンサユニットとデータの収集を行うサーバユニットを試作し、クラウド上のサーバへの通信試験を行った。<sup>[1]</sup>

具体的には、計測試験のために無線通信機能を有するセンサユニットおよびサーバユニットを試作し、両者間の通信試験を行った。また、センサユニットで計測したデータをサーバユニットへ送信し、収集したデータをクラウド上のサーバへの送信を行った。そのために必要となるハードウェアと通信プログラムの試作を行った。さらに、複数のセンサを用いた際の負荷試験を行った。

平成 28 年度は、センサネットワークの拡張、データ収集方法の改良、サーバシステムの改造、クラウド上へのサーバへの通信試験を行った。

### 2.1 センサユニットおよびサーバユニットの試作

図 1 に示すセンサユニットおよびサーバユニットを

試作し、データの収集を行うセンサネットワークシステムを構築した。

センサユニットでは計測機能と通信機能を持つセンサデバイスとしてディジ インターナショナル株式会社<sup>[3]</sup>製の通信モジュール XBee<sup>[4]</sup>を用いている。また、サーバユニットでは組み込み用ワンボードコンピュータである RaspberryPi<sup>[5]</sup>に XBee を接続して使用している。

XBee は自律的にネットワークを構成できる無線デバイスであり、デジタル入出力・アナログ入力・PWM 出力の機能を有しているので、簡易にセンサユニットを構成することができる。

XBee は予め以下のいずれかのモードに設定して使用する。

- センサネットワーク全体を統括する 1 基のコーディネータ
- データ計測を行う複数のエンドデバイス
- データの計測を行うとともに通信の中継を行う複数のルータ

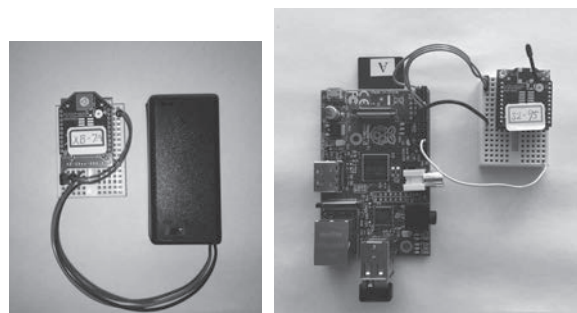


図 1 センサユニットおよびサーバユニット

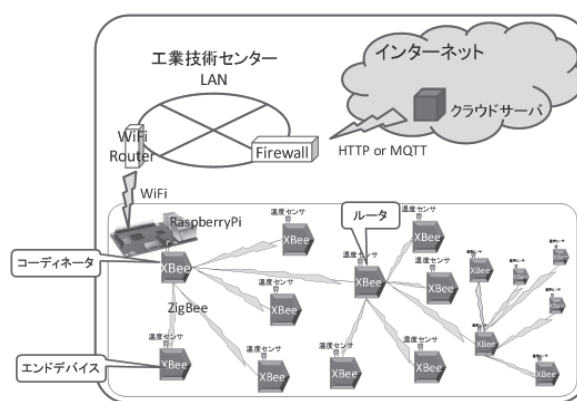


図 2 システム構成

XBee 相互の通信は ZigBee<sup>[6]</sup> に準拠した 2.4GHz 帯の無線通信により行われ、コーディネータの機能によりネットワーク構成が自律的に構成される。

XBee は 2.7 ~ 3.3V の直流電源で駆動できるため、

電池（単3×2本）による駆動が可能であり、後述の RaspberryPi からの電源供給による駆動も可能である。また、XBee のアナログ入力は0～1200mVの範囲で可能であるので、対応する範囲の電圧出力を有するセンサであれば、XBee のアナログ入力に直接接続することが可能であり、システムの簡略化を図ることができる。今回の実験で主に使用した温度センサ LM61BIZ は-30℃～+100℃の温度を+300mV～+1600mVの電圧として出力するので、-30℃～+60℃の温度を計測可能である。

サーバユニットには、ARM プロセッサを搭載したシングルボードコンピュータである Raspberry Pi（平成27年度は Model A、平成28年度は Raspberry Pi 2 Model B）を使用した。OSには Raspbian を使用し、センサネットワークとの通信のために Raspberry Pi のシリアル入出力（UART）に、コーディネータとして設定した XBee を接続し、XBee の電源（3.3V）も Raspberry Pi から供給している。

サーバユニットの Raspberry Pi に WiFi インターフェイスを装着して、工業技術センターの LAN 上に設置した無線ルータに WiFi 通信により接続し、ここを介してクラウド上のサーバにデータの送信を行った。

サーバユニットには以下の機能を持っている。

- センサネットワークを介して送信されるデータを UART に接続した XBee（コーディネータ）から読み出す
- 必要なデータを抽出・選定・整形する
- クラウド上のデータベースへデータを送信する（そのために必要な認証処理を行う）

XBee との通信プログラムおよびクラウド上のサーバへの通信プログラムは C 言語あるいは python により記述し、オープンライブラリである XBee-2.1.0<sup>[7]</sup>（現在の最新バージョンは XBee-2.3.1）および libxbee3<sup>[8]</sup> を使用している。

## 2.2 通信試験

通信試験として、サーバユニットからセンサユニットへコマンドを送信し、そのレスポンスをサーバユニット上で確認した。送信するコマンドは以下の2種類を用いた。

‘NI’ コマンド：XBee の識別子（Node Identifier）を読み出す

‘IS’ コマンド：XBee に接続されている入出力の状態を読み出す

‘NI’ コマンドはセンサネットワーク上にある全ての XBee に対し送信されるもので、XBee が固有に持っている 8 バイトの ID が送信される。また、‘IS’ コマンドは特定の XBee の ID を指定して入出力の状態の送信を求めるものであり、指定した XBee に接続されている入出力の状態が送信される。ルータを介さないで直接通信を行うことが出来る XBee の場合は約 12msec で返信される。

いずれも正常に通信が行うことが可能であることを確認し、複数（1～7）の XBee について通信を行うことが可能であることも確認した。

次に、‘IS’ コマンドにより XBee から読み出したセンサデータをクラウドサーバへ送信する試験を行った。

クラウドサーバには株式会社スカイディスク<sup>[9]</sup>のセンサクラウド GINGA を利用した。専用に用意されている http インターフェイスを利用して、クラウドサーバ上のセンサ情報（センサの ID、名称、設置場所等）の登録・更新、および、センサ計測データの送信・記録・読出を行った。また、MQTT による AWS Kinesis<sup>[10]</sup>への送信を行った。

システムの構成を図3に示す。

なお XBee の設定については、サーバユニットは API Coordinator モード、センサユニットは常時通信を行うことが出来る API Router あるいは間欠動作が可能な API End Device モードを使用している。

図4に3点の温度を継続して計測し、クラウドサーバへの送信を行った際の例を示す。

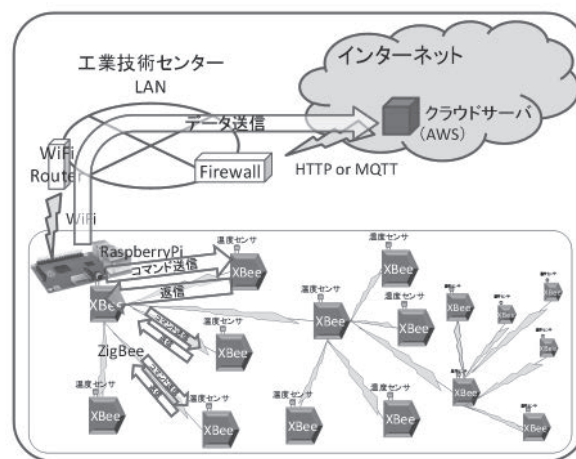


図3 通信試験のシステム構成

基本的な機能に限定したセンサユニットとサーバユニットを試作し、センサ情報の取得と計測データの収集、および、クラウド上のサーバへの通信を行うこと



が可能であることを確認した。

通信プロトコルとして HTTP、MQTT のいずれでも通信を行うことが可能であり、GINGA クラウドへの通信、AWS Kinesis への送信を行うことが可能であった。

また、センサネットワーク上のセンサユニットの数および通信頻度を増やした際のサーバユニットへの通信負荷試験において、6 パケット / 秒の通信頻度においてもセンサネットワークの通信に支障はなかったが、サーバユニットからクラウドサーバへの http プロトコルによる通信においては処理速度が十分ではなかったために、数時間でサーバユニットのメモリがオーバーフローを発生した。センサユニットを減らし、3 パケット / 秒の通信試験においても徐々に通信の遅延が発生し累積していくため、数日でサーバユニットのプログラムが停止した。今後は効率的な通信方法の検討、計測データの種類の拡張と信頼性の向上、長時間運用のための電源の確保を検討するとともに、クラウドの効率的な活用方法を検討していく。

### 2.3 クラウドサーバへのデータ転送

センサネットワークを介して収集したデータをサーバユニットを用いて解析し、必要なデータを選定してクラウド上のデータベースへ転送するシステムを構築した。

ハードウェアについては、通信試験で用いたセンサユニットおよびサーバユニットとほぼ同様の構成である。

通信試験で用いた 'IS' コマンドによる入力状態の読出しはサーバユニットからセンサユニットへコマンドを順次送って返信を待つポーリング方式であることから、通信に要する時間が長くなる問題がある。そのため、センサユニットの XBee において定期的に計測データを送信する割り込み方式に変更することとし

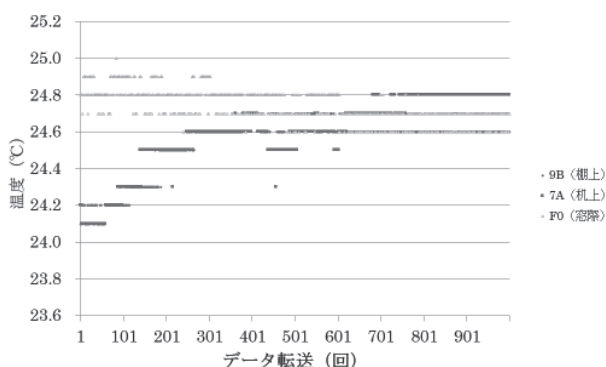


図4 通信試験の計測例

た。具体的には XBee をスリープモード (間欠動作) に設定して、センサからの入力値を定周期 (1分に1回、等) にコーディネータ (サーバユニット) に送信することとした。

サーバユニットではセンサユニットから送信されるデータを受信するごとにクラウドサーバへメッセージとして送信する。送信する内容は、デバイス名 (サーバユニットの名称)、センサ名 (XBee の ID)、データ名 (入力信号名)、データ値 (計測温度値)、時刻とした。

収集したデータの蓄積を行うためのクラウド上のサーバとしては、企業等で導入しやすいクラウドサービスを活用することも目的として、①アマゾン ウェブ サービス ジャパン株式会社の AWS (Amazon Web Services) を利用したサーバ、②マイクロソフト社の Azure [11] を利用したサーバの2つの商用クラウドサービスを用いた。

サーバユニット上で XBee からのデータ読み込みおよびクラウド上のサーバへの通信を行うプログラムは、Linux (Raspbian) 上で動作する python により記述しており、ライブラリとして、XBee からの通信については標準の XBee ライブラリを、クラウドへの通信は urllib2 [12] および Azure SDK をそれぞれ使用した。クラウドサーバへの通信プロトコルとしては、汎用性が高い HTTP (Hypertext Transfer Protocol) と軽量の通信プロトコルである MQTT (MQ Telemetry Transport) を使用した。

試験に用いた試作システムの構成を図5に示す。

クラウドサーバでは送信される計測データをデータベースに格納するために、図6に示すように Microsoft Azure の複数の機能を組み合わせてシステムを構築している。具体的には、マイクロソフト社の Azure にお

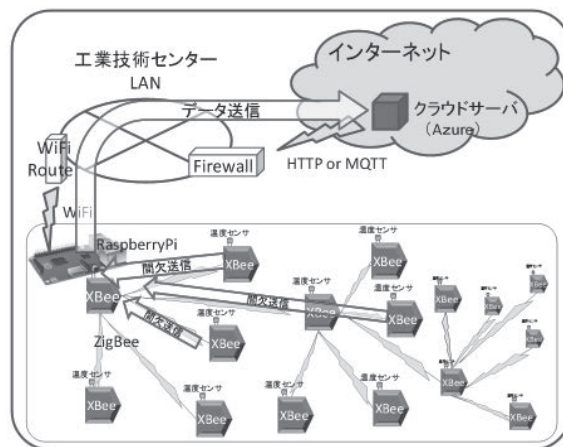


図5 試作システムおよびクラウドサービスの利用

ける IoT Hub、Stream Analytics、SQL Database の機能を組合せて用いた。それぞれの機能を以下に示す。

- IoT Hub：登録済みデバイスからのメッセージを受信し、キューに保存する
- Stream Analytics：IoT Hub からのデータストリームを取り込み、その中から必要な項目を抜き出し、型を指定して SQL Database へ出力する
- SQL Database：クラウド上のリレーショナルデータベース サービスへデータを格納する

IoT Hub はサーバユニットとクラウドサービスとの間でセキュリティで保護された信頼性がある双方向通信を実現するものであり、サーバユニットから Stream Analytics へのメッセージ送信に用いている。その通信内容は専用のプログラム DeviceExplorer を用いて監視することができる。その表示例を図7に、送信データの例を図8に示す。各データに時刻、デバイス名、センサ名、データ名、データ値が含まれていることが確認できる。

SQL Database に格納した大量のデータは Microsoft SQL Management Studio を用いて管理し、Microsoft が提供する対話型のデータ視覚化ツール表示 PowerBI を用いて表示した。

計測試験の解析例を図9に示す。センサユニットに接続した温度センサによるデータ収集とクラウドサーバへのデータ転送を安定的に実施できることを確認した。

### 3. 結 言

計測機能と通信機能を持つ無線通信ユニット XBee を用いてセンサユニットを試作し、センサユニットを相互に多数接続することによりセンサネットワークを構築した。また、Linux が動作する組込みシステム RaspberryPi を用いてデータ収集とクラウド上のサーバ

へ通信を行うサーバユニットを試作し、これを用いてセンサユニットから送信されるデータを計測してクラウドサーバへの送信試験を行った。

その結果、データ計測およびクラウドサーバへの送信を安定して実施することができるシステムを構築した。しかし、センサ数の増加、計測間隔の短縮に伴う通信量の増加により、サーバユニットの処理能力およびクラウド上のサーバへの転送速度の制約が生じることが確認された。推測される影響は、センサユニットが増加するに伴い、通信の衝突によるデータの欠測が発生すること、サーバの処理能力およびクラウドへの転送速度が不足することである。

また、計測条件によりエンドデバイスおよびルータの電源の確保が必要となるので、今後は産業技術総合研究所等との共同研究により継続して実証試験を行うとともに、県内企業への技術移転を進めていく。

さらに、サーバユニットの RaspberryPi から無線ルータへの接続が途切れるという問題も確認された。これは、XBee 間の通信と WiFi 通信が同じ 2.4GHz 帯を使

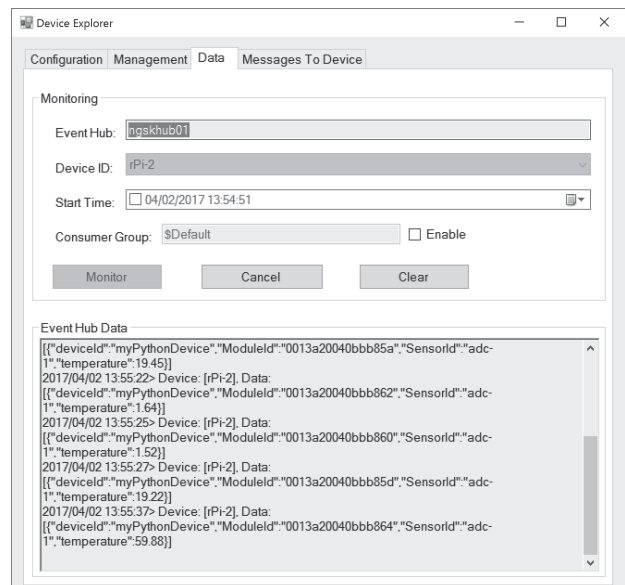


図7 DeviceExplorer の表示例



図8 IoT Hub への送信データ例

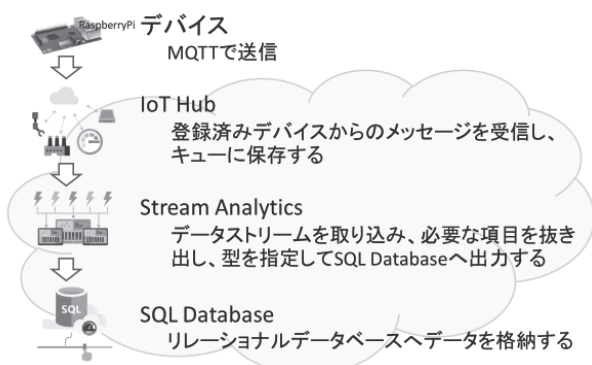


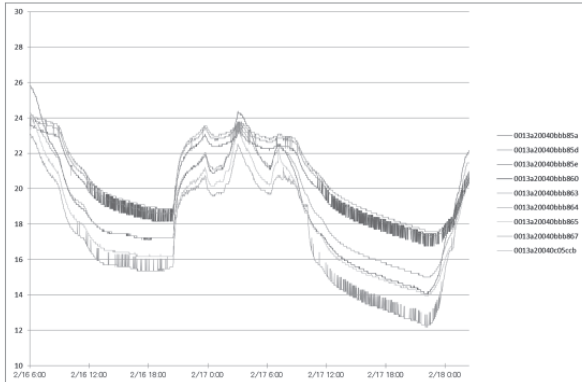
図6 クラウドサーバのシステム構成

用しているためと推測され、サーバユニットと無線ルータの間の通信に5GHzの無線LANあるいは有線LANを使用することとした。

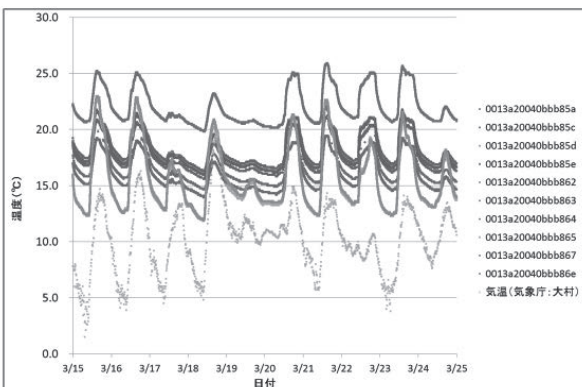
なお、センサユニットからの計測データ送信は定期的かつ非同期に行っているため、センサユニットの数の増加に伴って通信の衝突の確率が徐々に増加する。対策としては、通信の間隔を伸ばすことにより通信の衝突の確率を低減すること、通信間隔をセンサユニットごとにわずかに変えて通信の衝突を継続させないこと等が考えられる。ただし、XBeeのように通信の再送機能を実装していないデバイスを用いる場合は、ある程度の欠測を見込んだシステム設計が必要である。

### 参考文献

- [1] 藤本和貴, 長崎県工業技術センター研究報告, No.45, 33-35 (2016).
- [2] 経済産業省「2016年版ものづくり白書」
- [3] デイジインターナショナル:  
<http://www.digi-intl.co.jp/>, Accessed 2016.
- [4] XBee : <http://www.digi-intl.co.jp/products/wireless-wired-embedded-solutions/zigbee-rf-modules/zigbee-mesh-module/xbee-zb-module.html>, Accessed 2016.
- [5] Raspberrypi : <https://www.raspberrypi.org/>, Accessed 2016.
- [6] ZigBee : <http://www.zigbee.org/>, Accessed 2016.
- [7] <https://pypi.python.org/pypi/XBee/2.1.0>, Accessed 2016.
- [8] <https://github.com/attie/libxbee3>, Accessed 2016.
- [9] <https://skydisc.jp/>, Accessed 2016.
- [10] AWS Kinesis : <https://aws.amazon.com/jp/kinesis/>, Accessed 2016.
- [11] Microsoft Azure : <https://azure.microsoft.com/ja-jp/>, Accessed 2016.
- [12] urllib2 : <http://docs.python.jp/2/library/urllib2.html>, Accessed 2016.



(a) 短期



(b) 長期

図9 計測試験の解析例