

IoT デバイスからのシームレスデータ集約システムの開発

西尾啓汰^{†1} Shay Gregory^{†2} 鏑木崇史^{†1} 栗原陽介^{†1}

概要: 高齢社会において健康管理, 異常検知, 行動モニタリングなどを実現するためには多種大量のデータをシームレスに収集し, ビッグデータとして使いやすい形式で集約することが必要である. そのためには, 環境で分散的に設置されているさまざまなセンサからデータをいかに効率的に収集するかが重要となる. 本研究では安価な Raspberry Pi を用いて各センサから計測される信号を収集し, 大量に分散する情報をネットワーク経由でクラウドサーバへ集約する枠組みを構築している.

キーワード: Internet of Things(IoT), データ集約, ビッグデータ

A Development of Seamless Data Collection System from IoT Devices

KEITA NISHIO^{†1} Shay Gregory^{†2} TAKASHI KABURAGI^{†1}
YOSUKE KURIHARA^{†1}

Abstract: In order to achieve health control, anomaly detection, and behavior monitoring in aging society, it is necessary to collect large amounts of various data seamlessly and consolidate them as a big data for good usability. For that reason, it is important how efficiently we collect data from various sensors eccentrically located. In this paper, we propose a framework which enables us to collect signals from each sensor and consolidate information dispersed widely into a cloud server through network by inexpensive Raspberry Pi.

Keywords: Internet of Things(Iot), Data Collection, Big Data

1. はじめに

近年, 日本の総人口は減少を続け, 平成 28 年度の人口は前年と比較して 21 万人減少した. 一方, 65 歳以上の高齢人口は前年比 57 万人も増加している. 同年度の高齢化率は 27.7% となっていて, 4 人に 1 人以上が高齢者となっている[1]. このような高齢社会において, 高齢者の健康問題, 認知症などの様々な疾病による異常行動などが社会的な問題となっている. したがって, 高齢者の健康管理, 異常検知, 行動モニタリングなどのシステムが求められている.

それらを実現するためには多種大量のデータをシームレスに収集し, ビッグデータとして使いやすい形式で集約することが必要である. 例えば, 栗原らは電力供給を必要としないセンサを用いて, アトピー患者をモニタリングし, 睡眠時の掻破行動を検知するシステムを提案している[2]. さらに, Hao Dong らは深層学習を用いて, 睡眠段階の評価において 83.35%の正答率を記録した[3].

そのためには, 環境で分散的に設置されているさまざまなセンサからデータをいかに効率的に収集するかが重要となる.

本研究では安価な Raspberry Pi 3 を用いて各センサから計測される信号を収集し, 大量に分散する情報をネットワーク経由でクラウドサーバへ集約する枠組みを構築してい

る.

2. システム概要

提案するシステムの概要を図 1 に示す.

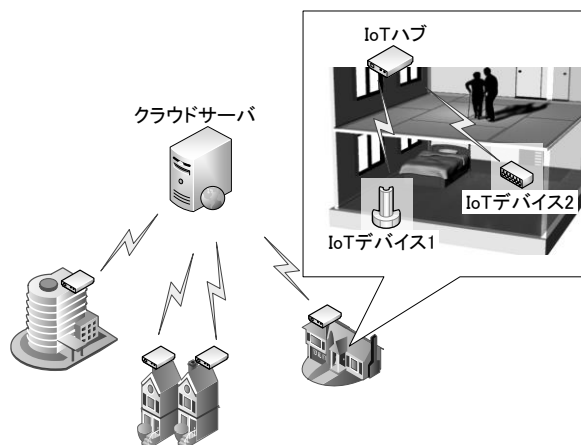


図 1 システムの概要

取得されるデータは環境的に分散しており, かつその量は膨大である. したがって, それらのデータは扱いやすいデータ形式に整えられた上で, ネットワークを経由してクラウドサーバに収集されるべきである. また, クラウドサーバには膨大な量のデータが集約されるため, 扱いやすい形にデータを処理する過程は, ネットワークを経由する前

^{†1} 青山学院大学
Aoyamagakuin University
^{†2} テキサス工科大学

Texas Tech University

の段階でローカルになされるべきである。さらに、膨大な量のセンサから出力されるデータを集約するため、データを転送するシステムの設置コストは小さくならなければならない。

したがって本稿では、様々なセンサから取得される性質の異なる信号を、信号の特性に合わせてローカルに信号処理を行うことで扱いやすい形に変換し、安価な Raspberry Pi model B を用いてネットワーク経由でクラウドサーバに集約するシステムを提案する。

3. 実装

提案するシステムを実装したものを図 2 に示す。

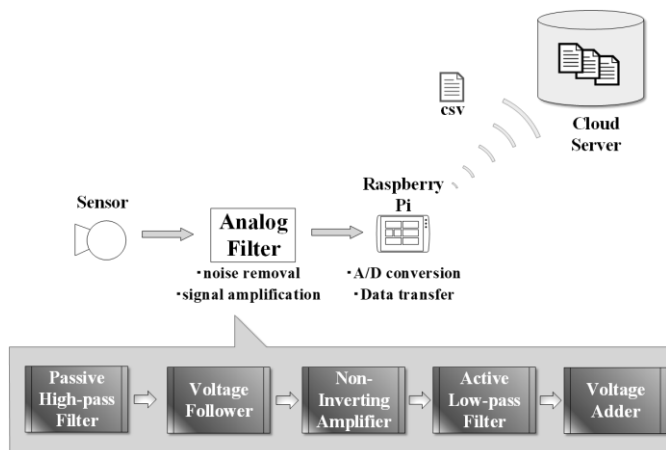


図 2 システムの概要

センサから出力される信号は多くのノイズを含み、取得したい信号成分は微小なものとなっている。したがって、アナログフィルタに通し、それらのノイズを減減し、取得したい成分を増幅する必要がある。フィルタから出力されるアナログ信号は AD コンバータにより離散化され、Raspberry Pi に取り込まれる。取り込んだデータは Raspberry Pi 上で csv データとして出力され、Python を用いてそのデータはクラウドサーバへと同期される。

3.1 アナログフィルタ

アナログフィルタはパッシブローパスフィルタ、ボルテージフォロワ、非反転増幅回路、アクティブローパスフィルタ、加算回路から構成される。

センサから出力される信号は多くの周波数成分を含んでいて、その中には直流成分も含まれる。その信号をそのまま増幅させると、取得したい成分だけでなく直流成分も増幅され、大きなオフセットをもった信号になってしまう。信号の電圧がオペアンプの許容電圧を超えてしまうと飽和してしまう為、オペアンプによるフィルタに通す際には直流成分が除去されていなければならない。したがって、最初にセンサの出力信号をパッシブハイパスフィルタに通す。ハイパスフィルタのカットオフ周波数 f_{ch} は式(2. 1)によって決定される。カットオフ周波数とは、通過帯域に比べて利得が 3dB 下がる周波数のことである。

$$f_{ch} = \frac{1}{2\pi CR} \quad (2. 1)$$

取得される信号をフィルタによって処理する際、センサの出力インピーダンスが、フィルタの入力に対して著しく大きな値を取ることが多い。すると、センサからの出力信号が目減りしてフィルタに入力されるという現象が発生する。ボルテージフォロワはセンサからの出力インピーダンスとフィルタへの入力インピーダンスを等しくすることにより、これを回避することができる。

ボルテージフォロワによりインピーダンスが適切に調整された信号は、その後非反転増幅回路を通り、極性を変えずに電圧を増幅される。非反転増幅回路はオペアンプと 2 つの抵抗によって実現される。入力電圧と出力電圧をそれぞれ V_i 、 V_o とし、出力からのフィードバック抵抗の抵抗値を R_f 、反転入力端子に接続され、接地される抵抗の抵抗値を R_s とすると、それらの関係は式(2. 2)に示される。

$$V_o = \frac{R_s + R_f}{R_s} V_i \quad (2. 2)$$

非反転増幅回路により電圧を増幅された信号は、取得したい信号成分だけでなく高周波のノイズ成分を含んでいる。信号を周波数解析を行う際、解析したい周波数成分より高い周波数成分にノイズを持っていると、高周波ノイズが解析したい周波数帯域にも影響を及ぼしてしまう。これはエイリアシングと呼ばれる。このエイリアシングを回避するため、信号に対して高周波成分を減減させるアクティブローパスフィルタを実装する。アクティブローパスフィルタはオペアンプと、抵抗とキャパシタそれぞれ 2 つにより構成される。抵抗の抵抗値 R_1 、 R_2 とキャパシタの容量 C_1 、 C_2 によってカットオフ周波数 f_{cl} が式(2. 3)のように決定される。

$$f_{cl} = \frac{1}{2\pi\sqrt{R_1C_1R_2C_2}} \quad (2. 3)$$

高周波ノイズ成分を減減した信号は、最後に加算回路を通る。アナログフィルタからの出力はその後 Raspberry Pi の GPIO により取り込まれる。GPIO は負の電圧を許さないが、信号は負の値を取り得る。したがって、適切な電圧をオフセットとして信号に持たせた後で Raspberry Pi に取り込む必要がある。したがって、加算回路により信号に対してオフセットを持たせる処理を行う。加算回路はオペアンプと複数の抵抗により構成される。

以上のアナログフィルタにより、センサからの出力信号は、ノイズ成分が減減され、取得したい信号成分は適切に増幅され、Raspberry Pi に正しく入力される信号となった。

3.2 Raspberry Pi

まず Raspberry Pi とデータを同期するクラウドサーバとを Secure Shell(SSH)により接続し、公開鍵によって認証しておく。

アナログフィルタからの出力は、SPI 接続のアナログデジタルコンバータによって離散化され、GPIO を通して Raspberry Pi に取り込まれる。

Raspberry Pi に入力されたデジタル信号は Python3. 0 で作成したコードにより csv ファイルとして出力され、Raspberry Pi 上のディレクトリ Dir 内に保存される。その後、Linux 上の rsync コマンドを用いて、クラウドサーバ上の任意のパスに Raspberry Pi 内の Dir 全体を同期する。

4. 実証実験

システムの実証実験の様子を図 3 に示す。

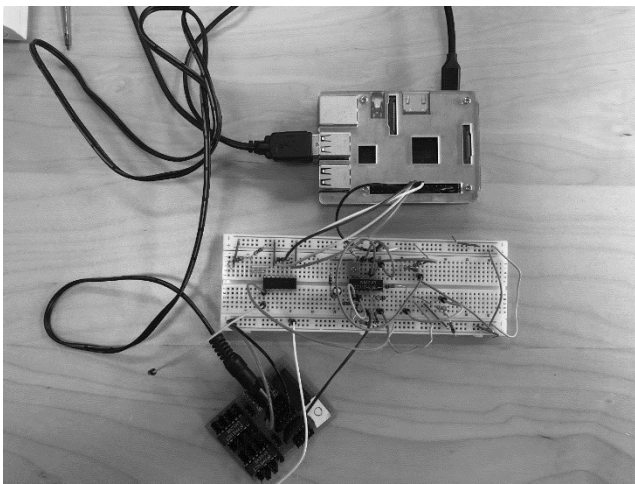


図 3 実装されたシステムの様子

4.1 アナログフィルタの実証

アナログフィルタの実装に伴い、低消費電力クワッド汎用オペアンプ LM324N(Texas Instruments Inc.)を使用した。LM324N によってボルテージフォロワ、非反転増幅回路、アクティブローパスフィルタ、加算回路を実現した。なお、LM324N には MCW03(Minmax Technology Co., Ltd.)を用いて $\pm 15V$ の両電源を供給した。

まず、直流成分の除去が目的である為、 f_{ch} が極めて小さな値となるよう、 $R=10.1M\Omega$ の抵抗と $C=10\mu F$ のキャパシタを用いて、 $f_{ch}=0.0015757Hz$ のパッシブハイパスフィルタを実装した。

次に、ボルテージフォロワはオペアンプの出力を反転入力に接続することにより実現した。

非反転増幅回路については、 $R_s=470\Omega$ の抵抗、 $R_f=0\Omega \sim 50k\Omega$ の可変抵抗を使用することによって実現した。これにより、センサからの出力電圧に対応して、 R_f の抵抗値を変化させることにより、増幅率が 1 倍から 107 倍の範囲で変化し、センサの電圧を適切に増幅することが可能となる。

アクティブローパスフィルタは $R1=R2=100\Omega$ 、 $C1=100\mu F$ 、 $C2=47\mu F$ とすることにより、 $f_{cl}=23.2Hz$ とした。

さらに、 $+15V$ の正電源から抵抗を用いて分圧し、信号に

0.4688V を加えた。これは、Raspberry Pi の GPIO への入力が 0~3.3V に制限されている為である。また、直流成分はハイパスフィルタによって減衰されているが、非反転増幅回路の増幅率によっては Raspberry Pi への入力に影響を与えかけないため、加算電圧は GPIO の制限電圧の中心値よりも小さく設定している。

実装したアナログ回路の動作を確認するため、実際に 5 つの信号を入力して実証実験を行った。この実証実験は CONTEC 社の A/D コンバータ AI-1608AY-USB を用いて行った。

入力信号の 1 つ目はファンクションジェネレータ AFG-2105(GW Instek)により生成した周波数 1Hz、振幅 1mV、オフセット 0v の正弦波である。これは呼吸を模した信号である。この信号の各フィルタからの出力を図 4 に示す。5 つのグラフは上から入力信号、ハイパスフィルタの出力、非反転増幅回路の出力、ローパスフィルタからの出力、加算回路からの出力である。

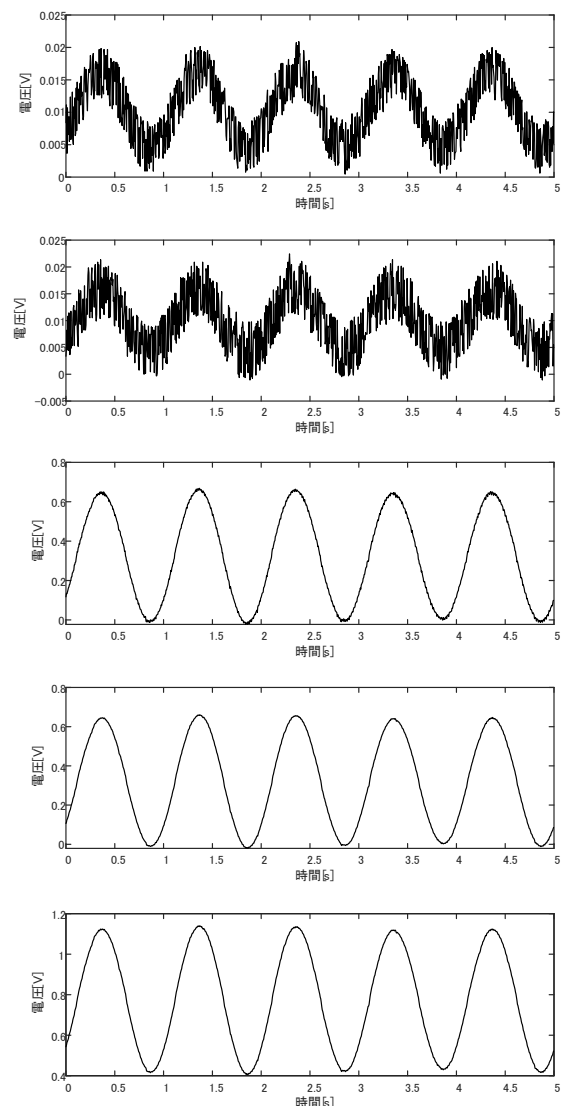


図 4 呼吸を模した信号の結果

加算回路の出力を見てみると、その前後で正しく 0.

4688Vのオフセットが加えられることがわかる。また、1Hzは $f_{cl}=23$ 。2Hzより小さい為、振幅の減少は確認されない。

2つ目の信号はファンクションジェネレータ AFG-2105(GW Instek)により生成した周波数 50Hz, 振幅 1mV, オフセット 0v の正弦波である。この信号は高周波ノイズを模している。その結果を図 5 に示す。

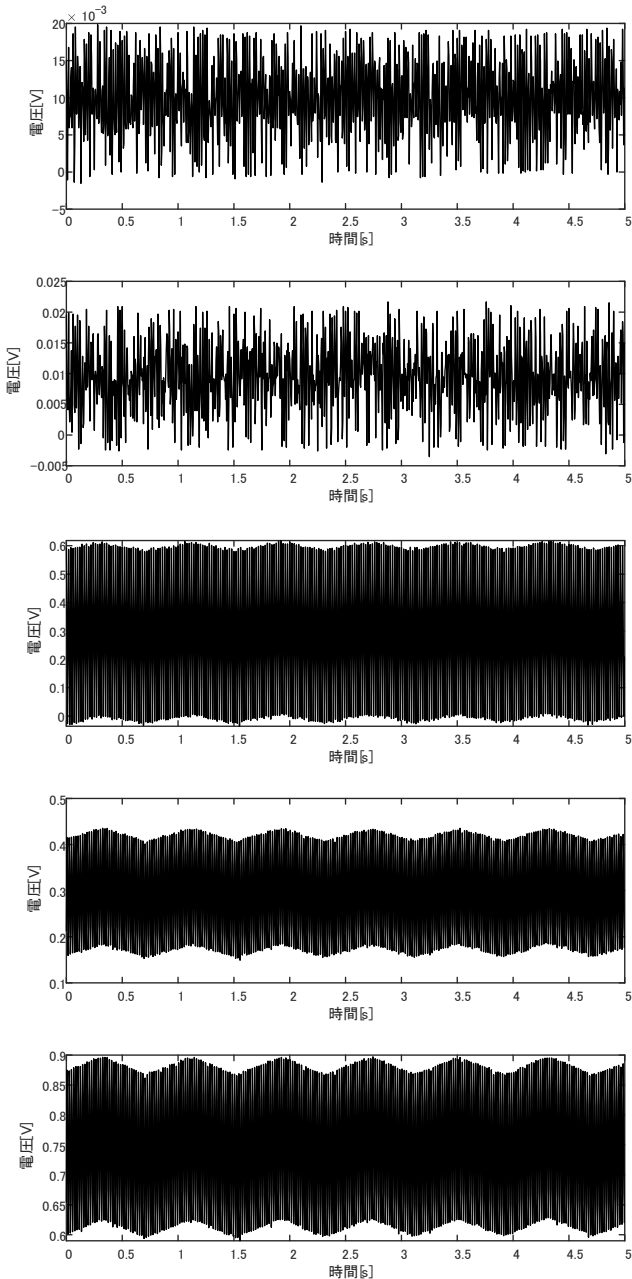


図 5 ノイズを模した信号の結果

上から3つ目と4つ目のグラフを見比べると、入力信号の周波数は f_{cl} を超えているため、ローパスフィルタによって振幅が小さく抑えられていることがわかる。

3つ目の信号はファンクションジェネレータ AFG-2105(GW Instek)により生成した周波数 10Hz, 振幅 1mV, オフセット 0v の矩形波である。これは高周波ノイズを仮定している。その結果を図 6 に示す。

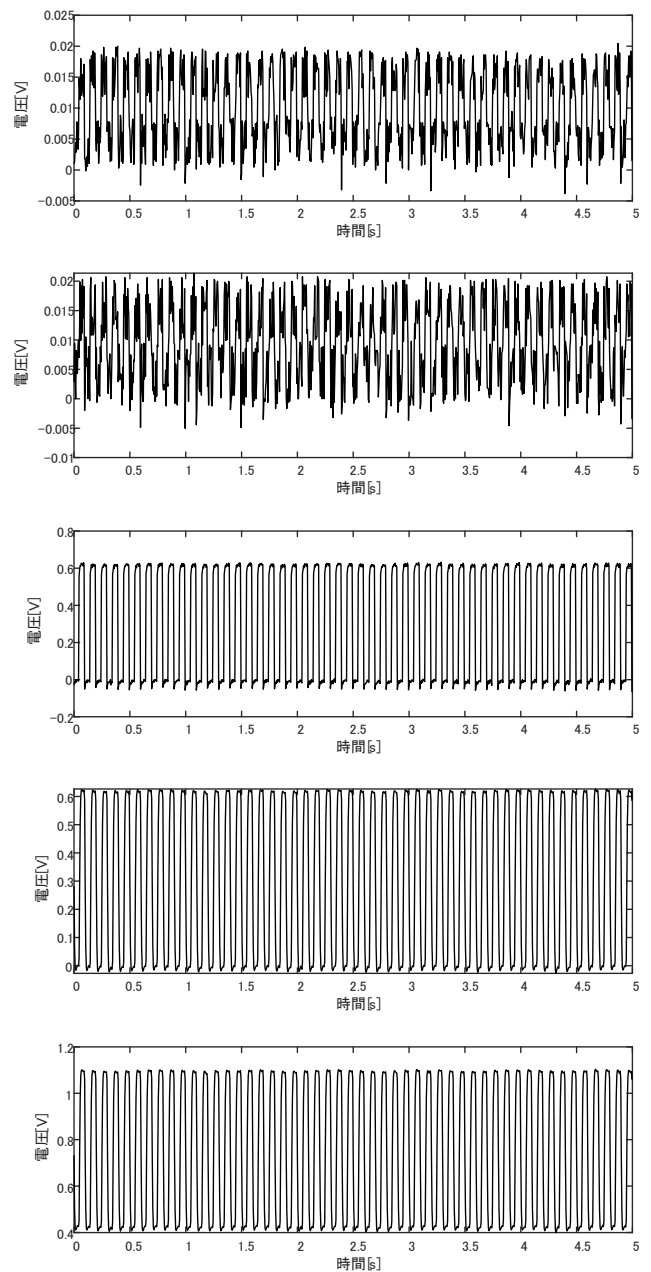


図 6 矩形波の結果

矩形波はフーリエ級数展開した際、基本波の周波数の n 倍の成分が、基本波の振幅の n 分の 1 となることが知られている。矩形波をフィルタに通したものをフーリエ変換を行い、振幅スペクトルを求めた。そのスペクトルを図 7 に示す。

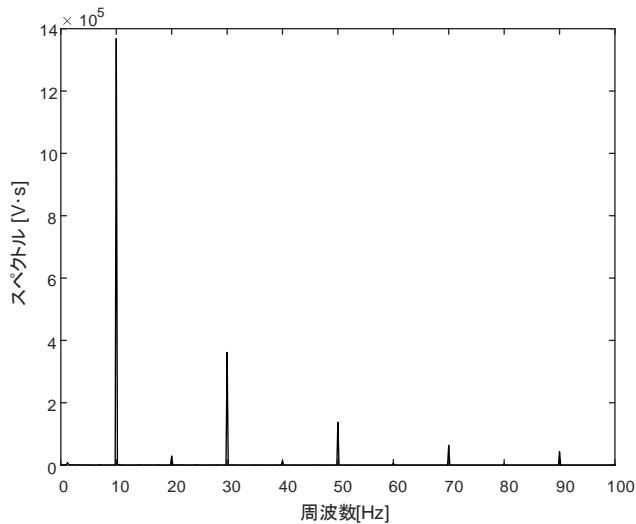


図 7 フィルタ後の矩形波の振幅スペクトル

図 7 を見てみると、10Hz の基本波の奇数倍の周波数成分にスペクトルが立っていることが確認できる。さらに、スペクトルの大きさは指数関数的に小さくなっている。これにより、ローパスフィルタが正常に動作していることがわかる。

4 つ目の信号はコンデンサマイクを使用した高感度圧力センサを、被験者の寝たベッドマットの下に設置し取得した信号である。このセンサはベッドマットを通して被験者の身体の振動を取ることができる。その結果を図 7 に示す。

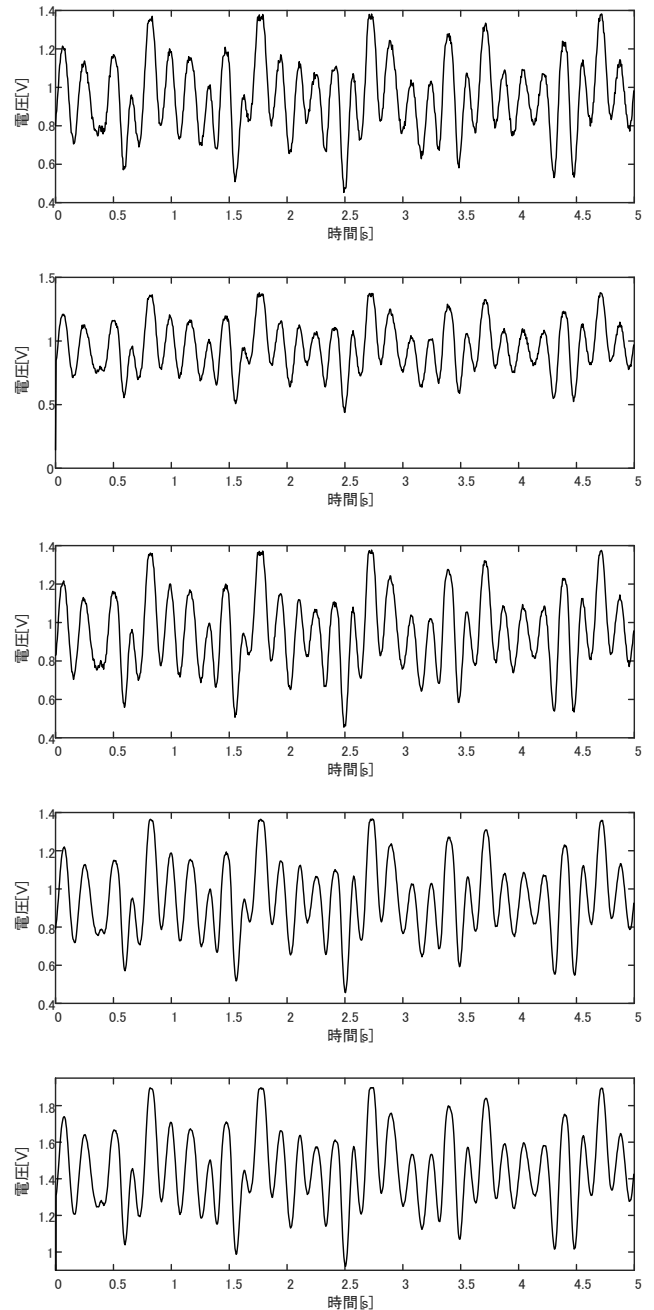


図 7 高感度圧力センサによる信号の結果

5 つ目のグラフを見てみると、約 1 秒ごとに振幅が大きくなっていることがわかる。これは被験者の脈波成分を取得できているものと考えられる。

5 つ目の信号は圧電セラミックを使用したセンサをベッドの脚に設置したセンサによる出力信号である。このセンサはベッドの脚にかかる圧力の変化を取ることができ、人間の生体振動を捉えることができる。この信号をフィルタにかけた結果を図 8 に示す。

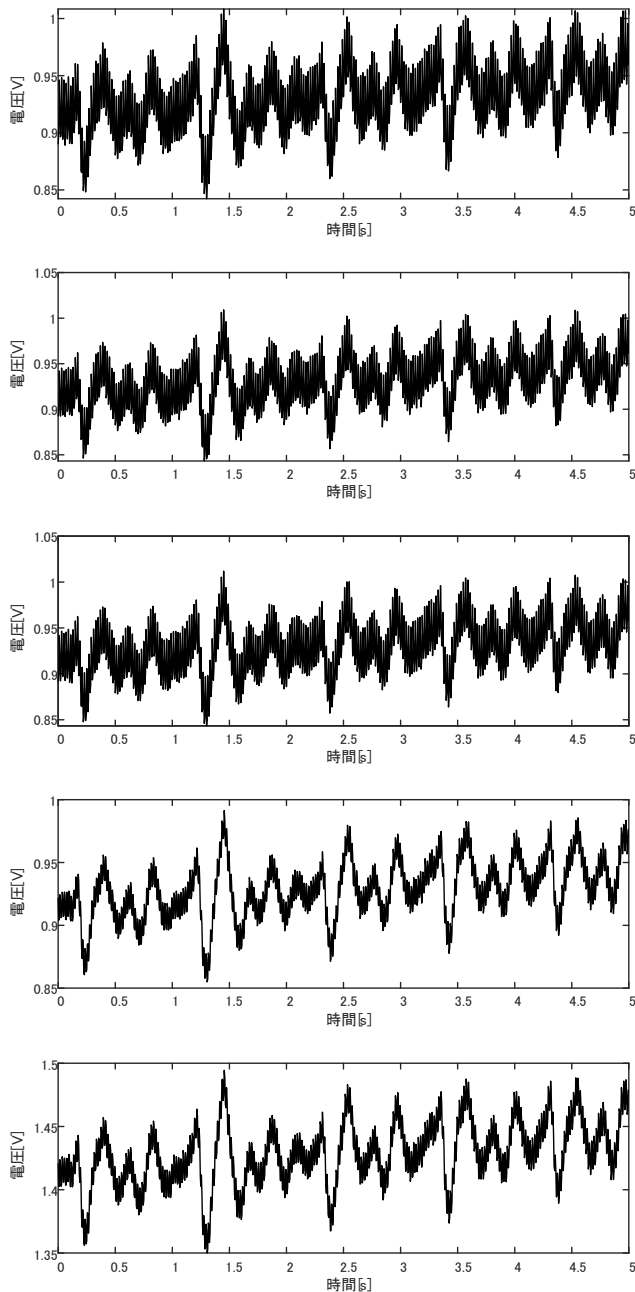


図 8 圧電セラミックによる信号の結果

3 目と 4 目目のグラフを見比べると、高周波ノイズ成分が十分小さく抑えられていることがわかる。

4.2 Raspberry Pi の実証

アナログフィルタからの出力は逐次比較型 12 ビット・アナログ・デジタルコンバータ MCP3208(Microchip Technology Inc.)を通して Raspberry Pi に取り込んだ。

まず、MCP3208 と Raspberry Pi を用いて A/D 変換を行い取得したデータについての検証を行う。リファレンスとして CONTEC 社の AI-1608AY-USB を用いた。

4.1 章において入力した 5 つの信号について、提案手法により取得したデータとリファレンスによって取得されたデータを図 9 から図 13 に示す。上の図がリファレンスによる取得データ、下の図が提案手法による取得データを示

している。

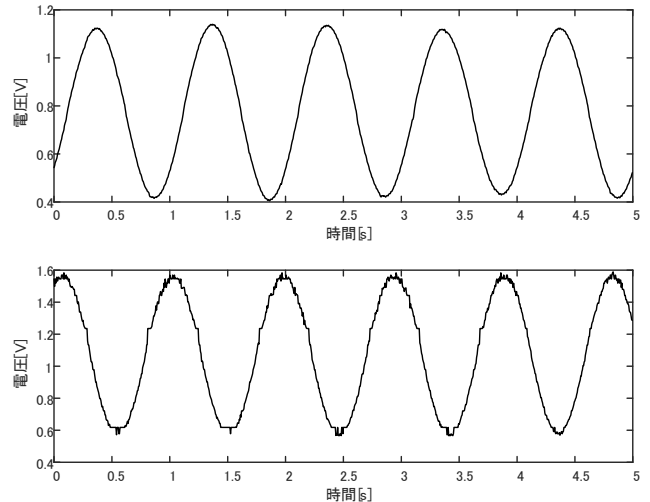


図 9 1Hz の正弦波の比較

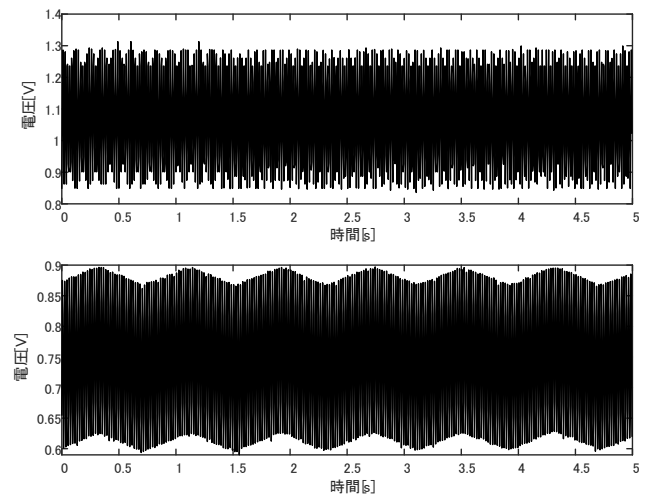


図 10 50Hz の正弦波の比較

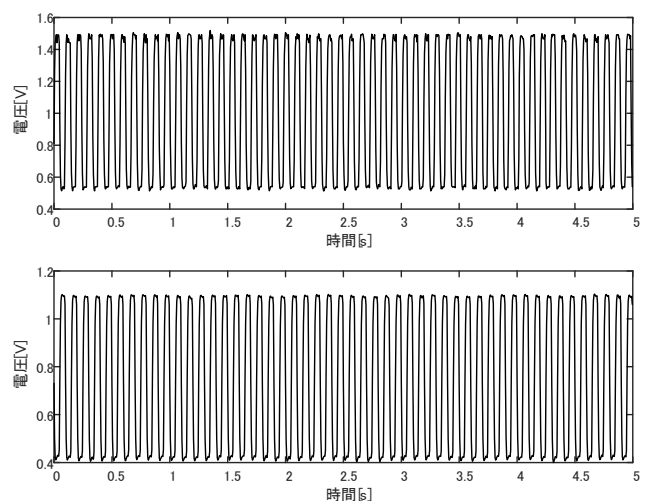


図 11 10Hz の矩形波の比較

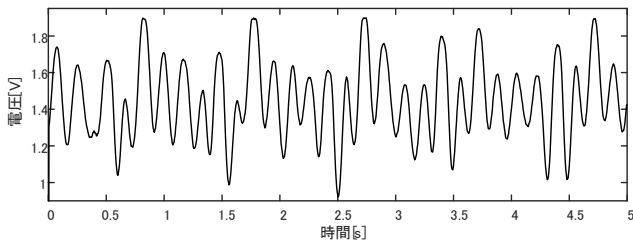
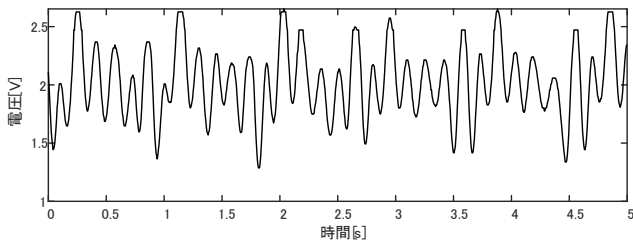


図 12 圧力センサの信号の比較

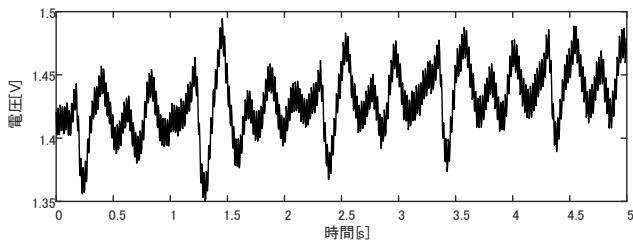
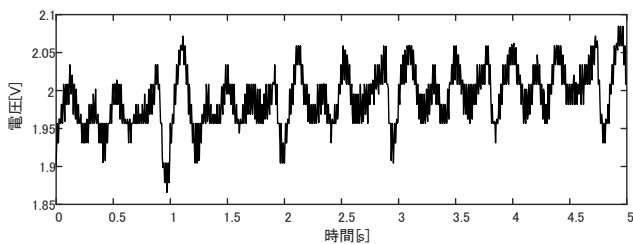


図 13 圧電セラミックの信号の比較

5つの信号とも、提案手法により正しくデータの取得が行われていることがわかる。

次に、Raspberry Piによるクラウドサーバとのデータ同期の実証実験を行う。

Python3.0によって、SPI接続されたMCP3208が離散化された信号をcsvファイルvalue.csvとして出力するプログラムを作成した。

次に、出力したvalue.csvをSSH接続したクラウドサーバに収集する。Terminal上でLinuxのコマンドrsyncを実行することによりクラウドサーバにデータを同期する。value.csvを含むディレクトリ全体を同期することによりこれを実現した。また、rsyncのオプションとして-zを指定することにより、同期するデータサイズが大きくなったときにも対応できるように配慮した。

5. 考察

高齢化社会において、さまざまな特徴を持ったデータが収集されるが、今回提案したシステムのアナログフィルタは、そのカットオフ周波数や増幅率など、フィルタの特性

を適切に調整することによって、それを扱いやすい形に変えることができる。さらに、大量のデータが取得されるようになり、データ取得に伴うコストが重要となってきた。

データ同期の際に、Linux上のrsyncコマンドを用いて実現した。rsyncの特徴として、同期元ディレクトリと同期先ディレクトリとの差分のみを送信するという特徴がある。断片的に取得されるデータの収集というのを考えると、ディレクトリ全体を送信してしまうと、それまでに取得されたデータが多くなればなるほど、同期にかかる時間も長くなると考えられる。その点、rsyncを用いることにより、累積データサイズが大きくなっても、送信時間が著しく長くなることを避けることができる。

6. むすび

本稿ではアナログフィルタにより信号の特性に合わせてローカルに信号処理を行い、安価なRaspberry Piを用いてクラウドサーバへとデータを集約するシステムを提案した。実証実験の結果、処理したデータをcsvファイルとしてRaspberry Pi上に書き出し、それをクラウドサーバに同期することに成功した。これは異常検知やモニタリングを行う上で不可欠なものである。今後の展望としては、Linux上のcronetabを活用することにより、ADLに関するデータを自動的に取得するシステムの構築などが考えられる。

参考文献

- [1] “統計からみた我が国の高齢者(65歳以上)”, 総務省統計局, 2017, p. 1-3, <http://www.stat.go.jp/data/topics/pdf/topics103.pdf>, (参照 2018-08-02)
- [2] Yosuke Kurihara, Takashi Kaburagi, and Kajiro Watanabe: Sensing Method of Patient's Body Movement Without Attaching Sensors on the Patient's Body Evaluation of “Scratching Cheek,” “Turning Over and Scratching Back,” and “Scratching Shin”, IEEE Sensors Journal, Vol. 16, Issue 23, pp. 8372-8278(2016)
- [3] Hao Dong, Akara Supratak, Wei Pan, Chao Wu, Paul M. Matthews, and Yike Guo: Mixed Neural Network Approach for Temporal Sleep Stage Classification, IEEE TRANSACTIONS ON NEURAL SYSTEMS AND REHABILITATION ENGINEERING, Vol. 26, No. 2, pp. 324-333 (2018)