

人間を測るためのワイレスセンサ開発
ー 人を測ってモノを創る ー
第 0 版

平成 2 8 年 3 月

人間情報応用チーム

北海道立総合研究機構
産業技術研究本部 工業試験場
製品技術部 デザイン・人間情報グループ

はじめに

ポケットに入れたセンサが自分の体を自動で測り、無線ネットワークで健康を管理する、そんな時代がここ数年で一気に花開きつつあります。超小型のデジタル無線とセンサ、省電力なマイクロコンピュータ、高容量の超小型電源といった技術が開発され、ウェアラブルセンサを市場化する環境が整ったのです。そして、その市場はこの数年間で爆発的に拡大しています。人を測る技術は、健康の向上だけでなく家電の制御やユーザインタフェースなど新たなモノ・サービス開発に対する巨大な潜在力を秘めています。

ウェアラブルセンサの研究開発は、中小企業や個人レベルでも簡単にはじめられる時代になりました。安価な開発環境が整い、アイデアとちょっとした技術さえあれば、ほんの少額の投資で健康センサやアプリを研究開発できるようになったのです。

とはいえ、それでもセンサの試作を進めるためには相応の知識と準備が必要です。そこで私たちは、「安く作って安く失敗できる」ウェアラブルセンサの試作技術をこの冊子で簡単に解説し、普及したいと考えています。それは、様々なタイプの人々が参入することで、いわゆる技術者には思いもよらない、全く新しいアイデアが実現すると考えるからです。思いもよらないセンサを作って未来を切り開きましょう。

北海道立総合研究機構 産業技術研究本部 工業試験場
製品技術部 デザイン・人間情報グループ

目次

第 1 章	ウェアラブルセンサを開発する.....	1
1.1	概要.....	1
1.2	ウェアラブルセンサのデータ処理フロー.....	1
1.3	処理負荷の振り分け.....	2
第 2 章	センサの種類.....	5
2.1	運動や方向を測るセンサ.....	5
2.1.1	加速度センサ.....	5
2.1.2	ジャイロセンサ.....	6
2.1.3	地磁気センサ（コンパス）.....	6
2.2	圧力を測るセンサ.....	6
2.2.1	絶対圧センサ.....	6
2.2.2	差圧センサ.....	7
2.3	温度を測るセンサ.....	7
2.3.1	サーミスタ.....	7
2.3.2	サーモパイル.....	7
第 3 章	Bluetooth 通信が可能なハードウェアプラットフォーム.....	9
3.1	LightBlue® Bean.....	9
3.2	mbed™ HRM1017.....	10
3.3	Shimmer™.....	11
第 4 章	ソフトウェアによるデータ処理.....	13
4.1	フィルタ.....	13
4.1.1	バタワースローパスフィルタ.....	13
4.1.2	バタワースハイパスフィルタ.....	20
4.1.3	バタワースバンドストップフィルタ.....	22
4.1.4	バタワースバンドパスフィルタ.....	23
4.2	デジタルフィルタの実装.....	26
4.2.1	2 次形式によるプログラミング.....	26
4.2.2	片方向フィルタと双方向フィルタ.....	28
4.3	アナログセンサのサンプリングに対する注意点.....	28
4.3.1	エイリアシング.....	28

目次

4.3.2	入力インピーダンス	28
第 5 章	開発事例	31
5.1	乳牛用血中カルシウム濃度測定システム	31
5.2	ワイヤレス圧力センサの機能試作	35
5.3	筋電位計測によるバスドラム用キックペダルの評価	38
参考文献	41

第1章 ウェアラブルセンサを開発する

1.1 概要

ウェアラブルセンサを開発するためには、センサやマイクロコンピュータといったハードウェアと、コンピュータ内部でデータを処理するファームウェアやアプリケーションといったソフトウェアが必要となる。近年では、ハードウェアの開発環境には Arduino や mbedTM といったファームウェアも含めた開発プラットフォーム（統合開発環境）が整備され、迅速な試作が可能となってきた。以後の章では、これらのプラットフォームを活用して試作していく手法について述べる。

1.2 ウェアラブルセンサのデータ処理フロー

ワイヤレス生体計測センサのデータ処理過程は、おおむね図 1-1 のようにセンサ→データ変換→コンピュータ処理、といった流れになる。これは生体計測に限らず、他のセンサにおいても同様である。図 1-1 の左から見ていくと、最初に筋電や脈波などを実際に計測するアナログセンサがある。心電計であれば電極（電位検出器）が、光電脈波計であればフォトセンサ（光検出器）がこれにあたる。

次に、アナログセンサで電気信号に変換された生体情報を AFE（アナログフロントエンド）に送り、目的に適合するアナログデータ処理を行いマイクロプロセッサ (Micro Processing Unit; MPU) に信号を送る。AFE とは、アナログ信号をマイクロコンピュータに取り込む前に、センサの特性に合わせて調整を行う回路を指す。アナログセンサで得られた微弱な電気信号を増幅してフィルタによりノイズを除去し、A/D 変換（アナログ／デジタル変

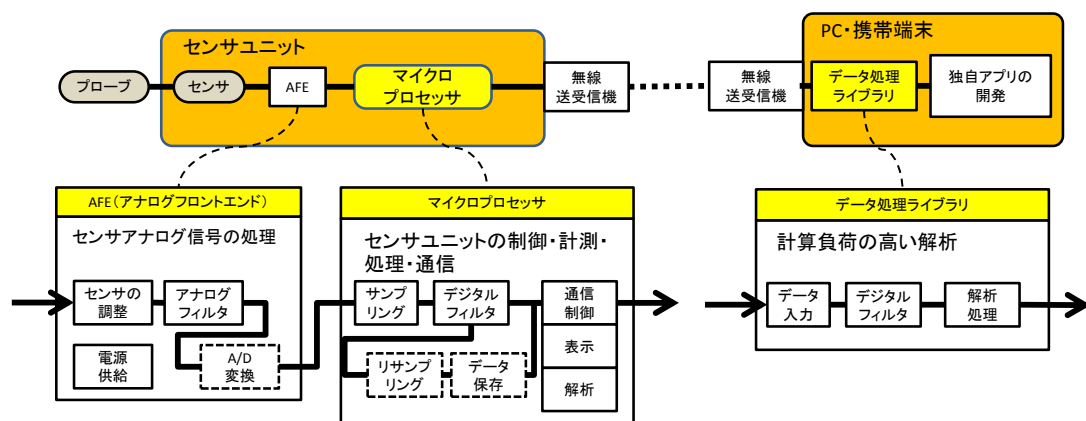


図 1-1 生体計測センサのデータ処理フロー

換)を行い MPU と通信するなど様々な調整を行う。近年では、センサと AFE をすべてパッケージ化され、センサ IC として製品化されているものも多い。

MPU は CPU (Central Processing Unit)、マイクロコンピュータ (マイコン) などとも呼ばれ、内部に書かれたプログラム (ファームウェア) に従い様々なデータ処理や制御を行うモジュールである。近年は高速化・多機能化され、各種通信機能や A/D 変換、D/A 変換、無線通信機能などがワンチップ化されていることも多い。MPU では、AFE から送られた信号をサンプリングし、デジタルフィルタや解析・表示・通信制御・データ保存などを行う。PC や携帯端末との通信制御も MPU が行う。

PC や携帯端末では、センサユニットから送られた生体情報を豊富な演算パワーとソフトウェアライブラリを用いて加工し、解析や表示など自身の目的にかなう独自アプリケーションを開発する。

1.3 処理負荷の振り分け

生体計測において、センサユニットと PC 携帯端末(場合によってインターネットサーバ)のどのプロセッサにデータ処理をどのように振り分けるかは、計測の目的や場面によって異なってくる。センサで計測された生データをそのまま保存するのか、必要な情報だけを演算して圧縮保存するのか、データ量を犠牲にしてでもバッテリー消費を抑えるのか、これらをよく吟味する必要がある。中でも注意すべきは、サンプリングレートと通信速度である。

サンプリングレートは、センサがデータを収集する頻度であり、収集するデータの最大周波数によって決まる一方、MPU 内の計算量や通信速度によって制限される。温度や湿度のように変化が緩やかなものであればサンプリングレートを $1 \sim 0.01\text{Hz}$ (数秒～数分) と長く

表 1-1 センサユニット内の計算処理過程と MPU 計算量・無線通信量の関係

計算処理過程	MPU 計算量	無線 通信量	例
(生データ)	小	大	心電位、筋電位
デジタル 周波数フィルタ	↓	↑	例：ノイズ除去 (ハムノイズ)
解析処理	大	小	例： 心拍数 (ピーク検出) 筋活動量 (移動平均)

取れ、MPU 内部の計算時間を十分に確保でき、バッテリー消費も抑えられる。一方、筋電位では 500-2000Hz と高頻度でサンプリングする必要があるため、MPU 内の処理が追いつかない場合も出てくる。特に複雑な計算処理（三角関数、べき乗根、指数・対数の入る統計処理など）は非常に重く、次のサンプリングまでに処理が終わらない可能性もある。その場合は MPU 上内の計算処理を減らして PC や携帯端末に処理を委ねたり、サンプリングレートそのものを下げたり、といった事も検討すべきである。また、計算処理の軽いアルゴリズムを用いることも大事である。

通信速度は、センサから PC への情報供給量の上限値となるとともに、バッテリー消費量にも大きな影響がある（表 1-1）。通信速度は通信規格によって上限があり、また通信環境によって制限される。例えば Bluetooth Low Energy(BLE)は接続性と低バッテリー消費を主眼に策定された規格であり、そのトレードオフとして通信速度が 1Mbps、実質 50kbps 程度と遅くなっている [1]。高速通信が必要な場合は、Wifi や Bluetooth 2.1～3.0 を検討すべきである。また、これらの規格や仕様には上限の通信速度が記載されているが、通信環境などの影響を考えると実用上の通信速度はその 1/5～1/10 程度を上限に考えるべきである。また、電波出力による電力消費は、MPU そのものの電力消費よりはるかに大きい。バッテリー消費を考えるなら、まずは無線通信の頻度を下げるのが効果的である。例えば、心拍計を開発するのであれば、心電位の生波形をそのまま電波出力するより、MPU 上で心拍間隔を計算し心拍数のみを電波に乘せる方がバッテリー消費を抑えられる。

第2章 センサの種類

センサには、測る対象によって様々な種類や方式がある。近年では、センサ単体だけでなく、増幅や調整、デジタル化まで行う IC 化されたワンパッケージ型のセンサも多く出ている。機能ここでは主に、市販されているセンサの概要を紹介する。

2.1 運動や方向を測るセンサ

2.1.1 加速度センサ

加速度センサは、加速度を測定する。現在よく用いられているのは MEMS 加工による半導体チップのタイプである。加速度センサは、センサが設置された物体の加速度や重力加速度を測定する。加速度センサには 1 軸つまり 1 方向のみ計測可能なもののほか、2 軸、3 軸のタイプがあり、3 軸では x、y、z 各々の軸の加速度が測定できる。重力加速度を用いて、鉛直軸に対する傾斜を測定する事も可能である。

代表的なセンサ：

Analog Devices ST ADXL345Z （3 軸）（図 2-1）

<http://www.analog.com/jp/products/mems/mems-accelerometers/adxl345.html>

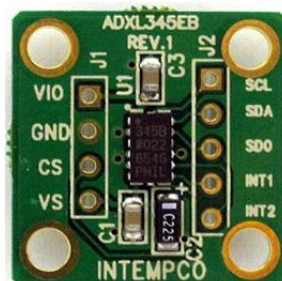


図 2-1 3 軸加速度センサ ADXL345（Analog Devices 社）。図はセンサ評価キット EVAL-ADXL345Z、図中央の IC が ADXL345 本体である。

（<http://www.analog.com/jp/products/mems/mems-accelerometers/adxl345.html>）

2.1.2 ジャイロセンサ

ジャイロセンサは、角速度（回転の速さ）を測定する。センサ内に MEMS 加工された振動子が回転するときに発生するコリオリ力を検出し、角速度に変換する。

代表的なセンサ：

ST Microelectronics L3GD20H （3 軸）

http://www.st.com/web/jp/jp/catalog/sense_power/FM89/SC1288/PF254039

2.1.3 地磁気センサ（コンパス）

地球の磁場の方向と大きさを 3 軸それぞれについて測定する。地球の磁場に対するセンサの 3 次元姿勢が測定できる。

代表的なセンサ：

ST Microelectronics LSM303D （3 軸地磁気、3 軸加速度センサ搭載）

http://www.st.com/web/jp/catalog/sense_power/FM89/SC1449/PF253884

2.2 圧力を測るセンサ

2.2.1 絶対圧センサ

空気圧を測定する。代表的なものとしては、大気圧を測定するセンサがある。近年は分解能が向上し、1Pa、高度にして 10cm の差を検知できるものが出てきている。

代表的なセンサ：

村田製作所 ZPA2326-0311A-R

<http://www.murata.com/ja-jp/products/sensor/baropressure>

Bosch BME280

https://www.bosch-sensortec.com/bst/products/all_products/bme280

2.2.2 差圧センサ

2本のポート間の圧力差を検出する。膜式と熱線式（フローセンサ）がある。膜式はポート間に薄膜が配置されており、膜にかかる圧力差を電圧に変換する。ポート内に流れは発生しない。フローセンサは、ポート内をごくわずかに空気が流れるようになっており、センサ内部にある熱源からの流れる熱を近傍の温度センサが検出し、それを差圧に変換する。

代表的なセンサ：

FreeScale MP3V5010（膜式）

<http://www.nxp.com/ja/products/sensors/pressure-sensors:DRSNSPRSSR>

Sensirion SDP600（フローセンサ方式）

<http://www.sensirion.co.jp/products/differential-pressure-sensors>

2.3 温度を測るセンサ

2.3.1 サーミスタ

測温抵抗体とも呼ばれる。センサ部抵抗値が温度の関数となっており、この抵抗値を計算することで温度が計算できる。近年ではIC化されていることも多く、その場合はセンサからアナログ電圧やデジタル通信（I2CやUART等）により出力される。

代表的なセンサ：

村田製作所 NXFT15（サーミスタ、アンプ・補正等必要）

<http://www.murata.com/ja-jp/products/thermistor>

Texas Instruments LM35DZ（補正機能付きワンパッケージ温度センサ、アナログ出力）

<http://www.tij.co.jp/product/jp/LM35>

Texas Instruments HDC1000（温湿度センサモジュール、デジタル出力）

<http://www.tij.co.jp/tool/jp/hdc1000evm>

2.3.2 サーモパイル

非接触温度センサとも呼ばれ、赤外光のエネルギーを電圧に変換するセンサである。特定の条件下では出力値が温度の関数になる。放射現象を活用したセンサであることから極めて短時間で計測が可能であり、例えば耳式体温計などの温度センサとして活用される。

代表的なセンサ：

OMRON D6T-44L-06

<https://www.omron.co.jp/ecb/products/sensor/11/d6t.html>

第3章 Bluetooth 通信が可能なハードウェアプラットフォーム

ウェアラブルセンサを開発するには、センサの情報を処理しデータを送信するハードウェアが必要である。これらを一から開発するとなれば、ハードウェアの設計、基板の作成、部品の実装、テスト、といったことを実行しなければならず、かなりの知識とコストがかかる。これらに全くの徒手空拳で投資するのはかなりの勇気が必要である。

そこで開発の初期段階では、周辺回路や通信機能が整備されたハードウェアプラットフォーム（統合開発環境）の利用をお勧めする。近年では、Arduino や mbed™ といったプラットフォームが出てきており、自分の要望に近いマイコンボードを購入してファームウェアの開発を進められるので、迅速かつ低コストな機能試作が可能となっている。

ここでは、ウェアラブルセンサのハードウェア試作に用いるプラットフォームについて紹介する。ここで紹介するものの他にも、多数のプラットフォームが製品化されているので、ぜひ検討してほしい。

3.1 LightBlue® Bean

LightBlue® Bean は、Punch Through Design 社（USA）が開発した Arduino 互換のマイコンボードで、Bluetooth Low Energy (BLE) 通信機能と 3 軸加速度センサを内蔵している（図 3-1）[2]。このボードに搭載された BLE 無線モジュール LBM313 は日本国内の電波法技術基準適合証明（技適）を取得済である¹。電源用に 2032 型リチウムコイン電池のホルダが装備されており、45.5 × 20.3 × 8.38 mm とコンパクトなサイズながらボード単体での動作が可能となっている。プログラムの書き替えも PC やスマートホンから BLE 経由で行える。ボード上には 2 本の A/D または I²C ポートと 6 本の I/O ポートの他、回路部品を組み付けられるスルーホールが空けられている。

プログラミングは、Arduino 言語およびコンパイラを用いて行う。Arduino は Atmel 社の AVR マイクロコンピュータをベースに入出力ポートを備えたボードと Arduino IDE（Integrated Development Environment; 統合開発環境）から構成されるマイコンデバイス開発環境である。IDE には、プログラミング用のエディタやコンパイラが含まれており、無償で提供されている。Arduino のプログラムは「スケッチ」と呼ばれ、C++ 言語を簡略化したような文法で簡潔に記述できる。Arduino は、動作させるまでに低コストかつ短期間で済み、初心者でも大きな負担なく行えることが特徴で、様々な種類の製品が広く普及している。

¹総務省のサイトから技適取得済か否かの確認ができる。（総務省：技術基準適合証明等を受けた機器の検索 <http://www.tele.soumu.go.jp/giteki/SearchServlet?pageID=js01>）

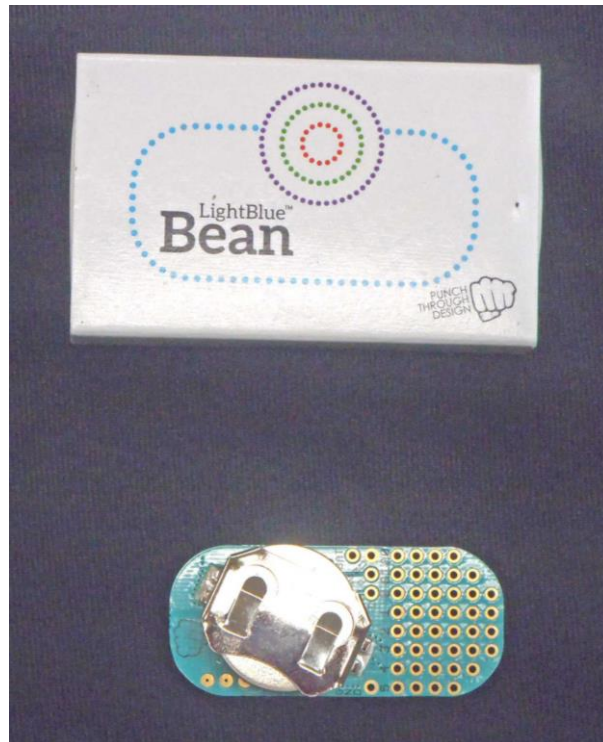


図 3-1 LightBlue® Bean (Punch Through Design 社)

Punch Through Design 社のサイト：

LightBlue® Bean <https://punchthrough.com/bean>

3.2 mbed™ HRM1017

HRM1017 は、ホシデン社が開発した mbed™ 互換のマイコンボードで、BLE 通信機能が搭載されている（図 3-2 左）。ピッチ変換基板にはんだ付けされたボードがスイッチサイエンス社で販売されている。バッテリーは別途接続する必要があるが、図 3-2 右に示すモジュール本体は非常にコンパクトでしかも ARM® Cortex™ M0 CPU を内蔵しファームウェアを書き込めるため、本格的な試作で小型化する際に非常に有利である。プログラムの書き替えは PC から USB 接続により行う。

mbed™ とは、ARM 社のマイクロコンピュータを搭載した試作用ワンボードマイコンとソフトウェア開発環境から構成される総合開発環境である。mbed™ 対応マイコンボードは様々なメーカから多数市販されており、使用する機能に応じて選択することができる。ソフトウェア開発環境はオンラインで無償提供されており、ブラウザ上でプログラミングとコンパイルが可能となっている。プログラミング言語は C++ で、ブラウザ上で様々なソフトウェア

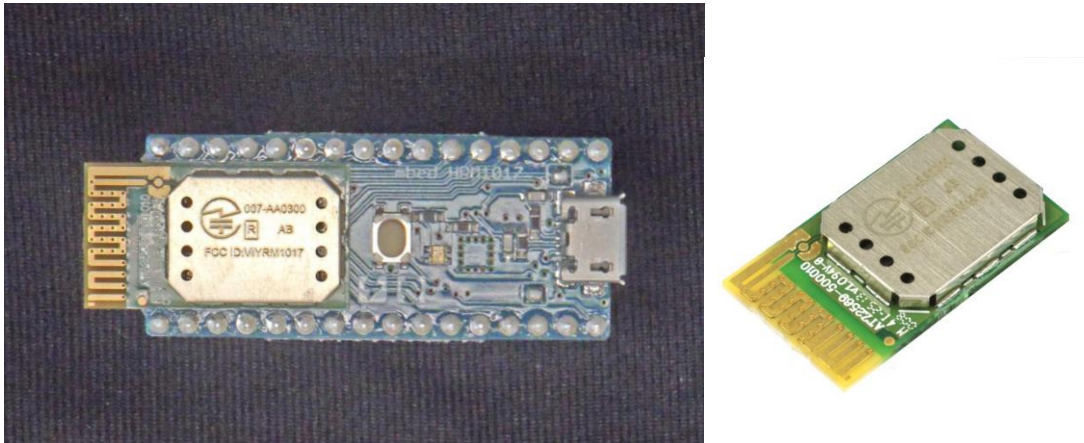


図 3-2 mbed™ HRM1017（スイッチサイエンス社／ホシデン社）。（左：ピッチ変換基板付きボード 17.8x47x6.2mm（ピン部除く）、右：BLE モジュール 12.7x23.0x2.6 mm、<https://www.switch-science.com/catalog/1755/>）

モジュールが提供されており、デベロッパーの開発負担低減が図られている。

スイッチサイエンス社のサイト：

HRM1017	https://www.switch-science.com/catalog/1755/
mbed™開発環境	https://developer.mbed.org/

3.3 Shimmer™

Shimmer™は Shimmer 社（アイルランド）が開発したワイヤレス生体計測機器用モジュールで、現在のバージョンは Shimmer3 である（図 3-3）。MPU には MSP430（Texas Instruments）が搭載されている。標準モジュールで 9 軸センサ（3 軸の加速度、ジャイロ、地磁気）を備えており、これらのデータを用いて 3 次元姿勢を算出できる。これらに加え、Shimmer3 は気圧センサ、温度センサを標準で備えている。また心電位、筋電位、脈波、皮膚電気反応、ブリッジアンプが拡張モジュールとして用意されており、新規にハードウェアを開発しなくとも生体計測システムの試作が可能である。内部には 2GB のマイクロ SD カードが標準搭載され、大量のデータを保存できる。通信は、BLE では転送速度が十分でないため Bluetooth 2.1+EDR（内蔵 Bluetooth モジュールは Microchip 社の RN-42、技適取得済）の SPP プロファイルを用い、仮想シリアル通信でデータ転送を行う。ファームウェアは標準で数種類用意されており、リアルタイム計測やロギングなど、用途によって変更できる。ファーム

ウェアは Tiny-OS と C 言語の 2 種類のバージョンが用意されている。ソースは公開されており、これをカスタマイズして開発することも可能である。アプリケーション開発用の API(Application Programming Interface ソフトウェア開発のための部品のようなもの)も充実しており、C#や Android 用 API の他、LabView、Matlab 用のドライバも用意されている。

Shimmer 社のサイト：

Shimmer <http://www.shimmersensing.com>



図 3-3 Shmmer3 + EMG/ECG モジュール (Shimmer 社)

第4章 ソフトウェアによるデータ処理

4.1 フィルタ

フィルタとは、ここでは時系列で得られた信号（例えば加速度、角速度、温度、圧力など）に含まれる余分な信号を除去する回路やソフトウェアを指す [3] [4] [5]。時系列の生信号、すなわち一切の加工がされていない信号には、通常はノイズなど観測に望ましくない信号も含まれるため、それらをフィルタで除去していく。極めて微弱なためノイズが重畳しやすい生体情報計測にとってフィルタは大変重要な技術であり、これがなければ正確な計測はほぼ不可能である。

時系列信号処理の基本は、ローパスフィルタである。ローパスフィルタとは、その名の通り周波数の低い信号のみを通過させるフィルタである。一般に、時系列信号は多くの周波数が含まれており、その中の高い周波数の信号のみをカットするのがローパスフィルタである。例えば、1Hz の波と 100Hz の波が混在する信号が観測された場合、カットオフ周波数（フィルタで落とす周波数の境界値）が 10Hz のローパスフィルタを使えば 1Hz の信号のみを抽出できる。

フィルタには様々な形式があるが、ここでは MPU 上での計算量が少なく最もポピュラーな無限インパルス応答 (Infinite impulse response; IIR) バタワースフィルタのみを取り扱う。

4.1.1 バタワースローパスフィルタ

バタワースフィルタとは、「最大平坦特性フィルタ」とも呼ばれる周波数フィルタで、信号が通過する帯域が最大限平坦、つまり通過前後での信号の大きさの変化が最も小さくなるように設計されている。図 4-1 に示す正規化バタワースローパスフィルタ²を例にとると、カットオフ周波数より小さい周波数の信号はほぼ 0 dB、つまりフィルタ通過前後における振幅変化がほとんどないが、カットオフ周波数では振幅比が-3dB（実数に変換すると $10^{-3/10} \approx 0.5$ 倍）に減衰し、それより大きい周波数では急激に減衰する。この正規化バタワースローパスフィルタは、ハイパス、バンドパス、バンドストップとすべてのバタワースフィルタの原型となる。

バタワースフィルタを設計するには、まずフィルタの極や零点などフィルタの特性を表すパラメータを求める。 N 次正規化バタワースローパスフィルタのラプラス変換 $H(s)$ を極 s_k ($k=1\dots N$) で表す。極とは、伝達関数 $H(s)$ が無限大に発散する点の事を言う。実際にはこ

² 「正規化」とは、一般に式中の重要なパラメータを“1”となるように変換する事をいう。ここでは、カットオフ周波数を $\omega_c = 1(\text{rad/s})$ とすることを正規化と呼んでいる。

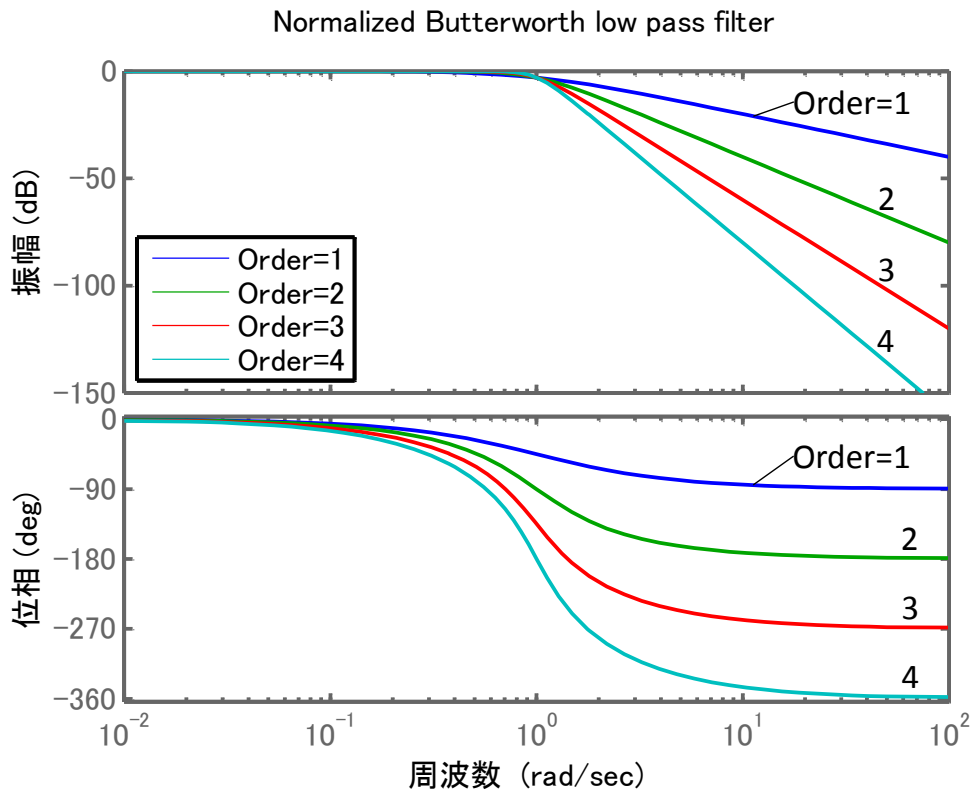


図 4-1 正規化バタワースローパスフィルタのボード線図

ボード線図は、横軸に周波数、縦軸に振幅と位相を取り、伝達関数の特性を表した図である。上図は、フィルタ通過前後の振幅比を表し、下図は位相の変化を表す。正規化バタワースローパスフィルタでは、カットオフ周波数 $\omega_c = 1(\text{rad/s})$ から周波数が増加するに従い、フィルタ通過後の振幅が急激に減少する事がわかる。

の点より、周波数応答(特定の周波数信号を入力したときの増幅率や位相変化)を示す $H(j\omega)$ 、すなわち s 平面の虚軸上の値が重要である。

$$H(s) = \frac{1}{\prod_{k=1}^N (s - s_k)} \quad (4-1)$$

この極 s_k は、次に示す正規化バタワース多項式の解で与えられる(バタワース多項式の詳細は参考文献 [5]等を参照)。

$$1 + (-1)^N s^{2N} = 0 \quad (4-2)$$

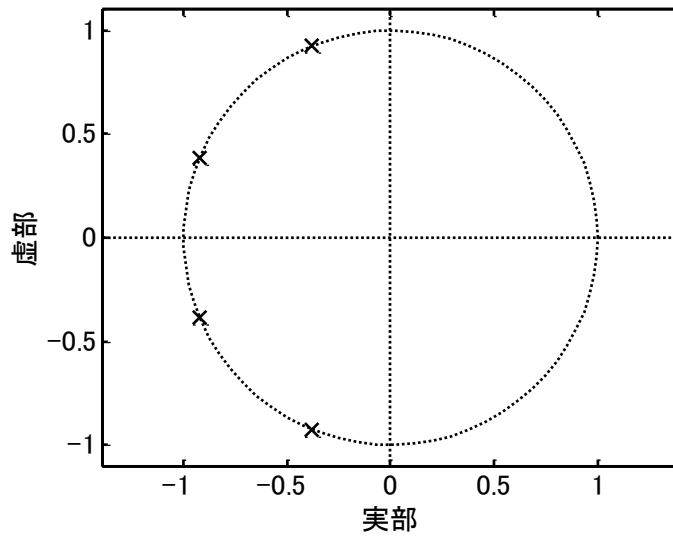


図 4-2 s 平面上に表した正規化バタワースローパスフィルタの極（4 次）

s 平面とは、複素平面（横軸に s の実数、縦軸に虚数を取る）上に、 $H(s)$ をプロットしたものである。 $H(s)$ も複素数となる。図中の“ \times ”は「極」と呼ばれ、 $H(s)$ の絶対値 $|H(s)|$ が無限大に発散する点を表す。

式(4-2)の解はオイラーの公式 $e^{j\varphi} = \cos \varphi + j \sin \varphi$ を用いて $1 = e^{2\pi k}$ 、 $-1 = e^{j(2k+1)\pi}$ (k は整数) とおき、解 $s_k = e^{j\varphi_k}$ を求めると、

$$s_k = \begin{cases} e^{\frac{jk\pi}{N}} & (N: \text{奇数のとき}) \\ e^{\frac{j(2k+1)\pi}{2N}} & (N: \text{偶数のとき}) \end{cases} \quad (4-3)$$

となる。このうち、 $\frac{\pi}{2} < \varphi_k < \frac{3\pi}{2}$ 、つまり s_k の実数が負となる安定解が N 次正規化ローパスバタワースフィルタの極となる。

$$s_k = \begin{cases} e^{\frac{j(N+2k-1)\pi}{2N}} & (N: \text{奇数のとき}) \\ e^{\frac{j(N+2k)\pi}{2N}} & (N: \text{偶数のとき}) \end{cases} \quad (4-4)$$

この解を複素平面（ラプラス変換の変数名に従い「 s 平面」とよぶ）上に図示すると、図

4-2 のようになる。

次数 $N=1$ のときは、極 $s_1 = e^{j\pi} = -1$ であるから、フィルタのラプラス変換は式(4-1)より

$$H_1(s) = \frac{1}{s - e^{j\pi}} = \frac{1}{s + 1} \quad (4-5)$$

となる。

$N \geq 2$ のときは共役³な極同士を掛け合わせ、係数が実数となるように扱おうと便利である。 N が偶数のとき、図 4-3 に示すように式(4-4)の k 番目と $N+k-1$ 番目の解が共役となる。その両者から積を取った 2 次ローパスフィルタ $H_k(s)$ は、

$$\begin{aligned} H_k(s) &= \frac{1}{(s - s_k)(s - s_{N+k-1})} \\ &= \frac{1}{(s - s_k)(s - \bar{s}_k)} \\ &= \frac{1}{s^2 + 2 \cos \varphi_k s + 1} \end{aligned} \quad (4-6)$$

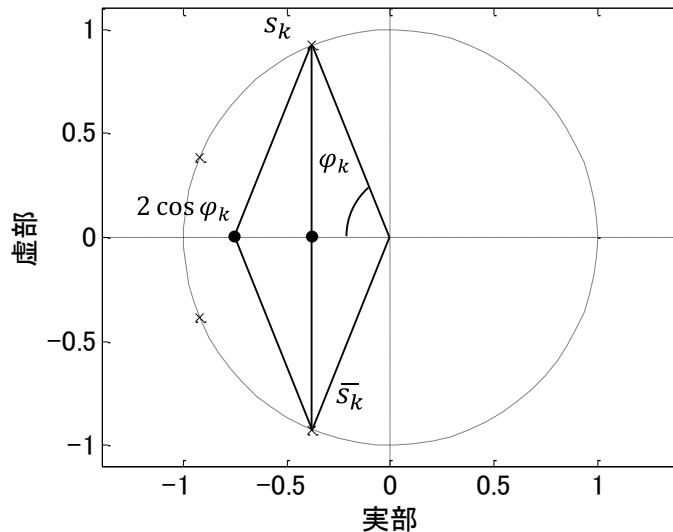


図 4-3 共役な極の積による伝達関数係数の実数化

³ 共役とはある複素数の虚数部の符号が反転したものを指し、 $z = x + jy$ の共役複素数は $\bar{z} = x - jy$ となる。 $z\bar{z} = |z|^2 = x^2 + y^2$ 、 $z + \bar{z} = 2x$ となる性質がある。

となり、係数は実数のみとなる。式中の φ_k は図 4-3 に示すように、s 平面の負の実数軸からみた角度である。この φ_k を昇順に並べ、番号 k を振り直すと次の通りとなる。

$$\varphi_k = \begin{cases} \frac{(k-1)\pi}{N} & (k = 1, \dots, (N-1)/2, \text{ N:奇数の時}) \\ \frac{(2k-1)\pi}{2N} & (k = 1, \dots, N/2, \text{ N:偶数の時}) \end{cases} \quad (4-7)$$

以上から、正規化バタワースローパスフィルタのラプラス変換は、次の通りとなる。

- N が奇数の時($N = 1$)

$$H(s) = \frac{1}{s - e^{j\varphi_1}} = \frac{1}{s - e^{j\pi}} = \frac{1}{s + 1} \quad (4-8)$$

- N が奇数の時($N \geq 3$)

$$H(s) = \frac{1}{s + 1} \prod_{k=2}^{(N-1)/2} \frac{1}{s^2 + 2 \cos \varphi_k s + 1} \quad (4-9)$$

$$\varphi_k = (k - 1)\pi/N \quad (k = 1, \dots, (N-1)/2)$$

- N が偶数の時

$$H(s) = \prod_{k=1}^{N/2} \frac{1}{s^2 + 2 \cos \varphi_k s + 1} \quad (4-10)$$

$$\varphi_k = (2k - 1)\pi/2N \quad (k = 1, \dots, N/2)$$

これで得られたラプラス変換はアナログ計測に用いられるが、コンピュータ上でデジタル測定するには、飛び飛びの時間で計測する「離散時間」に対応する必要がある。そのため、得られたラプラス変換に $z = e^{sT}$ (T はサンプリング間隔) とする「 z 変換」という変換を施す (z 変換にはさらに多くの説明が必要となるので省略する)。デジタルフィルタでは、エイリアシング⁴対策のため次に示す z 変換の近似型である双一次変換がよく用いられる。

$$s = \frac{1}{T} \ln(z) \approx \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}} \quad (4-11)$$

ラプラス変換に双一次変換を施し、デジタルフィルタの係数を求めていく。2 次フィルタの一般式を

$$H(s) = \frac{a_0 s^2 + a_1 s + a_2}{s^2 + b_1 s + b_2} \quad (4-12)$$

としたとき、これに双一次変換を施して整理すると次のようになる。

⁴ サンプリング定理より、サンプリングレートの半分以上の周波数の信号が、それより低い周波数の信号として誤って重畳する現象。これを防ぐには、ナイキスト周波数 (サンプリングレートの 1/2 の周波数) 以上の周波数信号をカットする必要がある。

$$H(z) = c \frac{(a_0 h^2 + a_1 h + a_2) + (-2a_0 h^2 + 2a_2)z^{-1} + (a_0 h^2 - a_1 h + a_2)z^{-2}}{1 + c(-2h^2 + 2b_2)z^{-1} + c(h^2 - b_1 h + b_2)z^{-2}} \quad (4-13)$$

ただし、 $c = 1/(h^2 + b_1 h + b_2)$, $h = 2/T$ である。この式を任意のカットオフ周波数 ω_c で z 変換したい場合は、 $h = 2/\omega_c T$ とすれば直ちに得られる。

ここで、双一次 z 変換における分子分母の係数を

$$H(z) = \frac{A_0 + A_1 z^{-1} + A_2 z^{-2}}{1 + B_1 z^{-1} + B_2 z^{-2}} \quad (4-14)$$

とおくと、各係数は、

$$\begin{aligned} A_0 &= c(a_0 h^2 - a_1 h + a_2) \\ A_1 &= c(-2a_0 h^2 + 2a_2) \\ A_2 &= c(a_0 h^2 + a_1 h + a_2) \\ B_0 &= c(h^2 - b_1 h + b_2) \\ B_1 &= c(-2h^2 + 2b_2) \end{aligned} \quad (4-15)$$

となり、のように、ラプラス変換の分子分母の係数とサンプリング周期から求められる。

一次フィルタの場合は、一般式を

$$H(s) = \frac{a_0 s + a_1}{s + b_1} \quad (4-16)$$

とおくと、双一次変換は次のようになる。

$$H(z) = c \frac{(a_0 h + a_1) + (-a_0 h + a_1)z^{-1}}{1 + c(-h + b_1)z^{-1}} \quad (4-17)$$

ただし、 $c = 1/(h + b_0)$, $h = 2/T$ であり、

$$H(z) = \frac{A_0 + A_1 z^{-1}}{1 + B_1 z^{-1}} \quad (4-18)$$

とおいたときの各係数は、次の通りとなる。

$$\begin{aligned} A_0 &= c(-a_0 h + a_1) \\ A_1 &= c(a_0 h + a_1) \\ B_0 &= c(-h + b_1) \end{aligned} \quad (4-19)$$

4.1.2 バタワースハイパスフィルタ

バタワースハイパスフィルタは、高周波域の信号を通過し、低周波域をカットする（図 4-4）。その極はバタワースローパスフィルタのそれと一致し、異なるのは原点に零点があることのみである（図 4-5）。正規化バタワースハイパスフィルタのラプラス変換は、ローパスフィルタの式(4-8)～(4-10)の s に $1/s$ を代入することで簡単に得られ、その係数はローパスフィルタの逆順となる。

2 次フィルタの一般式

$$H_k(s) = \frac{a_0 s^2 + a_1 s + a_2}{b_0 s^2 + b_1 s + b_2} \quad (4-20)$$

に $s \rightarrow 1/s$ を代入すると、

$$H_k(s) = \frac{a_2 s^2 + a_1 s + a_0}{b_2 s^2 + b_1 s + b_0} \quad (4-21)$$

となり、ハイパスフィルタの係数が逆順となることがわかる。式(4-15)に係数を逆順に当てはめれば、ハイパスフィルタにおける z 変換の各係数が直ちに得られる。

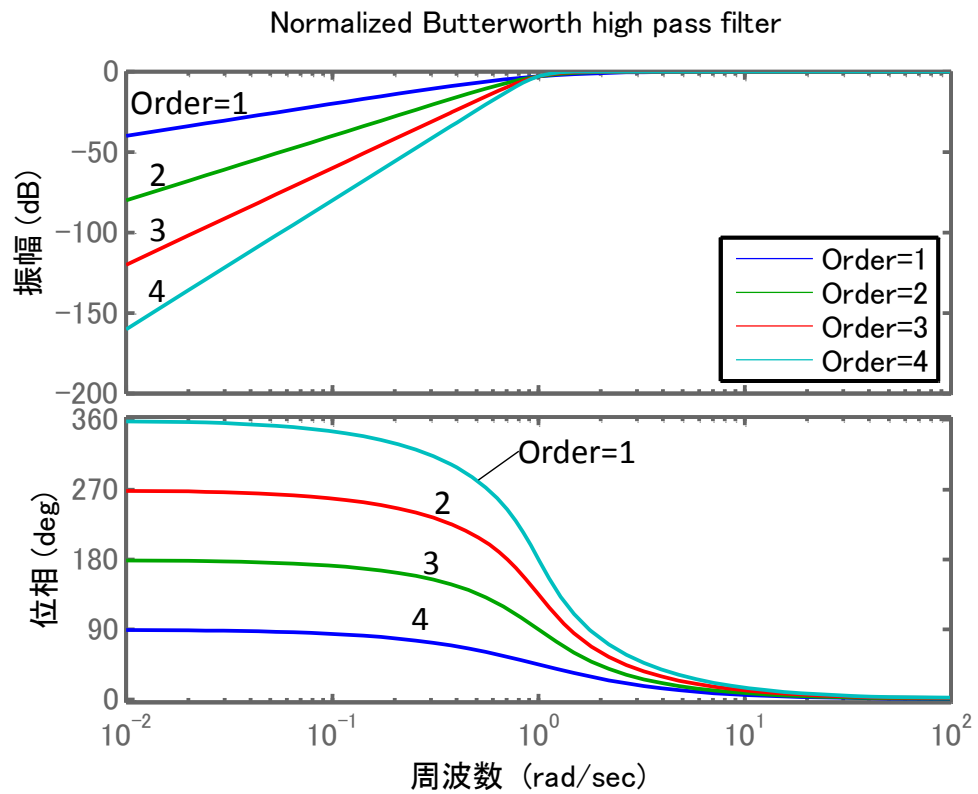


図 4-4 正規化バタワースハイパスフィルタのボード線図

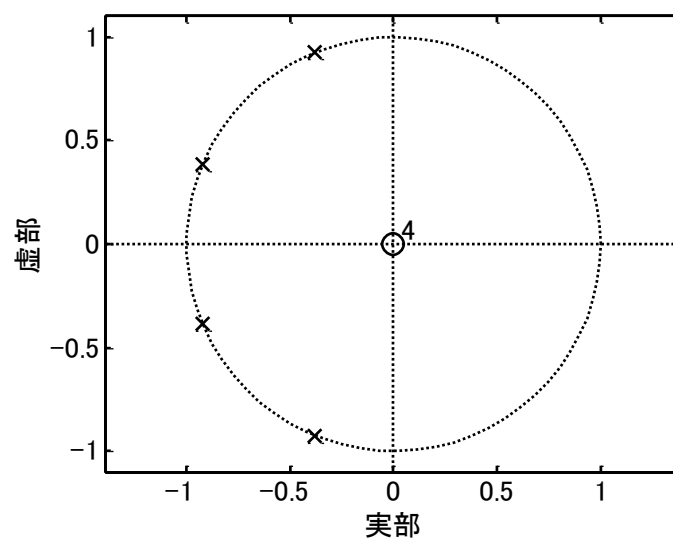


図 4-5 正規化バタワースハイパスフィルタの極と零点（4次）

4.1.3 バタワースバンドストップフィルタ

カットオフ中心周波数 ω_c 、カットオフバンド幅 $B(\text{rad/s})$ 、選択度 $Q = \omega_c/B$ が与えられた時、正規化バタワースローパスフィルタに $s \rightarrow 1/Q \left(s + \frac{1}{s}\right)$ を代入すると、バンドストップフィルタが得られる。式(4-6)に示す2次正規化バタワースローパスフィルタの伝達関数 $H_{Sk}(s)$ に $s \rightarrow 1/Q \left(s + \frac{1}{s}\right)$ を代入すると、次の4次バンドストップフィルタが得られる。

$$H_k(s) = \frac{(s^2 + 1)^2}{s^4 + \frac{2}{Q} \cos \varphi_k s^3 + \left(\frac{1}{Q^2} + 2\right) s^2 + \frac{2}{Q} \cos \varphi_k s + 1} \quad (4-22)$$

この分母=0 の解が、バンドストップフィルタの極となる。分母の式は s^2 について係数が対称となることから、式を次のように置くことができる。

$$H_k(s) = \frac{(s^2 + 1)^2}{(s^2 + 2a_k \cos \eta_k s + a_k^2) \left(s^2 + \frac{2}{a_k} \cos \eta_k s + \frac{1}{a_k^2}\right)} \quad (4-23)$$

この分母=0 の解は $s_k = a_k e^{\pm j\eta_k}$, $\frac{1}{a_k} e^{\pm j\eta_k}$ で表される。この式の分母を $H_{Dk}(s)$ として展開すると、

$$\begin{aligned} H_{Dk}(S) = & s^4 + 2 \left(a_k + \frac{1}{a_k}\right) \cos \eta_k s^3 \\ & + \left(a_k^2 + \frac{1}{a_k^2} + 4a_k \cos^2 \eta_k\right) s^2 \\ & + 2 \left(a_k + \frac{1}{a_k}\right) \cos \eta_k s^2 + 1 \end{aligned} \quad (4-24)$$

となる。これと式(4-22)は同じものであるから、その係数には次の関係が成り立つ。

$$\begin{cases} \frac{1}{Q} \cos \varphi = \left(a_k + \frac{1}{a_k}\right) \cos \eta_k \\ \frac{1}{Q^2} + 2 = a_k^2 + \frac{1}{a_k^2} + 4a_k \cos^2 \eta_k \end{cases} \quad (4-25)$$

この方程式を解き、 a_k と $\cos \eta_k$ を φ_k と Q で表すと次のようになる。

$$a_k = \frac{\sqrt{1+4Q^2}}{2\sqrt{2}Q} \left(\sqrt{\frac{1-4Q^2}{1+4Q^2}} + C + \sqrt{1+C} \right) \quad (4-26)$$

$$\cos \eta_k = \frac{\sqrt{1+4Q^2}}{2\sqrt{2}Q} \sqrt{1-C} \quad (4-27)$$

$$\text{ただし、} C = \sqrt{1 - \cos^2 \varphi_k \left(\frac{4Q}{1+4Q^2} \right)^2} \quad (4-28)$$

これを式(4-23)に当てはめれば、バンドストップフィルタの係数が得られる。さらにこの係数を式(4-15)に代入することで、 z 変換の係数が得られる。

また、1次のローパスフィルタの式(4-8)をバンドストップフィルタに変換すると、

$$H(s) = \frac{s^2 + 1}{s^2 + \frac{1}{Q}s + 1} \quad (4-29)$$

となる。

4.1.4 バタワースバンドパスフィルタ

バンドパスフィルタは、バンドストップフィルタの分子のみが変わる。2次バタワースフィルタの伝達関数

$$H_k(s) = \frac{1}{s^2 + 2 \cos \varphi_k s + 1} \quad (4-30)$$

に $s \rightarrow Q \left(s + \frac{1}{s} \right)$ を代入して得られるバンドストップフィルタは次の通りになる。

$$H_{pk}(s) = \frac{\left(\frac{1}{Q}s \right)^2}{s^4 + \frac{2}{Q} \cos \varphi_k s^3 + \left(\frac{1}{Q^2} + 2 \right) s^2 + \frac{2}{Q} \cos \varphi_k s + 1} \quad (4-31)$$

式(4-22)と比較してわかるとおり、バンドパスフィルタについては、バンドストップフィルタの分子が $s^2 + 1$ から $\frac{1}{Q}s$ に入れ替わっただけである。

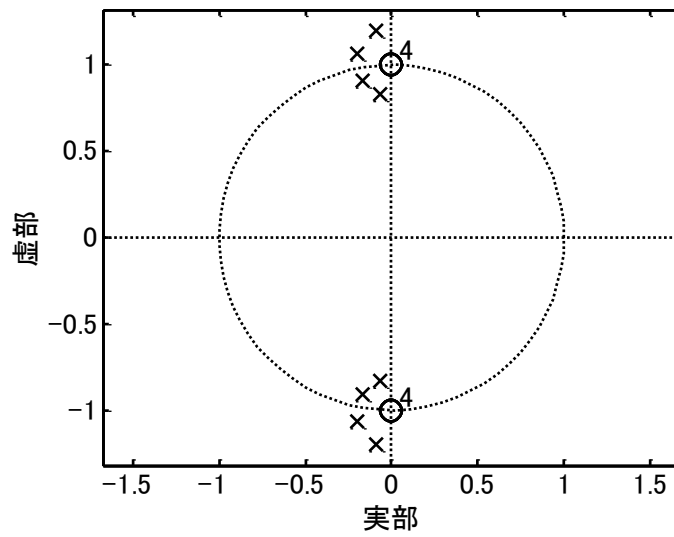


図 4-6 正規化バタワースバンドストップフィルタの極と零点 (4 次、バンド幅 0.4)

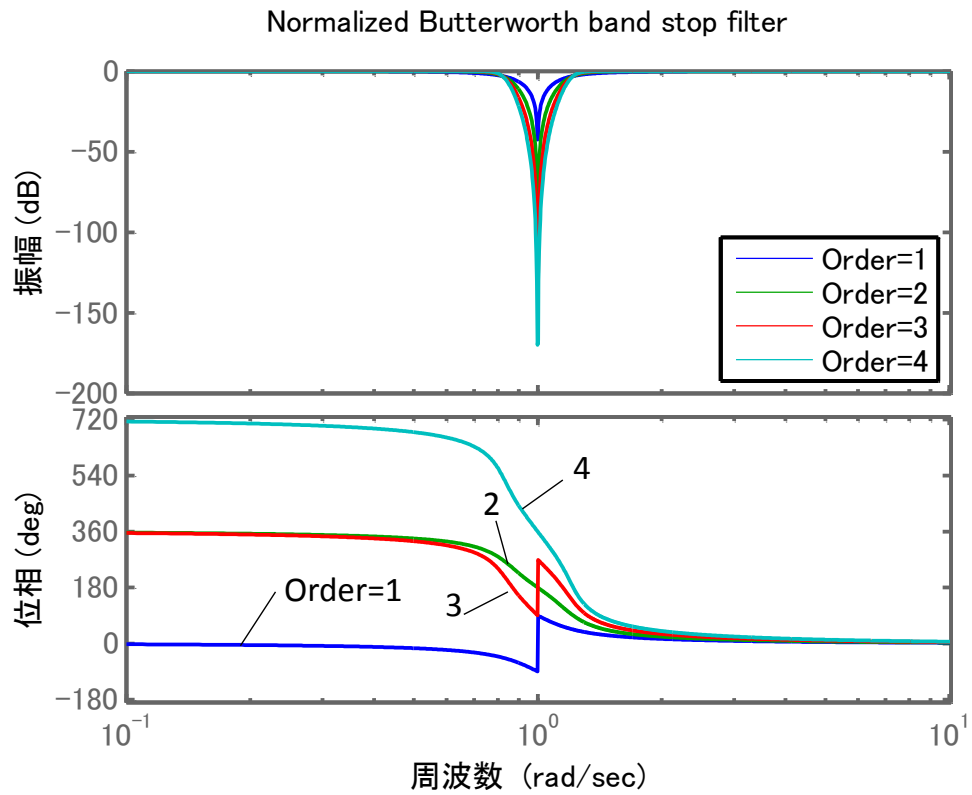


図 4-7 正規化バタワースバンドストップフィルタのボード線図 (バンド幅 0.4)

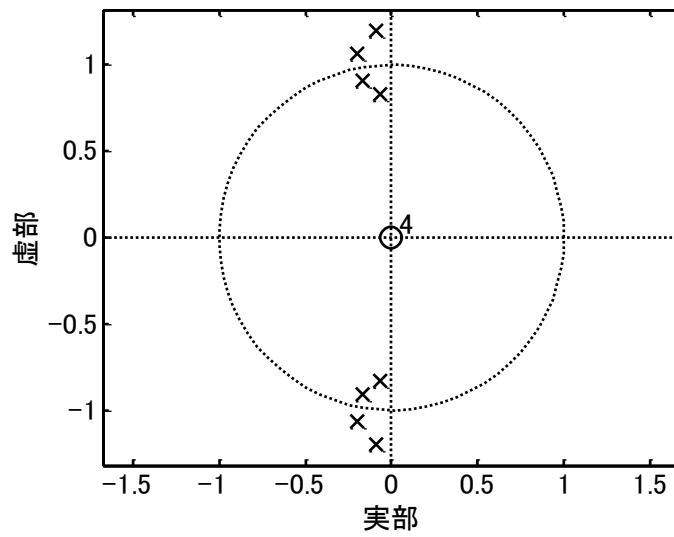


図 4-8 正規化バタワースバンドパスフィルタの極と零点 (4 次、バンド幅 0.4)

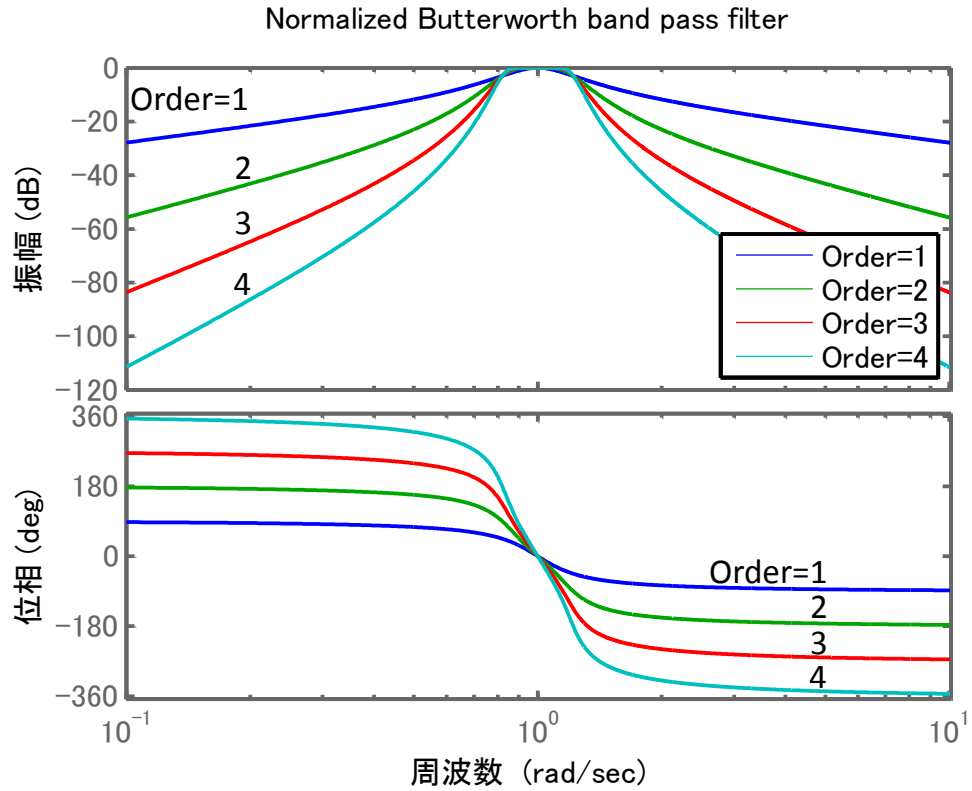


図 4-9 正規化バタワースバンドパスフィルタのボード線図 (バンド幅 0.4)

4.2 デジタルフィルタの実装

4.2.1 2次形式によるプログラミング

前章で求められたデジタルフィルタをプログラム上に実装する。入力 x に対し 2 次フィルタを適用した応答式の z 変換を

$$Y(z) = H(z)X(z) \quad (4-32)$$

とする。これを展開するとサンプリングされた入力時系列 x_n と出力 y_n の関係は式(4-14)より

$$y_n(1 + B_1z^{-1} + B_2z^{-2}) = x_n(A_0 + A_1z^{-1} + A_2z^{-2}) \quad (4-33)$$

となる。 z 変換の性質より $u_nz^{-1} = x_{n-1}$ であるから、式(4-33)は

$$y_n = A_0x_n + A_1x_{n-1} + A_2x_{n-2} - B_1y_{n-1} - B_2y_{n-2} \quad (4-34)$$

と表され、過去の入出力から現在の出力値 y_n が得られる。

さらに、これをプログラム上で計算回数と必要メモリを最小化するように整理する。中間変数 u_{0n}, u_{1n} を用いて式(4-34)を分解すると、次のようになる。

$$\begin{cases} y_n = u_{0n}z^{-1} & + A_0x_n \\ u_{0n} = u_{1n}z^{-1} - B_1y_n & + A_1x_n \\ u_{1n} = & -B_2y_n + A_2x_n \end{cases} \quad (4-35)$$

図 4-10 は式(4-35)をブロック線図で表したもので、この形式は「直接形Ⅱ転置形」と呼ばれる。図中の z^{-1} はプログラム上でバッファメモリ（遅延バッファ）として表現される。つまり、2 次フィルタの出力を計算するには、積算 5 回と加算 4 回の実施とメモリ 2 ブロックの用意をすればよい。これをプログラムで記述するとプログラム 4-1 のようになる⁵。

⁵ C#, Visual Studio2015 で作成したサンプルコードを GitHub よりダウンロードできる。
(<https://github.com/nakajima-hro/ButterworthFilter>)

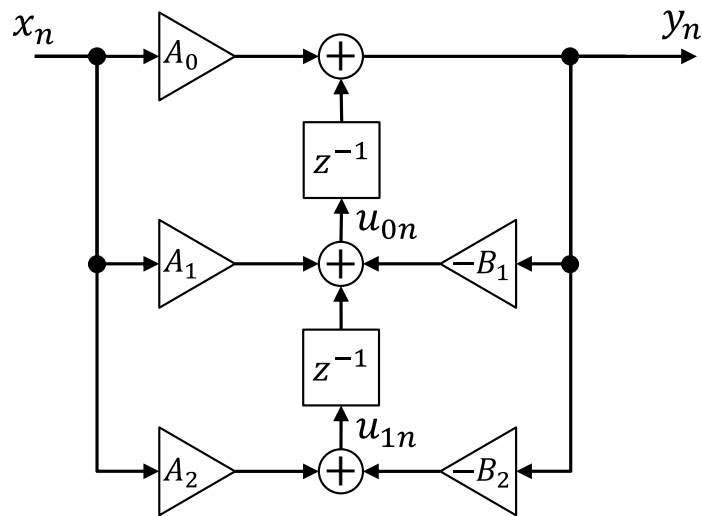


図 4-10 2 次フィルタのブロック線図（直接形 II 転置形）

プログラム 4-1 単純な 2 次フィルタクラスの例(C#)

```
class Filter
{
    public double a[3]; //z 変換された伝達関数の分子係数
    public double b[3]; //伝達関数の分母係数
    private double u[2]; //遅延バッファ（初期設定するなら、public としてよい）
    :
    // 2 次フィルタ（直接形 II 転置形、Transposed-Direct-Form-II）
    // 分子係数 a、分母係数 b を初期設定して用いる。
    // y = Filter(u);
    // x:入力
    // y:フィルタ後の出力
    public double Filter(double x)
    {
        double y = u[0] + a[0]*x;
        u[0] = u[1] - b[1]*y + a[1]*x; //u[0]は次の計算で用いられる。
        u[1] = -b[2]*y + a[2]*x; //u[1]は次の計算で用いられる。
        return y;
    }
}
```

4.2.2 片方向フィルタと双方向フィルタ

一般に、ローパスフィルタやハイパスフィルタをかけると、位相が変化する。端的な例を言えば、ローパスフィルタを正の時間方向に沿って掛けると時間が遅れ、例えば波の山や谷の時刻が遅くなる。この影響はフィルタの性質やカットオフ周波数に依存するが、位相の影響を取り除くには、時間の正方向にフィルタを掛けた後に、逆方向からもう一度フィルタを掛けるとよい。これを双方向フィルタといい、位相の遅れや進みが打ち消される。ただし、このフィルタは時間の逆方向、つまり未来から過去へフィルタを掛けることから、リアルタイムでは用いる事はできない。データの収集が終わった後に、位相変化をさせたくない場合に使用することになる。

一方、リアルタイムで用いる場合には、ことに人間の情報を見る場合には、位相の遅れよりも出力の見やすさ（フィルタ通過後のきれいな信号）のほうが状況判断に重要なので、リアルタイムグラフ表示には片方向フィルタで十分である。

データ保存には生データを用い、リアルタイムグラフはフィルタ通過後のデータを、保存後の処理には双方向／片方向のフィルタ処理を行うというのがよい。

4.3 アナログセンサのサンプリングに対する注意点

4.3.1 エイリアシング

生体信号に限らず、センサデータを A/D 変換サンプリングする際にはサンプリングレートに注意を払う必要がある。データ信号にはノイズを含め多くの周波数が混在しているが、その信号の最大周波数がサンプリングレートの $1/2$ を越えてはいけな~~い~~。これは、サンプリング定理と呼ばれるデジタル信号処理の基本原則で、もしそのまま信号をデジタル化すると、後工程で影響を除去することはできない（例えば、ある周期を持つサイン波と同じ周期でサンプリングを行うと、その信号はデジタル領域では一定電圧の信号になってしまい、元のサイン波の情報は失われる）。そのため、信号を取り込む前には、必ずアナログ回路のローパスフィルタでサンプリング周波数の $1/2$ 以上の周波数を十分に減衰させなければならない。

4.3.2 入力インピーダンス

次に、A/D 変換器につなぐアナログ回路の入力インピーダンスはできるだけ小さくすべきである。A/D 変換器にはごく小さな電気容量のコンデンサが内蔵されており、これに一定

時間でチャージされた後の電圧がデジタル変換される。この時、アナログ電圧の入力インピーダンスが大きい（端的に言えば電気抵抗が大きい）と、コンデンサに十分なチャージがされず、変換後の値が不正確になる事がある。特に短い間隔でサンプリングするときは注意が必要である。A/D 変換器のマニュアルに記載されている推奨設定を理解し、入力インピーダンスや A/D 変換時間を設定する必要がある。特に、心電計は 100～1000Hz と高速でサンプリングするため、気をつけなければならない。

第5章 開発事例

5.1 乳牛用血中カルシウム濃度測定システム⁶

乳牛用血中カルシウム(Ca)濃度測定システムは(図 5-1)、分娩前後の乳牛から計測された心電位の解析結果から統計解析により血中 Ca 濃度を推定するものである [6]。血中の Ca 濃度が低下すると心電位の波形が延長する現象を利用し、心電位からその濃度を測定することができる。心電位のわずかな変化をもとに推定するため、心電位計測には高い精度が求められる。

このシステム用のワイヤレス心電計の試作においては Shimmer3 のベースモジュールを用いた。アナログ周波数フィルタ(ローパスフィルタ、ハイパスフィルタ)を内蔵した高精度心電アンプを Shimmer3 の拡張ポートにある A/D コンバータに接続し、測定を行っている。Shimmer3 はファームウェアがよく整備されており、本システムの試作においては標準のものをそのまま用いた。Windows 用アプリケーションは LabVIEW を用いて開発をした。

データをマイクロプロセッサ(MPU)に取り込んだ後は、Bluetooth 通信により PC へデータ送信を行う。ここでは、できるだけ生データに近い状態でデータを送るため、特にデータ加工は行わない。データの通信量が多く BLE では通信速度が足りないため、ここでは Bluetooth 2.1 の SPP プロファイル(Serial Port Profile)を用いている。

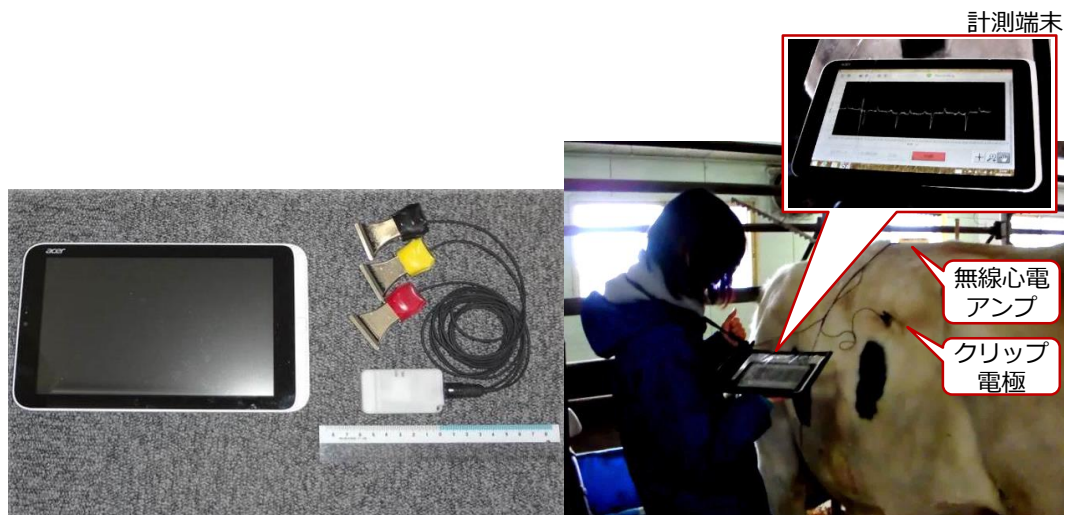


図 5-1 携帯型乳牛血中カルシウム濃度計測システムおよび計測風景

⁶ 本研究は、戦略的情報通信研究開発推進事業 (SCOPE)「地域 ICT 振興型研究開発」の研究助成によって行われた。http://www.soumu.go.jp/main_sosiki/joho_tsusin/scope/

PC がデータを受信した後は、プログラム上で表示部と保存解析部にデータが振り分けられる（図 5-3）。保存後のデータは、解析前にローパス、ハイパス、ノッチの双方向フィルタをかける。双方向フィルタとは、時間の進む方向にフィルタをかけた後、もう一度時間の戻る向きにかけるフィルタである。このようにフィルタをかけると、フィルタによる位相のずれが打ち消され、波の進み遅れがなくなるため、ピークやしきい値の時刻精度が重要となる場合に用いる事が多い。ただし、双方向フィルタは時間を戻すようにかけるため、リアルタイム計算で用いる事はできない。

乳牛の心電位の計測は、人間の場合よりはるかにノイズが混入しやすい。まず、牛は皮膚に体毛が密集しているため入力インピーダンスが非常に大きくなり、ハムノイズが混入しやすい。また、牛がじっとしていることもないため筋電位の混入や体動による基線揺らぎも大きい。そのため、ノイズ除去には必然的に強いフィルタを用いる事になる。

心電位の計測フィルタリング手順は次の通りである。心電センサで計測された心電位はアンプ内でローパス・ハイパスのアナログフィルタを通過したのち A/D 変換でサンプリングされる（図 5-2）。データ送信し PC が受診した後、心電位にさらにフィルタをかけていく（図 5-3）。ここではまずローパスフィルタをかける。心電位のカットオフ周波数はサンプ

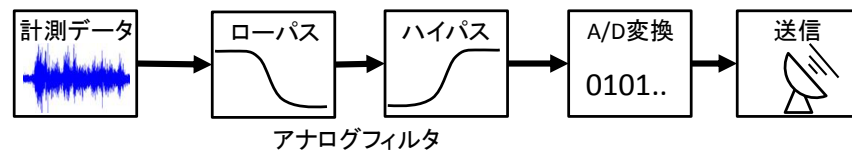


図 5-2 電極から MPU までのデータ処理過程

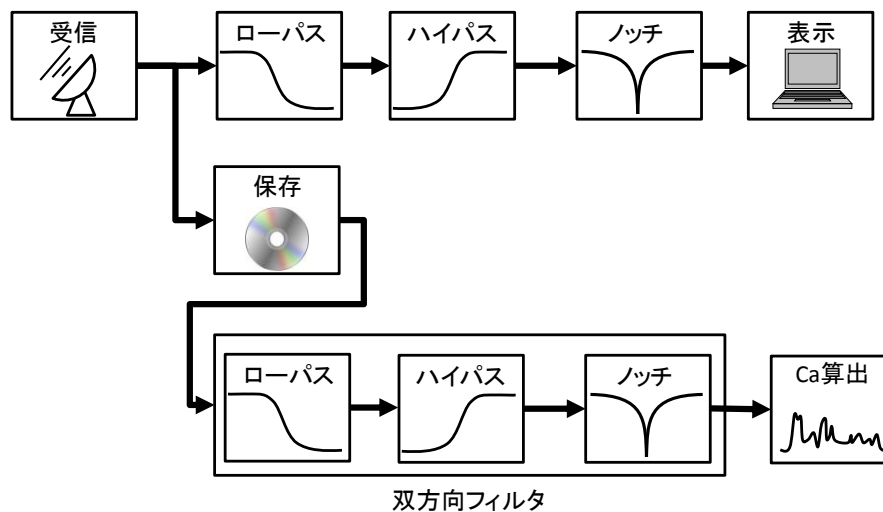


図 5-3 PC 側における心電位のデータ処理過程

リングレートと解析内容によって決まるが概ね 30～150Hz の間であり、詳細な解析を行う場合は数百 Hz とする場合もある。なお、心拍間隔を検出する程度であれば、30Hz でも十分である。

ローパスフィルタをかけた後は、ハイパスフィルタで基線揺らぎを除去する。通常はカットオフ周波数 0.1～0.5Hz とするが、乳牛の場合はさらに高い値でかけ、十分に体動の影響を除去する。ハムノイズは電源周波数（東日本では 50Hz、西日本では 60Hz）を中心周波数としたバンドストップフィルタをかける。50～60Hz 帯は心電位の主要な周波数成分も含むため、バンドストップフィルタのバンド幅はできるだけ狭くするべきである。これらのフィルタをかけデータをきれいにしたのち、ピーク検索による心拍間隔検出や波形特徴量の解析などを行う。

図 5-4 に、心電位にフィルタを適用した場合の効果を示す。①の生データに、②ハイパスフィルタをかけた場合、③ローパスフィルタをかけた場合、④バンドストップフィルタをかけた場合、それぞれで除去される周波数成分が異なるのが見て取れる。これらのフィルタをすべて当てはめた場合（図 5-4⑤）、基線揺らぎがなく高周波成分やハムノイズもないクリアなデータが得られる。

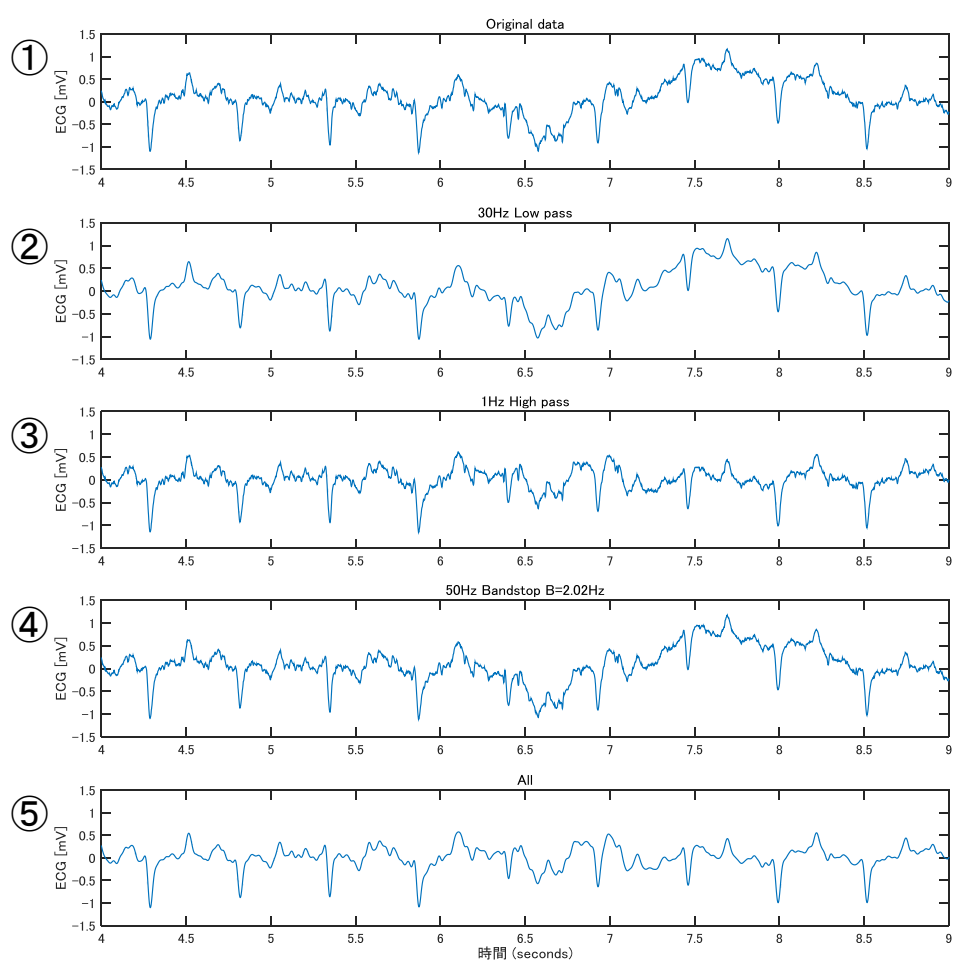


図 5-4 フィルタの種類による心電位の変化

上から①生データ、②ハイパスフィルタのみ（1Hz、双方向 4 次）、③ローパスフィルタのみ（30Hz、双方向 4 次）、④バンドストップフィルタのみ（50Hz、バンド幅 2.02Hz、双方向 4 次）、⑤フィルタをすべてかけた場合。

5.2 ワイヤレス圧力センサの機能試作

ある技術支援において圧力計測装置の機能試作を行った事例を紹介する。この事例では、配管内の圧力データをアナログ電圧で収集し、BLE モジュールを用いてタブレット PC に転送している。

BLE モジュールには Punch Through Design 社の LightBlue®（以下、Bean）を用いた。Bean は BLE 無線モジュールを搭載し、バッテリーは 2032 コイン型リチウムバッテリーのソケットが標準装備されている。ファームウェアは Arduino 言語で記述でき、短期間で機能試作を行う事ができる。本システムでは、Bean に標準搭載された A/D 変換ポートに圧力センサの出力ピンを接続し、10Hz のサンプリングレート、1Hz の 2 次バタワースローパスフィルタで出力電圧を収集、転送している。

PC には Windows10 タブレットを用い、アプリケーションを開発した。転送出力されたデータはフィルタ処理、RAW->圧力の処理を行った後、グラフ表示（ZedGraph）とファイルストリームによるデータ保存を行っている。

プログラム 5-1 に、サンプルプログラムを掲載する⁷。

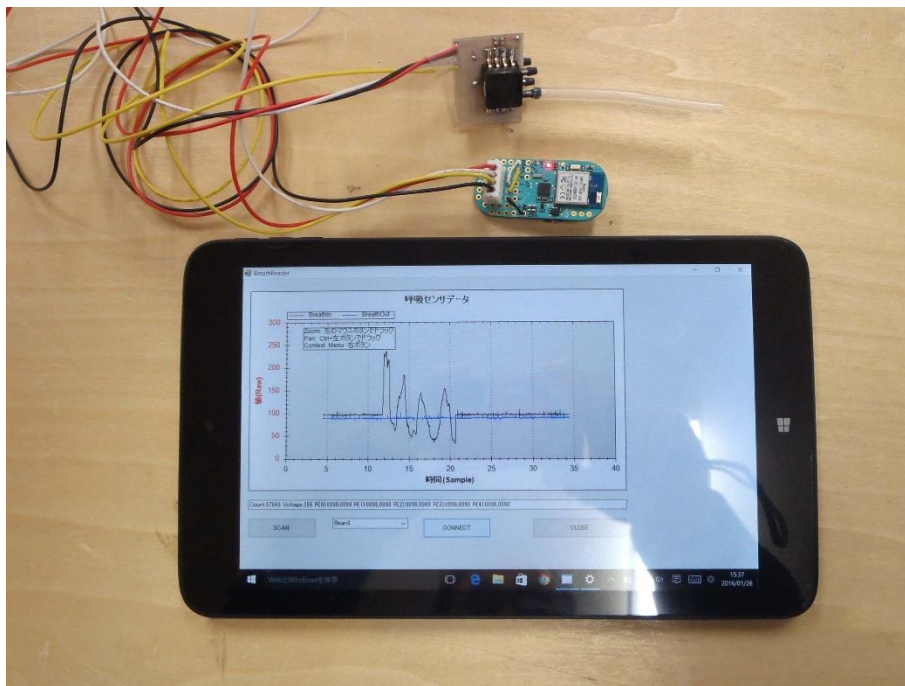


図 5-5 ワイヤレス圧力センサの試作機

⁷ GitHub より Bean ファームウェアと Windows アプリケーションのソースコードをダウンロードできる。（<https://github.com/nakajima-hro/BeanAccReader> , <https://github.com/nakajima-hro/BeanAccReaderApp>）

プログラム 5-1 LightBlue® Bean を用いたローパスフィルタ付き圧力センサ

```

#include <MsTimer2.h>

//データバッファ
uint8_t bleBuffer[6];

//A/D 変換データ
uint16_t an0;

//タイマ割込発生数カウンタ
uint16_t count = 0;

//割込中フラグ
bool readFlag = false;

//LED 点灯
static bool led = true;

// タイマ割り込みによるアナログポート読み込み
void ReadAD()
{
    //割込許可（タイマを止めないため）
    interrupts();
    readFlag = true;
    count++;

    //圧力センサ電圧の読み込み
    an0 = analogRead(A0);

    //LED 点滅
    if (count % 10 == 0) {
        led = !led;
        if (led) Bean.setLed(32, 0, 0);
        else Bean.setLed(0, 0, 0);
    }
}

void setup()
{
    //LED をオン
    Bean.setLed(32, 0, 0);

    //100ms ごとに割込み A/D 読取を起動。
    MsTimer2::set(100, ReadAD);
    MsTimer2::start();
}

// the loop routine runs over and over again forever:
void loop()
{
    //read フラグ（読み取りデータあり）が立っていればデータ送信
    if (readFlag) {

        //フラグリセット
        readFlag = false;

        //センサ情報をバッファに格納

        int16_t y = (int16_t) Filter(an0);
    }
}

```

```

//バッファカウンタ
uint16_t bufferCounter = 0;
bleBuffer[bufferCounter++] = count & 0xFF;
bleBuffer[bufferCounter++] = count >> 8;
bleBuffer[bufferCounter++] = y & 0xFF;
bleBuffer[bufferCounter++] = y >> 8;
bleBuffer[bufferCounter++] = an0 & 0xFF;
bleBuffer[bufferCounter++] = an0 >> 8;

//BLE 送信
Bean.setScratchData(1, bleBuffer, sizeof(bleBuffer));

//BeanLoader のバーチャルシリアルによる出力
Serial.print(count, DEC);
Serial.print("\t");
Serial.print(y, DEC);
Serial.print("\t");
Serial.print(accel.xAxis, DEC);
Serial.print("\n");
}
}

// 2次フィルタ（直接形 II 転置形、Transposed-Direct-Form-II）
// 分子係数 a、分母係数 b を初期設定して用いる。
// y = Filter(x);
// x:入力
// y:フィルタ出力
double Filter(double x)
{
    //z 変換された伝達関数の分子係数（2次ローパス、無次元カットオフ：0.2（カットオフ 1Hz））
    static const double a[3] =
        {0.06745527388907, 0.13491054777814, 0.06745527388907};
    //伝達関数の分母係数
    static const double b[3] =
        {1.0, -1.14298050253990, 0.41280159809619};
    //遅延バッファ
    static double u[2] = {0, 0};

    //フィルタ出力の計算
    double y = u[0] + a[0]*x;
    u[0] = u[1] - b[1]*y + a[1]*x;
    u[1] = -b[2]*y + a[2]*x;
    return y;
}

```

5.3 筋電位計測によるバスドラム用キックペダルの評価

この事例は、バスドラム用新型キックペダルを人間工学の観点から身体負担を評価したものである [7]。ある企業が開発した踏み力の軽い新機構のドラムペダルに対し、最も疲労の大きい脚部の筋電位を計測し、どの程度筋活動が低減したかを評価したものである。計測は、前脛骨筋（下腿前部にあり、つま先を上げる）と腓腹筋（下腿高部にあり、つま先を下げる）の2本の筋に対して行いった。

筋電位の計測処理においてフィルタリングの手順は基本的に心電位と同じである。しかし、筋電位と心電位では、周波数フィルタの区間が大きく異なる。心電位では一般的にローパスフィルタのカットオフ周波数が 30～150Hz、ハイパスでは 0.1～0.5Hz となるが、筋電位はローパスを 5～30Hz、ハイパスを 200～500Hz に設定する。ハムノイズが大きい場合に交流電源の周波数を中心周波数とするノッチフィルタをかける過程は心電位と同様である。

フィルタリング処理の後、筋活動量（筋活動の大きさ）を算出する。筋活動量を算出するには場合は、次に挙げる二つの方法のいずれかで筋電位を平滑化することが多い。一つ目の二乗平均平方根（Root Mean Square; RMS）は、式(5-1)、ブロック線図 5-7 に示すように、サンプリングデータを 2 乗して移動平均しその平方根をとるもので、統計学で言うところの標準偏差と同じ計算方法である。

$$RMS(n) = \sqrt{\frac{1}{m} \sum_{k=-(m-1)/2}^{(m-1)/2} x_{n-k}^2} \quad (5-1)$$



図 5-6 ドラムペダルの筋負荷測定試験

二つ目は整流移動平均（Average Rectified Value; ARV）で、単に移動平均と呼ばれることが多い。こちらは式(5-2)、ブロック線図 5-8 に示す通り、RMS のようにデータの 2 乗値ではなく絶対値を取り移動平均するもので、統計学の平均偏差と同じ計算方法である。この二つは計算方法が異なるが、両者の傾向は似通ったものとなる。

$$ARV(n) = \frac{1}{m} \sum_{k=-\frac{m-1}{2}}^{\frac{m-1}{2}} |x_{n-k}| \quad (5-2)$$

移動平均の区間は、動作の鋭さや求める時間方向の細かさによって変わるが、概ね 50-200ms の範囲で平滑するのが一般的である。オフラインでデータを計測する場合には中央移動平均、現時刻を中心として前後の点を移動平均するのが一般的である。後方移動平均（現時刻から過去の点を平均する）では、時間が窓幅の 1/2 だけ応答が遅れるためである。

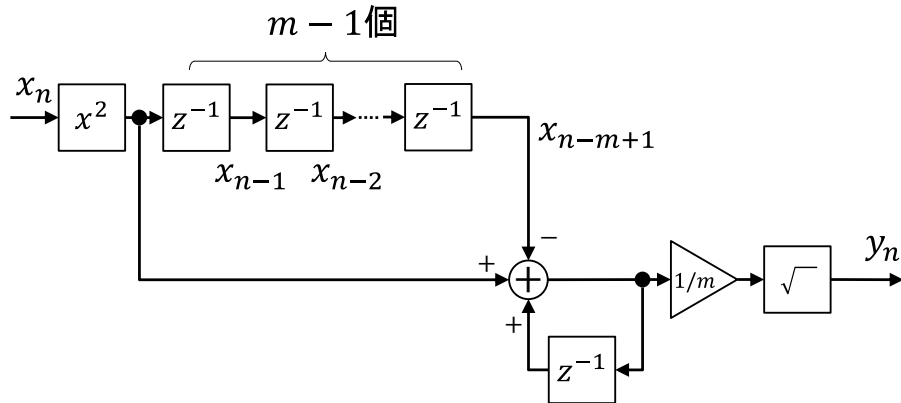


図 5-7 二乗平均平方根(Root Mean Square; RMS)のブロック線図

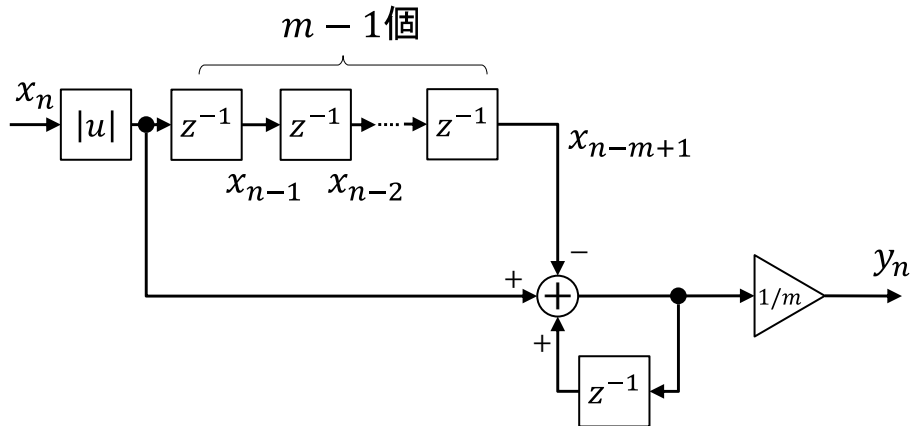


図 5-8 整流移動平均(Averaged Rectified Value; ARV)のブロック線図

ただし、算出された値の時刻を現時刻ではなく窓幅の $1/2$ だけ遅らせるのであれば、中央移動平均と同値となる。

図 5-9 に筋電位の処理過程を示す。1000Hz でサンプリングされた生データ①から順に②200Hz-4 次バタワースローパス、③20Hz-4 次ハイパス、④全波整流（絶対値）、⑤50ms 移動平均をかけ、筋活動量を算出した。このピーク値を比較することで、筋活動量の大きさが比較できる。

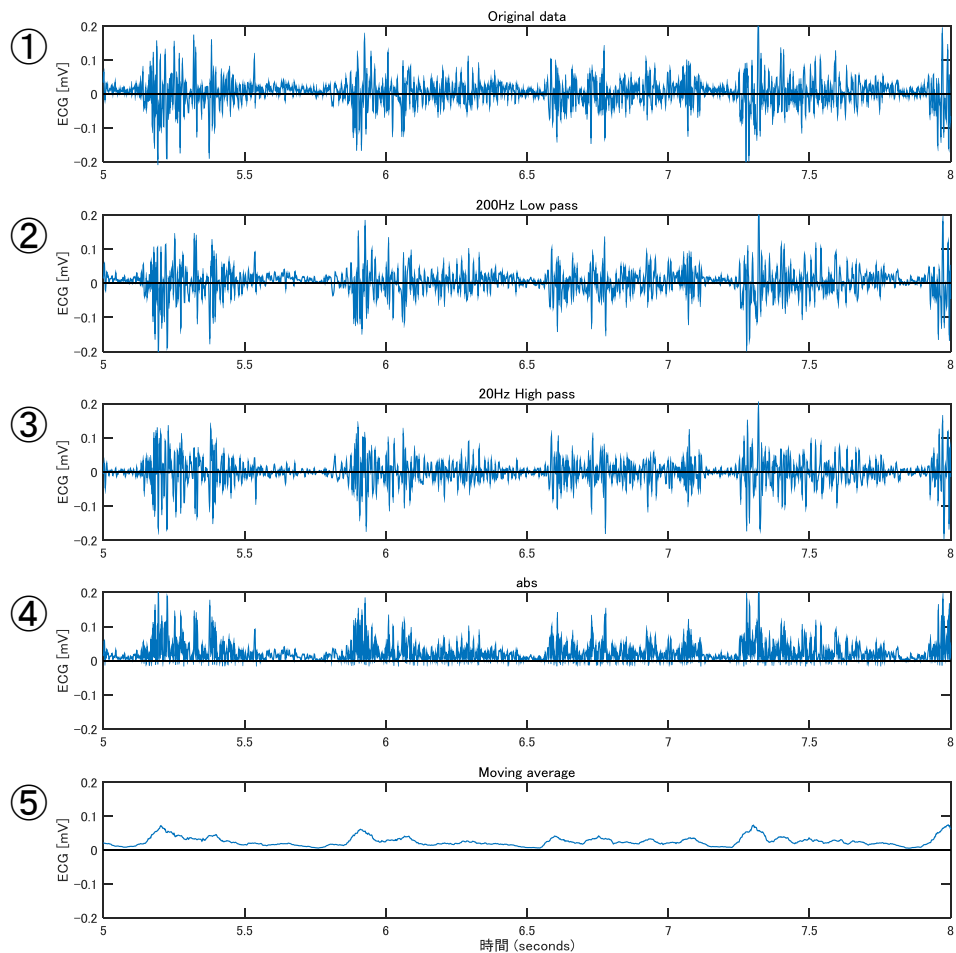


図 5-9 筋電位の整流移動平均値。上から順にフィルタをかけていった状態。①生データ→②ローパスフィルタ（200Hz、片方向 4 次）→③ハイパスフィルタ（20Hz、片方向 4 次）→④全波整流（絶対値）→⑤中央移動平均（51 点（50ms 幅））。

参考文献

- [1] K. Townsend, C. Cufi, Akiba , R. Davidson, Bluetooth Low Energy をはじめよう, オライリージャパン, 2015.
- [2] Punch Through Design, “ LightBlue Bean, ” [オンライン]. Available: <https://punchthrough.com/bean>.
- [3] 武部幹, デジタルフィルタの設計 (デジタルテクノロジーシリーズ (4)), 東海大学出版会, 1986.
- [4] 三上直樹, はじめて学ぶデジタルフィルタと高速フーリエ変換, CQ 出版社, 2005.
- [5] 三谷政昭, デジタルフィルタデザイン (デジタル信号処理シリーズ 3), 昭晃堂, 1987, p. 111.
- [6] 北海道立総合研究機構 工業試験場, 技術支援成果事例集 2015, 北海道立総合研究機構 工業試験場, 2015.
- [7] 北海道立総合研究機構 工業試験場, 技術支援成果事例集 2012, 第 巻 2012, 北海道立総合研究機構 工業試験場, 2012.
- [8] 木塚正博、増田正、木竜徹、佐渡山亜兵, 表面筋電図, 東京電機大学出版局, 2006.

謝 辞

本冊子は、北海道立総合研究機構札幌技術研究本部工業試験場 経常研究「人間計測応用製品の試作支援ツールの開発」（平成 26～27 年度実施）によって作成された。本ガイドの作成に協力頂いた皆様に、この場を借りて感謝の意を表す。

2016 年 3 月

北海道立総合研究機構 産業技術研究本部 工業試験場

中島 康博

人間を測るためのワイヤレスセンサ開発

ー 人を測ってモノを創る ー

2016 年 3 月 26 日 第 0 版発行

著 者 中島康博・桑野晃希・前田大輔・橋場参生

発行者 北海道立総合研究機構 産業技術研究本部 工業試験場 製品技術部

デザイン・人間情報グループ 人間情報応用分野