機械学習のための
信号処理

MathWorks
アプリケーションエンジニアリング部
竹本佳充

© 2016 The MathWorks, Inc.

Scania Emergency Braking System

Scania's Analytics-Driven Emergency Braking System

- Sensor fusion algorithms developed using machine learning trained with 80TB of driving data
- Algorithms deployed on a production embedded system using Model-Based Design

"Sensor fusion brings the sensor data together and, algorithmically, processes the combined sensor data to understand the real-time situation and apply the right control strategy to ensure braking occurs when necessary."

Celestion社
楽器用ラウドスピーカーの分類に機械学習技術を適用

課題
楽器用ラウドスピーカーの評価における、
マニュアルでの官能評価プロセスを自動化

ソリューション
ラウドスピーカーの高精度かつ高確度な分類のための
機械学習アルゴリズム構築にMATLABを使用

結果
- プロセスを自動化することにより、マニュアルで工数のかかる作業を数秒で実現
- 多種多様な機械学習および信号処理技術を評価し適用
- エラー率を0.1%程度に低減

Celestion社のラウドスピーカーテスト用リスニングルーム

“MATLABと Statistics and Machine Learning Toolboxの使用により, 一度データを用意すれば, 多種多様な分類手法を数分で評価することができました. もしそれらをマニュアルでコーディングしていたら、数年はかかっていたと思います. ”

Andrew Harper
Celestion

Link to article

1977 Original Recording

2016 with MATLAB

Remastering The Beatles' Only Live Recording

“The Beatles Live at the Hollywood Bowl” Remastered and Remarkable

- Used MATLAB, Signal Processing Toolbox, and related examples already written in MATLAB
- Developed tool that uses spectrum analysis, optimization, and musical modelling

“It doesn’t exist as a software program that is easy to use. It’s a lot of Matlab, more like a research tool. There’s no graphical front end where you can just load a piece of audio up, paint a track, and extract the audio. I write manual scripts, which I then put into the engine to process.”

- James Clarke, Abbey Road Studios

CellScope's Oto consumer otoscope

CellScope's Oto Diagnoses Ear Infections with a Smartphone

- Visual analysis of the eardrum, by comparing with a database of 10,000 expertly classified images
- Developed using a full spectrum of MATLAB tools:
  - Computer vision
  - Machine learning
  - Bioinformatics
  - Parallel computing
  - Code generation

"What is great about having such an attachment is the power to immediately share the image with the patient and better explain to them what you are seeing..."

Agenda

- 背景
- センサー信号の場合
- 音声信号の場合
- 画像の場合
- 関連情報
- まとめ

Agenda

- 背景
- センサー信号の場合
- 音声信号の場合
- 画像の場合
- 関連情報
- まとめ

MATLABによるアナリティクスフロー

アクション

知識

情報

データ

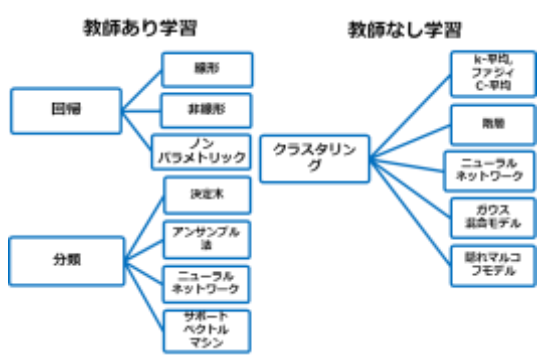
計測器

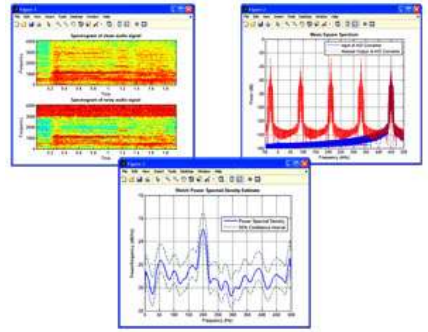
意思決定 & 設計
- レポート & Apps
- 拡張性のある展開
- 設計最適化

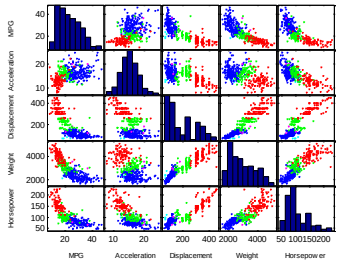
理解
- 分析
- 周波数 & 時間領域
- 予測分析
- 推定

整理
- フィルタリング
- 信号解析
- データ処理
- プロット

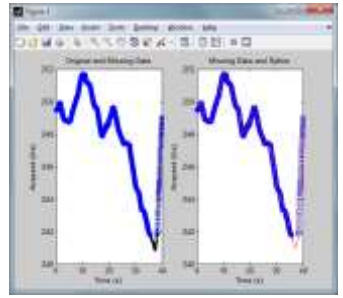
観測
- 検出
- 収集
- 状態
- データ取得







Key	B	Y	Z
1	1.1	0.1	0.2
4	1.4	NaN	NaN
7	1.7	0.7	0.8
9	1.9	NaN	NaN



データアナリティクス・ワークフロー：「機械学習」と「信号処理」

アクション

意思決定 & 設計
- レポート & Apps
- 拡張性のある展開
- 設計最適化

知識

理解
- 分析
- 周波数 & 時間領域
- 予測分析
- 推定

情報

整理
- フィルタリング
- 信号解析
- データ処理
- プロット

データ

観測
- 検出
- 収集
- 状態
- データ取得

計測器

種類	センサー	音	画像
サービス	防災 ヘルスケア 生産ライン 各種故障予知	モバイルアプリ 公共サイン音 異音検査 定量評価	セキュリティ 自動運転 医療診断
新たな知見	固有の周期性 異常値 特異なパターン	音声認識 構文解析 音質	画像認識 オブジェクト分類 トラッキング
特徴量	平均値 標準偏差 AR係数 周波数特性	インパルス応答 フォルマント LPC MFCC	HOG SURF FREAK BRISK
データ収集	各種センサーデバイス 測定器、データ収集I/O テキスト、バイナリ	マイク WAVE FLAC MPEG-4	カメラ JPEG MPEG WMV

機械学習の適用範囲

信号処理の適用範囲

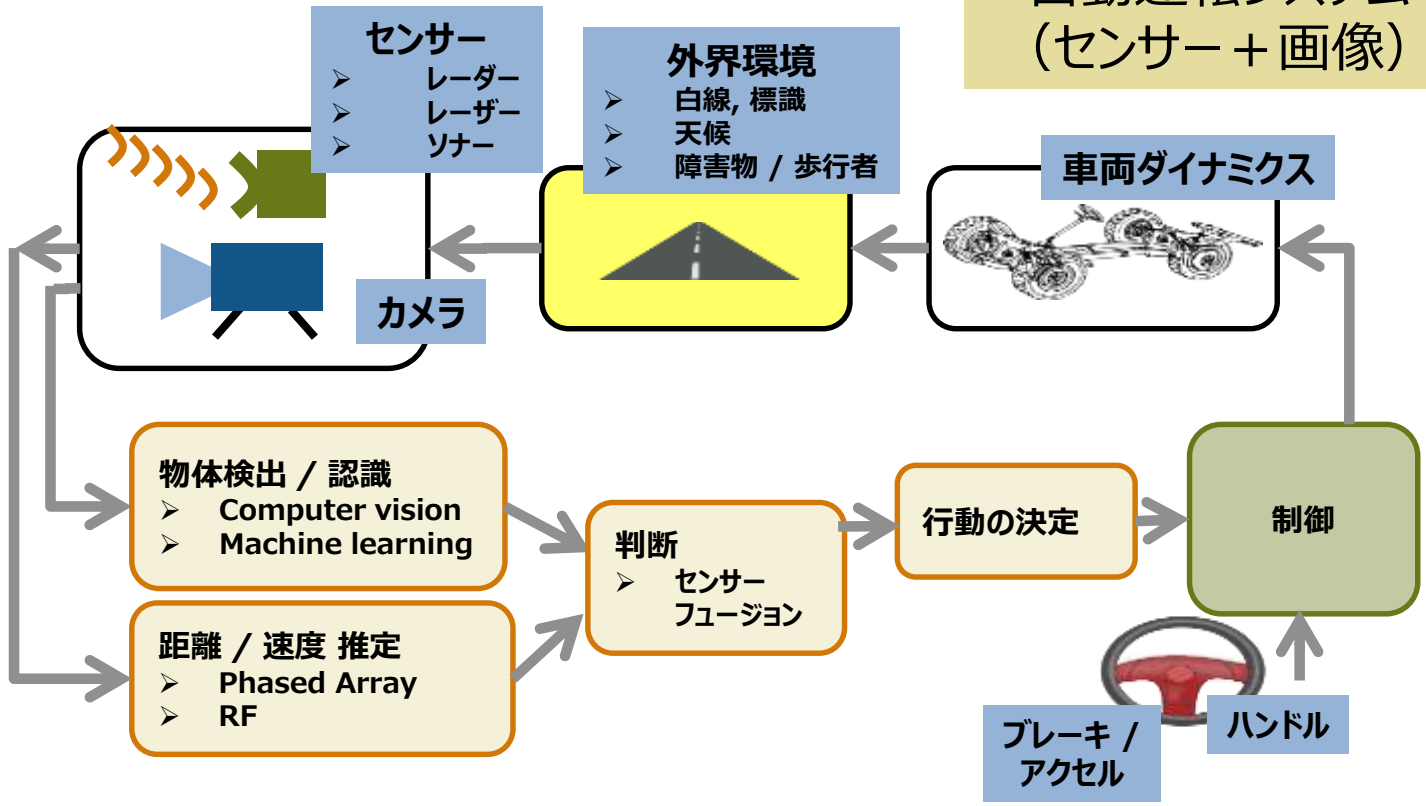
何故、センサー、音、画像？

メディカル分野の例
（音＋画像）

iSonea

… the AirSonea® technology detects wheeze patterns in an **image** created from recorded breath **sounds**...

自動運転システム
（センサー＋画像）



複合的な要素技術の
応用が必要とされる時代

➔ MATLABは
多分野の専門ツールを提供

Agenda

- 背景
- センサー信号の場合
- 音声信号の場合
- 画像の場合
- 関連情報
- まとめ

センサー信号の取得

Data Acquisition Toolbox™

- データ収集ボードへのアクセス
  - アナログI/O
  - デジタルI/O

- サポートベンダー
  - National Instruments など

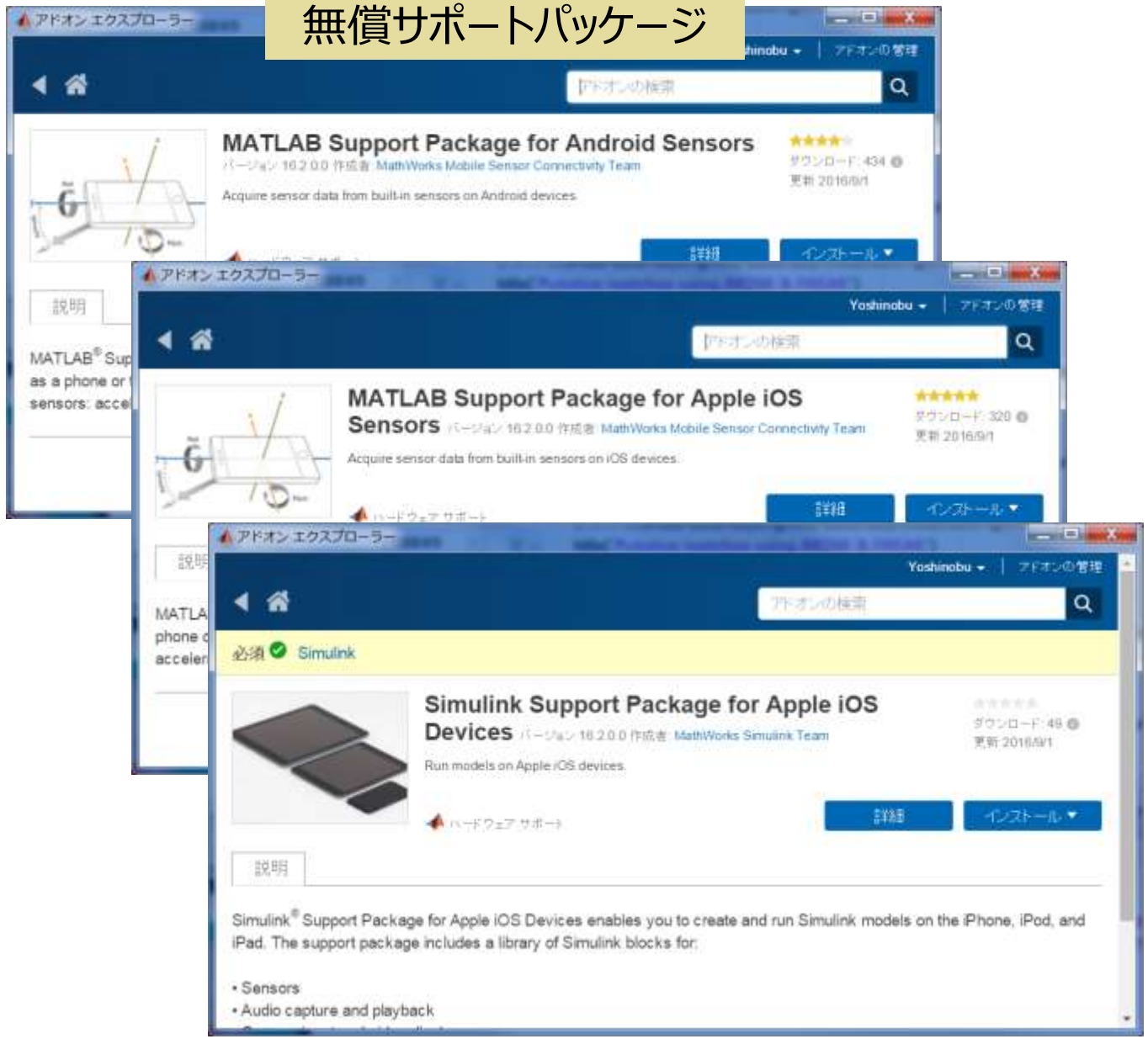
Instrument Control Toolbox™

- 測定器へのアクセス
  - オシロスコープ
  - 信号発生器
  - スペアナ

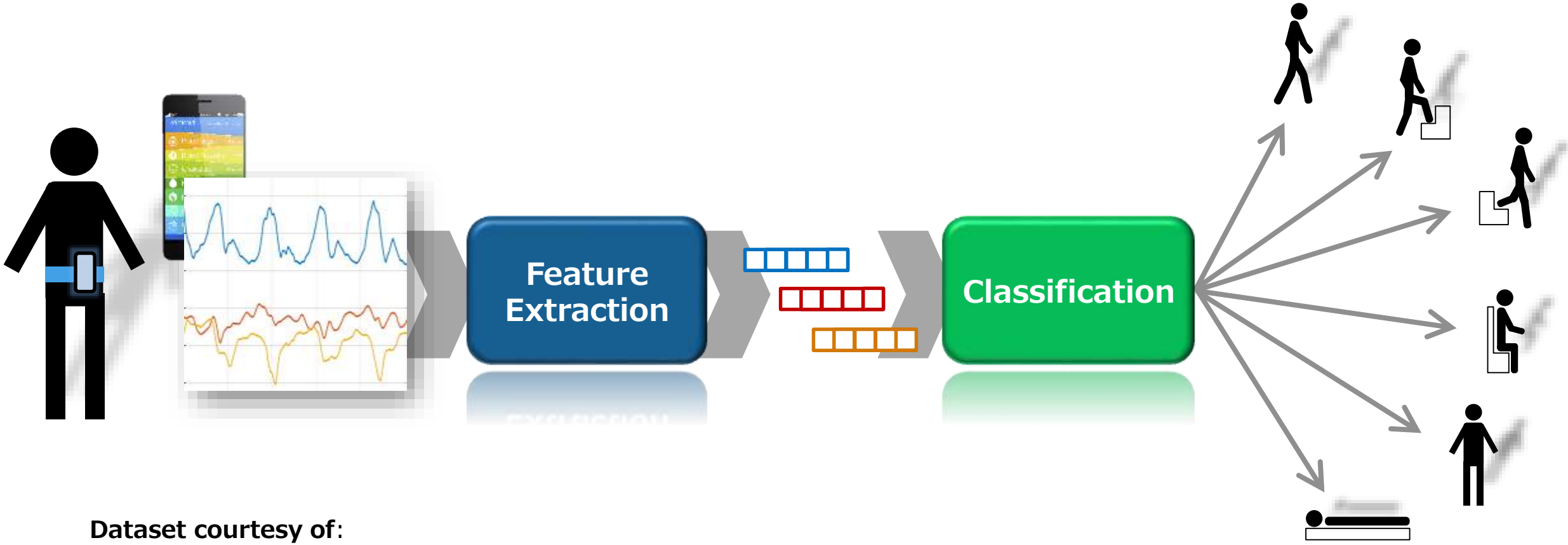
    :

- シリアル通信(RS-232など)

無償サポートパッケージ



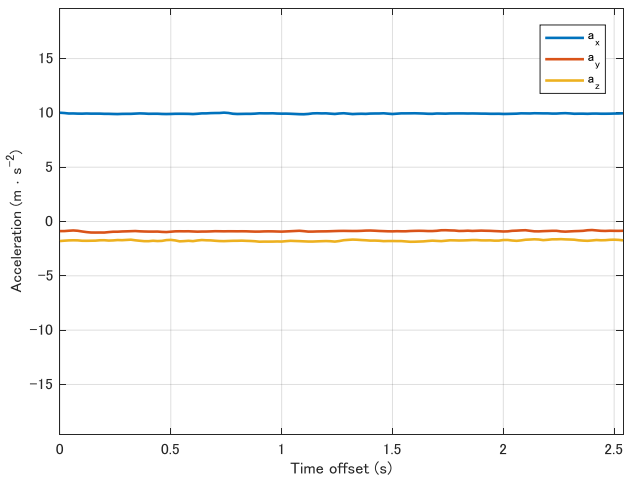
例: 人の活動解析とクラシフィケーション

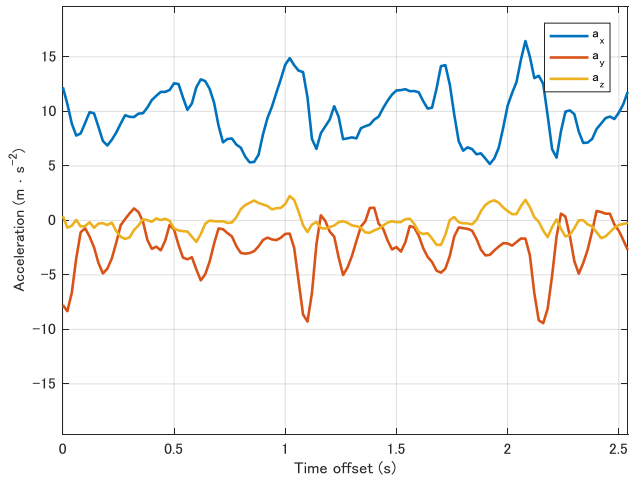


Dataset courtesy of:
Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra and Jorge L. Reyes-Ortiz.
Human Activity Recognition on Smartphones using a Multiclass Hardware-Friendly Support Vector Machine.
International Workshop of Ambient Assisted Living (IWAAL 2012). Vitoria-Gasteiz, Spain. Dec 2012
http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones

クラシフィケーションのための信号解析
加速度センサー情報による人の活動認識

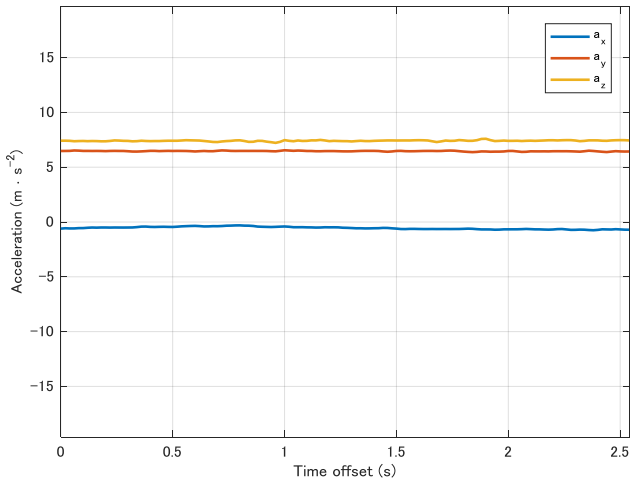
直立の
状態

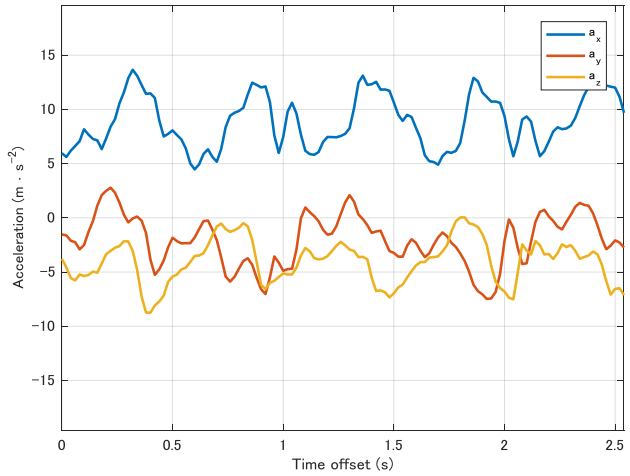




歩いている
状態

寝転んだ
状態

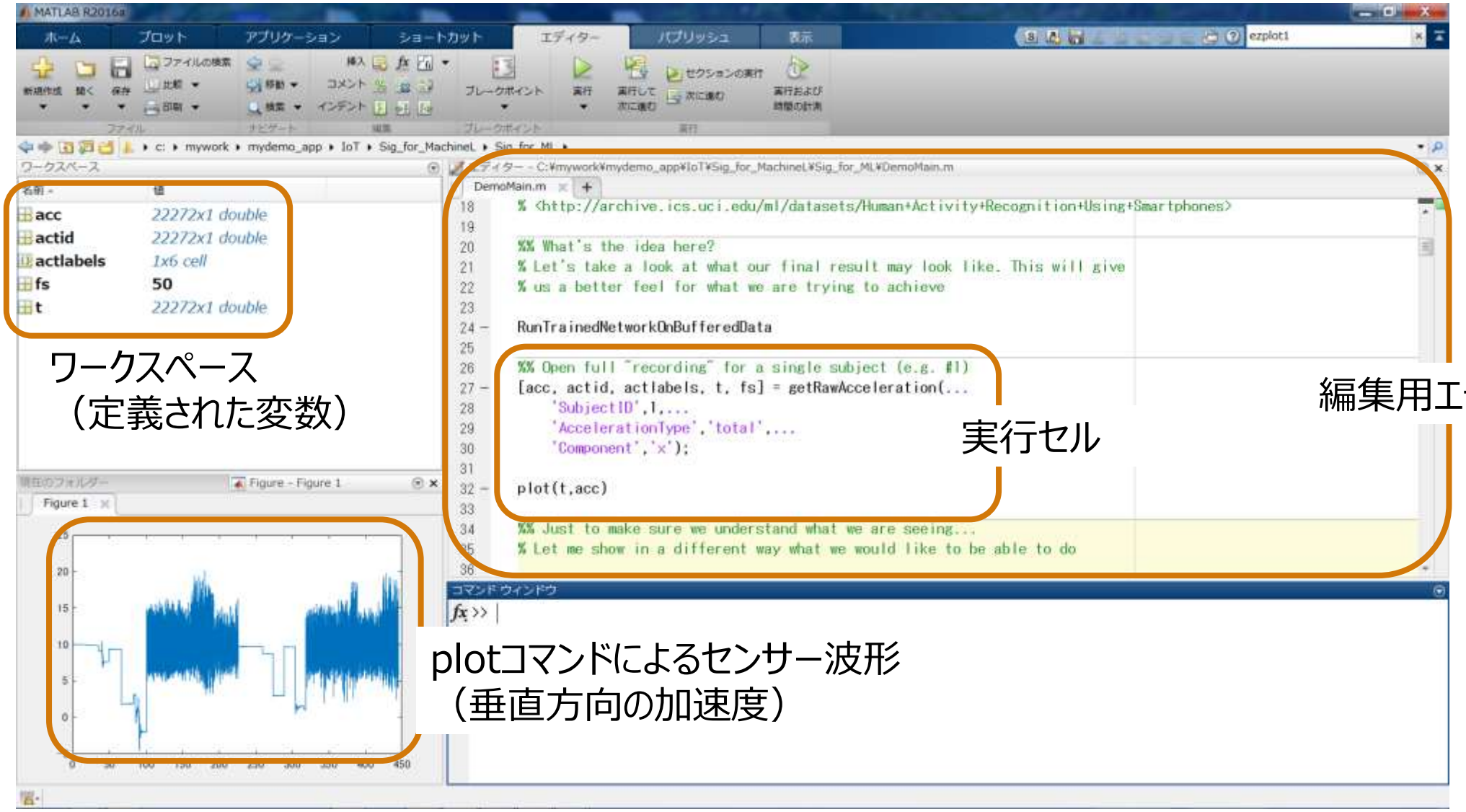




階段を上っている
状態

Demo: クラシフィケーションのための信号解析

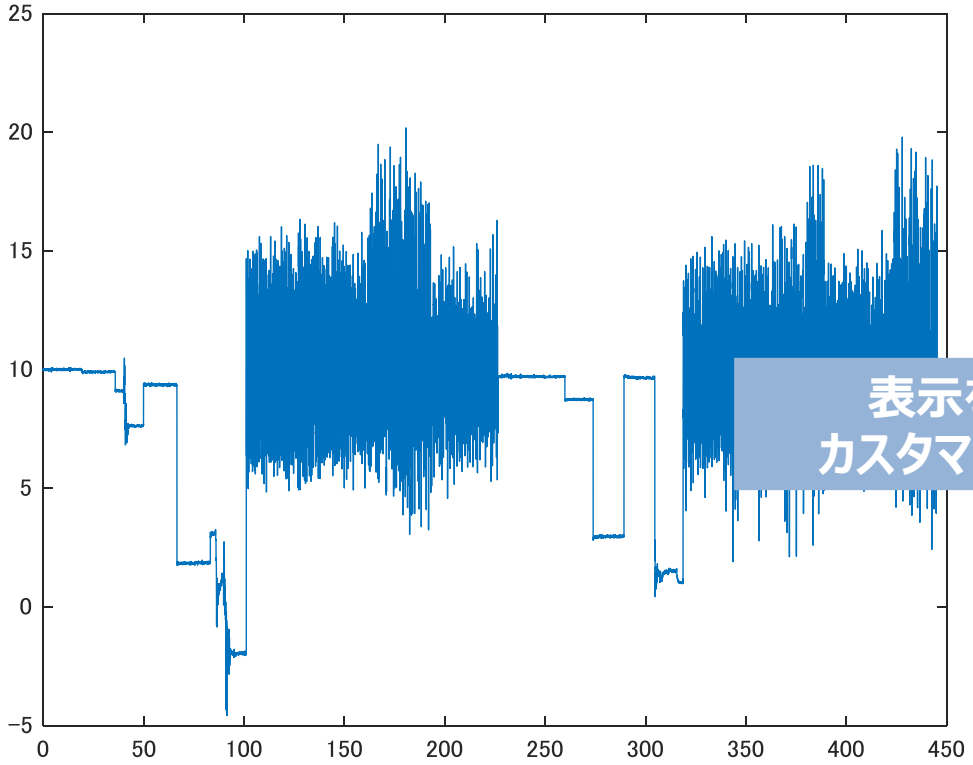
MATLAB環境



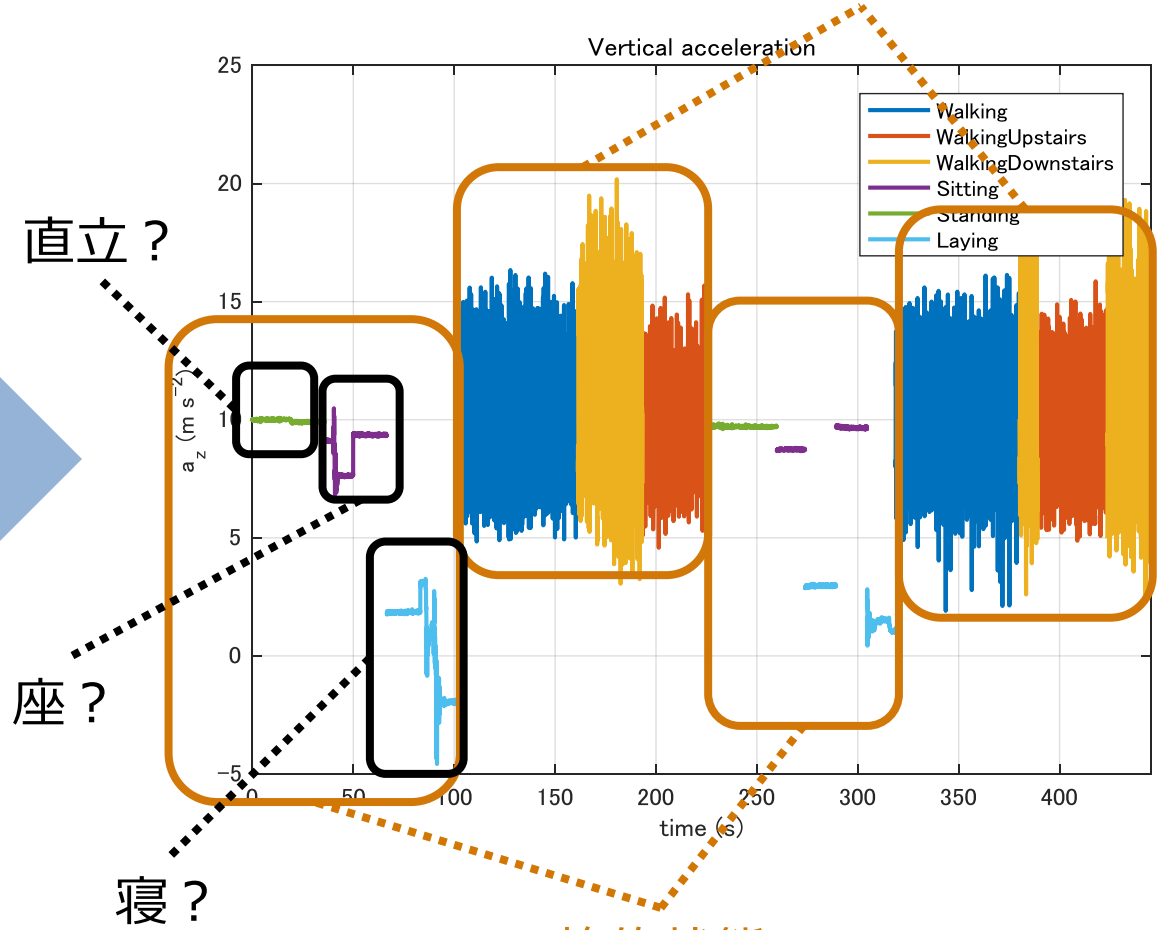
Demo: クラシフィケーションのための信号解析

可視化（視覚的情報からの予想）

動的状態
- 歩行?
- 階段を上る?
- 階段を下る？

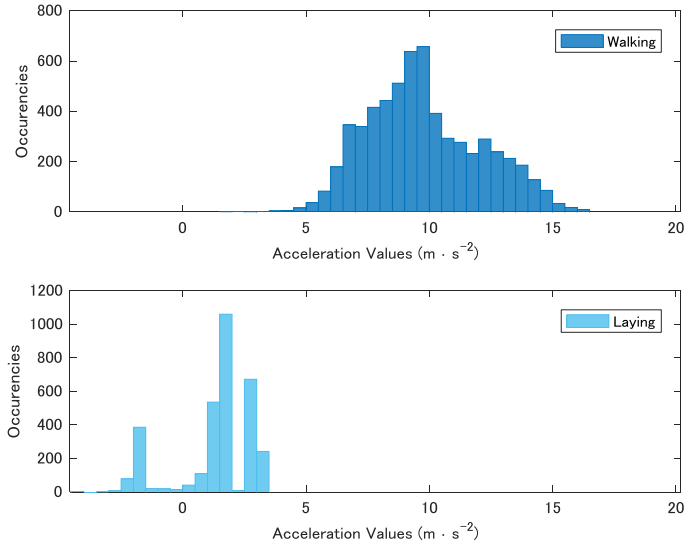


表示を
カスタマイズ

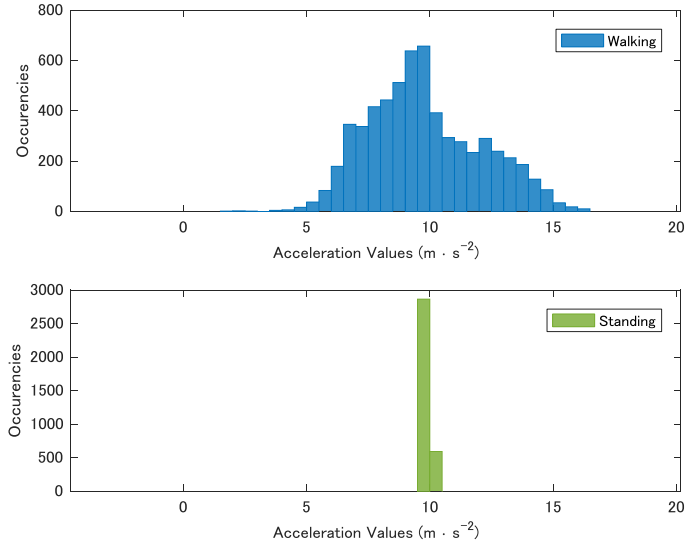


Demo: クラシフィケーションのための信号解析

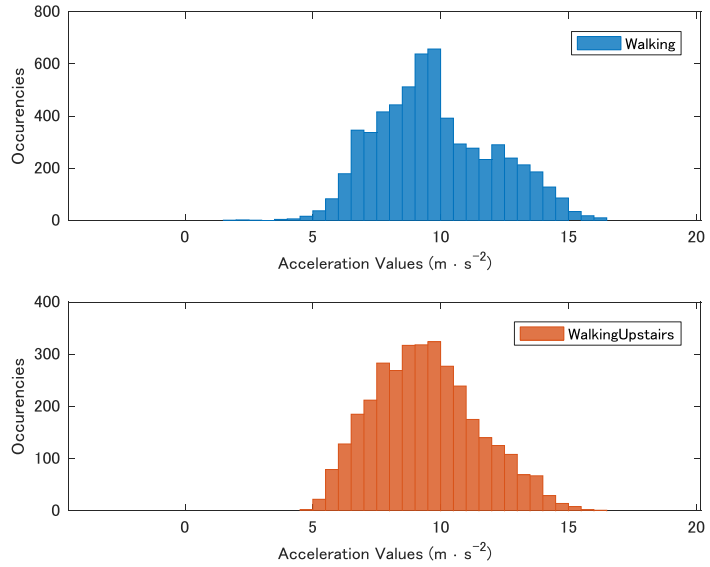
ベーシックな特徴量：ヒストグラムの平均値、RMS等



歩く vs. 寝る
↓
平均値で識別



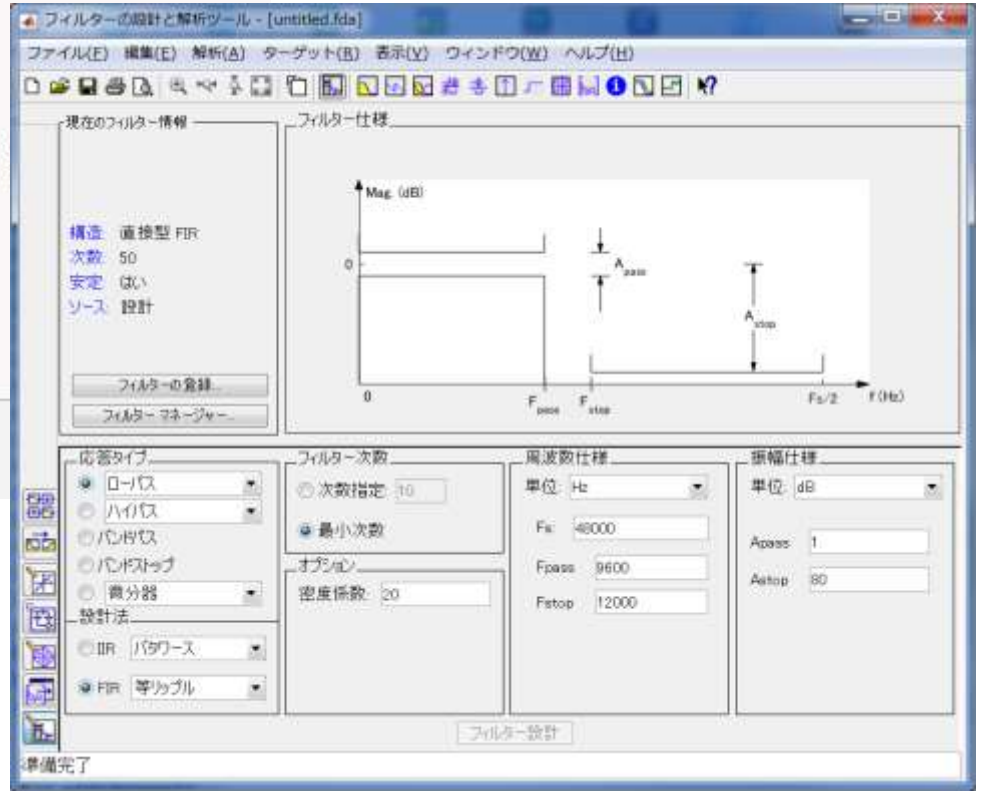
歩く vs. 立つ
↓
標準偏差やRMSで識別



歩く vs. 階段を上る
↓
何で識別する？

Demo: クラシフィケーションのための信号解析

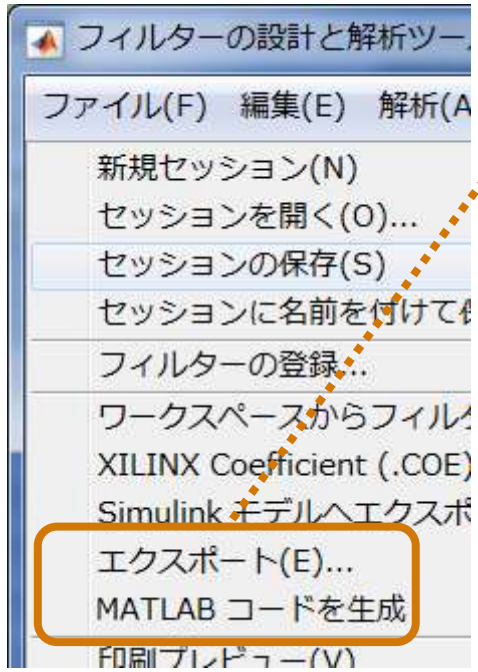
前処理：重力の影響を除去（フィルタ設計環境）

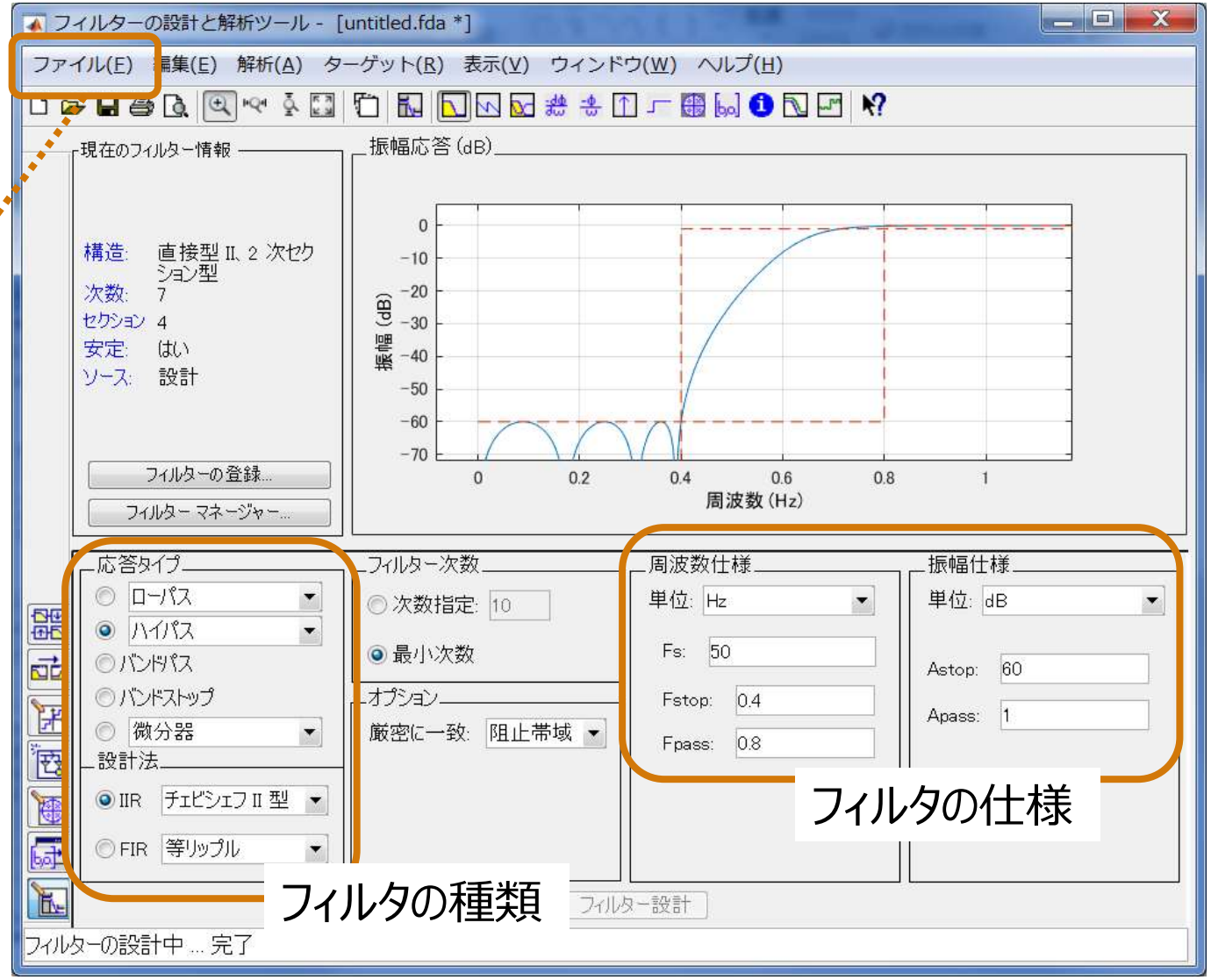


>>fdatool

Demo: クラシフィケーションのための信号解析

フィルタ設計GUI





プログラム内でフィルタを適用する方法
①エクスポート：
フィルタ係数を出力
②MATLABコードを生成：
フィルタ処理関数の自動生成

Demo: クラシフィケーションのための信号解析

フィルタ設計環境(cont'd)

前頁「②MATLABコードを生成」を選択した場合

DemoMain.m　myfilter.m

自動生成された自作フィルタ関数

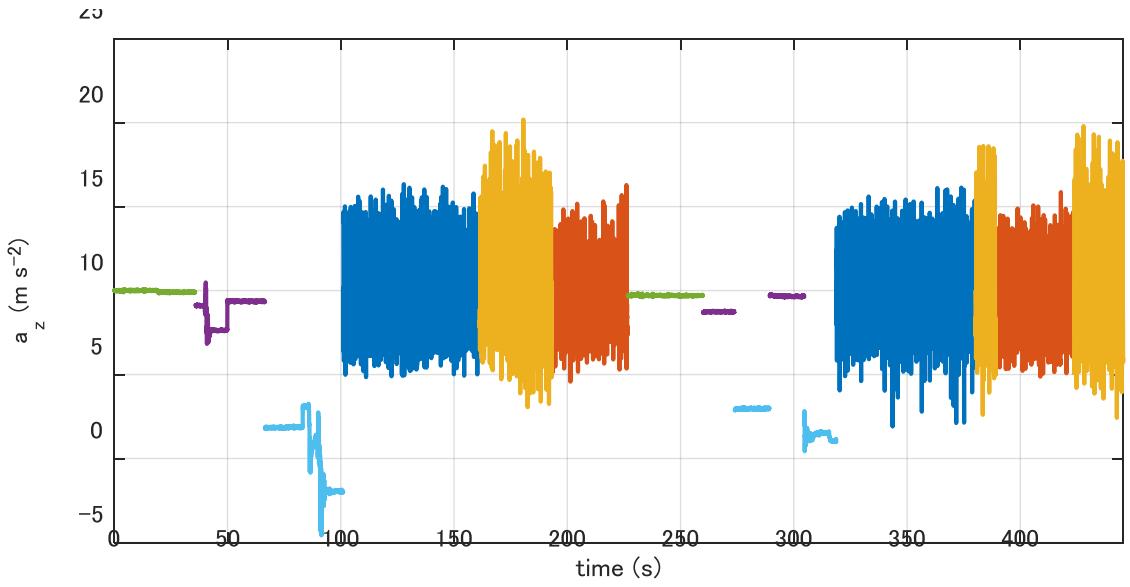
```
function Hd = myfilter
%MYFILTER Returns a discrete-time filter object.

% MATLAB Code
% Generated by MATLAB(R) 8.4 and the Signal Processing Toolbox 6.22.
% Generated on: 19-Dec-2014 17:23:22

% Chebyshev Type II Highpass filter designed using FDESIGN.HIGHPASS.

% All frequency values are in Hz.
Fs = 50;  % Sampling Frequency

Fstop = 0.4;          % Stopband Frequency
Fpass = 0.8;          % Passband Frequency
Astop = 60;           % Stopband Attenuation (dB)
Apass = 1;            % Passband Ripple (dB)
match = 'stopband';   % Band to match exactly

% Construct an FDESIGN object and call its CHEBY2 method.
h  = fdesign.highpass(Fstop, Fpass, Astop, Apass, Fs);
Hd = design(h, 'cheby2', 'MatchExactly', match);
```

フィルタの仕様

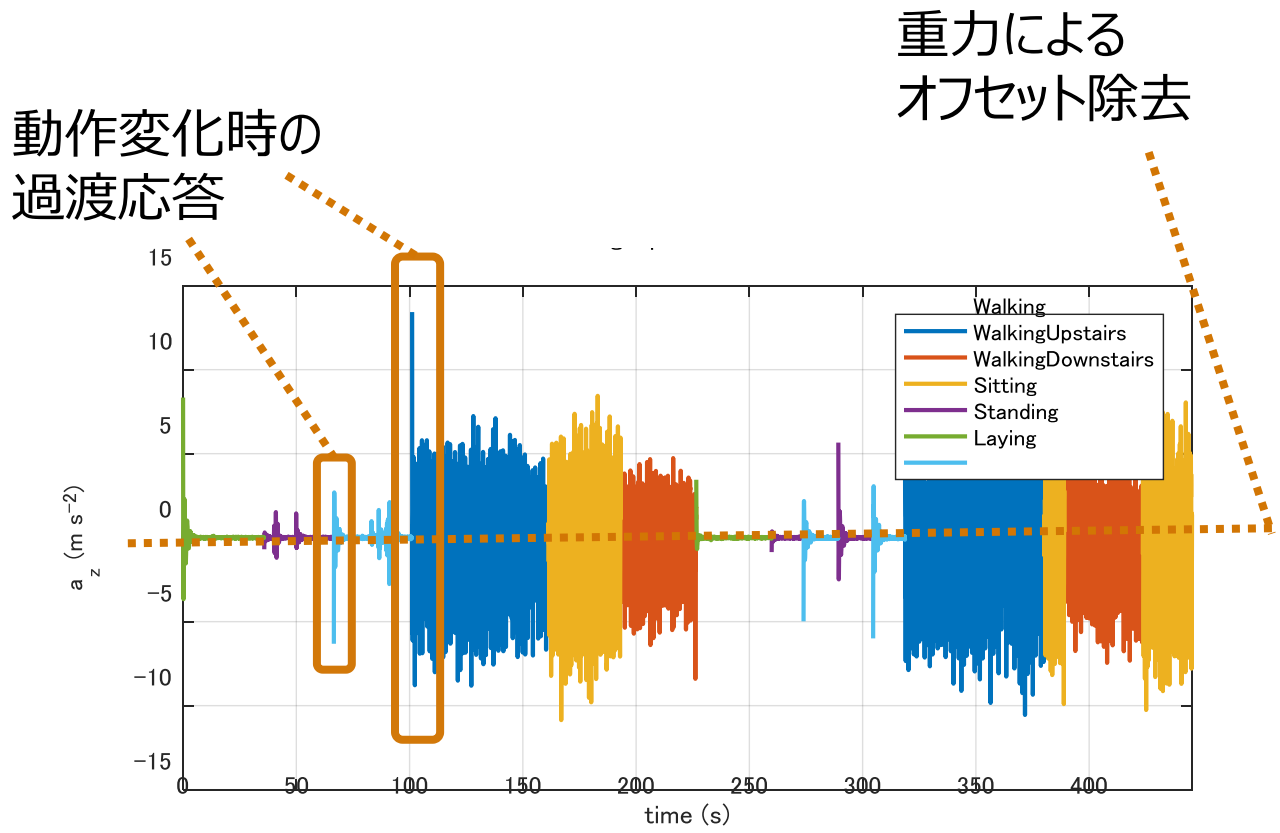
```
e.g.)
% フィルタオブジェクト定義
H = myfilter;
% フィルタ処理
y = filter(H,data);
```

Demo: クラシフィケーションのための信号解析

体の動きによる成分の取得



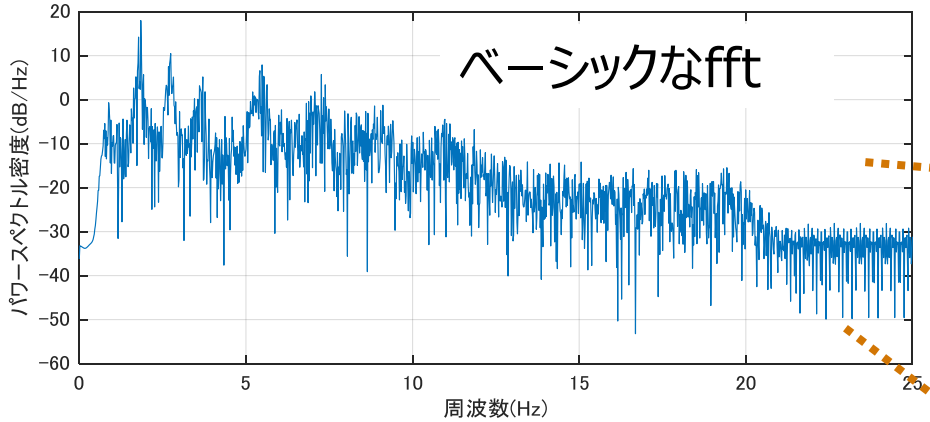
オリジナル
（ハイパスフィルタ処理前）

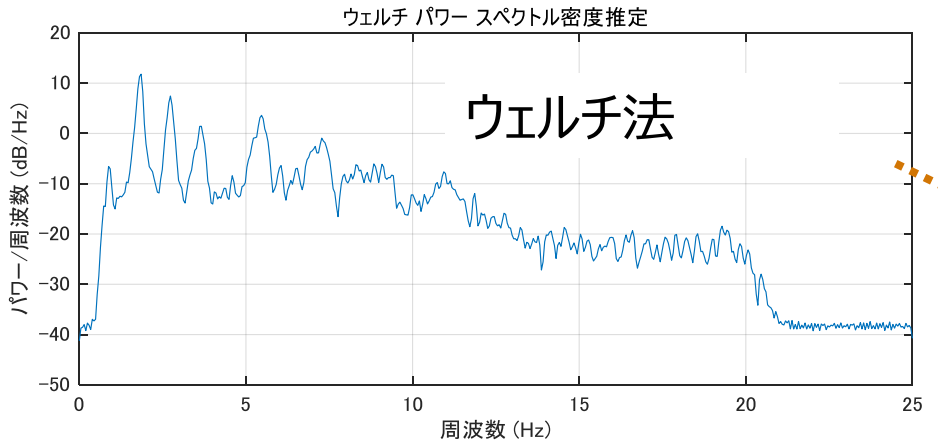


体の動きによる寄与分
（ハイパスフィルタ処理後）

Demo: クラシフィケーションのための信号解析

周波数ドメインでの解析





自作スクリプト

periodogram関数

pwelch関数

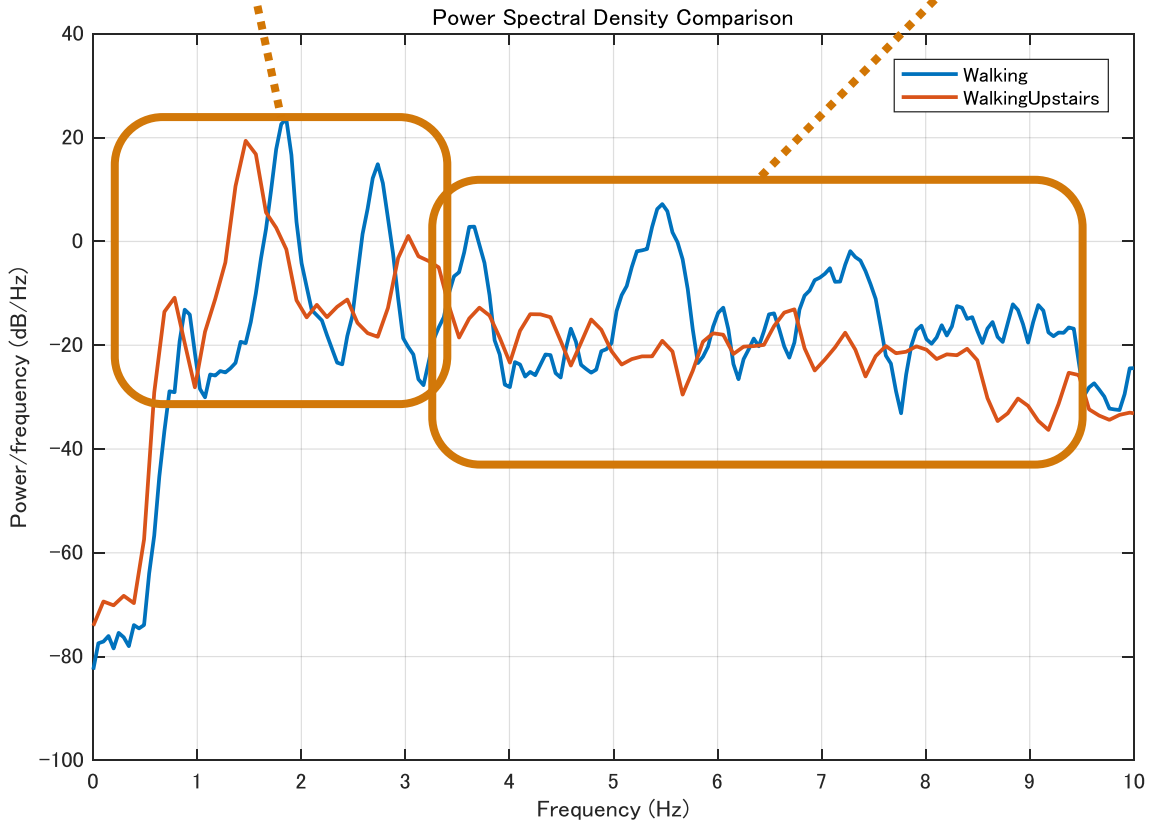
```
151 –  N = length(abw);
152 –  Y = fft(abw);
153 –  Y = Y(1:N/2 + 1);
154 –  Yp = (1/(fs*N)) * abs(Y).^2;
155 –  Yp(2:end-1) = 2*Yp(2:end-1);
156 –  f = 0 : fs/N : fs/2;
157
158 –  subplot(2,1,1), plot(f, 10*log10(Yp))
159 –  xlabel('周波数(Hz)')
160 –  ylabel('パワースペクトル密度(dB/Hz)')
161 –  grid
162
163 –  subplot(2,1,2)
164
165    %% using Periodogram
166 –  periodogram(abw,[],[],fs)
167
168    %% using Welch method
169 –  pwelch(abw,[],[],[],fs)
170
```

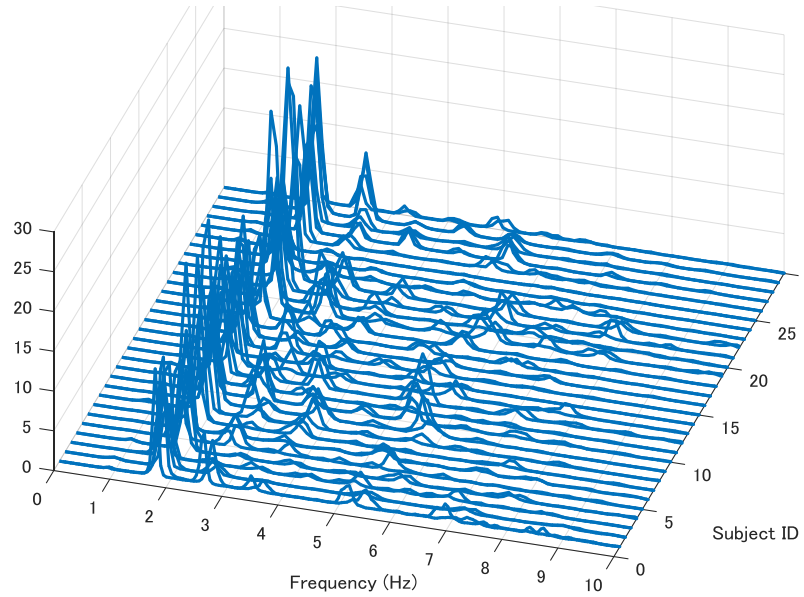
Demo: クラシフィケーションのための信号解析

周波数ドメインでの解析: 各動作の比較、特徴量の獲得

「階段を上る」の方が全体的に左より
→一周期の時間が長い
→動作が**遅い**

「階段を上る」の方が高調波の減衰が大きい
→ピュアな正弦波に近い
→動作が**スムーズ**



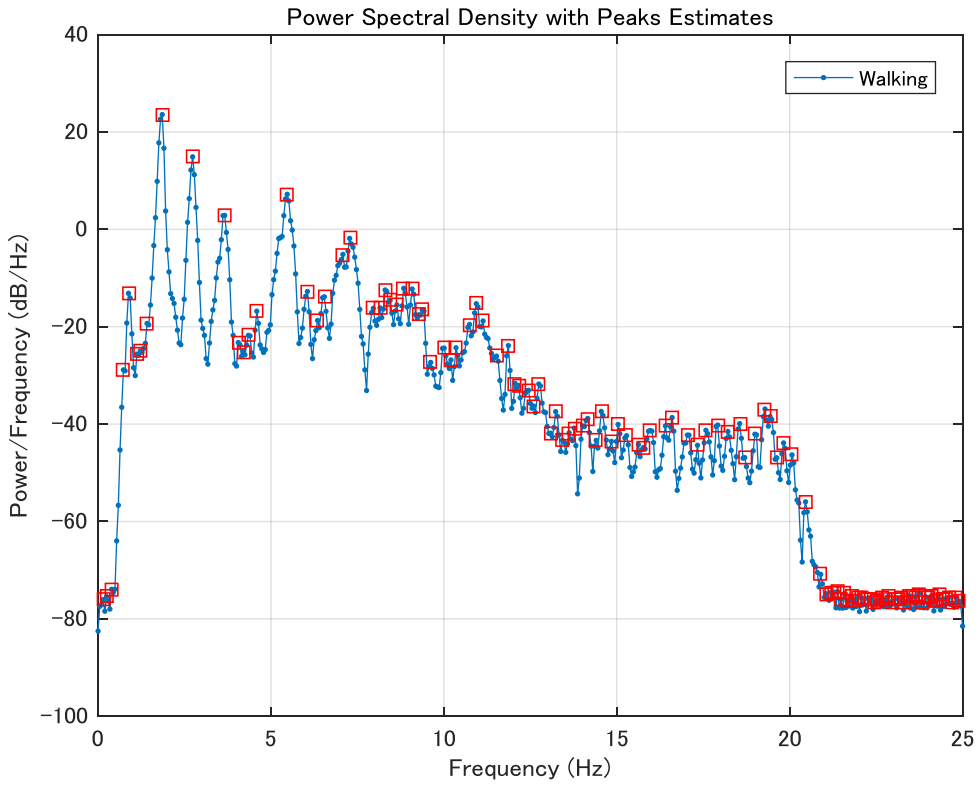


他のデータでも同傾向であることを、
3Dプロットで確認

Demo: クラシフィケーションのための信号解析

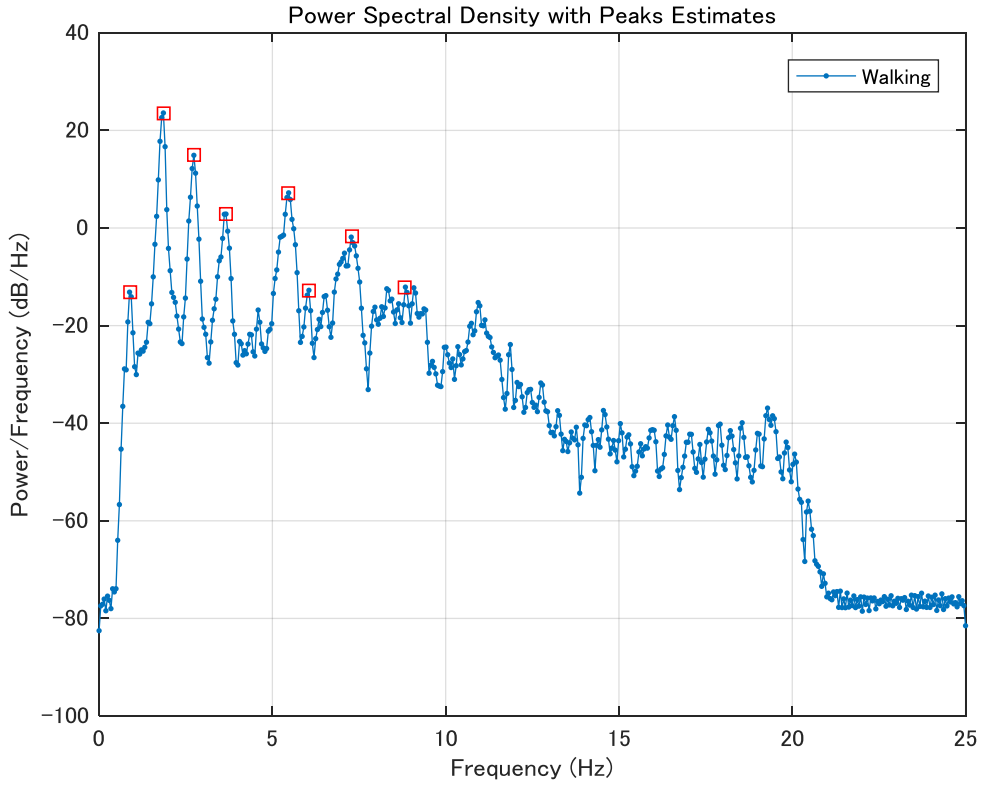
周波数ドメインでの解析: 特徴量の自動取得

制約なしの場合




```
>>findpeaks(p);
```

ピークの数、ピーク間の距離、
プロミネンス等の制約を与えた場合

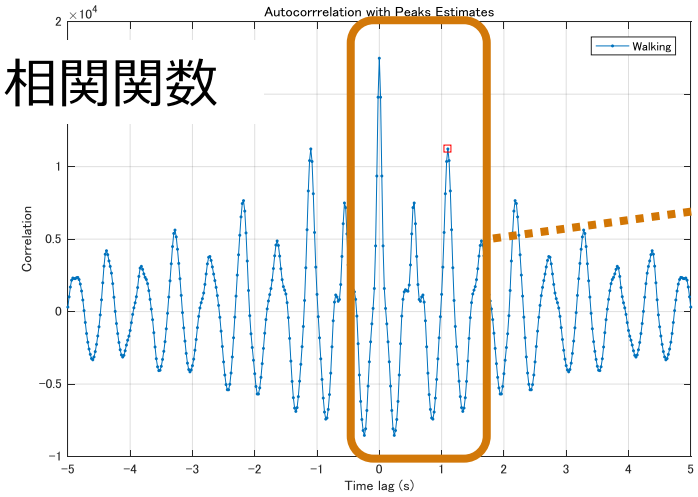


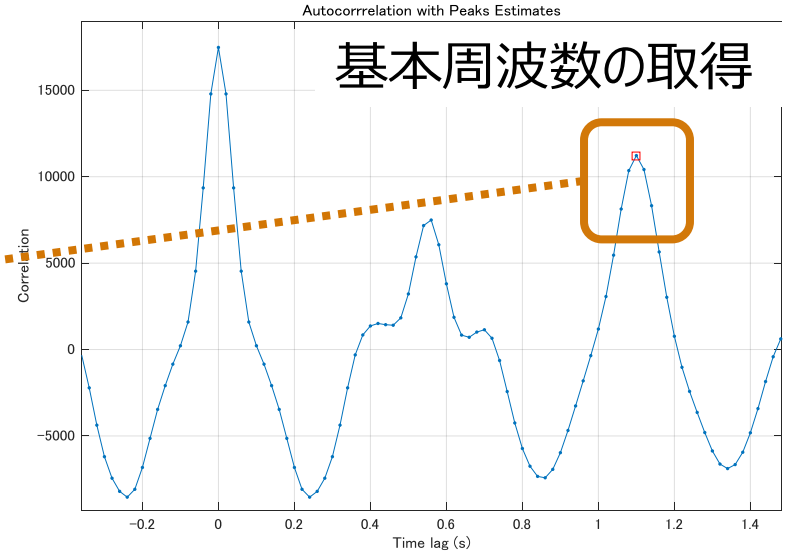

```
>>findpeaks(p,'minpeakprominence',5,...
    'minpeakdistance',0.15);
```

Demo: クラシフィケーションのための信号解析

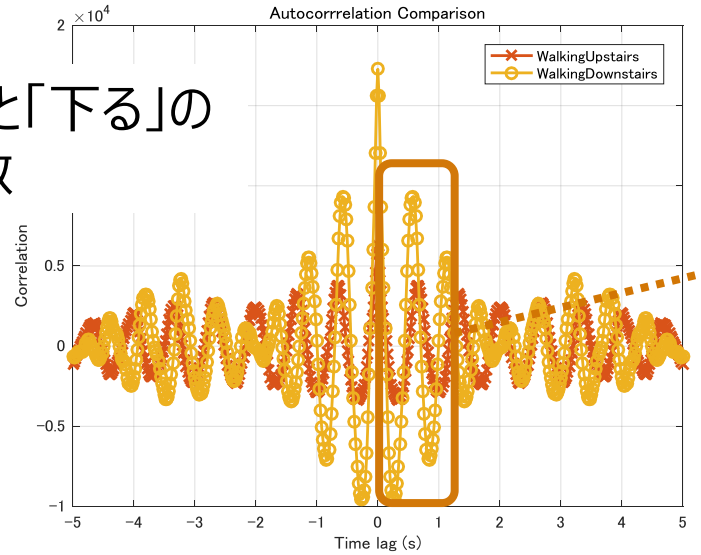
自己相関関数:基本周波数の取得等

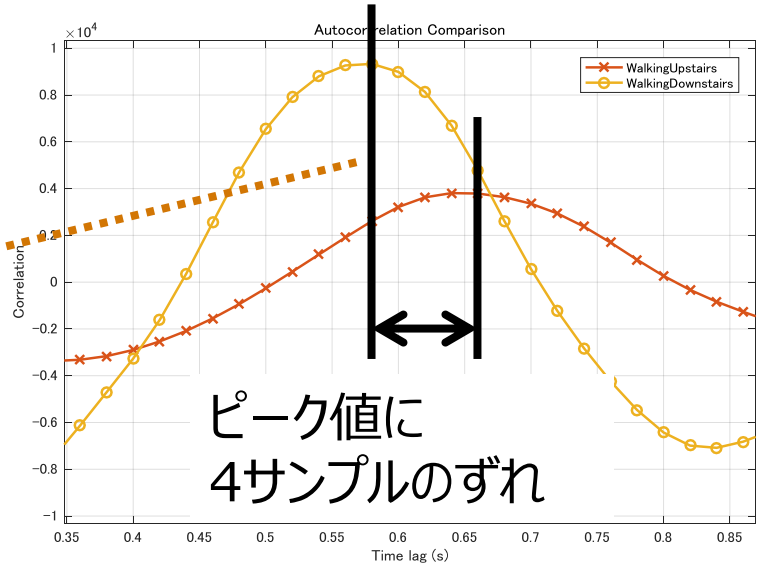
「歩く」の自己相関関数





階段を「上る」と「下る」の
自己相関関数





Demo: クラシフィケーションのための信号解析

3軸加速度センサからの、特徴量一括取得

特徴量一括取得用関数（総数66）

```
function feat = featuresFromBuffer(atx, aty, atz, fs)

persistent fhp
if(isempty(fhp))
    fhp = hpfilter;
    fhp.PersistentMemory = false;
end

feat = zeros(1,66);

abx = filter(fhp,atx);
aby = filter(fhp,aty);
abz = filter(fhp,atz);

%%
feat(1) = mean(atx);
feat(2) = mean(aty);
feat(3) = mean(atz);

%%
feat(4) = rms(abx);
feat(5) = rms(aby);
feat(6) = rms(abz);

%%
feat(7:9) = covFeatures(abx, fs);
feat(10:12) = covFeatures(aby, fs);
feat(13:15) = covFeatures(abz, fs);

%%
feat(16:27) = spectralPeaksFeatures(abx, fs);
feat(28:39) = spectralPeaksFeatures(aby, fs);
feat(40:51) = spectralPeaksFeatures(abz

%%
feat(52:56) = spectralPowerFeatures(abx, fs);
feat(57:61) = spectralPowerFeatures(aby, fs);
feat(62:66) = spectralPowerFeatures(abz, fs);
```

前処理（フィルタリング）

特徴量1（平均値）

特徴量2（RMS）

特徴量3（自己相関）

特徴量4（ハーモニクス）

特徴量5（エネルギー）

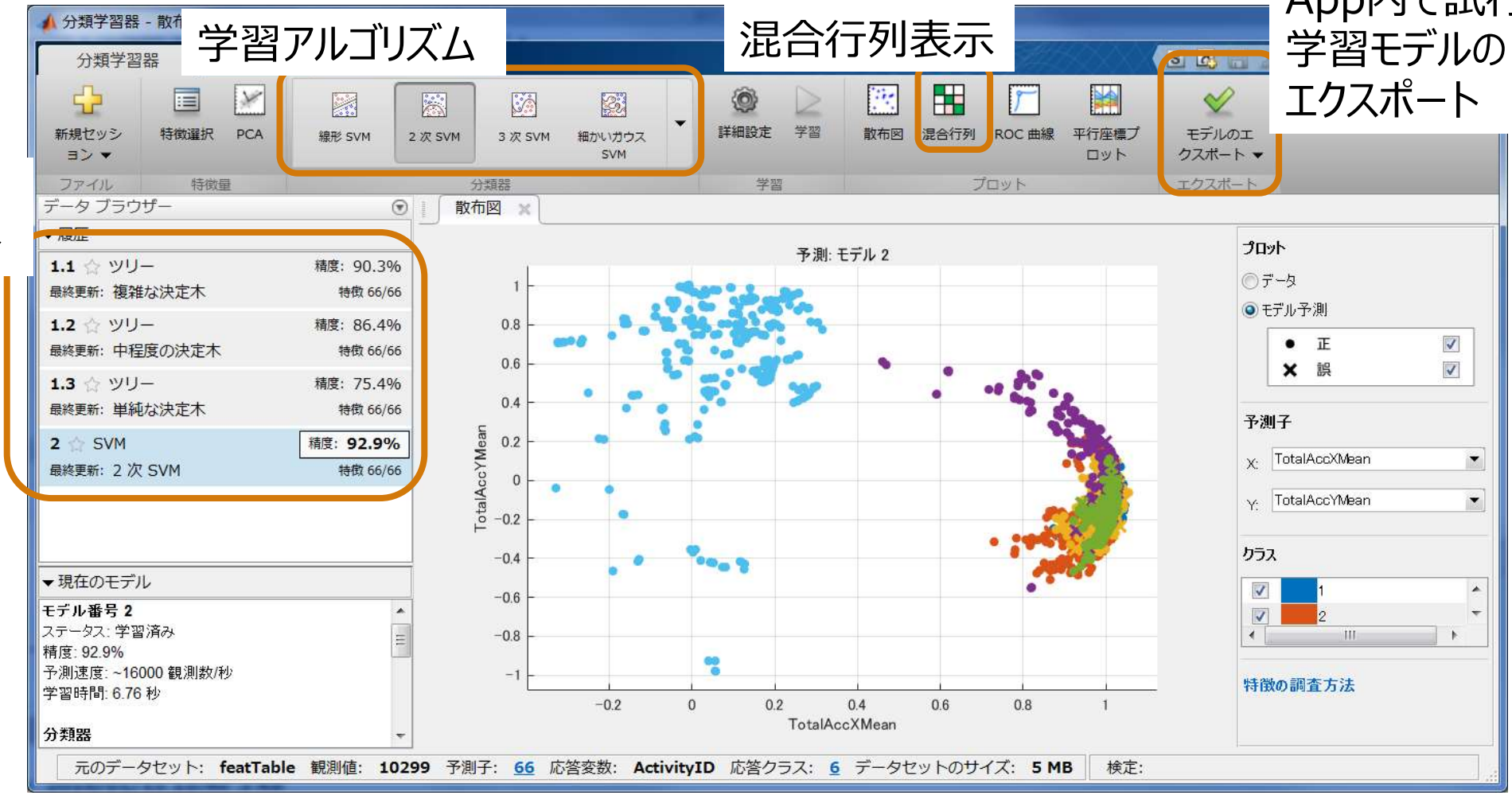
Demo: クラシフィケーションのための信号解析
classficationLeanerアプリによる機械学習

学習アルゴリズム

混合行列表示

App内で試行した
学習モデルの
エクスポート

試行済み
学習履歴

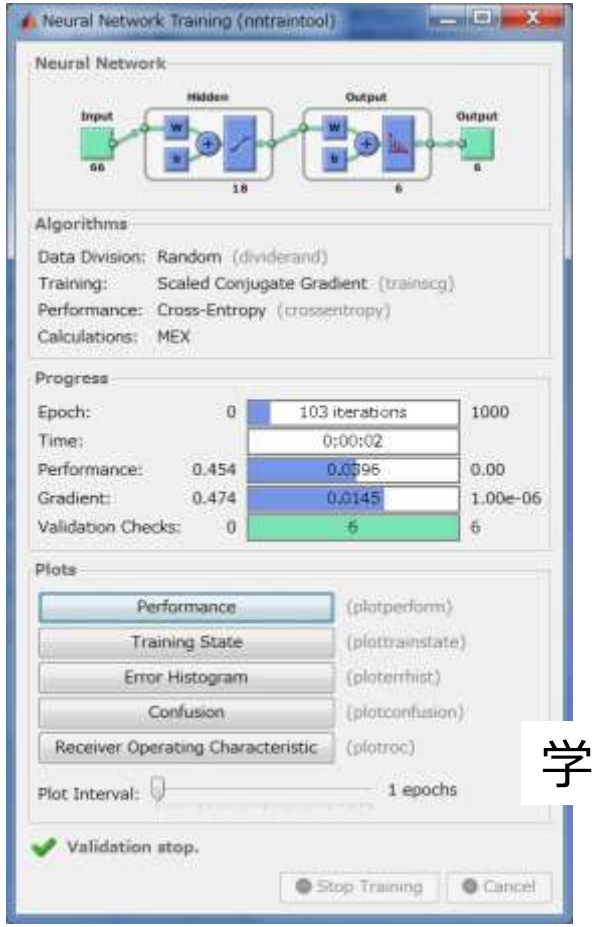


>>classificationLearner

Demo: クラシフィケーションのための信号解析

ニューラルネットワークによるクラシフィケーション

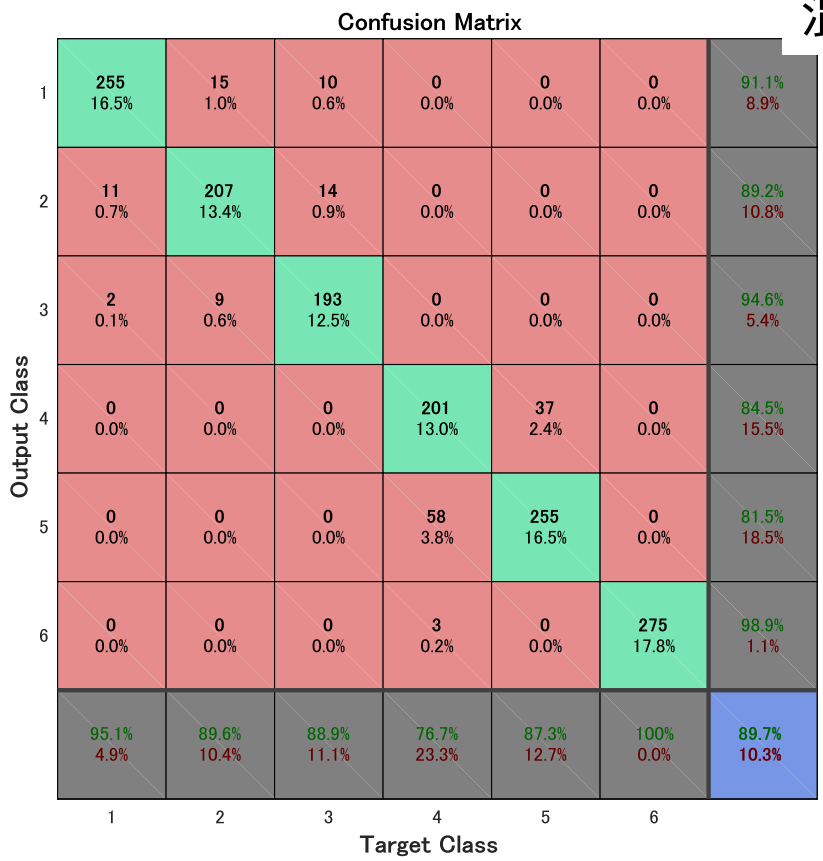
```
>>net = patternnet(n);
>>net = train(net, X, targets);
```



学習の様子

```
>>plotconfusion(targets, outputs)
```

混合行列



補足：センサーデータ前処理に使える新機能

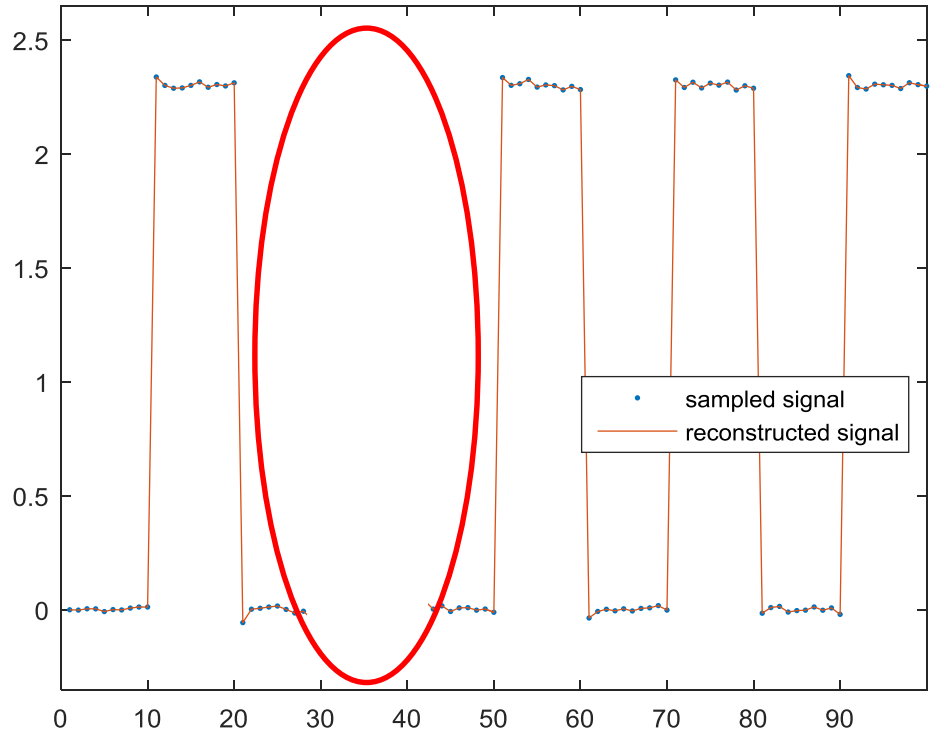
R2016a
R2016b

ギャップフィリング

R2016a

ARモデリングによる欠落したデータの復元: fillgaps

- `Resample()`で大規模欠落データの補間
- 複数のギャップのサポート
- 赤池情報量規準を元に自動でARモデルの次数を決定
- 非定常モデルをオプションで選択可能

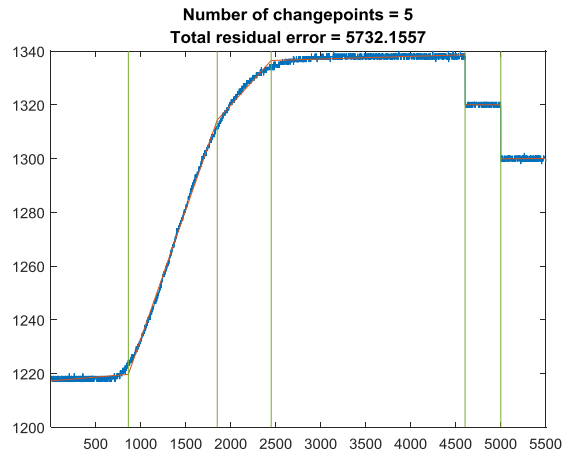



```
» load clockex
» x(29:42) = NaN;
» fillgaps(x)
```

局所統計量の変化点の検出

信号の統計的性質の変化を検出: findchangepts

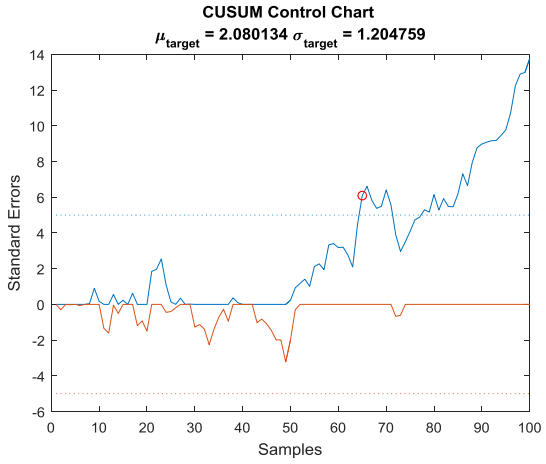
R2016a



- 局所的な統計量の変化に応じた信号分割を実施
  - 統計量として平均、分散、平均と分散、線形回帰をサポート

```
» load('engineRPM.mat','x')
» findchangepts(x,'Statistic','linear', …
» 'MinThreshold',var(x))
```

- cusum は累積和を使って局所平均値の変化点を検出



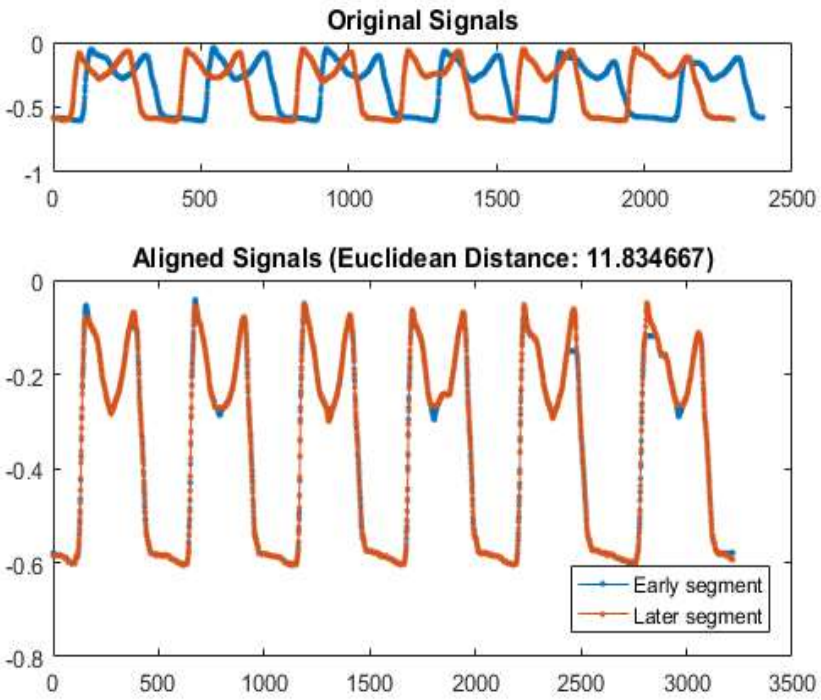

```
» load('meanshift','x')
» cusum(x)
```

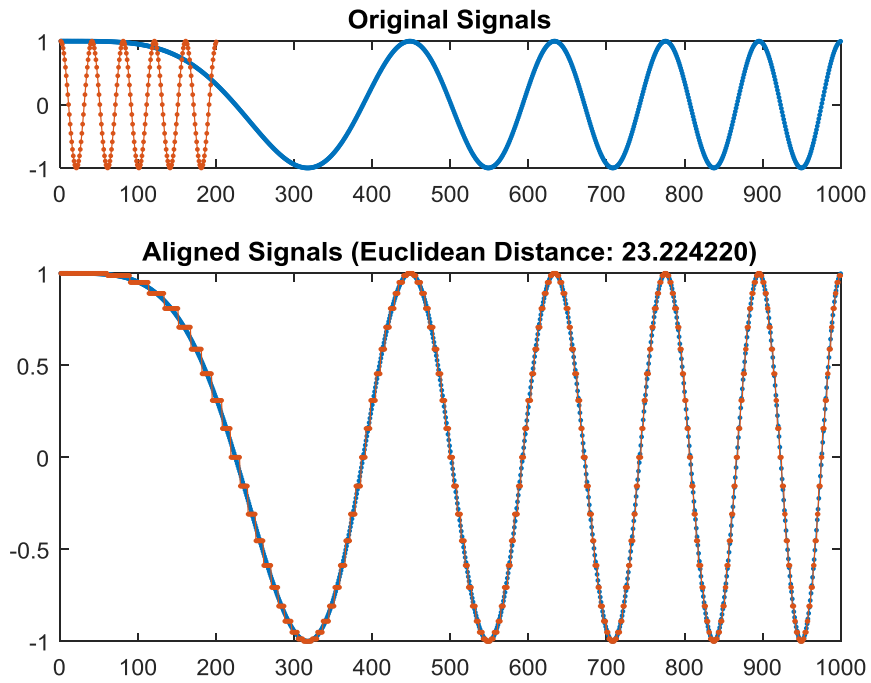
ダイナミックタイムワーピング

信号の伸長、アライン、比較: dtw

R2016a

- 2つの信号の軌跡の類似性を測定
- タイムアラインメントをオプションで指定可能
- 時系列データの分類に使われる






```
» dtw(x,y)
```

実信号間編集距離

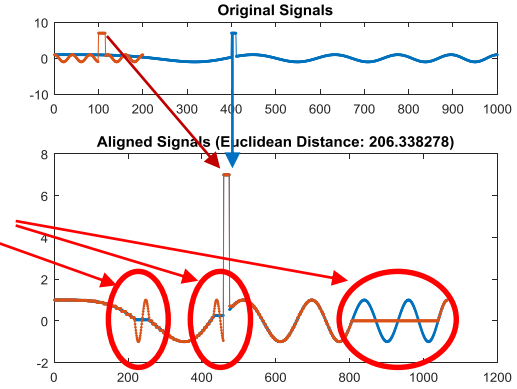
実信号間の調整: edr

R2016b

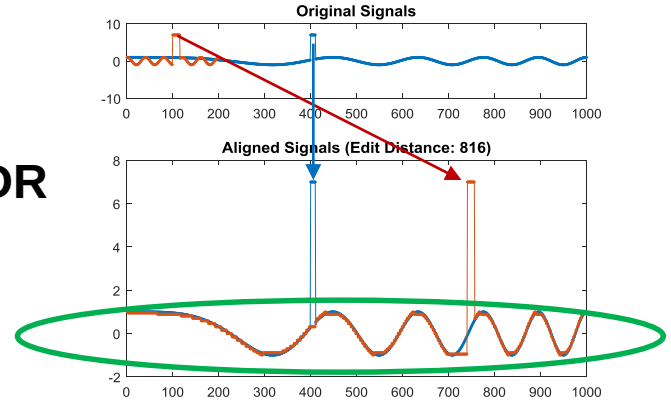
- 動的タイム ワーピング(DTW) の拡張機能
- 局所的な外れ値があった場合に正しく信号を整列
- 追加サポート
  - 複素信号
  - N次元信号

DTW

ミスアライン



EDR



適切なアライン

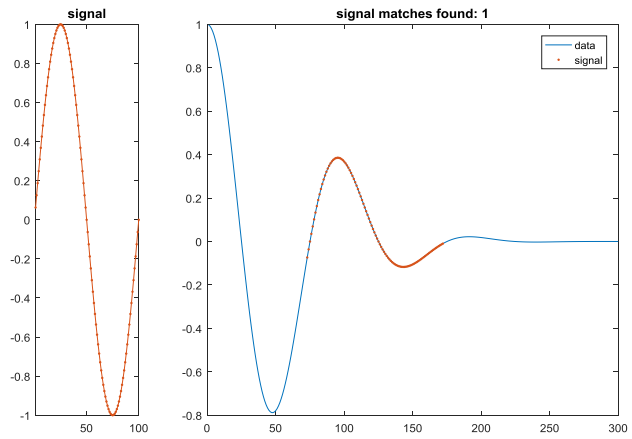
```
» x = chirp(0:999,0,1000,1/100);
» y = cos(2*pi*5*(0:199)/200);
» x(400:410) = 7;
» y(100:115) = 7;
» edr(x,y,.1)
```

類似性サーチでデータ内の信号を見つける

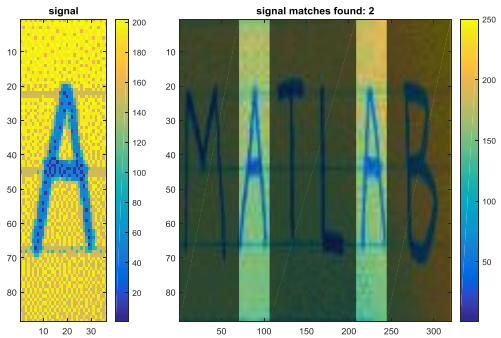
R2016b

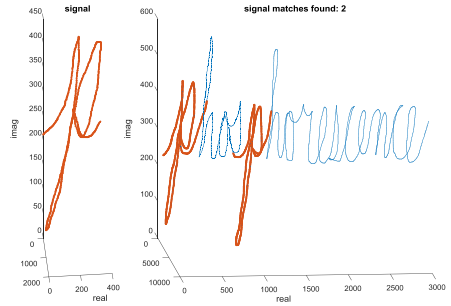
信号の最適にマッチする箇所を探す: findsignal

- 任意の次元の実数/複素数ベクトルの比較
- 複数のマッチング
- 有名なメトリックス (Euclidean, Squared Euclidean, Symmetric Kullback-Liebler)
- 様々な正規化 (Zero-mean, Unit norm, average power)




```
» data = exp(-((1:300)/100).^2).*cos(2*pi*(1:300)/100);
» signal = sin(2*pi*(1:100)/100);
» findsignal(data,signal)
```





Agenda

- 背景
- センサー信号の場合
- 音声信号の場合
- 画像の場合
- 関連情報
- まとめ

Celestion社事例

MathWorks Accelerating the pace of engineering and science

製品 ソリューション アカデミア サポート コミュニティ（英語） イベント

Technical Articles and Newsletters

Overview Search Technical Articles Newsletters Cleve's Corner Collection Sign Up

Applying Machine Learning Techniques to Classify Musical Instrument Loudspeakers

By Andrew Harper, Celestion

Celestion loudspeakers have powered the performances of many noted guitar and bass players, including legends such as Jimi Hendrix. Deciding whether a loudspeaker is good enough for professional musicians is a lengthy and painstaking process. Each speaker has its own unique sound based on a combination of sonic characteristics, such as midrange character and brightness. Evaluating a musical instrument

density function reveals considerable overlap for good and bad speakers (Figure 3).

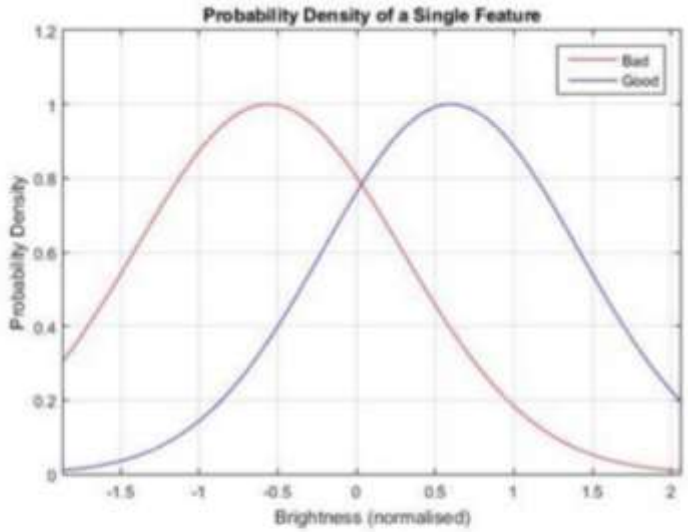


Figure 3. Brightness probability density function for good speakers (blue) and bad speakers (red). There is still considerable overlap between good and bad results, despite the clear difference in the means.

The result could be predicted, but with very limited certainty. Our idea was that in combining many results, the degree of certainty would increase. Rather than a linear 'hurdle' type approach, machine learning would enable us to consider the values of all features at once and then calculate the overall probability of a good result

http://jp.mathworks.com/company/newsletters/articles/applying-machine-learning-techniques-to-classify-musical-instrument-loudspeakers.html?s_tid=srchtitle

celestion matlab

Celestion社事例抜粋： 特徴量の選択、得られた教訓等

評価した特徴量

- I developed and examined **more than 50 features** over several months. These included energy features, such as the **energy center**; spectral features, such as **high-frequency roll-off** and **brightness**; and psychoacoustic features, such as **sharpness**.

心理音響値の定量化

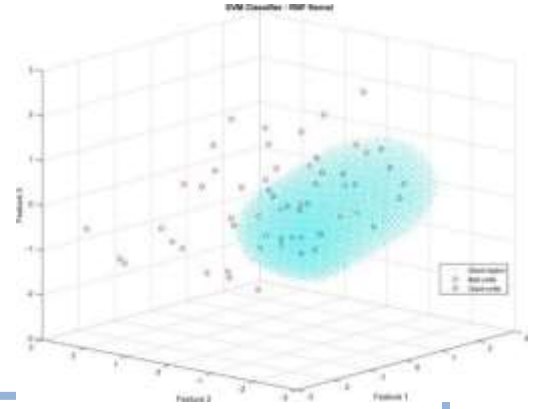
- To compute psychoacoustic values, I used MATLAB, Signal Processing Toolbox, and DSP System Toolbox™…. I applied functions for **smoothing**, **convolution**, **peak finding**, and computing **Welch's power spectral density (PSD)** estimates, among others.

三つの教訓

- The first is that there is value in applying **simpler** algorithms such as linear logistic regression, even if they end up performing poorly compared to SVM, neural networks, or ensemble methods.
- The second lesson is that with a long and open-ended process like **feature extraction**, having access to readily available **signal processing functions** is virtually a **necessity**
- The third lesson is that **no** single machine learning technique works well with **all** datasets.

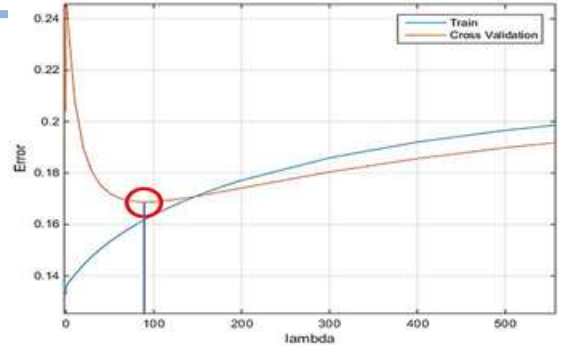
Celestion社事例抜粋： 使用ツール

Using **MATLAB®**, **Statistics and Machine Learning Toolbox™, and Signal Processing Toolbox™**, I developed an application that uses machine learning algorithms to classify one loudspeaker model as either good or bad.



To compute psychoacoustic values, I used MATLAB, **Signal Processing Toolbox, and DSP System Toolbox™** to implement established formulas for psychoacoustics from a range of textbooks.

I used **Optimization Toolbox™** to find an optimal value for*lambda* and minimize the error on both sets.



I used **Audio System Toolbox™** to stream real-time audio directly from MATLAB, created the main UI for playing and selection…

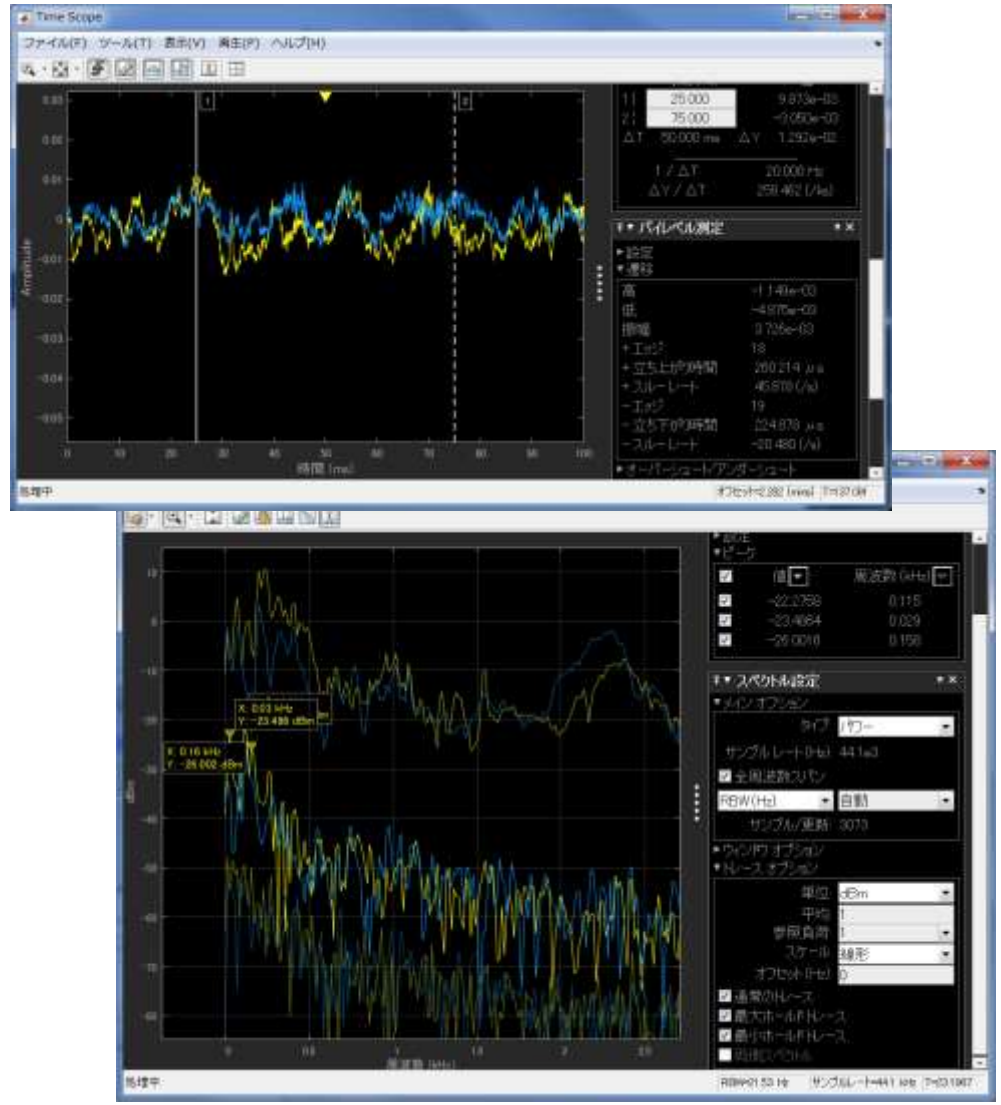
Audio System Toolbox™

- オーディオシステムの開発と
  シミュレーション
  - オーディオエフェクト用ライブラリ
  - ラウドネス測定
  - ASIOドライバサポートによる
    低レイテンシリアルタイムシミュレーション
  - マルチチャネルI/O
  - MIDIコントローラによるパラメータチューニング

- パラメータ調整と官能評価
  - audioTestBenchで波形やスペクトルの可視化と
    パラメータチューニング
  - MATLABコードからVST Plugin生成・取り込み

例: AudioSystem Toolboxによるデータ取得

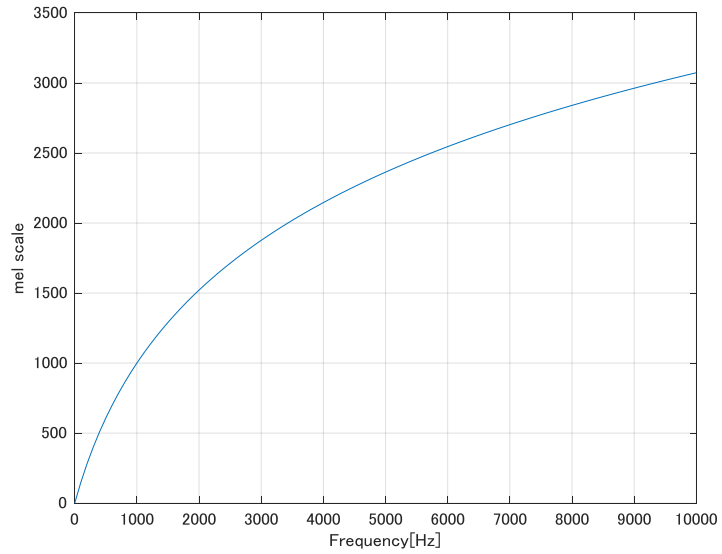
```
%% オブジェクトの定義
v1 = audioDeviceReader

%% サウンド入力パラメータ設定
v1.SampleRate     = Fs;
v1.NumChannels  = NumC;
v1.Driver    = 'ASIO';
v1.Device  = 'device name';

%% 可視化オブジェクト
hts = dsp.TimeScope('SampleRate', Fs);
hsa = dsp.SpectrumAnalyzer('SampleRate',Fs);

%% スタート
for ii = 1:500
    data = step(v1);     % 取得データの変数への代入
    step(hsa, data)
    step(hts, data)
end
```



メル周波数ケプストラム係数：
MFCC(Mel-Frequency Cepstral Coefficients)

ケプストラム（英: Cepstrum）：
（信号の）ケプストラムとは、（信号の）フーリエ変換の対数（位相アンラッピングを施したもの）をフーリエ変換したもの。スペクトルのスペクトルとも呼ばれる。

応用：
… 人間の声や音楽の信号を表す特徴ベクトルとしても使われている。この場合、スペクトルをまずメル尺度を使って変換する。その結果はメル周波数ケプストラム係数（MFCC）と呼ばれる。これは話者認識やピッチ抽出アルゴリズムなどに応用されている。…



メル尺度
（実周波数と知覚音高の関係）

（出展：フリー百科事典「ウィキペディア(Wikipedia)」, “ケプストラム”より抜粋）

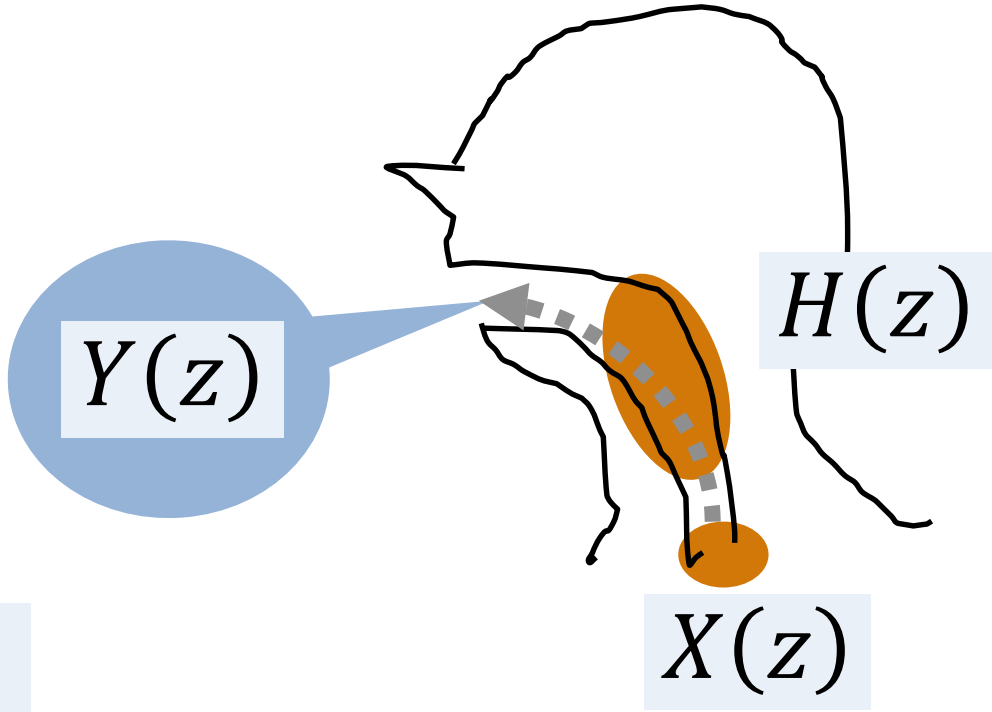
ソース・フィルターモデル

ソース：声帯励振
（音源）

合成音

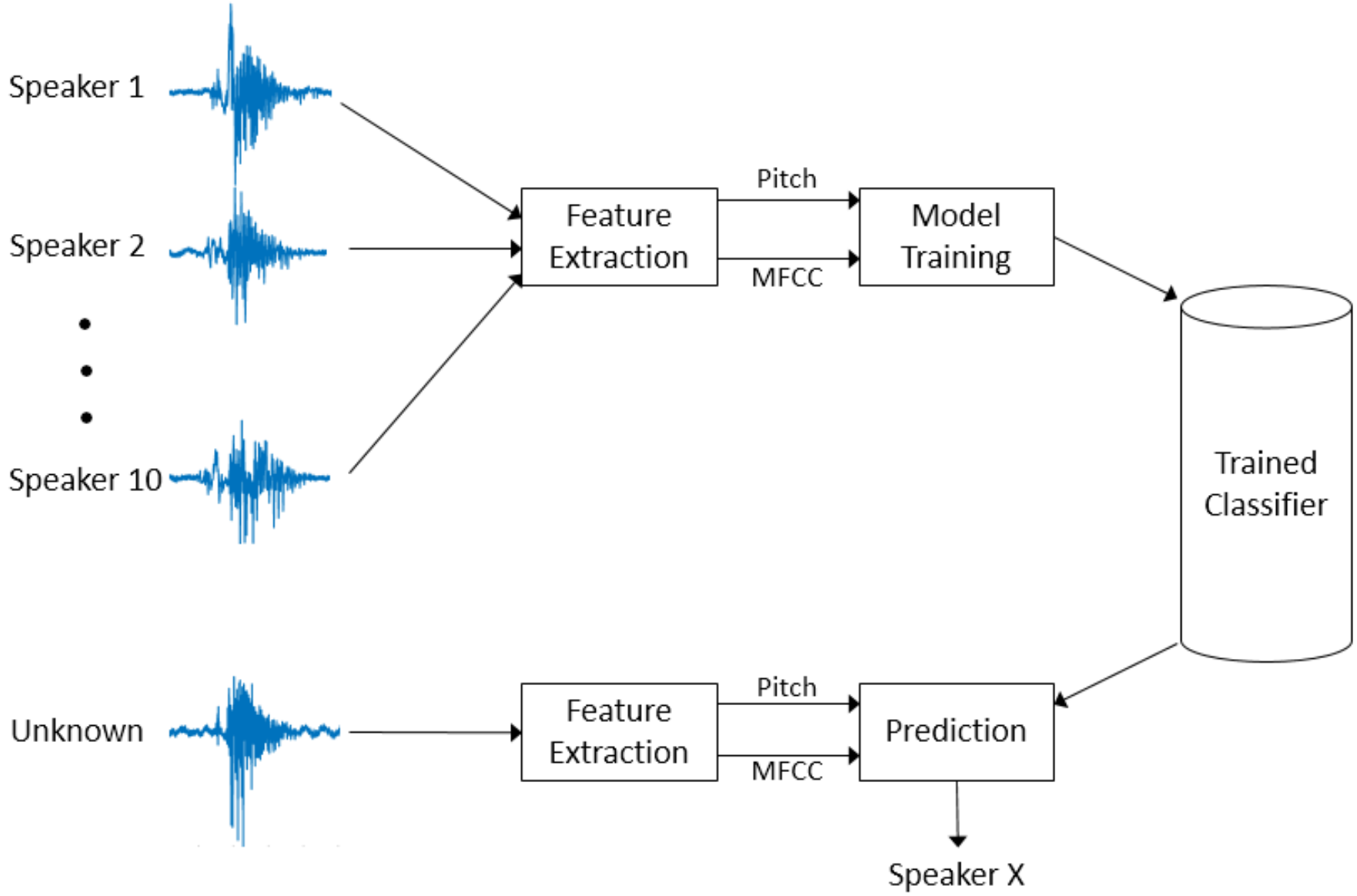
$$Y(z) = H(z)\ X(z)$$

フィルタ：声道
（MFCC）

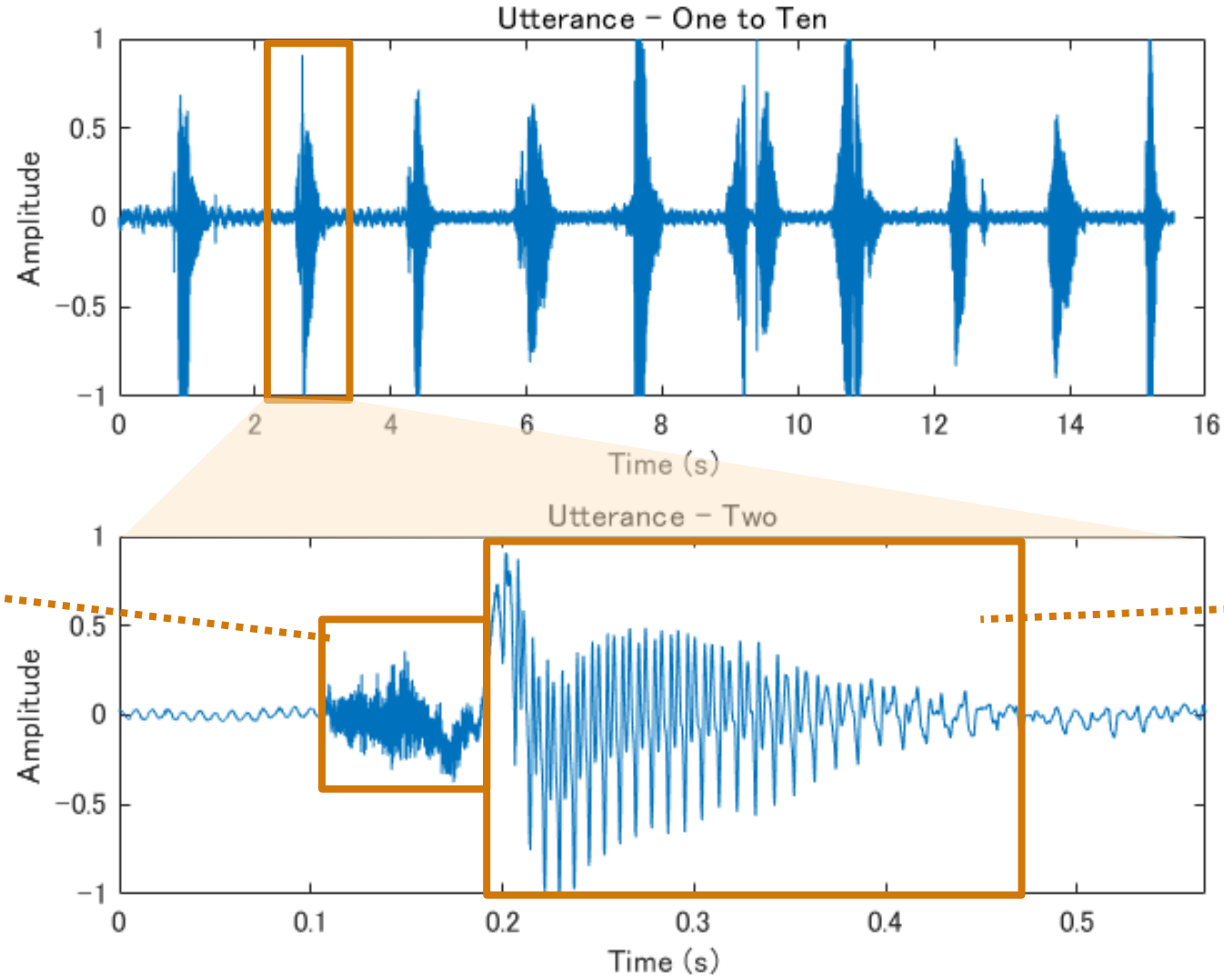


肺からの空気が**声帯**を震わせ
その信号が**声道**を通り**発音**する

demo：MFCCとPitchを用いた話者の推定



音声"Two"の、有声音部と無声音部



子音/T/
(無声音部)

母音/UW/
(有声音部)

ゼロクロスの頻度大：無声音
ゼロクロスの頻度小：有声音

有声音部のピッチ推定: 相互相関法

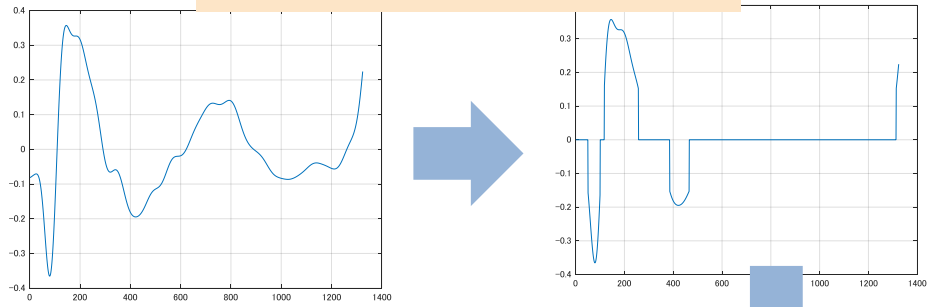
1.音声信号にセンタークリップ処理
➔声道の応答に起因する微小な振動成分を除去

2.信号分解能確保のためアップサンプリング
➔信号を補間する

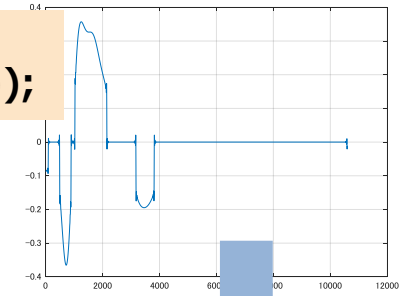
3.自己相関を計算
➔サンプルのラグを判定

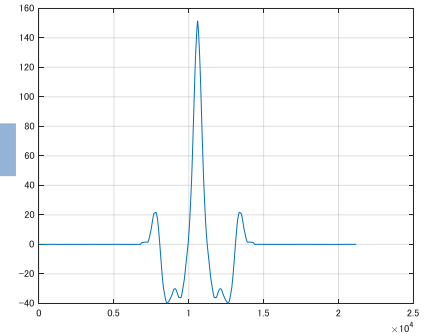
4.ラグ値からピッチ周期を換算

```
% Center-clip signal
x(abs(x)<clipLevel) = 0;
```

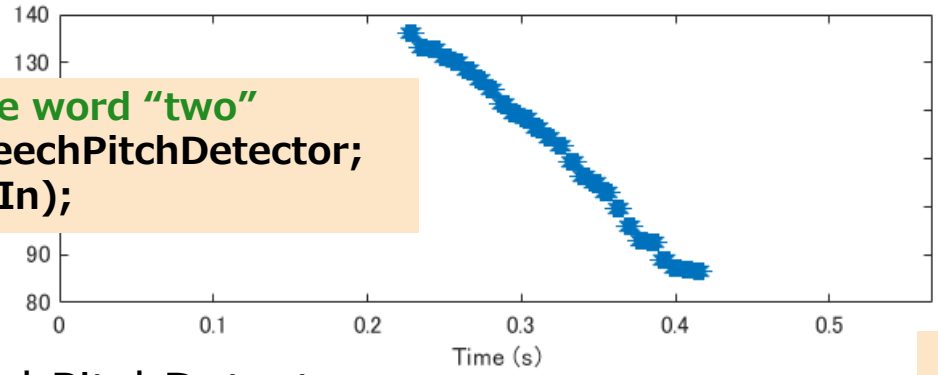


```
% Upsample the signal for
x = step(FIRInterpolatorFour,x(:,1));
```





```
% Perform autocorrelation
y = xcorr(x);
```




```
% Apply pitch detection to the word “two”
pD = audiopluginexample.SpeechPitchDetector;
[~,pitch] = process(pD,audioIn);
```

>> edit audiopluginexample.SpeechPitchDetector

データセット（Census Database: AN4）

http://www.speech.cs.cmu.edu/databases/an4/

Databases

- MicArray
- AN4
- Let's go
- CMU-SIN
- PDA
- RM1

CMU Robust Speech Recognition Group: Census Database

This database, also known as AN4 and as the Alphanumeric database, was recorded internally at CMU circa 1991. It has been used in several theses over the years.

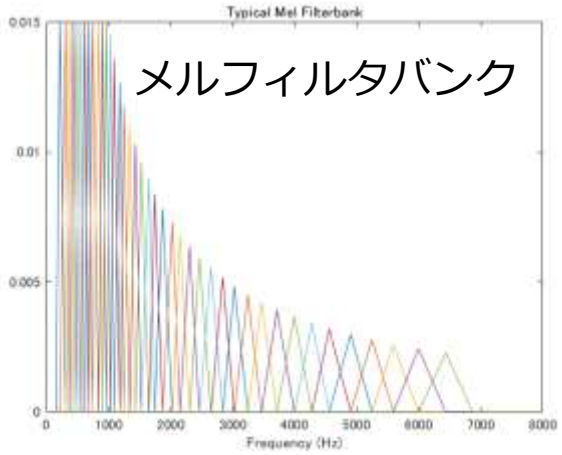
This database is made available subject to the license terms.

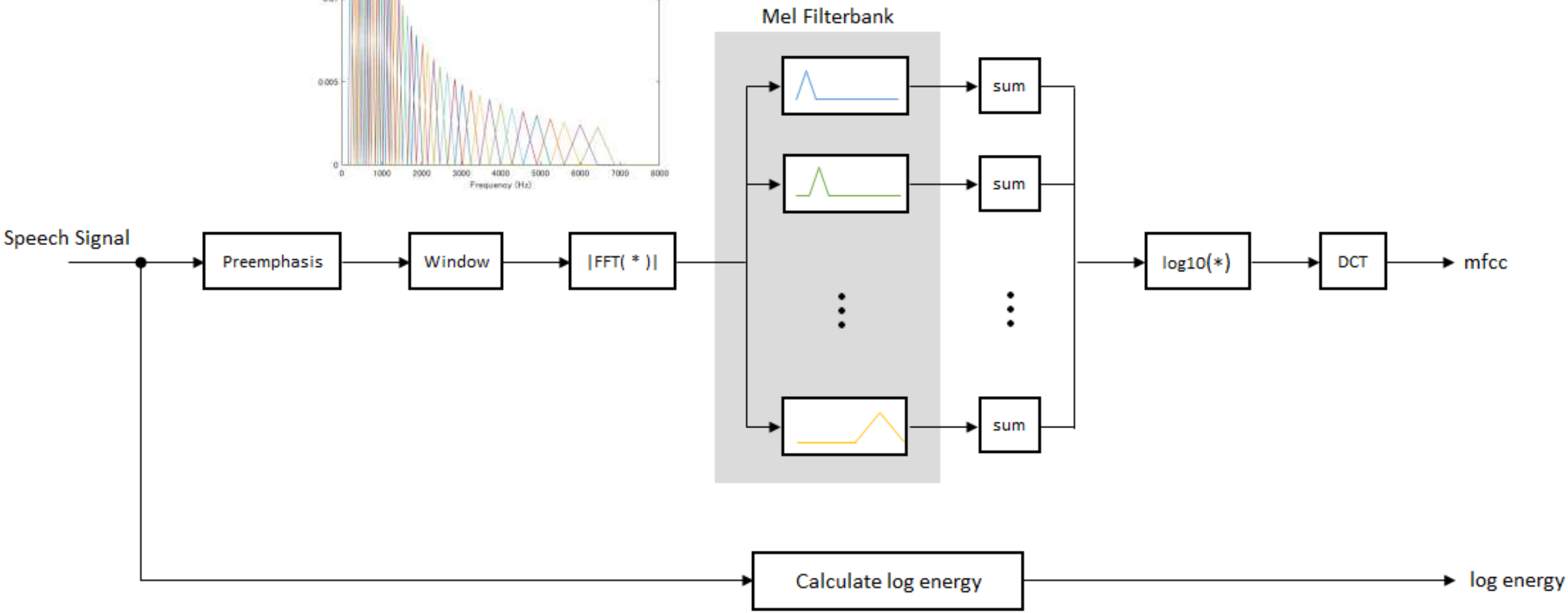
CMU Census Database

- Description of database
  - README
- Database (audio files and transcriptions)
  - Raw audio (.raw) format, big endian byte order (64 M)
  - Raw audio (.raw) format, little endian byte order (64 M)
  - NIST's Sphere audio (.sph) format (64 M)
  - Mel cepstra (.mfc) encoded format (14 M)

This page is maintained by Evandro Gouvêa (egouvea+cmusphinx@cs.cmu.edu)
The Robust Speech Recognition Group is part of the Sphinx Speech Group at Carnegie Mellon University

MFCC演算ステップ


Typical Mel Filterbank
メルフィルタバンク
0.015
0.01
0.005
0
0
1000
2000
3000
4000
5000
6000
7000
8000
Frequency (Hz)


Mel Filterbank
Speech Signal
Preemphasis
Window
|FFT(*)|
sum
sum
sum
log10(*)
DCT
mfcc
Calculate log energy
log energy

音声全データ

テスト用
学習用
女性
fash
fbbh
fclc
fejs
ffmm
男性
mblb
mblw
mbmg
mcel
mcen

データセットから取得した特徴量（抜粋）

ピッチ　　MFCC1～13　　ラベル

	Filename	Pitch	MFCC1	MFCC2	MFCC3	MFCC4	MFCC5	MFCC6	MFCC7	MFCC8	MFCC9	MFCC10	MFCC11	MFCC12	MFCC13	Label
1	'an251-fash-b.raw'	210.8731	-4.0064	-1.0502	2.2130	1.5944	0.1636	0.5905	-0.1960	0.4518	-0.7614	-0.0169	-0.0412	-0.2525	0.0019	'fash'
2	'an251-fash-b.raw'	214.4054	-3.5244	-1.7537	1.7664	1.8763	0.1116	0.3083	-0.2133	0.3668	-0.5598	0.2236	-0.1668	-0.3488	-0.1594	'fash'
3	'an251-fash-b.raw'	213.3333	-2.9804	-1.5177	1.4508	1.5171	0.0459	0.2228	-0.1571	0.3317	-0.5655	0.0566	-0.2429	-0.4489	-0.2143	'fash'
4	'an251-fash-b.raw'	213.3333	-2.4698	-1.4726	1.3716	1.5239	0.0530	0.1686	-0.2833	0.1835	-0.6909	-0.0505	-0.2472	-0.3179	-0.1572	'fash'
5	'an251-fash-b.raw'	208.8091	-2.3291	-1.1027	1.3191	1.6805	-0.3529	0.1277	-0.5211	-0.0254	-0.6645	-0.0556	-0.3002	-0.2029	-0.2032	'fash'
6	'an251-fash-b.raw'	212.9784	-2.2898	-0.8004	0.9195	1.6846	-0.7346	0.1145	-0.6251	0.1587	-0.5014	-0.0298	-0.3634	-0.2166	-0.2122	'fash'
7	'an251-fash-b.raw'	207.4554	-2.3984	-0.7293	0.5258	1.5897	-0.7562	-0.0039	-0.6940	0.2292	-0.5705	0.0363	-0.3367	-0.0192	-0.0899	'fash'
8	'an251-fash-b.raw'	202.5316	-2.5082	-0.7673	0.2855	1.3773	-0.6114	-0.3232	-0.5542	0.3293	-0.6243	0.1565	-0.1616	0.2889	-0.0642	'fash'

ピッチとMFCCの値のスケールが異なるため
正規化が必要

データセットから取得した特徴量（抜粋: 正規化後）

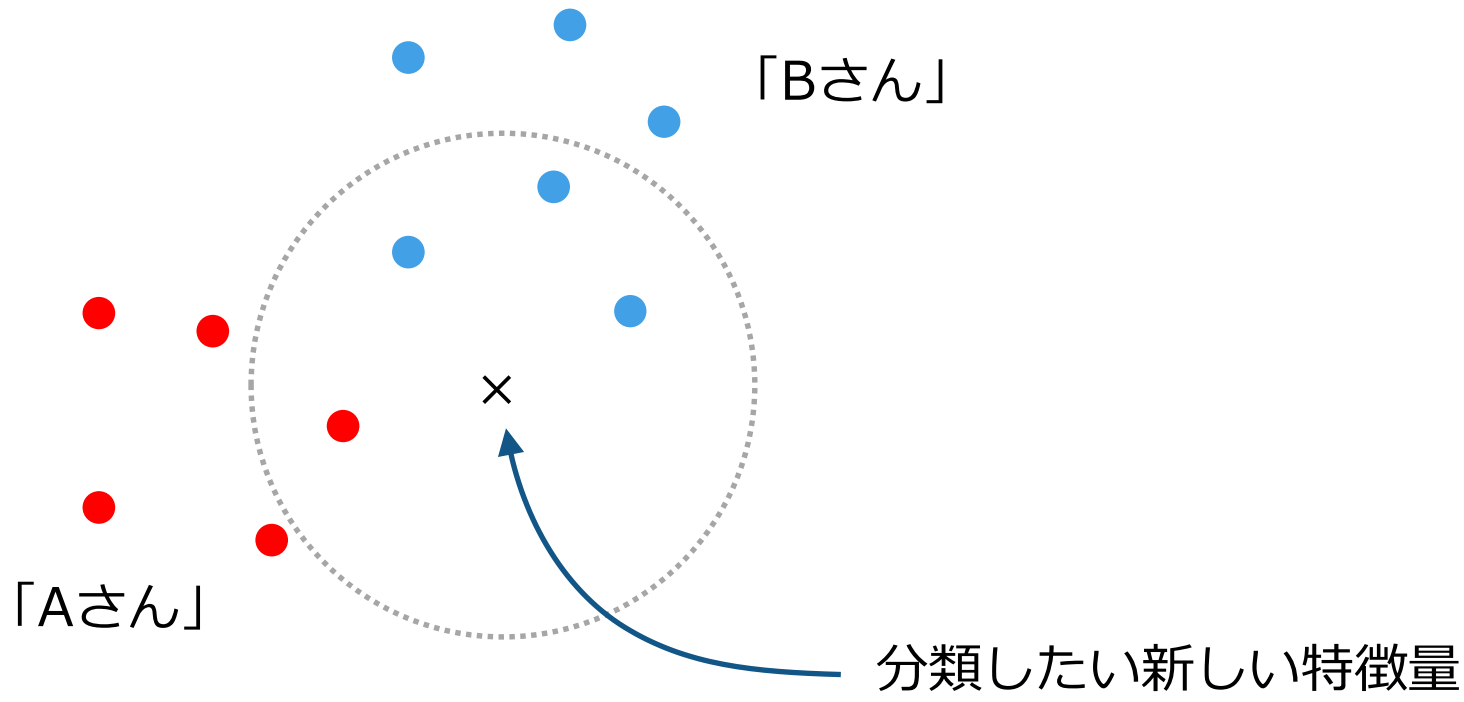
ピッチ　　MFCC1～13　　ラベル

	Filename	Pitch	MFCC1	MFCC2	MFCC3	MFCC4	MFCC5	MFCC6	MFCC7	MFCC8	MFCC9	MFCC10	MFCC11	MFCC12	MFCC13	Label
1	'an251-fash-b.raw'	0.9923	-1.7968	-1.7525	2.4287	0.3883	0.9017	0.8702	-0.2387	1.1208	-2.1363	-0.0567	-0.2265	-0.6421	-0.0378	'fash'
2	'an251-fash-b.raw'	1.0660	-1.5036	-2.3170	1.9590	0.6519	0.8234	0.2628	-0.2732	0.9087	-1.5951	0.5719	-0.6201	-0.9597	-0.6153	'fash'
3	'an251-fash-b.raw'	1.0437	-1.1728	-2.1276	1.6269	0.3161	0.7246	0.0788	-0.1611	0.8213	-1.6105	0.1352	-0.8587	-1.2898	-0.8116	'fash'
4	'an251-fash-b.raw'	1.0437	-0.8623	-2.0915	1.5436	0.3224	0.7352	-0.0378	-0.4127	0.4515	-1.9471	-0.1446	-0.8719	-0.8579	-0.6075	'fash'
5	'an251-fash-b.raw'	0.9492	-0.7767	-1.7946	1.4884	0.4688	0.1239	-0.1259	-0.8867	-0.0698	-1.8761	-0.1581	-1.0382	-0.4783	-0.7719	'fash'
6	'an251-fash-b.raw'	1.0363	-0.7528	-1.5521	1.0680	0.4727	-0.4508	-0.1544	-1.0941	0.3896	-1.4382	-0.0906	-1.2364	-0.5235	-0.8043	'fash'
7	'an251-fash-b.raw'	0.9210	-0.8188	-1.4950	0.6539	0.3840	-0.4834	-0.4092	-1.2315	0.5656	-1.6238	0.0824	-1.1525	0.1274	-0.3666	'fash'
8	'an251-fash-b.raw'	0.8182	-0.8856	-1.5255	0.4012	0.1854	-0.2653	-1.0963	-0.9527	0.8153	-1.7682	0.3964	-0.6039	1.1439	-0.2743	'fash'

```
>> m = mean(features{:,2:15});
>> s = std(features{:,2:15});
>> features{:,2:15} = (features{:,2:15}-m)./s;
```

KNN 分類器（K最近傍分類器）

原理：近くにある K個の点を探して、多数決を採る



KNN分類器（K最近傍分類器）

分類器： d = fitcknn(X, T, ‘NumNeighbors’, 4)

特徴量

1039.06	1691.41
230.469	3218.75
395.313	728.125
429.688	1410.16
910.156	1449.22
429.688	2344.53
457.031	1574.22

ラベル

1
2
5
3
1
4
3

予測： C = predict(d, Y)

予測されるラベル

1

新しい特徴量

1039.06	1464.84

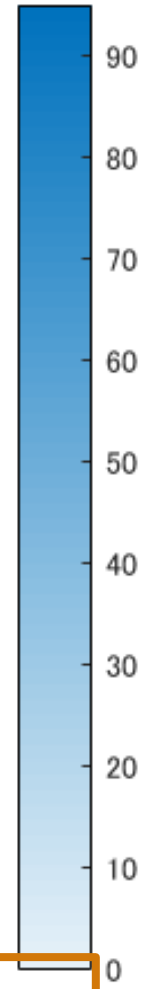
混合行列

HelperTrainKNNClassifier.m
（主要関数抜粋）

```
% Train a classifier
classificationKNN = fitcknn(...

% Perform cross-validation
partitionedModel = crossval(...

% Compute validation accuracy
validationAccuracy = 1 - kfoldLoss(...

% Compute confusion matrix
confMatrix = confusionmat(...
```

Confusion Matrix

話者（正解）	fash	fbbh	fclc	fejs	ffmm	mblb	mblw	mbmg	mcel	mcen
fash	95.01	1.22	0.1109	1.885	0.4435	0.1109	0.1109	0.2217	0.1109	0.7761
fbbh	0.9392	91.55	0.7735	1.823	2.762	0.3315	0.1657	0.6077	0.3867	0.663
fclc	0.2494	1.047	92.82	1.746	2.643	0.09975	0.09975	0.5985	0.3491	0.3491
fejs	0.5513	1.433	0.6064	94.21	1.929	0.05513	0.05513	0.7718	0.1103	0.2756
ffmm	0.356	2.492	1.513	1.78	91.86	0.4895	0.178	0.534	0.3115	0.4895
mblb	0	0.2691	0.2153	0.2153	0.7535	93.7	1.076	1.507	0.915	1.346
mblw	0.1143	0.3429	0.4571	0.6857	0	8	81.49	4.229	1.371	3.314
mbmg	0.09042	0.6781	0.9494	1.221	0.5877	0.9946	0.6781	92.27	0.9042	1.627
mcel	0	0.2939	0.5143	1.616	1.176	1.91	1.91	3.527	87.51	1.543
mcen	0	0.2117	0.2823	0.4234	0.8469	2.752	0.7057	2.329	1.059	91.39



話者（予測結果）

予測結果

HelperTestKNNClassifier.m
（主要部抜粋）

```
for idx = 1:length(Filenames)
    T = features_test(Fidx==idx,2:end);  % Rows that correspond to one file

    predictedLabels = string(predict(trainedClassifier,T(:,1:15))); % Predict
    totalVals = size(predictedLabels,1);

    [predictedLabel, freq] = mode(predictedLabels); % Find most frequently predicted label
    match = freq/totalVals*100;

    result_file.Filename = Filenames(idx);
    result_file.ActualSpeaker = T{1,15}{:};
    result_file.PredictedSpeaker = char(predictedLabel);
    result_file.ConfidencePercentage = match;

    result = [result; struct2table(result_file)]; %#ok
end
```

- 分割された音声データについて10クラスに予測
- 各予測結果の頻度スコアのトップを予測結果とする
- 頻度のパーセント換算値を”信頼パーセンテージ”とする
- 全テストファイルが実際の話者と合致

合致

	Filename	Actual Speaker	Predicted Speaker	Confidence Percentage
1	'cen1-fash-b.raw'	fash	fash	73.1844
2	'cen1-fbbh-b.raw'	fbbh	fbbh	56.0000
3	'cen1-fclc-b.raw'	fclc	fclc	77.0609
4	'cen1-fejs-b.raw'	fejs	fejs	87.0536
5	'cen1-ffmm-b.raw'	ffmm	ffmm	51.3410
6	'cen1-mblb-b.raw'	mblb	mblb	76.4368
7	'cen1-mblw-b.raw'	mblw	mblw	36.7188
8	'cen1-mbmg-b.raw'	mbmg	mbmg	77.6596
9	'cen1-mcel-b.raw'	mcel	mcel	63.5246
10	'cen1-mcen-b.raw'	mcen	mcen	82.9670
11	'cen2-fash-b.raw'	fash	fash	35.5140
12	'cen2-fbbh-b.raw'	fbbh	fbbh	64.7059
13	'cen2-fclc-b.raw'	fclc	fclc	78.3784
14	'cen2-fejs-b.raw'	fejs	fejs	86.9565
15	'cen2-ffmm-b.raw'	ffmm	ffmm	48.0114
16	'cen2-mblb-b.raw'	mblb	mblb	78.0822
17	'cen2-mblw-b.raw'	mblw	mblw	44.2478
18	'cen2-mbmg-b.raw'	mbmg	mbmg	78.0645
19	'cen2-mcel-b.raw'	mcel	mcel	50.3497
20	'cen2-mcen-b.raw'	mcen	mcen	54.3624

Agenda

- 背景
- センサー信号の場合
- 音声信号の場合
- 画像の場合
- 関連情報
- まとめ

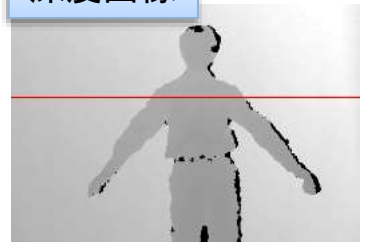
イメージの取得
Image Acquisition Toolbox™

- 業界標準のHWからの動画像取込み機能を提供
  - フレームグラバ (画像入力ボード)
    - Analog 入力
    - Camera Link® 入力
  - DCAM IIDC ® 1394 (FireWire)
  - GigE Vision ®互換カメラ
  - 一般的なUSB Webカメラ, IPカメラ(MATLABサポートパッケージ)
  - USB3 Vision互換カメラ(GenICam GenTL API経由)
  - Microsoft ® Kinect ® for Windows ®, Kinect ® for Windows ® v2
- カスタムアダプター開発キット付属
- Simulinkカメラブロックセット

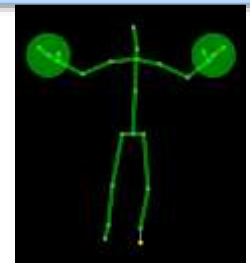
RGB画像＋骨格座標

認識した人

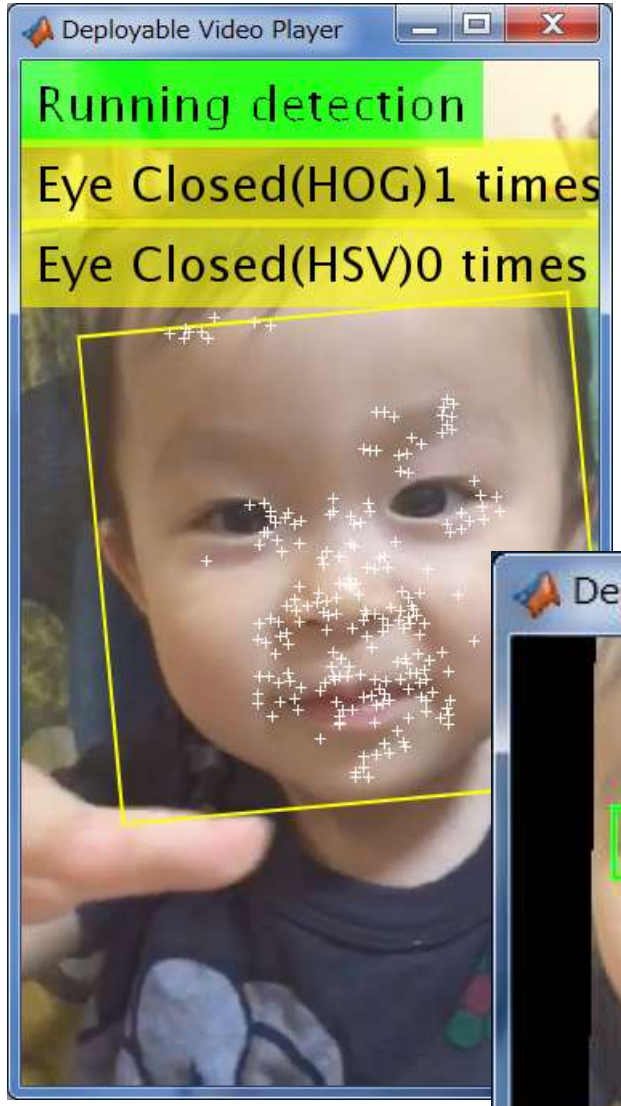
深度画像

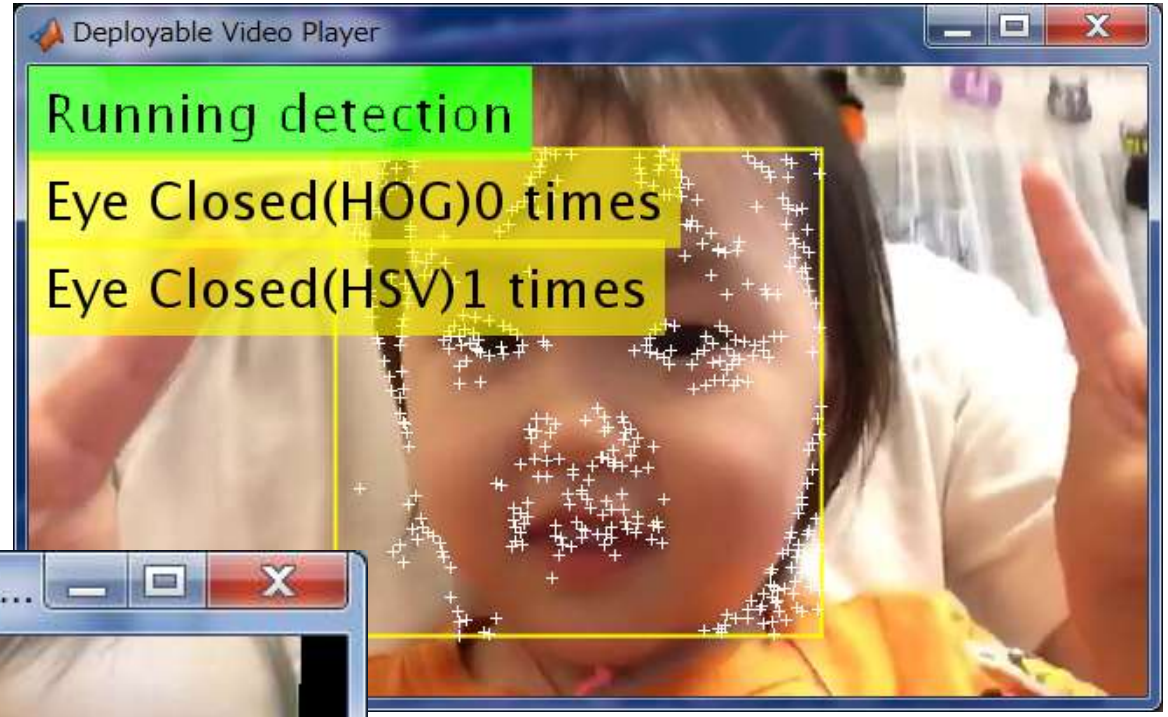


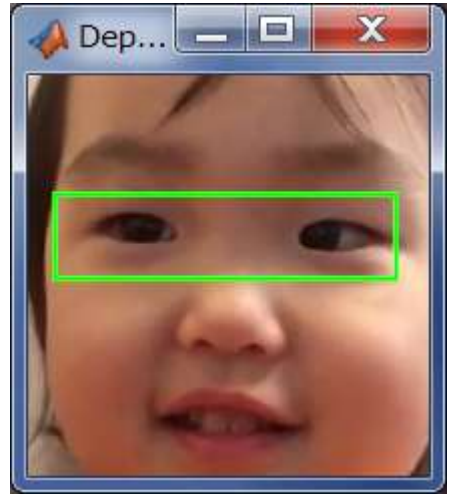
Kinect v2では
手の開閉状態も検出



Demo: HSV vs HOG
瞬きの検出に適した特徴量は？







■HSV特徴量（明度のヒストグラム）

```
>>hsv = rgb2hsv(img);
>>f2 = imhist(hsv(:,:,3))';
```

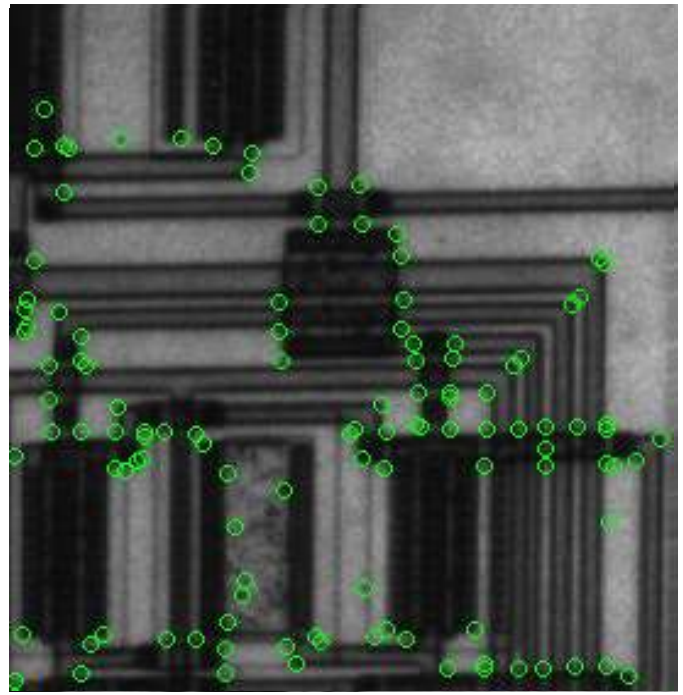
■HOG特徴量

```
>>f1 = extractHOGFeatures(img);
```

画像処理で用いられる特徴量

- **画像における局所特徴：**
  - エッジ、コーナー、ブロブ等の特徴的なパターン
  - 材質や色等が周辺と異なる場合に顕著に現れる

- **良い特徴とは？**
  - 検出が繰り返し可能
  - 特徴のパターンに特殊性がある
  - 位置推定が可能



Computer Vision System Toolbox

以下、ドキュメント抜粋

http://jp.mathworks.com/help/vision/ug/local-feature-detection-and-extraction.html

画像処理で用いられる特徴量（cont'd）

検出器と記述子の選択のための考慮事項

基準	推奨
特徴のタイプ	データに適した検出器を使用 ■例1) バクテリア細胞のイメージ: ブロブ検出器 ■例2) 都市部の航空写真: コーナー検出器
目的: ■キー ポイントのマッチング ■分類	■HOG 、SURF:分類用途 ■BRISK 、FREAK:イメージ間の対応、レジストレーション
歪み(スケールの変化、回転)のタイプ	■スケールの変化が生じない場合:コーナー検出器 ■スケールや回転などの大きな歪みがデータに含まれる場合: SURF 特徴検出器および記述子(ただし計算量は多い)
パフォーマンス要件: ■リアルタイム パフォーマンスが必要 ■正確度と処理速度	バイナリ記述子は、勾配ベースの記述子と比べて処理は高速。正確度の面で劣る。正確度を高めるには、同時に複数の検出器と記述子を使用

画像処理で用いられる特徴量（cont'd）

特徴のタイプに基づく検出関数の選択

検出器	特徴のタイプ	関数	スケール依存
FAST [1]	コーナー	detectFASTFeatures	×
最小固有値アルゴリズム [4]	コーナー	detectMinEigenFeatures	×
Harris [3]	コーナー	detectHarrisFeatures	×
SURF [11]	ブロブ	detectSURFFeatures	○
BRISK [6]	コーナー	detectBRISKFeatures	○
MSER [8]	強度が一様な領域	detectMSERFeatures	○

記述子メソッドの選択

記述子	バイナリ	関数とメソッド	不変要素		一般的な使用例	
			スケール	回転	画像間の対応	分類
HOG	×	extractHOGFeatures(I, ...)	×	×	×	○
LBP	×	extractLBPFeatures(I, ...)	×	○	なし	○
SURF	×	extractFeatures(I, 'Method','SURF')	○	○	○	○
FREAK	○	extractFeatures(I, 'Method','FREAK')	○	○	○	×
BRISK	○	extractFeatures(I, 'Method','BRISK')	○	○	○	×
■ ブロック ■ キー ポイントの周りのピクセル近傍	×	extractFeatures(I, 'Method','Block')	×	×	○	○

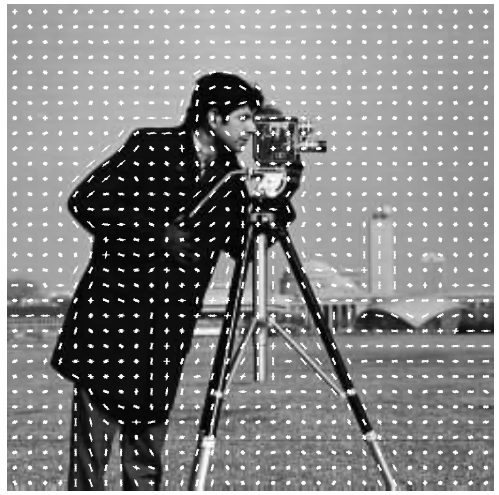
[1] Rosten, E., and T. Drummond. “Machine Learning for High-Speed Corner Detection.” 9th European Conference on Computer Vision. Vol. 1, 2006, pp. 430–443.
[3] Harris, C., and M. J. Stephens. “A Combined Corner and Edge Detector.” Proceedings of the 4th Alvey Vision Conference. August 1988, pp. 147–152.
[4] Shi, J., and C. Tomasi. “Good Features to Track.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. June 1994, pp. 593–600.
[6] Leutenegger, S., M. Chli, and R. Siegwart. “BRISK: Binary Robust Invariant Scalable Keypoints.” Proceedings of the IEEE International Conference. ICCV, 2011.
[8] Matas, J., O. Chum, M. Urba, and T. Pajdla. "Robust wide-baseline stereo from maximally stable extremal regions."Proceedings of British Machine Vision Conference. 2002, pp. 384–396.
[11] Bay, H., A. Ess, T. Tuytelaars, and L. Van Gool. “SURF: Speeded Up Robust Features.” Computer Vision and Image Understanding (CVIU). Vol. 110, No. 3, 2008, pp. 346–359.

画像処理で用いられる特徴量：例題（分類）

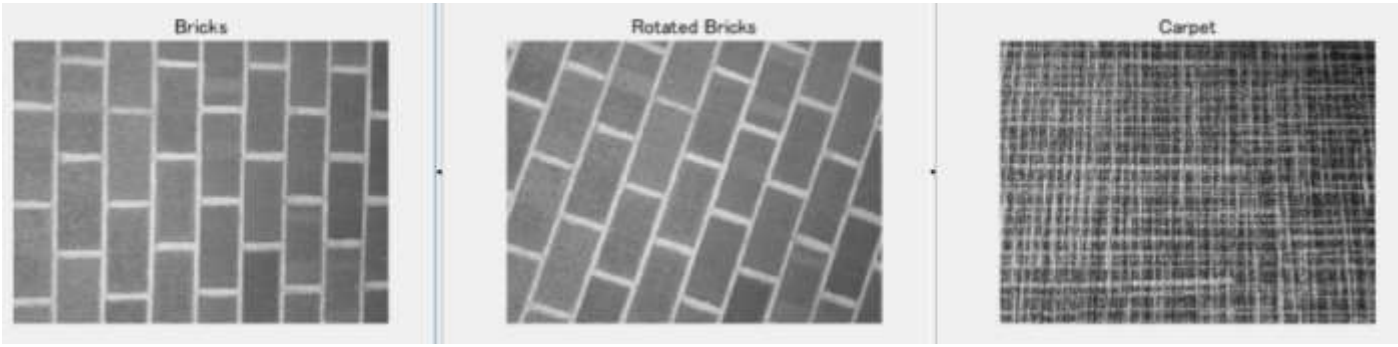
- 分類器の作成：**trainCascadeDetector**：
  - **Haar, LBP, HOG**に対応

- **HOG**:
  - オブジェクトの全体的な形状のキャプチャ向き
  - 人、自動車の検出等

- **Haar, LBP**:
  - スケールの細かいテクスチャ向き
  - 材質の識別、顔の検出等

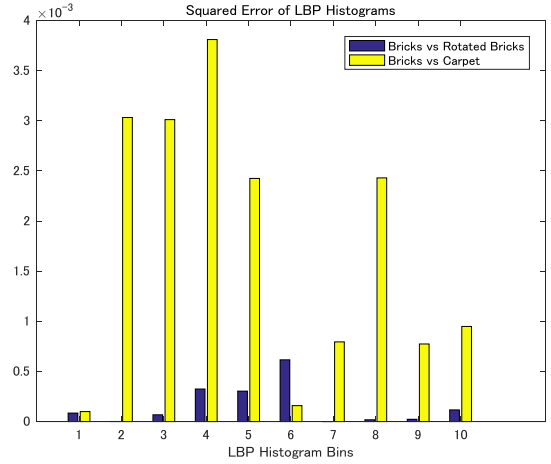


HOG特徴量



レンガ　　レンガ（回転）　　カーペット

LBP特徴量による
テクスチャの判別



- 黄：レンガ vsレンガ（回転）
- 青：レンガ vs カーペット

特徴点ベースのワークフロー

R2011b
R2012a
R2013a
R2013b
R2014a
R2015b

- 特徴点のマッチングを用いた、物体認識

対象物体

SURF/Harris/FAST/BRISK等の特徴点検出

SURF/FREAK/BRISK/HOG/LBP等の特徴量抽出

RANSACを用いた対応点推定

物体の検出

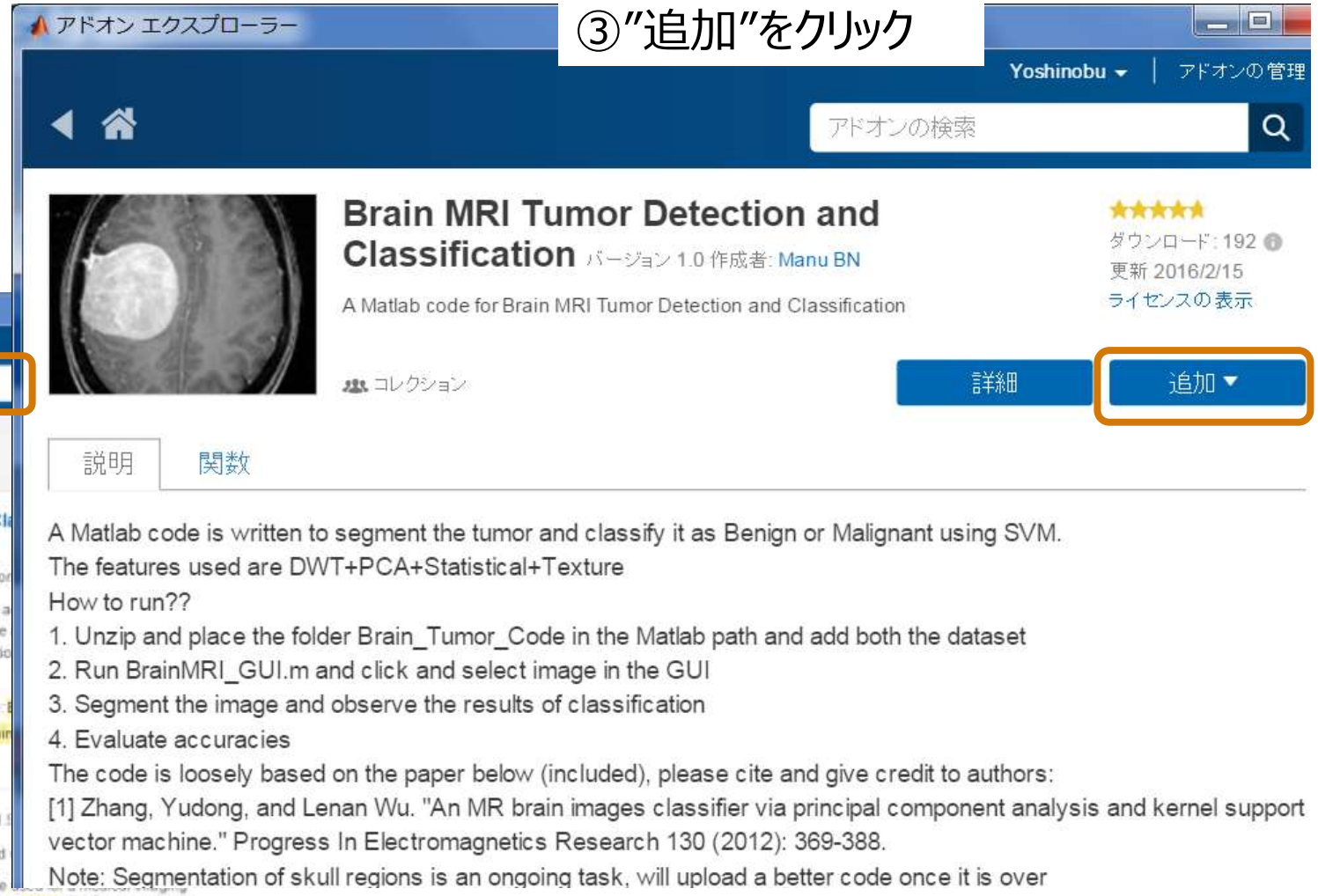
- 特徴点ベースのレジストレーション（位置合せ）

特徴量抽出の例題：（アドオンの追加）

①”アドオンの入手”をクリック

②キーワードを入力

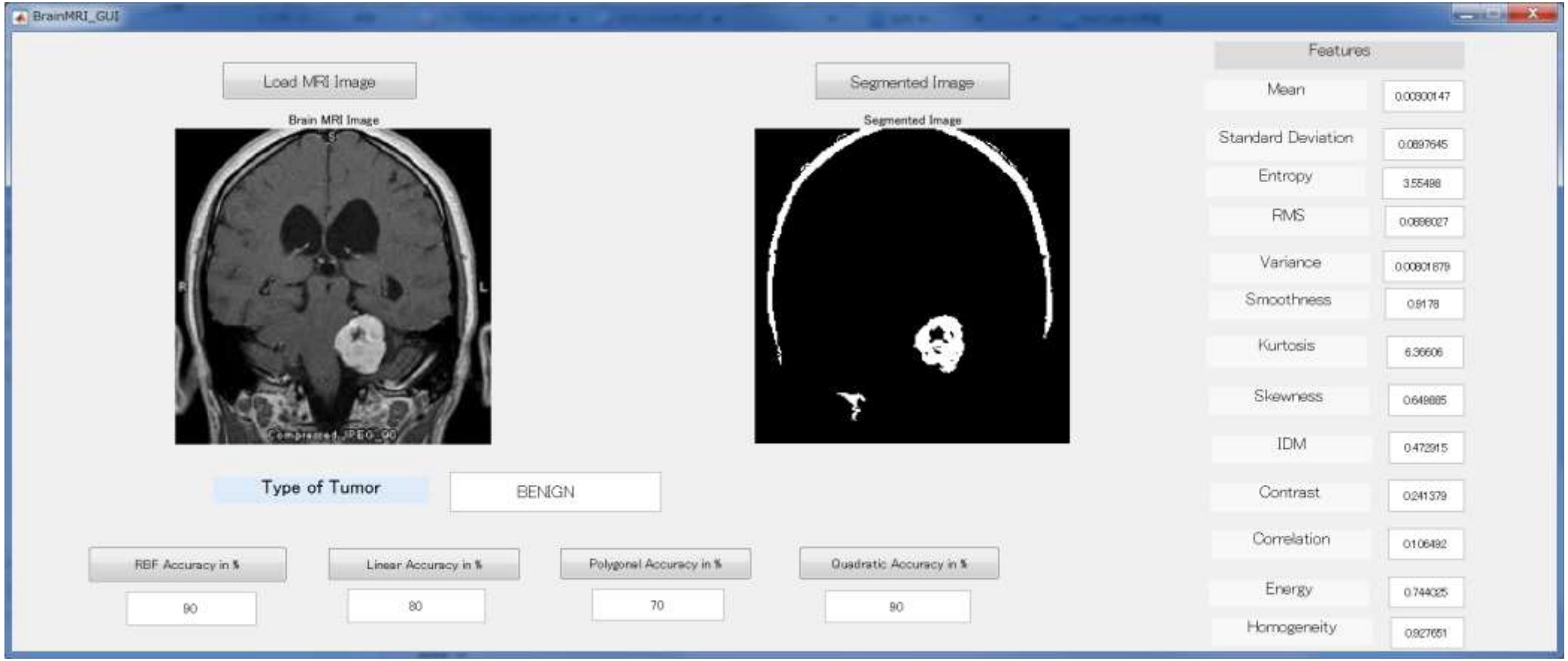
③”追加”をクリック



https://jp.mathworks.com/matlabcentral/fileexchange/55107-brain-mri-tumor-detection-and-classification

Brain MRI GUI

腫瘍部データの離散二次元Waveletから特徴量を抽出し、
腫瘍の良性・悪性の判定に応用した例（ **Wavelet Toolbox**を使用）

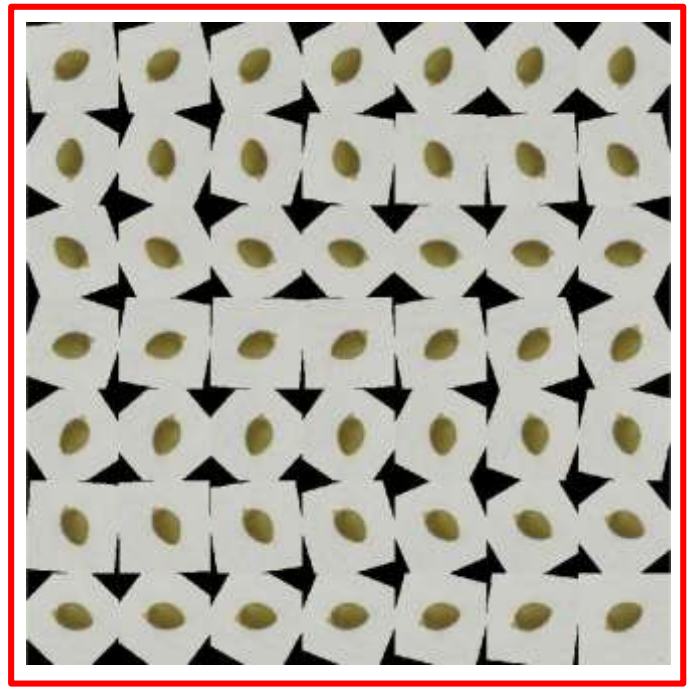


https://jp.mathworks.com/matlabcentral/fileexchange/55107-brain-mri-tumor-detection-and-classification

学習用画像の拡張

左右反転

任意の
反転



学習用画像の拡張

物体の色味の変更

物体の外形の変形

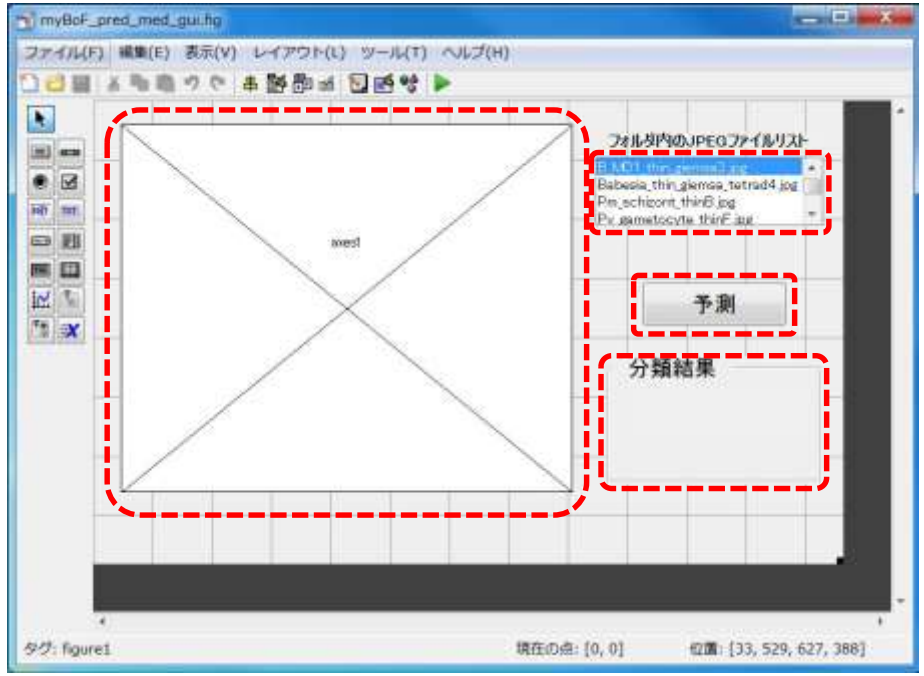
画像操作のさまざまな関数で、容易に画像の拡張を行っていただけます

補足：機械学習アルゴリズムの実装

ユーザーインターフェース

ユーザーインターフェース
設計環境 (GUIDE)

```
>> guide
```

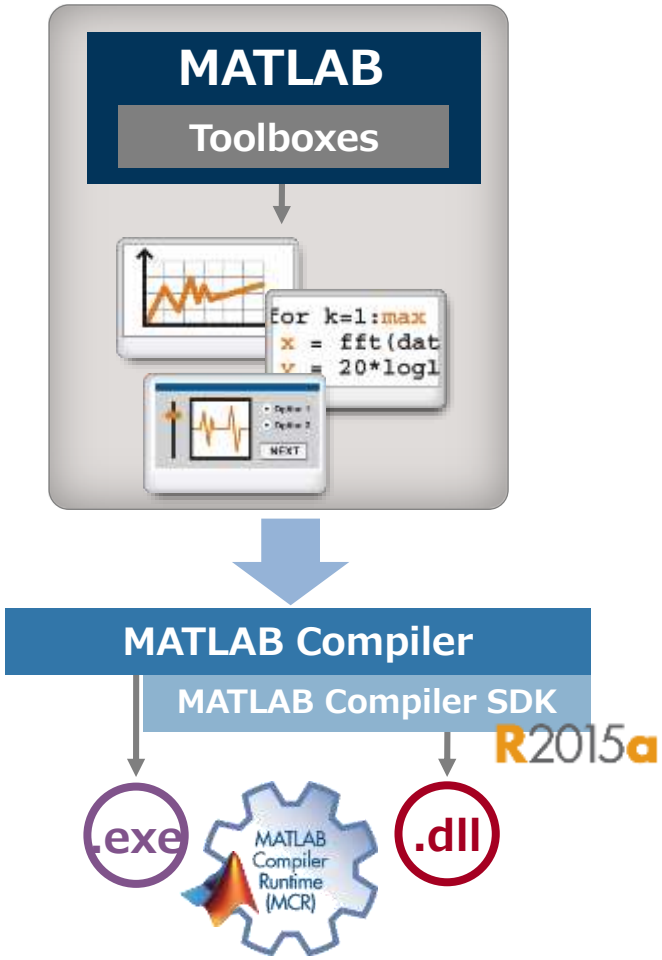


.fig ファイルが生成される

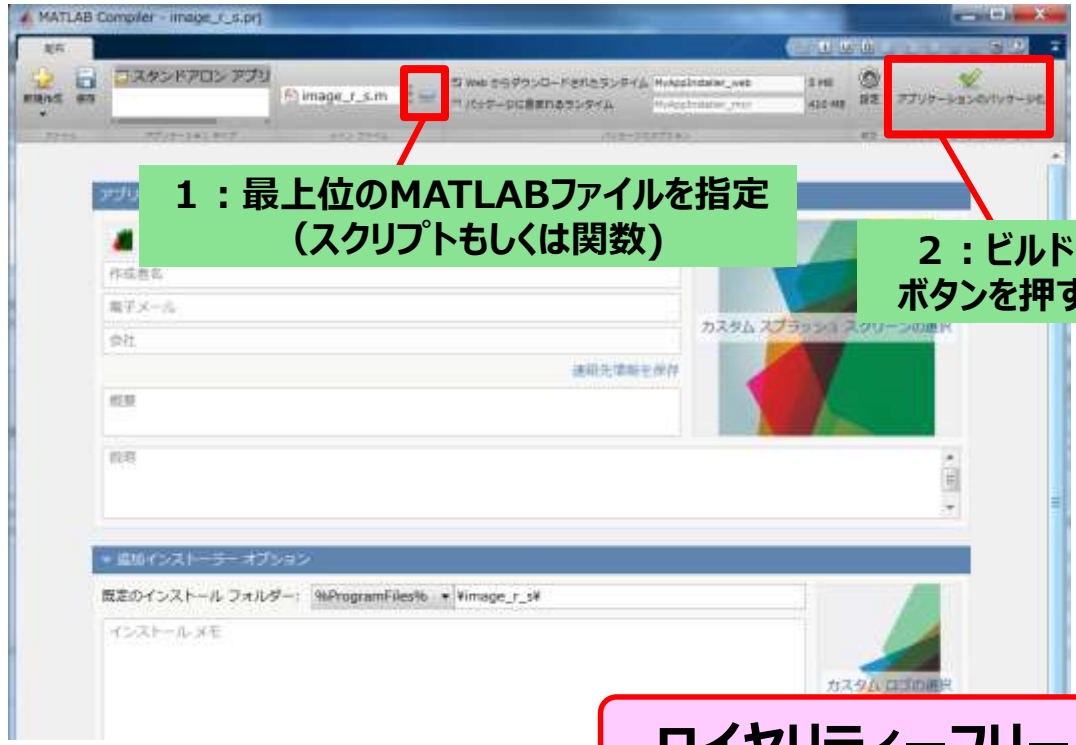
用いた画像のソース： http://www.cdc.gov/dpdx/index.html (Centers for Disease Control and Prevention)

補足：機械学習アルゴリズムの実装

MATLAB Compiler




```
>> deploytool
```
もしくは　アプリケーションタブから起動



ロイヤリティーフリー

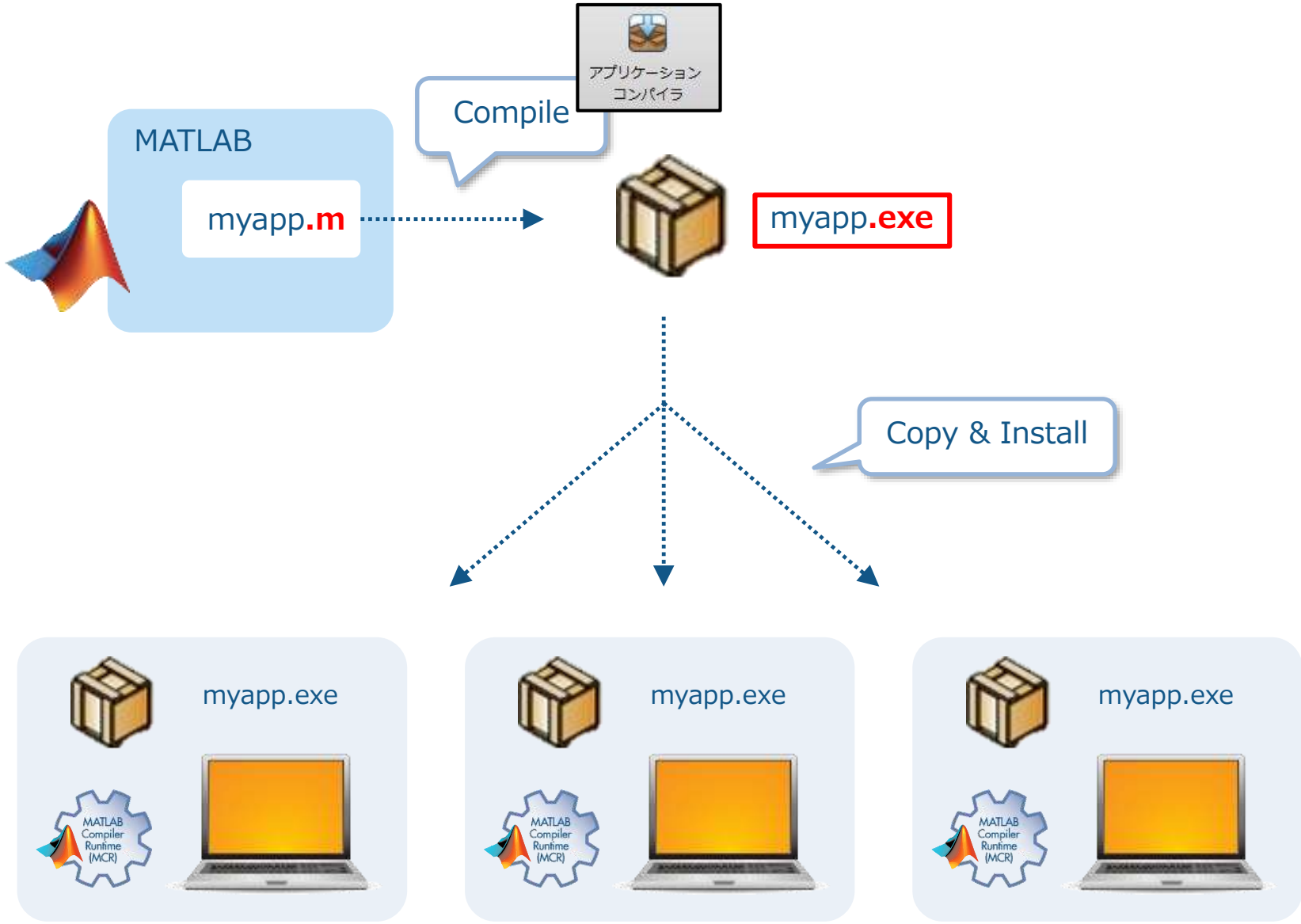
ほぼすべての関数をサポート ・ GUIやFigureも対応

制限：http://www.mathworks.co.jp/help/compiler/limitations-about-what-may-be-compiled.html

対応Toolbox関数：http://www.mathworks.co.jp/products/compiler/supported/compiler_support.html

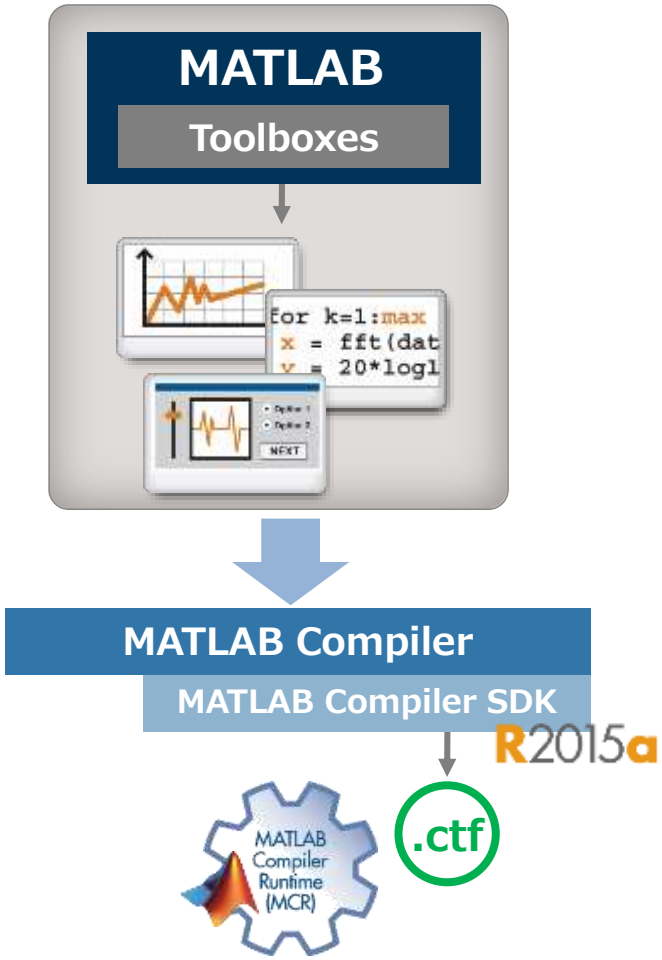
補足：機械学習アルゴリズムの実装

MATLAB Compiler



補足：機械学習アルゴリズムの実装

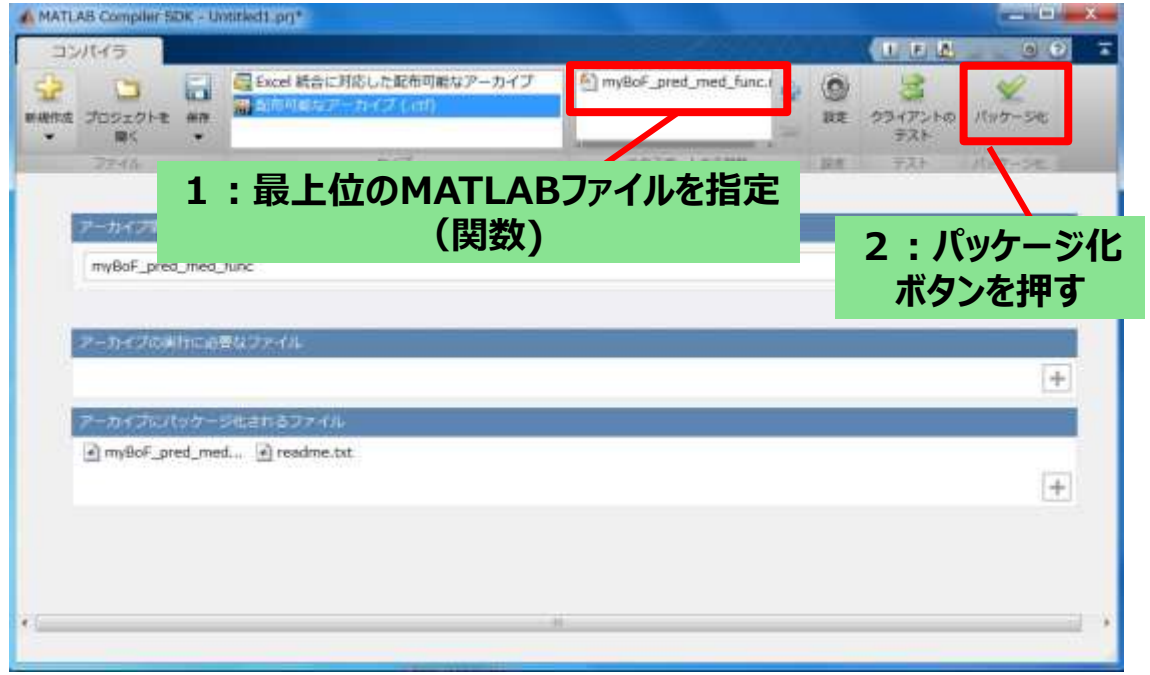
MATLAB Production Server



ほぼすべての関数をサポート

```
>> productionServerCompiler
```

もしくは　アプリケーションタブから起動



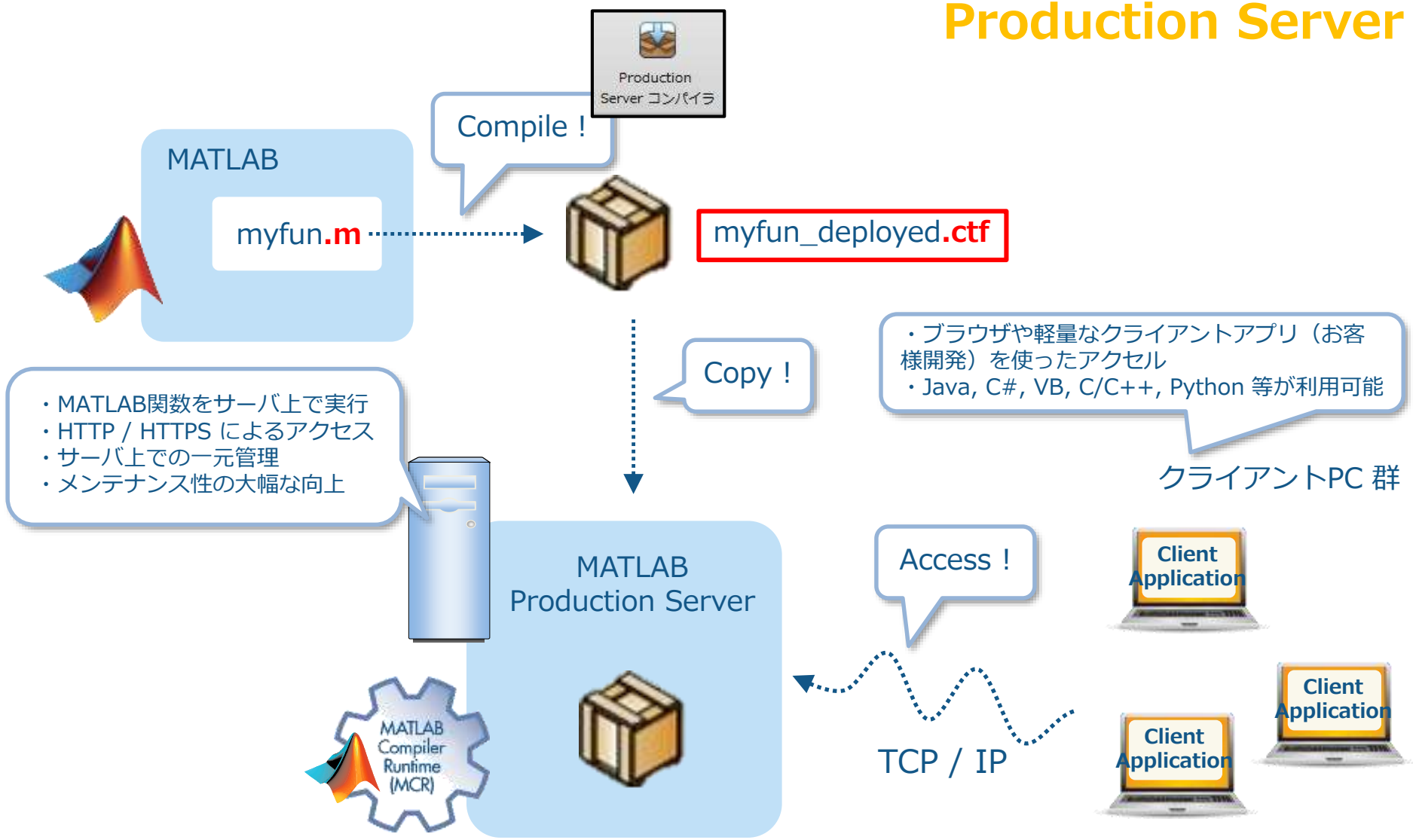
ロイヤリティーフリー

制限：http://www.mathworks.co.jp/help/compiler/limitations-about-what-may-be-compiled.html

対応Toolbox関数：http://www.mathworks.co.jp/products/compiler/supported/compiler_support.html

補足：機械学習アルゴリズムの実装

MATLAB Production Server



/auto_deploy/myfun_deployed.ctf

Agenda

- 背景
- センサー信号の場合
- 音声信号の場合
- 画像の場合
- 関連情報
- まとめ

エッジデバイス実装: Simulinkハードウエア連携機能(cont'd)

Arduino® Nano, Uno, Mega 2560

Raspberry Pi™
Raspberry Pi™2

Apple® iOS Devices

LEGO® MINDSTORMS® EV3

LEGO MINDSTORMS NXT

KINECT™ for Windows®

RTL-SDR Radio

Samsung® GALAXY Android™

BeagleBoard /PandaBoard

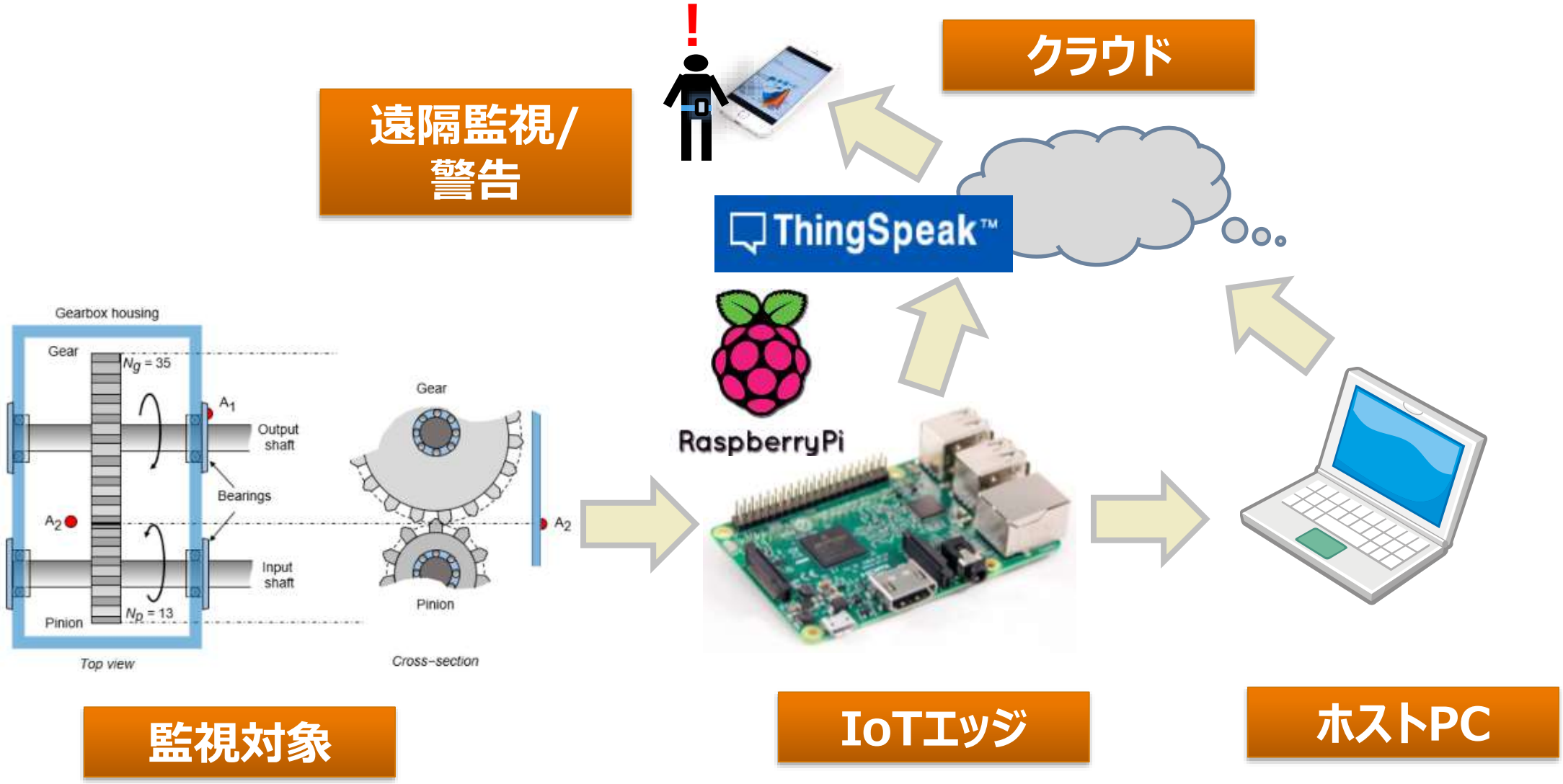
Digilent® Analog Discovery™

TurtleBot™

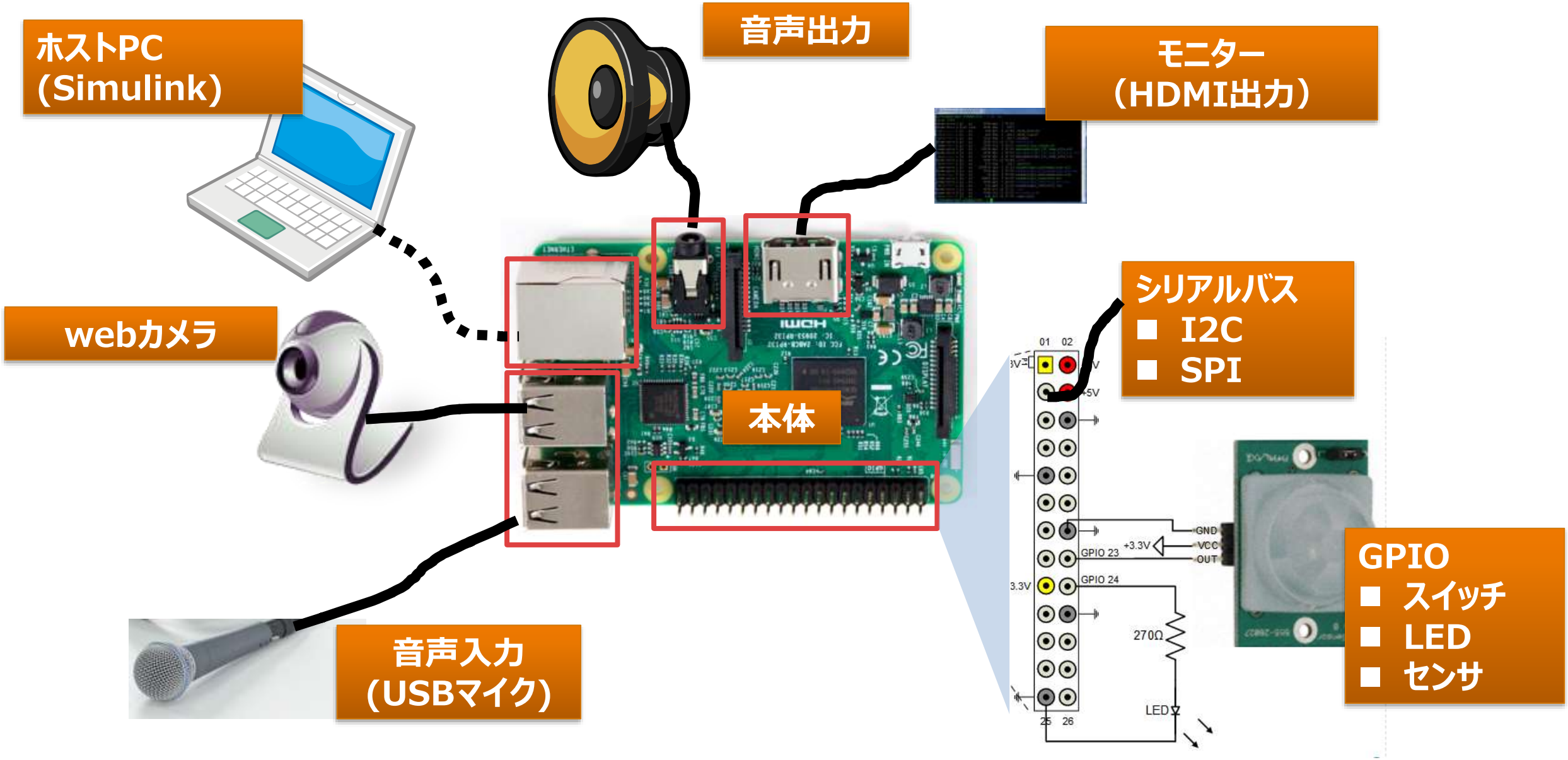
KINECT™ for Windows® V2

サポートされているハードウェア一覧
http://jp.mathworks.com/hardware-support

簡易IoTシステム概観



Raspberry Piの概観、インターフェイス



Raspberry Piの歴史

年	RasPi トピック	Simulink Support Package
2008	■ エベン・アプトン氏Raspberry Pi財団設立	
2011	■ 試作品完成	
2012	■ 初代Raspberry Pi リリース ■ 初日二時間で**1**万台を売りきる ■ 初日だけで**10**万台の発注	
2013	■ 累計**200**万台	■ リリース（R2013a）
2014	■ 累計**300**万台 ■ Compute moduleリリース	■ ThingSpeak Write (14a)
2015	■ Raspberry Pi 2, Zeroリリース ■ 累計**500**万台(Feb)	■ Pi 2対応 (15a) ■ MAC OS対応 (15a)
2016	■ Raspberry Pi 3 リリース ■ 愛知県稲沢市で製造開始 ■ 累計**1,100**万台(Nov)	■ Pi 3対応 (16a) ■ Linux OS対応 (16b)
2017	■ Compute module 3, Zero Wリリース ■ 累計**1,500**万台(Jul)	■ I2C,SPI,Serial（17a) ■ Audio read, logging (17b)
2018	■ 累計**1,900**万台(Mar)	■ Pi Zero W対応 (18a)

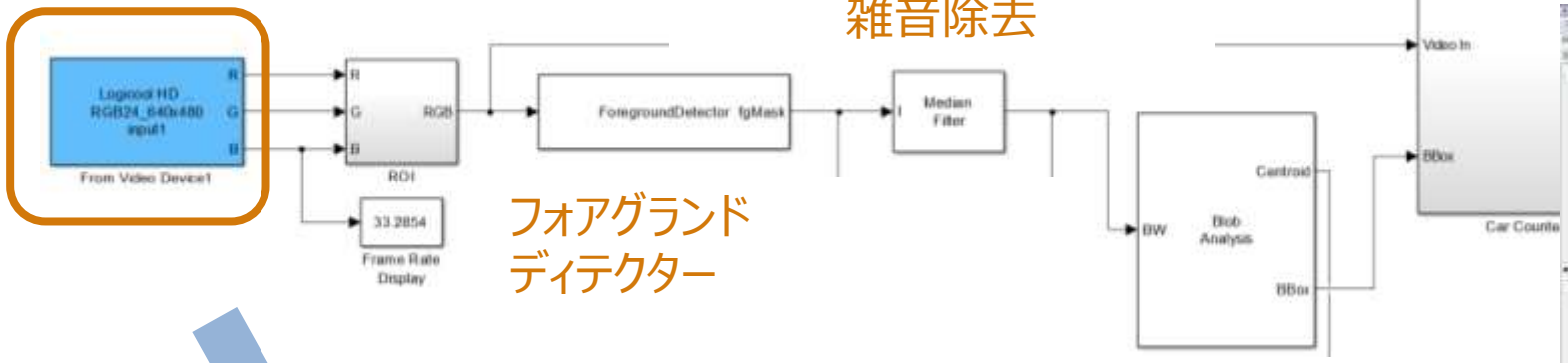
https://en.wikipedia.org/wiki/Raspberry_Pi

- リリース当初は、コンピュータサイエンスの授業利用を想定
- 学生をモチベートする基本方針
  - プログラミング可能
  - 楽しい
  - 小型で丈夫
  - 安価
- 現状教育用と産業用は半々(産業用途が増加傾向)

Raspberry Piによるトラフィックモニタ、分析

Raspberry Piによるトラフィックモニタ、分析（cont'd）

PCとWebカメラで動作チェック

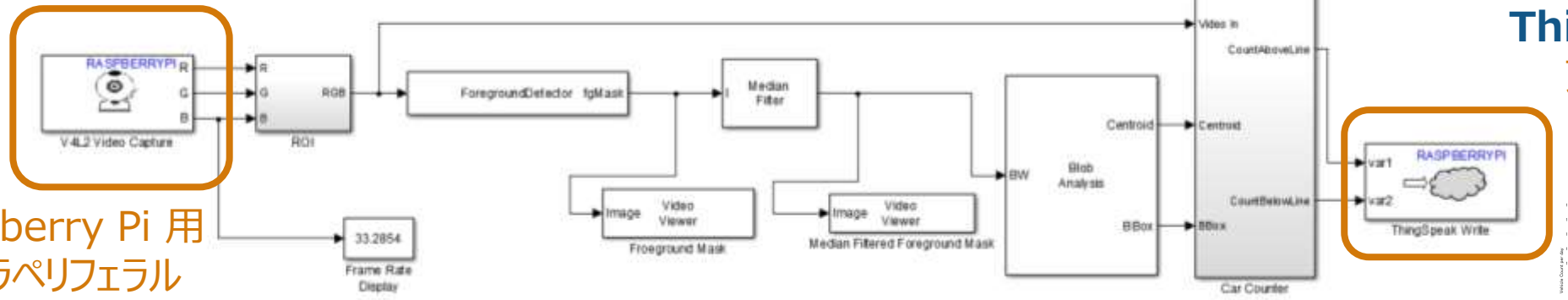


雑音除去

フォアグランドディテクター

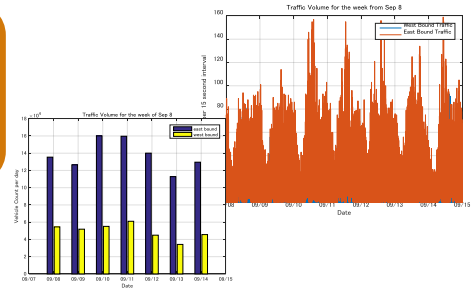
Blob解析

台数計測

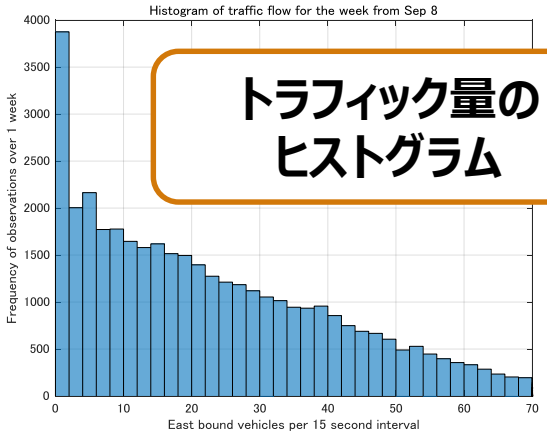


Raspberry Pi 用カメラペリフェラルブロック

必要なデータを**ThingSpeak**でアップロード

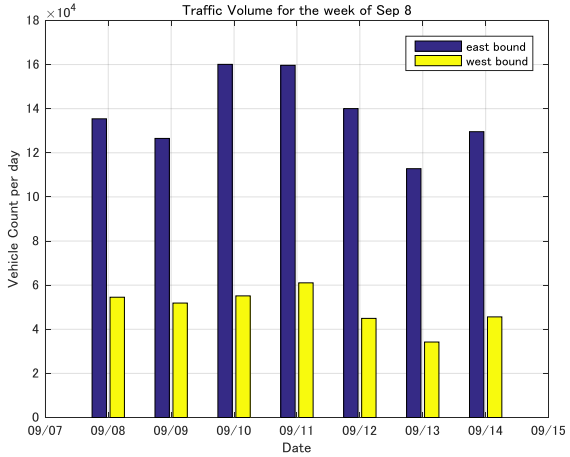


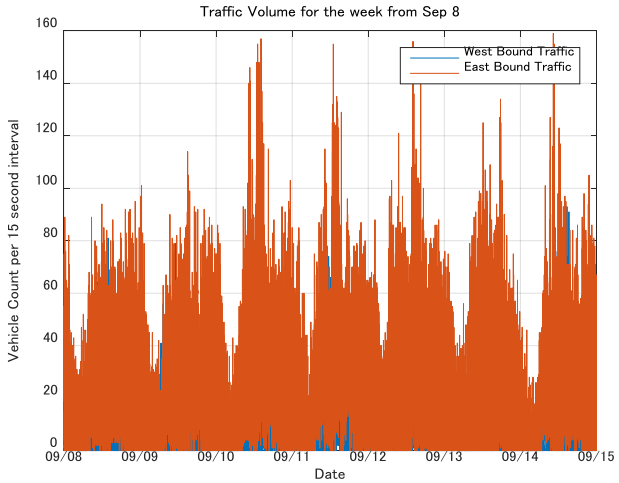
Raspberry Piによるトラフィックモニタ、分析（cont'd）



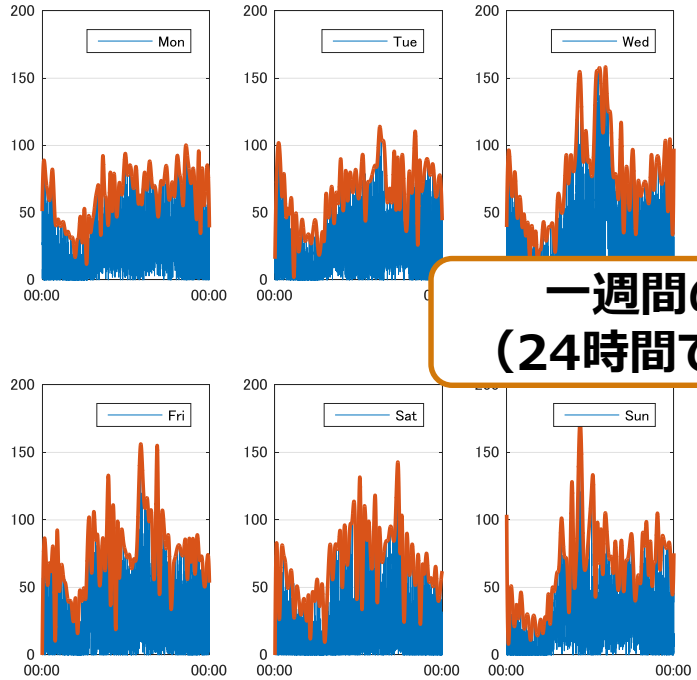
トラフィック量の
ヒストグラム

一週間の交通量
（日ごとの推移）



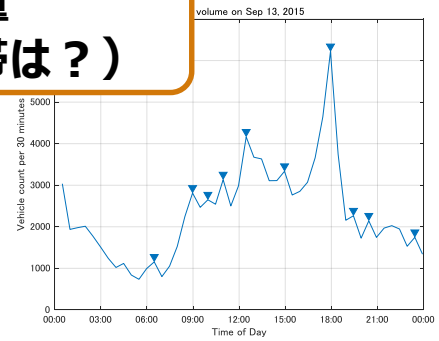


桜通の1週間の交通量
（西行きvs東行き）



一週間の交通量
（24時間で見た推移）

日曜日の交通量
（交通量の多い時間帯は？）



データ収集&解析を手軽に！ ThingSpeak

データ収集＆解析Webサービス

収集

コマンド 1行
専用ブロック

可視化＆解析

-WEB上でデータの確認が可能
-公開範囲Private/Public選択
-MATLAB関数で解析＆可視化

アクション

-csv形式でデータ保存
-条件を満たしたらツイート等
様々なアクション機能

IoTプラットフォームThingSpeak利用のメリット

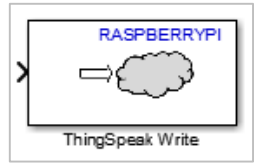
- ●インターネットでどこからでもデータ確認&解析
- ●Arduino/Raspberry Piとの活用で、低価格な実験系を構築
- ●プログラムの完了/データ異常をスマホでチェック

●無料ではじめられます！！

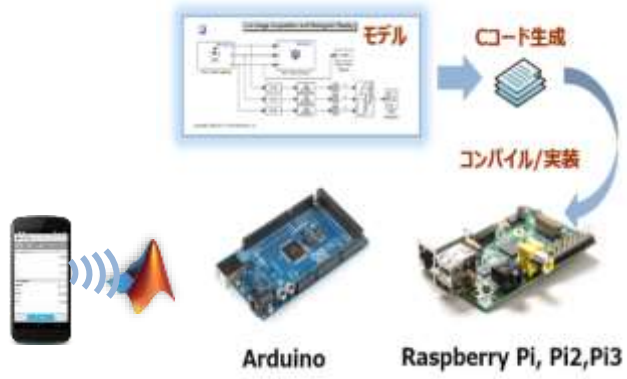
●準備は簡単3ステップ

-アカウント/チャネル作成
-関数群のインストール
-専用コマンド/ブロック
で簡単利用

thingSpeakWrite



●低価格HWとの相性が◎



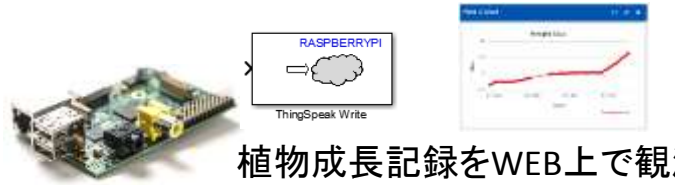
ThingSpeakはいろいろな場面で活用できます

実験での活用

測定器の自動制御にthingSpeakWrite
コマンド1行追加で無信号状態をすぐに把握。「数時間後気づいたらテストが進んでいなかった」を避け、離れた実験室との無駄な行き来もなくなります。

気象・農水産分野での活用

人やPCでやっていた温度/湿度等のデータ収集や植物の成長などの単純かつ長期間の計測。Arduino/Raspberry pi + ThingSpeakにより、数千円で自動観測システムを構築できます。



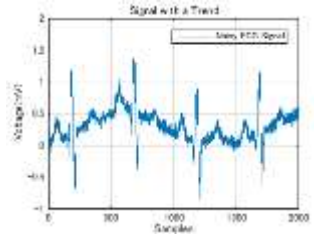
植物成長記録をWEB上で観測

情報・統計解析分野での活用

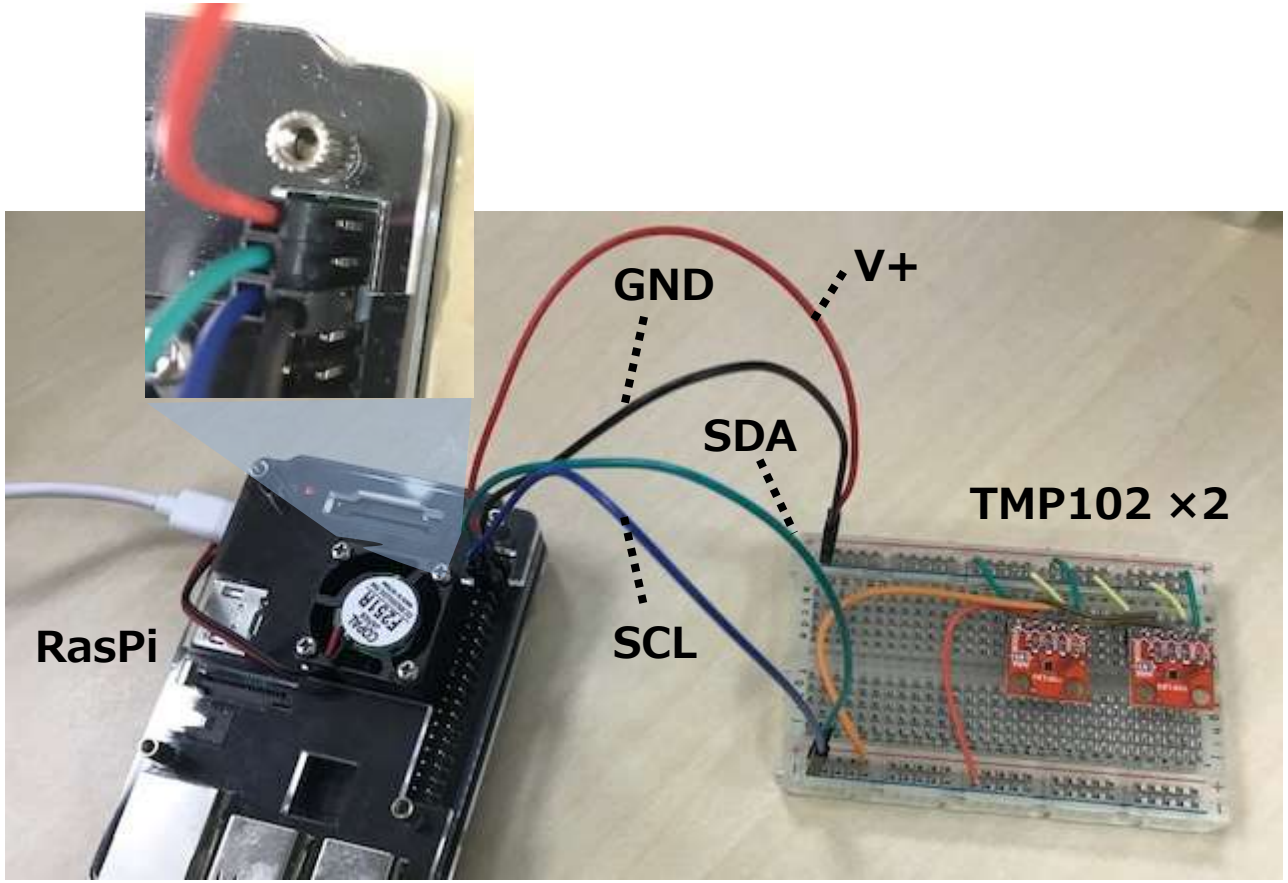
カメラ画像から車を検知するコンピュータビジョンのアルゴリズムをRaspberry Piに実装し、道路の交通量を観測。ThingSpeakでは取得した統計データを「時間ごと」「曜日ごと」などのソートやMATLAB関数を用いて解析することができます。

医療・スポーツ科学分野での活用

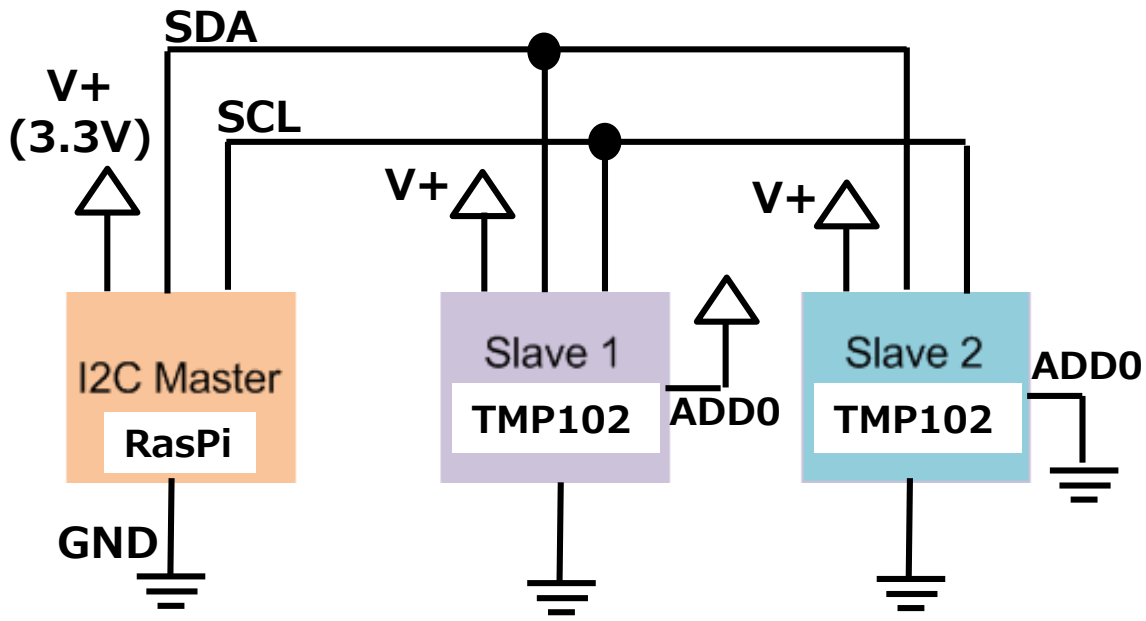
生体信号モニタリングで必要な長時間のデータ収集もThingSpeak上で行うことができます。その他、加速度、温度、位置情報のデータ等関連パラメータの同時収集/解析に利用可能。異常値に対してアクション機能も活用できます。



例：温度センサーのモニタリング、アラートの配信
～Raspberry Pi とセンサデバイス間の配線～



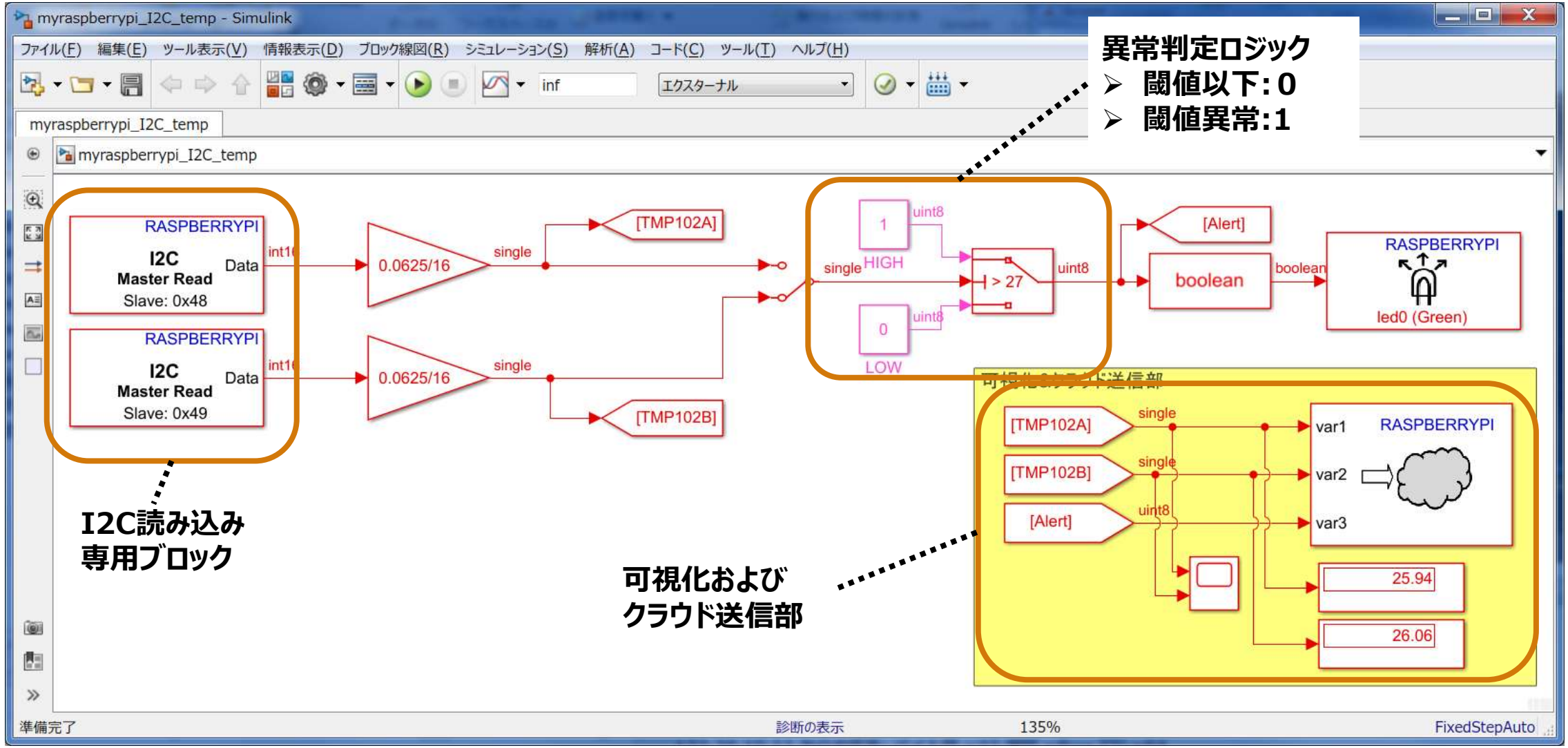
RasPiと温度センサ間の配線図



tmp102アドレスピン設定
（同時に4台まで接続可能）

アドレス	ADD0ピン
0x48	GND
0x49	V+
0x4a	SDA
0x4b	SCL

例：温度センサーのモニタリング、アラートの配信
～Simulinkモデル概観～



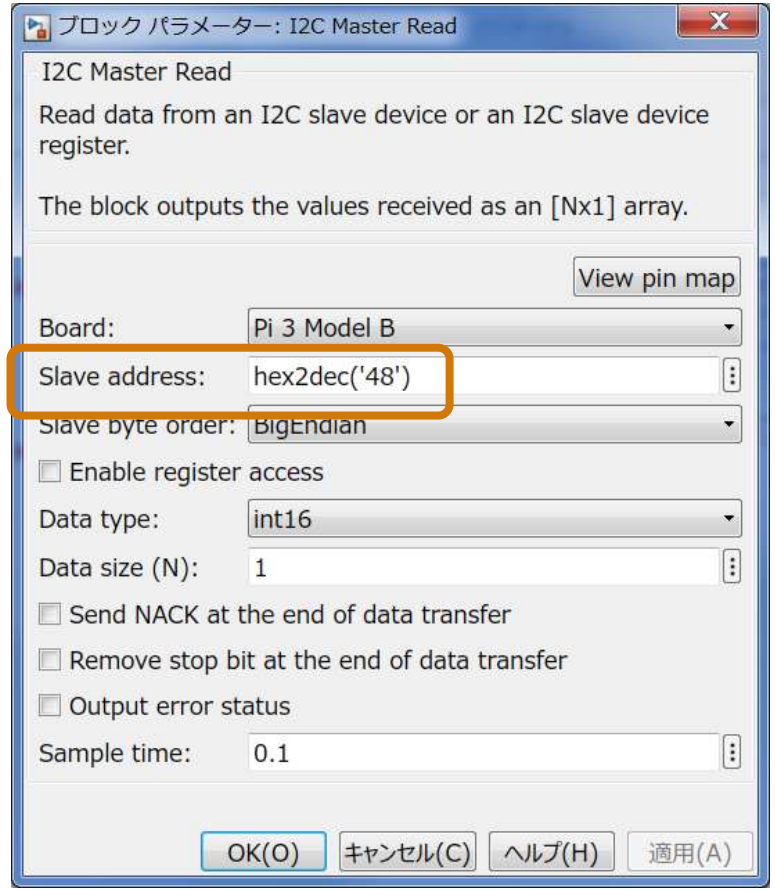
異常判定ロジック
- 閾値以下：0
- 閾値異常：1

I2C読み込み
専用ブロック

可視化および
クラウド送信部

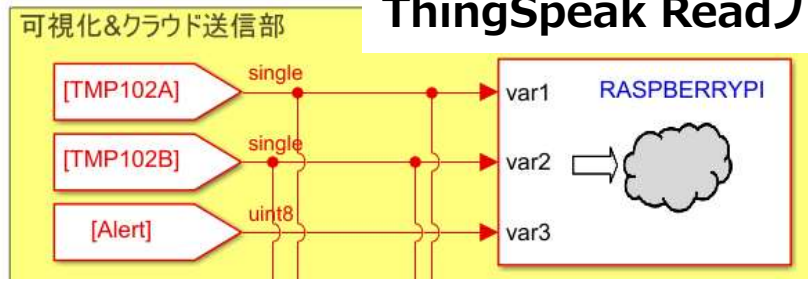
例：温度センサーのモニタリング、アラートの配信
～専用ブロックのパラメータ設定～

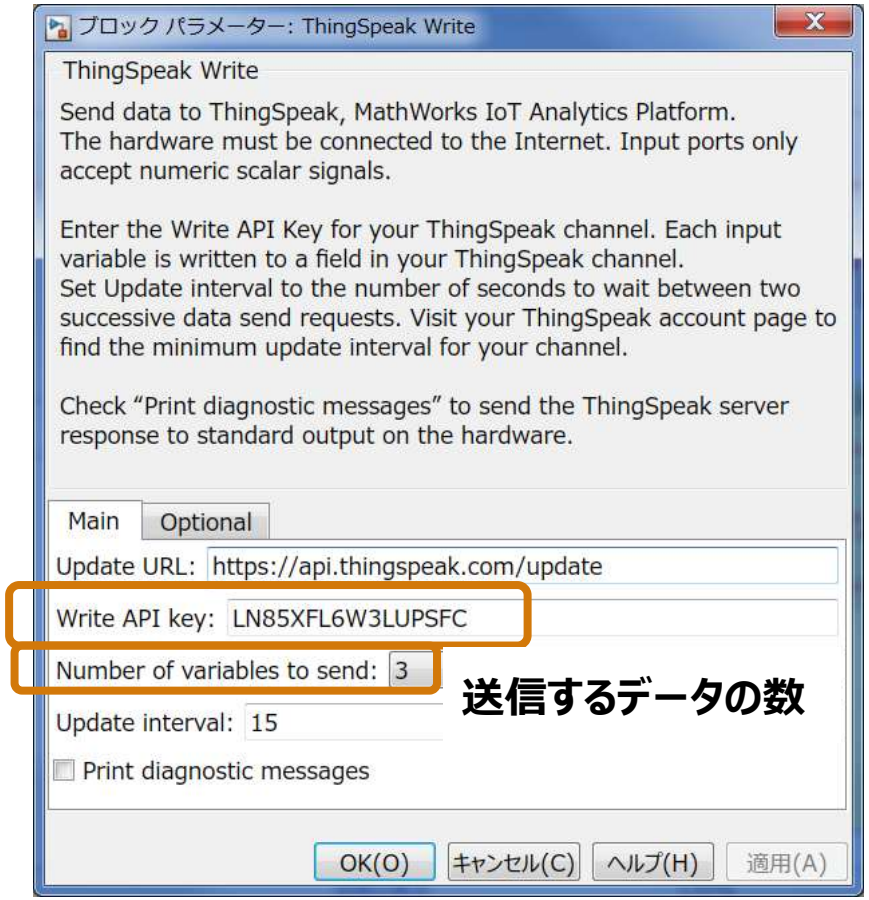
I2C Master Read ブロック



スレーブのアドレス

ThingSpeak Readブロック

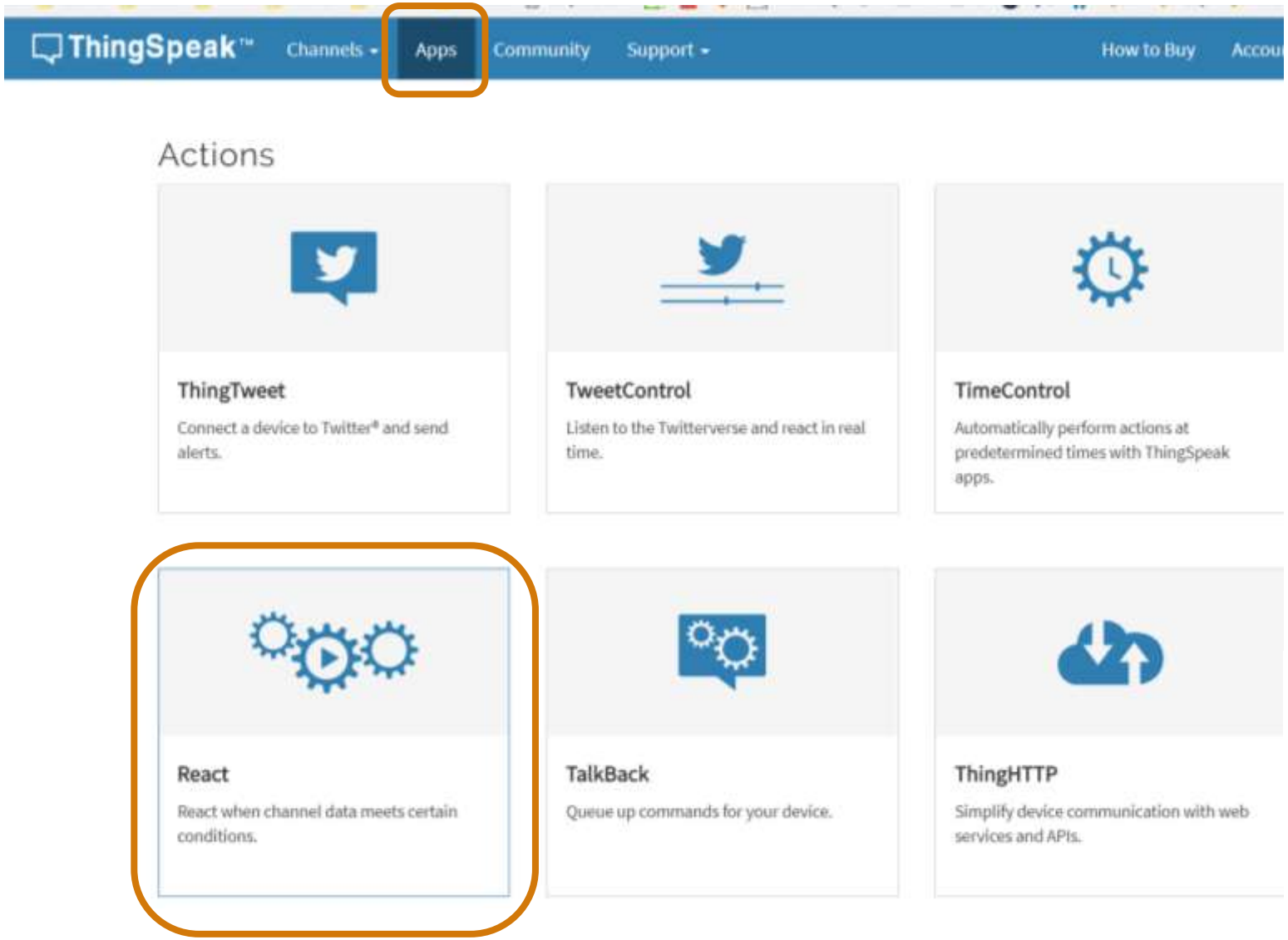




書き込み用API

送信するデータの数

例：温度センサーのモニタリング、アラートの配信
～Tweetの条件設定～



Test Frequency: On Data Insertion

Condition: If channel I2C temp (380633)

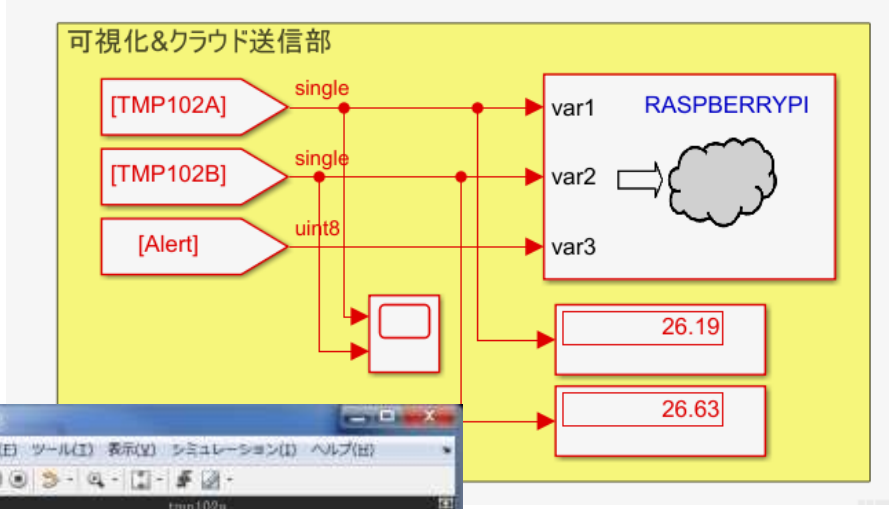
監視対象: field 3 (Alert)

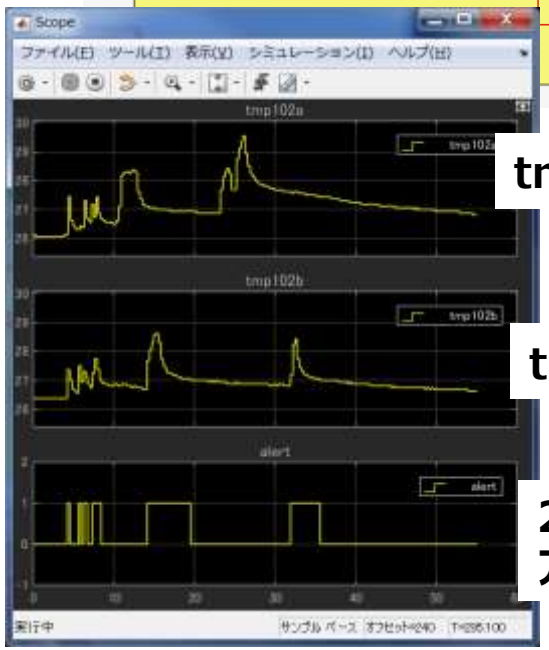
条件: is equal to 1

アクション（Tweetを送信）: ThingTweet

then tweet: 温度が27°を超えました

例：温度センサーのモニタリング、アラートの配信
～実行結果～



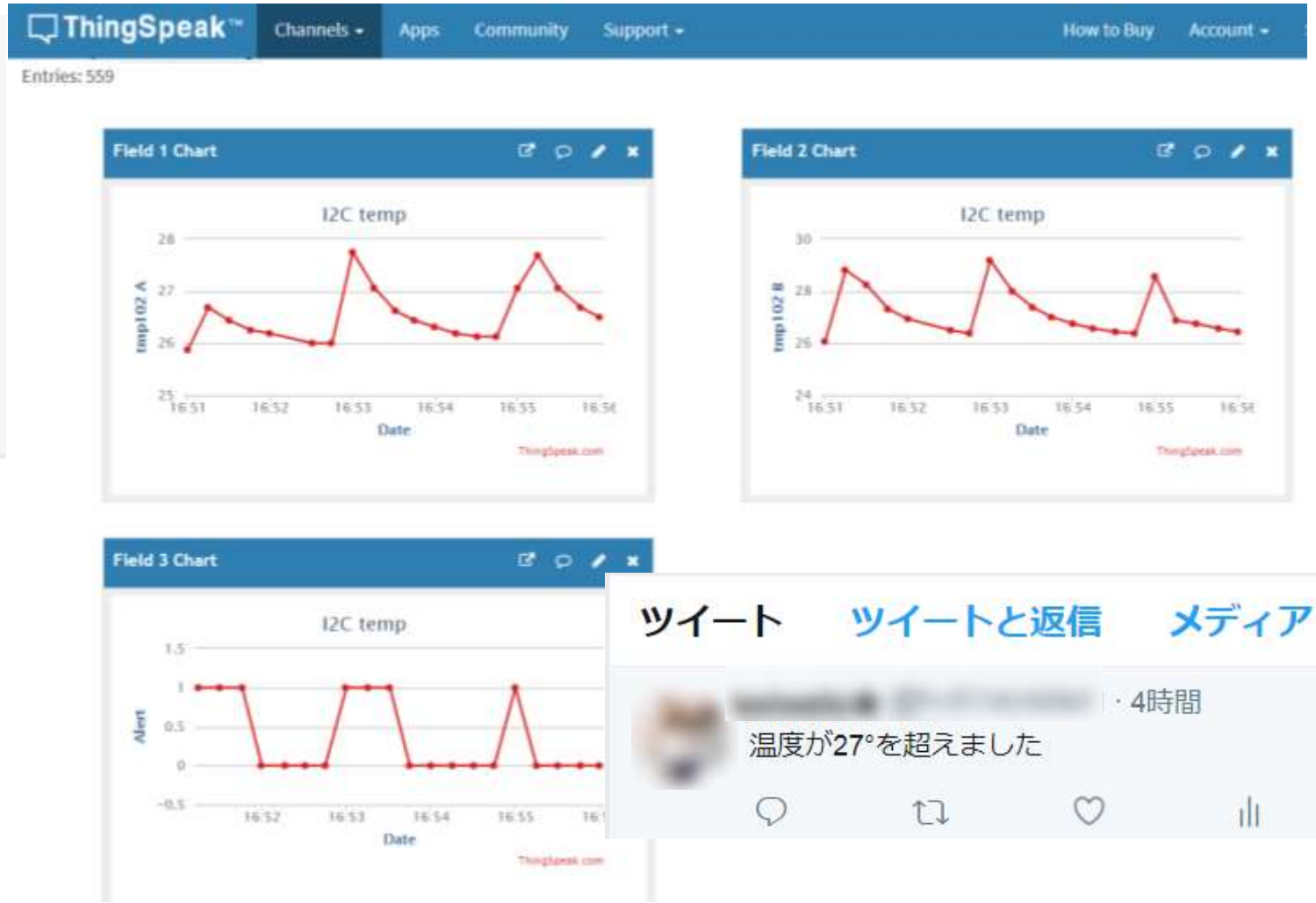


tmp102A

tmp102B

27°を超えた時
アラート

エクスターナルモードでの
実行結果



クラウド（Thingspeak）での可視化と
アラートの配信

Agenda

- 背景
- センサー信号の場合
- 音声信号の場合
- 画像の場合
- 関連情報
- まとめ

使用ツール

<table>
<tr><th>種類</th><th>センサー</th><th>音</th><th>画像</th></tr>
<tr><td>配布、
データ・
アグリゲート</td><td colspan="3">MATLAB Compiler
MATLAB Production Server
Thingspeak</td></tr>
<tr><td>機械学習</td><td colspan="3">Statistics and Machine Learning Toolbox
Neural Network Toolbox</td></tr>
<tr><td>前処理、
特徴量抽出</td><td colspan="2">Signal Processing Toolbox
DSP System Toolbox</td><td>Image Processing Toolbox
Computer Vision System Toolbox</td></tr>
<tr><td>データ収集</td><td>Data Acquisition Toolbox
Instrument Control Toolbox</td><td>Audio System Toolbox</td><td>Image Acquisition Toolbox</td></tr>
</table>

まとめ

- 背景
  - 事例の紹介
- センサー信号の場合
  - 信号処理関数で特徴量の試行錯誤を効率化
- 音声信号の場合
  - ユーザ事例、Audio System Toolboxのご紹介
- 画像の場合
  - 学習用データの拡張
- 関連情報
  - エッジデバイス実装、Thingspeakのご紹介

機械学習用途には信号処理ツールをお忘れなく