

現場ですぐ試すAI実装 ～データ取りや特性確認等の**泥臭い**ところを中心に～

MathWorks

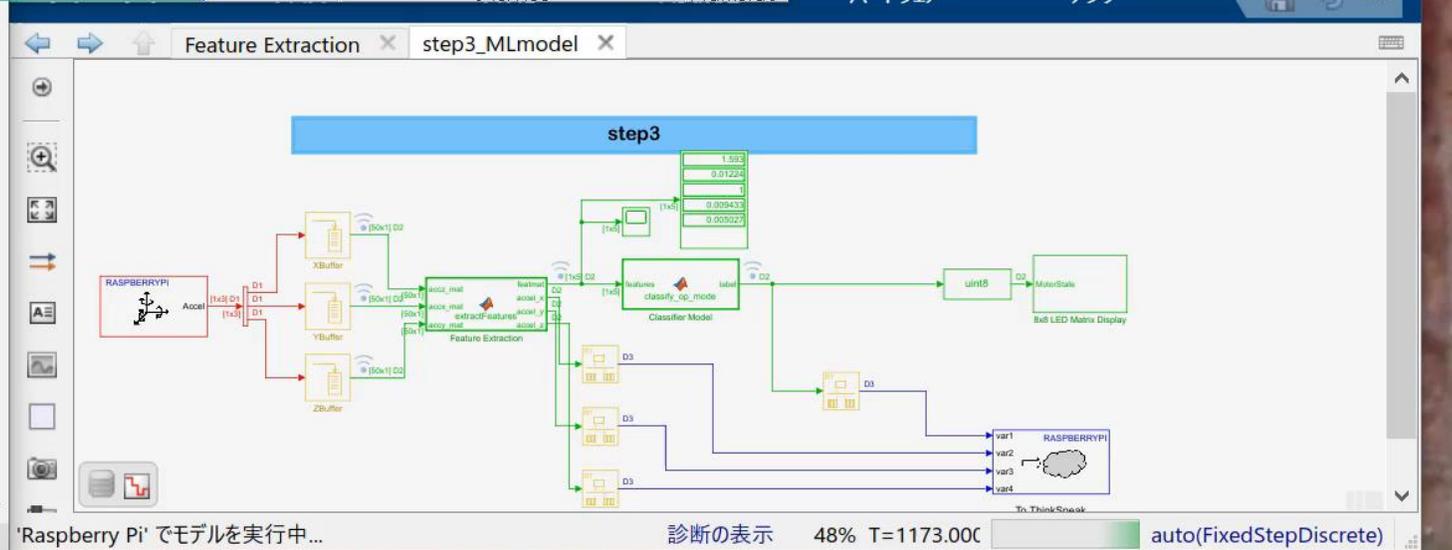
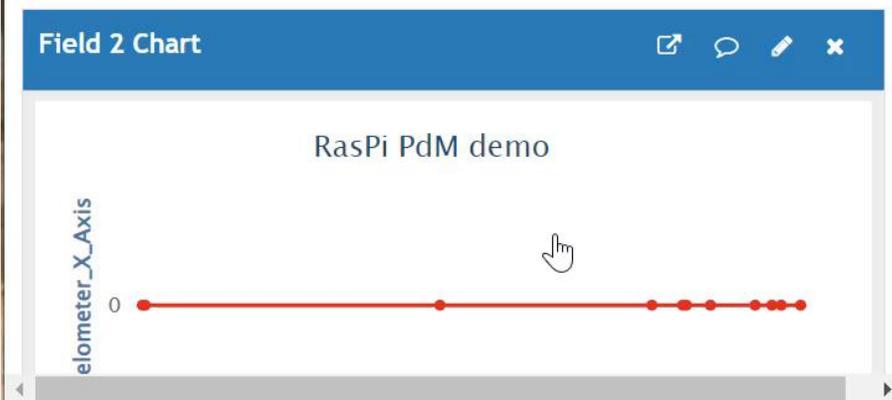
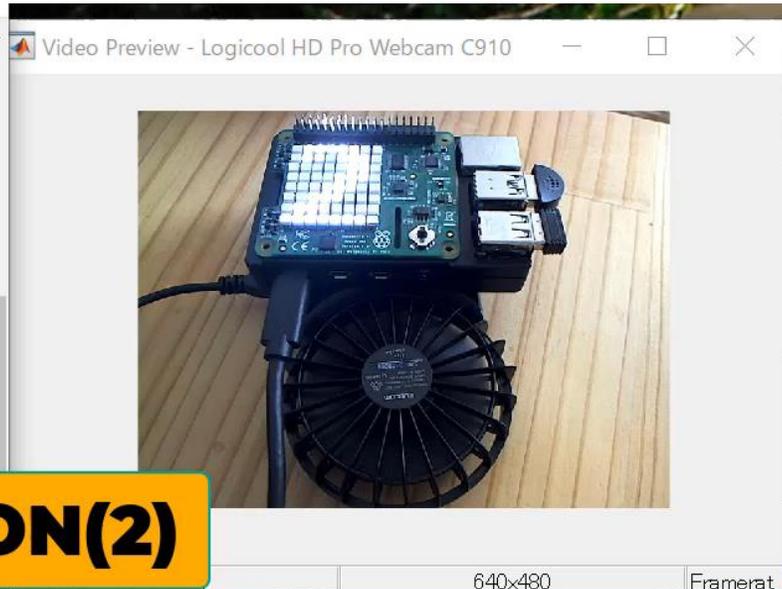
アプリケーションエンジニアリング部

竹本佳充

Demo



ON(2)



はじめに：目的とゴール

- **Raspberry Pi**を使ってファンの状態を予測し、結果を以下でモニターします

- **LED**表示
- **クラウド**による遠隔監視

- この例題をご覧頂くには…

- **信号処理**の基礎知識、**機械学習**の基礎知識は不要です
- （実運用時には）現場の**知見**が必要です

- この例題をご覧頂くことで…

- データの取得からシステム実装までの**一連流れ**を知ることができます
- 実体験に戻づく**泥臭ポイント**（DP）をご覧いただけます



AIシステム開発のワークフロー

1. データセットの作成

2. 前処理と変換

3. 予測モデルの開発

4. 実システムへの展開

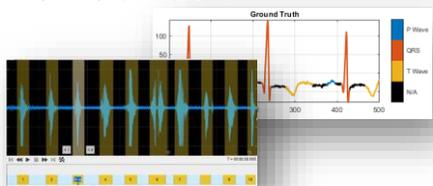
■ データソース



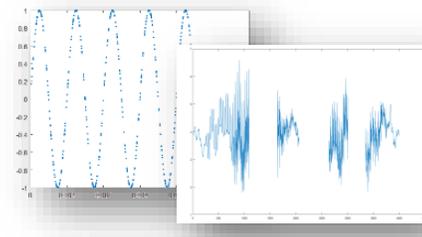
- シミュレーション
- オグメンテーション



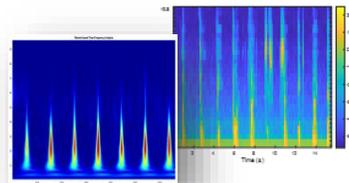
■ データラベリング



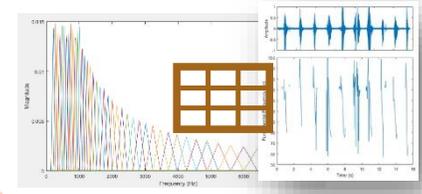
■ 前処理



■ 変換



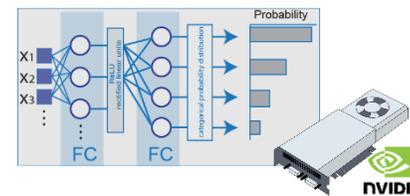
■ 特徴抽出



- リファレンスモデルの流用
- オリジナルモデルの作成



■ GPUによる高速な学習



■ ハイパーパラメータの解析とチューニング



■ デスクトップアプリ



■ エンタープライズシステム

Java
MATLAB
C/C++
Python

■ エッジ（組み込み）デバイス



Agenda

- Step0: 準備
 - ドキュメントの和訳
 - H/Wの準備
- Step1: 加速度センサーデータの収集
 - データ収集用モデルの準備
 - センサーデータの収集
- Step2: ON/OFFの判定
 - データの観察
 - モデルの実行
- Step3: ON/OFF/NGの判定
- まとめ

Step0

準備

デモのベースとなるドキュメント

<https://jp.mathworks.com/help/supportpkg/raspberrypi/ref/perform-predictive-maintenance-for-rotating-device-using-machine-learning-algorithm-on-raspberry-pi.html>

🔍 raspberry pi predictive maintenance

The screenshot shows the MathWorks Help Center interface. At the top, there is a search bar with the text 'ヘルプセンターを検索' and a 'サポート' button. Below the search bar, there are navigation tabs for 'Documentation', 'Examples', 'Functions', and 'Blocks'. The main content area displays the title 'Perform Predictive Maintenance for Rotating Device Using Machine Learning Algorithm on Raspberry Pi R2022a'. The text below the title describes the example, mentioning the Simulink Support Package for Raspberry Pi Hardware and the use of a machine learning algorithm for predictive maintenance. A diagram of the Simulink model is shown, featuring blocks like 'Acad', 'Buffer', 'Feature Extractor', 'Classify_3D_model', 'Classify Model', 'LED', and 'MAT LED Matrix Display'. Below the diagram, there are instructions on how to configure the 'X, Y, and Z Buffer (DSP System Toolbox) Block Parameters' dialog box, including setting the 'Output buffer size' to 50 and the 'Buffer overlap' to 20.

MathWorks®

Help Center

ヘルプセンターを検索

サポート 🔍

Documentation Examples Functions Blocks

Perform Predictive Maintenance for Rotating Device Using Machine Learning Algorithm on Raspberry Pi R2022a

This example shows how to use the Simulink® Support Package for Raspberry Pi™ Hardware to perform predictive maintenance on a rotating device using a machine learning algorithm. This example for predictive maintenance is a piece of equipment so that you can detect and prevent equipment failure.

In this example, the operational modes are divided into four modes: Stop, Balanced, Imbalanced, and Fault. This example uses these operational modes on the rotating device. The Raspberry Pi hardware is used for sensing the vibrations using the

« Documentation Home

« Simulink Support Package for Raspberry Pi Hardware

« Modeling

« Simulink Support Package for Raspberry Pi Hardware

« Run on Target Hardware

Perform Predictive Maintenance for Rotating Device Using Machine Learning Algorithm on Raspberry Pi

ON THIS PAGE

« Documentation Home

« Simulink Support Package for Raspberry Pi Hardware

« Modeling

« Simulink Support Package for Raspberry Pi Hardware

« Run on Target Hardware

Perform Predictive Maintenance for Rotating Device Using Machine Learning Algorithm on Raspberry Pi

ON THIS PAGE

Prerequisites

Required Hardware

Configure these parameters in the X, Y, and Z Buffer (DSP System Toolbox) Block Parameters dialog box.

1. Set the **Output buffer size** parameter to 50.
2. Set the **Buffer overlap** parameter to 20.

ドキュメントの和訳

The screenshot shows the MATLAB R2022a environment. The main window displays a document with the following content:

Raspberry Piを用いた機械学習アルゴリズムによる 回転機器の予知保全

この例では、Simulink® Support Package for Raspberry Pi™ Hardwareを使用して、機械学習アルゴリズムを用いて回転機器の健全性を予測・監視する方法を示しています。この例は、あらゆる回転装置や機器の予知保全に使用することができ、故障する前に修理することができます。

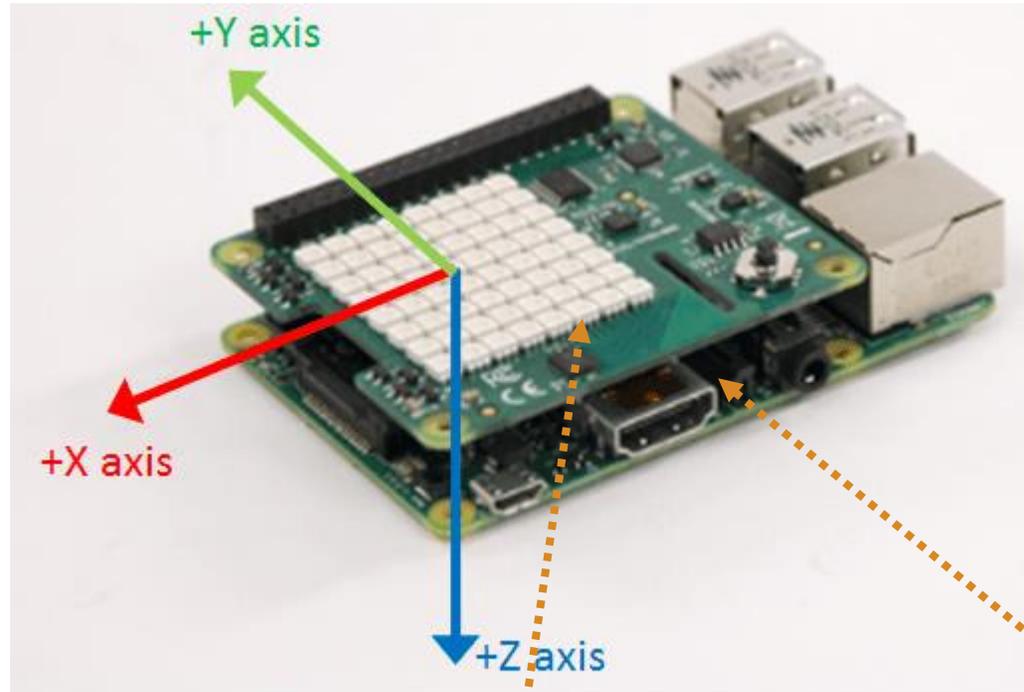
この例では、回転機器の動作状態を4つのモードに分けています。Stop、Block、Normal、Rotor Imbalancedの4つのモードを8×8のRGB LEDディスプレイとSense HATのLSM9DS1センサーを使用して監視しています。回転機器の振動データから特徴を抽出することができ

必要なハードウェア

- Raspberry Pi ボード
- なにかしらの動く部品。この例では回転ファンを使用。
- Sense HAT シールド
- 接続用のケーブル
- デバイス正常動作を阻害するなにかしらのもの。この例ではボール紙を使用
- 接着剤、もしくはテープ

At the bottom of the screenshot, there is a blue callout box with the text: **DP1: 便利なツールを知る**

Raspberry Piと周辺の備品



- 今回使用するもの
 - Raspberry Pi4 Model B/8GB
 - Micro SDカード
 - SenseHAT
 - なにかしら振動するもの（小型FAN等）
 - USBケーブル（給電用）
 - モニター、HDMIケーブル（SSHでも可）
 - Windows PC

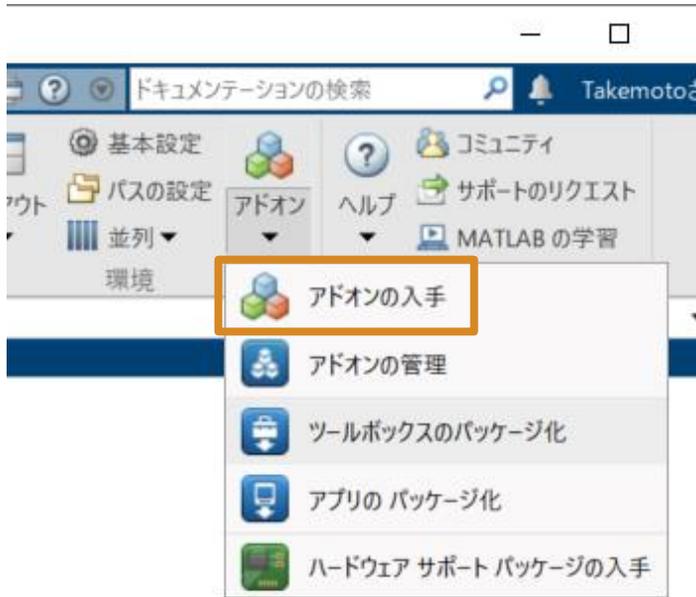
- SDカードはけちらない
- モニターはあればbetter

- SenseHAT
 - 加速度センサー
 - LEDマトリクス

- Raspberry Pi 本体

DP2: 備品の選定は意外と重要

サポートパッケージのインストール



タイプでフィルター		ベンダーでフィルタ	
ツールボックスと製品	24	Adimec	1
アプリ	3	Android	2
Simulink モデル	45	Arduino	29
✓ ハードウェア サポートパッケージ	319	ARM	15
オプション機能	3	National Instruments	13
関数		✓ Raspberry Pi	9

①条件で絞る

ハードウェア サポート パッケージ (9)

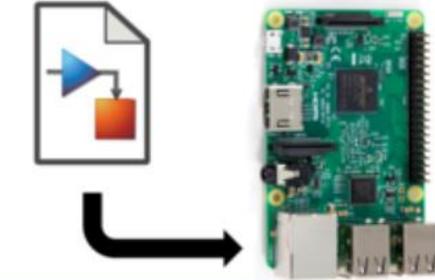


インストール済み

MATLAB Support Package for Raspberry Pi Hardware

Acquire sensor and image data from your Raspberry Pi.

ダウンロード 62.1K 件



インストール済み

Simulink Support Package for Raspberry Pi Hardware

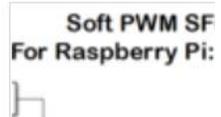
Run models on Raspberry Pi.

ダウンロード 62.5K 件



②Simulink Support Package …を選択

③あとは指示通りに…
(WiFiの使用がお勧め)



Step1

データの収集

データ収集用モデルの作成: movie

MATLAB R2022a

ドキュメンテーションの検索 Takemotoさん

ホーム プロット アプリ

新規 スクリプト 新規 ライブスクリプト 新規 開く ファイルの検索 比較

データのインポート データのクリーニング ワークスペースの保存 ワークスペースのクリア

お気に入り コードの解析 実行および時間の計測 コマンドのクリア

Simulink レイアウト 基本設定 パスの設定 並列 環境

ヘルプ コミュニティ サポートのリクエスト MATLAB の学習 リソース

E:\mwork2\mydemo_app\PdM\RasPiFan\modified22av2

現在のフォルダー

名前	更新日	サイズ	タイプ
step2_rawdata.mat	2022/04/26 18:24	17 KB	MAT ファイル
data_prep.mlx	2022/05/12 13:31	4 KB	ライブ スクリプト
step1_daq.slx	2022/04/20 9:30	30 KB	Simulink モデル

step2_rawdata.mat (MAT ファイル)

ワークスペース

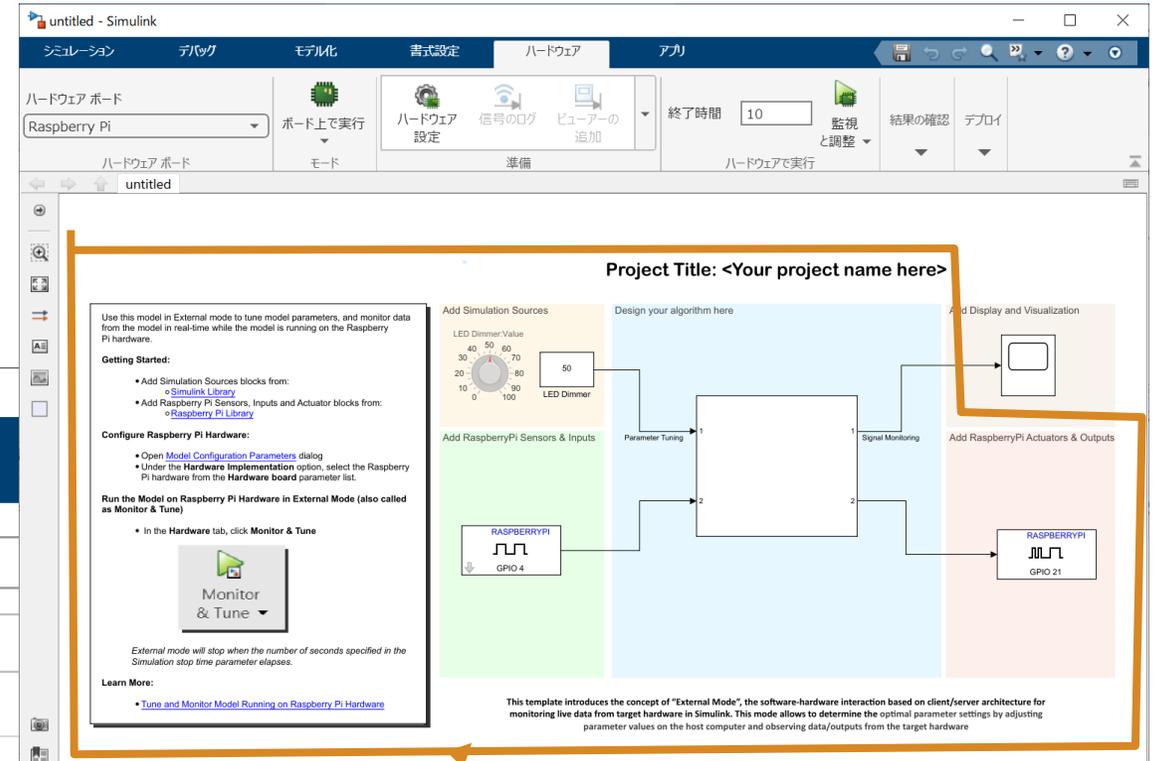
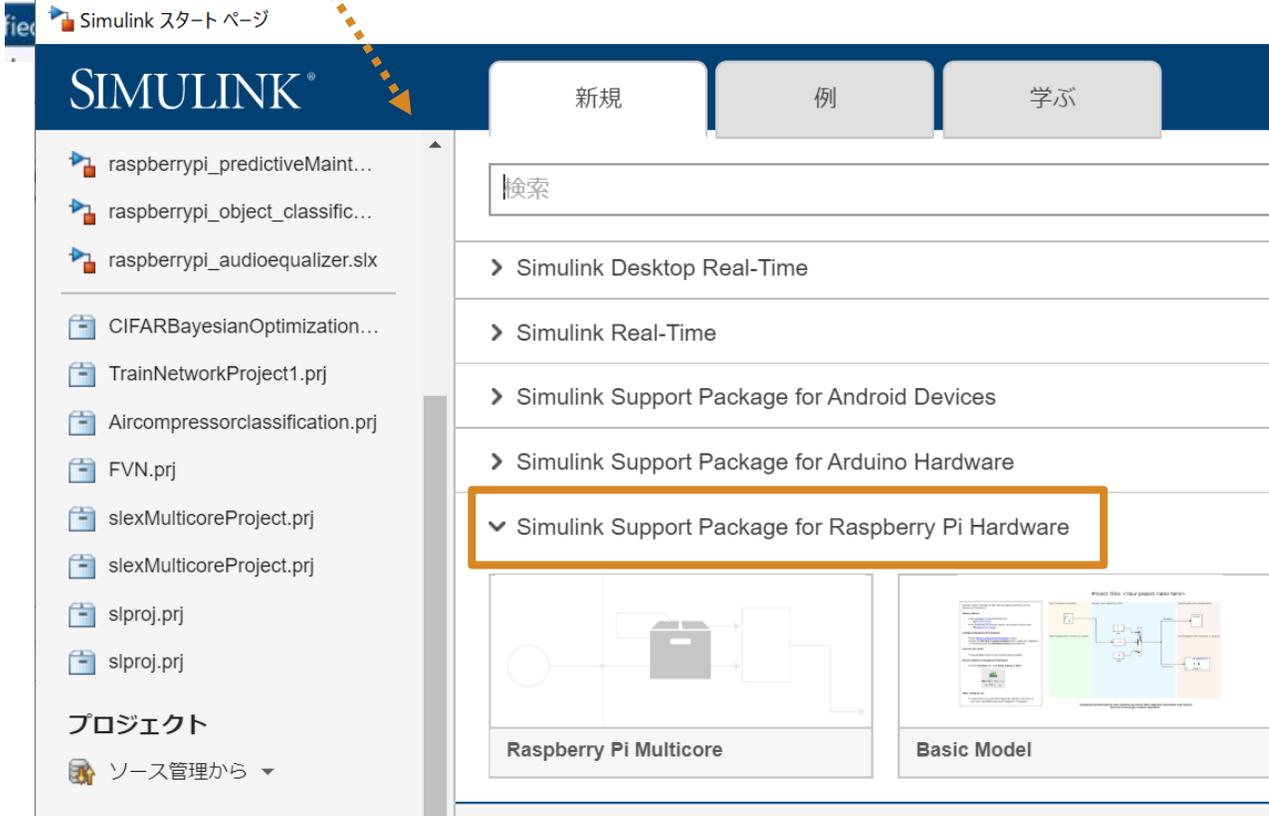
名前	値	サイズ

コマンド ウィンドウ

```
fx >>
```

データ収集用モデルの作成

① Simulinkの起動



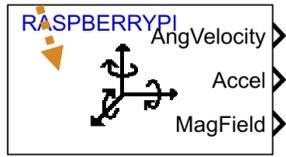
② Raspberry Pi実装用
モデルテンプレートを開く
(今回scopeブロック以外は
使用しないので削除)



データ収集用モデルの作成(cont'd)

The image shows two Simulink library browser windows and a Simulink model window. The left window shows the 'Simulink Support Package for Raspberry Pi Hardware/Sense HAT' category, with the 'LSM9DS1 IMU Sensor' block highlighted. The right window shows the 'Simulink/Sinks' category, with 'Display', 'Scope', and 'To Workspace' blocks highlighted. The Simulink model window shows the 'untitled' model with the 'LSM9DS1 IMU Sensor' block and three output blocks: 'display', 'Scope', and 'simout'. Dotted orange arrows connect the highlighted blocks in the library browser to their respective blocks in the model and to the explanatory text boxes.

SenseHAT
xyz加速度センサー



displayブロック
(センサーデータ値を表示)

Scopeブロック
(センサーデータをプロット)

To Workspaceブロック
(センサーデータをファイルに格納)

データ収集用モデルの作成(cont'd)

ブロックパラメーター: LSM9DS1 IMU Sensor

LSM9DS1 9-DoF IMU Sensor (mask) (link)

Measure linear acceleration, angular rate and magnetic field along X, Y and Z axis.

The block outputs acceleration as a [1X3] vector of double values in g (9.8 m/s²), angular rate as a [1X3] vector of double values in degrees per second (dps) and magnetic field as a [1X3] vector of double values in micro Tesla (μT).

View Axis Information

パラメーター

Board: Pi 4 Model B

Active sensors: Accelerometer

Advanced sensors: Accelerometer, Gyroscope and Magnetometer

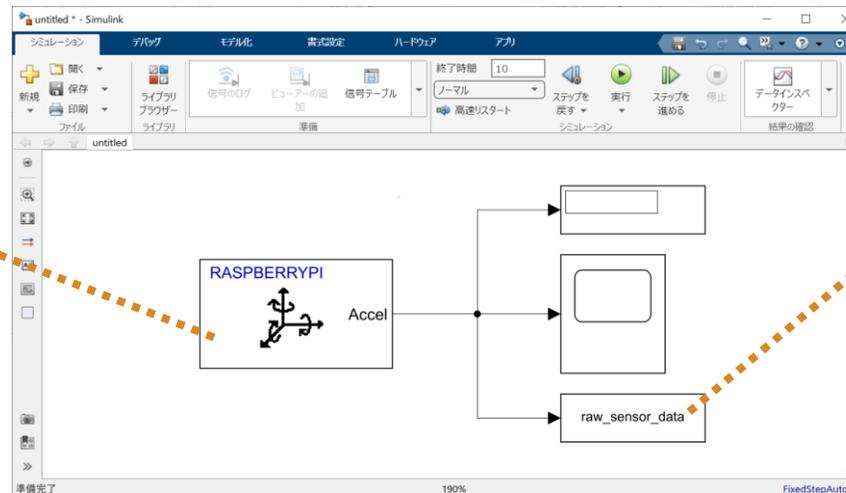
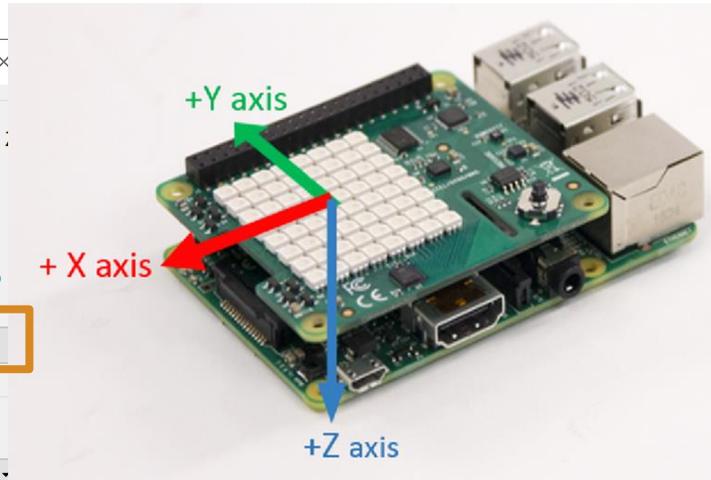
Accelerometer: Accelerometer and Magnetometer

Gyroscope: Gyroscope and Magnetometer

Magnetometer: Accelerometer, Gyroscope and Magnetometer

SampleTime: 0.02

OK(C) キャンセル(C) ヘルプ(H) 適用(A)



ブロックパラメーター: To Workspace

To Workspace

ワークスペース内で指定した時系列、配列、または構造体に入力を書き込みます。メモリーベースのシミュレーションの場合、データは MATLAB のベース ワークスペースに書き込まれます。データはシミュレーションが中止または一時停止するまで利用できません。

バス信号のログを作成するには、保存形式に "[時系列]" を使用します。

パラメーター

変数名: raw_sensor_data

データ点の制限: inf

間引き: 1

保存形式: 配列

2次元信号の保存形式: 2次元配列 (最初の次元と連結)

fi オブジェクトとして固定小数点データのログを記録する

サンプル時間 (継承 = -1):

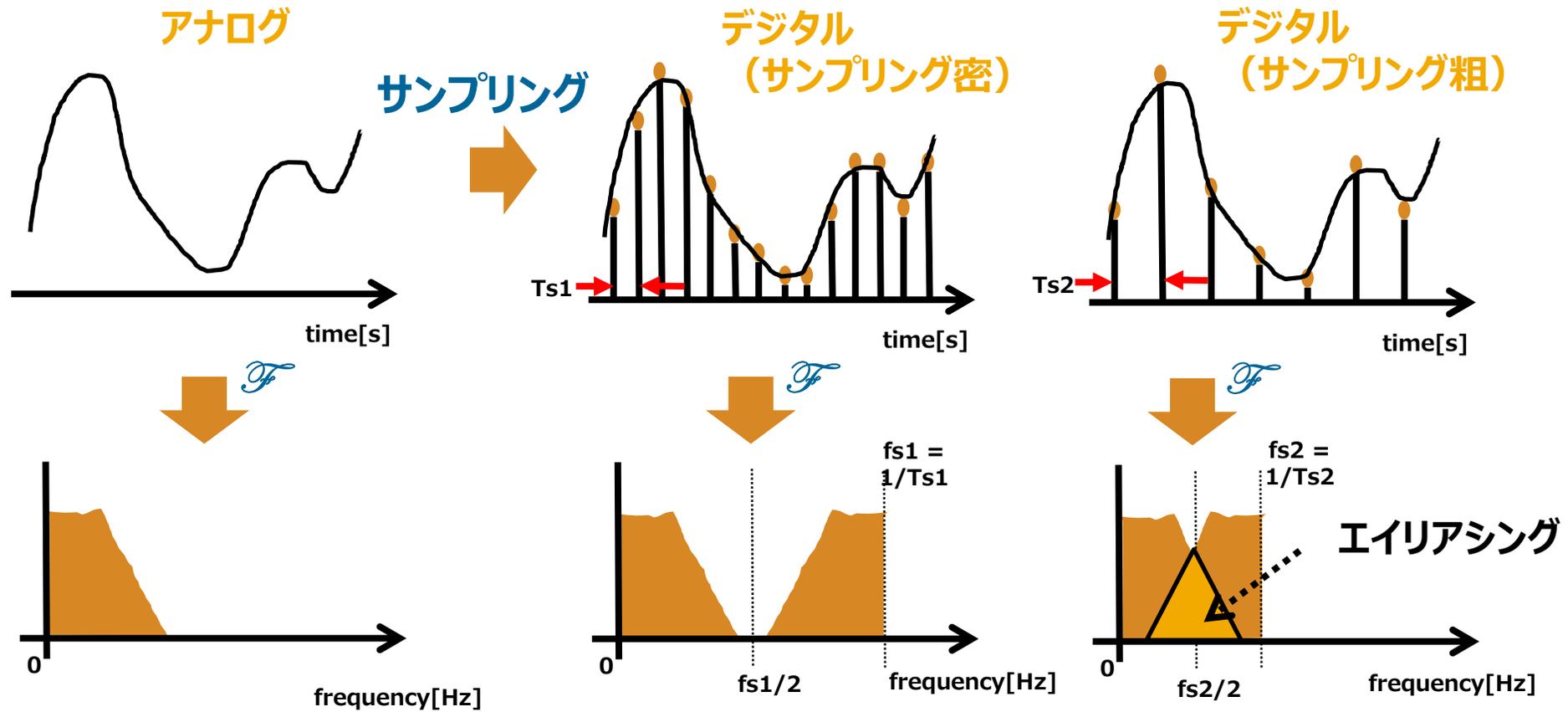
OK(O) キャンセル(C) ヘルプ(H) 適用(A)

加速度センサー設定

To Workspace設定

DP3: サンプル時間は適切に

補足：サンプリング定理

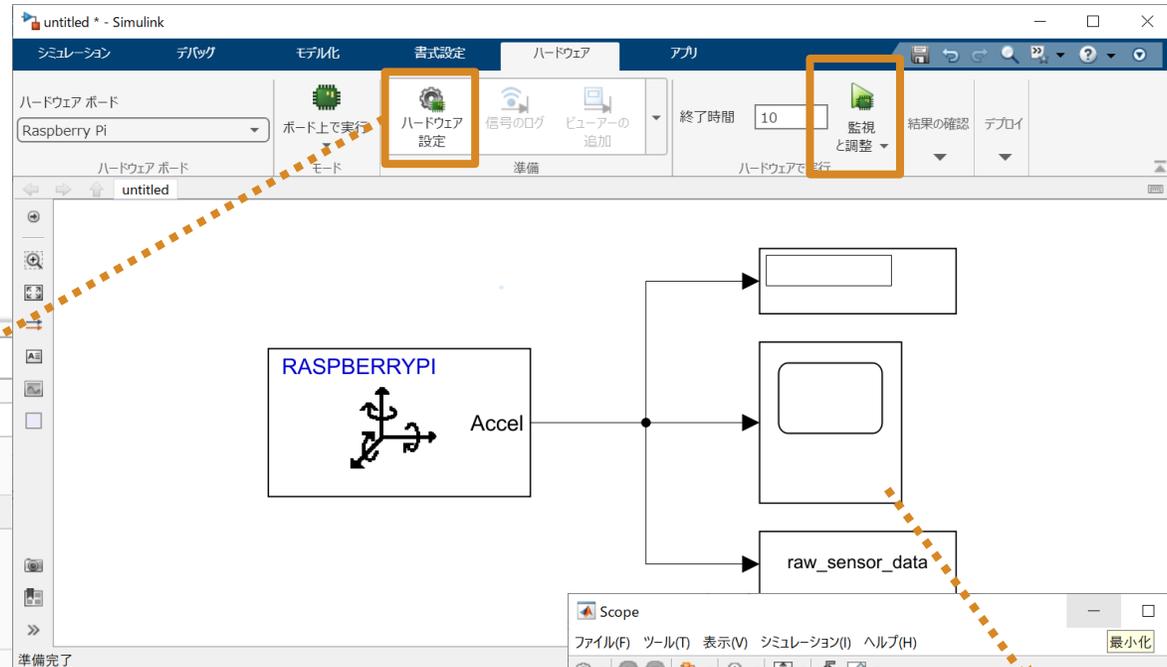


- T_s : サンプル時間 (s)
- f_s ($1/T_s$) サンプル周波数(Hz)
- $f_s/2$: ナイキスト周波数 (Hz)

e.g.) サンプル時間が0.02[s]の場合

- サンプル周波数: $1/0.02 = 50$ [Hz]
 - ナイキスト周波数: 25[Hz]
- ⇒25[Hz]以下の成分を観測

データ収集用モデルの作成(cont'd)



コンフィギュレーション パラメーター: untitled/Configuration (アクティブ)

検索

ハードウェアボード: **Raspberry Pi**

コード生成のシステムターゲット ファイル: [ert.tlc](#)

デバイスベンダー: ARM Compatible

デバイスの詳細

ハードウェアボード設定

オペレーティングシステム/スケジューラ

Target hardware resources

Groups

Board Parameters: **Device Address: 192.168.1.22**

Build options

SPI

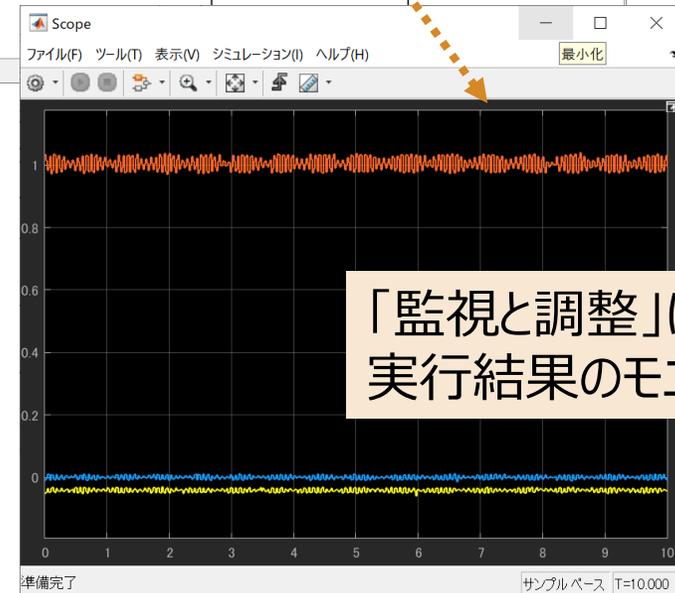
CAN

MQTT

Username: pi

Password: raspberry

Raspberry Pi設定



「監視と調整」による
実行結果のモニタリング

補足：MATLABで加速度センサーのデータをキャプチャする方法

ヘルプ

検索結果 - reaAcceler...
readAcceleration - Ma...

Help Center

R2022a のドキュメンテーションを検索

Documentation Examples Functions Apps

評価版 製品の更新

readAcceleration

Read acceleration measured by the Accelerometer along x, y, and z axes

R2022a
collapse all in page

Syntax

```
Acceleration = readAcceleration(mysh)
[Acceleration,Ts]= readAcceleration(mysh)
accelerationRaw = readAcceleration(mysh,'raw')
[accelerationRaw,Ts]= readAcceleration(mysh,'raw')
```

Description

`Acceleration = readAcceleration(mysh)` returns the acceleration measured by the Accelerometer along the x, y, and the z axis. [example](#)
The unit of measurement for the Accelerometer data is meter per second square (m/s²)

`[Acceleration,Ts]= readAcceleration(mysh)` returns the acceleration measured by the Accelerometer along the x, y, and the z axis with timestamp. [example](#)

`accelerationRaw = readAcceleration(mysh,'raw')` returns the acceleration measured by the Accelerometer along the x, y, and the z axis. The 'raw' argument is to specify that the output should be a uncalibrated raw data. [example](#)

`[accelerationRaw,Ts]= readAcceleration(mysh,'raw')` returns the acceleration measured by the Accelerometer along the x, y, and the z axis with timestamp. The 'raw' argument is to specify that the output should be a uncalibrated raw data. [example](#)
The full scale range for the Accelerometer is +/- 2g
Sense HAT IMU sensor axis information is shown below.

MATLAB Support Package
for Raspberry Pi Hardware

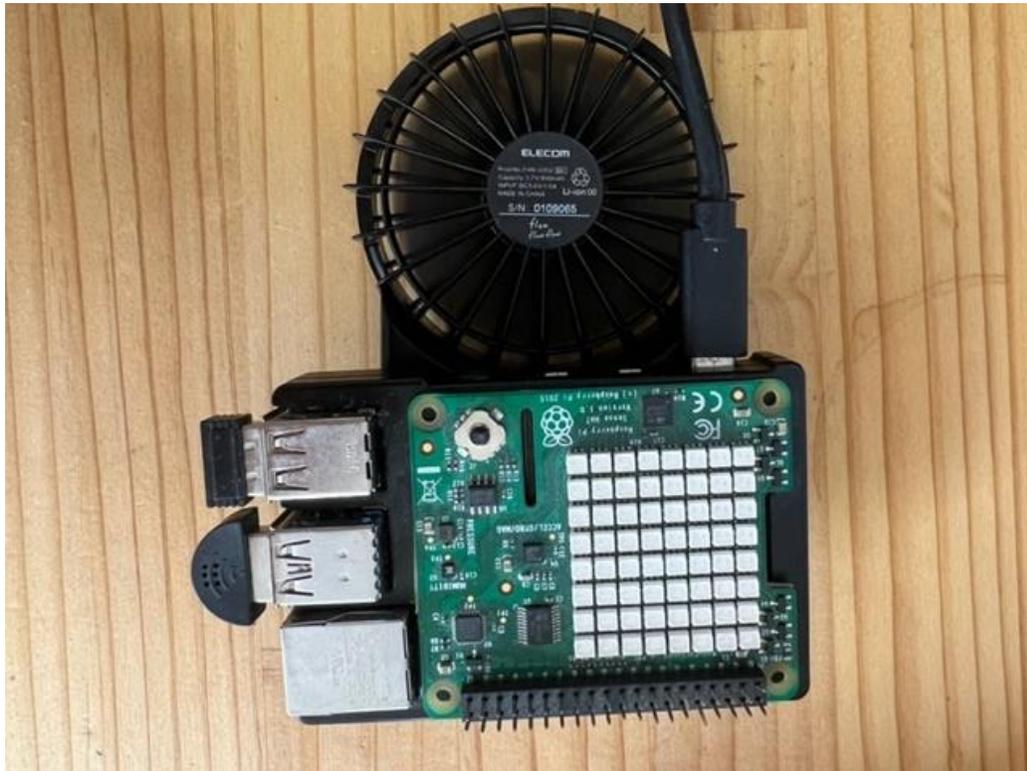
```
>>r = raspi; % Raspberry Pi オブジェクトの定義
>>mysh = sensehat(r); % sensehatオブジェクトの定義
>>[x,t] = readAcceleration(mysh); % x: 3軸加速度、t: タイムスタンプ
```

Step2 ON/OFFの判別

ON/OFFを判別する

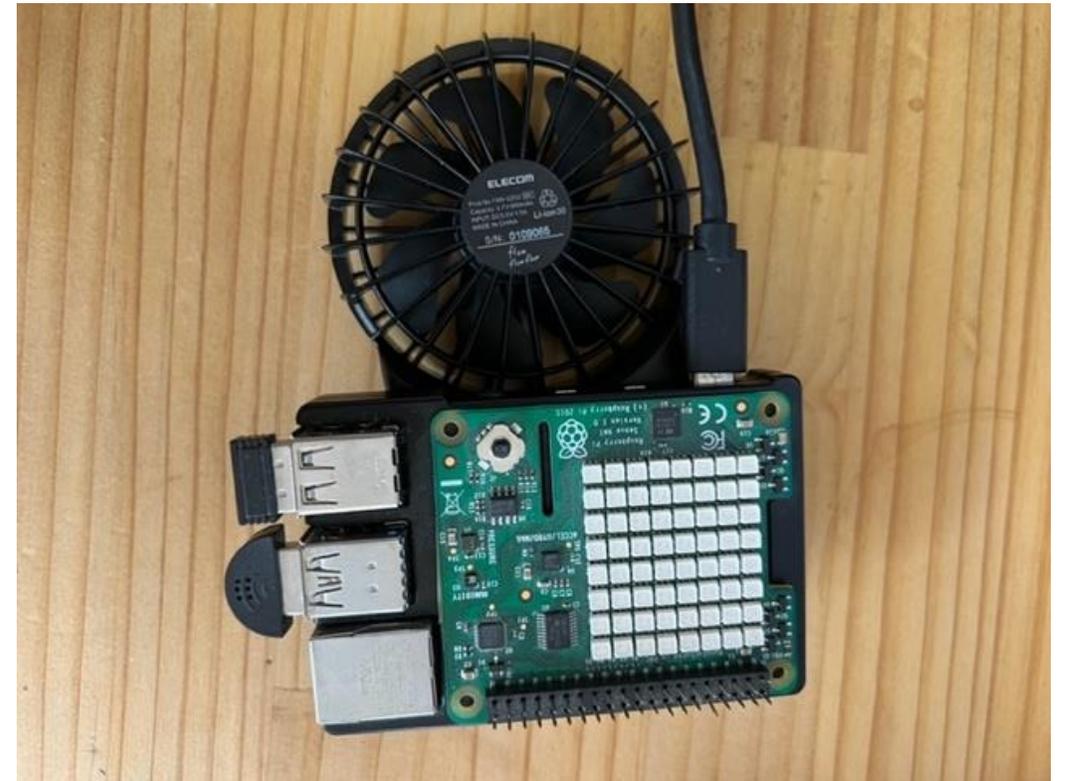
DP4: 題材はシンプルなものから

ON



回転状態
(正常)

OFF



停止状態

信号アナライザー (movie)

MATLAB R2022a

ホーム プロット アプリ

ドキュメンテーションの検索 Takemotoさん

アプリの設計 さらにアプリを取得 アプリのインストール アプリのパッケージ化

曲線フィッター 信号アナライザー Audio Test Bench フィルター デザイナー 信号ラベラー 信号多重解像度アナライザー 診断特徴デザイナー Sensor Array Analyzer 分類学習器 Raspberry Pi リソース モニター ディープ ネットワーク デザイナー

ファイル アプリ

E: \mwork2 \mydemo_app \PdM \RasPiFan \modified22av1

現在のフォルダー

名前	更新...	サイズ	タ...
slprj	2022...		フォル...
step1_daq_ert_rtw	2022...		フォル...
step2_simplemodel_ert_r.2022...			フォル...
step3_MLmodel_ert_rtw	2022...		フォル...
feature_for_simulink_test..2022...	2022...	2 KB	エディ...
step1_daq.elf	2022...	48 KB	ELF ...
step2_simplemodel.elf	2022...	406 ...	ELF ...
step3_MLmodel.elf	2022...	498 ...	ELF ...
feature_for_simulink_test..2022...	2022...	3 KB	関数
feature_for_simulink_test..2022...	2022...	2 KB	関数
feature_for_simulink_test..2022...	2022...	2 KB	関数
raspberrypi_pdm_extract..2022...	2022...	4 KB	関数
ClassificationLearnerSess..2022...	2022...	1.66 ...	MAT ...
DFD_App_SessionData....	2022...	281 ...	MAT ...
sigana_ONOFF.mat	2022...	11.1...	MAT ...
sigana_ONOFFNG.mat	2022...	11.6...	MAT ...
step1_rawdata.mat	2022...	11 KB	MAT ...
step2_rawdata.mat	2022...	17 KB	MAT ...
step2_rawdata_old.mat	2022...	18 KB	MAT ...

step1_rawdata.mat (MAT ファイル)

ワークスペース

名前	値
raw_sensor_data_off	1001x3 double
raw_sensor_data_on	1001x3 double

コマンド ウィンドウ

```
fx >>
```

データの観察：信号アナライザーの起動

① アプリタブ

② 「信号アナライザー」を選択

現在のフォルダー

名前	更新日	サイズ
modified	2022/02/25 9:...	
modified21bv0	2022/02/25 9:...	
modified21bv1	2022/02/25 9:...	
modified22av0	2022/02/25 9:...	
modified22av1	2022/04/18 10:...	
original22a	2022/02/25 9:...	
daq.mp4	2022/04/19 8:...	5.98 MB
sigana.mp4	2022/05/09 16:...	13.76 MB

③ 信号アナライザ起動

データの観察：信号の読み込みとサンプルレートの設定

信号アナライザー - 無題*

① drag & drop

② 時間軸の単位設定

③ サンプルレートで設定

名前	サイズ	クラス
raw_sensor_...	1001×3	double
raw_sensor_...	1001×3	double

時間値

選択した信号のサンプルレート、サンプル時間または時間値ベクトル

時間指定:

OK

時間値

選択した信号のサンプルレート、サンプル時間または時間値ベクトルを編集します

時間指定:

サンプルレート: Hz

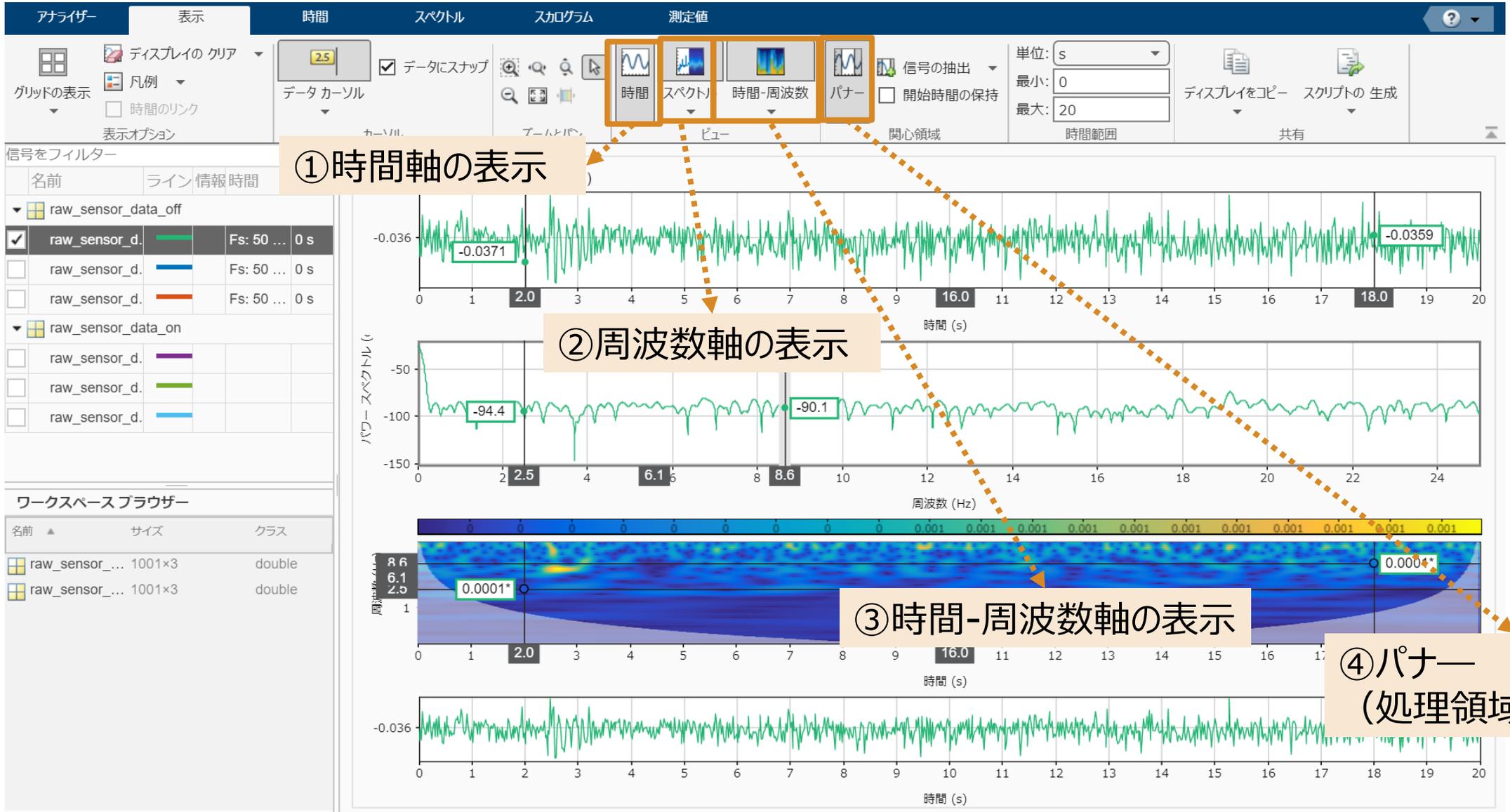
開始時間: s

OK キャンセル

データの観察：特性の可視化

DP5: 着目すべき"軸"を見極める

信号アナライザ - 無題*



データの観察：特性の比較、統計量の計算

信号アナライザ - 無題*

アナライザ 表示 時間 スペクトル 測定値

単位: s
 最小: 1.498896247e-
 最大: 1.984547461e-
 時間範囲

信号をフィルター

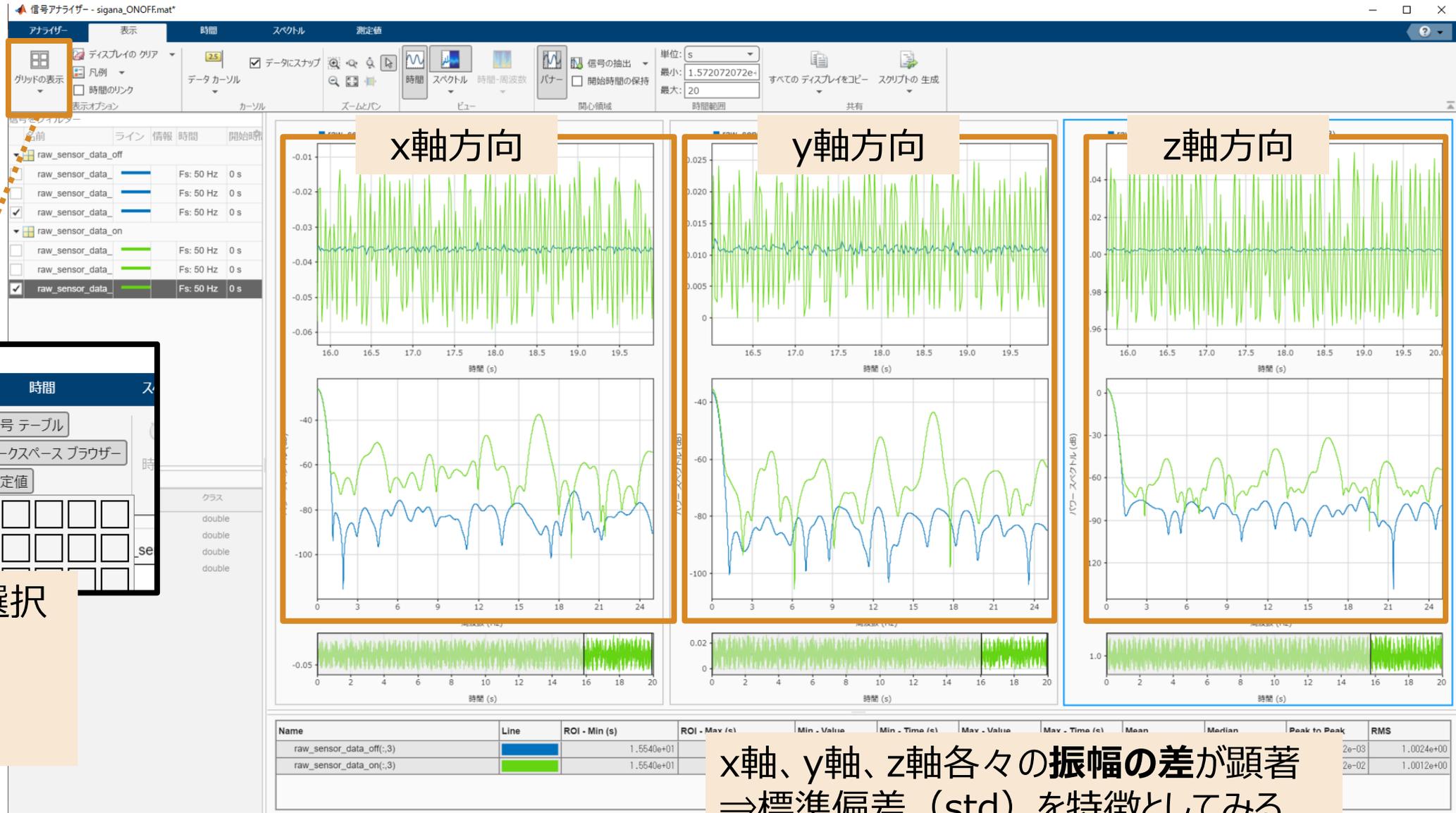
名前	ライン	情報	時間	開始
raw_sensor_data_off				
<input checked="" type="checkbox"/> raw_sensor	緑		Fs: 5...	0 s
<input type="checkbox"/> raw_sensor	青		Fs: 5...	0 s
<input type="checkbox"/> raw_sensor	赤		Fs: 5...	0 s
raw_sensor_data_on				
<input checked="" type="checkbox"/> raw_sensor	紫		Fs: 5...	0 s
<input type="checkbox"/> raw_sensor	緑		Fs: 5...	0 s
<input type="checkbox"/> raw_sensor	青		Fs: 5...	0 s

① 複数信号の
 選択で特性を比較
 ■ 緑 : OFF
 ■ 紫 : ON

② 各種統計量 (Max, Min, Peak to Peak, RMS等)

Name	Line	ROI - Min ...	ROI - Max...	Min - Value	Min - Tim...	Max - Value	Max - Tim...	Mean	Median	Peak to P...	RMS
raw_sensor_...	緑	1.4980e+01	1.9860e+01	-3.7781e-02	1.6440e+01	-3.4119e-02	1.5280e+01	-3.6228e-02	-3.6194e-02	3.6621e-03	3.6233e-02
raw_sensor_...	紫	1.4980e+01	1.9860e+01	-5.9021e-02	1.7960e+01	-8.6670e-03	1.7640e+01	-3.6322e-02	-3.6682e-02	5.0354e-02	3.9195e-02

データの観察 : ON vs OFF



信号アナライザー - 無題*

アナライザー 表示 時間

新規 開く 保存

ファイル

信号をフィルター

名前

信号 テーブル

ワークスペース ブラウザー

測定値

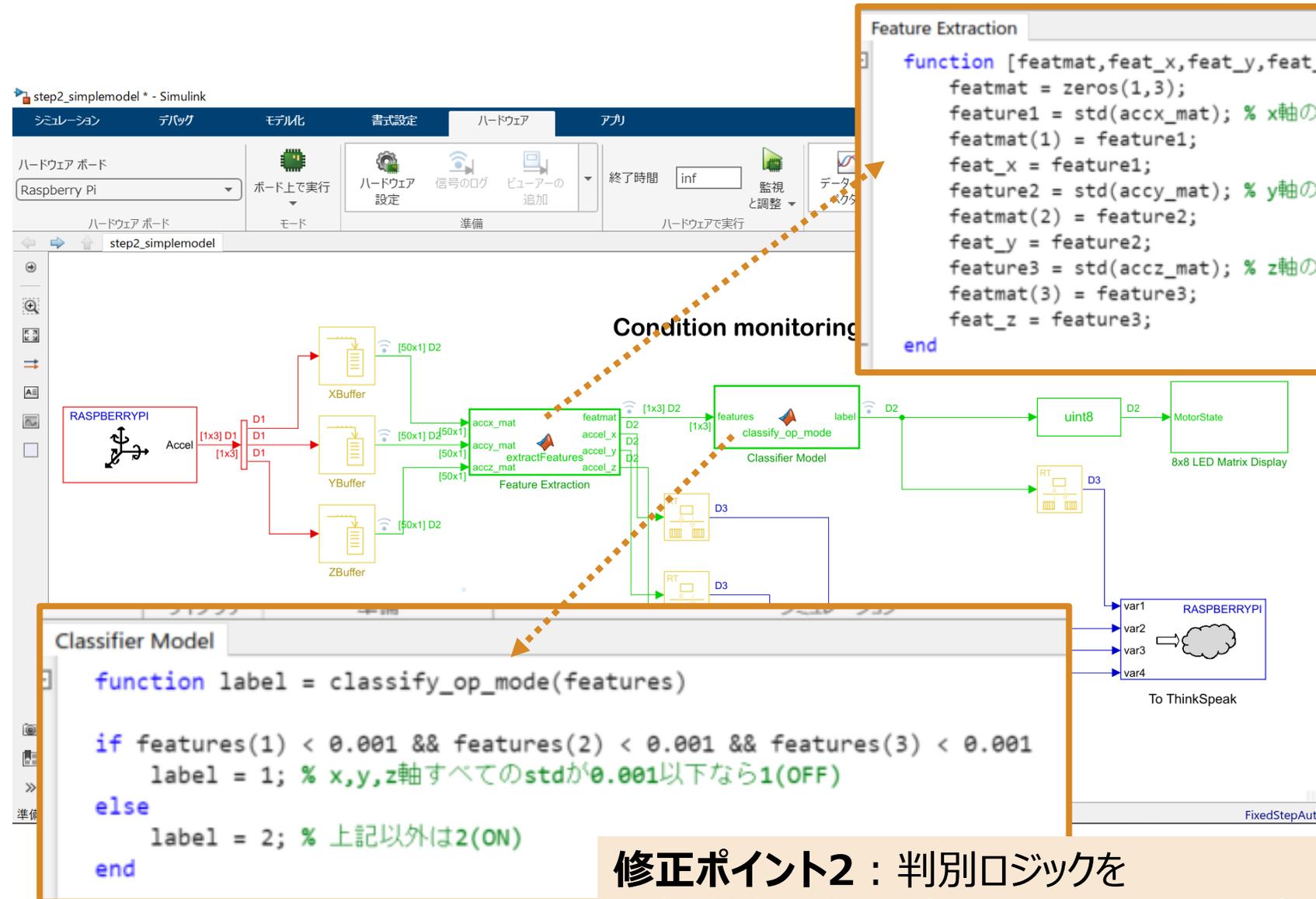
グリッドの表示

表示グリッドの選択

- 左 : x方向
- 中 : y方向
- 右 : z方向

x軸、y軸、z軸各々の**振幅の差**が顕著
⇒標準偏差 (std) を特徴としてみる

既存モデルをON/OFFの判別用モデルに修正 (修正は3ヶ所のみ)



```

Feature Extraction
function [featmat,feat_x,feat_y,feat_z] = extractFeatures(accx_mat,accy_mat,accz_mat)
    featmat = zeros(1,3);
    feature1 = std(accx_mat); % x軸のstd(標準偏差)
    featmat(1) = feature1;
    feat_x = feature1;
    feature2 = std(accy_mat); % y軸のstd
    featmat(2) = feature2;
    feat_y = feature2;
    feature3 = std(accz_mat); % z軸のstd
    featmat(3) = feature3;
    feat_z = feature3;
end
    
```

修正ポイント1: 特徴量をstd (標準偏差)に変更

DP6: あるものは活用する

```

Classifier Model
function label = classify_op_mode(features)

if features(1) < 0.001 && features(2) < 0.001 && features(3) < 0.001
    label = 1; % x,y,z軸すべてのstdが0.001以下なら1(OFF)
else
    label = 2; % 上記以外は2(ON)
end
    
```

修正ポイント2: 判別ロジックを閾値判定に変更 (OFF : 1 or ON : 2)

既存モデルをON/OFFの判別用モデルに修正 (ThingSpeak設定)



① MathWorks.com
トップページのmenuボタン

サイト内検索



ラーニング



MATLAB
Drive

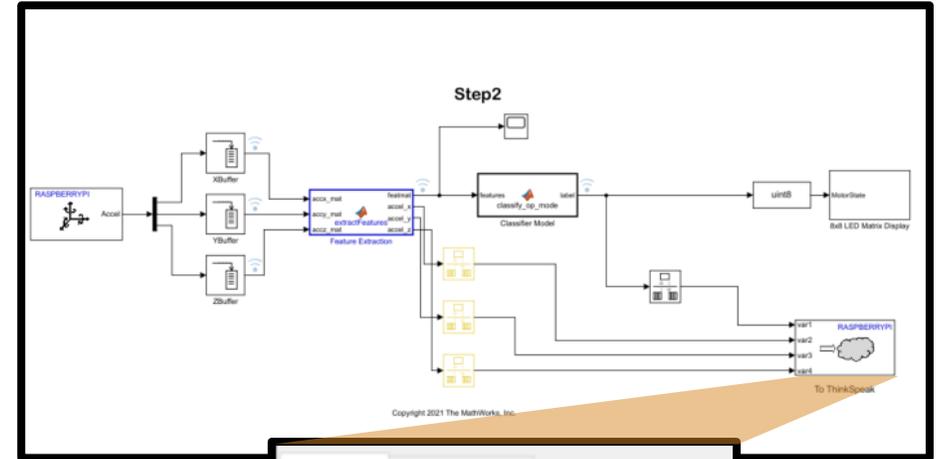


ThingSpeak



バグレ

② ThingSpeak
を選択



⑤ Write API keyを
ThingSpeak Writeブロックに
コピペ (修正ポイント3)

Main Optional

Write API key:

Number of variables to send: 4

Update interval: 0.1

Print diagnostic messages

analysis using MATLAB

③ アカウントを作成

Get Started For Free

Learn More

Private View Public View Channel Settings Sharing API Keys

Write API Key

Key

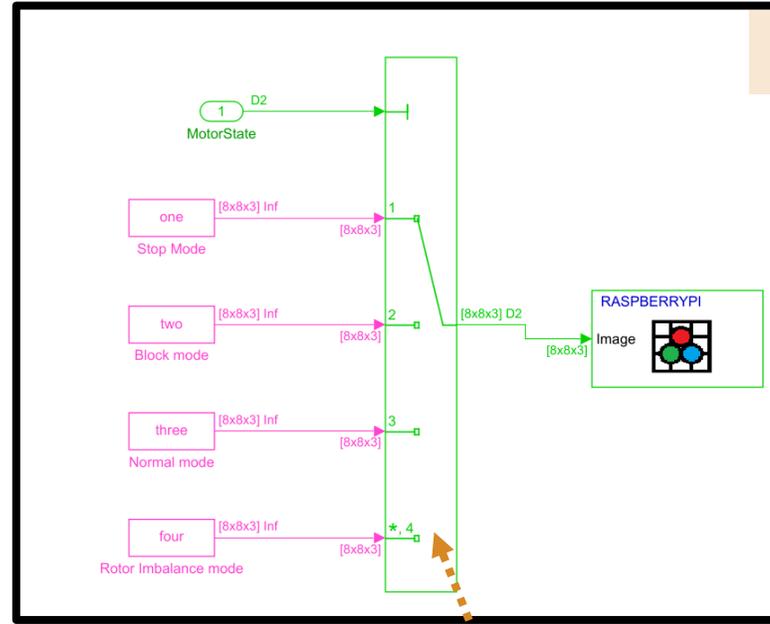
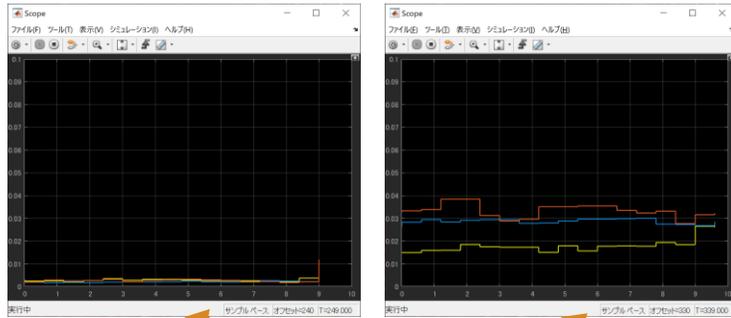
Help

API key
read da
generat

④ My Channelsを
作成

ON/OFFの判別用モデル

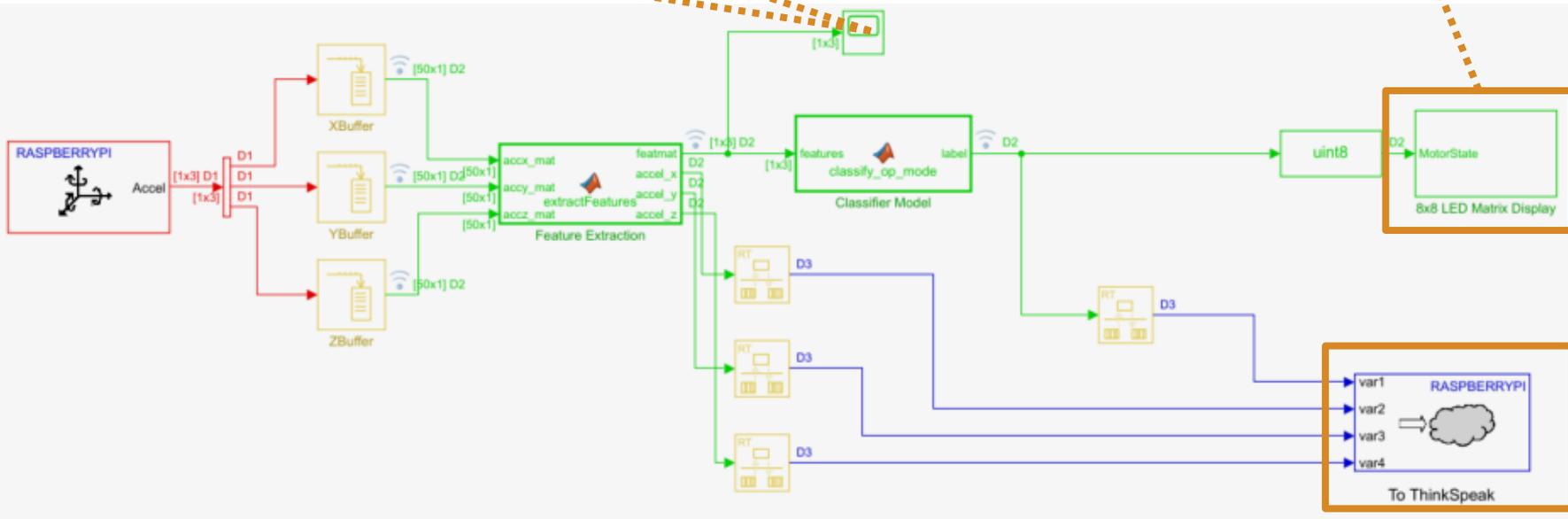
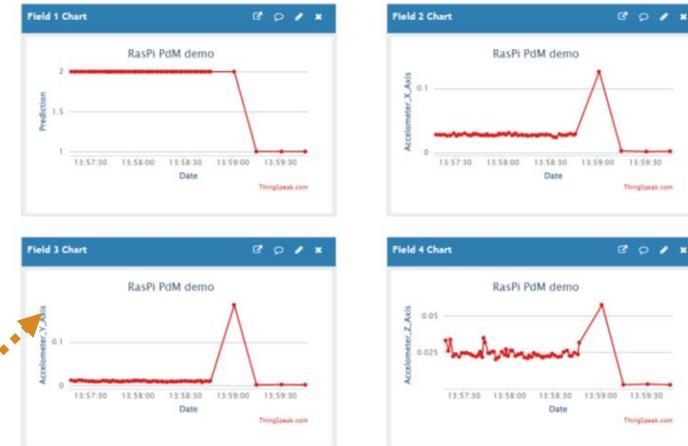
std値
左: OFF、右: ON



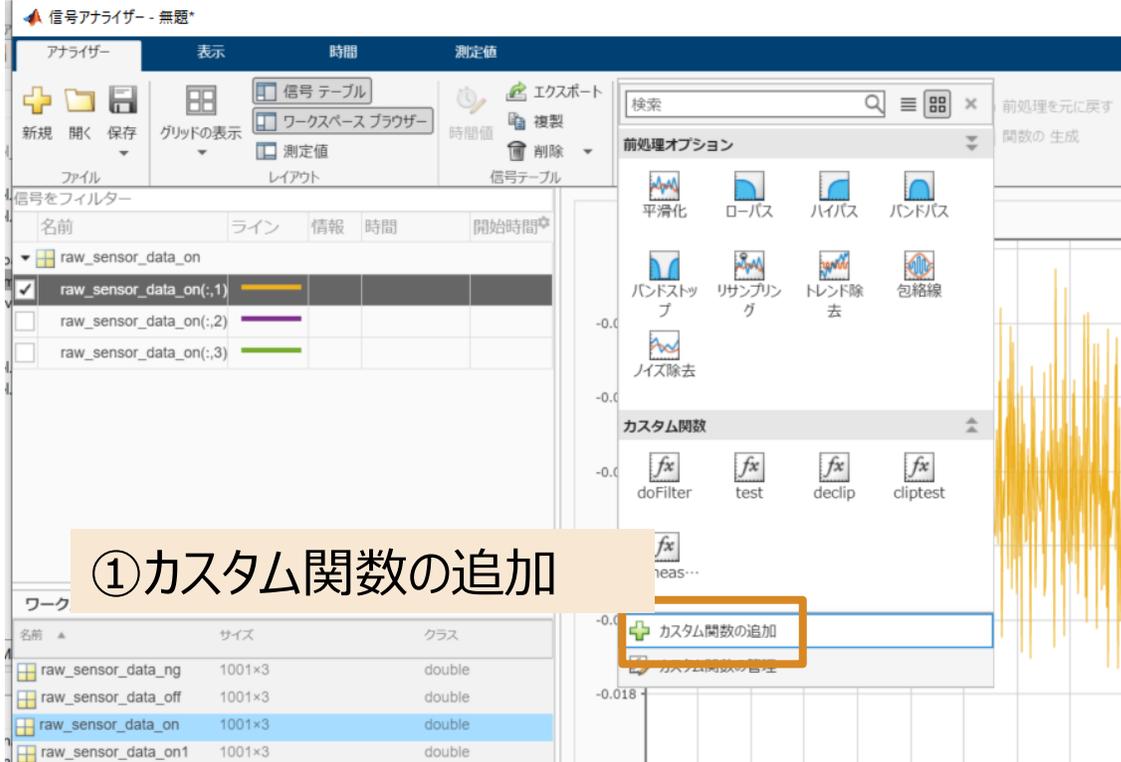
LEDマトリクスに表示



結果をクラウドで確認



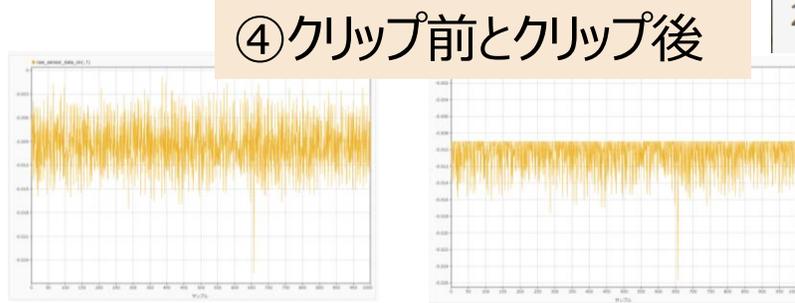
補足: カスタム関数の追加



```

1 function [y, tOut] = cliptest(x,tIn,varargin)
2
3 % 例1: 閾値以上の数値をクリップ
4
5 if isempty(tIn) % 「時間値」を「サンプル単位
6     tOut = [];
7 else % 「時間値」を設定した場合
8     t = tIn;
9     tOut = t;
10 end
11
12 if nargin<3
13     th = 1;
14 else
15     th = varargin{1}; % 引数が複数ある場合:
16 end
17
18 y = x;
19 y(x>th) = th;
20
21 end
    
```

② 追加する関数を選択 (信号のクリップ例)



DP7: ないものは作る

Step2 ON/OFF/NGの判別

ON/OFF/NGを判別する

ON



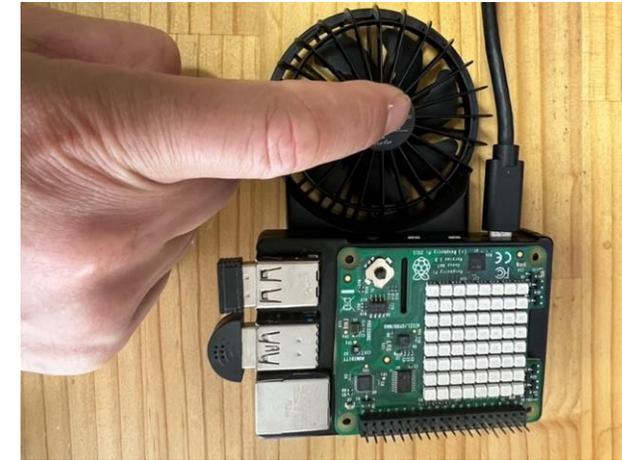
回転状態
(正常)

OFF



停止状態

NG



異常状態
(振動を加える)

データの観察（信号アナライザー）：ON, OFF, NG



3状態の判別に効果的な特徴は？

movie(診断特徴デザイナーと分類学習器)

MATLAB R2022a

ホーム プロット アプリ

アプリケーションの検索 Takemotoさん

アプリの設計 さらにアプリを取得 アプリのインストール アプリのパッケージ化

ファイル アプリ

曲線フィッター 信号アナライザー Audio Test Bench フィルターデザイナー 信号ラベラー 信号多重解像度アナライザー 診断特徴デザイナー Sensor Array Analyzer 分類学習器 Raspberry Pi リソース モニター ディープ ネットワーク デザイナー

E:\mwork2\mydemo_app\PdM\RasPiFan\modified22av2

現在のフォルダー

名前	更新日	サイズ	タイプ
slprj	2022/05/12 13...		フォルダー
step1_daq_ert_rtw	2022/05/16 13...		フォルダー
step2_simplemodel_e...	2022/05/16 13...		フォルダー
step3_MLmodel_ert_rtv	2022/05/12 14...		フォルダー
untitled_ert_rtw	2022/05/12 13...		フォルダー
crestfactor_shapafact...	2022/05/12 18...	1 KB	スクリプト
sense_hat_test.m	2022/05/16 10...	1 KB	スクリプト
step2_rawdata.mat	2022/05/12 14...	28 KB	MAT ファイル
trainedModel_svm.mat	2022/05/12 14...	6 KB	MAT ファイル
data_prep.mlx	2022/05/16 17...	4 KB	ライブ スクリプト
step3_MLmodel.slx.ori.	2022/05/11 9...	51 KB	ORIGINAL ファ...

コマンド ウィンドウ

```
fx >>
```

ワークスペース

名前	値
raw_sensor_data_ng	1001x3 double
raw_sensor_data_off	1001x3 double
raw_sensor_data_on	1001x3 double
raw_sensor_data_on1	1001x3 double
raw_sensor_data_on2	1001x3 double
tt	3x4 table
ttx	3x1 table
tty	3x1 table
ttz	3x1 table

Windows taskbar: 17:59

特徴量の評価 (診断特徴デザイナー)

%% timetable型で各状態 (OFF/ON/NG) のx軸データを定義し、tableとして結合

```
ttx = table([array2timetable(raw_sensor_data_off(:,1),'SampleRate',50));...
{array2timetable(raw_sensor_data_on(:,1),'SampleRate',50)};...
{array2timetable(raw_sensor_data_ng(:,1),'SampleRate',50)}});
ttx.Properties.VariableNames = {'accx'};
```

...(y,z軸データにも同様の処理)

%% xyz3つのテーブルを結合し、各状態に1,2,3のラベルを定義

```
tt = horzcat(ttx, tty, ttz);
tt.condition = [1;2;3]
```

>> tt

tt =

3 × 4 table

accx	accy	accz	condition
{1001 × 1 timetable}	{1001 × 1 timetable}	{1001 × 1 timetable}	1
{1001 × 1 timetable}	{1001 × 1 timetable}	{1001 × 1 timetable}	2
{1001 × 1 timetable}	{1001 × 1 timetable}	{1001 × 1 timetable}	3



診断特徴デザイナーを
起動

診断特徴デザイナーに入力する
データセットを用意

特徴量の評価 (診断特徴デザイナー: cont'd)

信号を分割する場合は
フレームポリシーを
"フレームベース"に設定

The screenshot displays the MathWorks diagnostic feature designer interface. On the left, the 'New Session' window shows 'tt' selected as the data set. The 'Source Variables' panel lists 'Time', 'Var1', and 'condition', with 'condition' highlighted as a 'Feature Variable'. The 'Feature Designer' window shows 'condition' set as a 'Feature Variable' and 'Time (seconds)' as the independent variable. The 'Frame Policy' dialog is open, with 'Frame-based' selected. The main plot shows 'accy/Var1' over time, with three conditions (1, 2, 3) overlaid. The x-axis is labeled 'Time' and the y-axis is 'accy/Var1'.

データセット

**"condition"を
"状態変数"とする**

**フレームポリシーを
"フレームベース"に設定**

x軸の波形

変数名	変数の型
Var1	信号
condition	特徴量

特徴量の評価 (診断特徴デザイナー) cont'd

診断特徴デザイナー

特徴デザイナー | 時間領域の特徴 | 信号の特徴

すべての特徴

<input checked="" type="checkbox"/> 平均	<input checked="" type="checkbox"/> 形状係数	<input checked="" type="checkbox"/> クレストファクター	<input checked="" type="checkbox"/> ピーク値	<input checked="" type="checkbox"/> S/N比
<input checked="" type="checkbox"/> RMS	<input checked="" type="checkbox"/> 尖度	<input checked="" type="checkbox"/> インパルス係数		<input checked="" type="checkbox"/> 全高調波歪み
<input checked="" type="checkbox"/> 標準偏差	<input checked="" type="checkbox"/> 歪度	<input checked="" type="checkbox"/> クリアランス係数		<input checked="" type="checkbox"/> SINAD

統計的な特徴 | インパルス性の特徴 | 高調波の特徴

診断特徴デザイナー

特徴デザイナー | ヒストグラム

現在のフレームポリシー: FRM_1 (サイズ: 1, レ...

現在の独立変数: Time (seconds)

抽出された特徴量のヒストグラム:

- accz_sigstats/Std
- accy_sigstats/Std
- accy_sigstats/ShapeFactor
- accx_sigstats/Std
- accx_sigstats/CrestFactor

xyz軸のデータについて
特徴を自動抽出

診断特徴デザイナー

特徴デザイナー | 特徴のランク付け

現在のフレームポリシー: FRM_1 (サイズ: 1, レ...

現在の独立変数: Time (seconds)

抽出された特徴量の重要度ランキング:

特徴量	1因子ANOVA
accx_sigstats/Std	407.6963
accy_sigstats/ShapeFactor	333.7944
accz_sigstats/Std	301.0917
accy_sigstats/Std	258.5721
accx_sigstats/CrestFactor	253.9930
accx_sigstats/ShapeFactor	230.9791
accx_sigstats/ImpulseFactor	195.3490
accx_sigstats/ClearanceF...	183.4278
	176.9880
	145.6617
	139.5382
	136.0597
	129.0261
	127.0797
	126.2292
	115.7770
	78.5148
	75.5474

抽出された特徴量の上位ランキング:

- accx_sigstats/Std
- accy_sigstats/ShapeFactor
- accz_sigstats/Std
- accy_sigstats/Std
- accx_sigstats/CrestFactor

抽出した特徴量の
上位ランキング

クレストファクタとシェイプファクタ

■ 平均値 : $\frac{1}{N} \sum_{n=1}^N |x_n|$

>>mean(abs(x))

■ RMS : $\sqrt{\frac{1}{N} \sum_{n=1}^N |x_n^2|}$

>>rms(x)

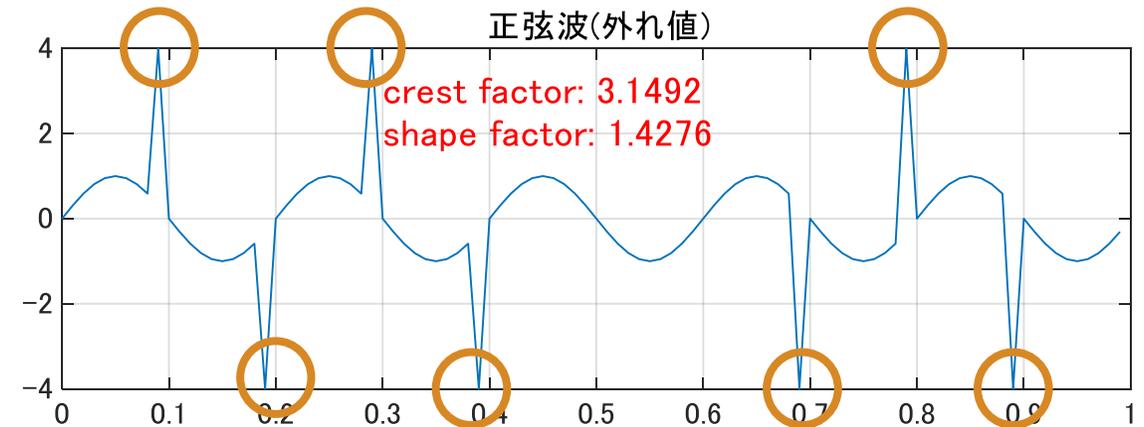
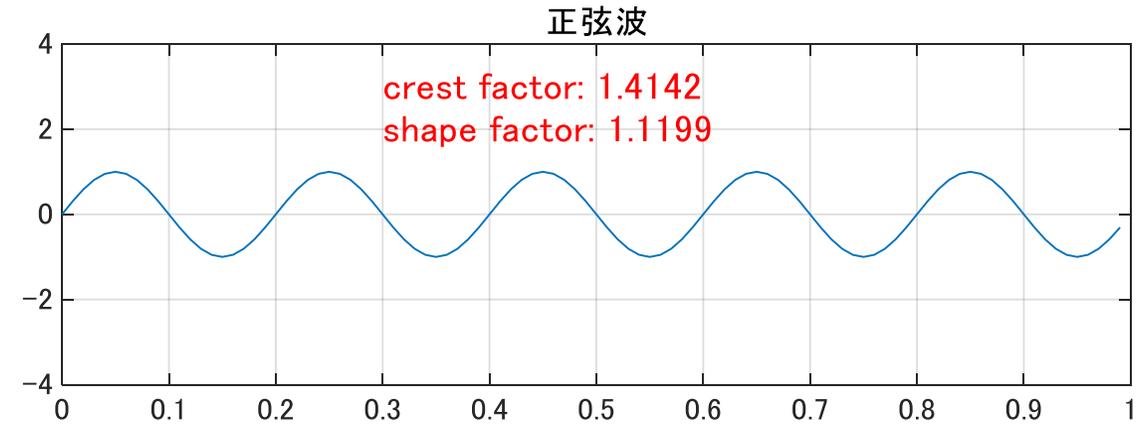
■ クレストファクタ : ピーク最大値/RMS

>>peak2rms(x)
(>>max(abs(x))/rms(x))

■ シェイプファクタ : RMS/平均値

>>rms(x)/mean(abs(x))

DP8: 代表的な統計量はおさえておく



e.g.) パルスノイズの影響

純正弦波に対して...

- クレストファクタ : 最大値増加にともない増大
- シェイプファクタ : RMSの増加にともない微増

特徴量の評価 (分類学習器)

ファイルからの新規セッション

データセット

データセット変数: FeatureTable1 (603x7 table)

応答: condition (double, 1 .. 3)

検証

検証方法: 交差検証

過適合を防止します。検定用に確保した分以外のデータについては、アプリはデータを区画に分割し、各区画の精度を推定します。

交差検証の分割数: 5

予測子

名前	タイプ	範囲
<input type="checkbox"/> FRM_1/TimeStart	duration	< 不適合 >
<input type="checkbox"/> FRM_1/TimeEnd	duration	< 不適合 >
<input checked="" type="checkbox"/> FRM_1/Var1_sigstats/CrestFactor	double	1.00034 .. 2.50882
<input checked="" type="checkbox"/> FRM_1/Var1_sigstats/ImpulseFactor	double	1.00034 .. 3.16553
<input checked="" type="checkbox"/> FRM_1/Var1_sigstats/Std	double	0.000324537 .. 0.0381214

すべて追加 すべて削除

データの準備方法

⚠ 応答変数が数値です。ユニークな値がクラスラベルとして解釈されます。

セッションの開始 キャンセル

予測子 (特徴量) の選択

予知保全のための分類学習器 - untitled*

分類学習器

すべてのクイック学習 すべてのモデル

モデル

並べ替え: モデル番号

1 ツリー ドラフト 最終更新: 複雑な木 特徴: 5/5

オリジナルデータセット: FeatureTable1

プロット

データ

モデル予測

予測子

X: FRM_1/accx_sigstats...

Y: FRM_1/accx_sigstats...

クラス

表示	順序
<input checked="" type="checkbox"/>	1
<input checked="" type="checkbox"/>	2
<input checked="" type="checkbox"/>	3

混同行列 (2次のSVM)

プロット

観測値の数

真陽性率 (TPR)

真陰性率 (FNR)

陽性の予測値 (PPV)

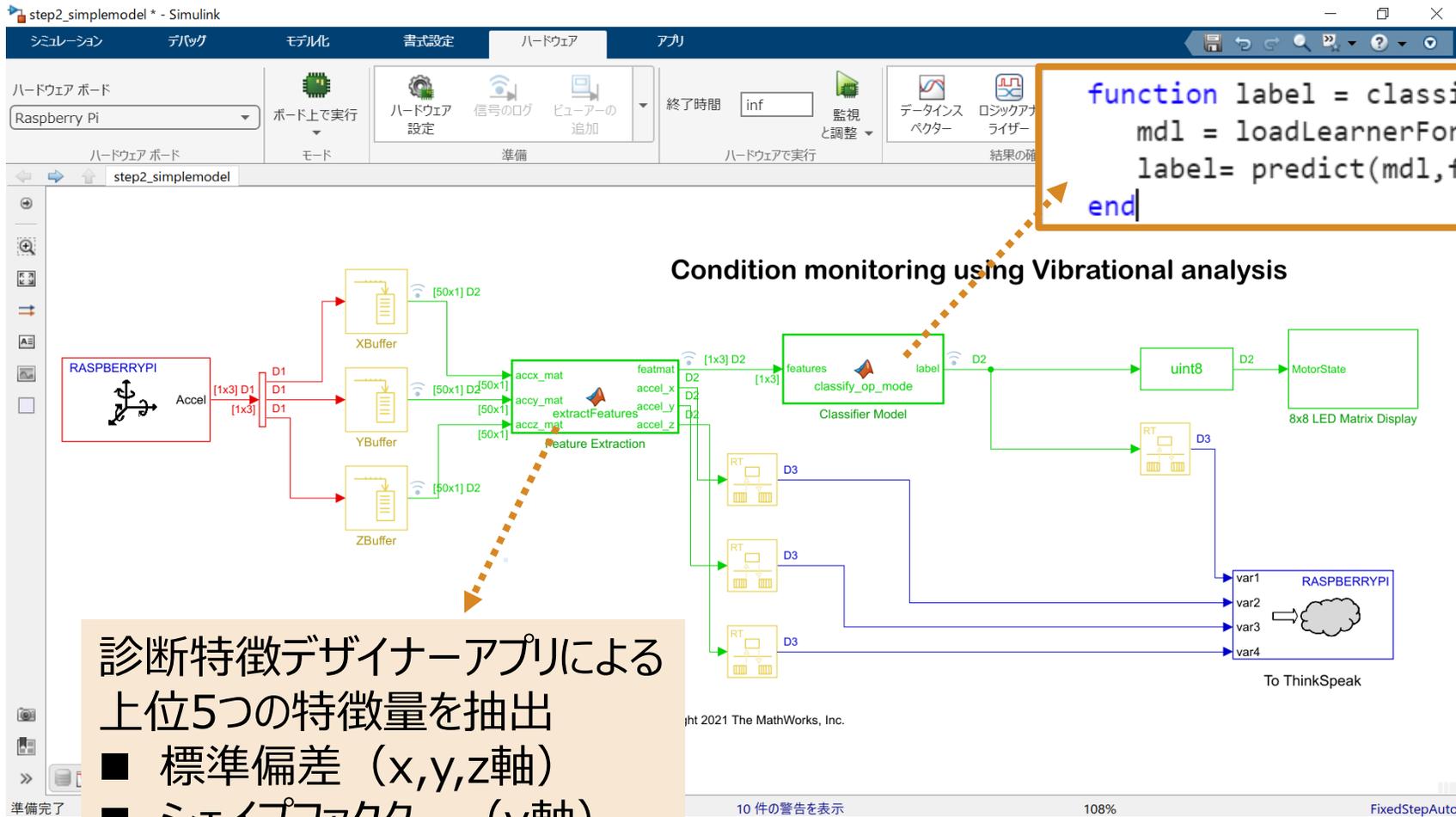
偽陽性率 (FDR)

混同行列とは

散布図

混同行列 (2次のSVM)

既存モデルをON/OFF/NGの判別用モデルに修正



```
function label = classify_op_mode(features)
    mdl = loadLearnerForCoder('trainedModel_svm.mat');
    label= predict(mdl,features);
end
```

分類学習器アプリで
学習済みのモデル

- OFF : 1
- ON : 2
- NG : 3

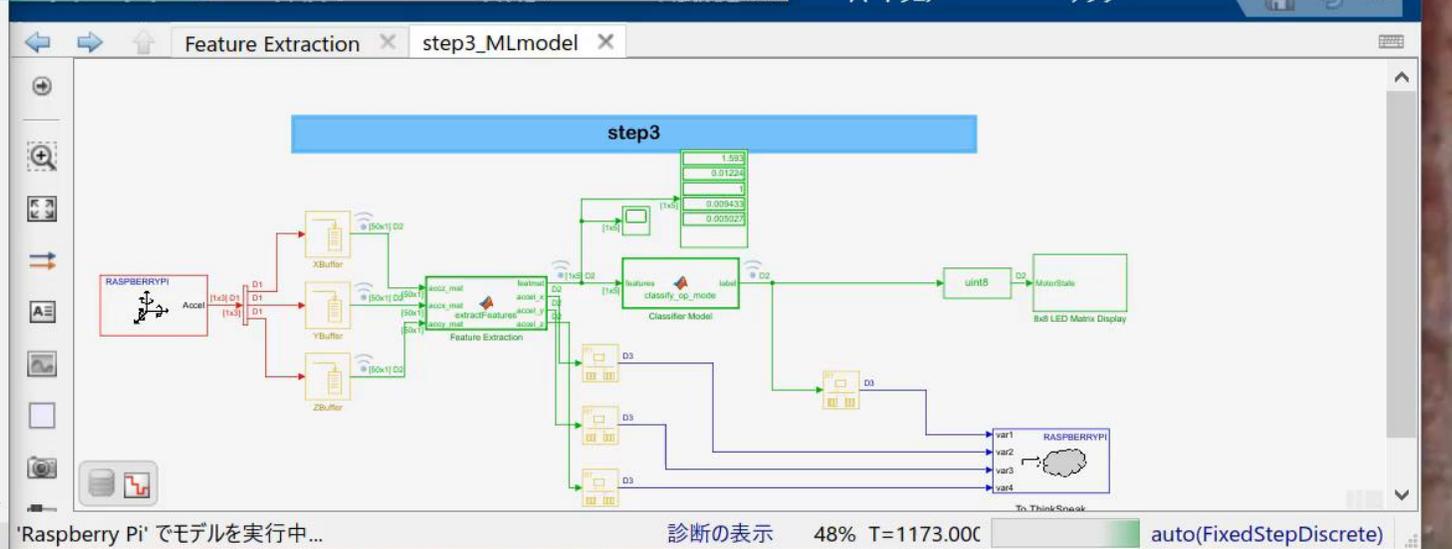
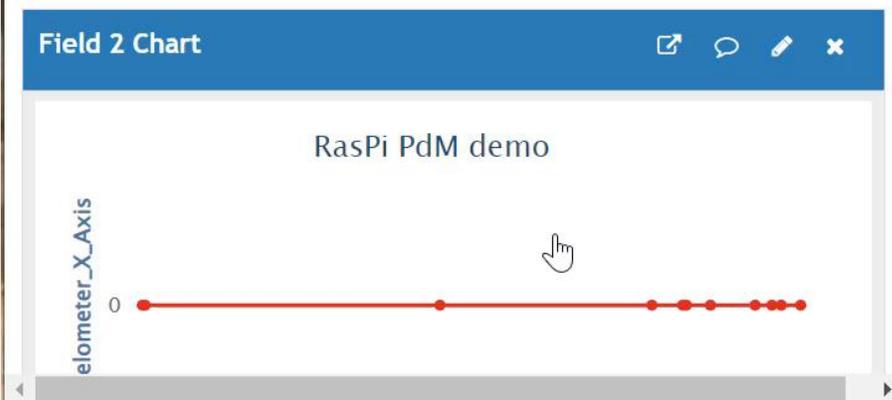
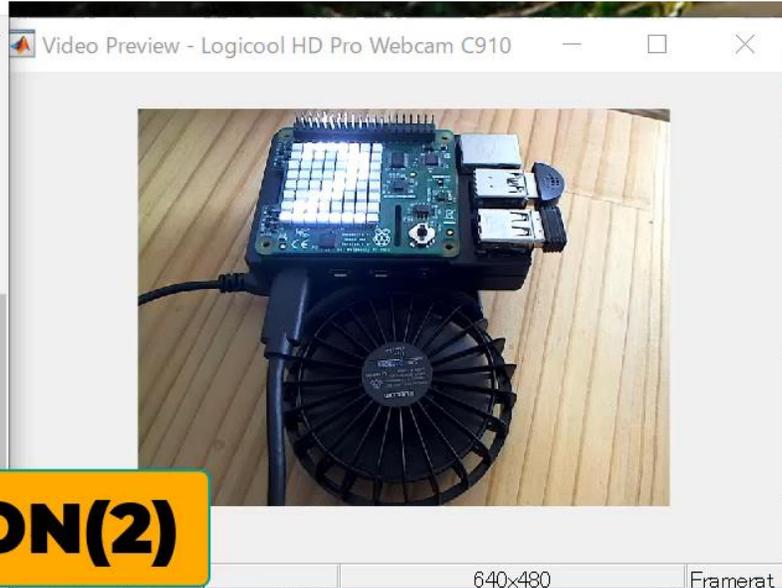
診断特徴デザイナーアプリによる
上位5つの特徴量を抽出

- 標準偏差 (x,y,z軸)
- シェイプファクター (y軸)
- クレストファクター (x軸)

Demo (再掲)



ON(2)



まとめ2

まとめ

■ 取り上げたトピック

- データの収集
- 信号特性の吟味
- 特徴量の試行錯誤
- Raspberry Piへの実装
- 遠隔監視の方法

泥臭8策

- 一、便利なツールを知る事
- 一、備品の選定に気を配る事
- 一、サンプル時間は適切に設定する事
- 一、シンプルな題材から試す事
- 一、着目すべき"軸"を見極める事
- 一、あるものは活用する事
- 一、ないものは作る事
- 一、代表的な統計量はおさえておく事

■ 今後の課題

- 周波数特徴量、時間-周波数特徴量等も試してみる
- 他の状態（風力の強弱）を追加してみる
- ディープラーニングで試してみる