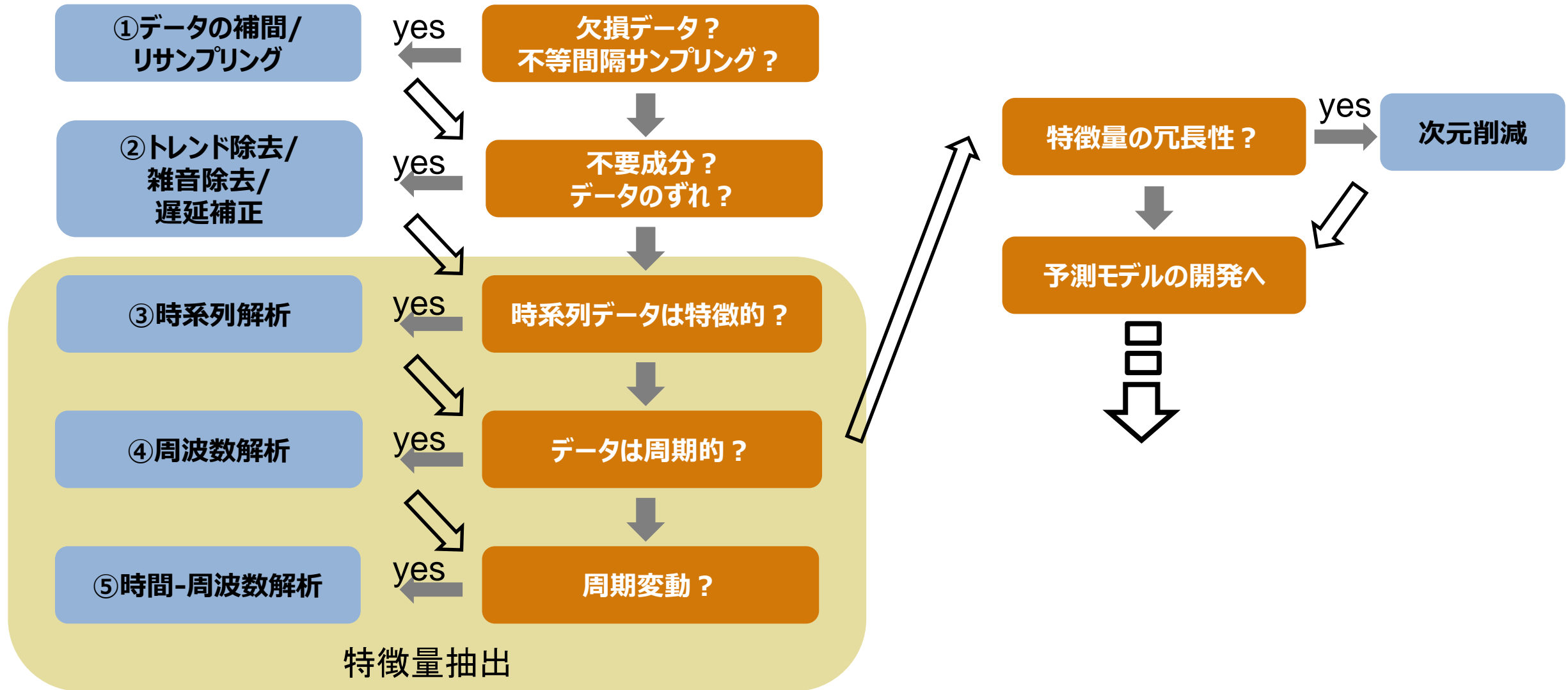


予知保全のための信号処理 ～前処理～

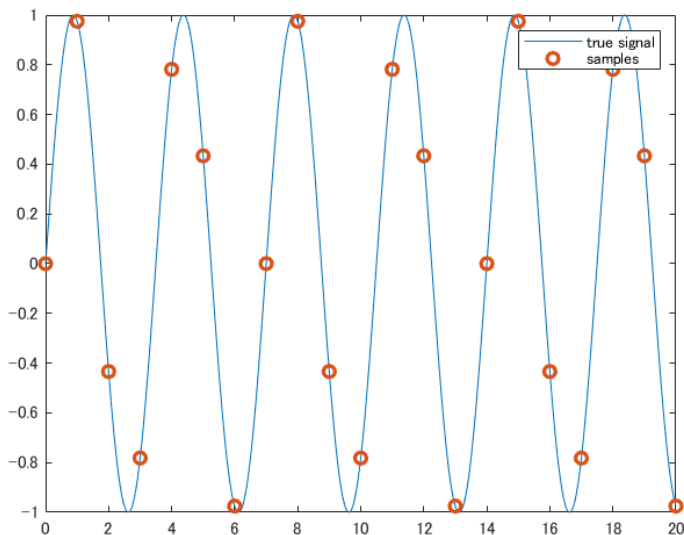
アプリケーションエンジニアリング部
竹本佳充

"データの前処理"ワークフロー



①データの補間: **interp1/resample**

測定波形と真値



% 線形補間/スプライン補間

```
>>y = interp1(t1,x1,t2,'method');
```

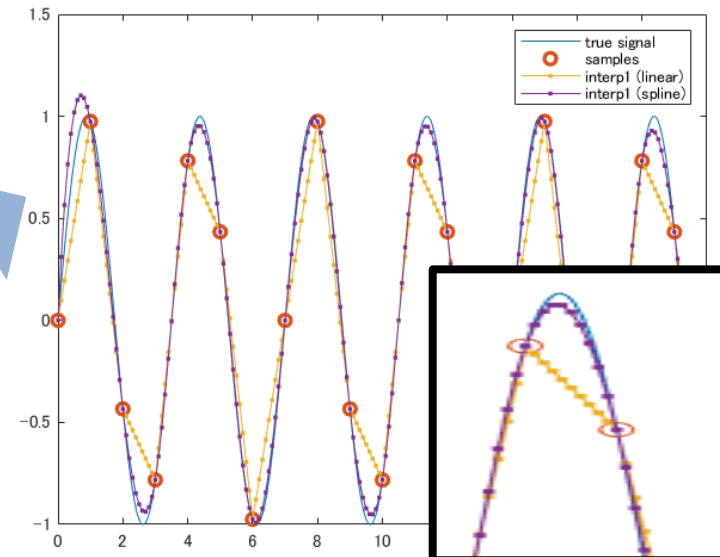
- t1,x1: 対象信号
- t2:補間後の時間ベクトル
- 'method': 補間メソッド
 - linear: 線形補間
 - spline: スプライン補間
- y: 補間後の信号

% アンチエイリアシングフィルタによるリサンプリング

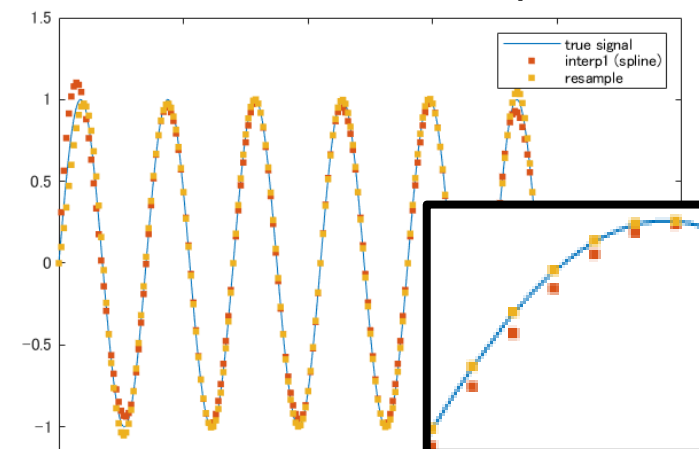
```
>>y = resample(x, p, q);
```

- x: 対象信号
- p,q: p/q倍でリサンプリング
- y: 補間後の信号

線形 vs スプライン補間

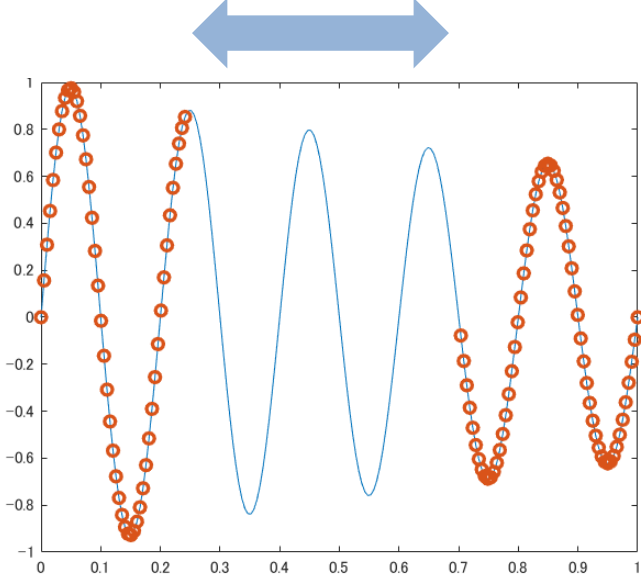


スプライン補間 vs resample関数

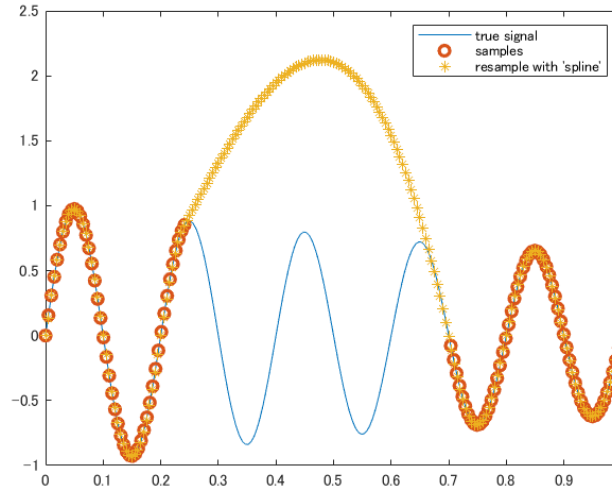


①欠損データの再現: **fillgaps**

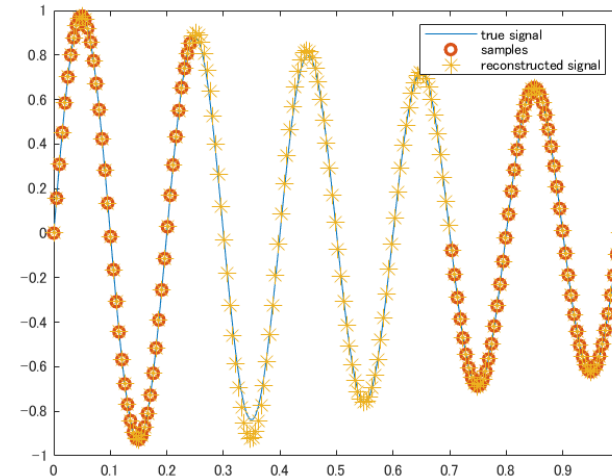
約0.4秒間の欠損波形と真値



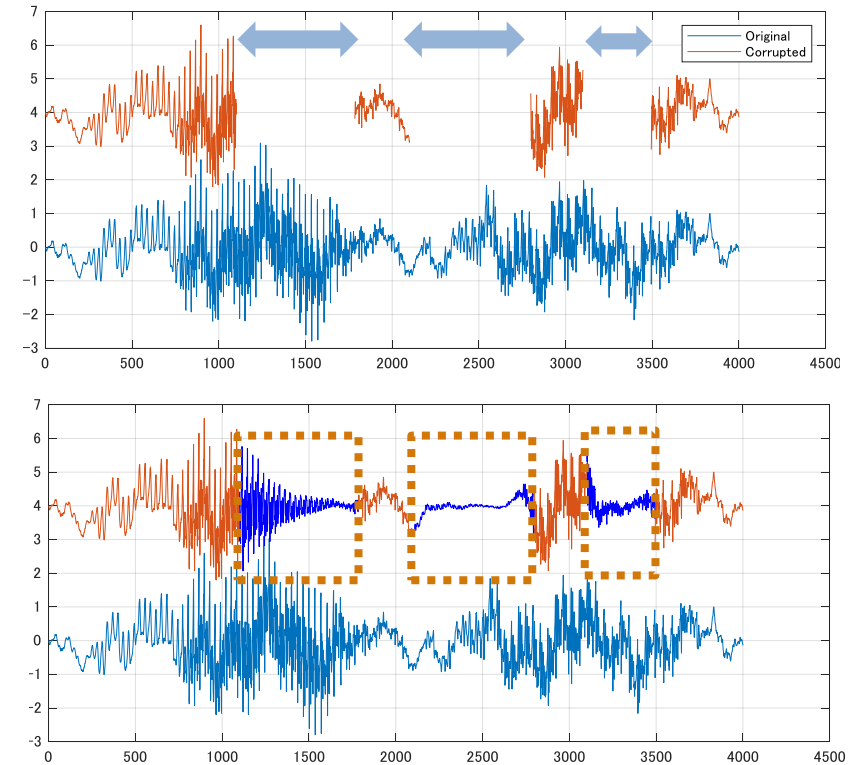
resample関数



fillgaps関数



音声信号への適用例



% 自己回帰モデルによる補間

>> **y = fillgaps(x,len);**

- x: 対象信号
- len: 推定に使用するデータ長
- y: 補間後の信号

■ 局所的な推定にも
使用可能

②トレンド・雑音を除去する: **polyfit/polyval/sgolayfilt**

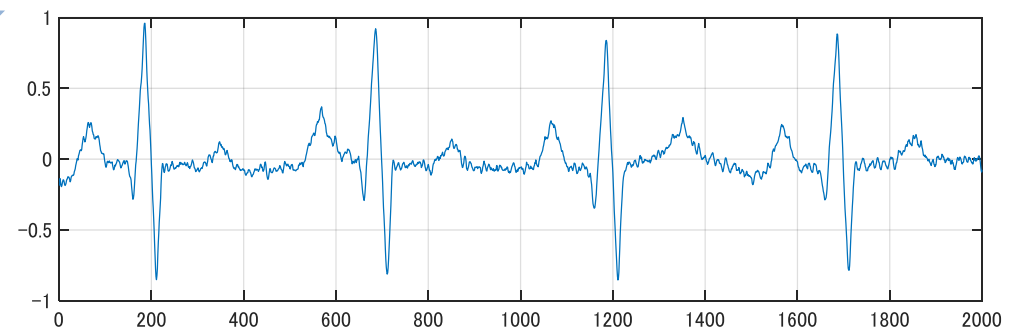
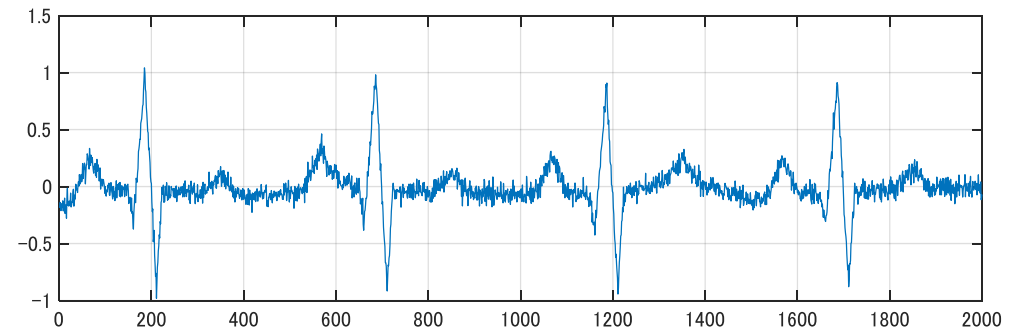
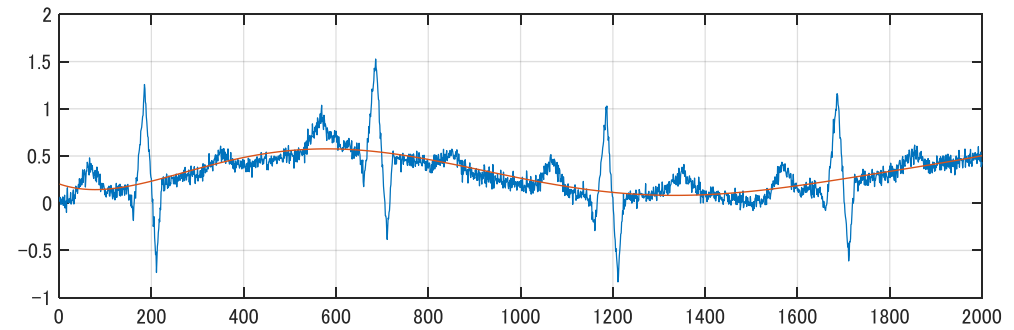
% 多項式近時とその評価

```
>>[p,s,mu] = polyfit(t,in,6);  
>>f_y      = polyval(p,t,[],mu);  
>>out      = in - f_y;
```

%Savitzky-Golayフィルタ (移動平均フィルタの一種)

```
>>out = sgolayfilt(in,7,21);
```

- 多項式近似でトレンド除去
- フィルタリング機能各種の活用



②注目したい成分を強調する: tsa

% TSA: 時間同期平均

>> tsa(x,fs,tPulse)

- x: 入力信号
- fs: xのサンプリング周波数
- tPulse: 同期タイミング信号

% e.g.) 雑音を含む正弦波の時間同期平均

>> fs = 500;

>> t = 0:1/fs:10-1/fs;

>> f0 = 1;

>> x = sin(2*pi*f0*t) + randn(size(t))/5;

>> tPulse = 0:1/f0:max(t); %同期タイミング

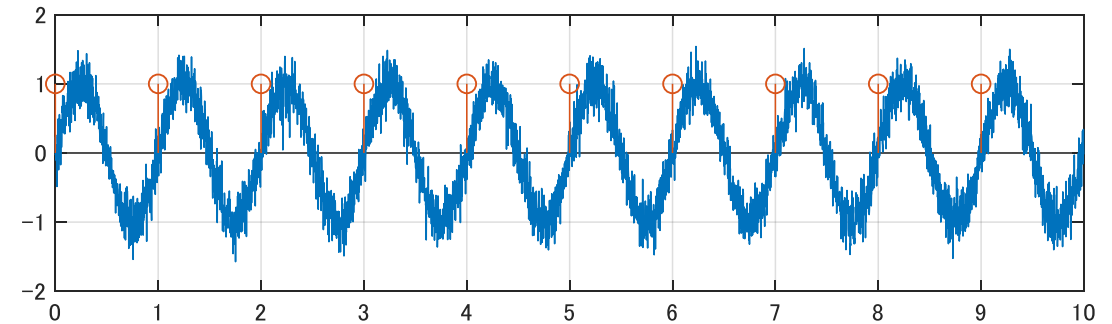
...

>> tsa(x,fs,tPulse) %1周期(デフォルト)

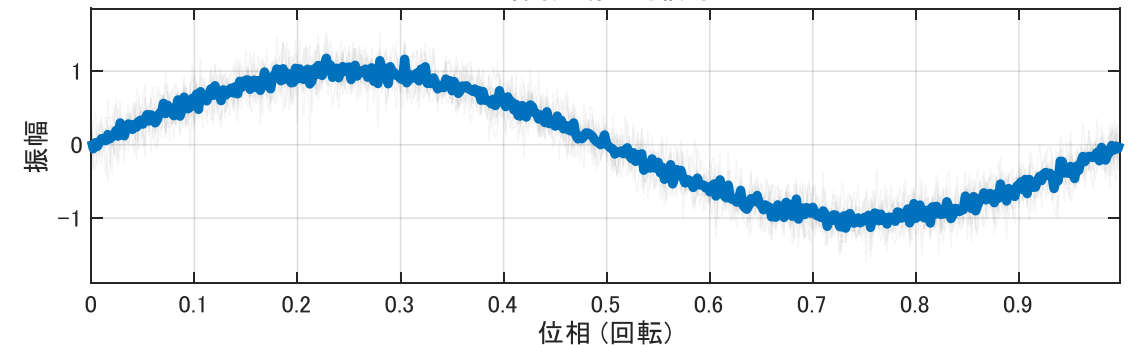
>> tsa(x,fs,tPulse, 'NumRotation, 2) %2周期

← 500点(1周期)

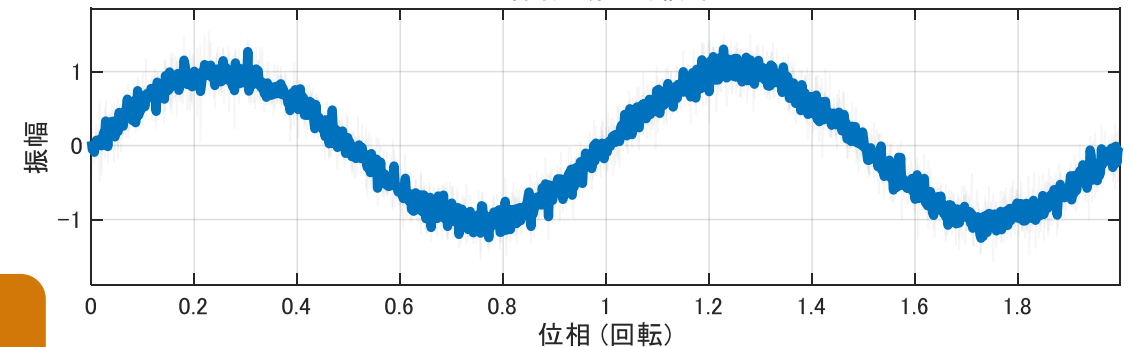
← 1000点(2周期)



時間同期平均信号



時間同期平均信号



R2017b

- ランダム雑音の様な、平均がゼロとなる成分除去に有効

②遅延を見積もる: **finddelay**

% 相互相関による遅延時間の見積もり

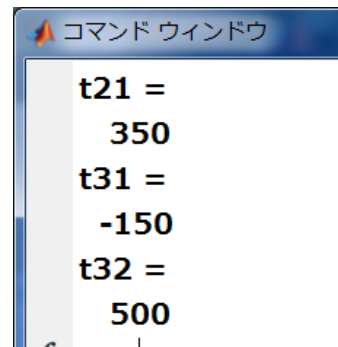
```
>> D = finddelay(x, y)
```

- **x, y**: 対象信号
- **D**: 二信号間の遅延時間

e.g.) 3信号間の遅延推定

```
>> t21 = finddelay(s2,s1)
>> t31 = finddelay(s3,s1)
>> t32 = finddelay(s2,s3)
```

推定された遅延量

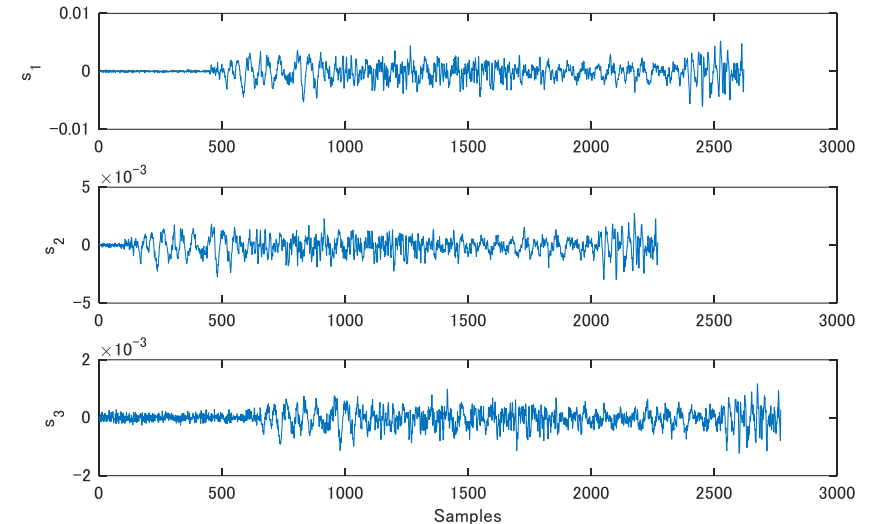


```
コマンド ウィンドウ
t21 =
    350
t31 =
   -150
t32 =
    500
```

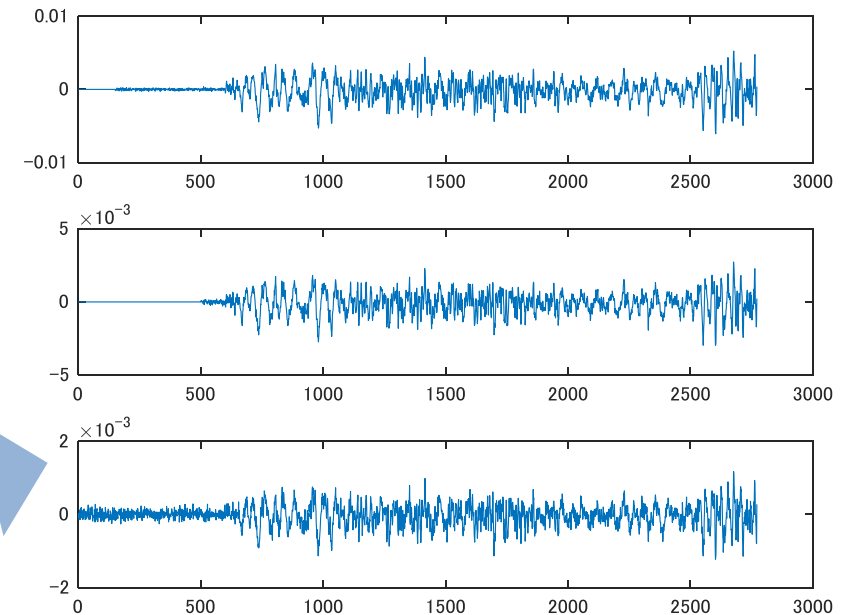
% 遅延量推定および信号をそろえる

```
>> [x1,x3] = alignsignals(s1,s3);
>> x2 = alignsignals(s2,s3);
>> subplot(3,1,1), plot(x1)
>> subplot(3,1,2), plot(x2)
>> subplot(3,1,3), plot(x3)
```

オリジナルのデータ



アライン後のデータ



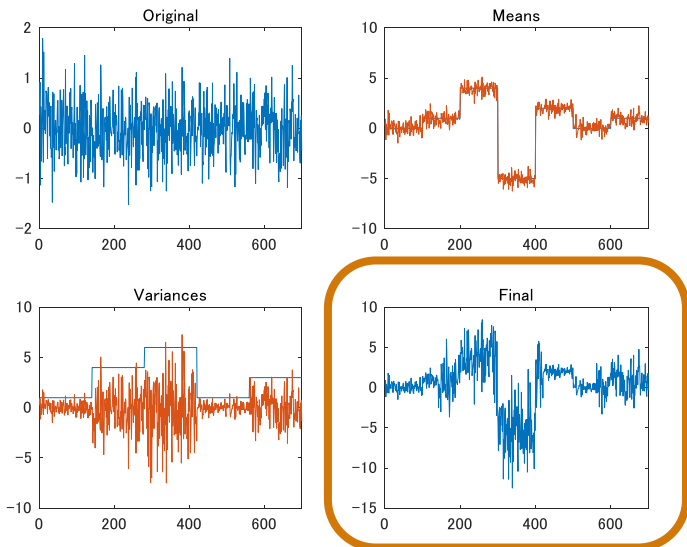
③変化点を見つける: findchangepts

% 信号の変化の大きい部分を検出
>>y = findchangepts(x)

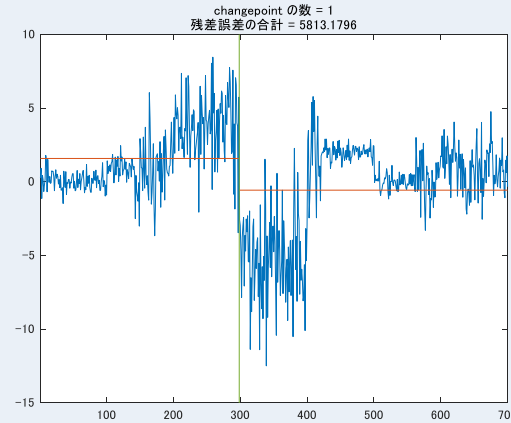
- **x**: 対象信号
- **y**: 変化点インデックス

e.g.)

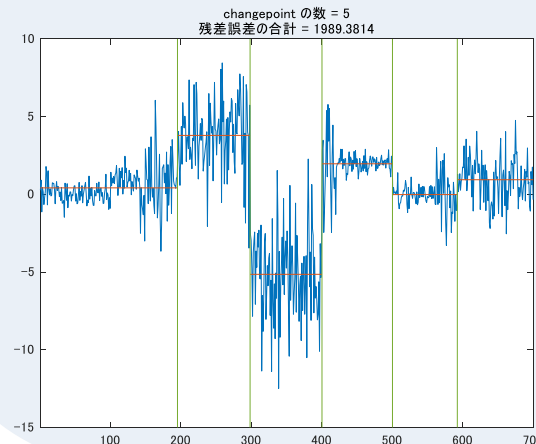
- 平均値: 7箇所に変化
- 分散: 5箇所に変化



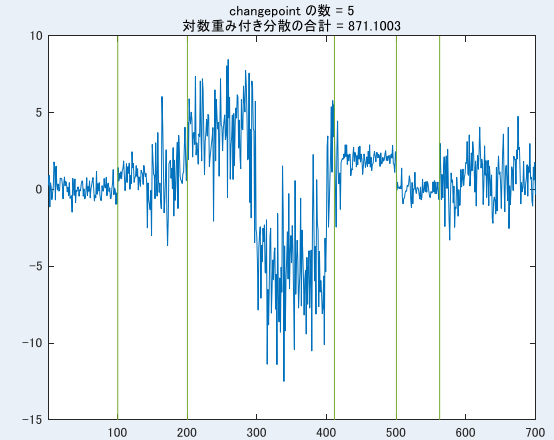
%% 制約なし
findchangepts(t)



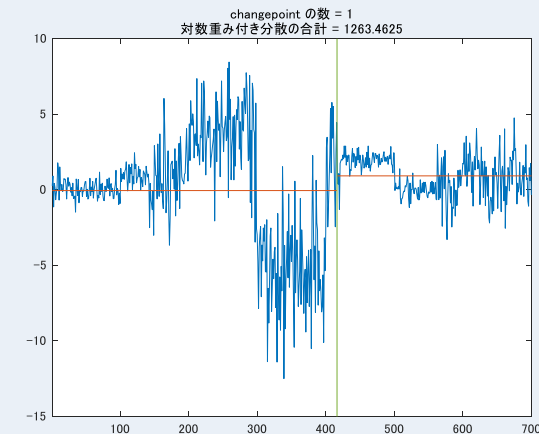
% 変化点の数で制約
findchangepts(t, 'MaxNumChanges', 5)



% RMSレベルが最も大きく変化する5点
findchangepts(t, 'MaxNumChanges', 5, ...
'Statistic', 'rms')



% 平均値と標準偏差が最も大きく変化する点
findchangepts(t, 'Statistic', 'std')

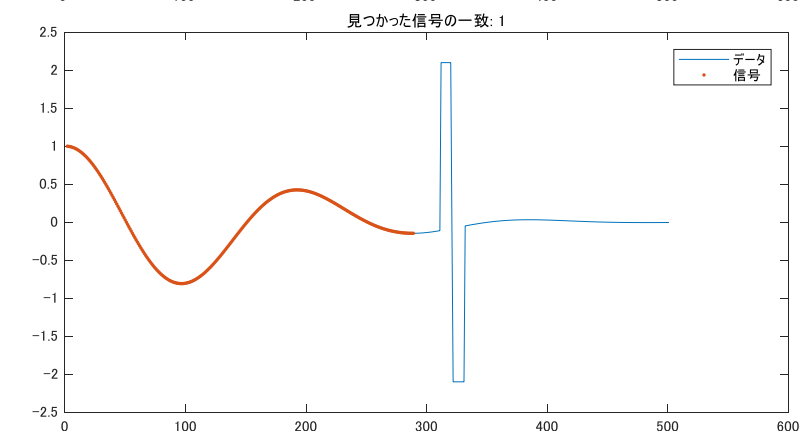
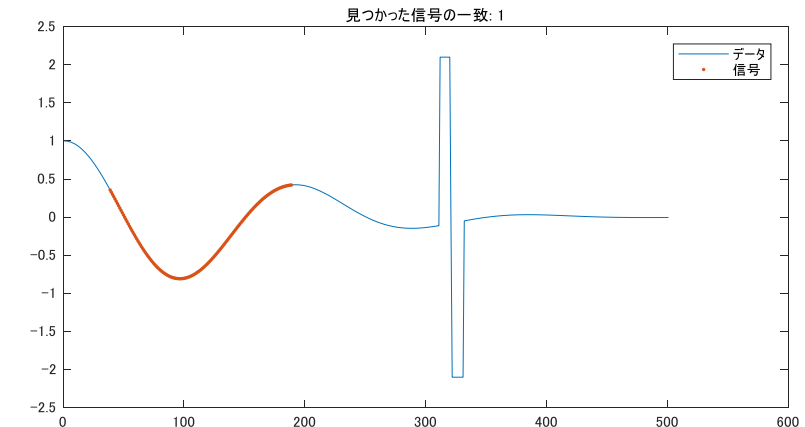
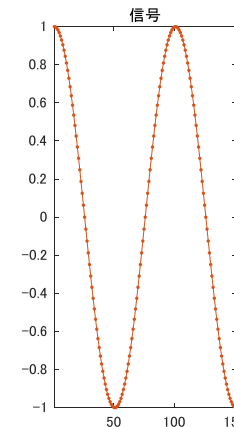


③似た成分を見つける: **findsignal**

% DTW, EDR等の類似性検索を応用した、信号の位置検出

>>[istart,istop,dist] = **findsignal**(x,signal,Name,Value)

- **x**: 対象信号
 - **signal**: 参照パターン
 - **Name, Value**: 検出条件名と、その値
-
- **istart, istop**: 検出部の開始/終了インデックス
 - **dist**: 距離



e.g.) 類似度判定におけるDTWの効果

>>findsignal(dt,signal,'Metric','absolute') %右上

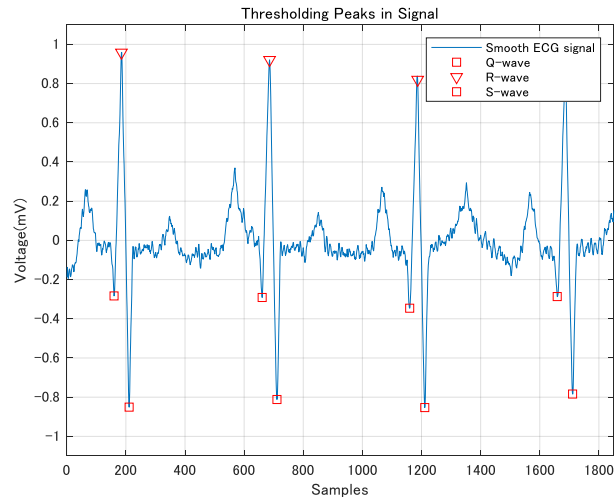
>>findsignal(dt,signal,'TimeAlignment','dtw','Metric','absolute') %右下

③④ ピークを見つける: **findpeaks**

% 信号のピーク探索

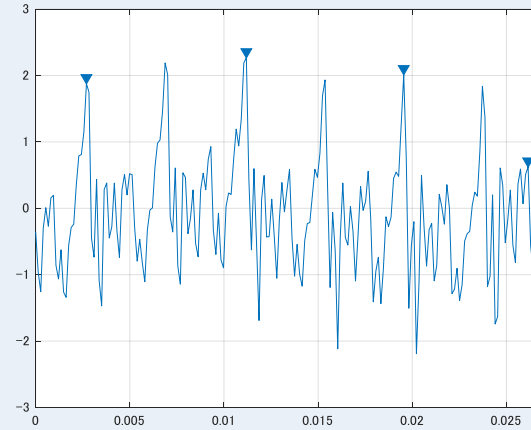
>> **findpeaks(x)**

■ **x**: 対象信号



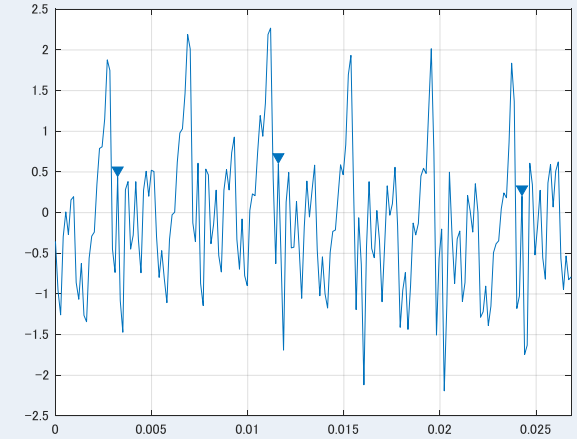
% ピーク間距離で制約

findpeaks(sig,Fs,'MinPeakDistance',0.005)



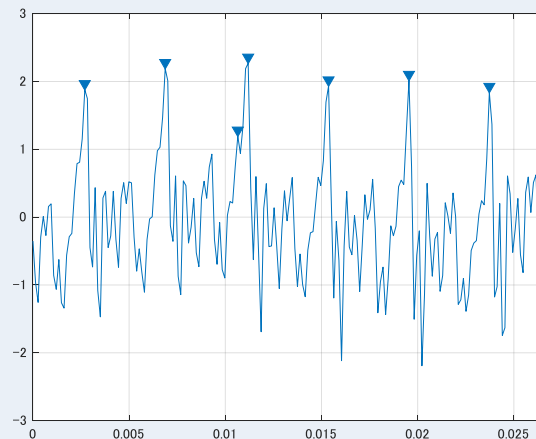
% 隣接データとの差で制約

findpeaks(sig,Fs,'Threshold',1)



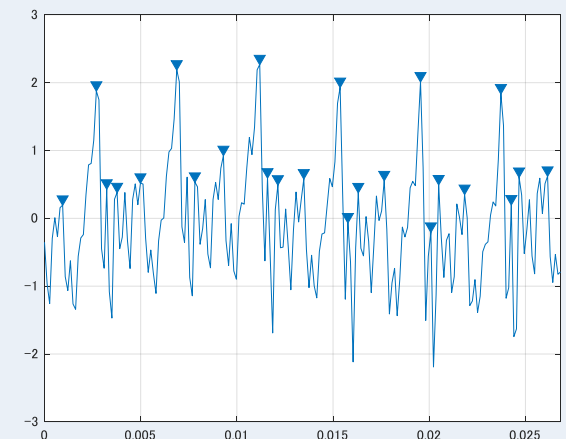
% ピーク最小値で制約

findpeaks(sig,Fs,'MinPeakHeight',1)



%% プロミネンス値 (左右の谷との高低差で制約)

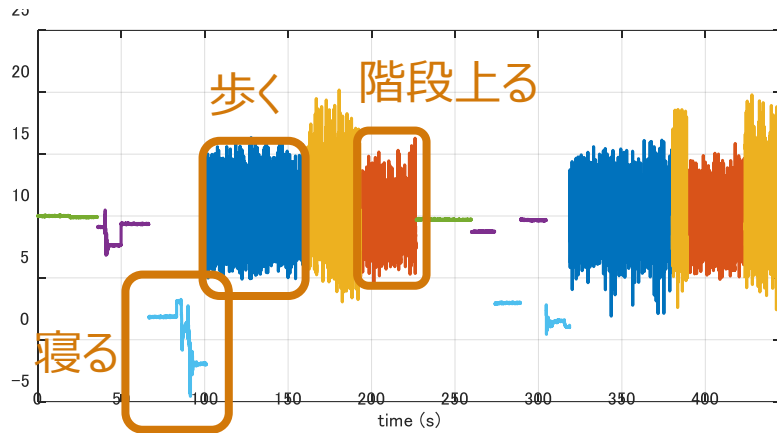
findpeaks(sig,Fs,'MinPeakProminence',1)



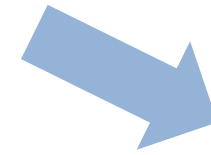
■ 各種制約条件の設定が可能

③④特徴量抽出

人の行動による
加速度センサーの
振る舞い

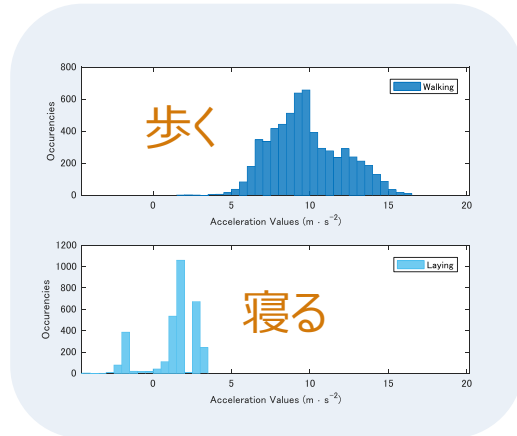


周波数解析

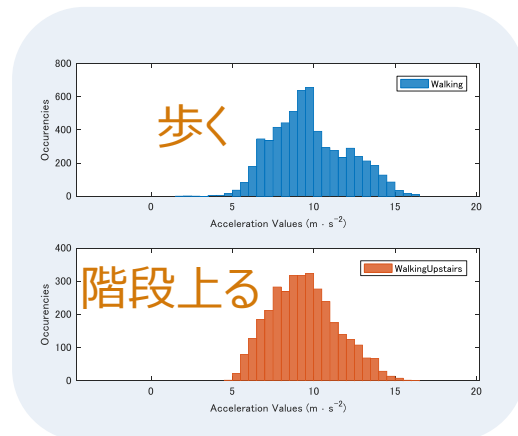


時間軸解析

% ヒストグラム
>> **histogram(x)**

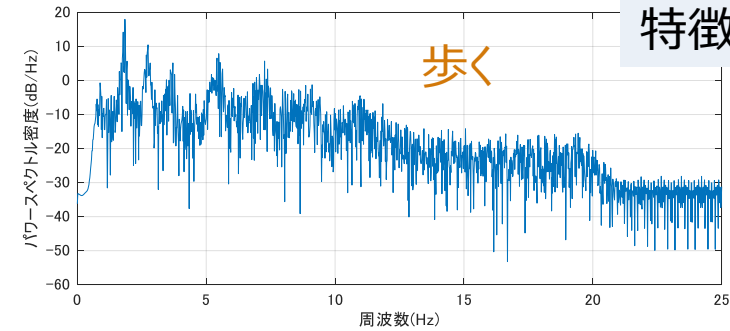


時間軸の情報で
分類可能



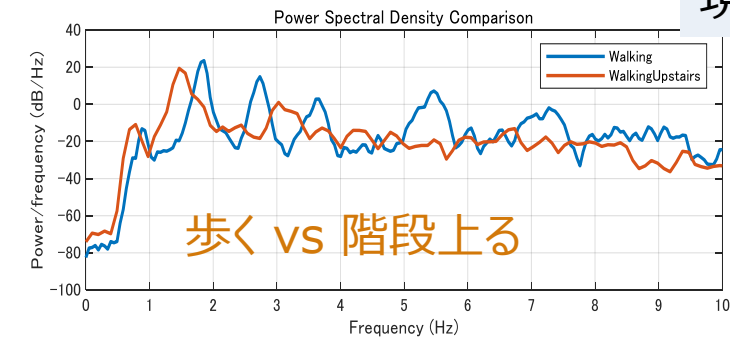
時間軸の情報だけでは
分類しきれない

% 周波数解析 (全区間fft)
>> **periodogram(x)**



特徴が現れる

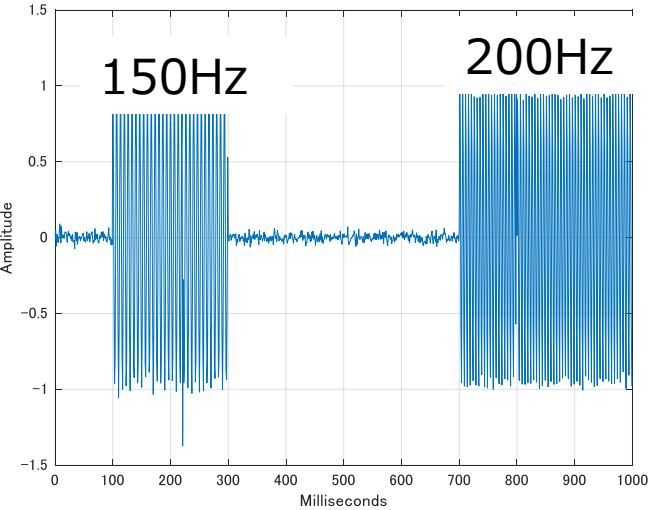
% 周波数解析 (短区間fft)
>> **pwelch(x)**



特徴が顕著に
現れる

■ 時間軸で十分な特徴が得られない場合、
周波数成分を考慮

⑤ 周期変動を持つ信号の特徴量



- 100ms-300ms: 150Hz
- 700ms-1s: 200Hz
- @222,800ms: ノイズ

% スペクトログラム

>> **spectrogram(x)**

% 連続ウェーブレット変換

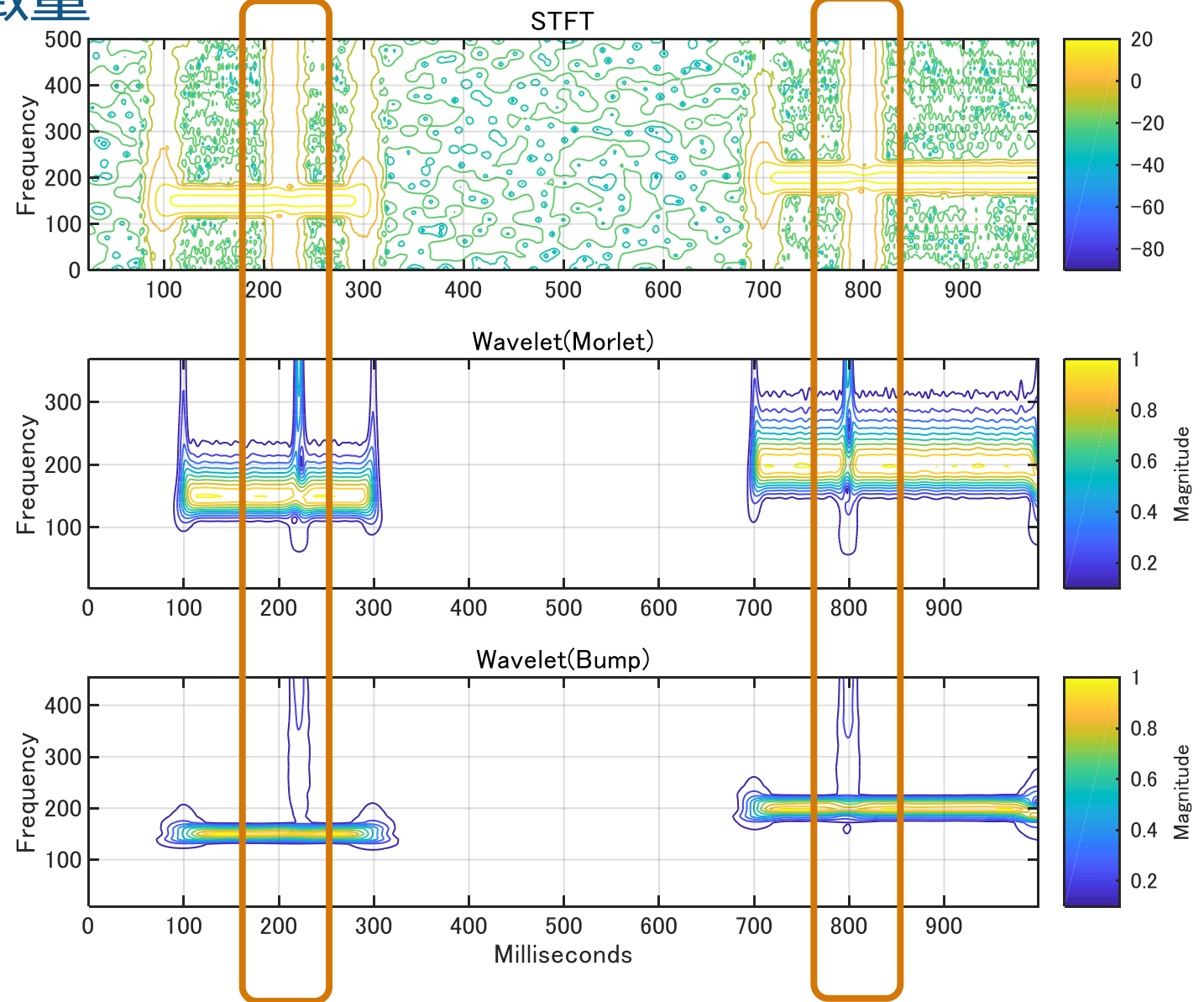
>> **cwt(x)**

スペクトログラム

ウェーブレット

■ 上: Morlet

■ 下: Bump



⑤ 周期変動を持つ信号の特徴量 (cont'd)

R2018b

■ **wvd**: ウィグナー分布および平滑化疑似ウィグナー分布

■ **ウィグナー分布 (連続信号)**

$$WVD_x(t, f) = \int_{-\infty}^{\infty} x\left(t + \frac{\tau}{2}\right) x^*\left(t - \frac{\tau}{2}\right) e^{-j2\pi f\tau} d\tau$$

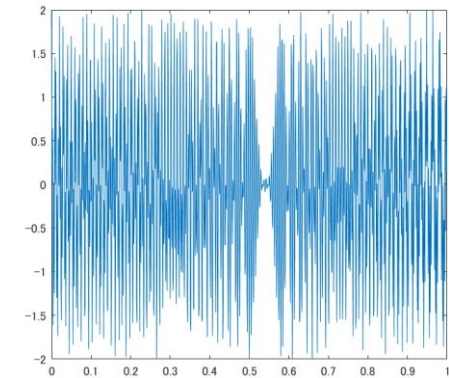
■ **ウィグナー分布 (離散信号)**

$$WVD_x(n, k) = \sum_{m=-N}^N x(n + m/2) x^*(n - m/2) e^{-j2\pi km/N}$$

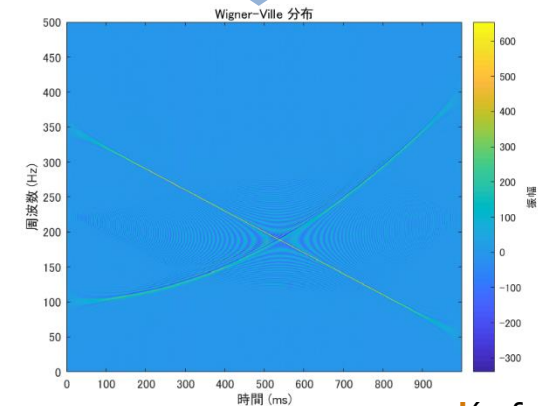
■ **平滑化疑似ウィグナー分布**

$$SPWVD_{g,H}(t, f) = \int_{-\infty}^{\infty} g(t) H(f) x\left(t + \frac{\tau}{2}\right) x^*\left(t - \frac{\tau}{2}\right) e^{-j2\pi f\tau} d\tau$$

■ チャープ信号
100→400Hz
350→50Hz

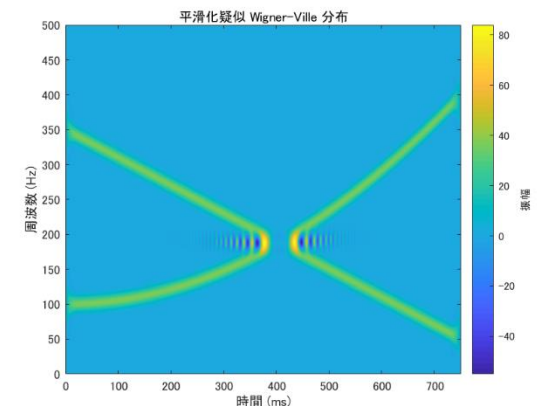


↓ `>> wvd(x,fs)`



■ **ウィグナー分布**

↓ `>> wvd(x,fs, ...'smoothedPseudo' ...)`



■ **平滑化疑似ウィグナー分布**

まとめ

■ 前処理

- 補間、リサンプリング
- 雑音除去
- 時間同期平均
- 遅延推定・補正

■ 主な使用ツール

- **Signal Processing Toolbox**
- **Wavelet Toolbox**

■ 特徴抽出

- 変化点推定
- 類似度
- ピーク検出
- 時系列分析
- 周波数分析
- 時間-周波数分析