Python Implementation and Extension of the Quench Heater Monitoring Framework

Christoph Obermair
1. Feature engineering → summarize historical data  
   e.g. resistance, characteristic time…

2. Classification → label features  
   e.g. ✔️, ☐️, no statement possible…

3. Analysis → comparison of features across time and circuits  
   e.g. clustering, trend analysis…
Presentation Outline:

1. Feature engineering
   a. Compression: Implementation and extension of the existing QH monitoring features

2. Classification
   a. Classification: Threshold based features classification to ✔ and ✗
   b. Comparison: Compare to LabVIEW classification and check differences with experts
   c. Extension: Extend existing classification methods

3. Trend analysis
   a. Analysis: Comparison of QH features across time and circuits

* Previously done in LabVIEW:
Z. Charifoulline et al., “Overview of the Performance of Quench Heaters for High-Current LHC Superconducting Magnets”, IEEE TAS 27(4), 06/2017
1. Feature engineering

a. Compression: implementation and extension of the existing QH monitoring features
1.a Compression: Which features are extracted?

GOAL: The features should summarize the properties of the exponential decay of the quench heater signals.

*contributed by Zinur Charfoulline
1.a Compression: Which features are extracted?

A. Initial values:
1. first = mean(medianf(data[0:19], w=3)) → save value for U, I and R
2. last = mean(medianf(data[−20:−1], w=3)) → save value for U, I and R

B. Characteristic time of exponential decay:
1. Charge approach: \[ \int_{\theta_0}^{\theta_1} f(t) d\theta = \int_{\theta_0}^{\theta_1} f_0(t) e^{-\frac{t}{\tau}} d\theta \] → save scalar \( \tilde{\tau} \) for U and I
2. Energy approach: \[ \int_{\theta_0}^{\theta_1} f^2(t) d\theta = \int_{\theta_0}^{\theta_1} f_0^2(t) e^{-\frac{t}{\tau}} d\theta \] → save scalar \( \tilde{\tau} \) for U and I
3. Linear regression: \[ \text{min}\left(f(t) - (p_0 + p_1 x)\right)_p \] → save scalar \( p_1 \) for U and I
4. Exponential fit: \[ \text{min}\left(f(t) - p_0 e^{-p_2(t-p_1)}\right)_p \] → save scalar \( p_2 \) for U and I
5. Change in characteristic time: \[ \frac{f(t)}{f(t')} = -\frac{\tilde{\tau}}{\tau} \frac{f_0(t) e^{-\frac{t}{\tilde{\tau}}}}{f_0(t) e^{-\frac{t'}{\tilde{\tau}}}} \] → save mean & std of vector \( \tilde{\tau} \) for U and I
1.a Compression: Which features are extracted?

C. Signal Similarity:

1. Normalize signals:

$$f^* = \frac{f - \min(f)}{\max(f) - \min(f)}$$

2. Euclidean distance within the signals:

$$\|f_i^* - f_j^*\|_2 = \sqrt{\sum_{t=0}^{T} (f_i^*(t) - f_j^*(t))^2} \quad \forall \ i, j = 1, 2, 3, 4$$

$$\begin{pmatrix}
0 & \|f_1^* - f_2^*\|_2 & \|f_1^* - f_3^*\|_2 & \|f_1^* - f_4^*\|_2 \\
\vdots & 0 & \|f_2^* - f_3^*\|_2 & \|f_2^* - f_4^*\|_2 \\
\vdots & \vdots & 0 & \|f_3^* - f_4^*\|_2 \\
\vdots & \vdots & \vdots & 0
\end{pmatrix} \rightarrow \text{save 6 values for } U, I \text{ and } R$$
1.a Compression: Which features are extracted?

D. Subtract normalized signals with normalized reference signal and look if they are out of a certain envelope

\[
|U_i'(t) - U_{i,ref}'(t)| < C_U e^{-\frac{t}{\tau}}; \quad i = 1,2,3,4 \quad \rightarrow \text{save}\% \text{ for which this is true}
\]

\[
|I_i'(t) - I_{i,ref}'(t)| < C_I e^{-\frac{t}{\tau}}; \quad i = 1,2,3,4 \quad \rightarrow \text{save}\% \text{ for which this is true}
\]

\[
|R_i'(t) - R_{i,ref}'(t)| < C_R e^{-\frac{t}{\tau}}; \quad i = 1,2,3,4 \quad \rightarrow \text{save}\% \text{ for which this is true}
\]

![Graphs showing feature extraction](image)
2. Classification

a. Classification: Threshold based features classification to ✔ and ✗
b. Comparison: Compare to LabVIEW classification and check differences with experts
c. Extension: Extend existing classification methods
2.a Classification: Threshold based signal classification

1. Difference across single component

- \( f_{\text{first}} = \text{mean(medianf(data[0:19], w = 3))} \)
- \( f_{\text{last}} = \text{mean(medianf(data[-20:-1], w = 3))} \)

Discharge within range?

- \( 780\text{V} < \text{first} < 980\text{V} \)
- \( 15\text{V} < \text{last} < 70\text{V} \)

Decay if:

- \( \text{last} - \text{first} > 20\text{V} \)

\( \text{NO Decay} \)

Similarity:

\( \| f_i' - f_j' \|_2 \)

2. Difference to other component (reference)

- Curve2Curve comparison:
  \[
  |U_i'(t) - U_{i,\text{ref}}'(t)| < C_U e^{-\frac{t_i}{\tau}} ;
  |I_i'(t) - I_{i,\text{ref}}'(t)| < C_I e^{-\frac{t_i}{\tau}} ;
  |R_i'(t) - R_{i,\text{ref}}'(t)| < C_R e^{-\frac{t_i}{\tau}} ;
  \]

\( \tau \) within range?

- \( \tau_i - \tau_{\text{ref},i} < \pm 3\text{ms} ; i = 1,2,3,4 \)

\( \text{OK} \) \( \times \) \( \text{NOT_OK} \)
2.b Comparison: Classify difference manually

- Voltage
- Current
- Resistance

→ Analysed Signal
→ Reference Signal
→ Analysed Signal - Reference Signal
2.b Comparison: Overview

For all 7140 PM entries from 2014 to 2018:

- Classification deviates ~0.6% from LabVIEW classification:
  - Different similarity measures
  - Critical NOT_OKs have all been detected by both methods →
  - Classification difference checked manually:
    - Manually labelled database contains 3130 OK, 116 NOT_OK and 3894 NO_Discharge
    - LabVIEW: 99.75% intersection with manually labelled database
      - Wrong classified signals: 1 OK, 17 NOT_OK*
    - My classification: 99.54% intersection with manually labelled database
      - Wrong classified signals: 24 OK, 9 NOT_OK

→ Next: comparison of features across time and circuits

*LabVIEW feature calculation got currently updated with signal normalization before comparison as well, and should therefore perform even better in the future.
2.c Extension: Comparison of CT calculation

<table>
<thead>
<tr>
<th>Approach</th>
<th>Explanation</th>
<th>Calculation Time / signal</th>
<th>Distribution [ms]*</th>
<th>Number of Outliers**</th>
</tr>
</thead>
<tbody>
<tr>
<td>Charge approach</td>
<td>$\int_{\theta_0}^{\theta_1} f(t) d\theta = \int_{\theta_0}^{\theta_1} f_0(t) e^{-\frac{t}{\tau}} d\theta$</td>
<td>0.01s</td>
<td>80.7 ± 46.3</td>
<td>40</td>
</tr>
<tr>
<td>Energy approach</td>
<td>$\int_{\theta_0}^{\theta_1} f^2(t) d\theta = \int_{\theta_0}^{\theta_1} f_0^2(t) e^{-\frac{t}{\tau}} d\theta$</td>
<td>0.01s</td>
<td>79.2 ± 12.6</td>
<td>40</td>
</tr>
<tr>
<td>Linear regression</td>
<td>$\min_{p}(f(t) - (p_0 + p_1x))$</td>
<td>0.05s</td>
<td>96.7 ± 6.9</td>
<td>2</td>
</tr>
<tr>
<td>Exponential fit</td>
<td>$\min_{p}(f(t) - p_0 e^{-p_2(t-p_1)})$</td>
<td>0.33s</td>
<td>89.5 ± 7.0</td>
<td>2</td>
</tr>
<tr>
<td>Change in characteristic time</td>
<td>$\frac{f(t)}{f(t)} = -\frac{f_0(t) e^{-\frac{t}{\tau}}}{f_0(t) e^{-\frac{t}{\tau}}}$</td>
<td>0.01s</td>
<td>93.9 ± 11.3</td>
<td>2</td>
</tr>
</tbody>
</table>

*Mean value of U&I Signals < 1s
**Considering 3246 PM entries with discharges > 1s
2.c Extension: Different ways for classification

Currently used thresholds:

<table>
<thead>
<tr>
<th>Threshold</th>
<th>Range threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Graph" /></td>
<td><img src="image2.png" alt="Graph" /></td>
</tr>
</tbody>
</table>

Threshold is chosen, by experts.

Possible extensions:

Linear Classifier

![Graph](image3.png)

Now we have access to historical data, one could also set the threshold, such that the distance to sample points is minimized.

\[ \text{e.g. } \min_{\omega_1} \| \omega_1 \| \]

Scalable to higher dimensions and higher order decision boundaries.
2.c Extension: Hard classification vs. soft classification

An overlap in the feature distribution can lead to false signal classification

In some of our features there is an overlap in feature distribution

\[ \text{mean}(\tau_i - \tau_{\text{ref}}); i = 1, 2, 3, 4 \]
2.c Extension: classify data using a Gaussian distribution

Instead of assigning a explicit state to the feature, one could also assign a probability to it.

\[ K(x, \mu) = e^{-\frac{(x-\mu)^2}{2\sigma^2}} = e^{-\frac{(0.25-0)^2}{2\times0.5^2}} = 0.882 \quad K(x, \mu) = e^{-\frac{(x-\mu)^2}{2\sigma^2}} = e^{-\frac{(0.25-1)^2}{2\times0.5^2}} = 0.325 \]

\[ P(\checkmark) = \frac{0.882}{0.882 + 0.325} \times 100\% = 73\% \quad P(\times) = \frac{0.325}{0.882 + 0.325} \times 100\% = 27\% \]

→scalable to more features
2.c Extension: Current workflow

In case of a wrong decisions, experts have to manually adjust the thresholds or the feature calculation.
2.c Extension: Workflow with feedback loop

Linear classification makes it possible incorporate expert knowledge into the classification progress

Proof of concept linear classifier:
- Only PM entries with discharges (3246) for classification
- A linear classifier needs data to learn from: Dividing dataset into training and validation set (50%/50%)
2.c Extension: Performance on validation set

- Performance (intersection with manually labelled dataset) on validation set:
  - 99.32% with the LabVIEW classification: 0 OK and 11 NOT_OK wrong classified
  - 98.82% with my classification: 11 OK and 8 NOT_OK wrong classified
  - 98.40% with linear classifier trained on my features*: 10 OK and 16 NOT_OK wrong classified

- Linking my classification and linear classifier with logic &:
  - 4 NOT_OK wrong classified

- With the used linear classifier* also the decision confidence is available
  - Can be used to adjust actions (E.g. send out warning at already 30% confidence)

*Actually a support vector machine with a Gaussian RBF kernel was used. Fine tuning of hyper parameter is still in progress, detailed information will be in my master thesis.
Trend analysis

a. Analysis: Comparison of QH features across time and circuits
3.a Analysis: Comparison across time and circuits

→ Browsing through data with dedicated SWAN-notebooks
3.a Analysis: Comparison across time and circuits

1. Prefiltering: Which PM entries are eligible for prediction?

<table>
<thead>
<tr>
<th>Eligibility</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>All PM entries 2014-2018</td>
<td>7140</td>
</tr>
<tr>
<td>All PM entries 2014-2018 with decay</td>
<td>3246</td>
</tr>
<tr>
<td>Entries with up-to-date reference</td>
<td>1929</td>
</tr>
<tr>
<td>No Hardware Commissioning tests</td>
<td>1906</td>
</tr>
<tr>
<td>Number of Magnets</td>
<td>1232</td>
</tr>
<tr>
<td>Average sample points</td>
<td>1.547078</td>
</tr>
</tbody>
</table>

2. Plotting features as a function of time
Conclusion & Outlook

→ Embedding of LabVIEW QH feature calculation
→ Extension with further features

→ Implementation of LabVIEW QH classification approach
→ Comparison of different feature calculation methods
→ New classification concept to incorporate expert knowledge

→ Browsing through data with dedicated SWAN-notebooks
→ Historical data for both quench heaters and busbar resistance available now

→ The same approach can be used for further applications
## 1. Feature Overview

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Explanation</th>
<th>Parameter Nr. Features/Parameter Nr. Of Features</th>
<th>Upper Threshold</th>
<th>Lower Threshold</th>
<th>Capped at</th>
</tr>
</thead>
<tbody>
<tr>
<td>I_MEAS</td>
<td>I_MEAS</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initial</td>
<td>mean(medianfilter(data[0:19,w=3]))</td>
<td>UIR</td>
<td>4</td>
<td>12</td>
<td>980,None,None 780,None,None</td>
</tr>
<tr>
<td>Last</td>
<td>mean(medianfilter(data[-20:-1,w=3]))</td>
<td>UIR</td>
<td>4</td>
<td>12</td>
<td>70,None,None 15,None,None</td>
</tr>
<tr>
<td>Charge Approach</td>
<td>characteristic time of exp. decay</td>
<td>UI</td>
<td>4</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Energy Approach</td>
<td>characteristic time of exp. decay</td>
<td>UI</td>
<td>4</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>LinReg</td>
<td>characteristic time of exp. decay</td>
<td>UI</td>
<td>4</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>CTime_mean</td>
<td>mean characteristic time of exp. decay</td>
<td>UI</td>
<td>4</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>CTime_std</td>
<td>std characteristic time of exp. decay</td>
<td>UI</td>
<td>4</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Similarity Matrix</td>
<td>euclidian distance within the signals</td>
<td>UIR</td>
<td>6</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>Similarity Matrix_normalized</td>
<td>euclidian distance within the signals normalized</td>
<td>UIR</td>
<td>6</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>Initial_Ref</td>
<td>mean(medianfilter(data[0:19,w=3]))</td>
<td>UIR</td>
<td>4</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>Last_Ref</td>
<td>mean(medianfilter(data[-20:-1,w=3]))</td>
<td>UIR</td>
<td>4</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>Charge Approach_Ref</td>
<td>characteristic time of exp. decay</td>
<td>UI</td>
<td>4</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Energy Approach_Ref</td>
<td>characteristic time of exp. decay</td>
<td>UI</td>
<td>4</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>LinReg_Ref</td>
<td>characteristic time of exp. decay</td>
<td>UI</td>
<td>4</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>CTime_mean_Ref</td>
<td>mean characteristic time of exp. decay</td>
<td>UI</td>
<td>4</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>CTime_std_Ref</td>
<td>std characteristic time of exp. decay</td>
<td>UI</td>
<td>4</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Similarity Matrix_Ref</td>
<td>euclidian distance within the signals</td>
<td>UIR</td>
<td>6</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>Similarity Matrix_normalized_Ref</td>
<td>euclidian distance within the signals normalized</td>
<td>UIR</td>
<td>6</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>C2c</td>
<td>substracted signals have to be within envelope</td>
<td>UI</td>
<td>4</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>C2c_doubleTCEEnvelope</td>
<td>substracted signals have to be within envelope</td>
<td>UIR</td>
<td>4</td>
<td>12</td>
<td>0,0,0</td>
</tr>
<tr>
<td>C2c_doubleTCEEnvelope_normalized</td>
<td>substracted signals have to be within envelope</td>
<td>UIR</td>
<td>4</td>
<td>12</td>
<td>0.1,0.1,1</td>
</tr>
</tbody>
</table>

**Total**: 233
1. Feature Compression

### Feature Compression

<table>
<thead>
<tr>
<th>Feature name</th>
<th>Explanation</th>
<th>Parameter</th>
<th>Nr. Features/Parameter</th>
<th>Nr. Of Features</th>
<th>Upper Threshold</th>
<th>Lower Threshold</th>
<th>Capped at</th>
</tr>
</thead>
<tbody>
<tr>
<td>energyApproach_dif</td>
<td>energyApproach - energyApproach_Ref</td>
<td>UI</td>
<td>4</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cTime_mean_dif</td>
<td>cTime_mean - cTime_mean_Ref</td>
<td>UI</td>
<td>4</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cTime_std_dif</td>
<td>cTime_std - cTime_std_Ref</td>
<td>UI</td>
<td>4</td>
<td>8</td>
<td>1200,100, None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>initial_dif</td>
<td>initial - initial_Ref</td>
<td>UIR</td>
<td>4</td>
<td>12</td>
<td>0.003</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SimilarityMatrix_dif</td>
<td>SimilarityMatrix - SimilarityMatrix_Ref</td>
<td>UIR</td>
<td>6</td>
<td>18</td>
<td>None,None, 20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SimilarityMatrix_normalized_dif</td>
<td>SimilarityMatrix_normalized - SimilarityMatrix_normalized_Ref</td>
<td>UIR</td>
<td>6</td>
<td>18</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td></td>
<td>72</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Feature Comparison

<table>
<thead>
<tr>
<th>Feature name</th>
<th>Explanation</th>
<th>Parameter</th>
<th>Nr. Features/Parameter</th>
<th>Nr. Of Features</th>
<th>Upper Threshold</th>
<th>Lower Threshold</th>
<th>Capped at</th>
</tr>
</thead>
<tbody>
<tr>
<td>energyApproach_dif_mean</td>
<td>mean(abs(energyApproach_dif))</td>
<td>UI</td>
<td>1</td>
<td>2</td>
<td>0.0025,0.0025</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cTime_std_dif_mean</td>
<td>mean(abs(cTime_std_dif))</td>
<td>UI</td>
<td>1</td>
<td>2</td>
<td>0.25,0.0025</td>
<td></td>
<td></td>
</tr>
<tr>
<td>initial_dif_mean</td>
<td>mean(abs(initial_dif))</td>
<td>UIR</td>
<td>1</td>
<td>3</td>
<td>25,25,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c2c_doubleTCEnvelope_mean</td>
<td>subtracted signals have to be within envelope</td>
<td>UIR</td>
<td>1</td>
<td>3</td>
<td>0.1,0.1,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SimilarityMatrix_dif_mean</td>
<td>mean(abs(SimilarityMatrix_dif))</td>
<td>UIR</td>
<td>1</td>
<td>3</td>
<td>1000,150,10</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td></td>
<td>13</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Reason for abs():
- Histograms are almost symmetric
- Values could cancel each other out (e.g. Similarity Matrix)