

# 機械学習手法を用いた XFELの自動調整

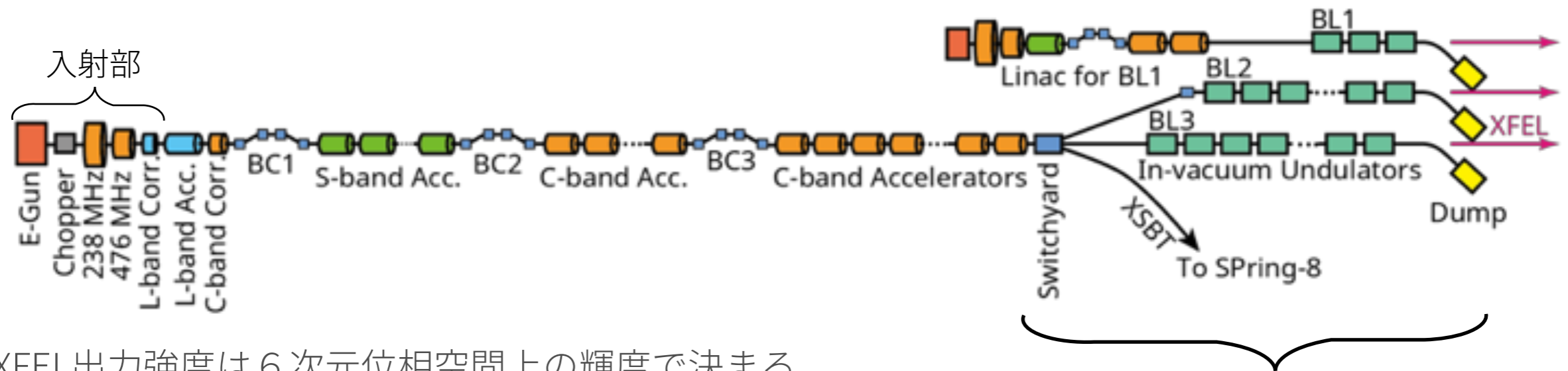
08/11/2021, 第18回日本加速器学会年会

岩井 瑛人

高輝度光科学研究センター/理化学研究所



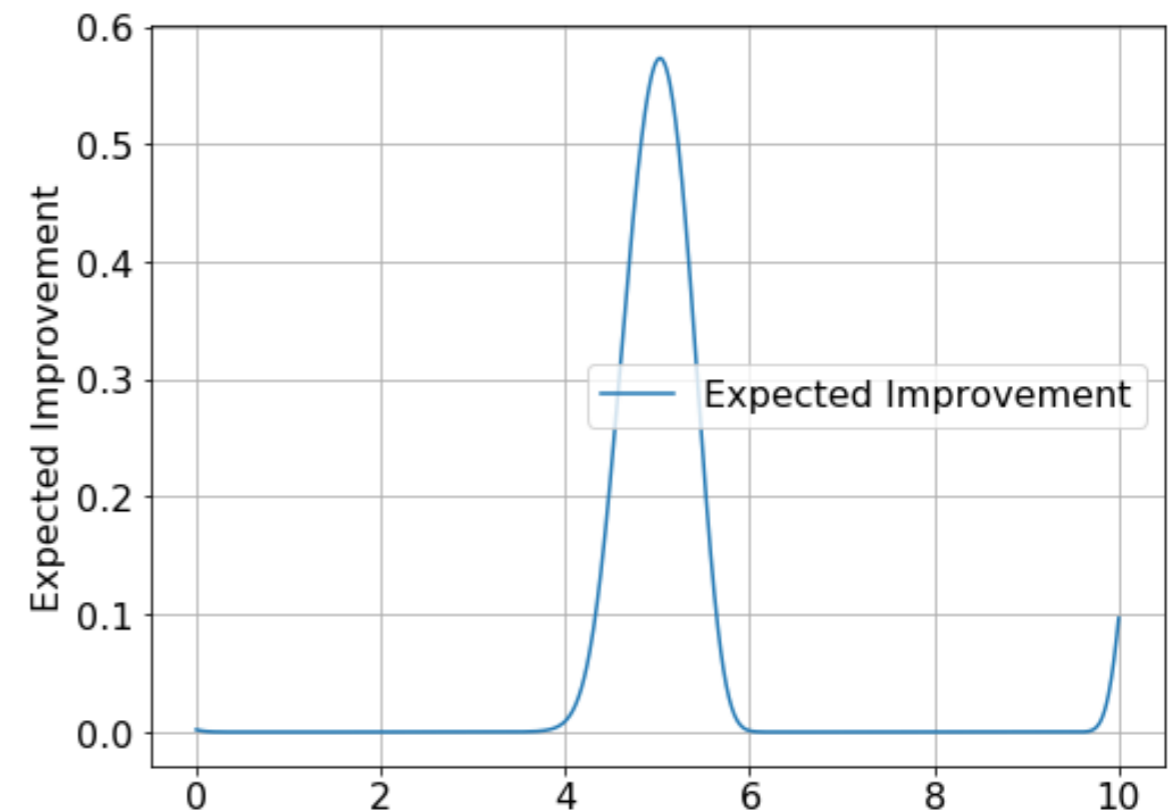
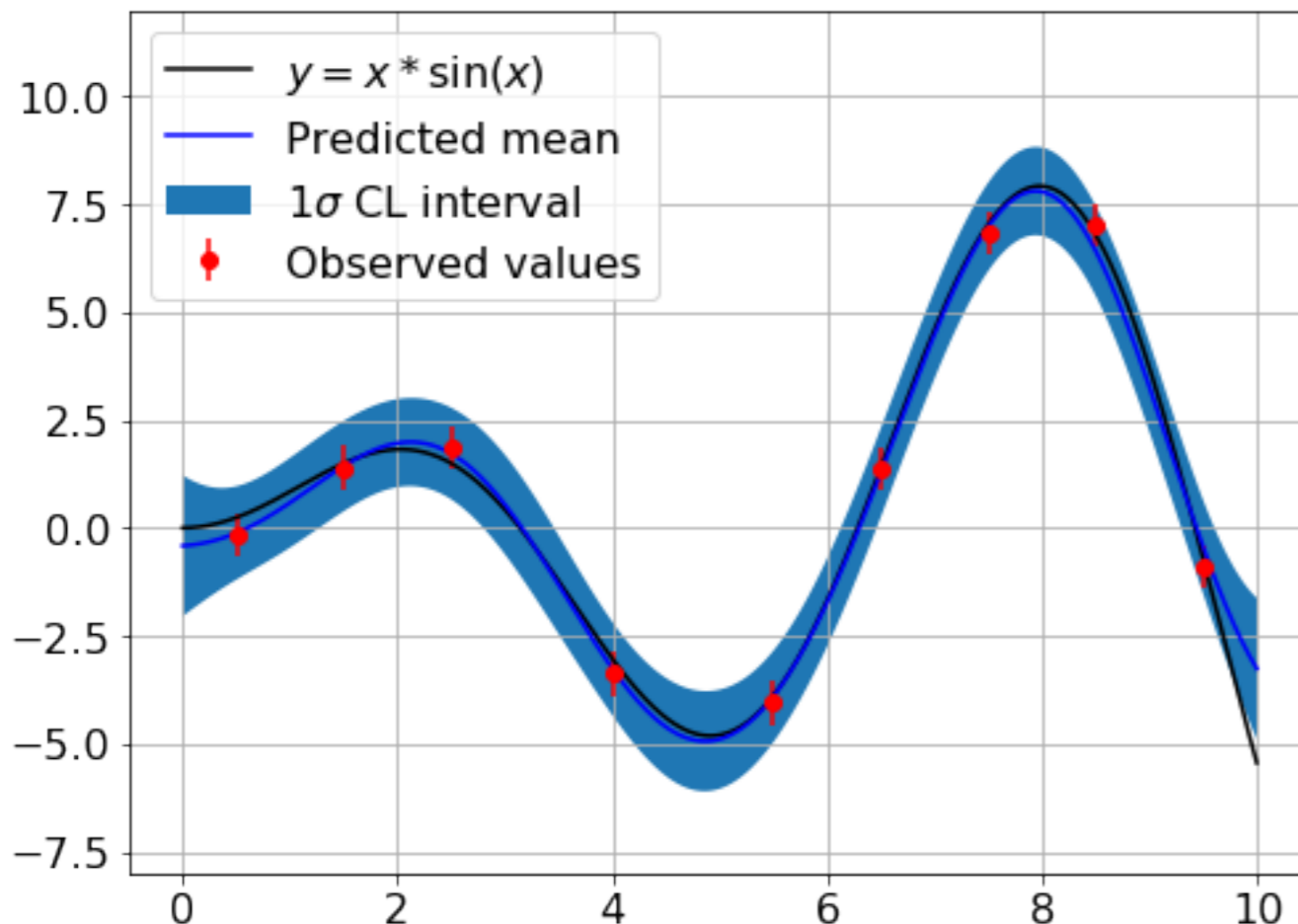
# XFEL/SACLA (→ motivation)



- ▶ XFEL出力強度は6次元位相空間上の輝度で決まる
  - これらを十分な精度でインラインで測る手法は未だ無し
  - SACLAの入射部は熱電子銃+速度圧縮のため均一性が良くはない (熱電子銃: 500 keV, エネルギーと速度 $\beta$ が非線形)
- ◎ “シュリンク”した平均値を用いても計算通りにはいかない (領域共通の課題)
- ➡ どうしてもビームを見ながらの難しい調整が必要になる
- ▶ 調整, 運転の合理化の必要性
  - 波長, ビーム条件が異なる3本のXFELビームライン(BL)の同時運転
    - 数日毎に必要な波長, ビーム条件などが個々に変わる
    - うち2本は主加速部までを共有, パルス毎に切替&振分
    - 2020年2月からSPring-8蓄積リングへの入射器の役割も兼ねる (→ SPring-8-II)
  - ビーム強度, 供給安定性が飽和傾向
    - より良いXFELビーム性能への要求 (パルス幅, 空間プロファイル, 各種レーザー指標の安定性…)

# 機械学習を用いた自動調整

- 機械学習手法の一つ, Gaussian Process Regressor (GPR; ガウス過程回帰) を用いて Optimizer を作成
  - 誤差, 不定性を持つ多次元空間のデータをうまく扱える
  - local min/max にハマりにくい
- Expected Improvement (EI)
  - GPR の中央値と不定性を用いて更新期待値を計算



# GPR Optimizer のテスト(1)

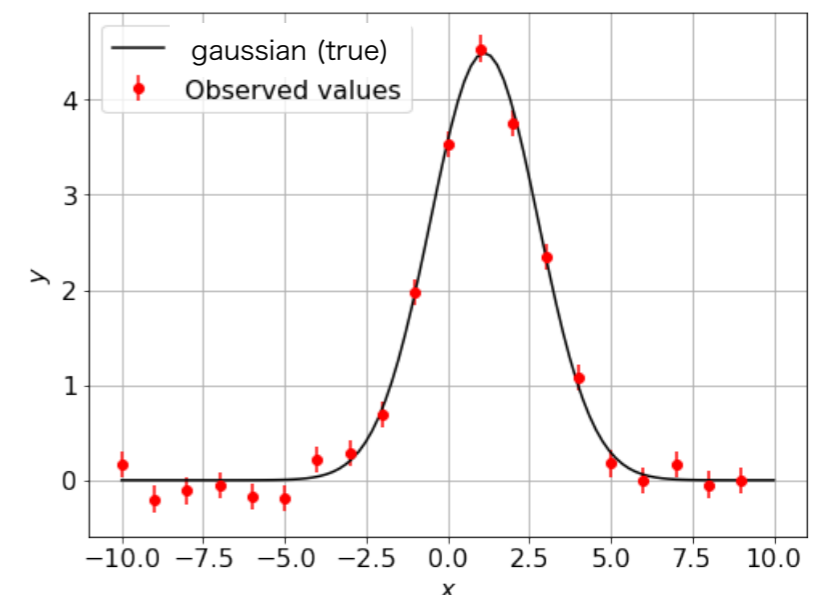
- 最適化シーケンス

1. パラメータ数, range に応じていくつかデータを収集  
(例えば 3 パラメータなら 7 点)
2. それらのデータを元に最初の GPモデルを生成
3. 最適化ループ

- loop
- 3.1. 与えられたパラメータ空間内で最大の EI を与えるパラメータの決定
  - 3.2. 最大 EI のパラメータを適用
  - 3.3. 当該条件でのサンプルデータを追加して, GPモデルをアップデート

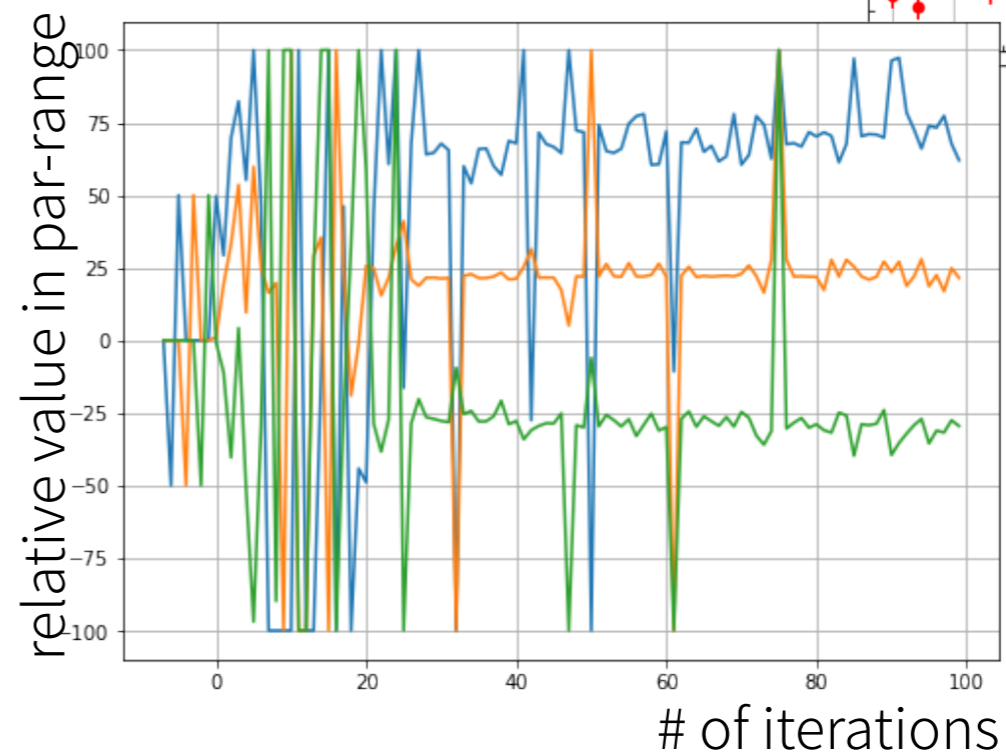
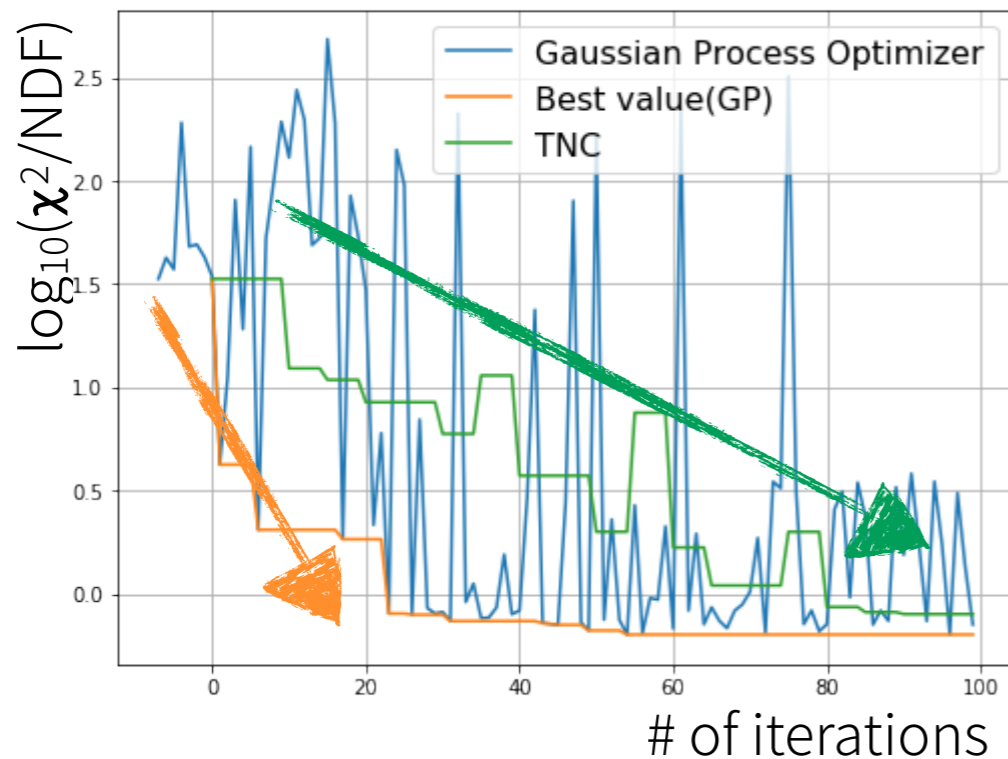
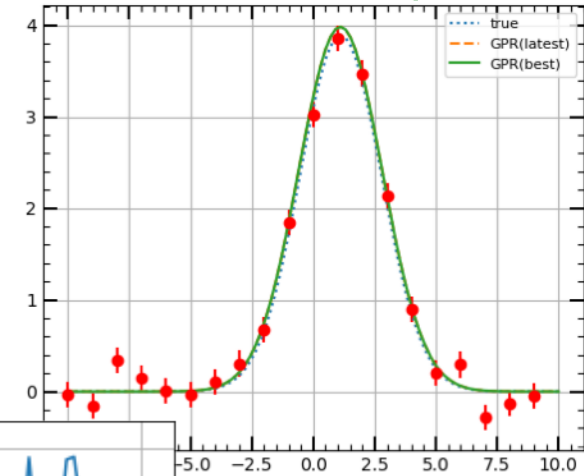
- この GPR-Optimizer を用いて誤差付きデータについてガウスフィットを行う

- パラメータ数: 3
- 目的関数: (reduced) chi-square



# GPR Optimizer のテスト(1) — GPR Optimizer

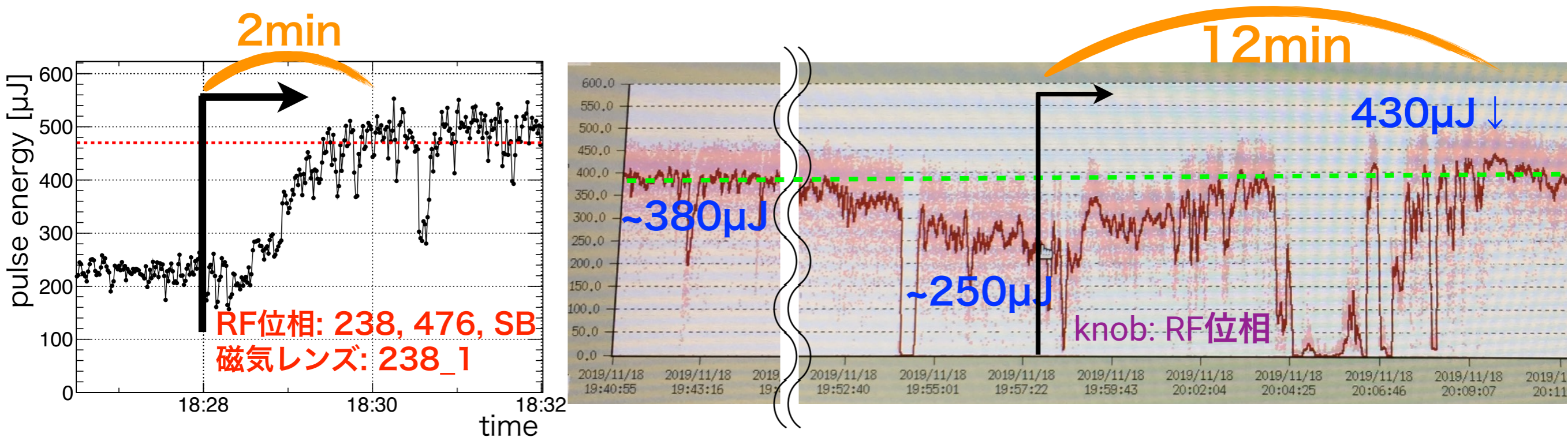
- データのガウスフィットを GPR Optimizer を用いて行う



- 既存/古典的な方法 (TNC; Truncated Neuton) と同等のフィット結果
  - 少ない”試行回数”で収束 (※加速器調整なら調整時間に相当)
  - ‘best’ な値に収束後にも大きなパラメータの”飛び”: GPRの特徴の一つ (local min/max にハマりにくいメリットの反面)

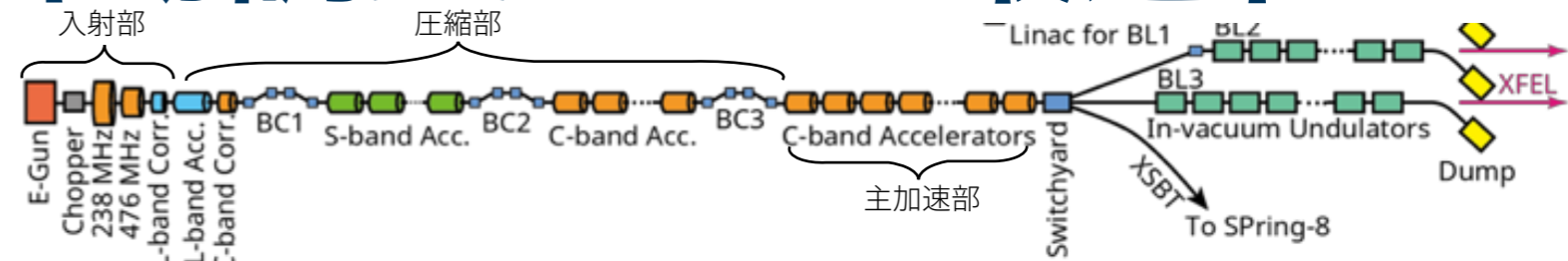
# GPR Optimizer のテスト(2)

- 実際にビームを用いたテスト(通常行われている調整と同様, XFEL出力強度を最大化)
  - 左: まずは少数のパラメータ, 意図的に出力を下げた既知のオフセット
  - 右: 位相ドリフトなど何らかの外乱により低下した出力の回復



- ▶ パラメータ数 ~ 調整”ノブ” の数を徐々に増やす (RF位相, Q/ML/ST磁石電流, アンジュレーターギャップ など)
- ▶ 課題: 計算資源リミットによるパラメータ数の上限 → **solved**
  - ✓ 効率的なコーディング, スマートなアルゴリズム
  - ✓ 専用の計算機 (コア, メモリ)
  - ✓ 他のライブラリも試す: scikit-learn → +BoTorch/GPyTorch

# 通常調整後の自動調整による最適化

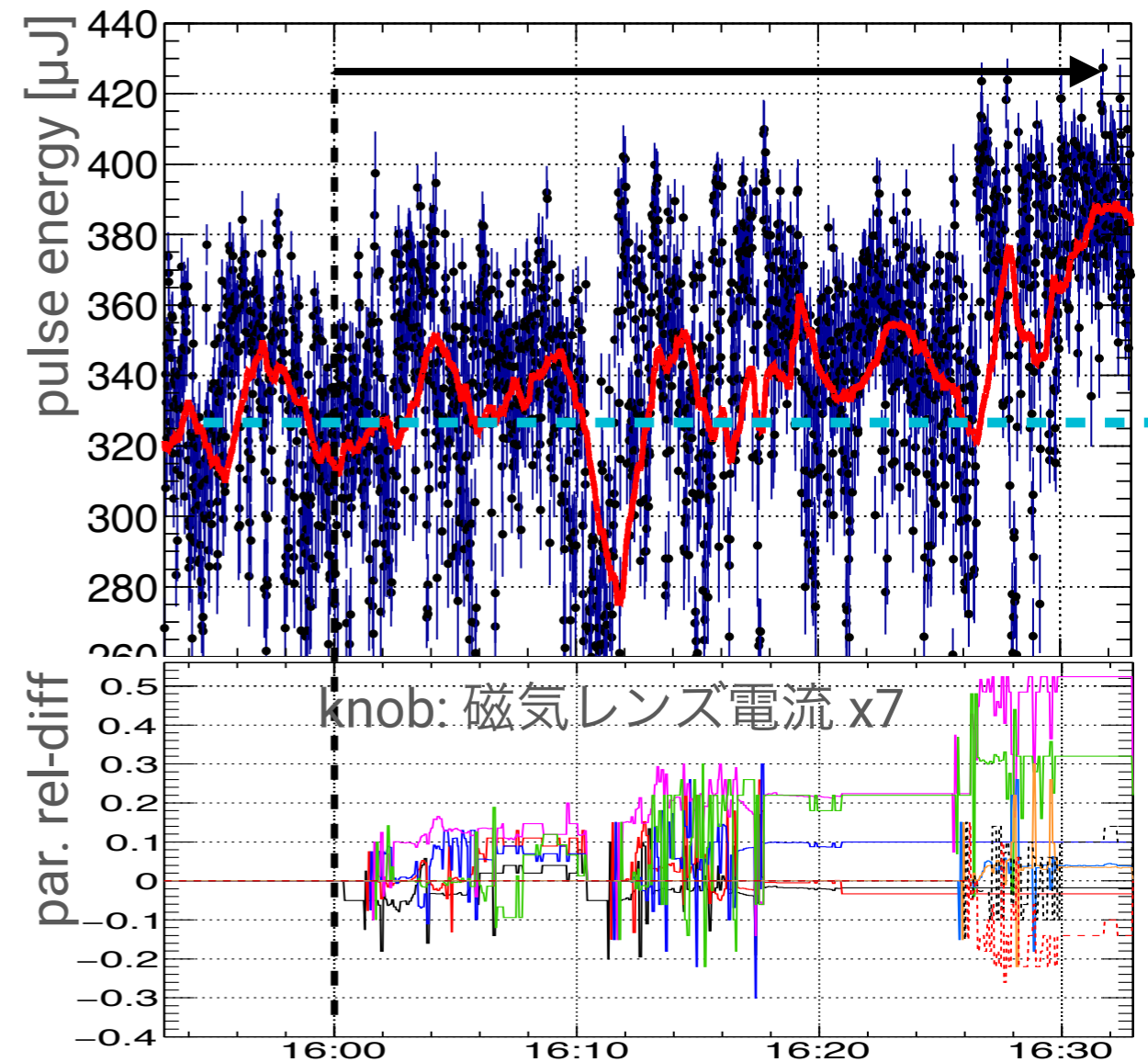
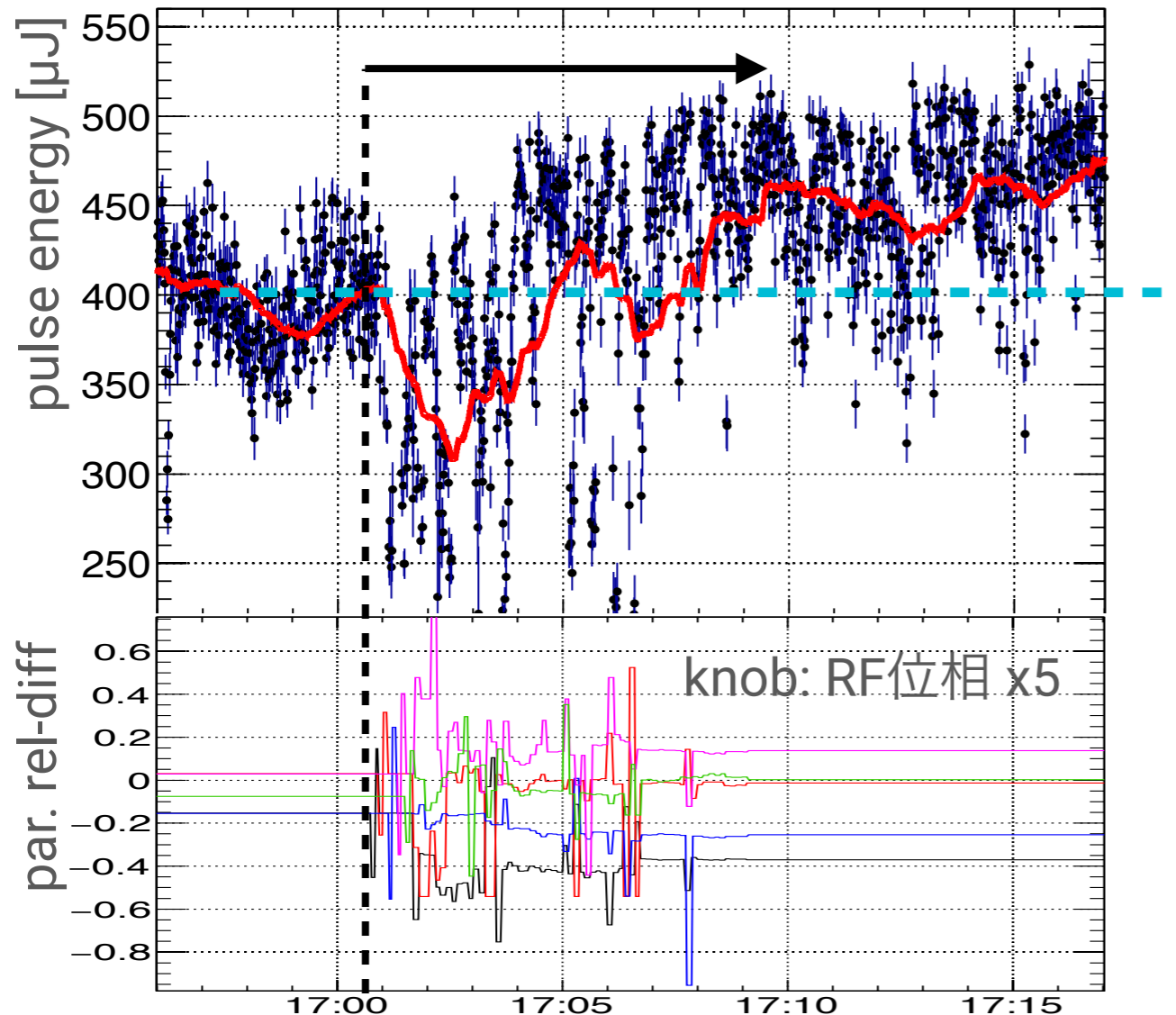


▶ 答えを知らない, 未知の“山頂”を見つけられるか?

✓ **左:** RF 5 位相 ~ バンチ圧縮の調整

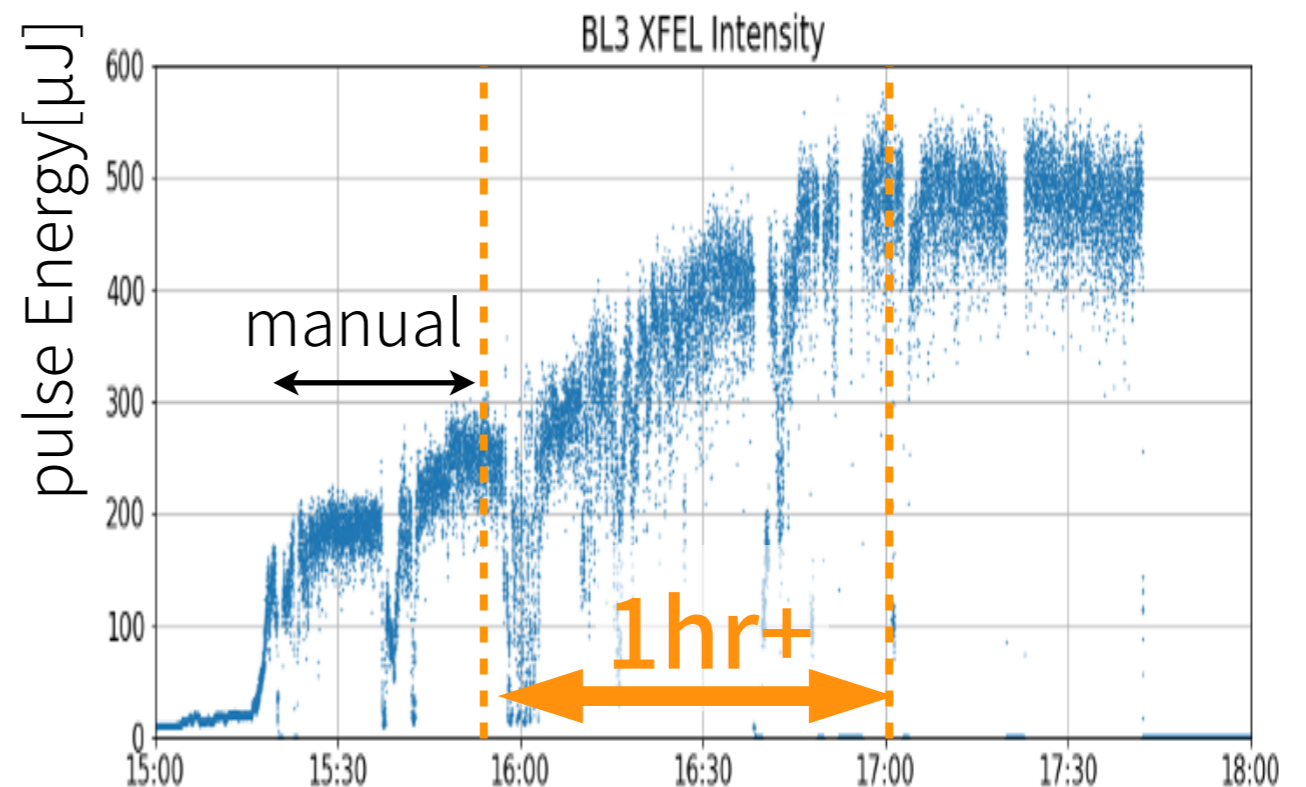
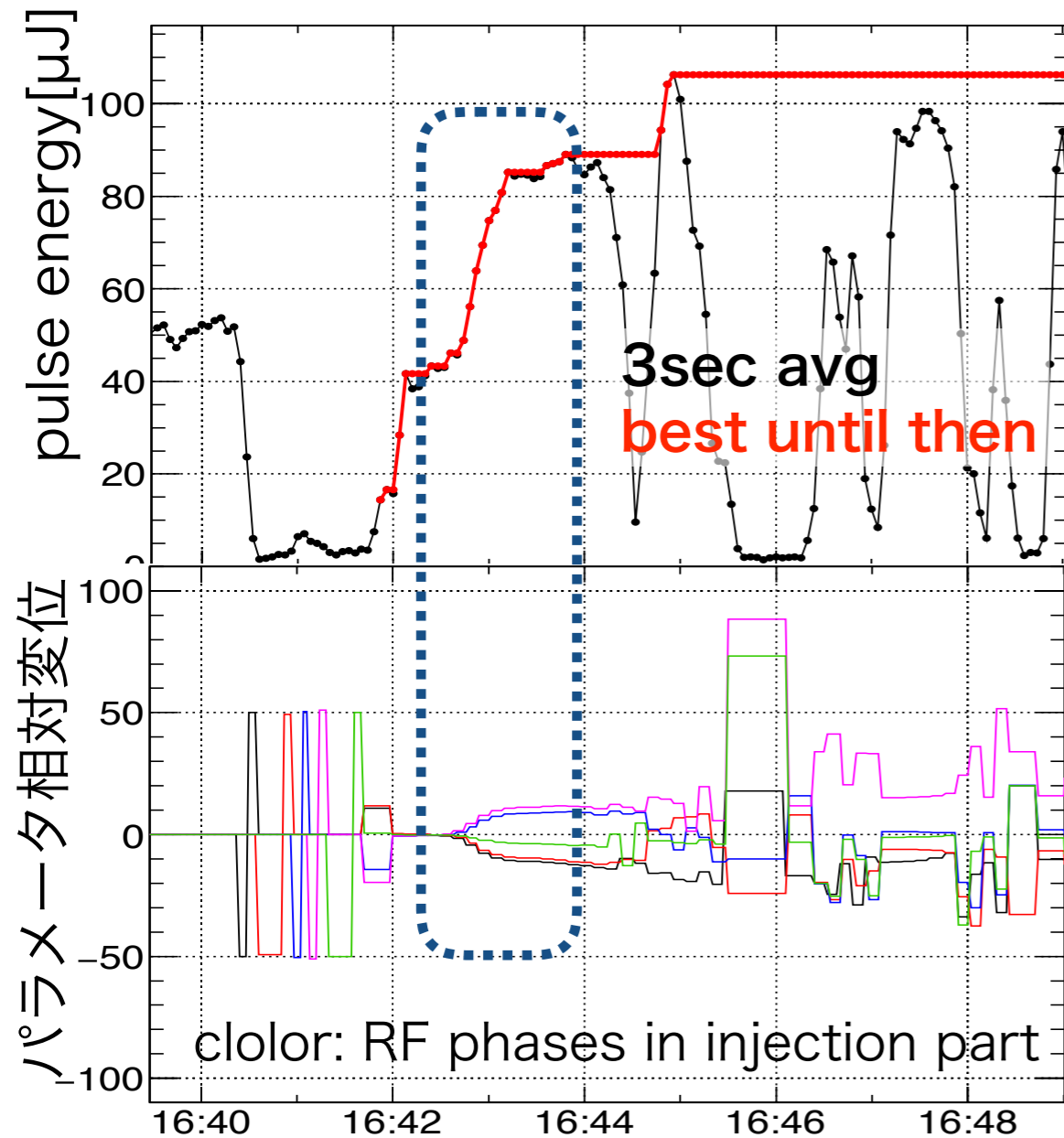
✓ **右:** 入射部磁気レンズ 7 台 (2 BL 共通部) ~ エンベロップの調整

— running avg (30sec)



# SACLA立ち上げ時の調整

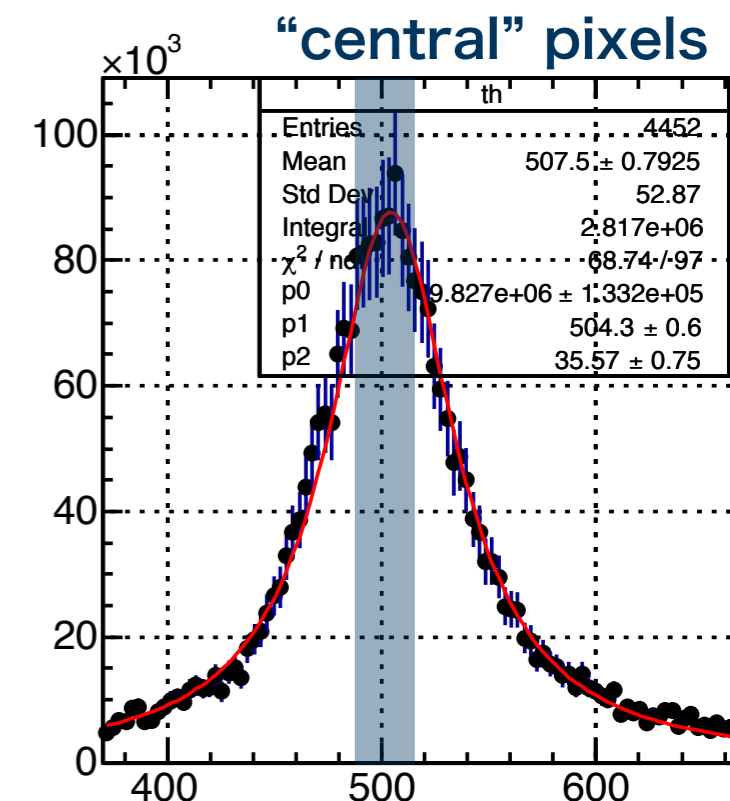
- ▶ **Left:** 電子銃(最上流)のカソード交換後の quick recovery  
- ML-like な振る舞いを観測
- ▶ **Right:** 長期停止後からの運転再開時の調整



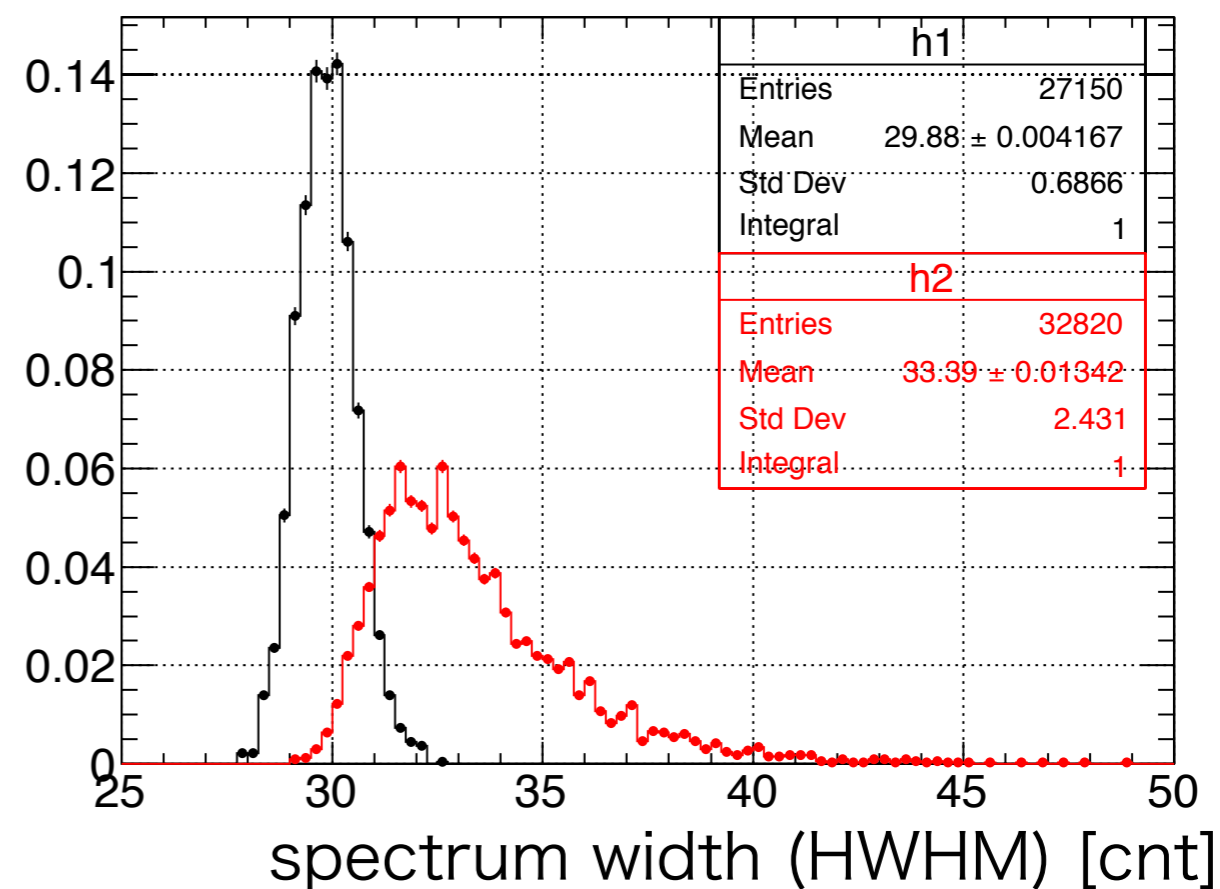
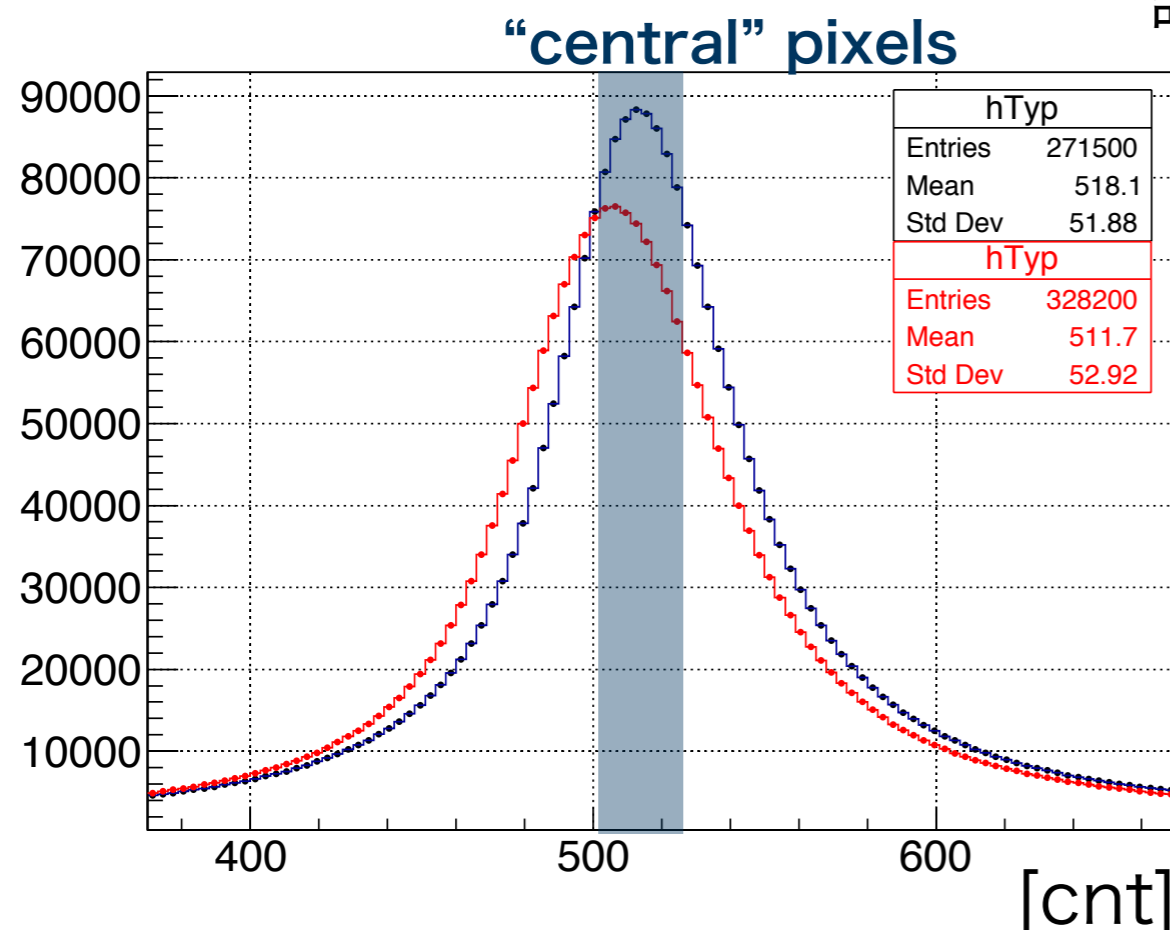


# スペクトル幅を考慮した調整

- スペクトル測定器のデータを指標として用いる
  - **Left:** スペクトルデータの単純平均  
(ピーク波長のパルス毎のフラつきは考慮していない)
  - **Right:** Cauchy でフィットした時のスペクトル幅



調整前  
調整後



# CUIベース → GUI化

- 複雑な Optimizer コア部と GUI/操作部を敢えて切り離して実装
  - 最適化アプリケーション(性能指標, 制御パラメータ)を運転員が任意設定, 開発できる
  - 今後コア部をアップデートしても汎用的に使える
  - Optimizer - GUI 間制御は redis を用いる

目的, セクション毎の  
制御パラメータ・テンプレート

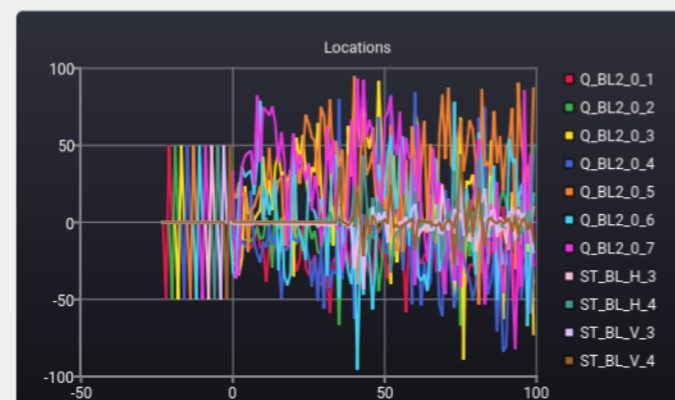
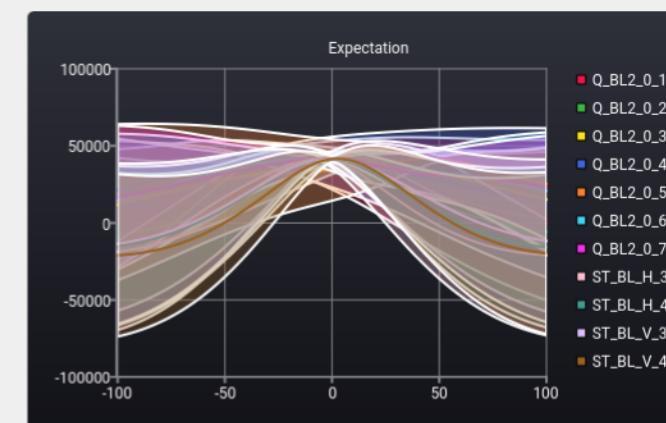
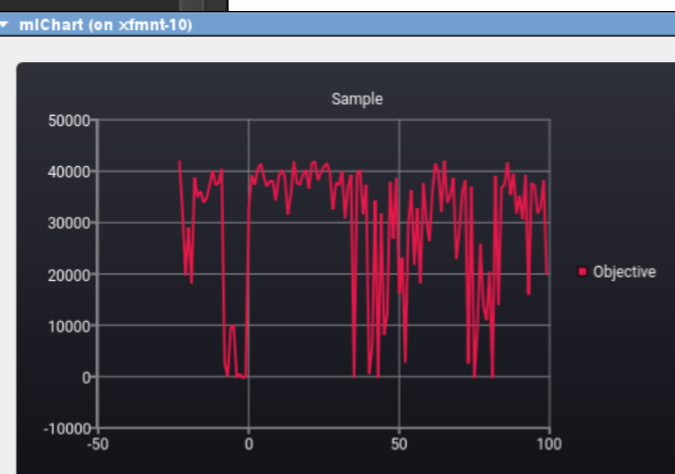
各機器の制御範囲, 待ち時間,  
制御情報などのリスト

The screenshot shows the main GUI window titled 'MainWindow (on xfmnt-10)'. It features a menu bar with 'File' and 'Tools'. Below the menu is an 'Optimization' section with buttons for 'Start', 'Stop max', and 'Stop default'. To the right, there are dropdown menus for 'BL:' (set to 'BL2') and 'Mode:' (set to 'Mode 2'), and a 'BL2 gain:' field set to '1.0'. A table lists optimization parameters with columns for Name, Min, Max, Unit, Wait, and Get. The first row is highlighted in blue.

Name	Min	Max	Unit	Wait	Get
1 Q_BL2_0_1	-0.5000	0.5000	A	2.500	get/xfel_mag_ps_q_bl2_0_1/current_dac
2 Q_BL2_0_2	-0.5000	0.5000	A	2.500	get/xfel_mag_ps_q_bl2_0_2/current_dac
3 Q_BL2_0_3	-0.5000	0.5000	A	2.500	get/xfel_mag_ps_q_bl2_0_3/current_dac
4 Q_BL2_0_4	-0.5000	0.5000	A	2.500	get/xfel_mag_ps_q_bl2_0_4/current_dac
5 Q_BL2_0_5	-0.5000	0.5000	A	2.500	get/xfel_mag_ps_q_bl2_0_5/current_dac
6 Q_BL2_0_6	-0.5000	0.5000	A	2.500	get/xfel_mag_ps_q_bl2_0_6/current_dac
7 Q_BL2_0_7	-0.5000	0.5000	A	2.500	get/xfel_mag_ps_q_bl2_0_7/current_dac
8 ST_BL_H_3	-0.5000	0.5000	A	3.000	get/xfel_mag_ps_st_h_bl3/current_dac
9 ST_BL_H_4	-0.5000	0.5000	A	3.000	get/xfel_mag_ps_st_h_bl4/current_dac

Below the table is a log window showing the following text:

```
2021-04-11T02:40:54 Start optimization.
2021-04-11T02:41:03 BL2_STATUS_NAME::INIT
2021-04-11T02:41:03 BL2_STATUS_NAME::WAIT
2021-04-11T02:41:03 BL2_MESSAGE::[Info] main >> The ML process is ready, initialized with BL_prefix: BL2
2021-04-11T02:41:03 BL2_MESSAGE::[Info] RedisClient >> Subscribe established: BL2_COMMAND
2021-04-11T02:41:09 BL2_COMMAND::RUN
2021-04-11T02:41:09 BL2_STATUS_NAME::RUN
2021-04-11T02:41:09 BL2_MESSAGE::[Info] main >> Starting ML process ...
2021-04-11T02:41:09 BL2_MESSAGE::[Info] main >> Processing noise measurement ...
2021-04-11T02:41:09 BL2_MESSAGE::[Info] SaclaOptimizerInterface >> Apply latest parameters ...
2021-04-11T02:41:09 BL2_STATUS_NAME::WAIT
2021-04-11T02:41:09 BL2_MESSAGE::[Info] myParamFunc >> The ML process is waiting for parameter changes ...
2021-04-11T02:41:09 BL2_MESSAGE::[Info] myObjFuncMode2 >> The ML process is waiting for beam evaluation ...
```



BL2 [dropdown menu] [Reset button]

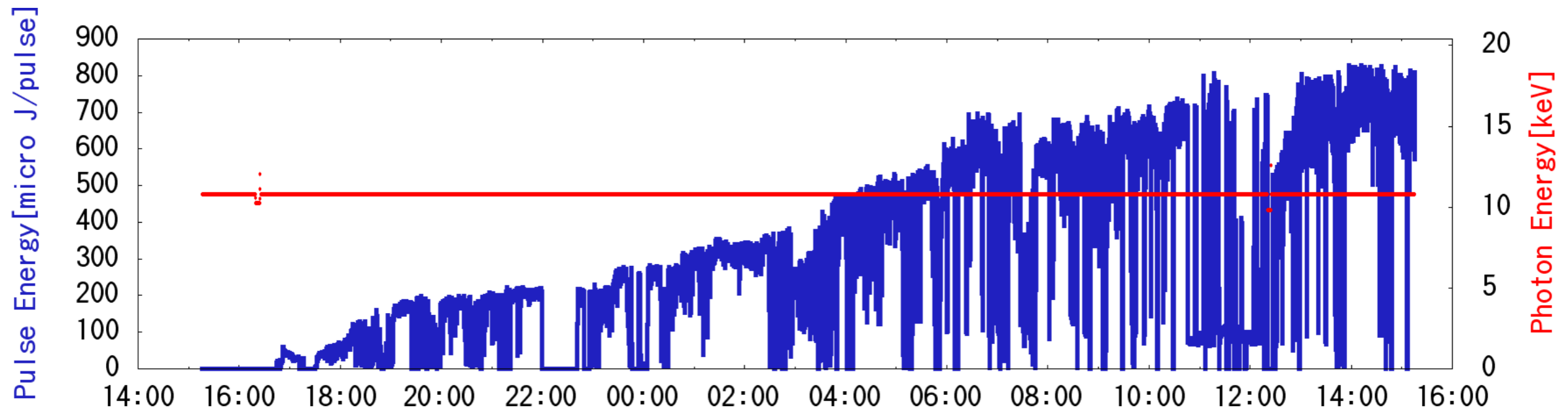
# 実際の加速器調整での利用

## ▶ 運転員によるGUIでの利用

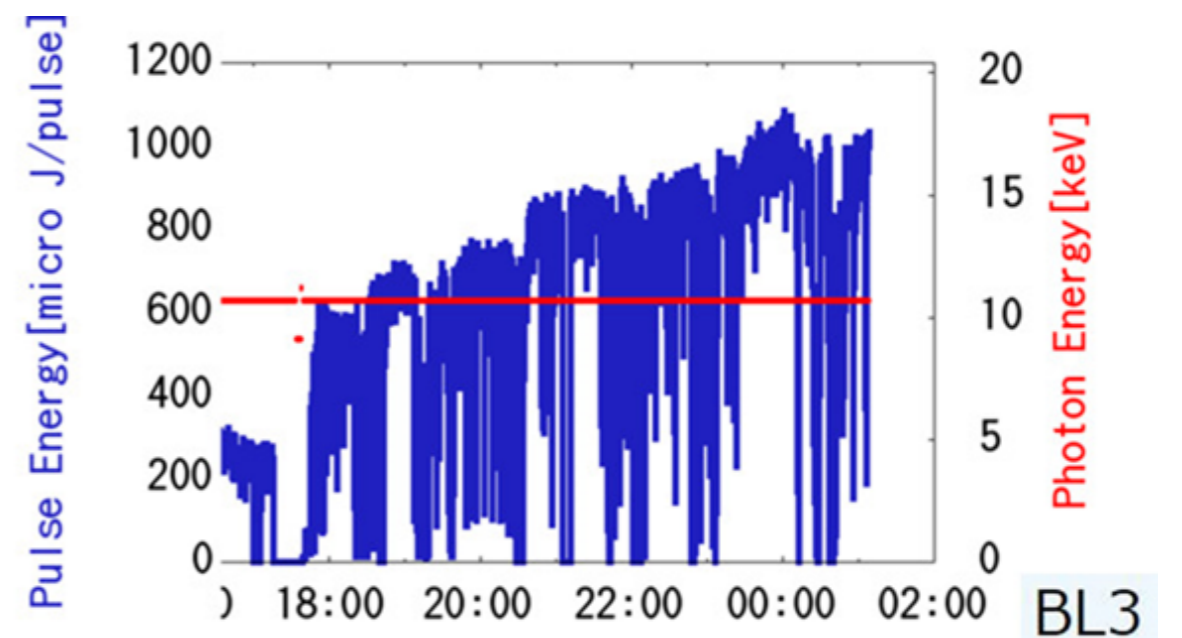
- ML-GUI は 既にも実戦投入されている
- 経験の浅い運転員から、熟練の運転員まで広く日常的に調整に利用されている

18:17 FELを確認

18:17以降、Machine Learningソフトでの調整を朝まで実施する。  
BL2/3 10keV



➡ 未踏の 1mJ@10keV 達成に貢献



**今後: 水平展開と垂直展開**

# Optimizer の実装

- Object指向, 階層型の実装 → 共同開発や将来的な改良を視野
  - 他の用途にも応用できるように、施設/用途毎の差分は基幹/コア部から分離
  - “ML core” 層は DQN などのさらに先進的なコアに置き換えられるように

gpoptimizer  
(w/ plots or GUI)

## “Application”

SACLA, SPring-8,  
XFEL, profile, efficiency,  
loss ... etc.

e.g.) Tune Injector, ID gaps etc.  
for different BLs

SaclaOptimizerInterface

OptimizerInterfaceBase

## “I/O”

another acc. lab,  
another experiment,  
another analysis ...

e.g.) The “Fitting test” was  
performed with the base-class

SaclaGPRegressor

(MLcore)

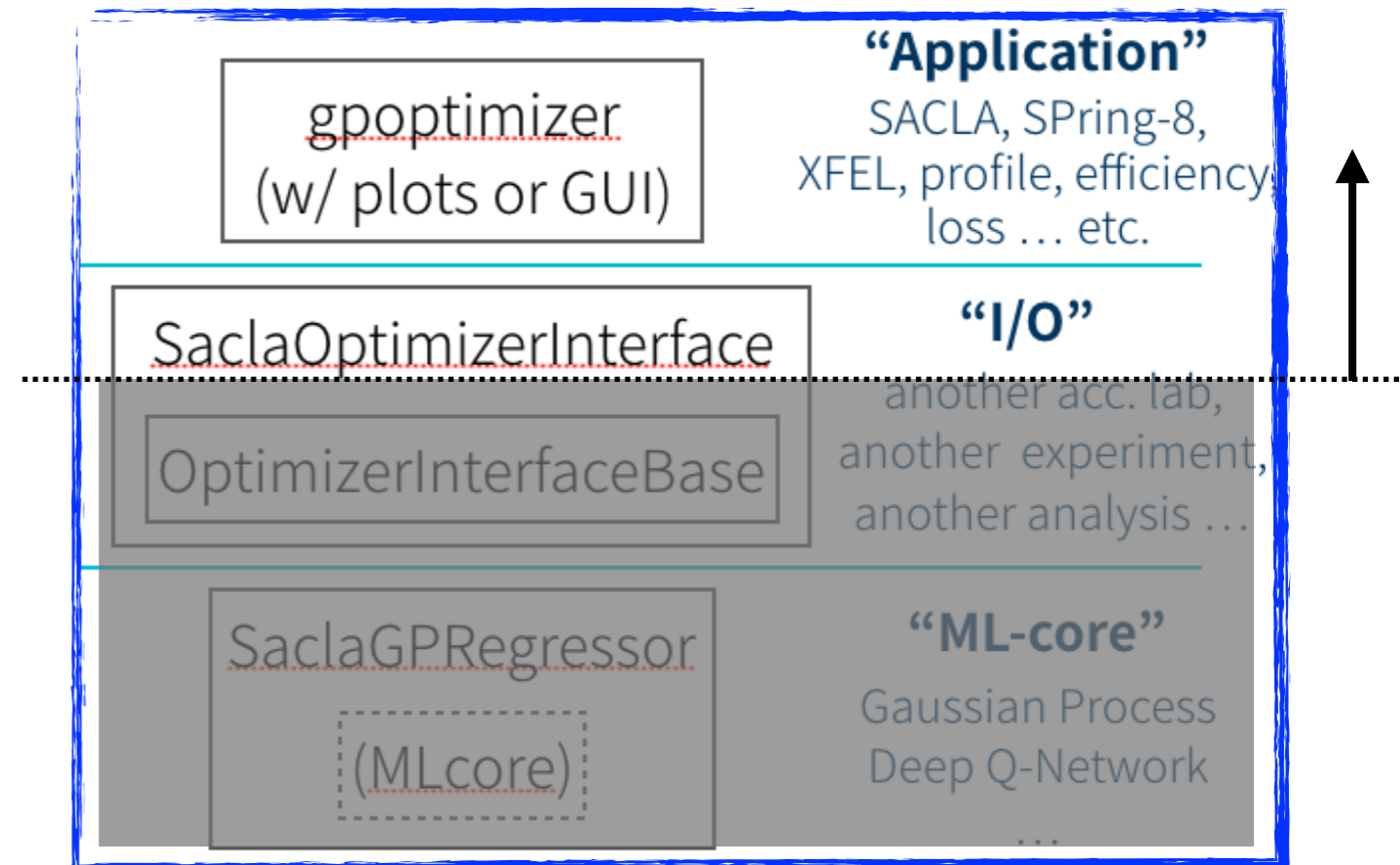
## “ML core”

Gaussian Process  
Deep Q-Network  
...

e.g.) “BoTorch/GPyTorch” version of  
GP-core was coded inherited from  
“scikit-learn” version

# 他施設, 他用途での応用

- RIBF@理研(和光) での応用試験 (理研西氏のトーク, 8/10 TUOA03)
  - 上位レイヤーの差分実装のみで, 自動調整試験ができた  
(互いに Optimizer 実装詳細, 加速器/用途の詳細をブラックボックス化したまま, かなり理想的に接続できた)



✓ 今後, 基幹/コア部 を含めて共同で開発を進める

➡ 他施設, 他用途での応用可能性を実証

# 垂直展開: 今後の開発項目

- 更に多様な用途

- YAG/SM 上の空間プロファイル最適化  
(熱電子銃カソード交換後のビーム調整用途)

- ...

- 更に洗練された性能指標 (BL系の方と協力)

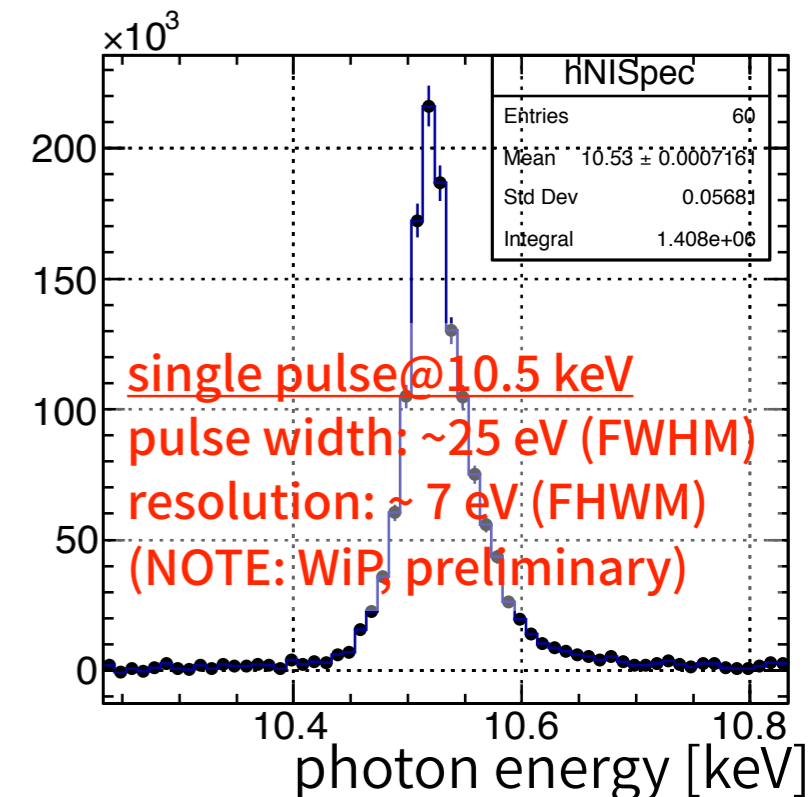
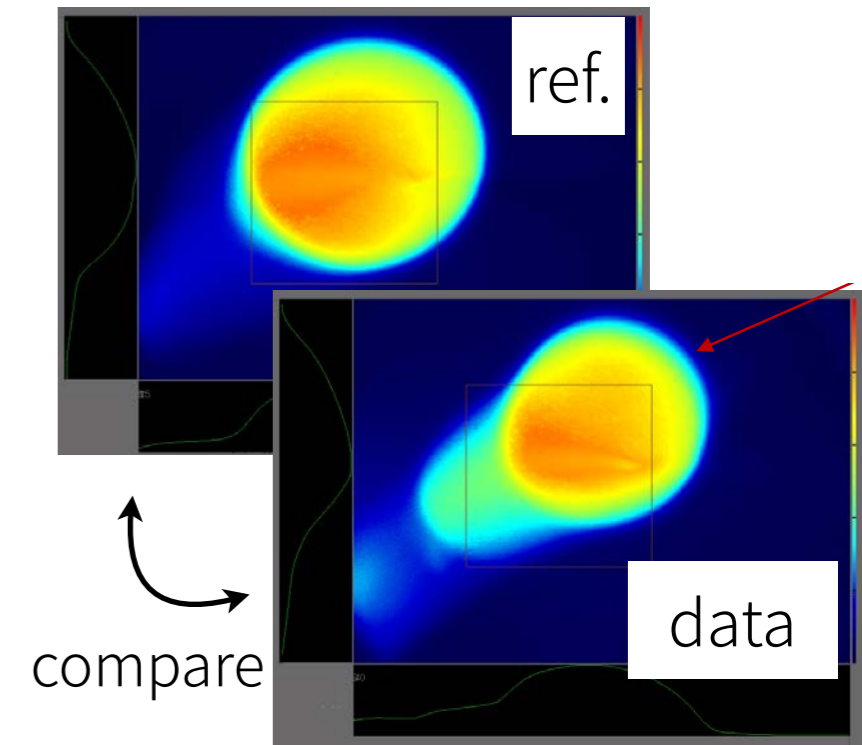
- パルス毎にパルス幅を測定, モニターが行える  
新しい inline spectrometer

- 各ユーザー実験に特化した性能指標

- ...

- 更に高度な機械学習手法の導入

- ...



# まとめ

▶ 難しい調整が必要なXFELの更なる高性能化と運転合理化を目指し、機械学習手法(GPR)を用いた自動調整手法を開発

➡ 実用化：運転員が実際の調整に常用

▶ 今後の展開

- 垂直展開：更なる高度化

- 更に多様な用途

- 更に洗練された性能指標

- 更に高度な機械学習手法の導入

- ➡ GPだけでなく、深層学習, 強化学習, 敵対的生成ネットワークなど, 過去に蓄積した経験を基にした機械学習手法を用いたコア/基幹部の開発

- 水平展開：国内で裾野を広げ, 基幹部の共同開発を狙う

- ➡ RIBF@理研(和光)での応用試験 (8/10 TUOA03)

◎ 新しい分野, 協力して開発を進められないでしょうか?    ご連絡歓迎致します