

ベイズ最適化による cERLビームの自動調整

KEK 高エネルギー加速器研究機構 加速器研究施設
帯名 崇、本田 洋介

前振り

本日の発表内容はベイズ推定の理論的詳細は含みません

- 1つ前の口頭発表（岩井瑛人さん：機械学習手法を用いたXFELの自動調整）と重複する内容が多いだろう、と想像しています
- 詳細はskipしてcERLでの活用例を紹介いたします（綺麗なデータより生々しい雰囲気を）

発表内容は

- いわゆる「やってみた」→「とても役に立った」というだけのお話です

しかしながら、

- 機械学習に興味はある。加速器での具体的な応用例を知りたい
- 日々のビーム調整に使えるのか、具体的にどうすれば良いのか知りたい。

という人にとって、本日の発表が何かのきっかけになれば幸いです

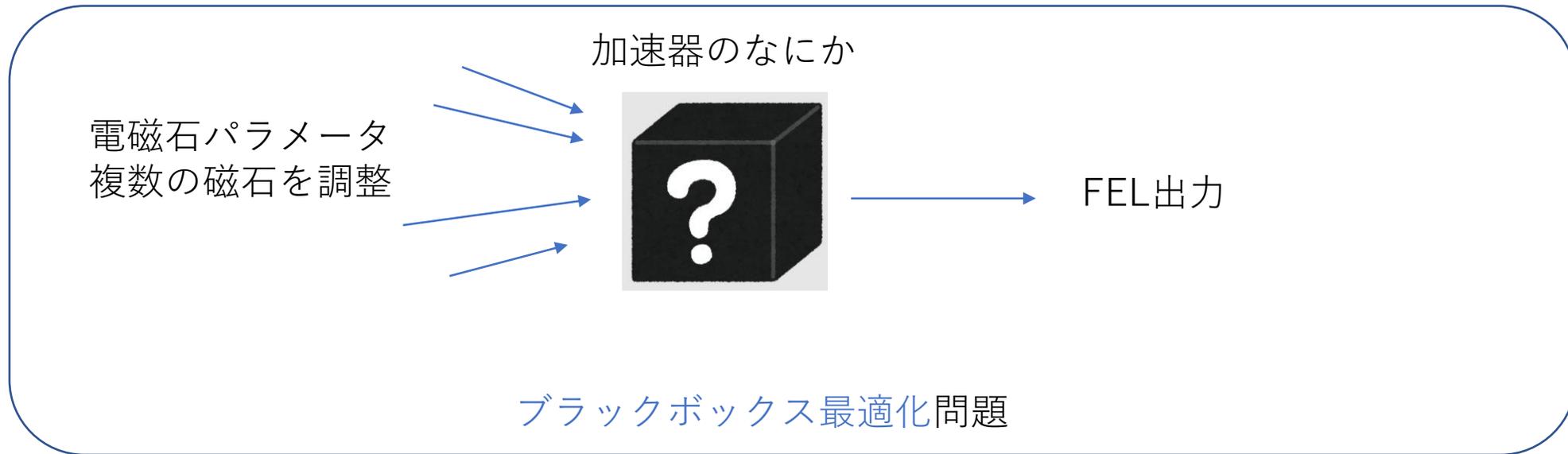
TOC

1. 前振り
2. ベイズ最適化（ガウス過程）の説明と加速器への応用例
 - 原理
 - ツールの紹介
 - 加速器への応用例
3. cERLの説明 & 機械学習が必要となった理由
4. cERLでの適用例
 - a. FELパワーの最大化
 - b. 4極電磁石の磁場中心通し
 - c. ビームサイズ調整
5. まとめ

ビーム調整の例

何かの調整パラメータを振って応答を最大化することを考える【モデルの知識無しで】

- 例えば、電磁石の値を振ってFEL強度を最大化（最小化）など



最小値を探索するには

✓ 全空間探索

- グリッドサーチ
- ランダムサーチ

✓ 効率的に探索する手法

- 最急降下法（確率的勾配降下法）
- ネルダーミード法（= ダウンヒルシンプレックス法）
- その他、多くの手法がある

- 完全に探索できる
- × 時間がかかる。探索空間が広すぎる。

- (比較的)速やかに最小を見つけることができる
- △ 初期値によっては極小に陥る場合あり

✓ 大量のデータを使った機械学習（特にDeep Learning）的手法：大流行中

データを集める + 学習するまで多くの時間を要する。

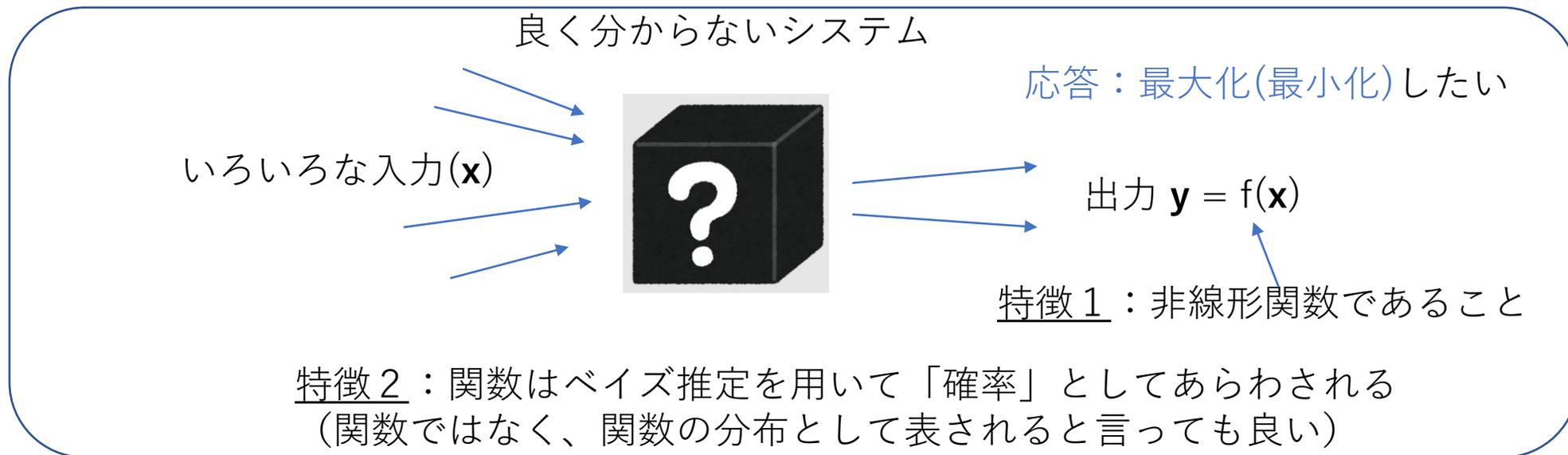
既存データがある場合でも一般的には「良質なデータ」が必要だったり、過学習の問題がある

※ simulationデータを使って学習して実機に適用する例もあるが現実との差異などが問題

 我々は機械学習のなかでも**ベイズ最適化（ガウス過程）**を使用したビーム調整法を採用

ベイズ最適化(Bayesian optimization)の話

人間が調整してきたことをプログラムで自動的にやらせる目的には良い手法
対象システムを「ブラックボックス」として取り扱う



ベイズ最適化は、少ない試行回数(標本個数) と最低限度の仮定 (カーネルの選択) から、確率的にブラックボックスを推定する手法。

- 1回の測定にコスト (お金、時間など) を要する場合や、探索空間が広すぎるときの指針を与えるときに威力を発揮する
- ベイズ最適化の1つであるガウス過程 (GP-EI) は極小に陥りにくい

ツールの紹介：GPyOptを使う例

【制御セッションなので参考のため記載しておく】

これで全部。モジュールのインストールもpipで簡単。意外と簡単に試せます。

```
import GPy
import GPyOpt
```

```
def targ_func(x):
    val = np.cos(2.3*x) + 0.5*x
    return val
```

} ブラックボックス関数：実際にはビームでの応答測定など

```
# __main__
bounds = [{'name': 'x', 'type': 'continuous', 'domain': (0,10)}]
```

type: 連続問題
domain: $0 < x < 10$ の範囲指定

```
# ベイズ推定用のオブジェクト作成
```

```
myBopt = GPyOpt.methods.BayesianOptimization(f=targ_func,
                                             domain=bounds,
                                             initial_design_numdata=5,
                                             acquisition_type='EI')
```

最初のランダム測定の数

```
# 最適化実行
```

```
myBopt.run_optimization(max_iter=15)
```

```
# 結果表示
```

```
print(myBopt.x_opt, myBopt.fx_opt)
```

```
# プロット
```

```
myBopt.plot_acquisition()
```

いい感じにプロットしてくれる

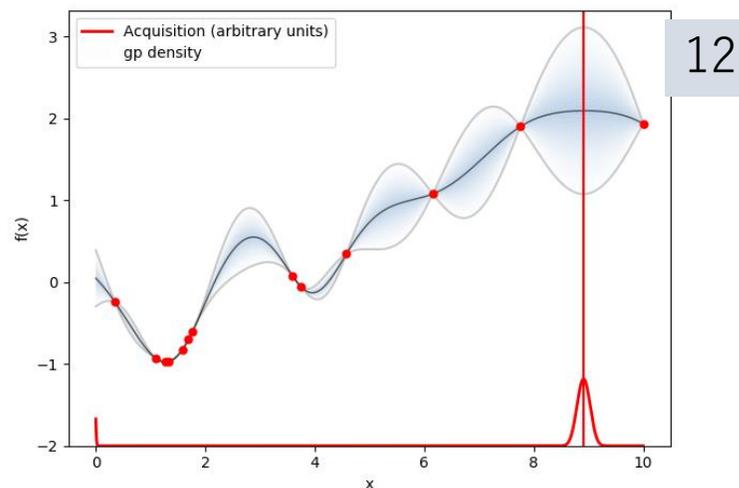
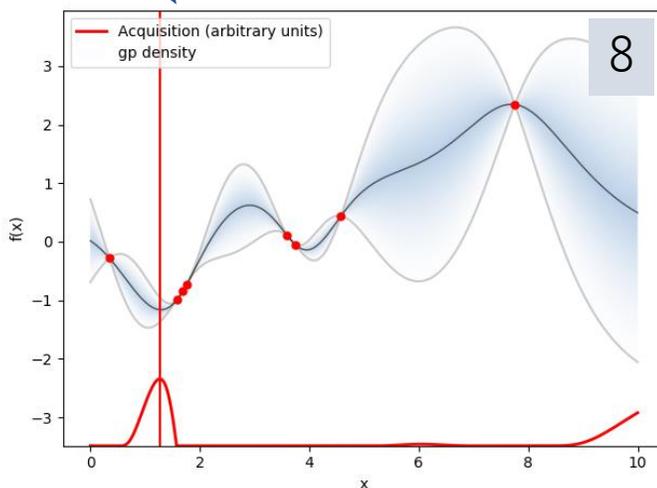
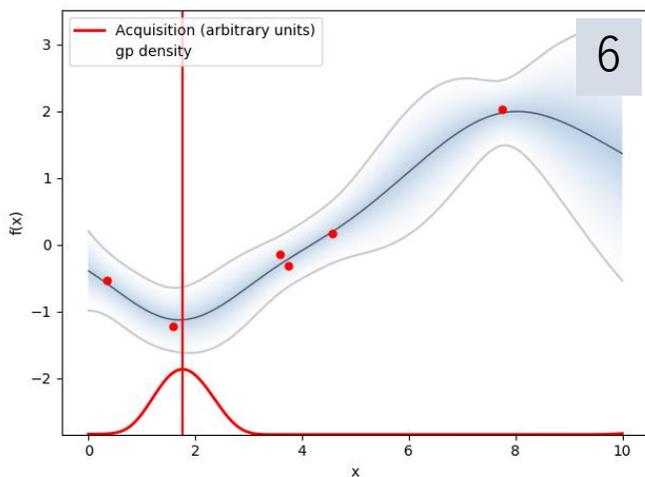
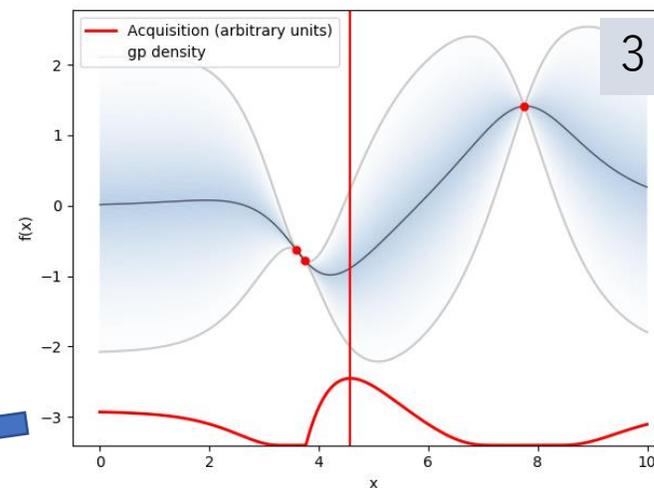
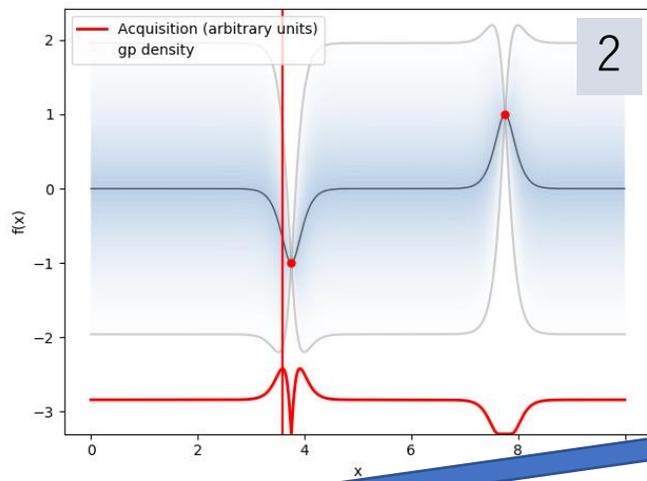
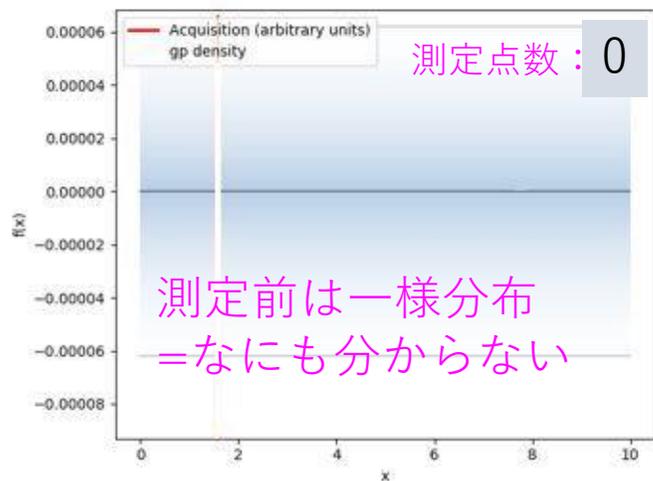
```
myBopt.plot_convergence()
```

獲得関数（最適化手法）
EI, MPI, LCBなどが指定できる

MPI: Maximum Probability Improvement
EI: Expected Improvement
LCB: Lower Confidence Bound

前頁の例を実行 ($0 < x < 10$ の区間で最小値探索)

- 赤色の点が測定した場所、測定値を縦軸にとり、推定された確率分布関数を示す。
- 赤色の縦線が「次に測定すべき場所」 = 最小付近を探しつつ + 不確定性 (幅) を小さく



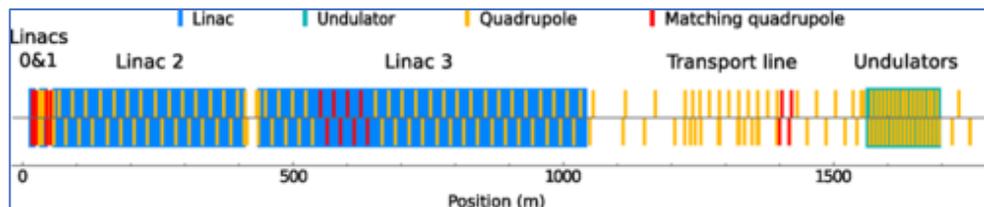
加速器での実用例

FEL調整で役立っているガウス過程とベイズ推定

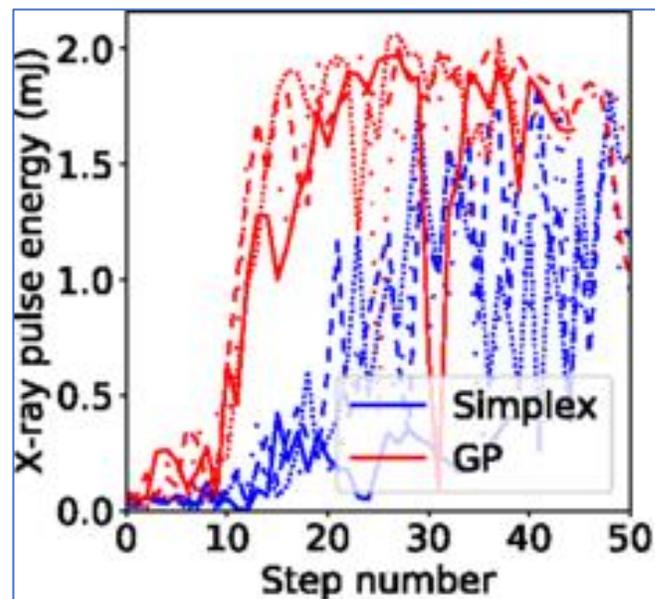
- SACLA → 岩井さんの発表
- LCLS2 : <https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.124.124801>

“Bayesian Optimization of a Free-Electron Laser”

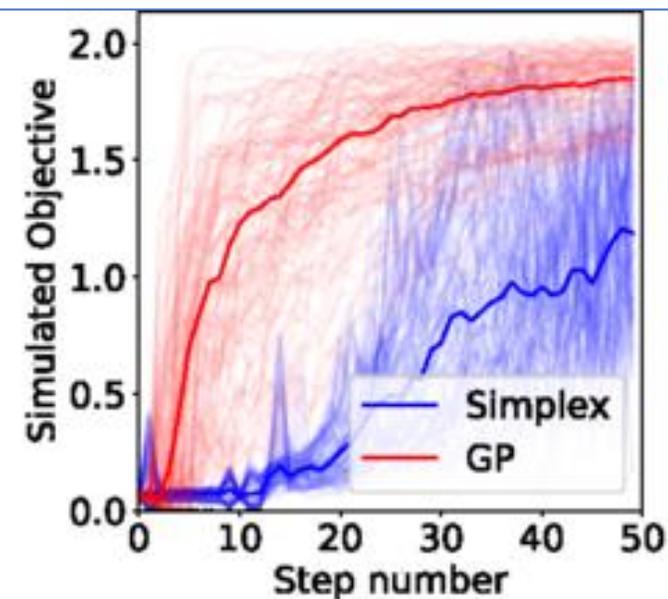
Gaussian Process Regressor (ガウス過程による回帰)



非線形応答の最適化
比較的少ない試行回数で最適解に到達



(a) Live 12 quadrupole optimization



(b) Simulated 12 quadrupole optimization

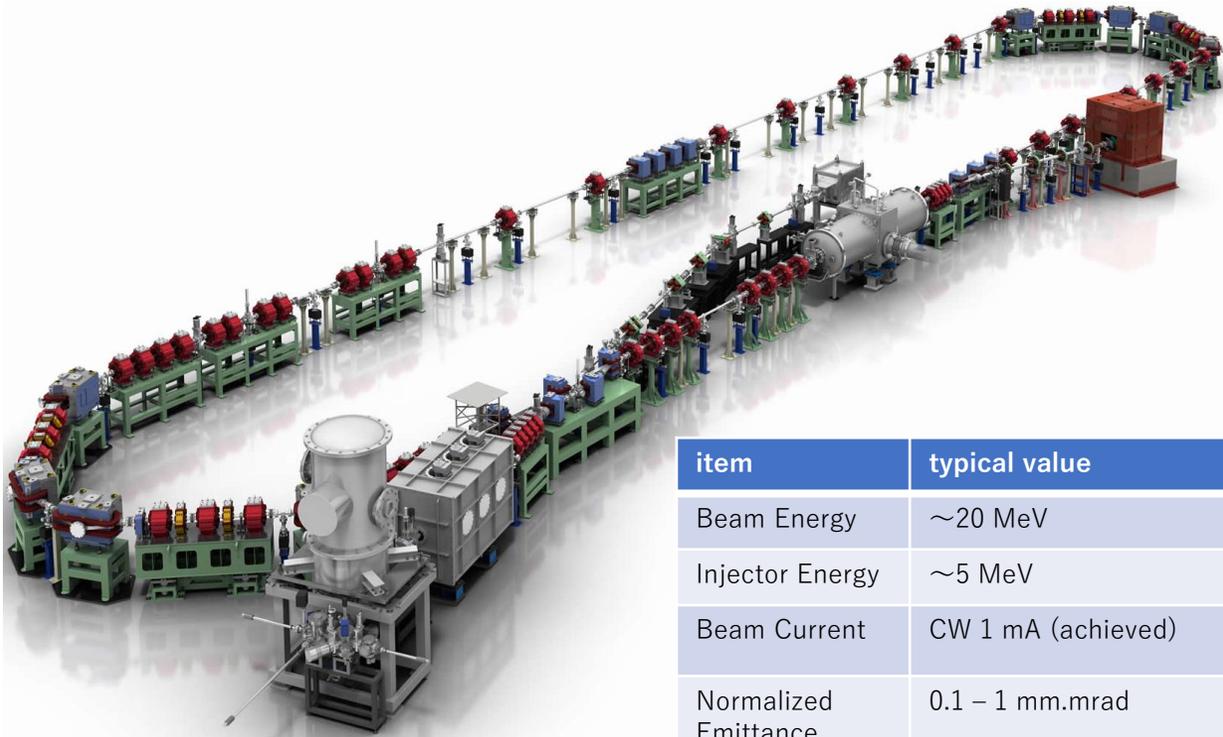
TOC

1. 前振り
2. ベイズ最適化（ガウス過程）のおさらい
 - 原理
 - ツールの紹介
 - 加速器への応用例
3. cERLの説明 & 機械学習が必要となった理由
4. cERLでの適用例
 - a. FELパワーの最大化
 - b. 4極電磁石の磁場中心通し
 - c. ビームサイズ調整
5. まとめ

4. cERLでの適用例を3つ紹介

- a. FELパワー最大化
- b. Q-Magnet磁場中心通し
- c. ビームサイズ調整

cERLの詳細については、初日の合同セッションで阪井さんが報告しています【MOPOA04】

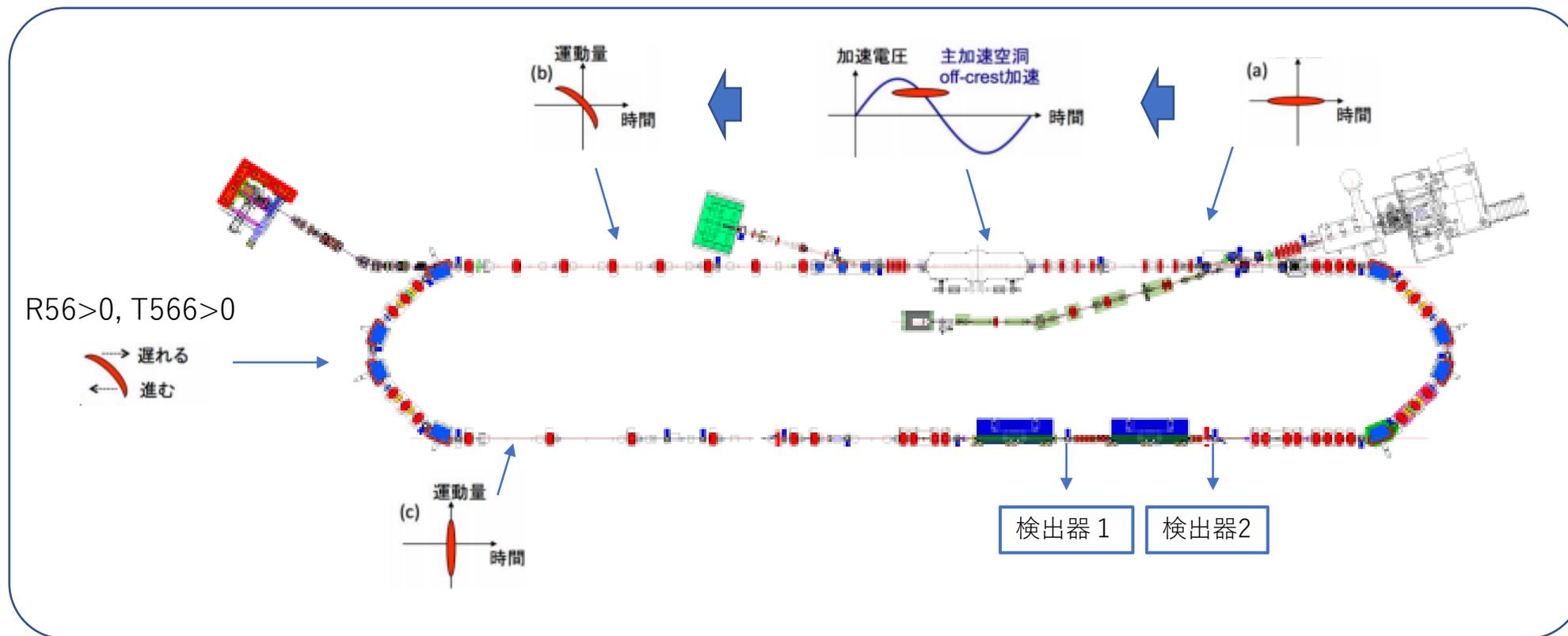


item	typical value
Beam Energy	~20 MeV
Injector Energy	~5 MeV
Beam Current	CW 1 mA (achieved)
Normalized Emittance	0.1 – 1 mm.mrad
Bunch Length	1 – 3 ps (typical) 100 fs (compressed)



4a) FELパワー最大化調整

基本はバンチ圧縮。加速勾配の肩にのせてエネルギーチャープを付ける→アークで圧縮



低エネルギーマシン (~20 MeV) ということもあり、ビーム調整は色々大変：
Space Charge(横・縦) / Velocity Bunching / 外部磁場 (地磁気、KEKB、etc) / 磁石のヒステリシス

機械学習(ML)による自動調整を導入した目的

- ビーム条件を変更するたびにFEL強度を最高値にもっていくように調整する必要がある
- その調整項目は多岐にわたる
 - Energy, R56, Dispersion, 挿入光源での軌道やビームサイズ・焦点など
 - ビーム条件を変えるたびに、おおよそ1~2シフトを要する作業
 - 手動調整では調整する「人」に依存してしまう && 人手は限られている



機械学習による自動調整を導入することで

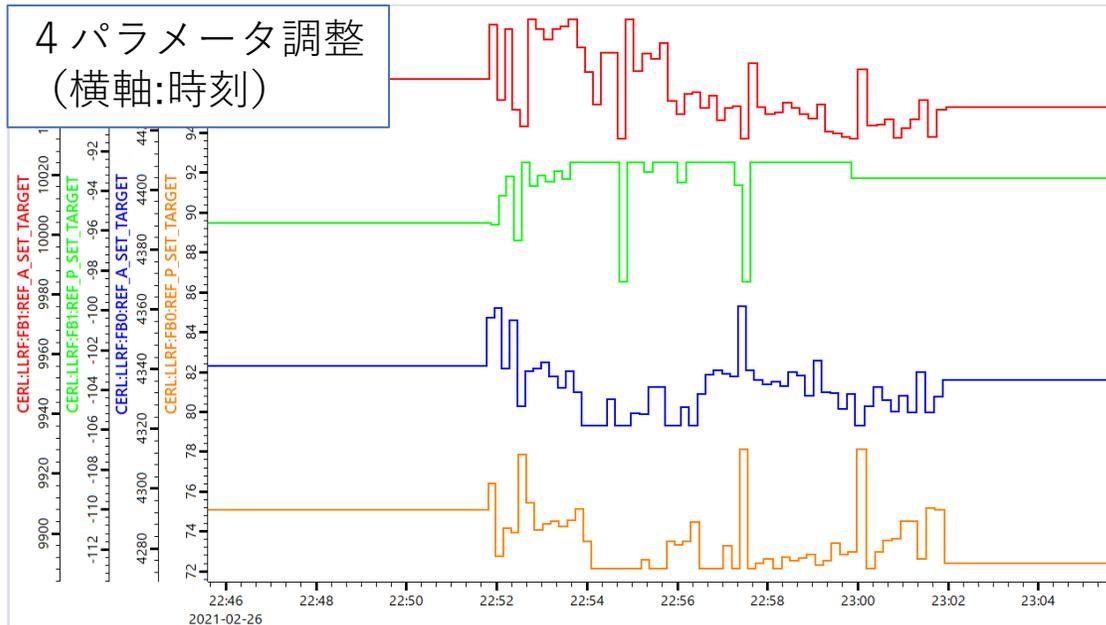
- 誰でも
- 簡単に
- 短時間に
- より高度な調整ができることを目指した

- ※ 計算機が自動調整しているあいだ、人間は「頭」を使って戦略を考えることができる
- ※ 限られた運転時間の中、何を優先するべきなのかを考えるのは人間の仕事

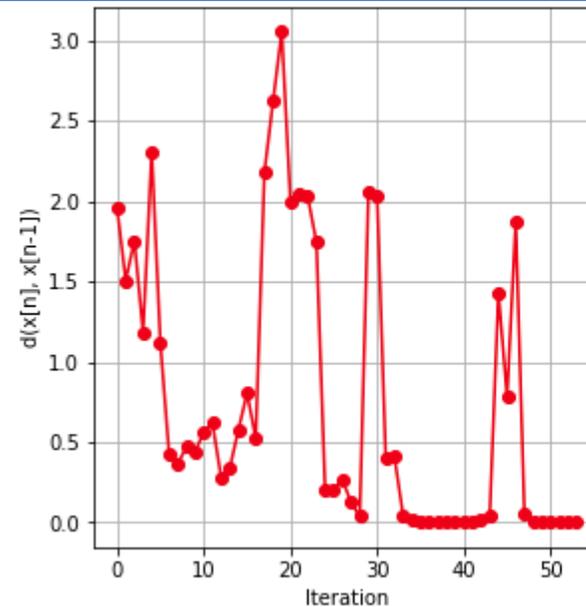
調整の様子：4パラメータを振っている例

- 赤外光はHgCdTe検出器(MCT)の電圧で測定
 - ショットごとのばらつきがあるため、1回の設定パラメータに対して出力を平均している
- 右図：横軸が試行回数,縦軸が評価値BEST（小さいときがFEL最大）→24回で最適解発見
- FEL最大を探しつつ、極大でないことを時々確認するような動き

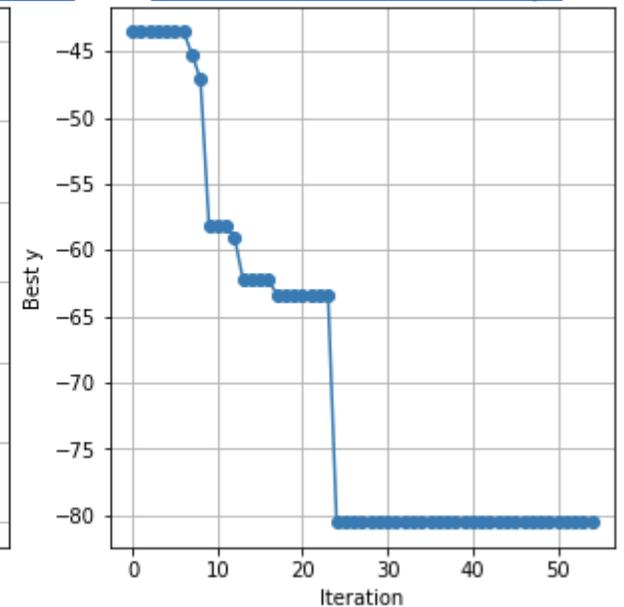
4パラメータ調整
(横軸:時刻)



隣接する測定点の4次元空間内での距離



評価関数のBEST値



課題：磁石のヒステリシスをどう扱うか？

→ 現在は調整速度優先でヒステリシスを考慮していない（設定誤差と同義）

GPによる調整レシピ例

調整手順/レシピを作成した（試行錯誤しながら決めている）。

調整手順: 新レシピ

* U1 を見ながら R56 (手動)

21C 詳細 → * U1 を見ながら R56 + SX (AI)

* U1 Steering (V/H 4 個)

All Screen →

U1 用 → * U1 を見ながら マッチング (QMIS01-05)

パラメータセツ

* U1-U2 間の中心通し

U2 用 → * U2 を見ながら マッチング (QMIS07-10)

パラメータセツ

* U2 を見ながら R56

* U2 Steering (V/H 4 個)

All Screen ? * U2 を見ながら FB0, FB1

* U2 を見ながら R56

* U2 を見ながら チャープ角スキャン (FB5) → Low gain でやる

* U2 を見ながら R56

* U2 を見ながら Steering (Horizontal 4 個)

* U2 を見ながら シケイン ^{Vertical}

* シケイン ON で R56 をやってみるのは良いかも

ビーム調整全体の基本思想は

1. U1を最大化してからU2
2. Transverse (電磁石 Q/Steering)
3. Longitudinal (R56など)
4. RF関連 (ML/Inj/Buncher)

とした。この戦略は人間が策定。

※詳細はMOP029(本田)

←に記載するような調整項目について、それぞれ機械学習(GP)で自動調整

各ステップでは探索範囲内に最大化できたポイントがあればよし、とするのが基本パターン。範囲の端まで行った場合はさらに探索範囲を拡大する手順とした。

調整プログラム：誰でも使えるようにするために

- 主に本田さんが準備した。
- 運転しながら効果的な組み合わせを変えたり、増やしたりするなど調整した
- Jupyter Notebook + GPyOpt で記述：Webインターフェースを活用

組み合わせノブ

- 組合せノブ係数セットツール(テンプレートをコピーして使ってください) ⇨ <http://erlserv5.cerl.kek.jp:26005/tree/OP>
- アーク部のR56調整
 - ARC1 Matching パネル を使う。
 - ⇨ http://erlserv5.cerl.kek.jp:26005/notebooks/OP/Combined_Knob/Arc1
- アーク部6極
 - Six Matching/パネルを 使う。
 - ⇨ http://erlserv5.cerl.kek.jp:26005/notebooks/OP/Combined_Knob/Arc1
- 北直線部最後を使ってアーク中央を調整するやつ
 - NorthQ Matching パネル を使う。
 - ⇨ http://erlserv5.cerl.kek.jp:26005/notebooks/OP/Combined_Knob/Nort
- 北直線部最後を使ってアーク中央を調整するやつ
 - NorthQ Matching パネル を使う。単純に個別の電磁石で。
 - ⇨ http://erlserv5.cerl.kek.jp:26005/notebooks/OP/Combined_Knob/Nort
- U2ステアリング
 - QMIS07,10で位置と角度
 - ⇨ http://erlserv5.cerl.kek.jp:26005/notebooks/OP/Combined_Knob/Und
- U1マッチング

自動調整

FELの調整に使うやつ

- Undulator MatchingのCombinedKnobの4パラメータを調整してU1のMCTを最大化するやつ⇨
- Undulator MatchingのCombinedKnobの4パラメータを調整してU2のMCTを最大化するやつ⇨
- アークR56とSXの3パラメータを調整してU1のMCTを最大化するやつ⇨ <http://erlserv5.cerl.ke>
- アークR56とSXの3パラメータを調整してU2のMCTを最大化するやつ⇨ <http://erlserv5.cerl.ke>
- アークR56とSXと分散漏れのパラメータを調整してU2のMCTを最大化するやつ⇨ <http://erlserv>
- パンチャーと入射器のLLRFの2パラメータを調整してU2のMCTを最大化するやつ⇨ <http://erlse>
- パンチャーと入射器のLLRFの4パラメータを調整してU2のMCTを最大化するやつ⇨ <http://erlse>
- ML1とML2のLLRFの4パラメータを調整してU2のMCTを最大化するやつ⇨ <http://erlserv5.cerl>
- ML2のLLRFの2パラメータ(振幅と位相)を調整してU2のMCTを最大化するやつ⇨ <http://erlserv5>
- U1上流のステアリング4つを調整してU1のMCTを最大化するやつ⇨ <http://erlserv5.cerl.kek.jp>
- U2上流のステアリング4つを調整してU2のMCTを最大化するやつ⇨ <http://erlserv5.cerl.kek.jp>
- U2上流の水平ステアリング4つを調整してU2のMCTを最大化するやつ⇨ <http://erlserv5.cerl.ke>
- U2上流の垂直ステアリング4つを調整してU2のMCTを最大化するやつ⇨ <http://erlserv5.cerl.ke>
- ソレノイドSL1とSL2を調整してU2のMCTを最大化するやつ⇨ <http://erlserv5.cerl.kek.jp:260>

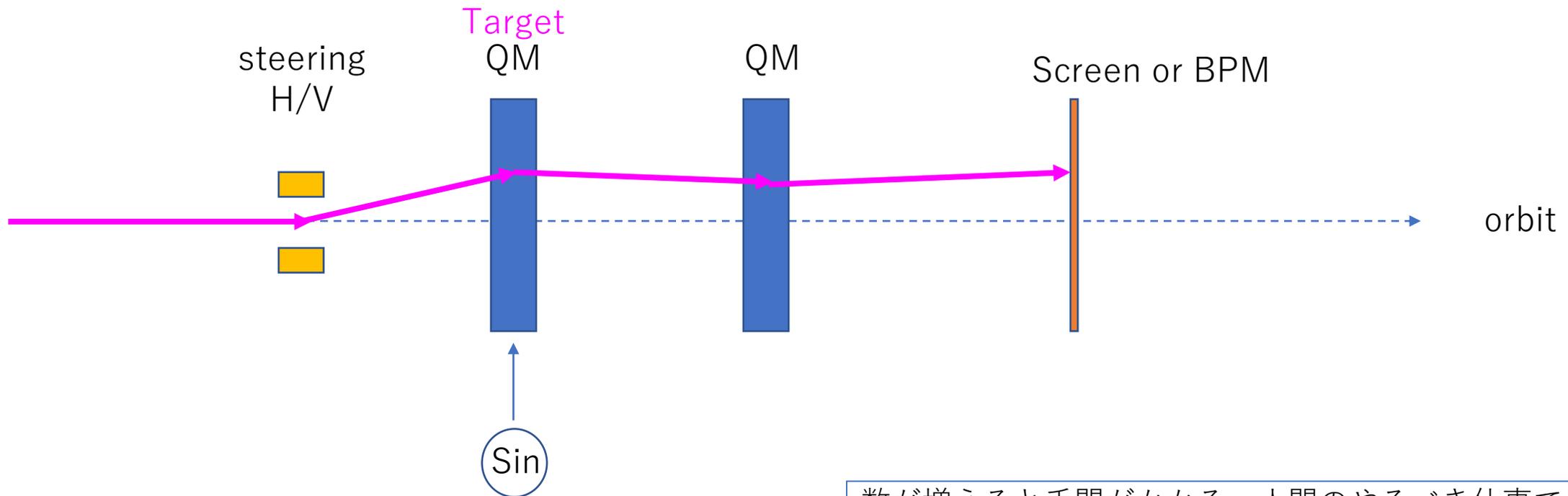
TOC

1. 前振り
2. ベイズ最適化（ガウス過程）の説明と加速器への応用例
 - 原理
 - ツールの紹介
 - 加速器への応用例
3. cERLの説明 & 機械学習が必要となった理由
4. cERLでの適用例
 - a. FELパワーの最大化
 - b. 4極電磁石の磁場中心通し
 - c. ビームサイズ調整
5. まとめ

4b) Q-Mag磁場中心通しについて

目的と手順

1. TargetとなるQMの磁場中心を通すのが目的
2. Target QMの電流値を適当に変える (sinなど)
3. 上流の補正電磁石 H/V を動かして、スクリーンで重心が動かない場所を探す

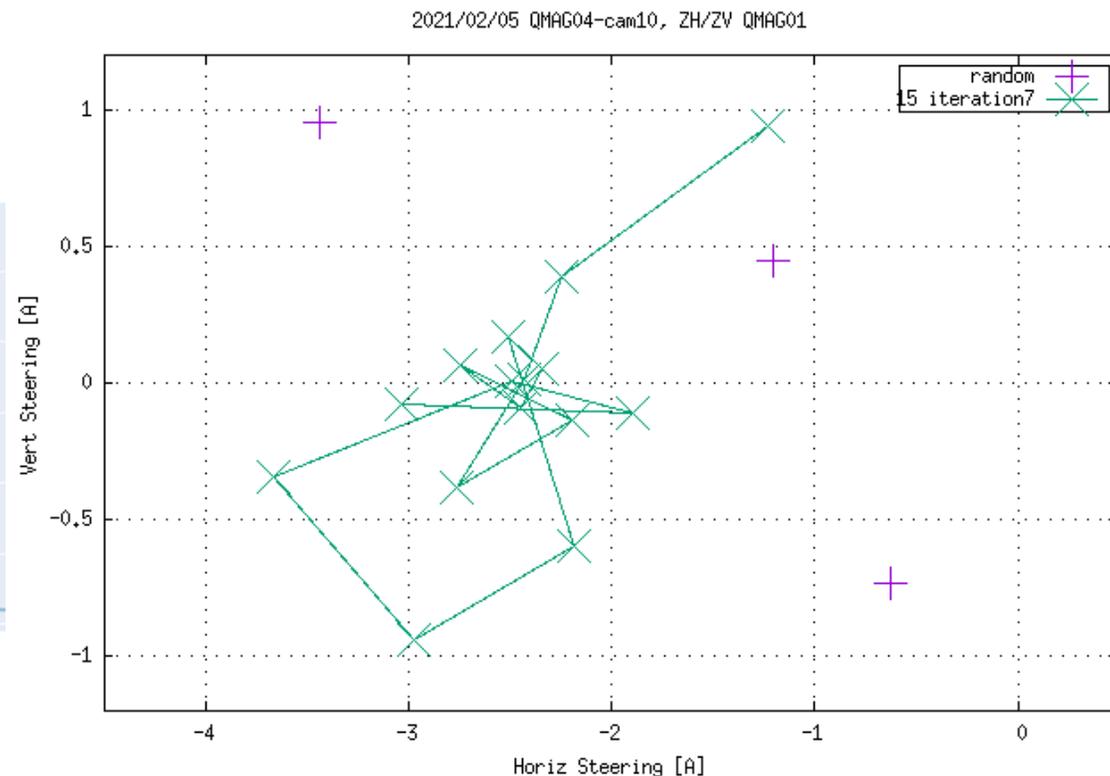
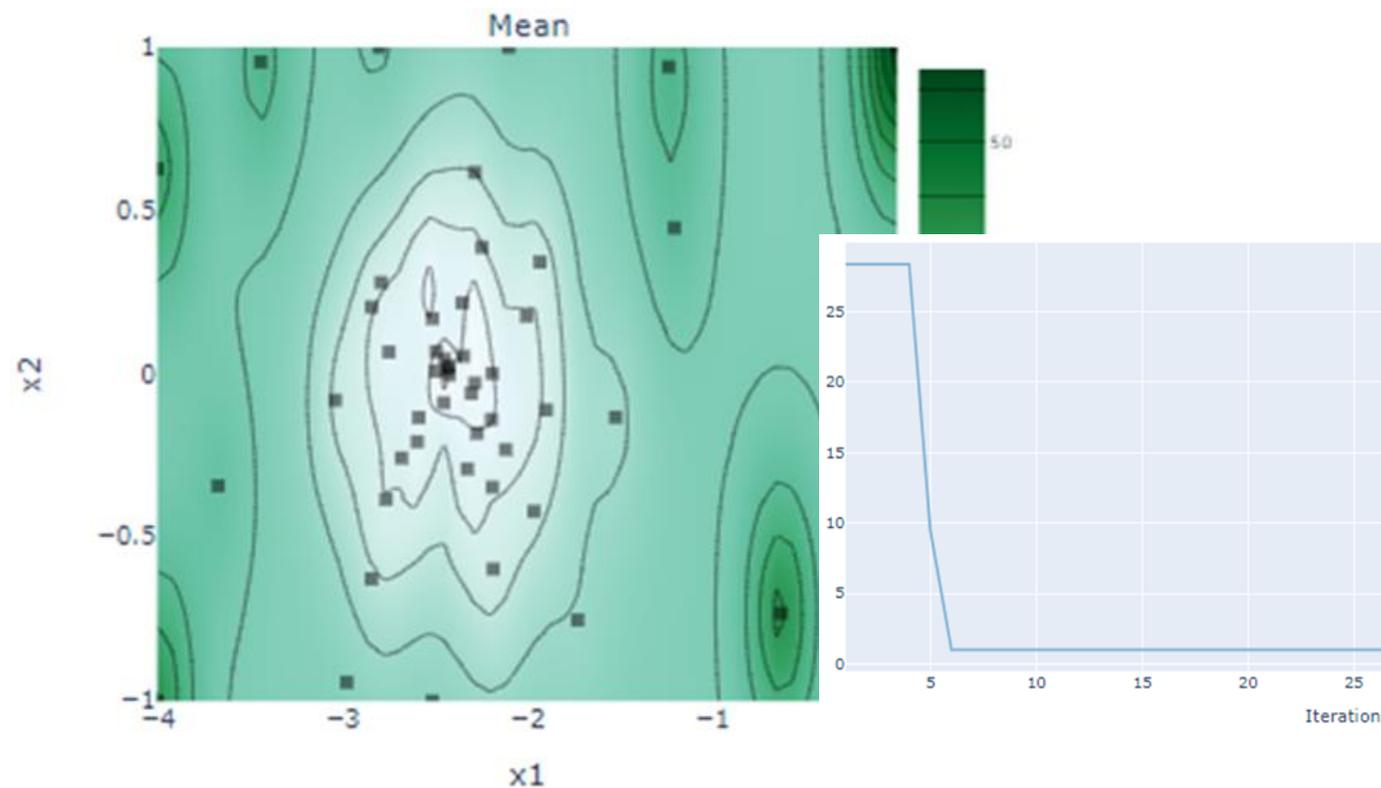


数が増えると手間がかかる→人間のやるべき仕事ではない

実際のビームを使った調整結果

Ax Platform を使用 (GPyTorch+BoTorch)

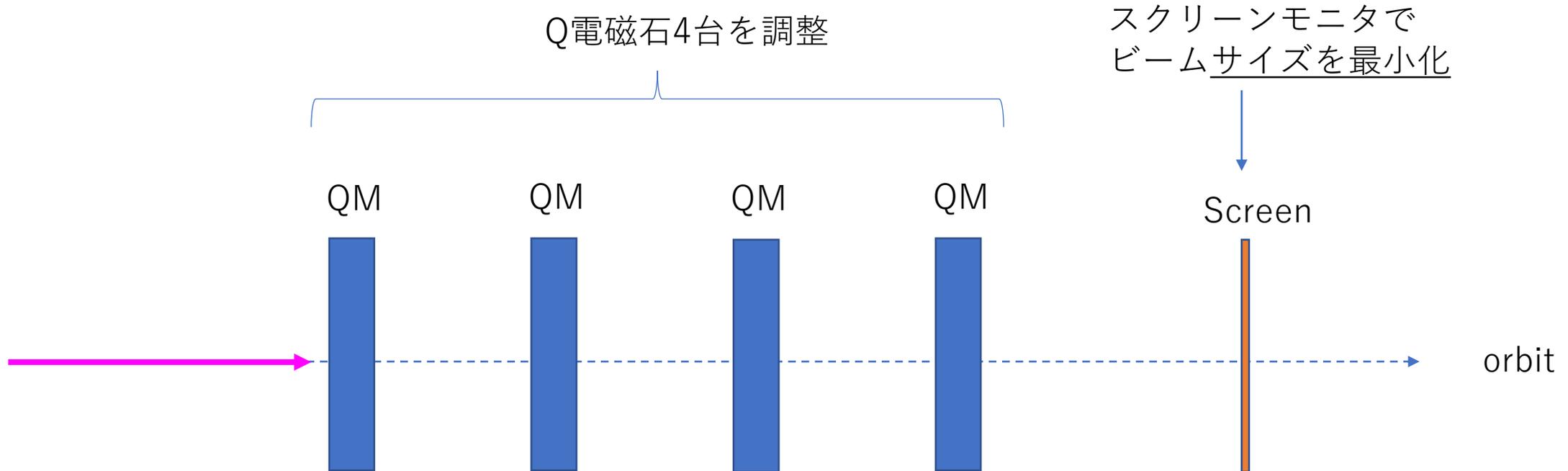
- Random (SOBOL) 3回+最大45回のiteration。実際には6手目にはほぼ最小に到達。



この例ならば機械学習である必然性は無く、ネルダーミードでもできる。
しかし例としては簡単が良い。

4c) ビームサイズ調整例

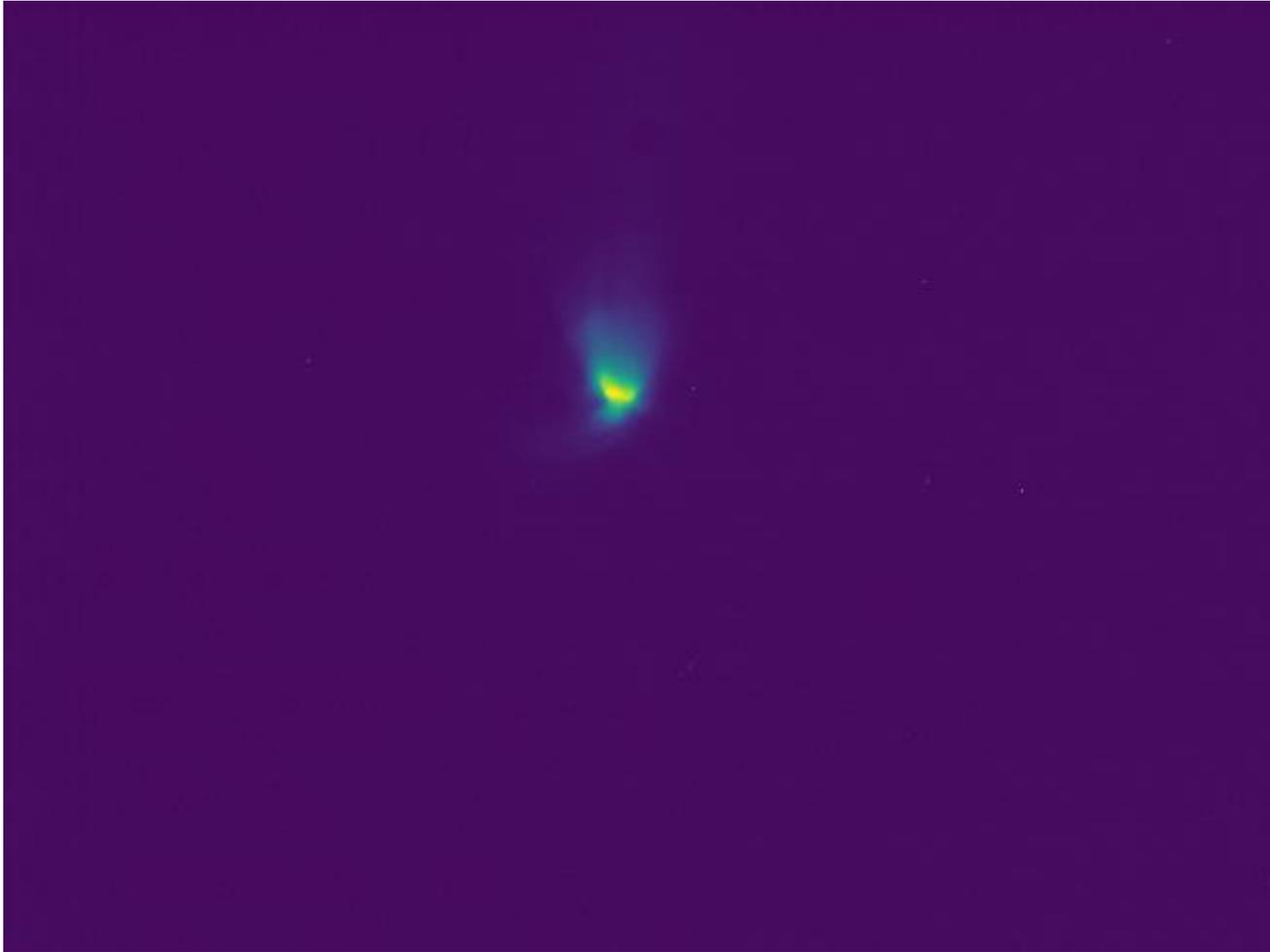
THz実験用にビームサイズを絞る調整（FEL調整とは別）



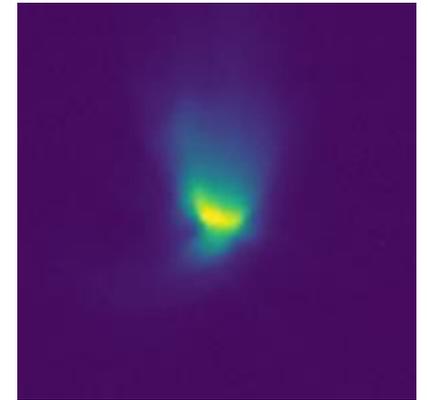
次ページで調整中の動画を紹介する

4c) ビームサイズ調整例

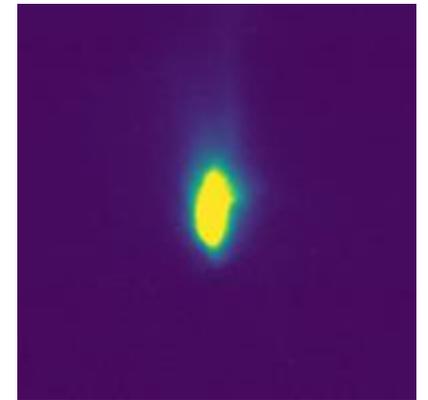
ビームサイズを小さくする調整 (for THz) を自動化 ↓



before



after



この図ではわざと
飽和(黄色) させている 22

まとめ

- ベイズ最適化手法をcERLビーム調整に適用した。本日は以下の3つを紹介。
 - a. FELパワー最大化
 - b. Q-Magnet磁場中心通し
 - c. ビームサイズ調整
- Gaussian Process によるFELパワー最大化は日々のビーム調整ツールとして有効だった
 - これ無しにはcERLのFEL調整は出来なかった
 - 電磁石はもちろん、RF位相/電圧などまで含め自動調整をおこなった
 - 2パラメータの場合に全空間探索とGP+EIとの比較を行ってほぼ同じ結果を得ている（時間の都合で本日は紹介していない）
- 最近は便利なツールがそろっており、簡単に試すことが出来る状況になっている
 - GPyOptの他PyTorch(AX)などを試している。他にもGPyFlow, Optuna など各種ある
 - 注：GPyOpt は開発終了が宣言されています。ノイズの取り扱いにも少し注意が必要。
- 今後/その他
 - ツールによる違い、カーネルの調整などでより早く調整ができるようになることを目指す
 - ネルダーミード法や、他の機械学習手法との比較は今後の課題
 - 磁石のヒステリシス効果をどうすべきか（探索時間優先 or 再現性優先 or 効果込で学習, etc)
 - 複数ステップの全自動化？