

SafeLineBO を用いたビーム調整テストの報告

Report on Beam Tuning Tests Using SafeLineBO

2025 12/9

理研仁科加速器科学研究所センター 西 隆博

共同研究者

森田 泰之

理研仁科加速器科学研究所センター

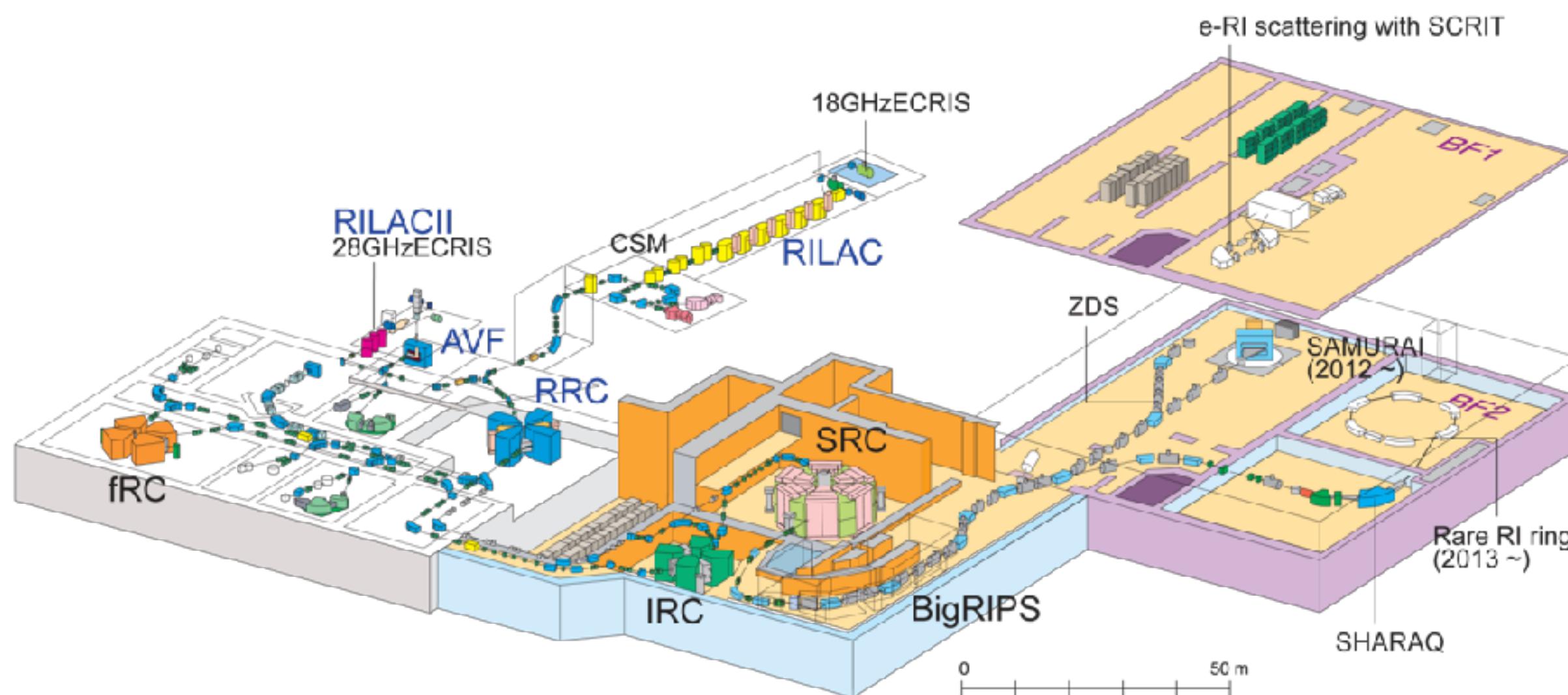
江川 弘行

理研最先端研究プラットフォーム連携(TRIP)事業本部

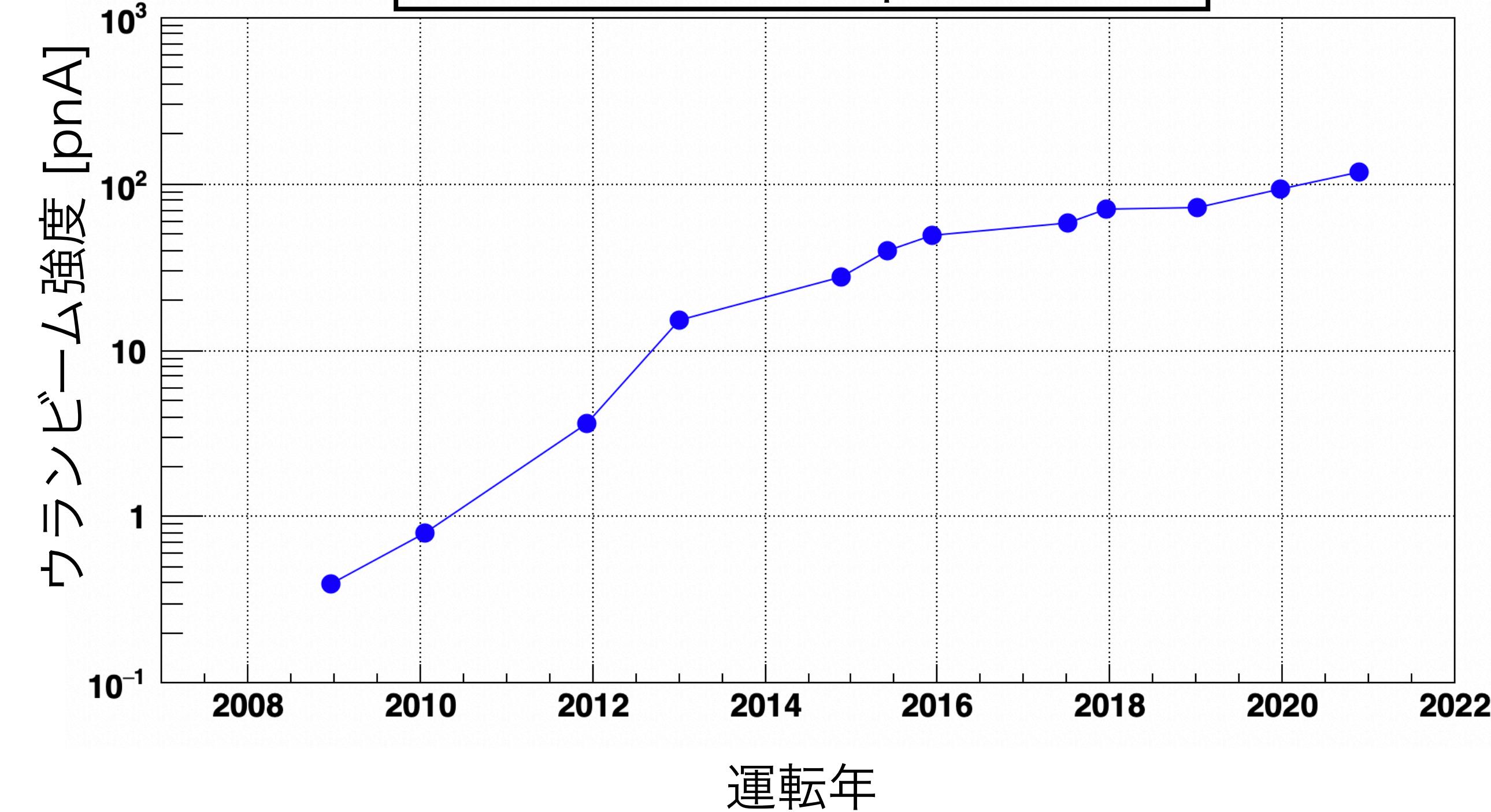


理研 RIBF と機械学習を用いたビーム輸送最適化

RI ビームファクトリーの俯瞰図



ウランビーム強度 [pnA] の変化



理化学研究所 RIBF は 2006 年より運転を開始し、現在では 100 pnA を超えるウランビーム (~345 MeV/u) を供給することで世界の原子核物理学を牽引している。その供給ビーム強度の増加率は年々鈍化している。
 → ビームラインの (600 を超えるパラメータの) 手動最適化の限界も一つの要因となっている。
 → Machine Learning を使った最適化でさらなる領域へ！
 ※ 大強度重イオンビームだと、ビームロス(熱)で律速される場合も。loss を 1% → 0.5% にすると、強度が2倍に！

RIBF における最適化プログラムに求める条件

- ・ 多数のパラメータの同時最適化
- ・ (機械学習としては) 少ない学習データに対応
- ・ 大強度ビームでも使えるビームの指標
- ・ 大強度ビームを “むやみに振り回さない” 安全システム
- ・ 環境の変化などに追随して常時パラメータ最適化するシステム

RIBF における最適化プログラムに求める条件

- ・多数のパラメータの同時最適化
- ・(機械学習としては) 少ない学習データに対応

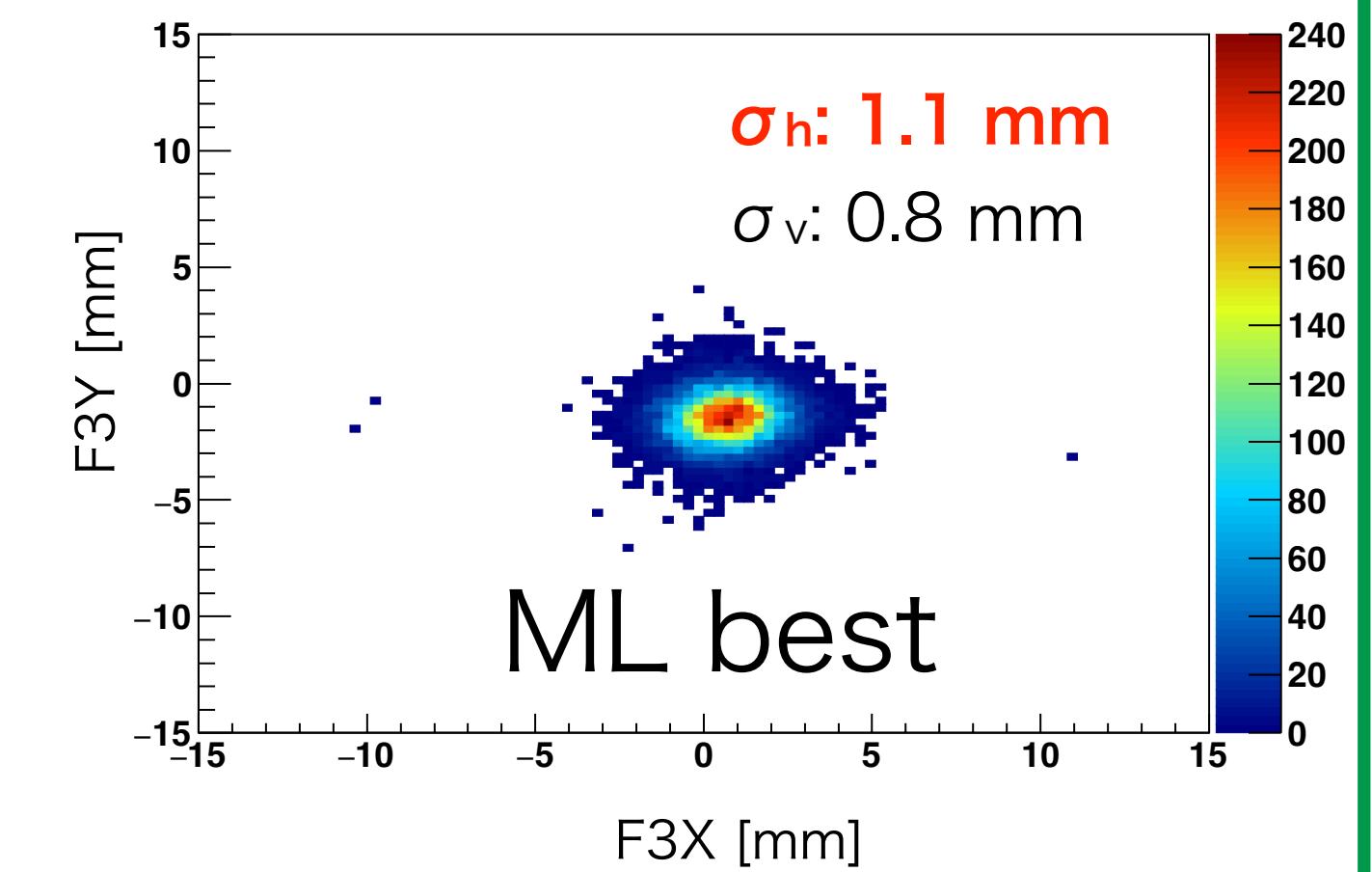
ガウス過程回帰を用いた
ベイズ最適化

- ・大強度ビームでも使えるビームの指標

荷電変換粒子

- ・大強度ビームを“むやみに振り回さない”安全システム
- ・環境の変化などに追随して常時パラメータ最適化するシステム

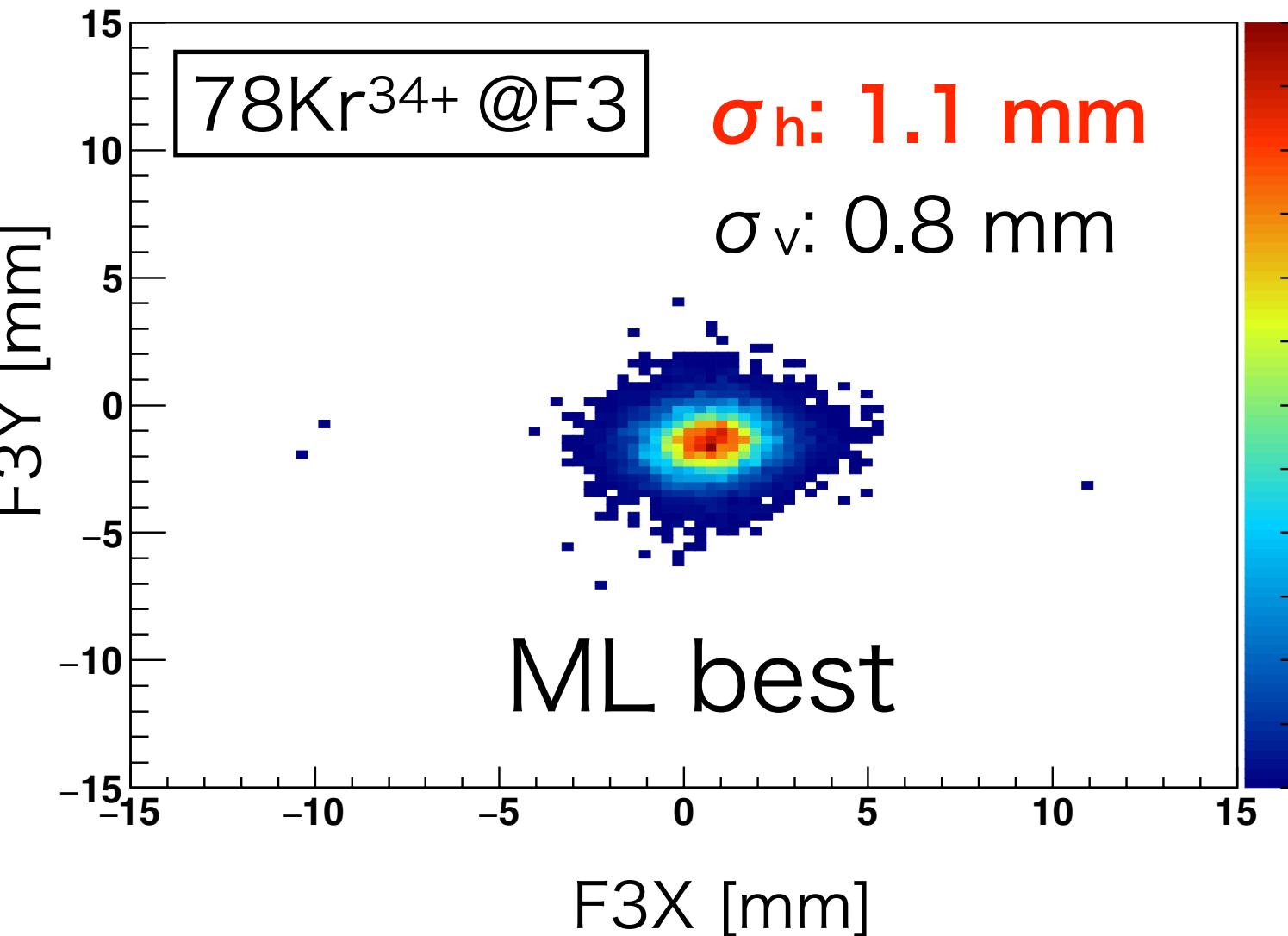
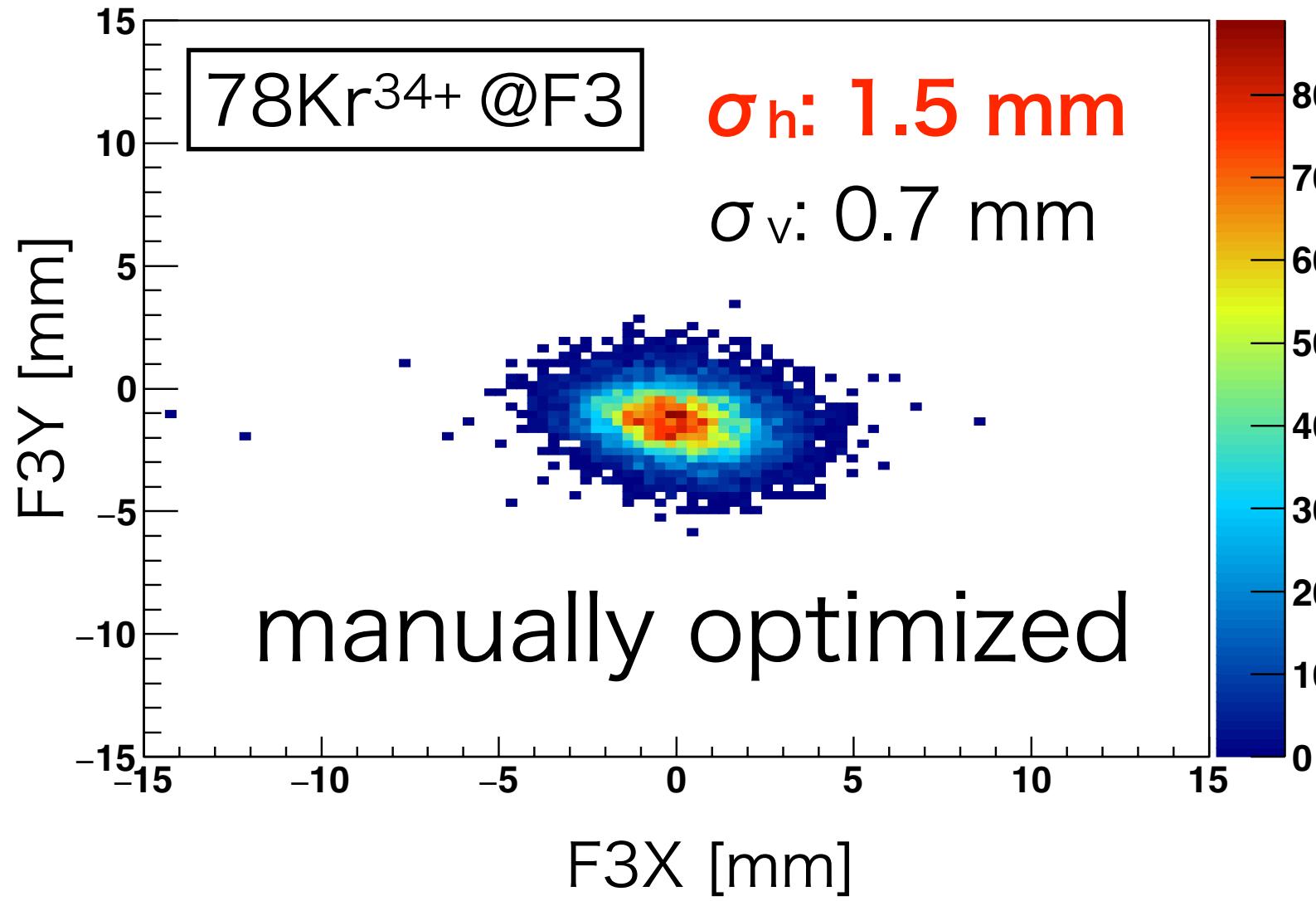
Kr³⁶⁺ + Be target
→ Kr³⁴⁺ で低強度化



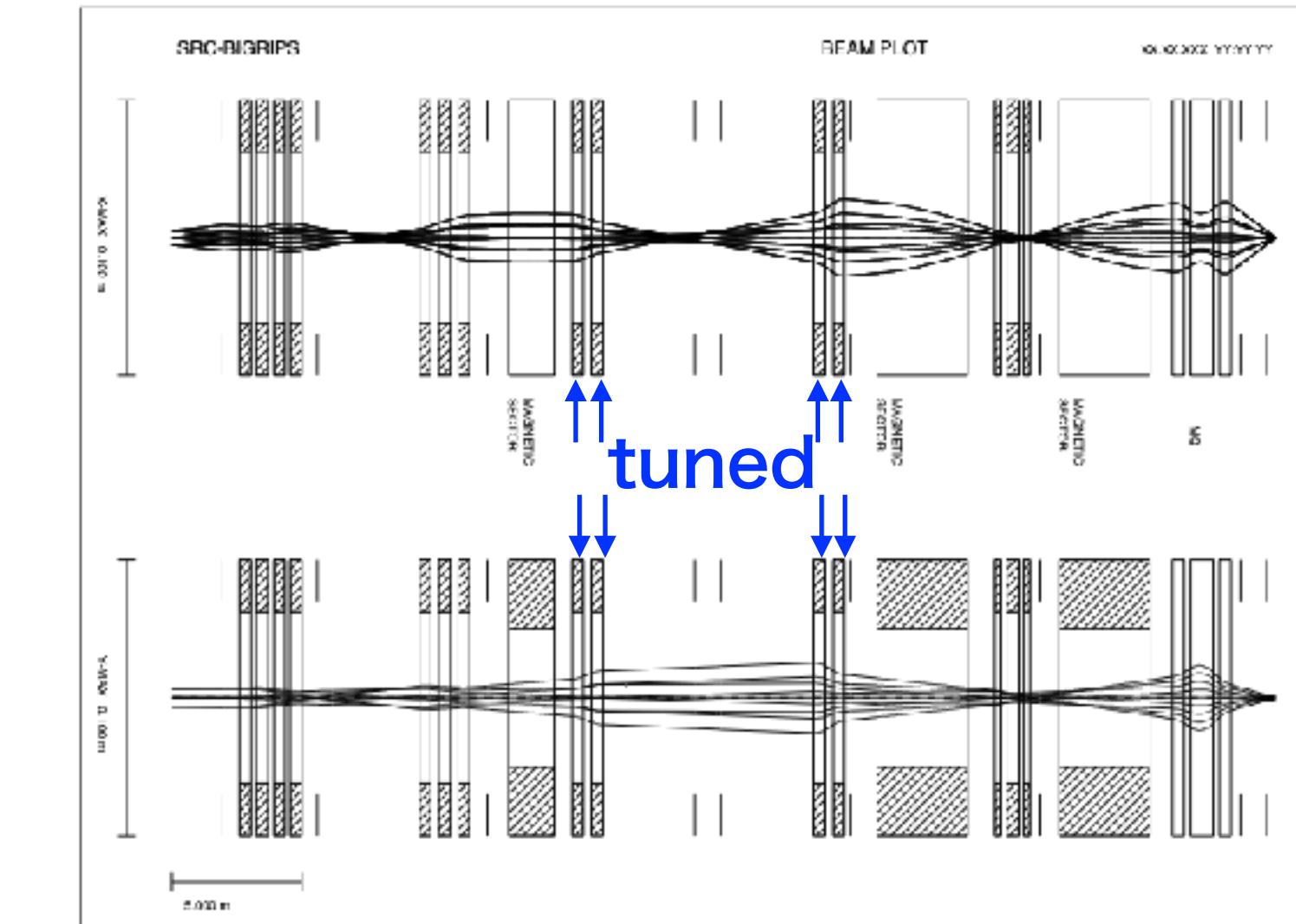
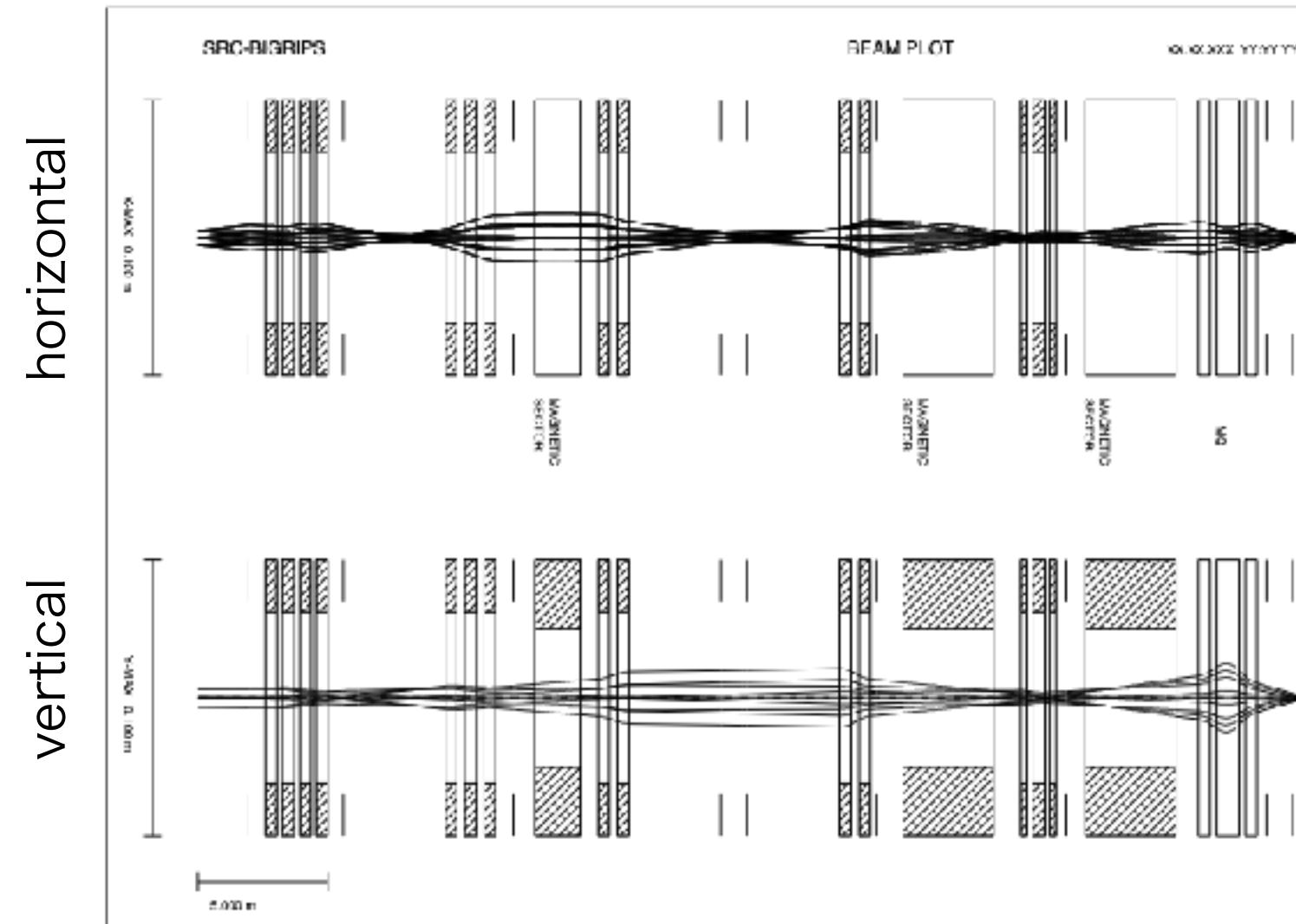
T. Nishi et al., Proc. of HIA2025, MSU, USA, TUP09, (2025).

“高強度”一次ビームで荷電変換粒子の測定による光学系最適化

Compare with manually optimized vs ML optimized optics



4つの四重極の最適化 / 30分
スポットサイズ ↓ 30%
通過率 はほぼ同程度



スポットサイズだけでなく、実験者のありとあらゆる解析結果を指標にビーム調整が可能となる。

RIBF における最適化プログラムに求める条件

- **多数のパラメータの同時最適化**
- (機械学習としては) 少ない学習データに対応
- 大強度ビームでも使えるビームの指標
- **大強度ビームを“むやみに振り回さない”安全システム**
- 環境の変化などに追随して常時パラメータ最適化するシステム

今日はこの2つをテーマに話す

多数のパラメータの同時最適化をどうやって実現するか？

- ・例えばRIBFだとパラメータは数百を超える。
- ・ガウス過程回帰をつかったベイズ最適化では、せいぜい 20 パラメータ？
(範囲にもよるが、以前行った最適化は 4 パラメータで 30分)

解決方法 ①: グループ分けをして少数のパラメータずつ最適化
→通称 “落ち穂拾い”

Pros: 何をやっているかがクリア。人がやっていることの自然な延長。

Cons: 異なるグループのパラメータを同時最適化できない。

解決方法 ②: 多次元空間の中から ”賢く” 部分空間を抜き出す
→ LineBO

Pros: 正しく動けば、多次元最適化を効率よく行うことが可能。

安全関数を用いた手法と組み合わせる→ SafeLineBO

Cons: 正しく感度のあるパラメータ(組)を見つけ出すことは可能？？

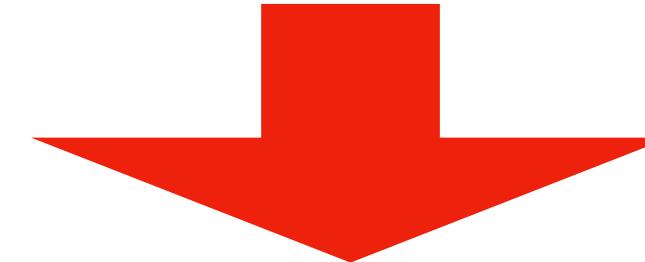
自動最適化のテスト@ RILAC / RILACII ~ RRC (昨年一部報告)

多次元最適化問題に対する解としての“落穂拾い”



落穂拾い@RIBF:

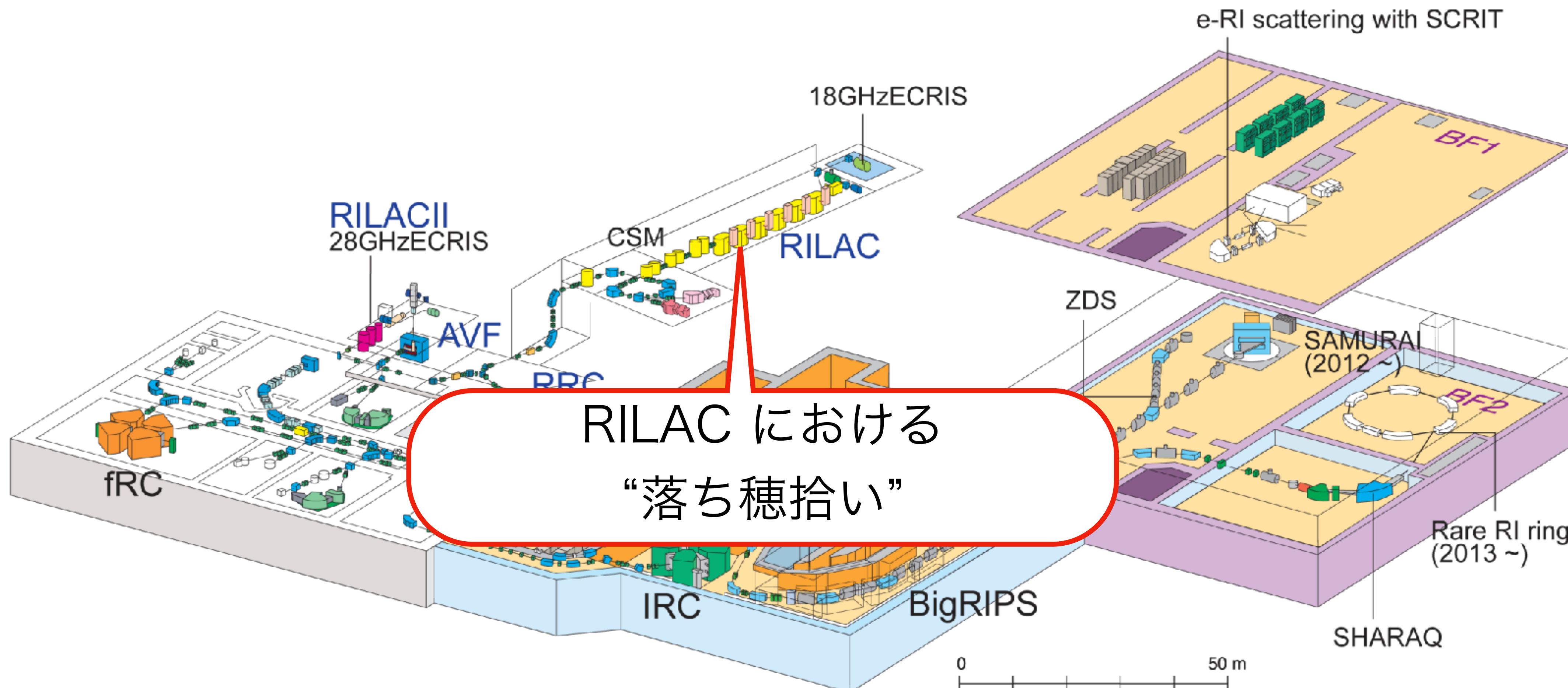
ある程度全てのパラメータを最適化したあと、最後に各パラメータを順番に微小に変化させ通過率などを上げる調整が残ってないか確認する作業



これならば現状の単純なベイズ最適化でも使える？

- ・パラメータの変更範囲を ~ 20%以内に
- ・活用 >> 探索 (局所最適化を目指す)
- ・扱いやすい四重極磁石/Str. を主に調整

自動最適化のテスト@ RILAC / RILACII ~ RRC



RILAC (線形加速器) における自動最適化のテスト (局所最適化)

2024年6月: 新しい核種 (^{2+}He) の常電導加速空洞パートにおける加速試験

→ 経験が浅く、まだ手動最適化されていない。ちょうどいい対象。

最適化する四重極磁石:

~ RFQ 下流から、加速空洞間 / 空洞内の四重極磁石, Steerer 磁石 (43パラメータ)

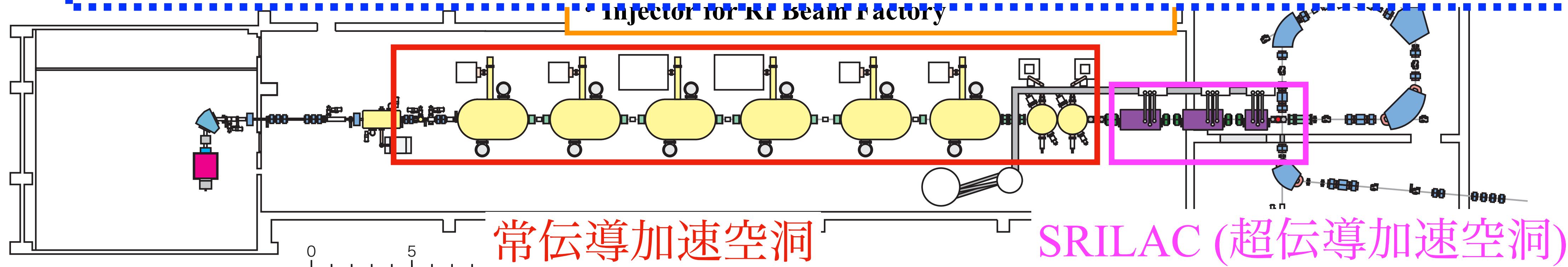
※Energy が測定できなかったため、RF の HV / 位相は触らず

目標:

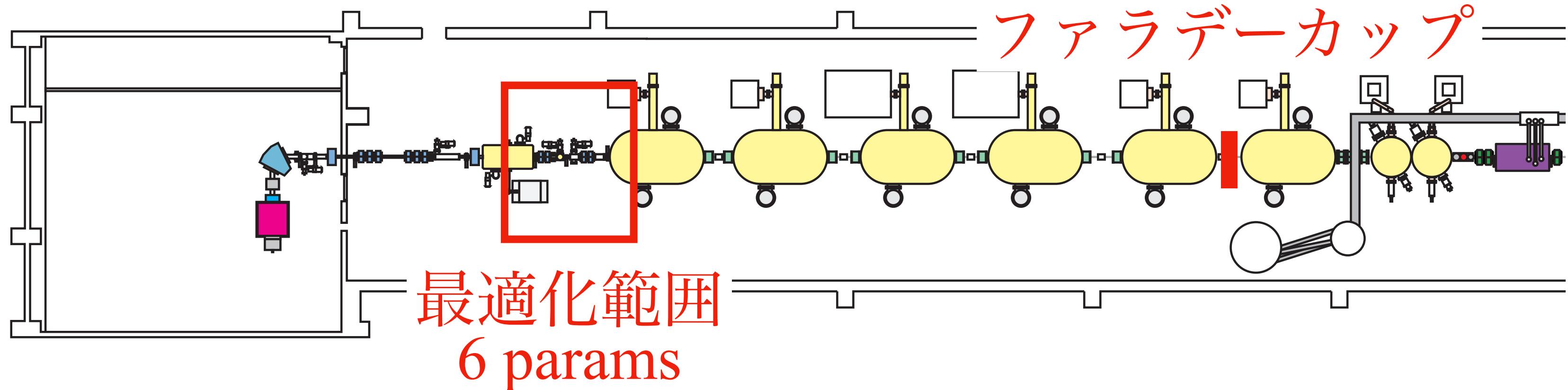
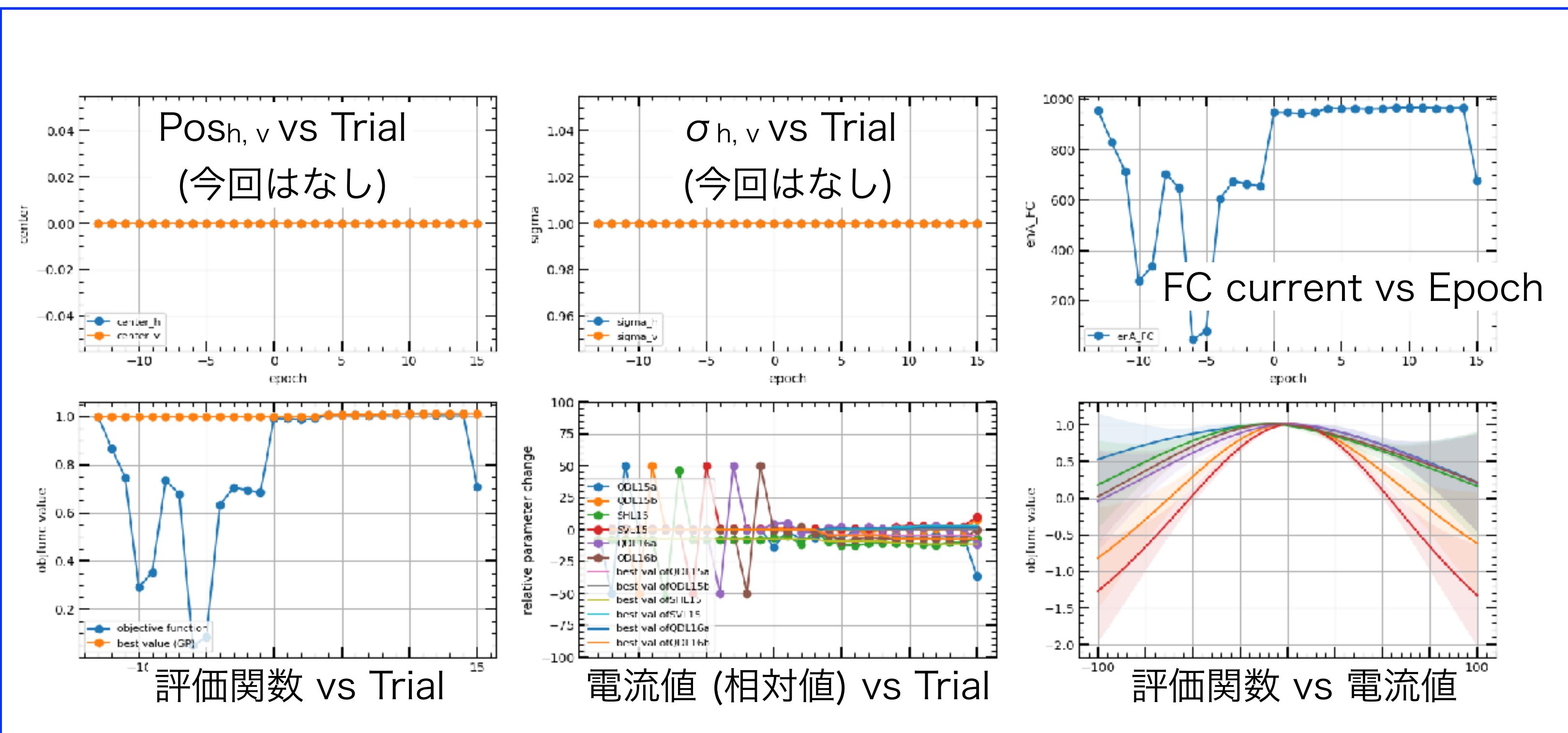
最下流ファラデーカップでの電流値↗

獲得関数:

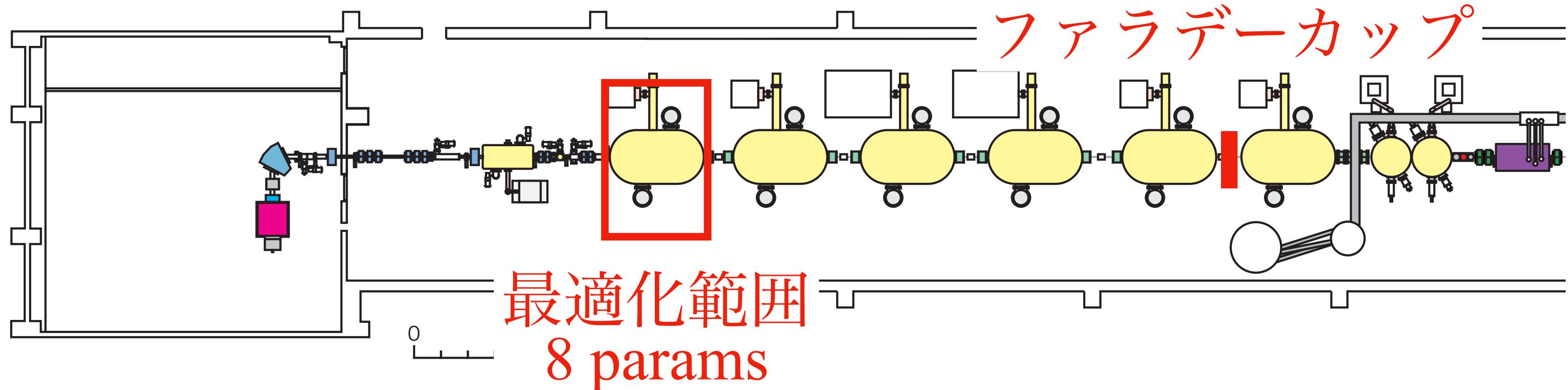
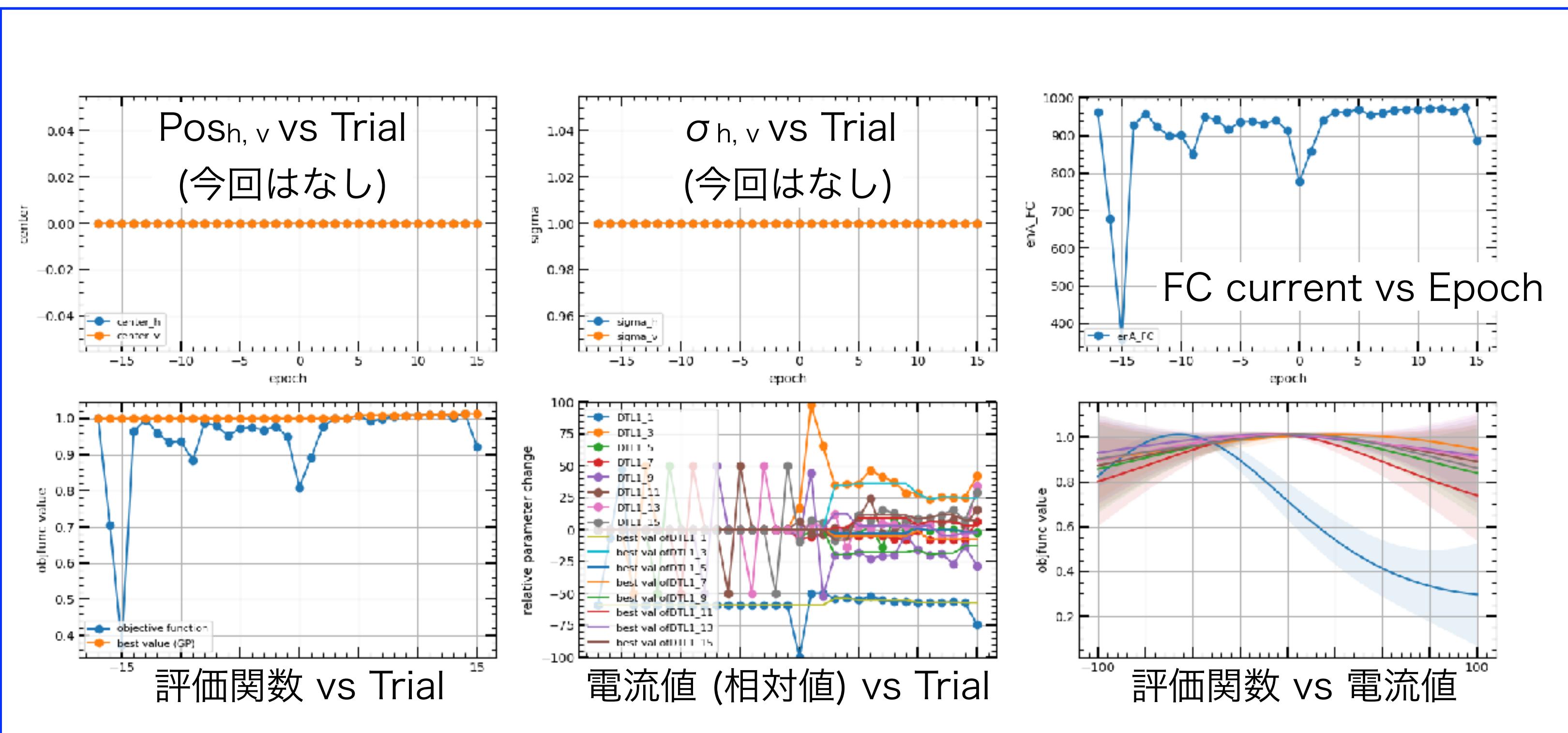
Lower Confidence Bound (LCB) / $\beta \sim 0.2$ (非常に活用寄り)



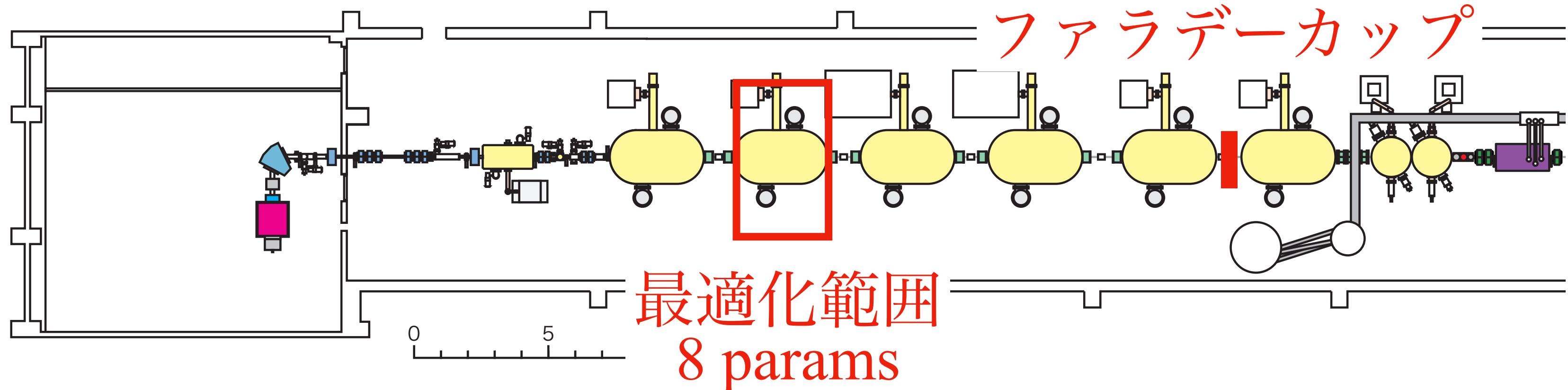
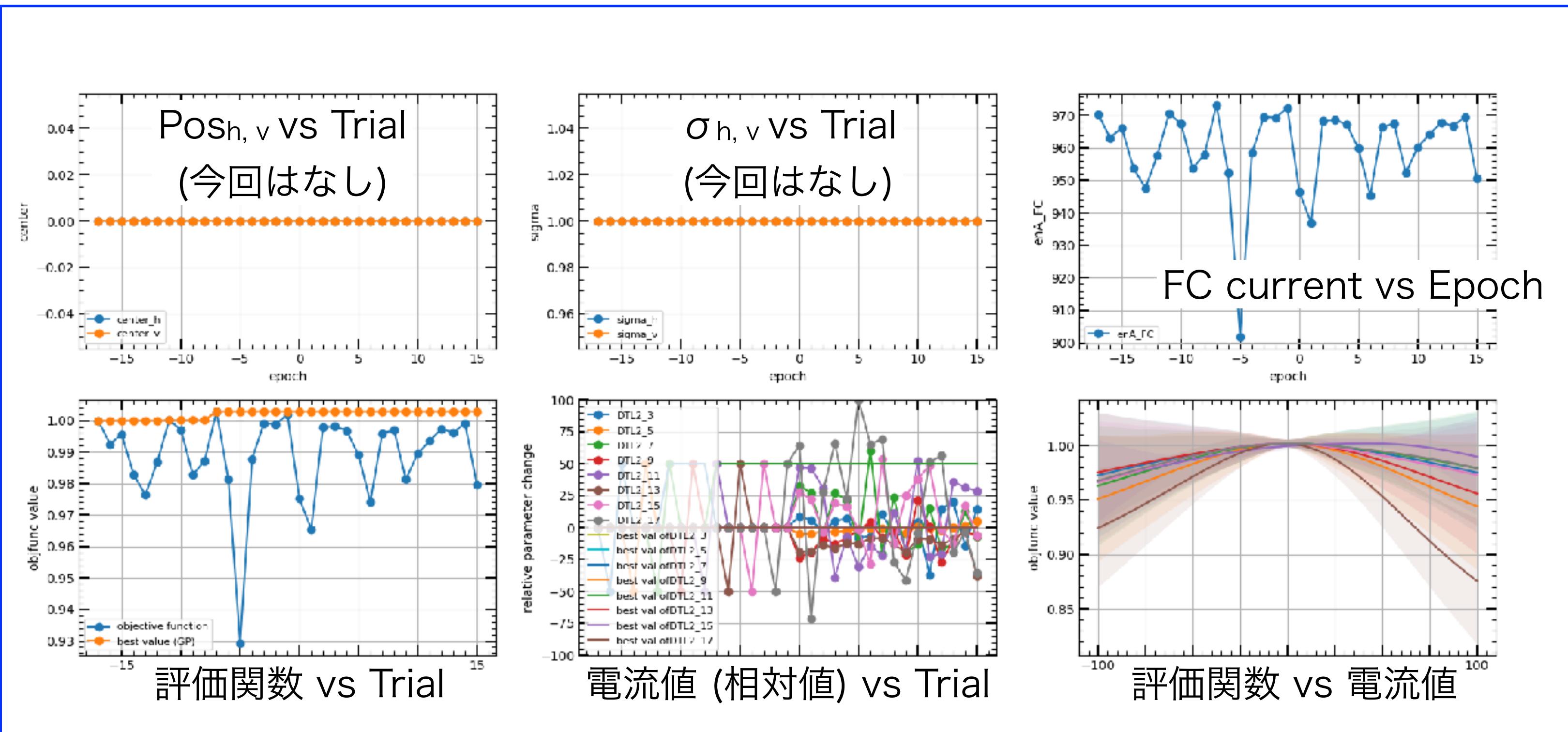
最適化の過程の画像



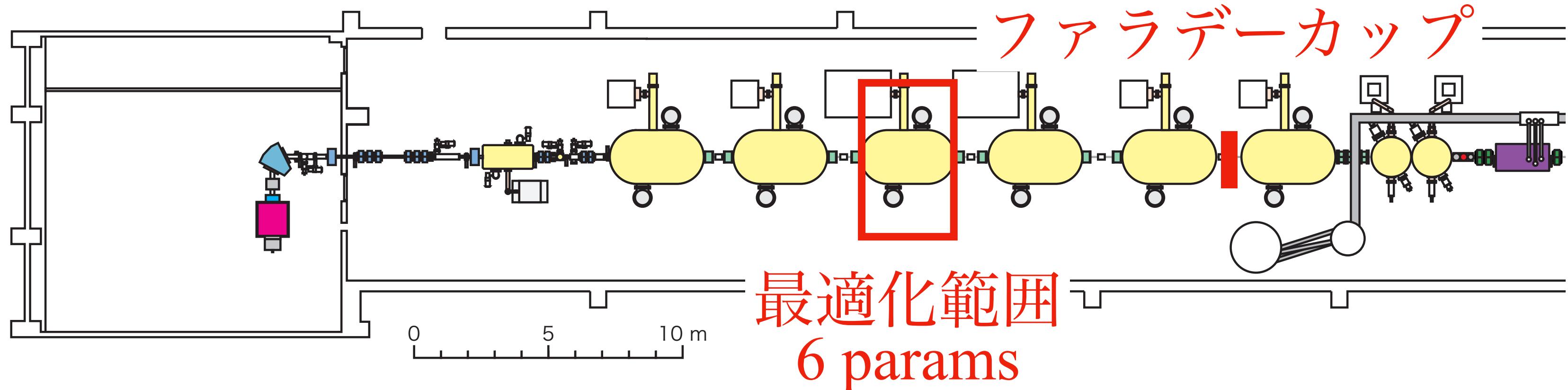
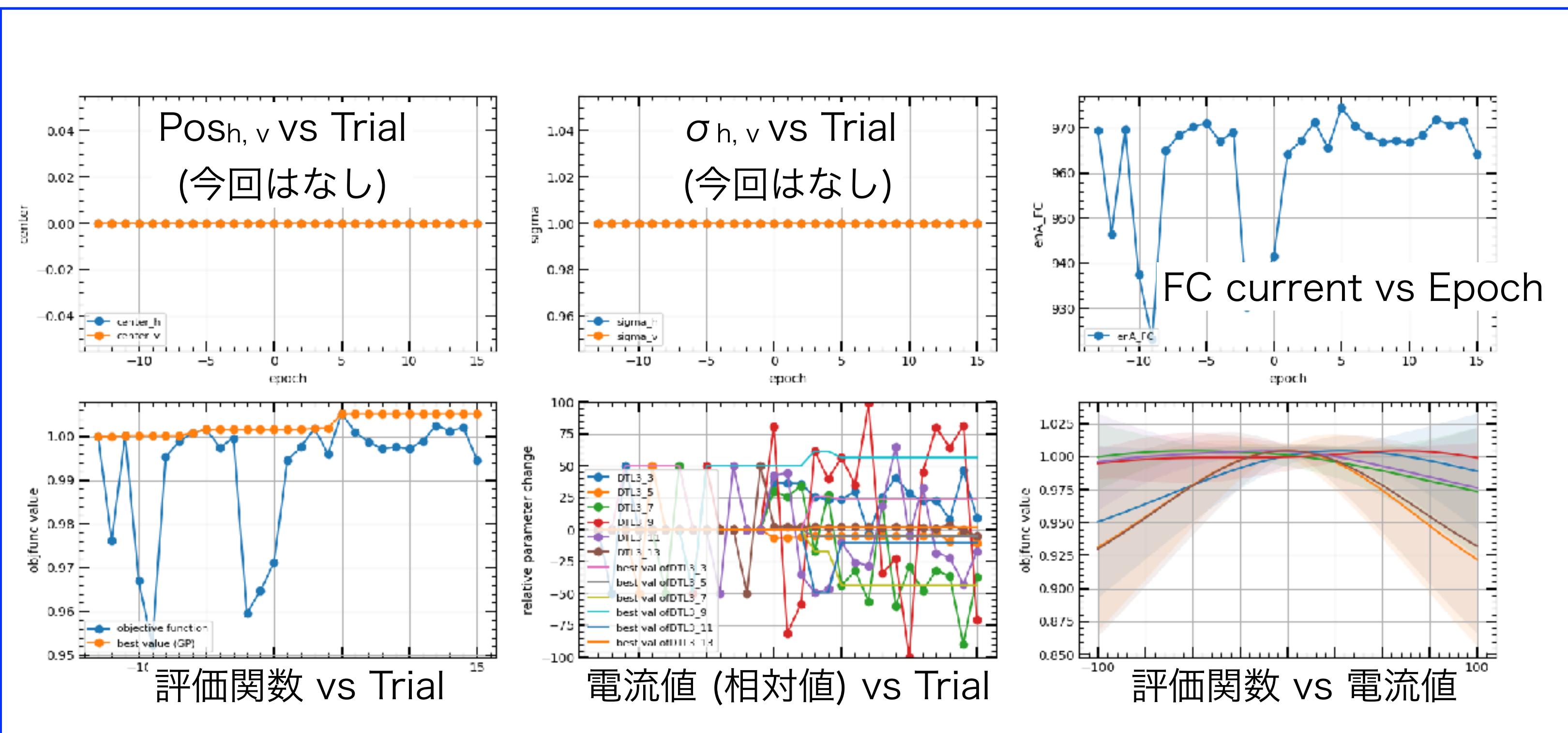
最適化の過程の画像



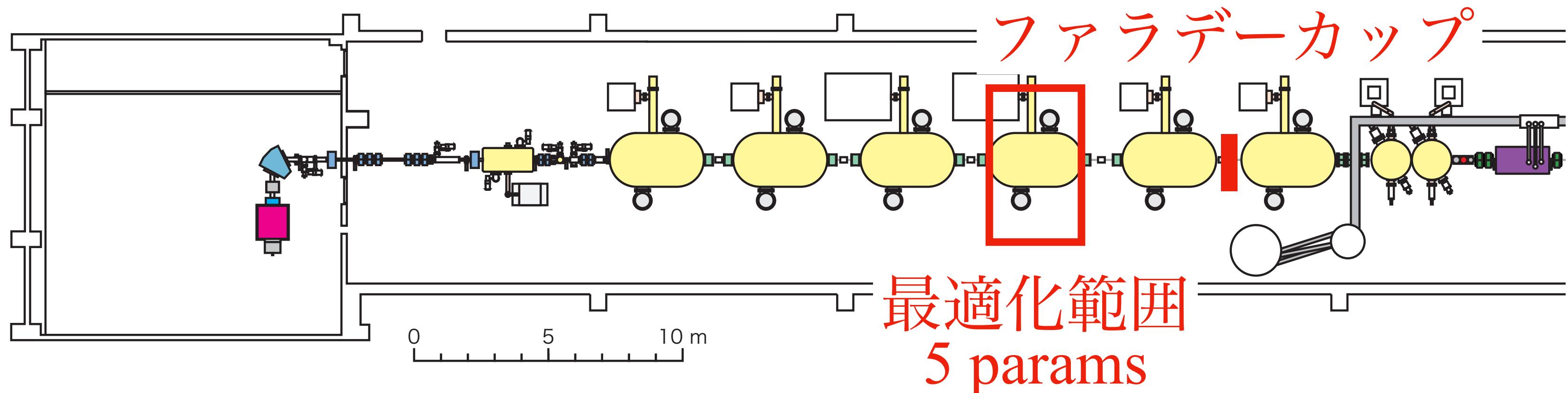
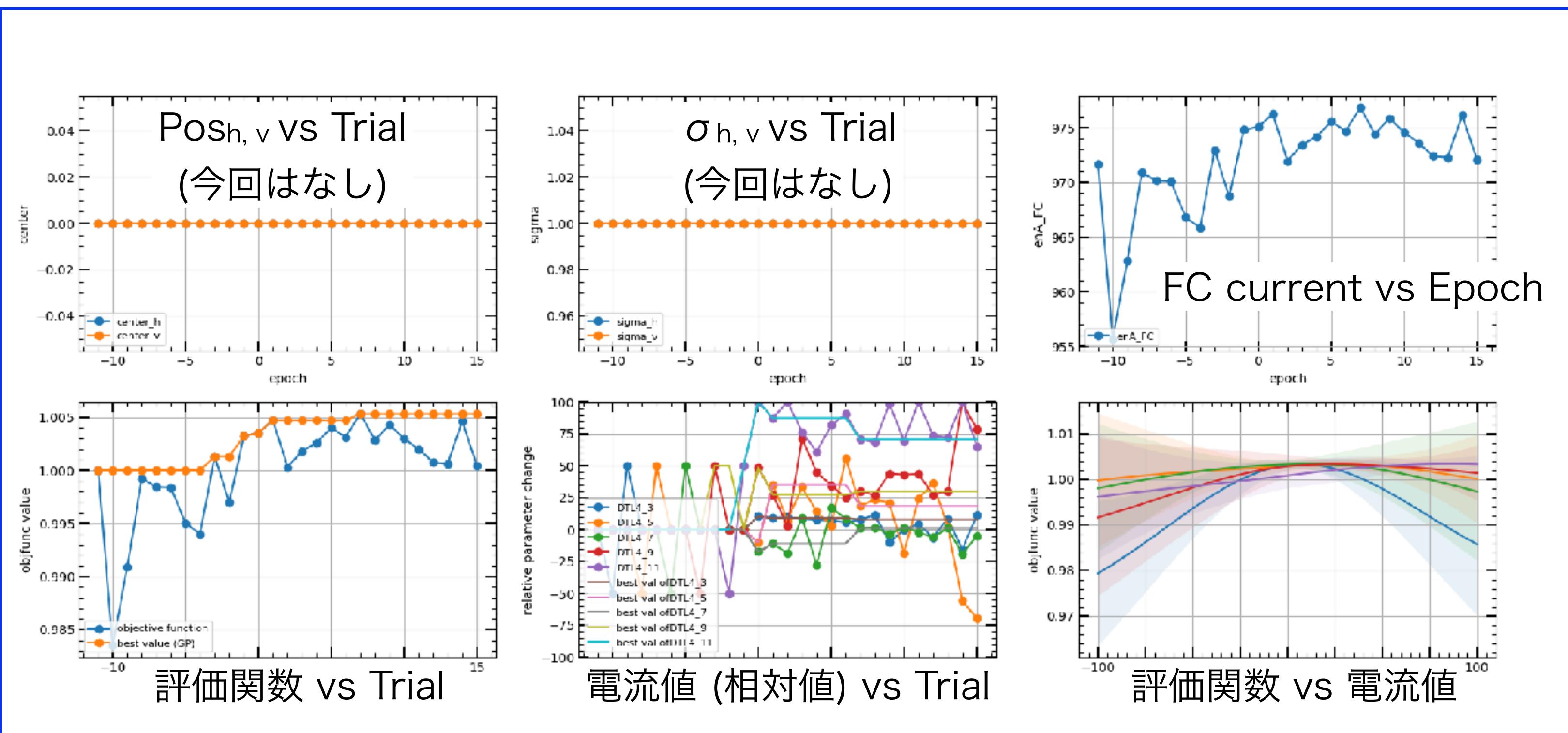
最適化の過程の画像



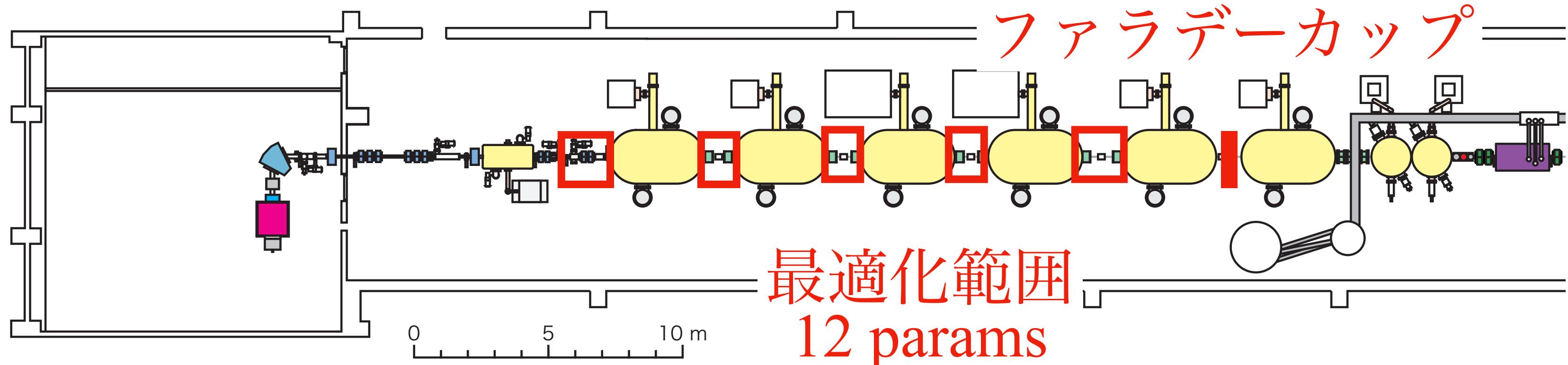
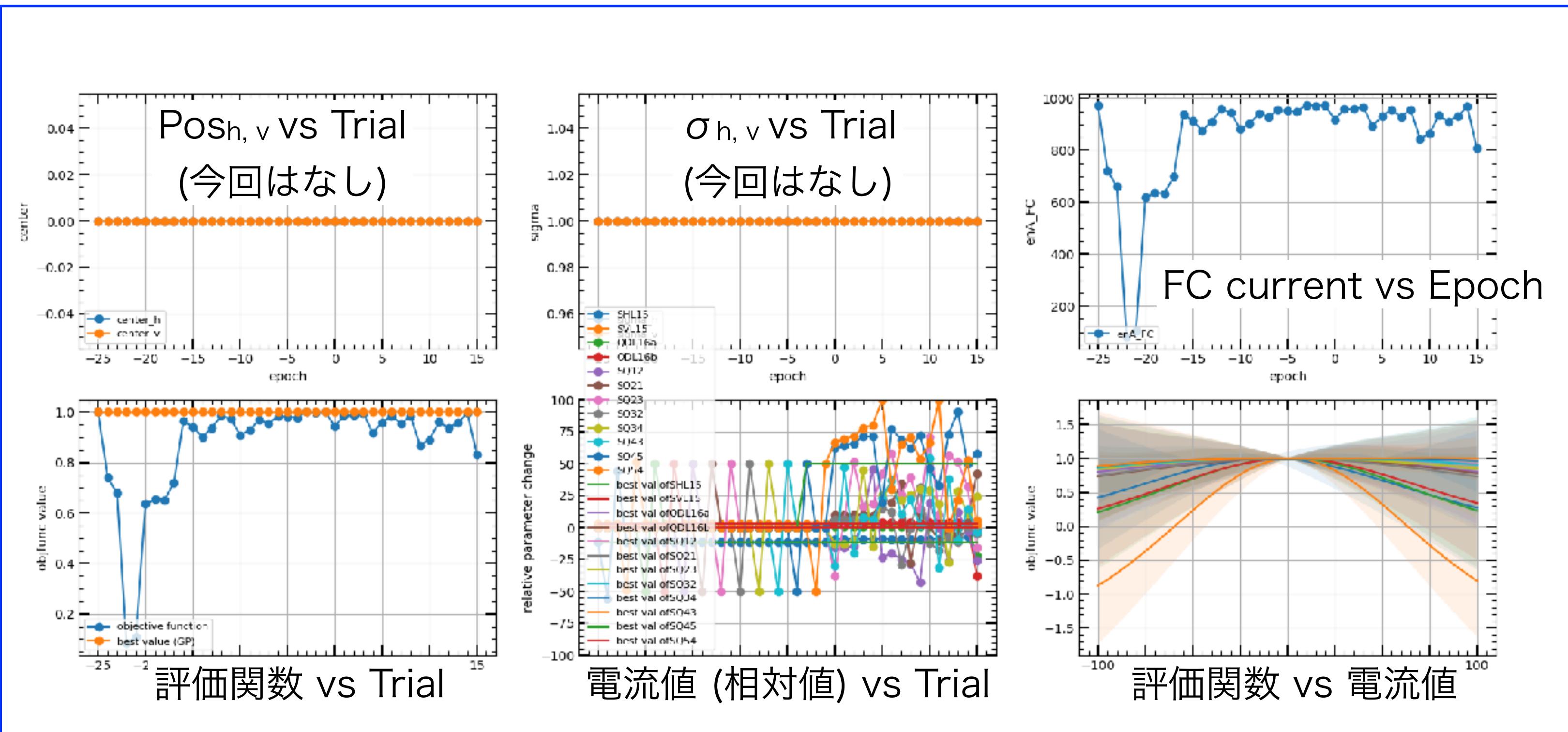
最適化の過程の画像

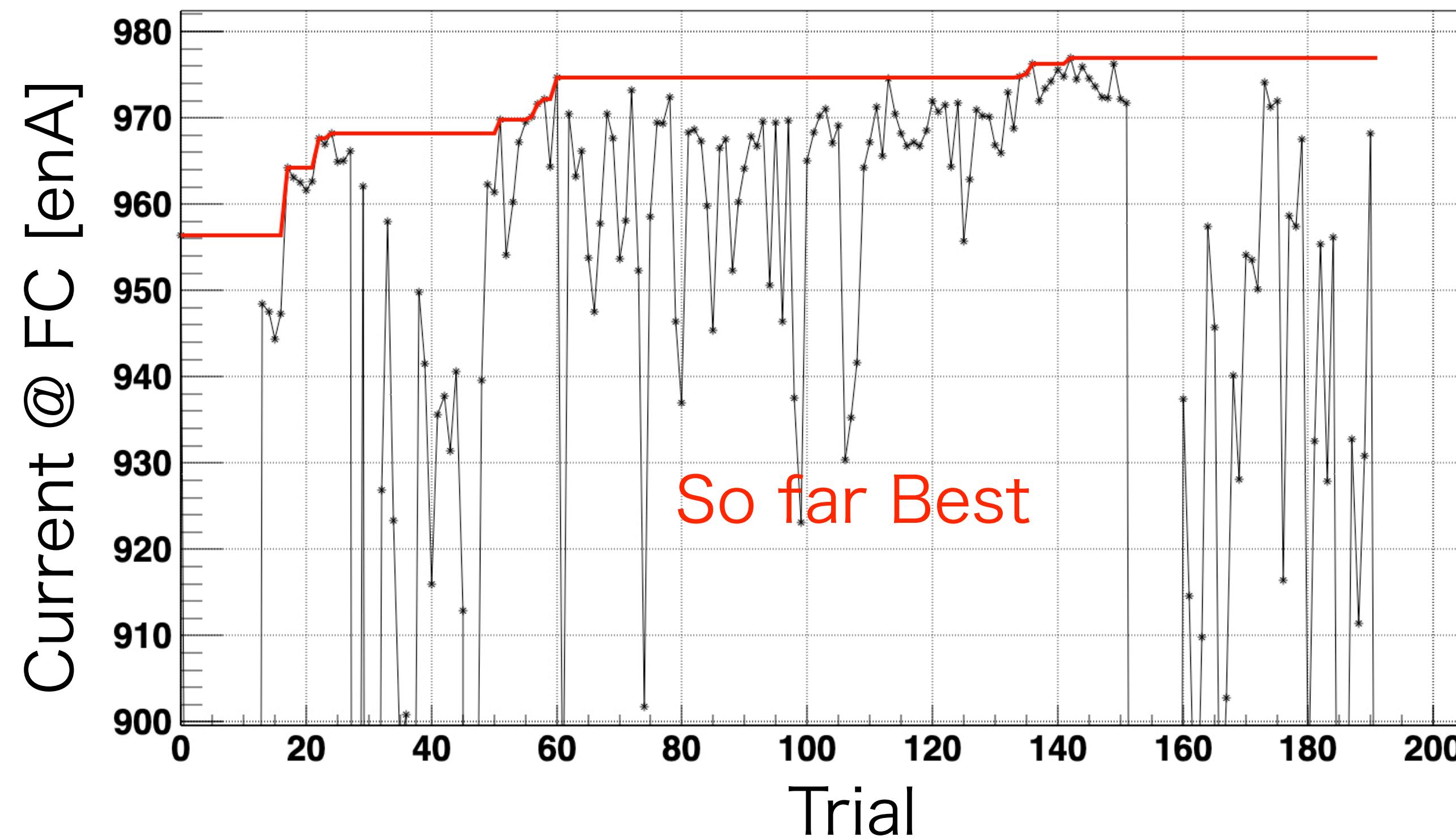


最適化の過程の画像



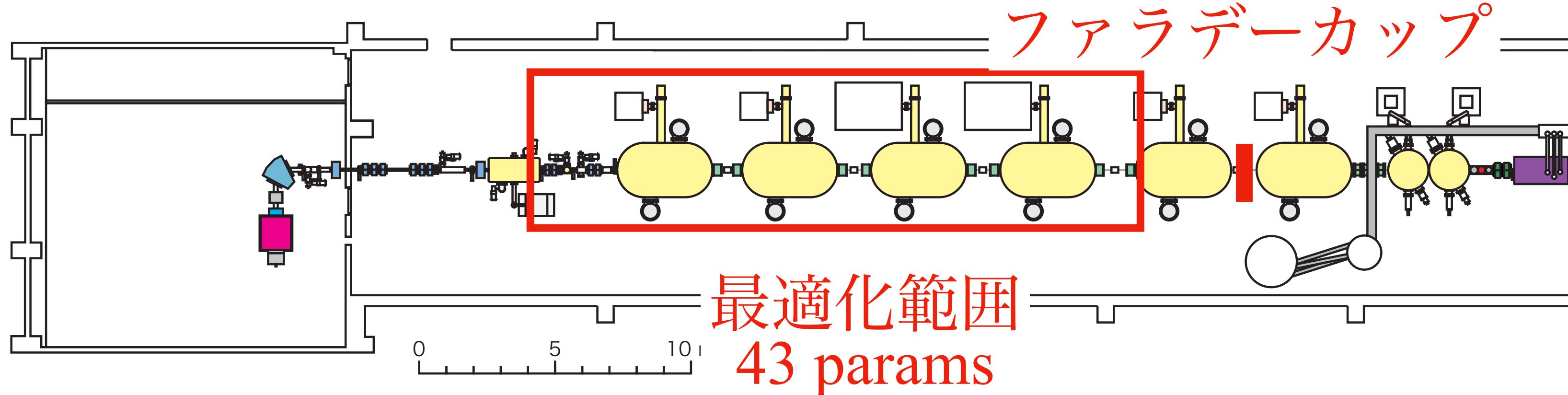
最適化の過程の画像

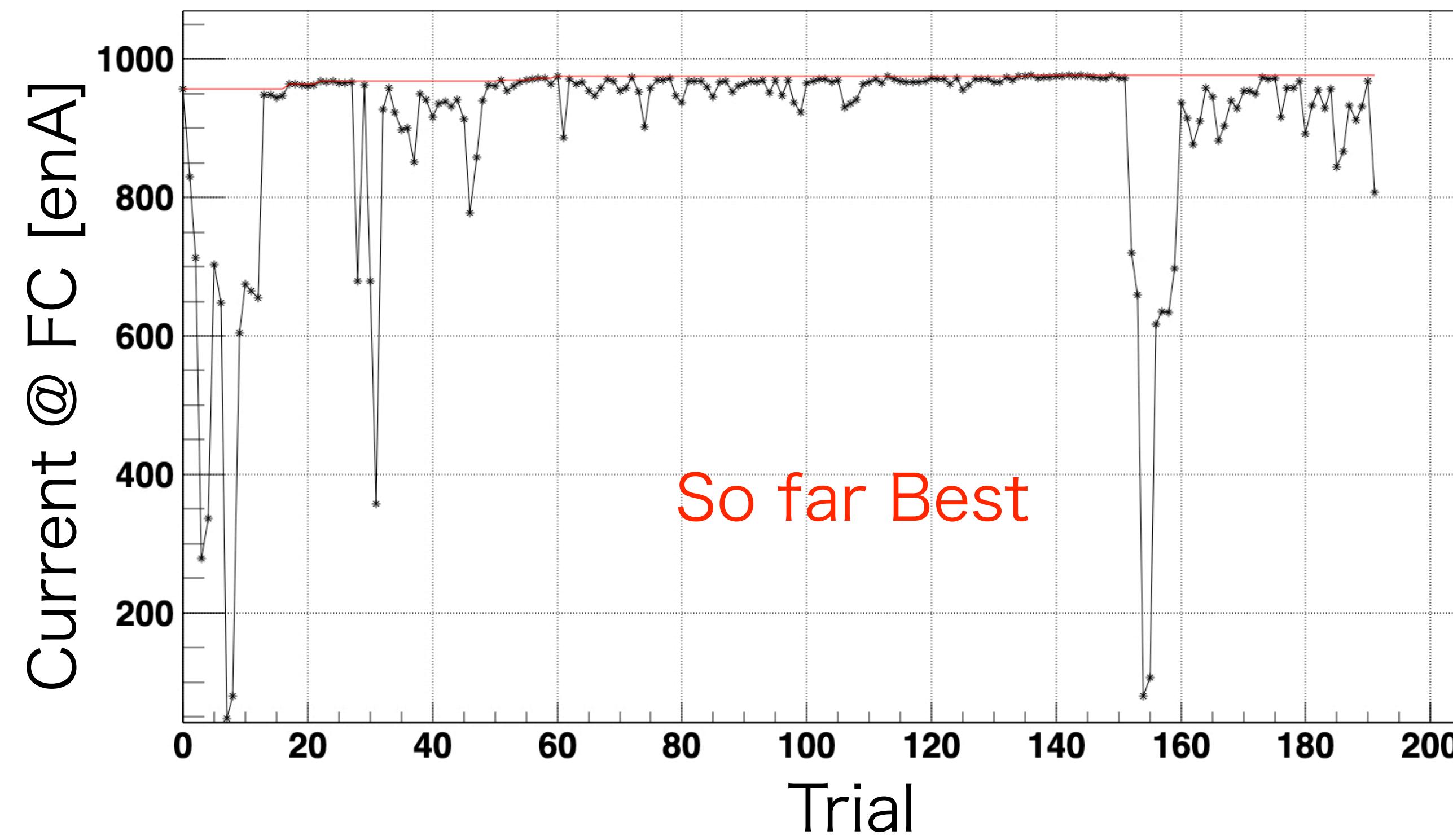




- ・ 総試行時間 ~ 20分 (完全自動化)
- ・ 総iTrial ~190回
(~ 30回試行 × 6 パートの最適化)
- ・ 測定値: 0.2 秒ごと3回のFCの測定の平均値
~ 数 enA のバタつき
- ・ 最終的に **956 enA → 976 enA (+2%)**

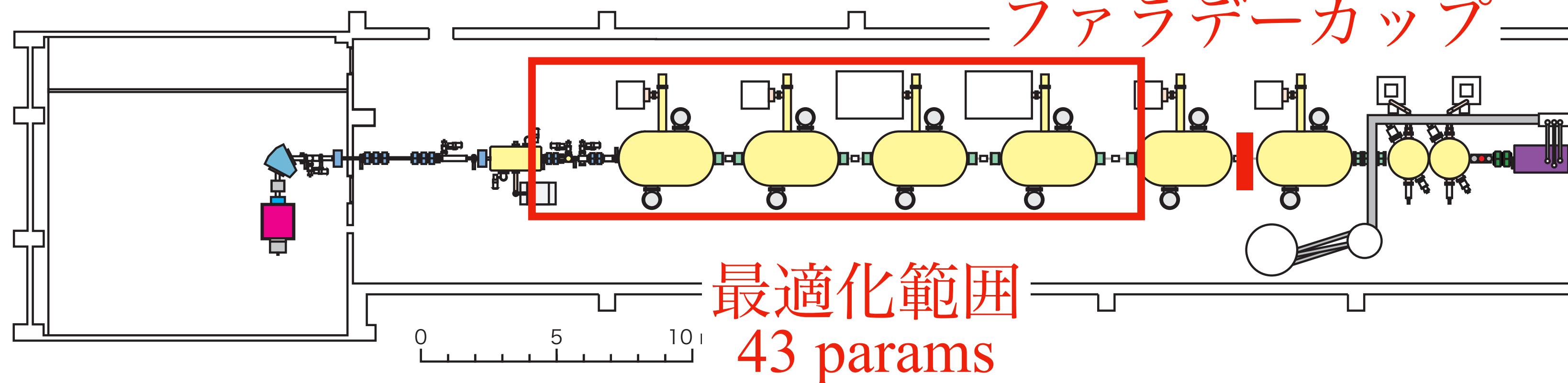
※ RF のパラメータを変更せず、また
ある程度オペレータによる最適化の
後であったことを考えれば悪くない結果



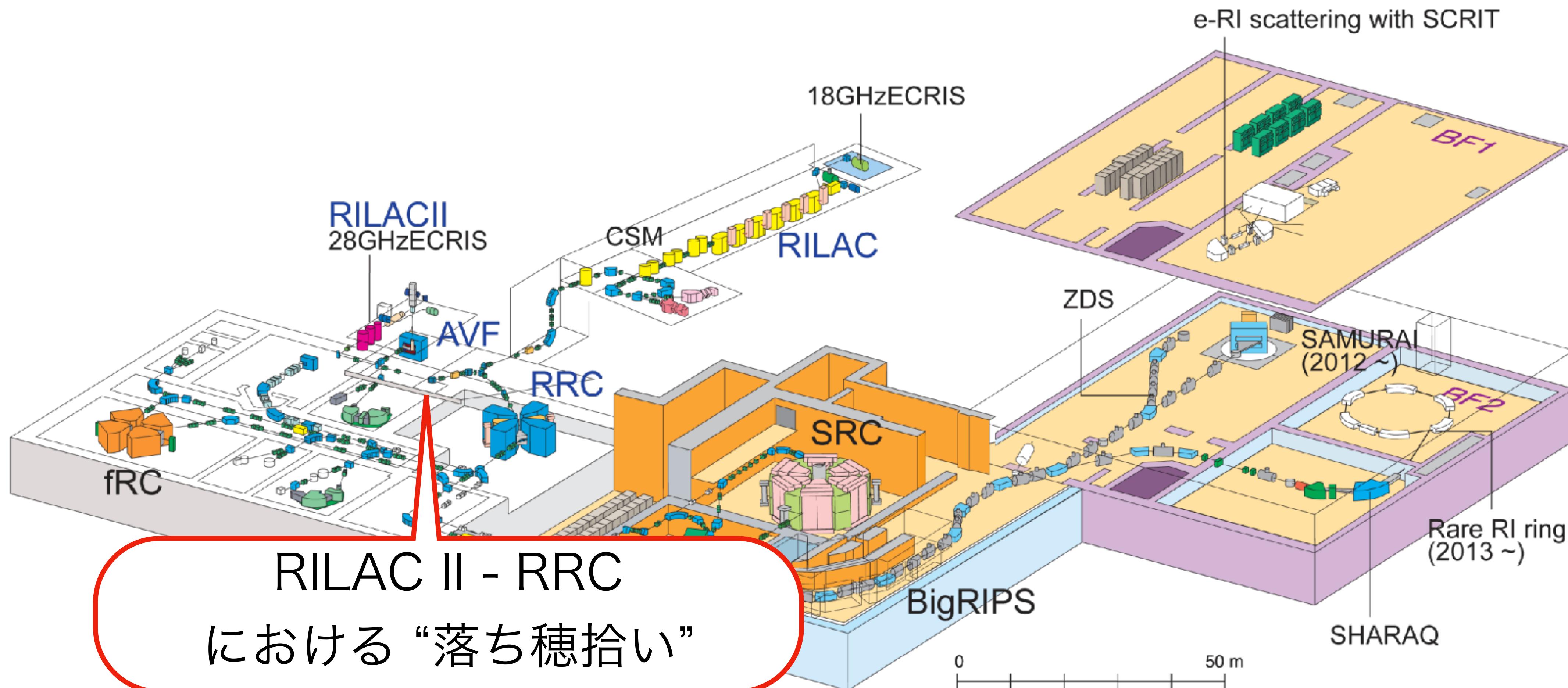


- ・ 総試行時間 ~ 20分 (完全自動化)
- ・ 総iTrial ~190回
(~ 30回試行 × 6 パートの最適化)
- ・ 測定値: 0.2 秒ごと3回のFCの測定の平均値
~ 数 enA のバタつき
- ・ 最終的に 956 enA → 976 enA (+2%)
- ・ **最大で 950 → 50 enA以下 まで低下**

**下流にある超伝導空洞まで
beam を通す場合、この
beamloss は許容されない**

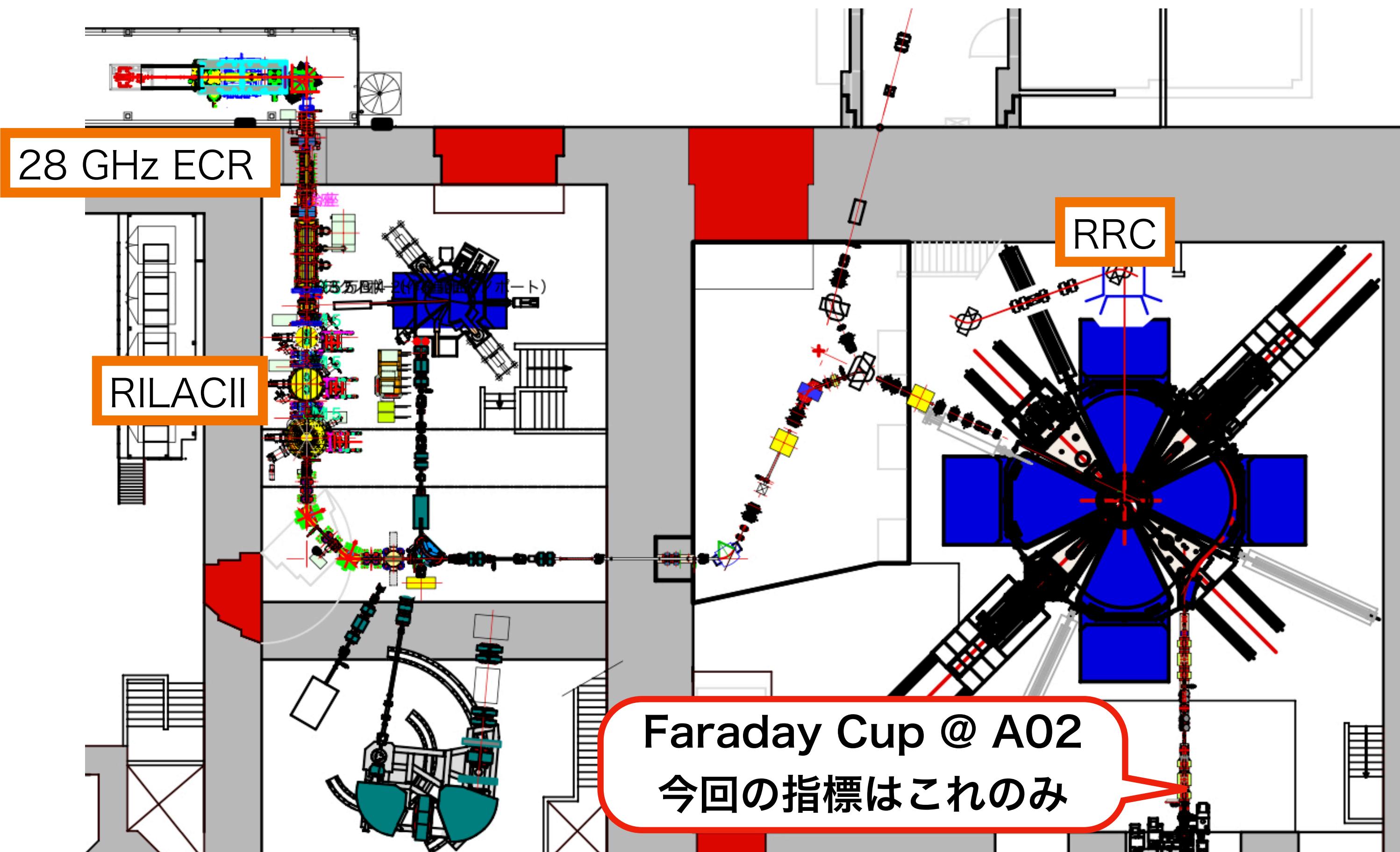


自動最適化のテスト@ RILAC / RILACII ~ RRC



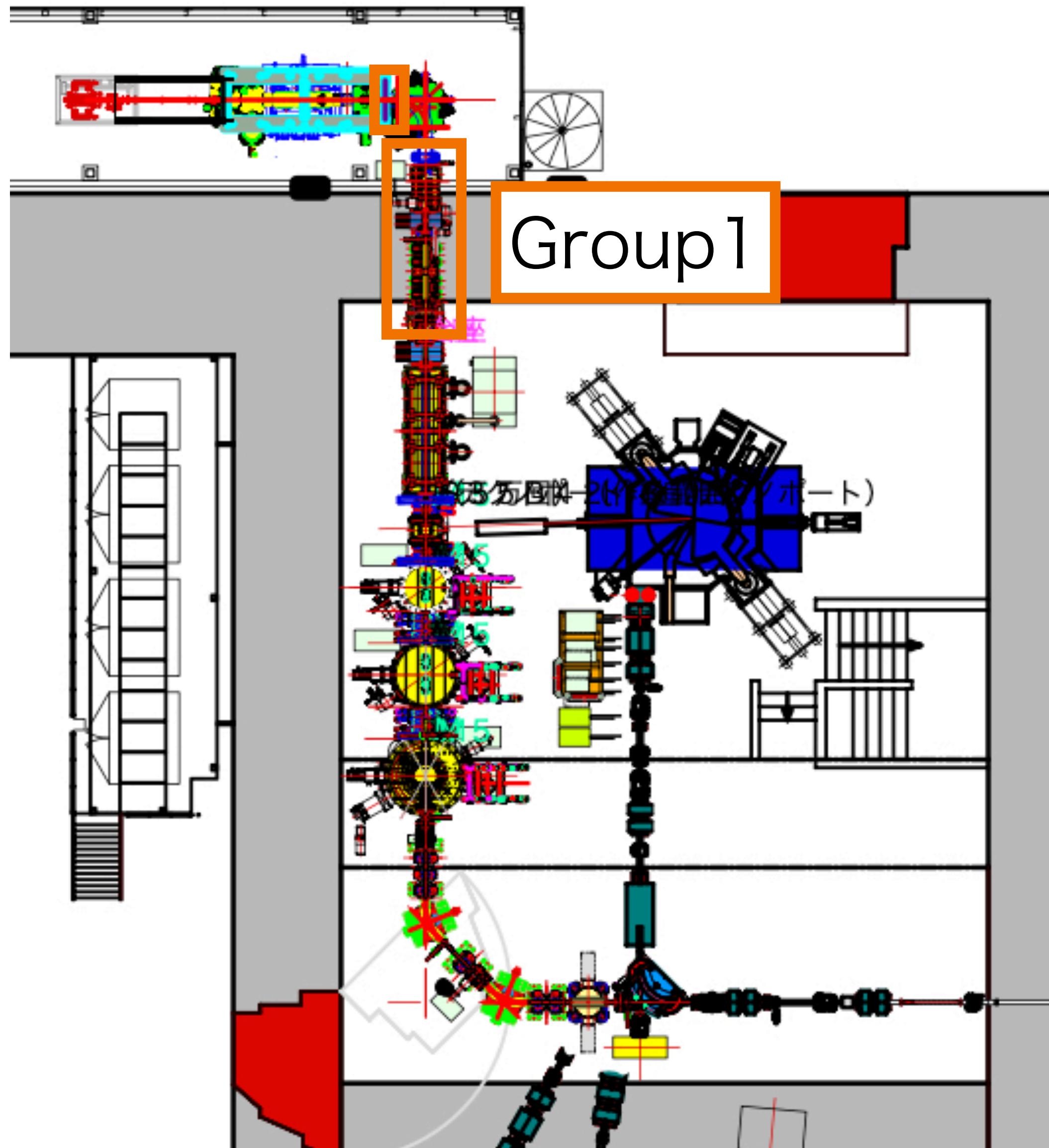
RILACII ~ RRC における”落ち穂拾い”

イオン源下流からサイクロトロン入射部の全 70 パラメータを最適化
 70 個のパラメータを 10 のパートに分割し、一度に最大10パラメータ調整
 指標はサイクロトロン下流 Faraday Cup のみ

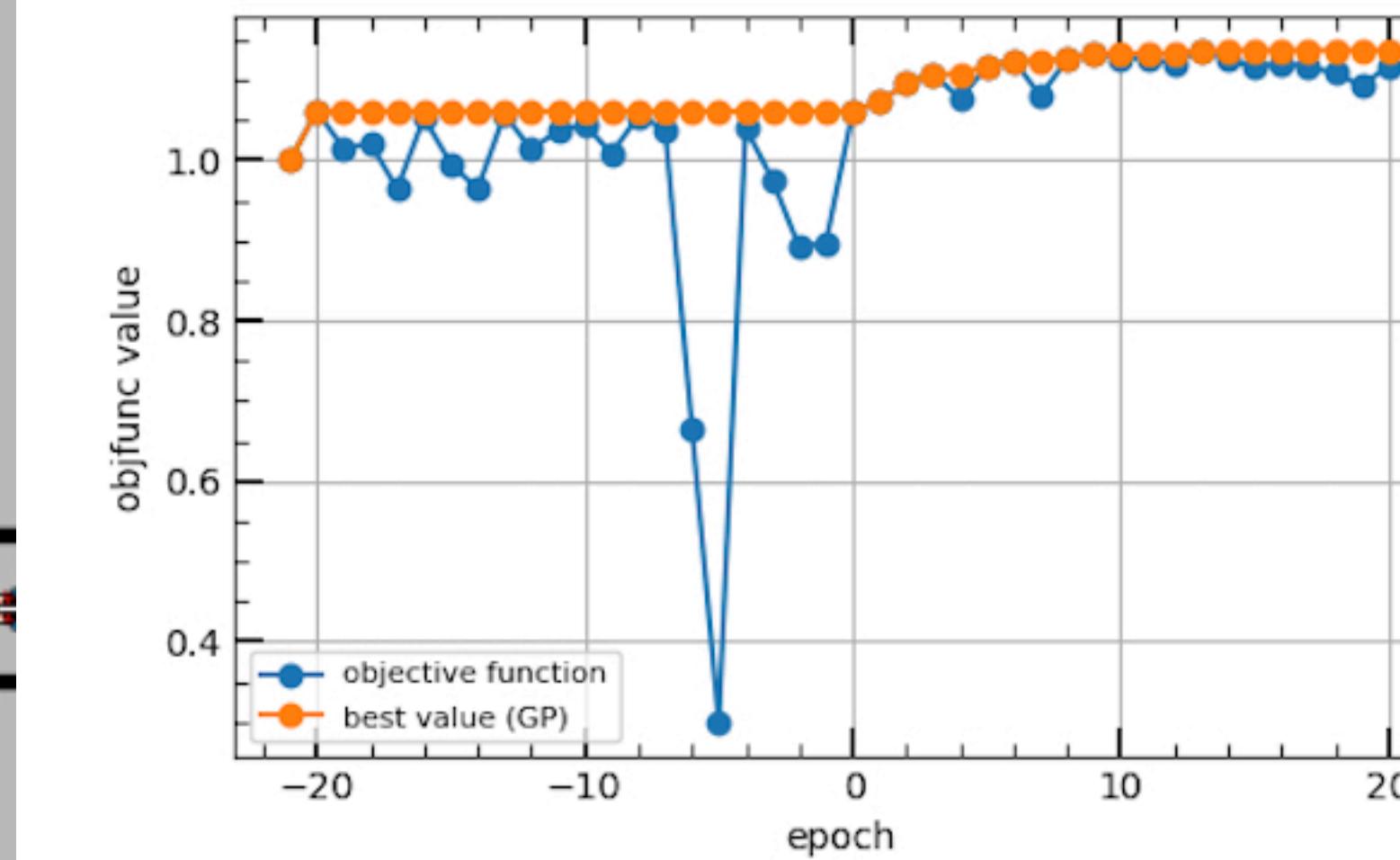


#	Quad	Steerer	Solenoid	Dipole	All
1	4	4	2	0	10
2	2	4	1	0	7
3	8	0	0	0	8
4	4	4	0	0	8
5	4	3	0	0	7
6	6	1	0	0	7
7	4	4	0	0	8
8	7	2	0	0	9
9	3	5	0	0	8
10	0	5	0	2	5

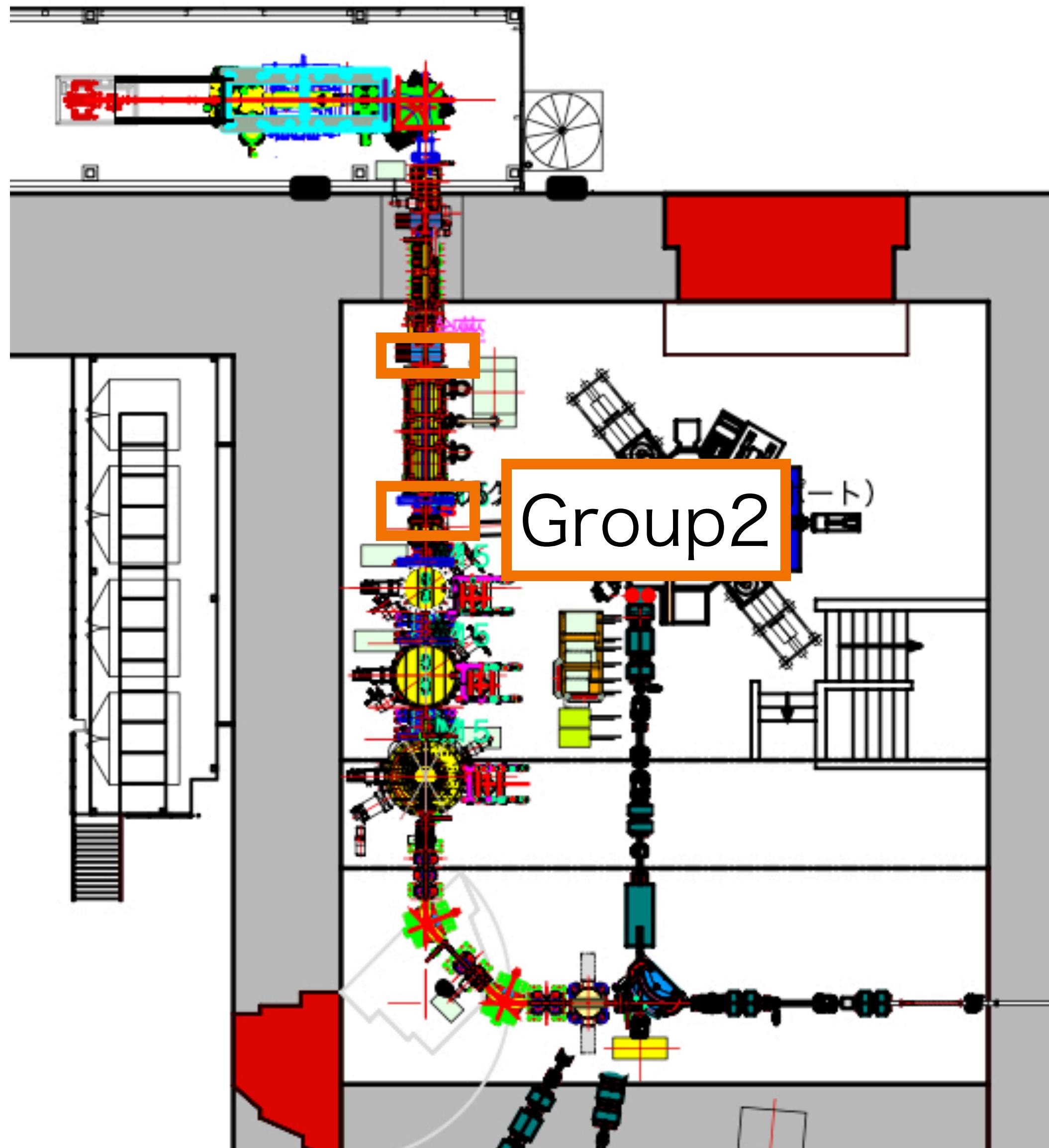
RILACII ~ RRC における”落ち穂拾い”



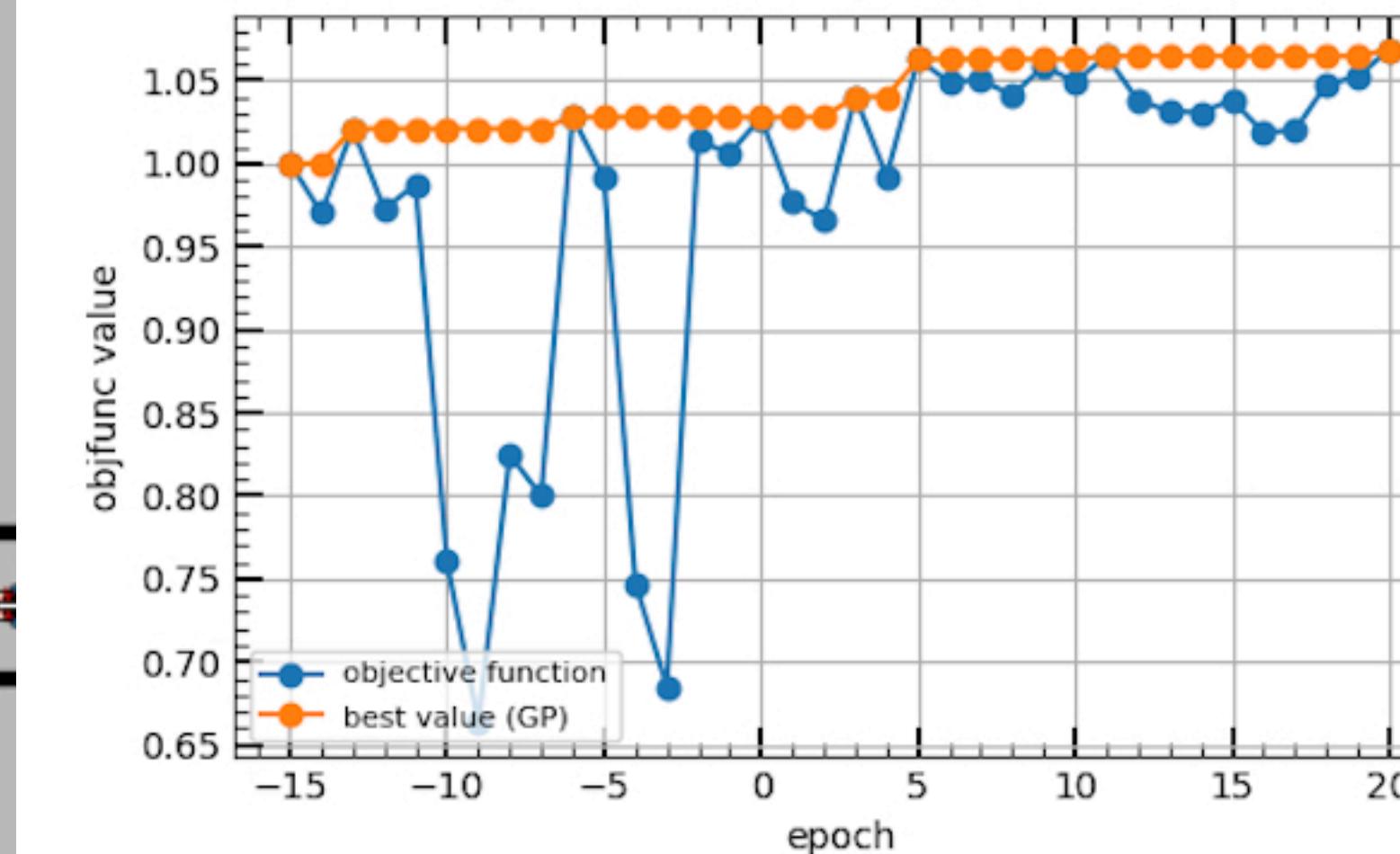
#	Quad	Steerer	Solenoid	Dipole	All
1	4	4	2	0	10
2	2	4	1	0	7
3	8	0	0	0	8
4	4	4	0	0	8
5	4	3	0	0	7



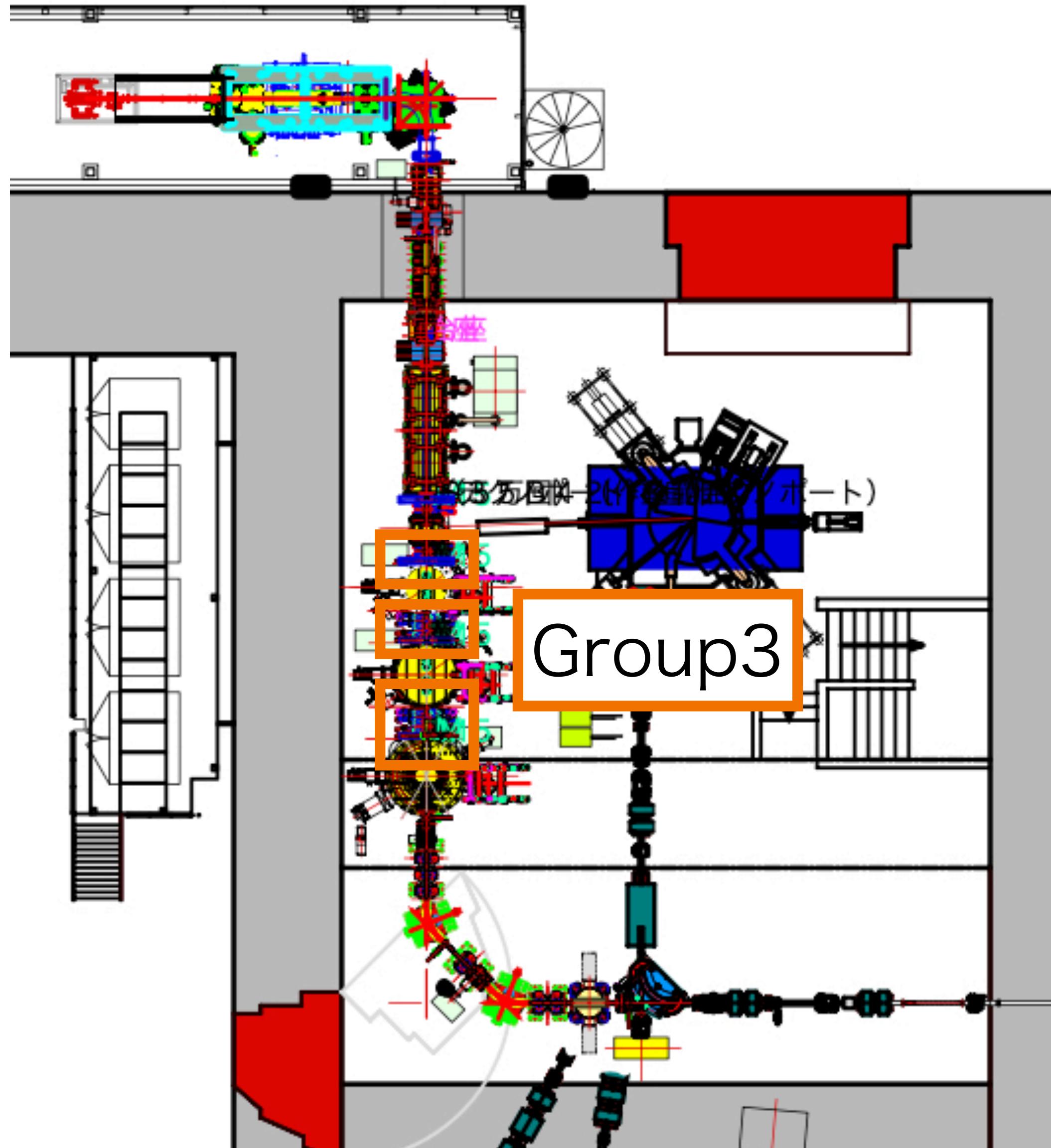
RILACII ~ RRC における”落ち穂拾い”



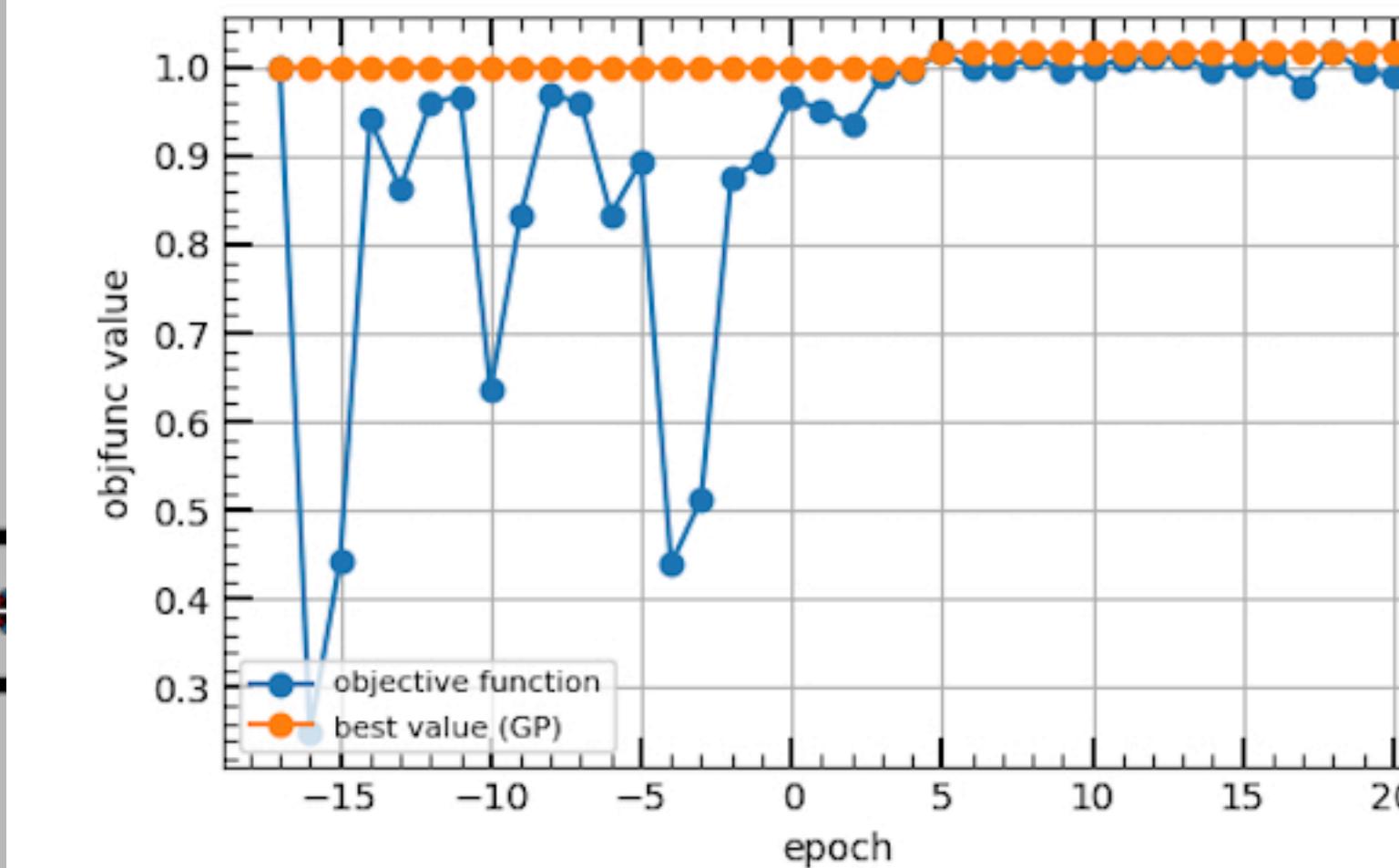
#	Quad	Steerer	Solenoid	Dipole	All
1	4	4	2	0	10
2	2	4	1	0	7
3	8	0	0	0	8
4	4	4	0	0	8
5	4	3	0	0	7



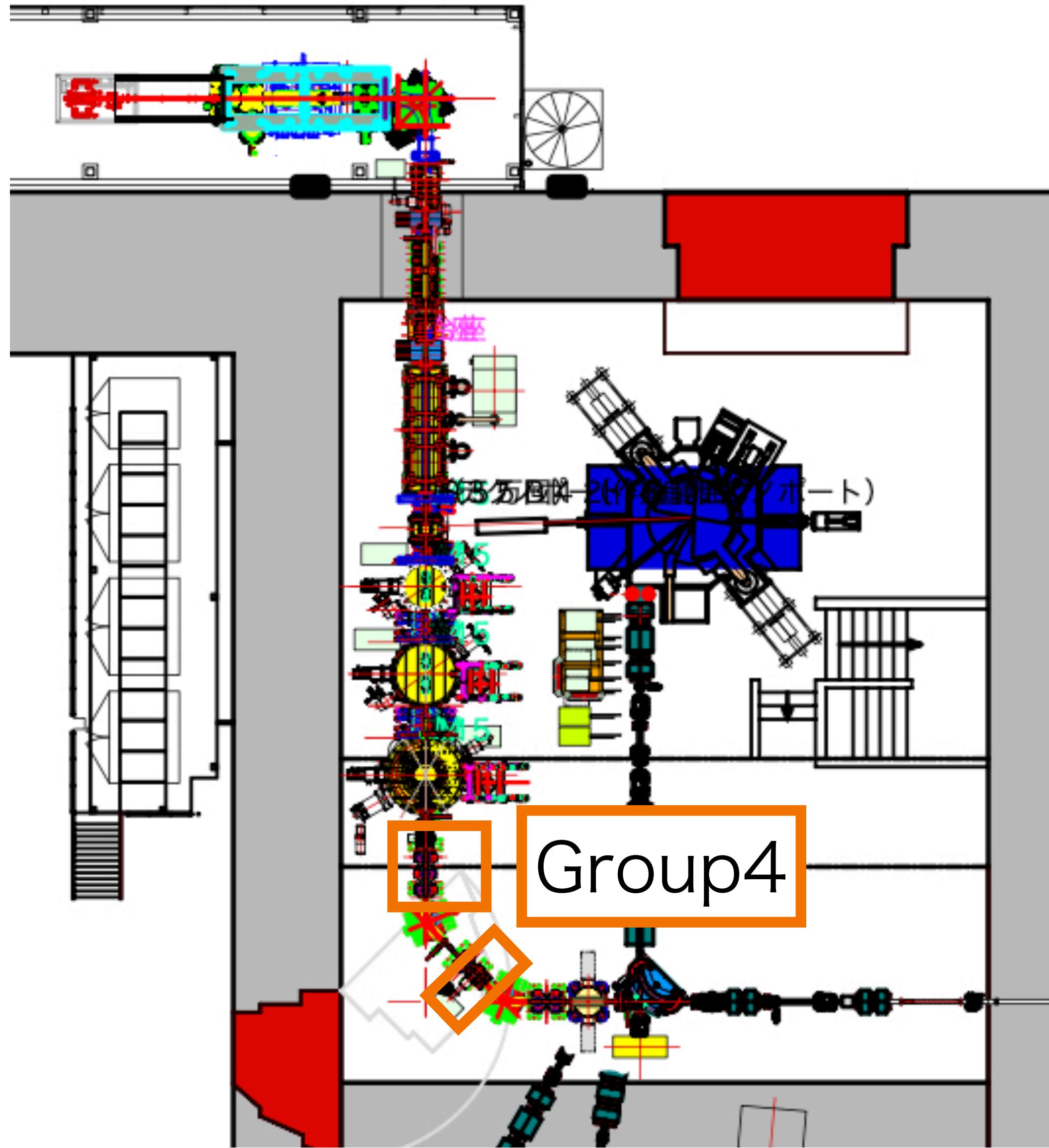
RILACII ~ RRC における”落ち穂拾い”



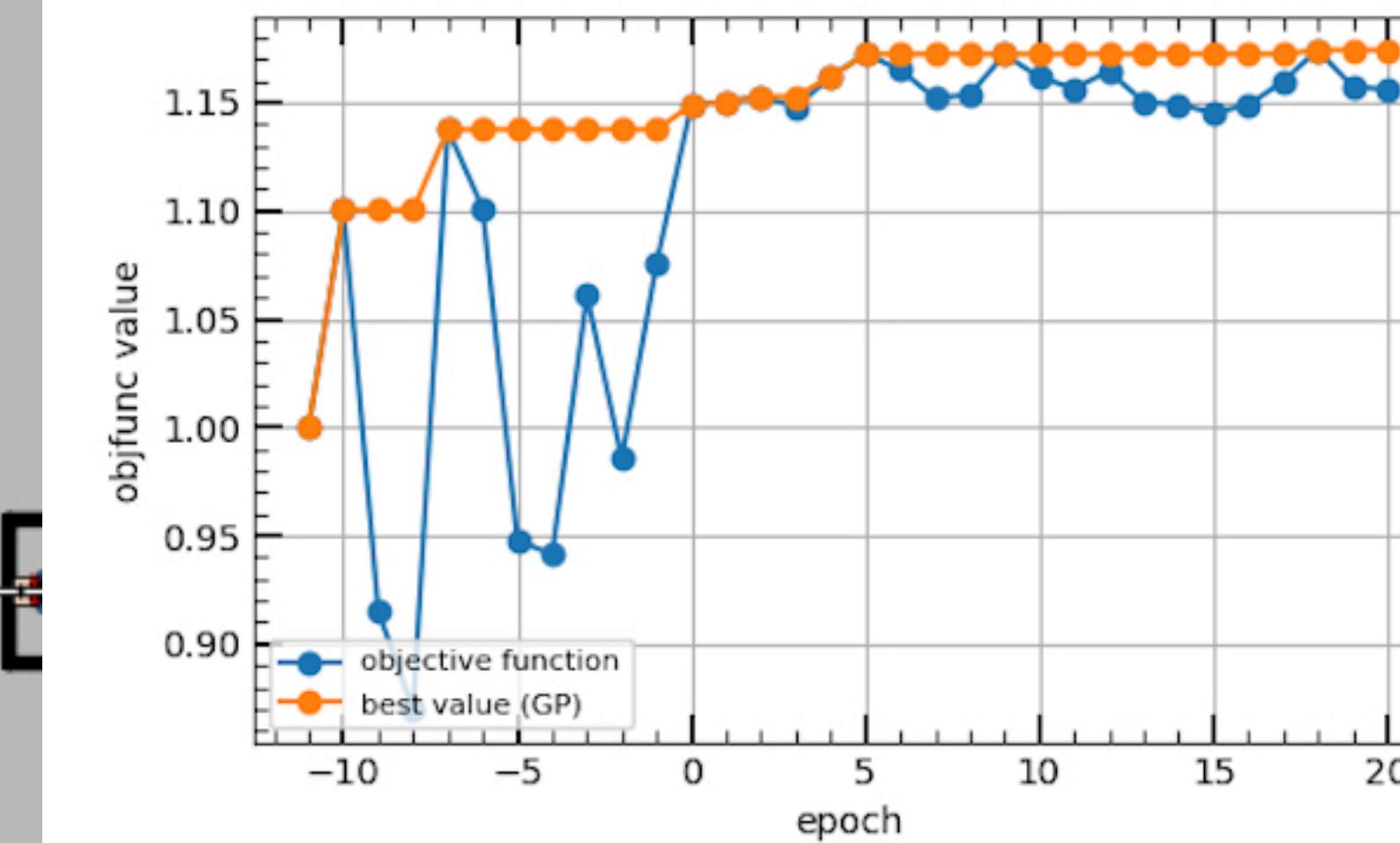
#	Quad	Steerer	Solenoid	Dipole	All
1	4	4	2	0	10
2	2	4	1	0	7
3	8	0	0	0	8
4	4	4	0	0	8
5	4	3	0	0	7



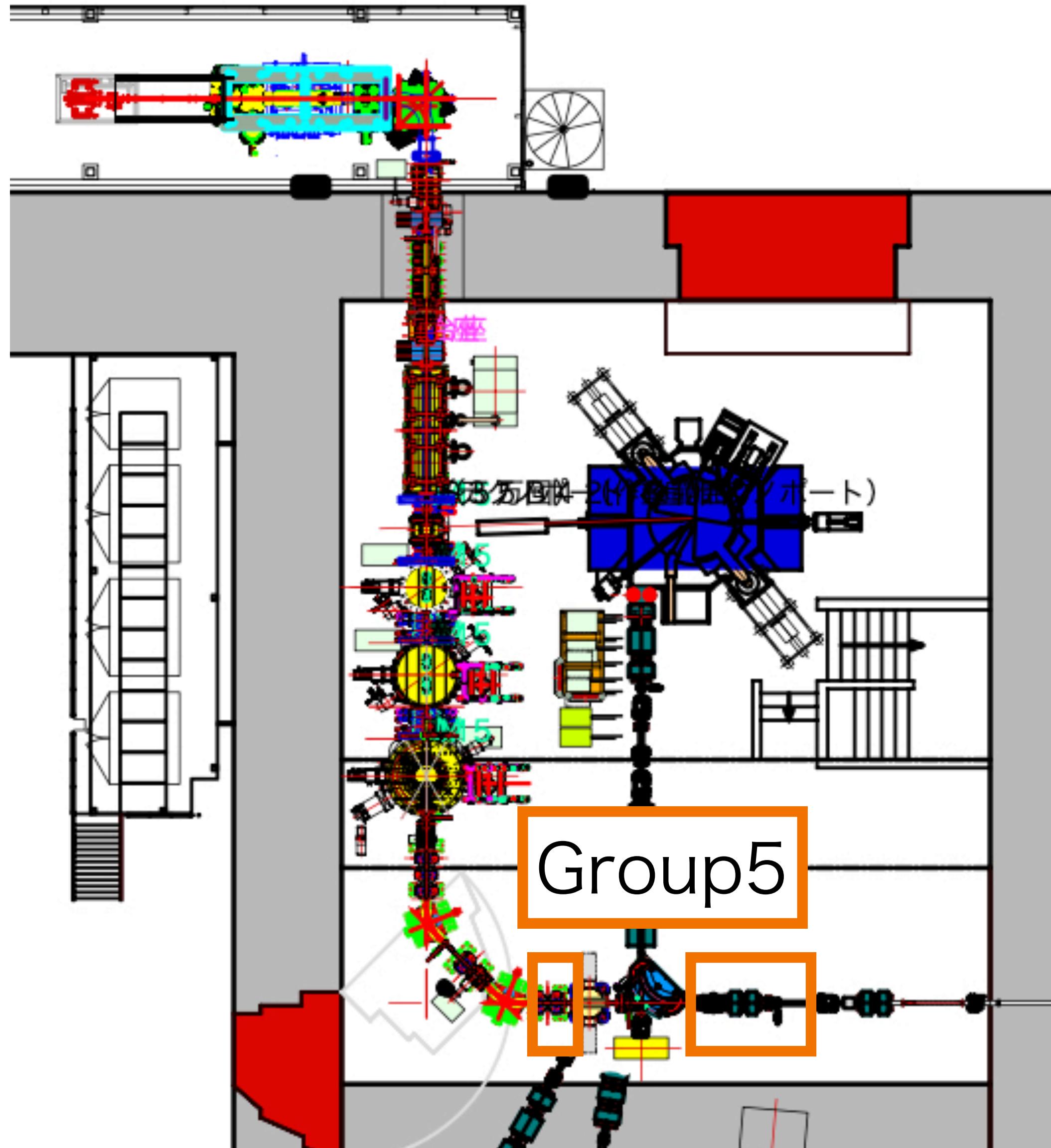
RILACII ~ RRC における”落ち穂拾い”



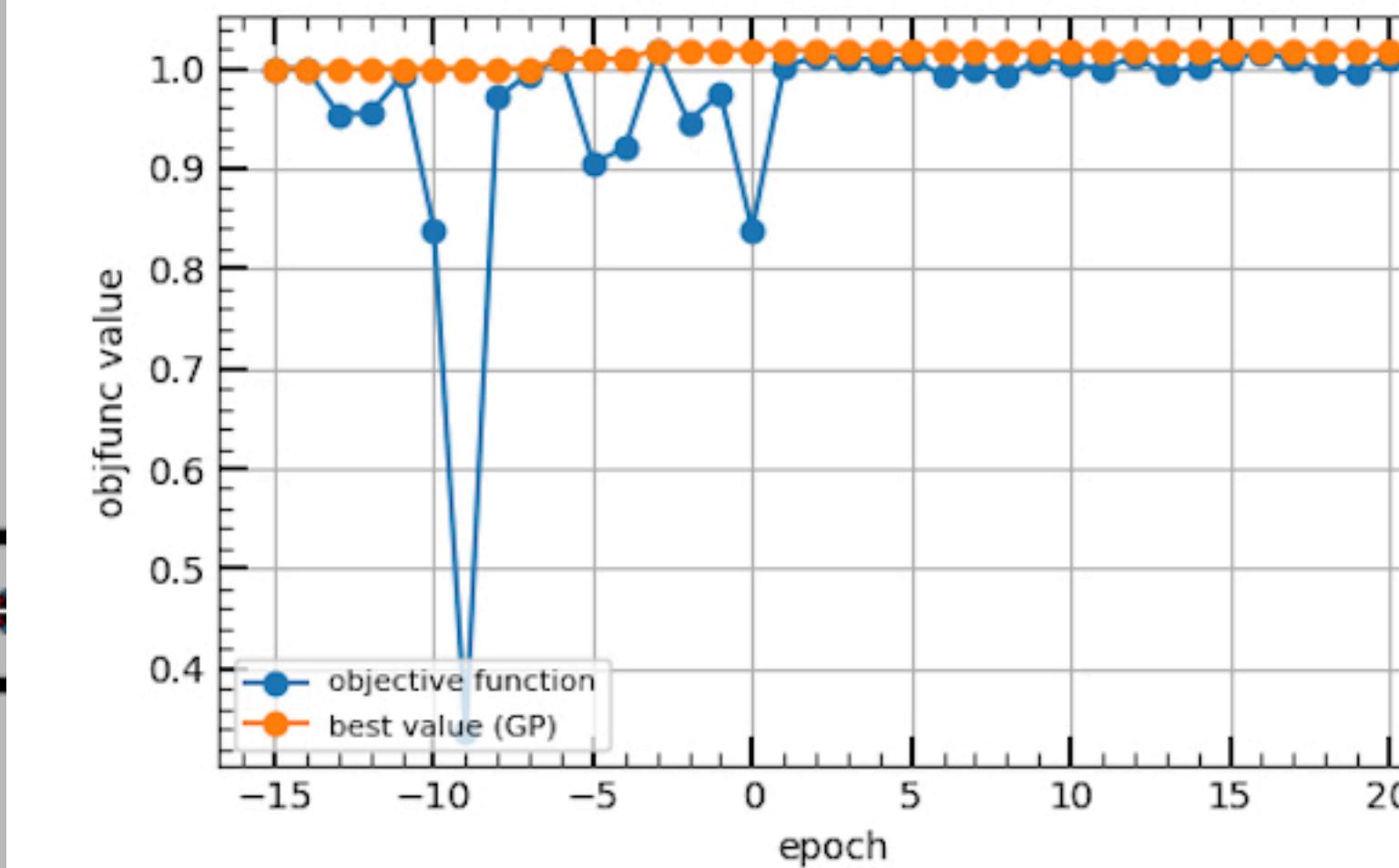
#	Quad	Steerer	Solenoid	Dipole	All
1	4	4	2	0	10
2	2	4	1	0	7
3	8	0	0	0	8
4	4	4	0	0	8
5	4	3	0	0	7



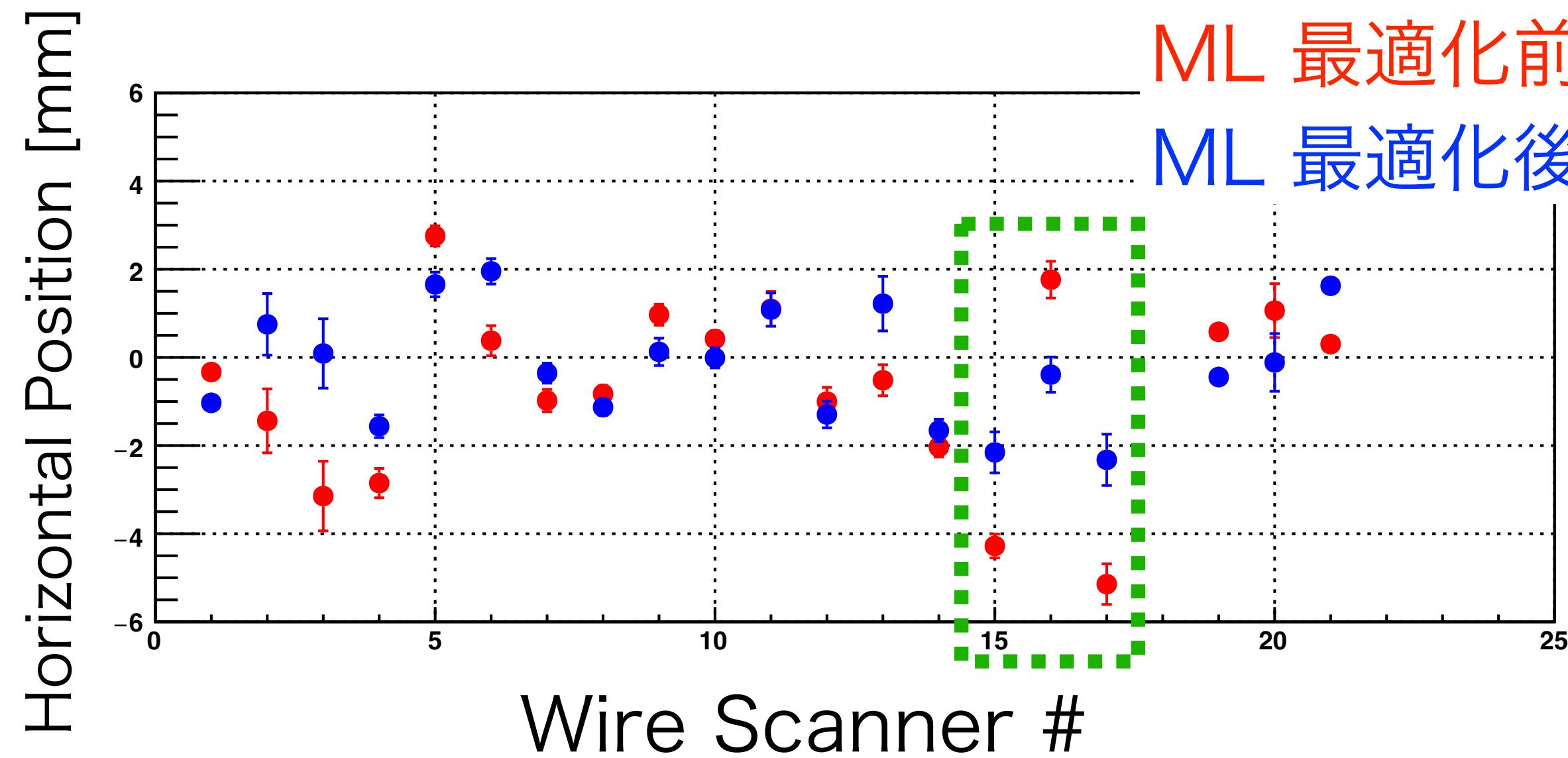
RILACII ~ RRC における”落ち穂拾い”



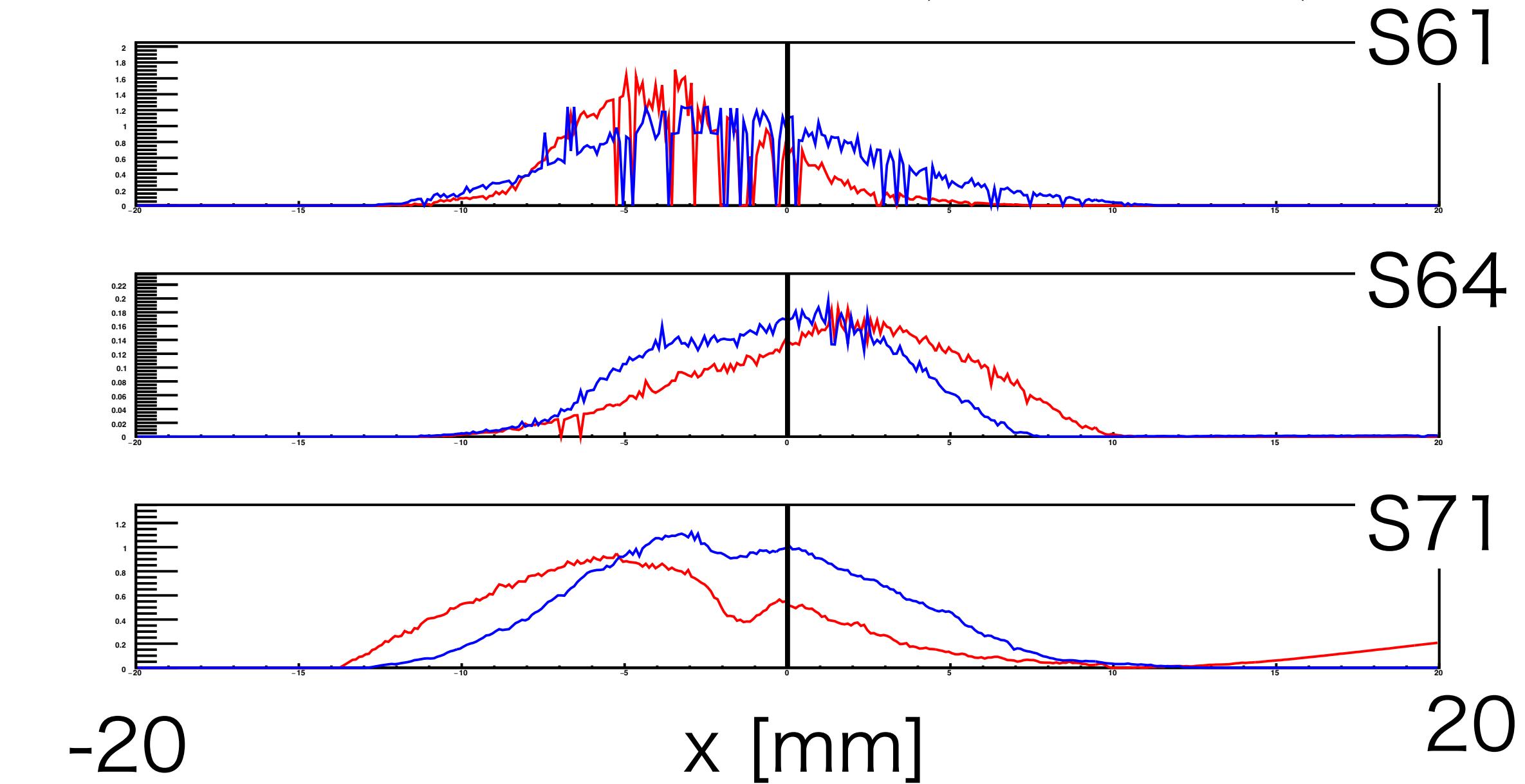
#	Quad	Steerer	Solenoid	Dipole	All
1	4	4	2	0	10
2	2	4	1	0	7
3	8	0	0	0	8
4	4	4	0	0	8
5	4	3	0	0	7



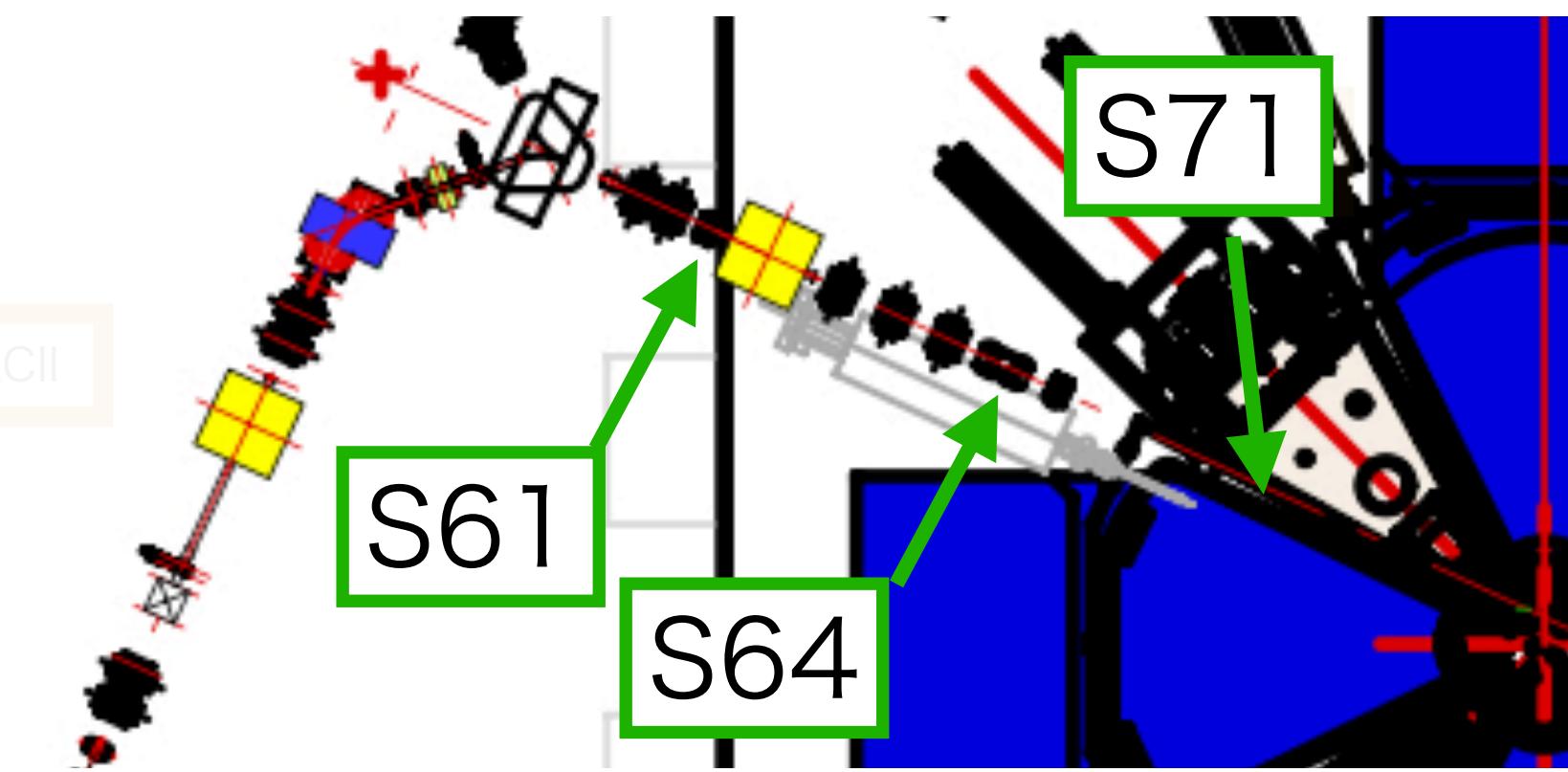
RILACII ~ RRC における”落ち穂拾い”



Beam Profiles (Horizontal)

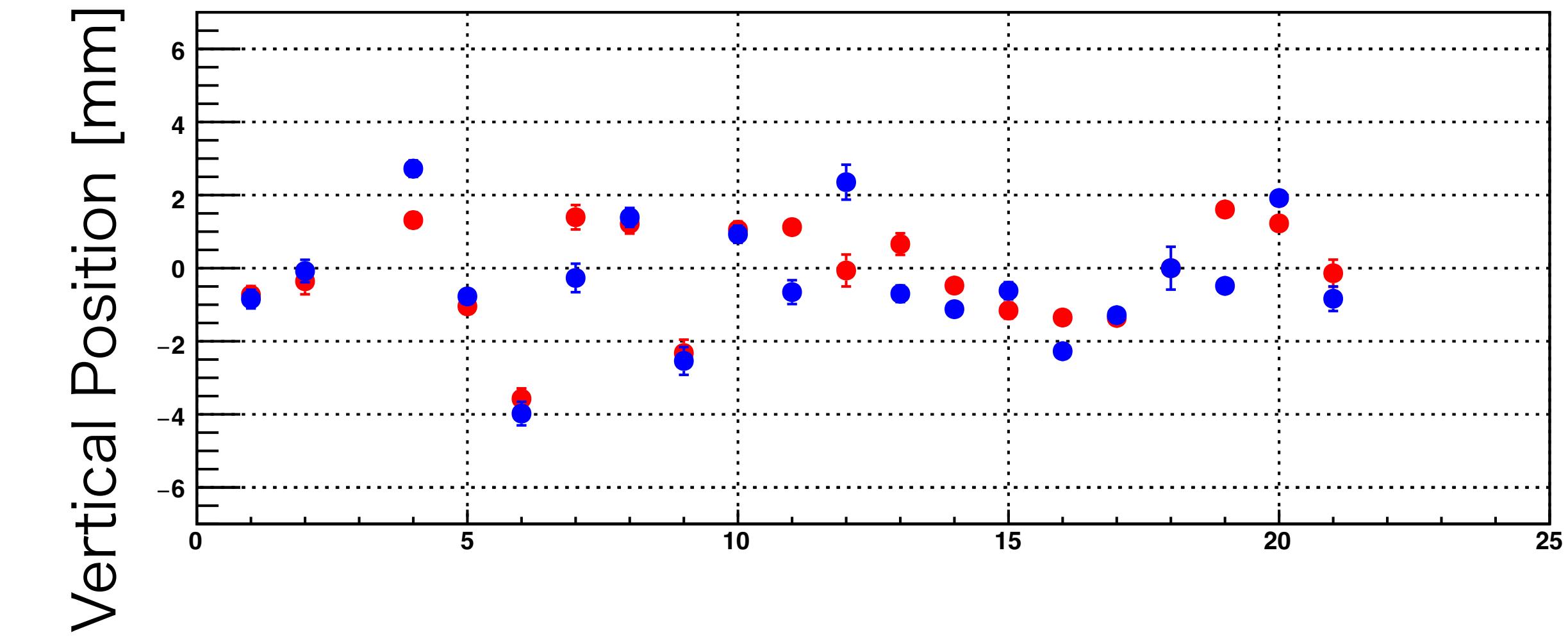
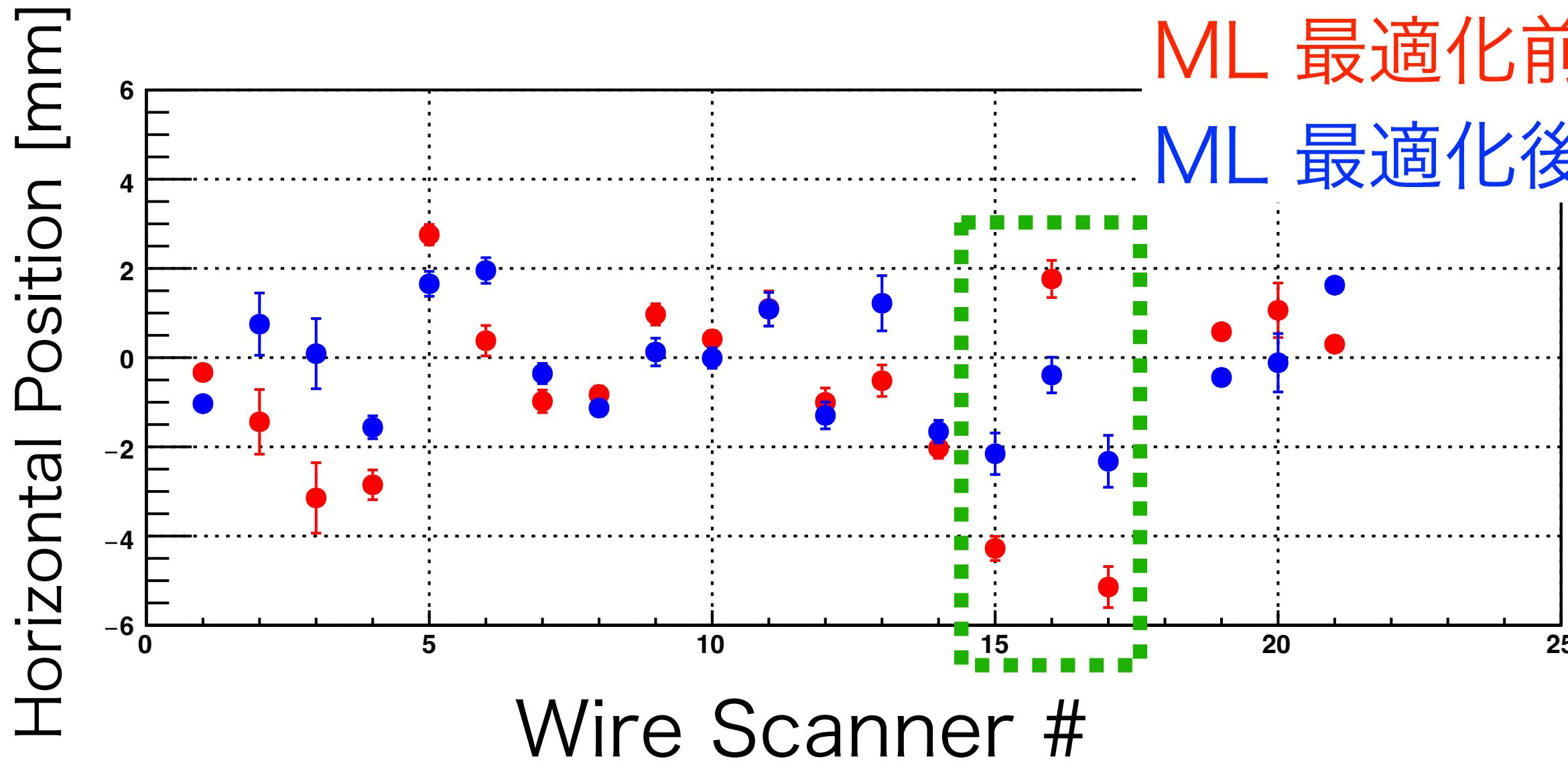


- 今回は Profile の軸は見ず、あくまで通過率のみ
しかし、最適化前後を見ると特に一部で beam の軸が
顕著に 0 に近づいていることが分かる。

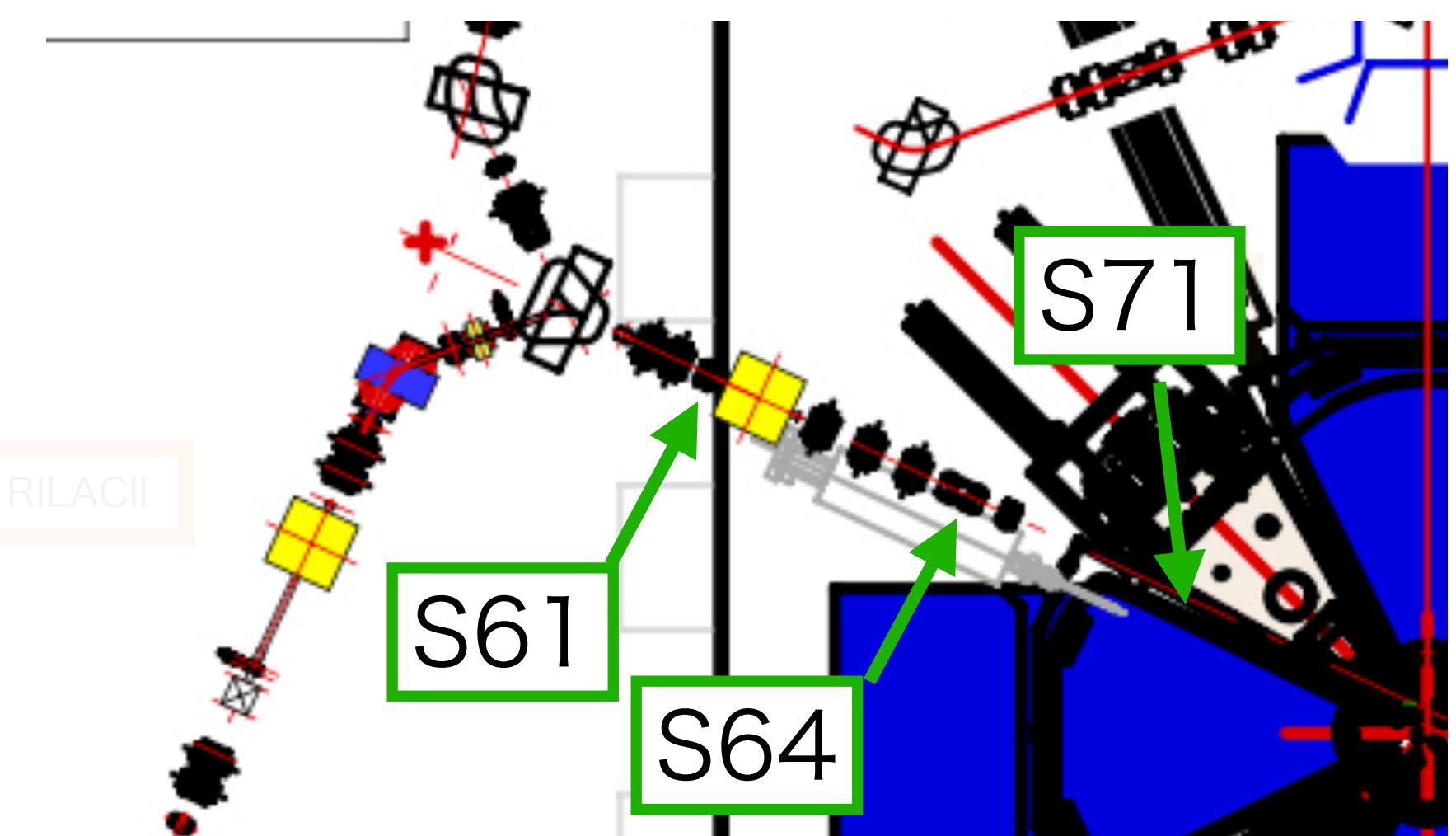


Faraday Cup @ A02
今回の指標はこれのみ

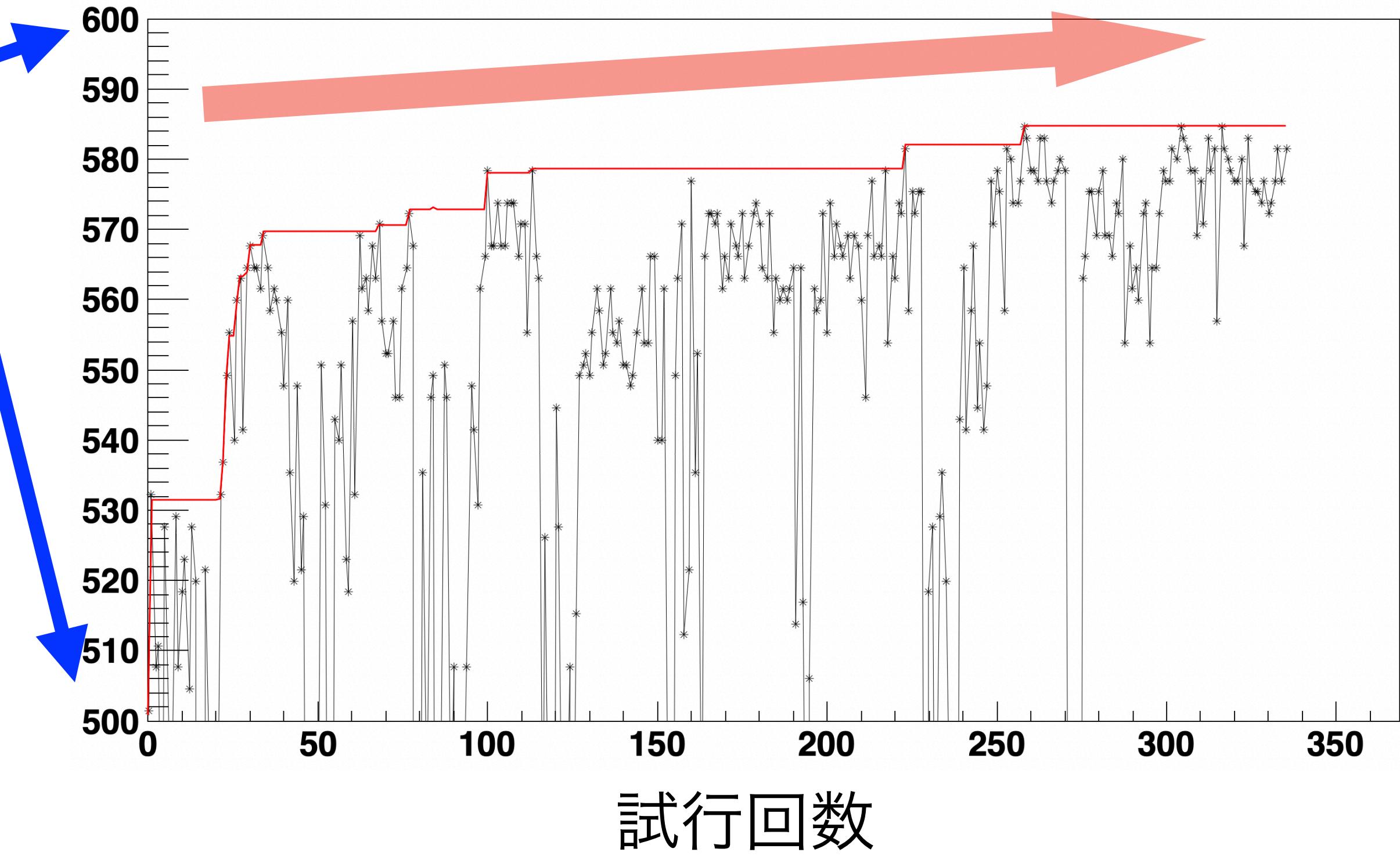
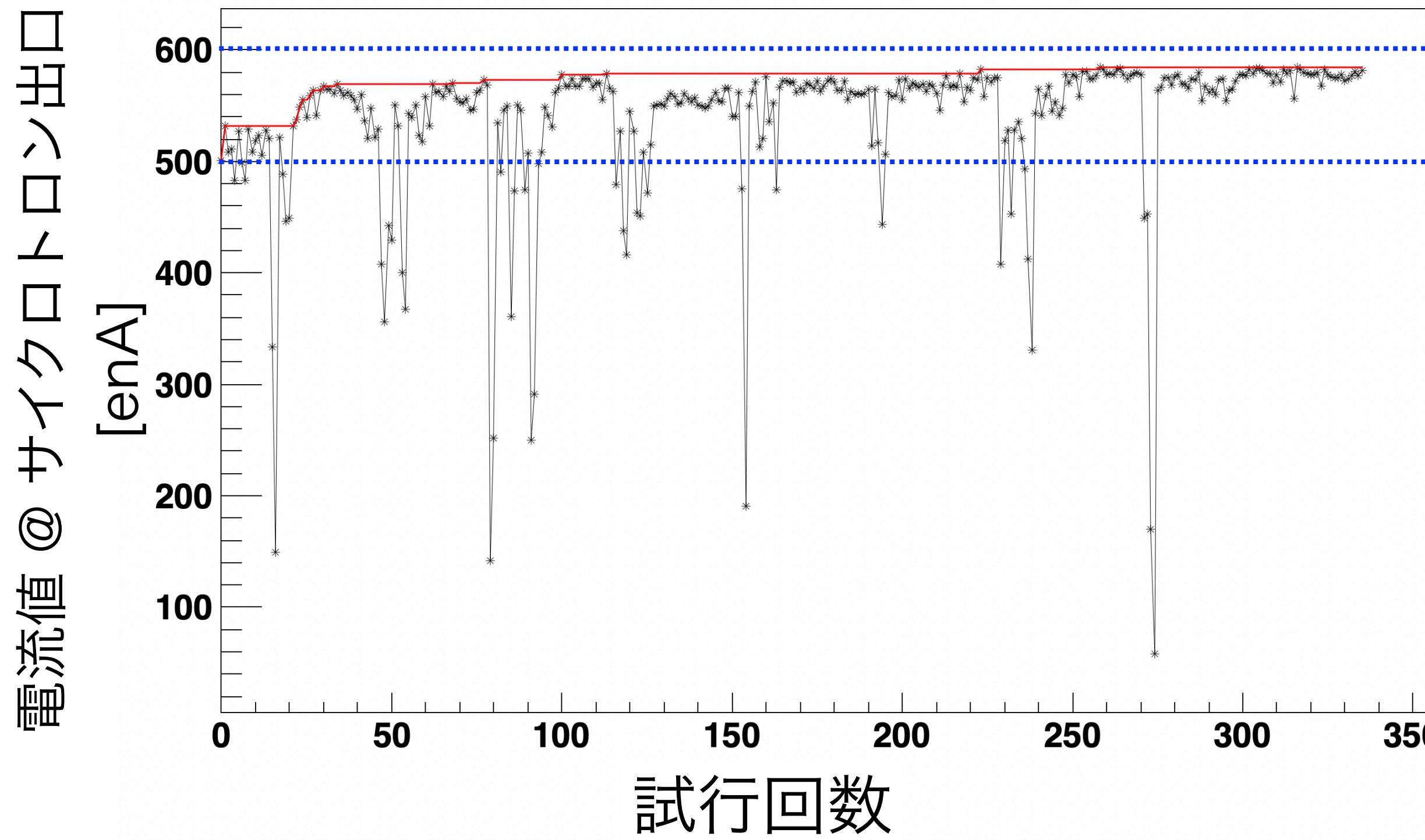
RILACII ~ RRC における”落ち穂拾い”



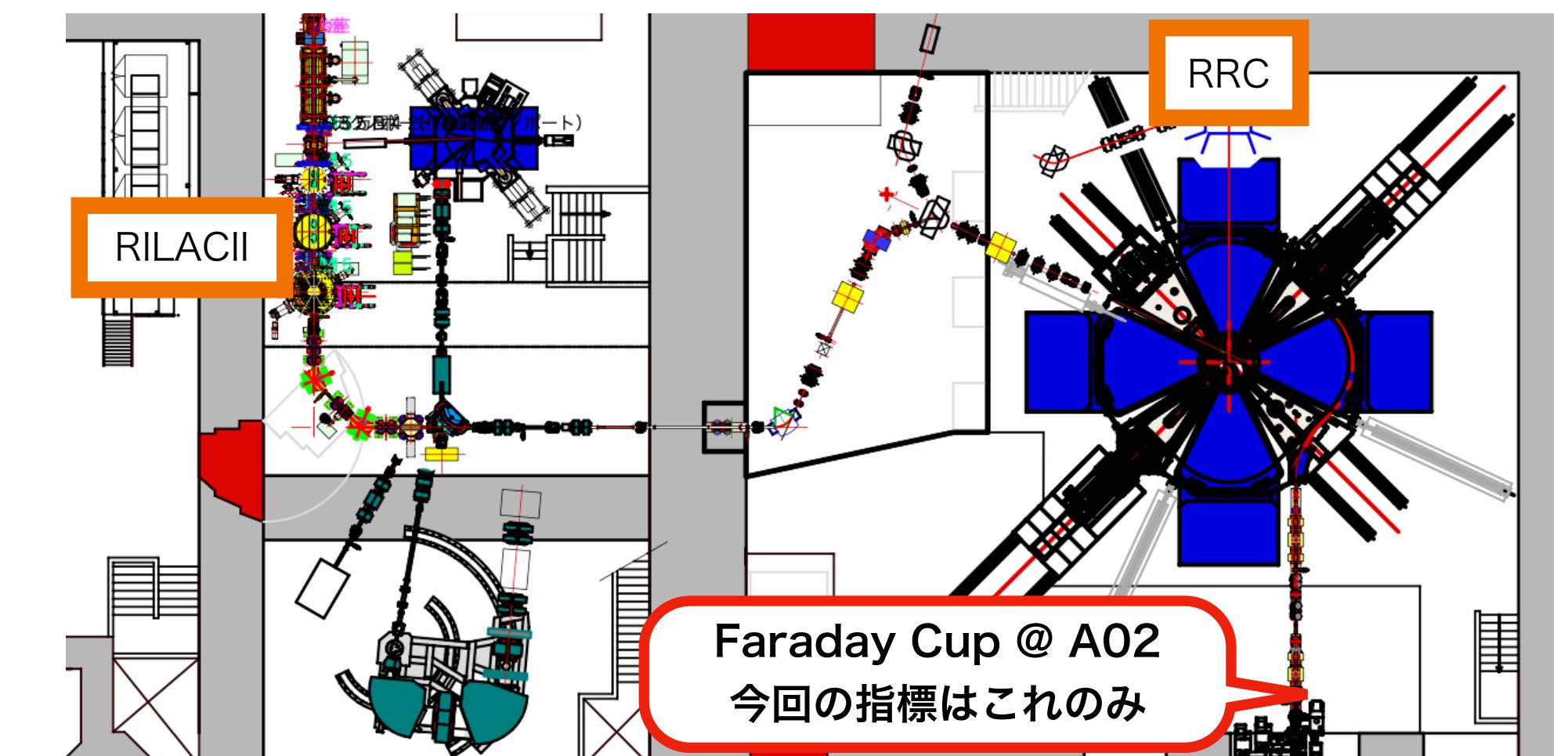
- 今日は Profile の軸は見ず、あくまで通過率のみしかし、最適化前後を見ると特に一部で beam の軸が顕著に 0 に近づいていることが分かる。
- なお、垂直方向については顕著な差は見られず。



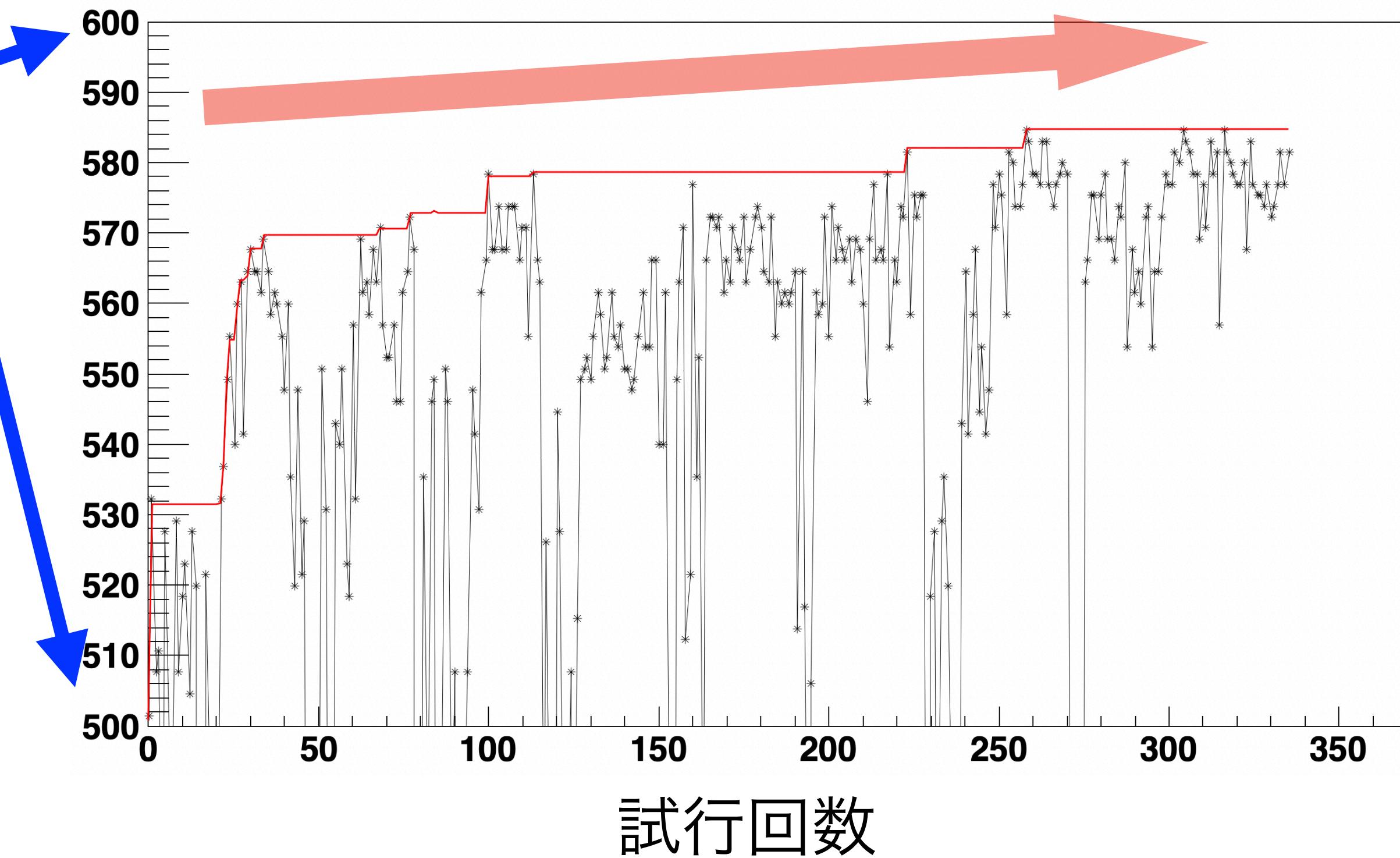
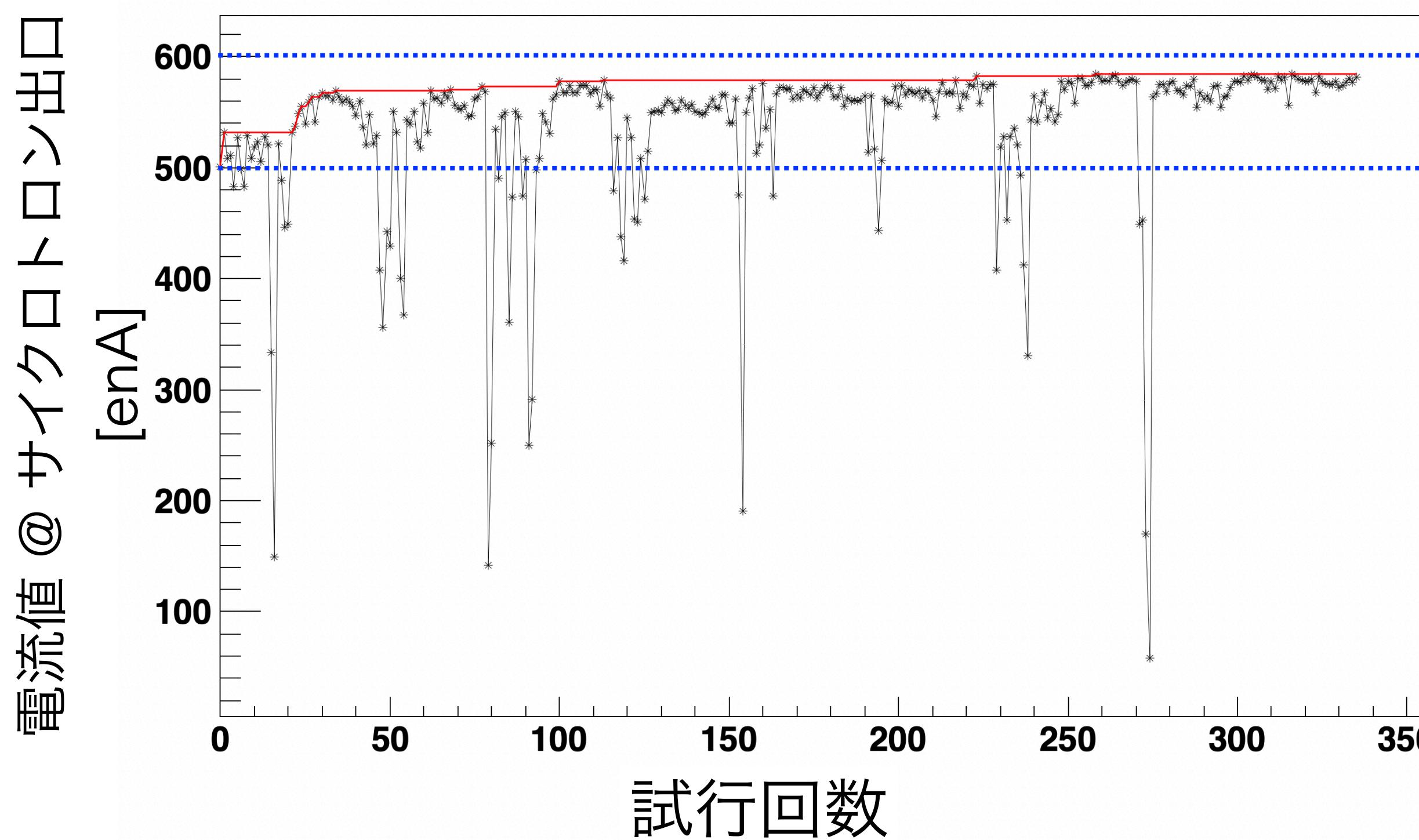
RILACII ~ RRC における”落ち穂拾い”



- ・測定値: 0.2 秒ごと5回のFCの測定の平均値
～数 enA のバタつき
- ・70 パラメータ / 335 試行、36分 (1試行 ~ 6.5秒)
- ・Faraday Cup で電流値が
520 ~ 530 enA → 570 ~ 580 enA (~ 10% ほどの上昇)
- ・ただし、最低値は ~ 50 enA (90% 口ス !)



RILACII ~ RRC における”落ち穂拾い”

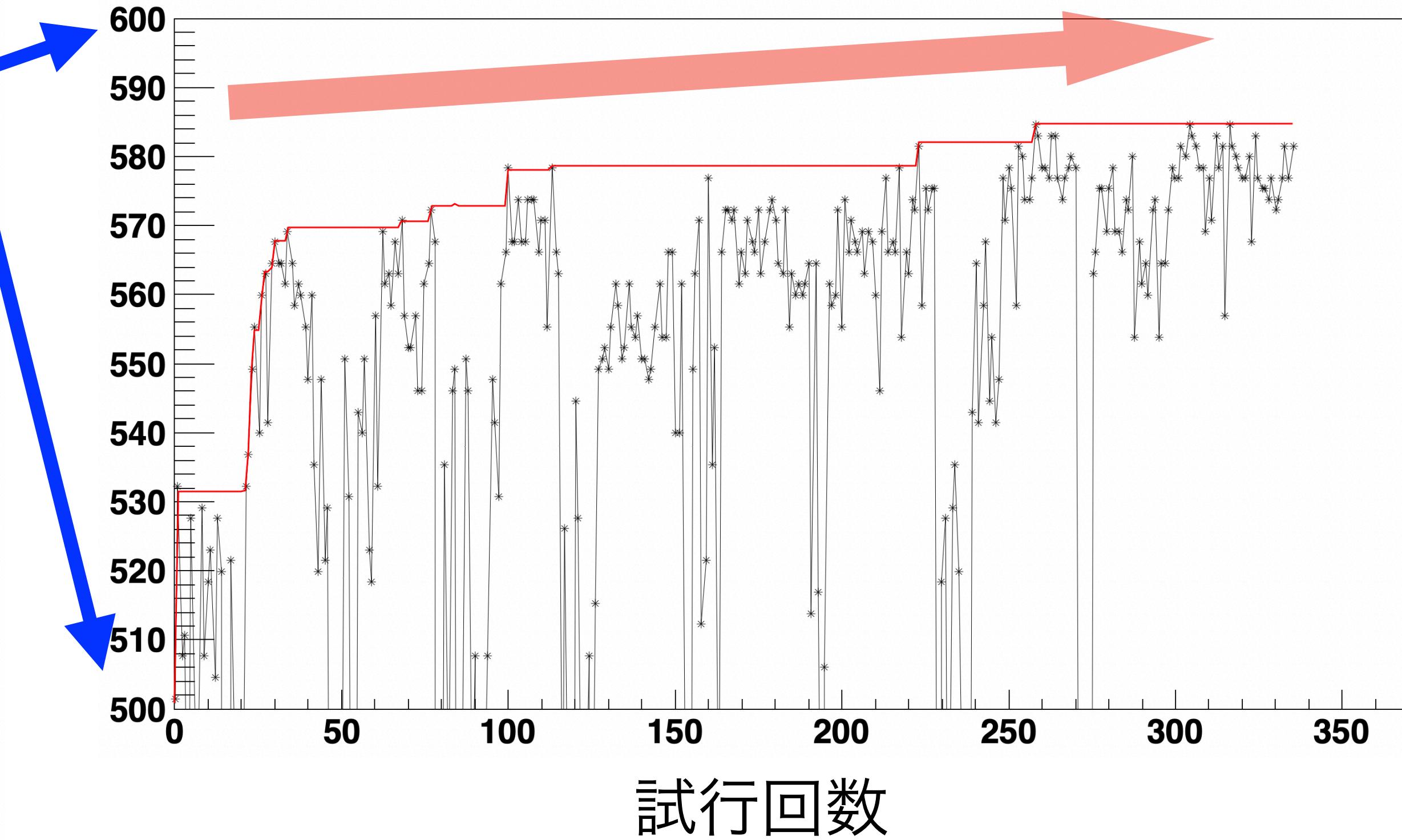
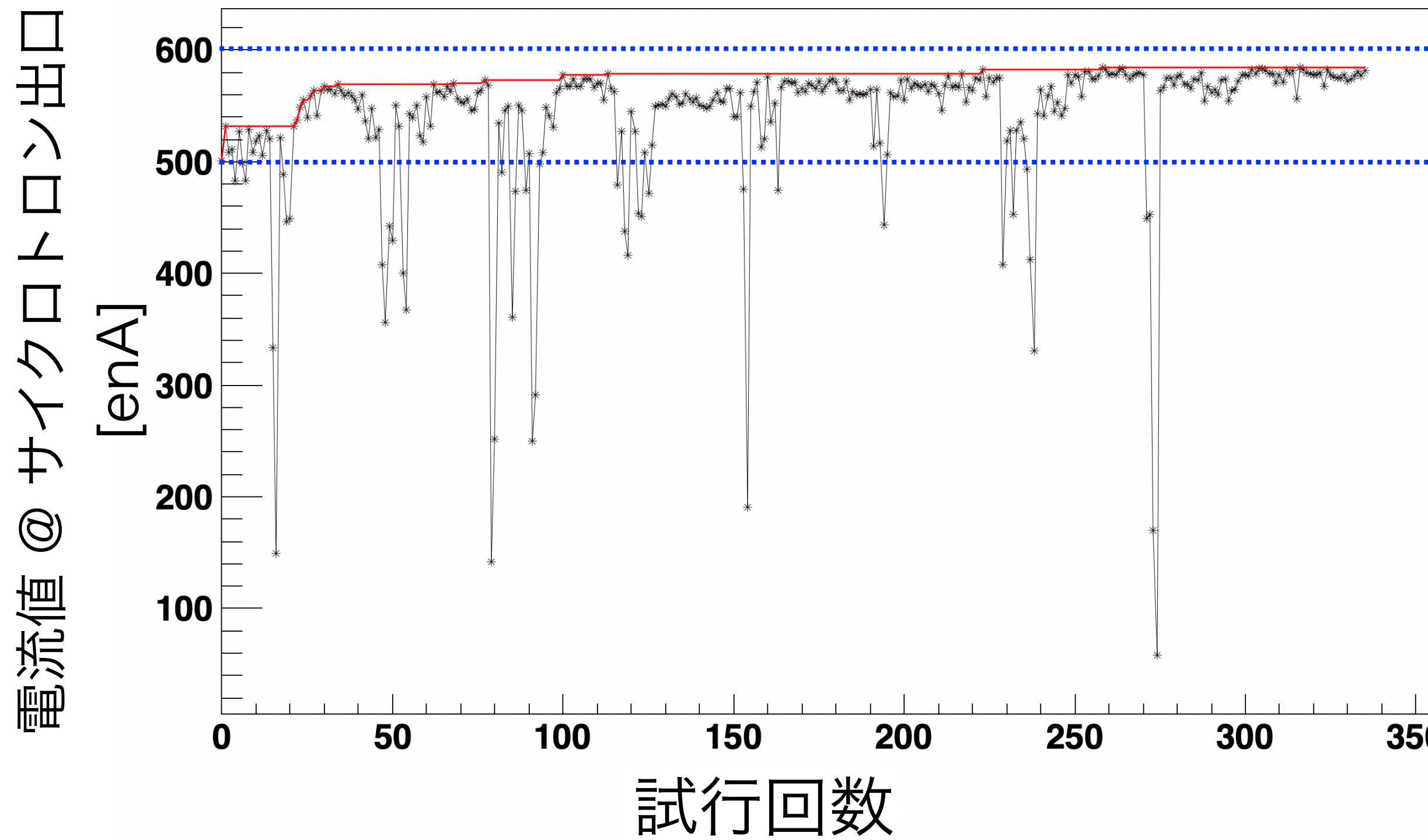


- ・測定値: 0.2 秒ごと3回のFCの測定の平均値
～数 enA のバタつき
- ・70 パラメータ / 335 試行、36分 (1試行 ~ 6.5秒)
- ・Faraday Cup で電流値が
520 ~ 530 enA → 570 ~ 580 enA (~ 10% ほどの上昇)
- ・ただし、最低値は ~ 50 enA (**90%ロス！**)

**RILAC と
ほぼ同じプログラム
制御・測定パラメータの名前と
範囲を変えただけで使える**



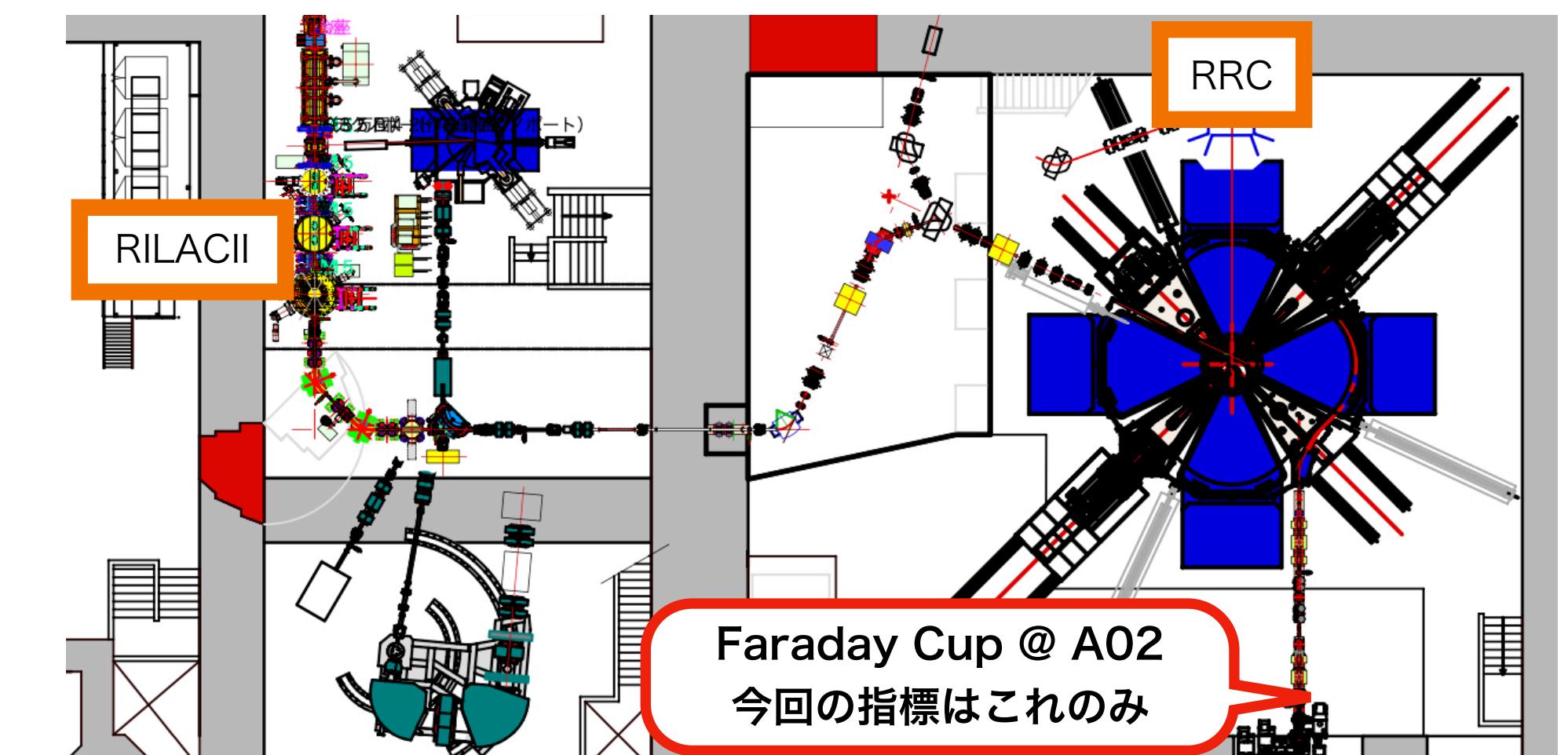
RILACII ~ RRC における”落ち穂拾い”



その他実用的な話として

- UI をちゃんと作る必要がある。
- 途中でセットした電流値が入らない機器がある。
- 他の要因でビームの電流値が変わることがある。
- パラメータのレンジを適切にしておかないと最適化がすすまない。

などがあるので、このあたりをクリアする必要あり。



多数のパラメータの同時最適化をどうやって実現するか？

- ・例えばRIBFだとパラメータは数百を超える。
- ・ガウス過程回帰をつかったベイズ最適化では、せいぜい 20 パラメータ？
(範囲にもよるが、以前行った最適化は 4 パラメータで 30分)

解決方法 ①: グループ分けをして少数のパラメータずつ最適化
→通称 “落ち穂拾い”

Pros: 何をやっているかがクリア。人がやっていることの自然な延長。

Cons: 異なるグループのパラメータを同時最適化できない。

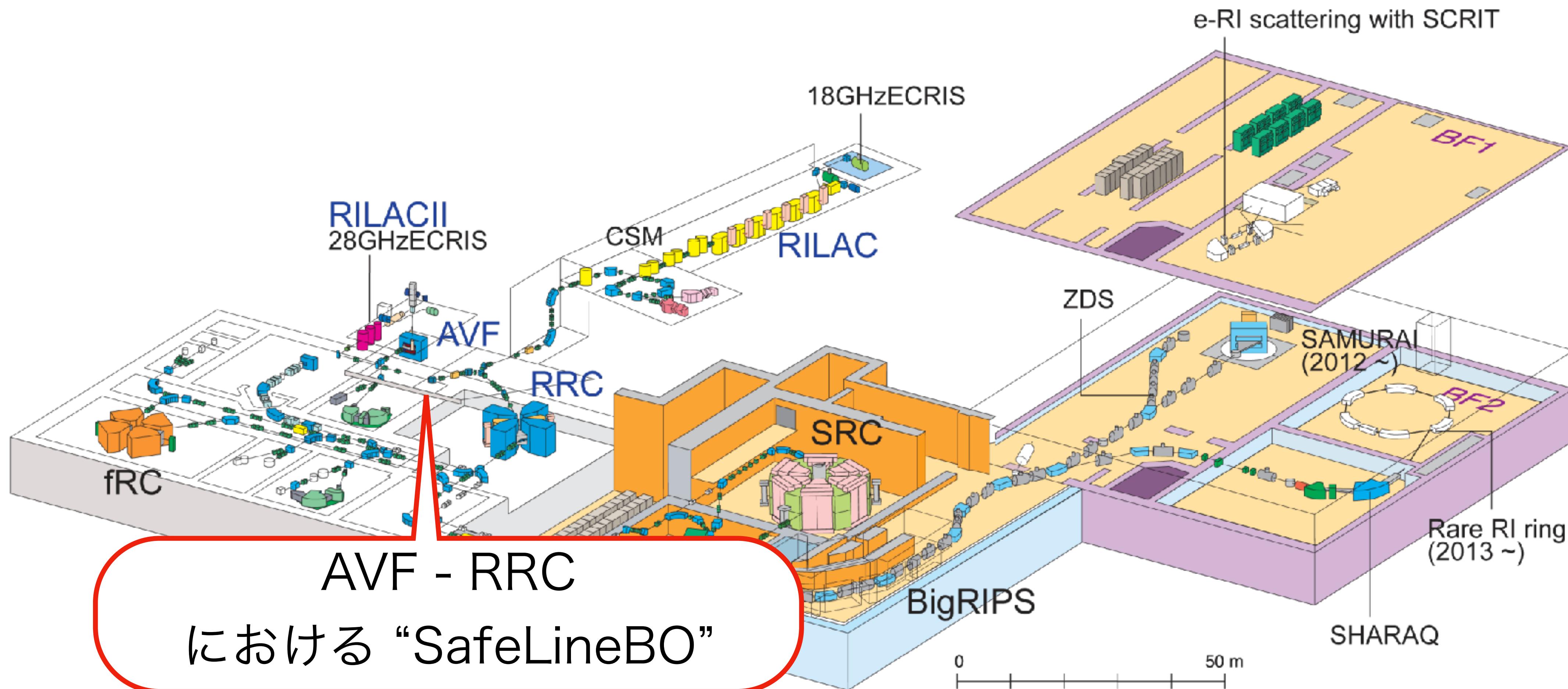
解決方法 ②: 多次元空間の中から ”賢く” 部分空間を抜き出す
→ (Safe)LineBO

Pros: 正しく動けば、多次元最適化を効率よく行うことが可能。

安全関数を用いた手法と組み合わせる

Cons: 正しく感度のあるパラメータ(組)を見つけ出すことは可能？？

自動最適化のテスト@ RILAC / RILACII ~ RRC



安全に自動調整を行うシステム: Safe optimization using LineBO

Objective

$$\max_{x \in \chi} f(x) \text{ (or } \min_{x \in \chi} f(x)) \quad \text{s.t.} \quad \begin{cases} g_1(x) \leq 0 \\ \vdots \\ g_l(x) \leq 0 \end{cases}$$

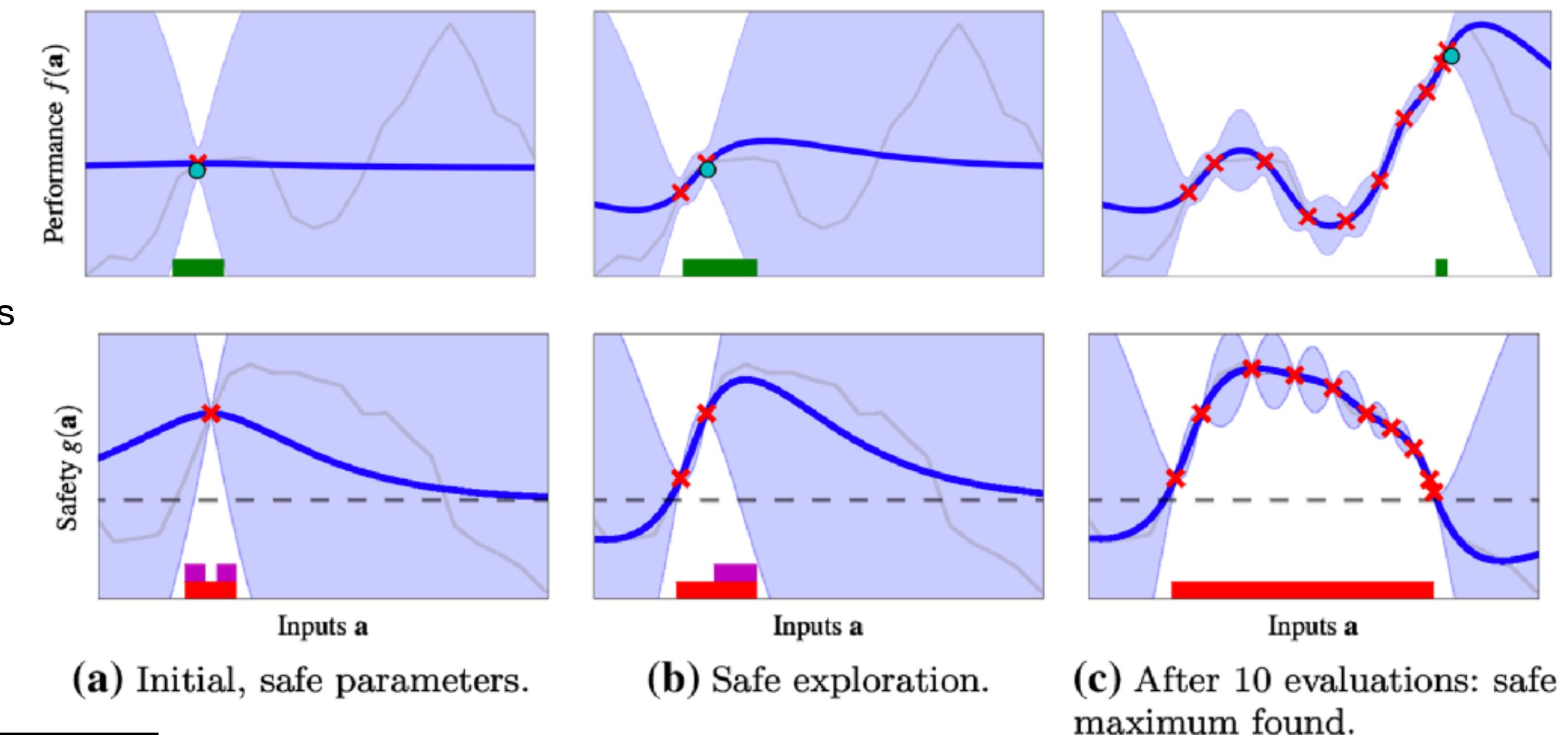
Approaches in the safe optimization community

Prepare distinct GP models for both the objective and the constraints
Evaluate within the safe set only

$$x_t = \arg \max_{x \in S_t^\tau} \text{UCB}_f(\hat{m}_t, x, \delta)$$

$$s.t. \quad S_t^\tau = \{x \in \chi : \max_{i \in [l]} \text{UCB}_{g_i}(\hat{m}_t, x, \delta) \leq -\tau\}$$

Margin



“安全関数”もモデル化

安全関数がある値以上の場所を探索

測定によって目的関数と同時に安全関数も更新

Felix Berkenkamp (2021)

安全に自動調整を行うシステム: Safe optimization using LineBO

計算コストの増大

→ Line Bayesian Optimizationで探索を効率的にする

モデル上で最も勾配の強い方向の1次元部分空間上の
ベイズ最適化問題を解く。最適解を見つけた後は、再び近
傍で勾配を計算、最も勾配の強い方向で改めて探索。

(A) CoordinateLineBO

各パラメータの軸に合わせて方向を選択

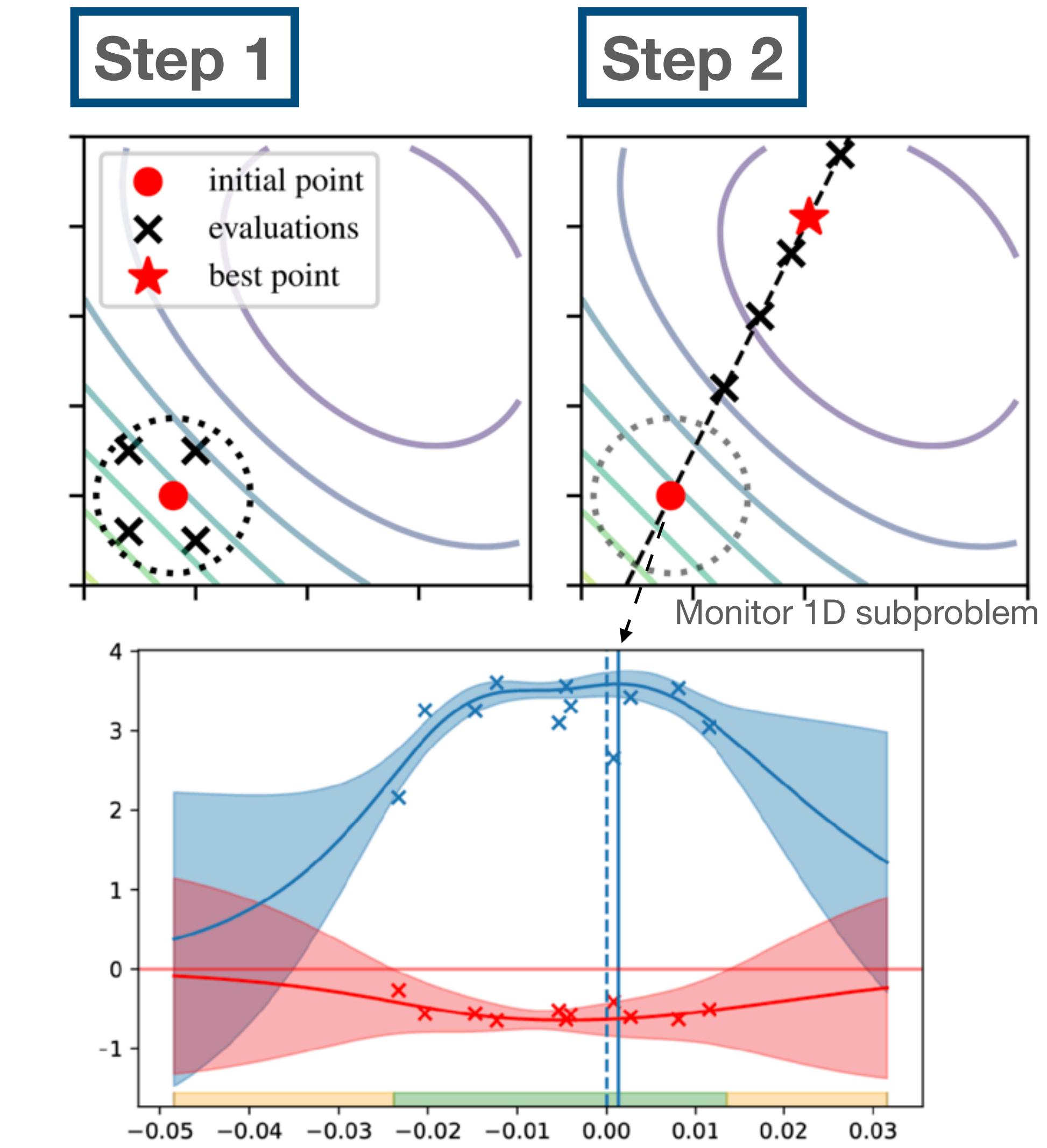
(ex. 磁石ごとに最適化) ※人間の最適化に近い

(B) Descent (Ascent) LineBO

近傍で勾配を計算、最も感度の高い方向に最適化

(C) RandomLineBO

Random に方向を選択

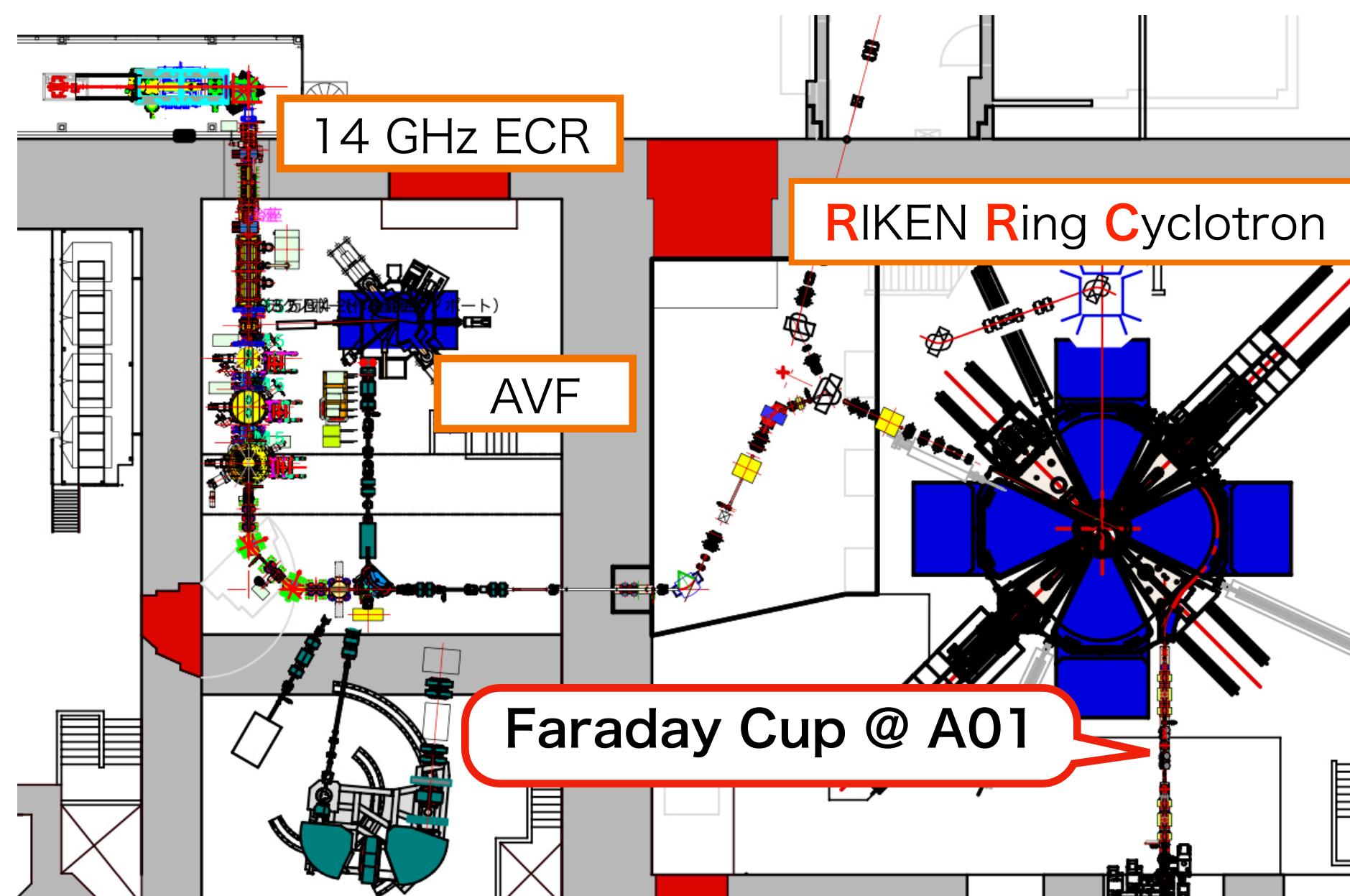


J. Kirschner, M. Mutný, N. Hiller, R. Ischebeck, and A. Krause, *36th Int. Conf. Mach. Learn. ICML 2019*, vol. 2019-June, no. Icml, pp. 5959–5971, 2019.

AVF ~ RRC における SafeLineBO テスト

イオン源から RRC 出口の FC まで。単純化のため、
今回は四重極磁石のみで 24 (23) パラメータ。

※ 今年 10 月に行った実験



事前準備:

藤井氏が整備してくれた SafeLineBO の
プログラムを岩井さんの GPopt に江川氏が実装
flag で SafeLineBO が選べるように。

※しかし numpy の ver. が古くて最新の
branch と merge できない。。。
適切なハイパーパラメータを見積もるため、
AVF から RRC 入口までのモンテカルロ
シミュレーションを作成。(gicosy + MOCADI)
このシミュレーションを基に、江川氏が
Optuna を用いてハイパーパラメータを最適化
(ベイズ最適化のためのベイズ最適化)

候補 1: $\beta \sim 6.7$

候補 2: $\beta \sim 0.3$

AVF ~ RRC における SafeLineBO テスト

イオン源から RRC 出口の FC まで。単純化のため、
今回は四重極磁石のみで 24 (23) パラメータ。

Safe Line BO の Procedure

信頼探索

現在の“最適値” A の周りで、安全な範囲内で探索を行って進むべき方向を探す。

ベストの値を試す

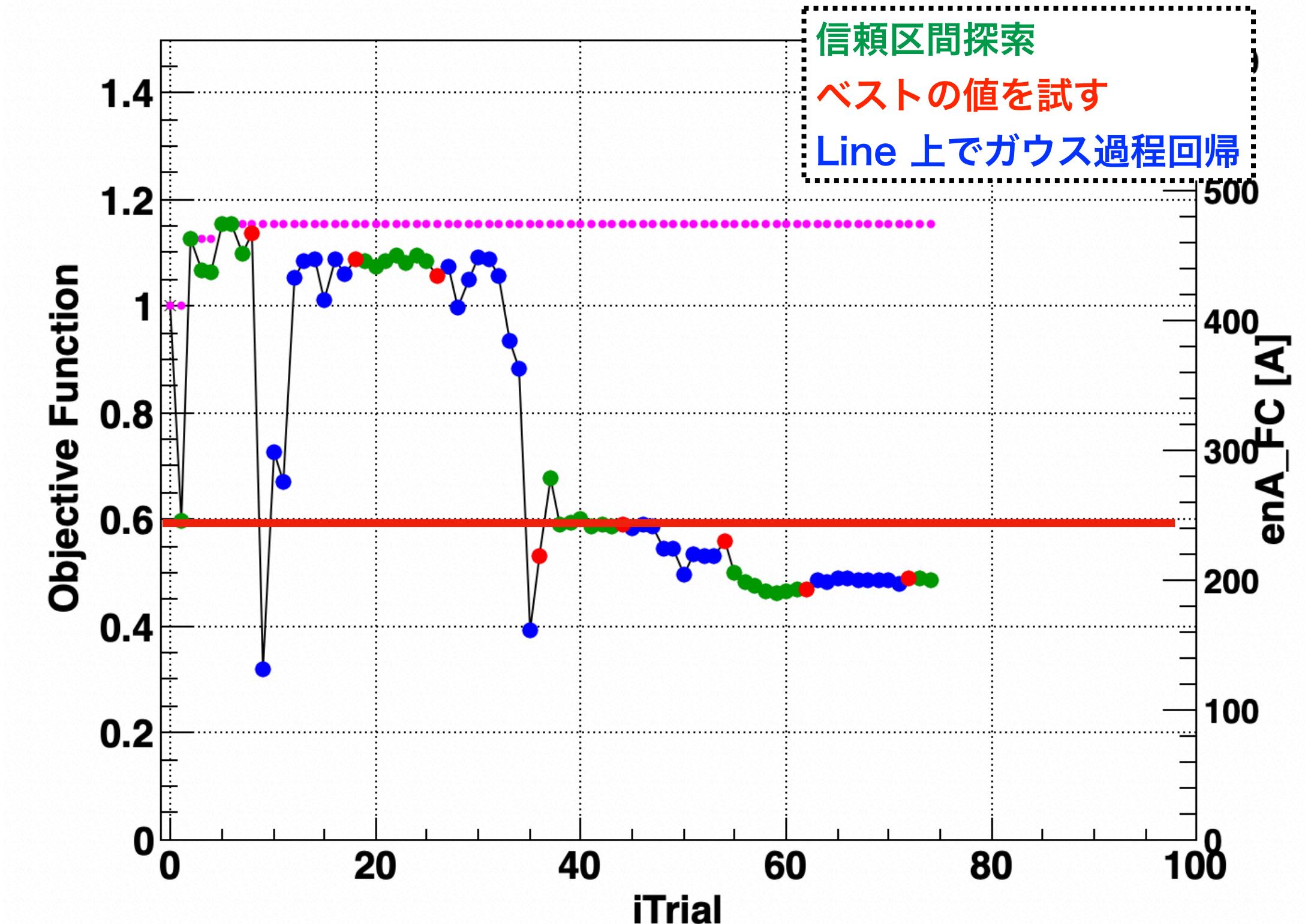
ここまでで見つけた最も良い候補点 B を試す
(これまでのデータで作ったモデルの最尤値)

Line 上でガウス過程回帰

A と B を結んだ方向でガウス過程回帰

このとき、安全閾数も計算し、安全な範囲内で探索

最初の SafeLineBO のテスト (23 パラメータ)
 $\beta = 6.7$



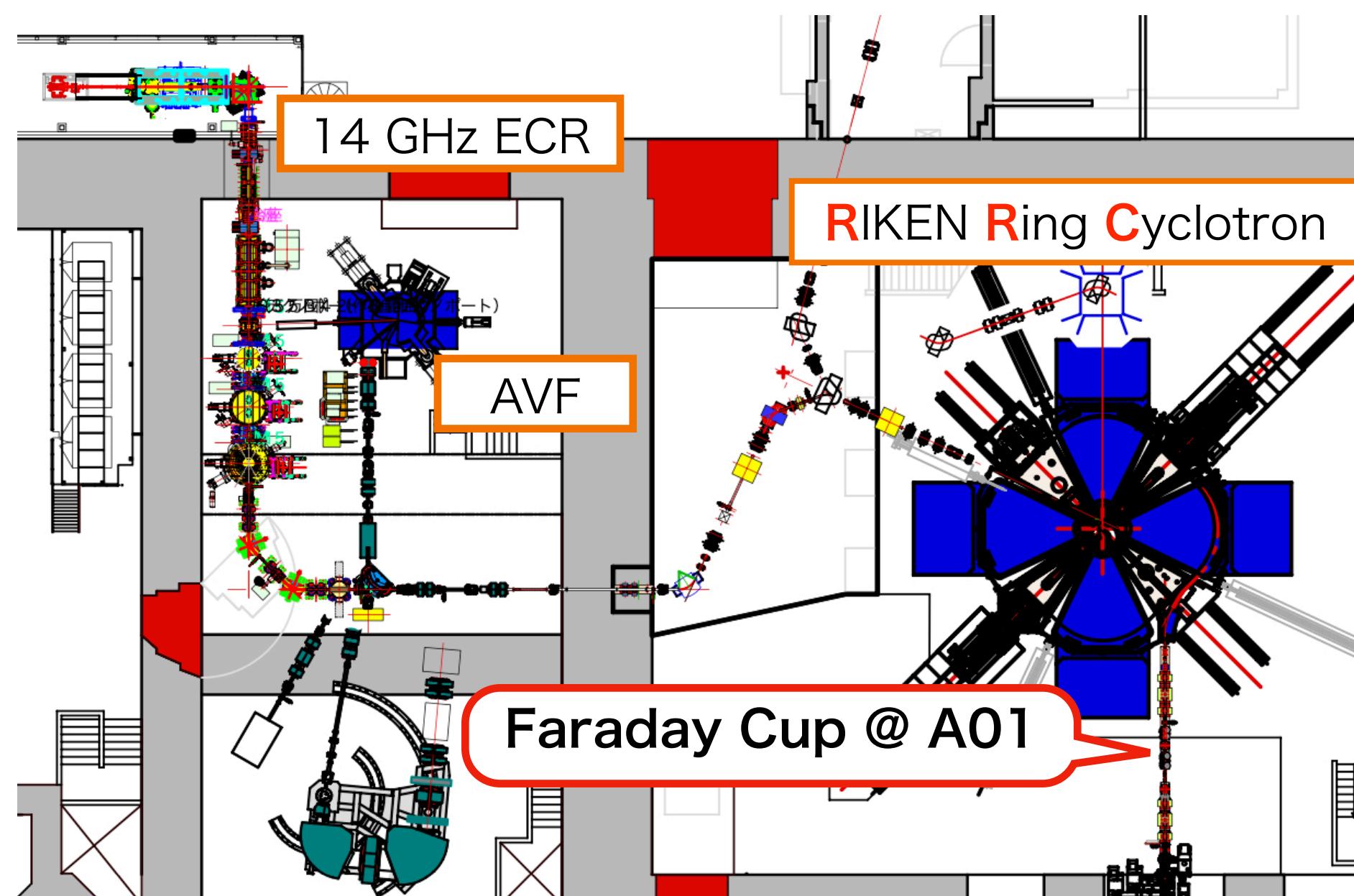
色々おかしい。。。。

- Line 上探索で、前の best に戻らない。
- 安全閾値を下回っている。

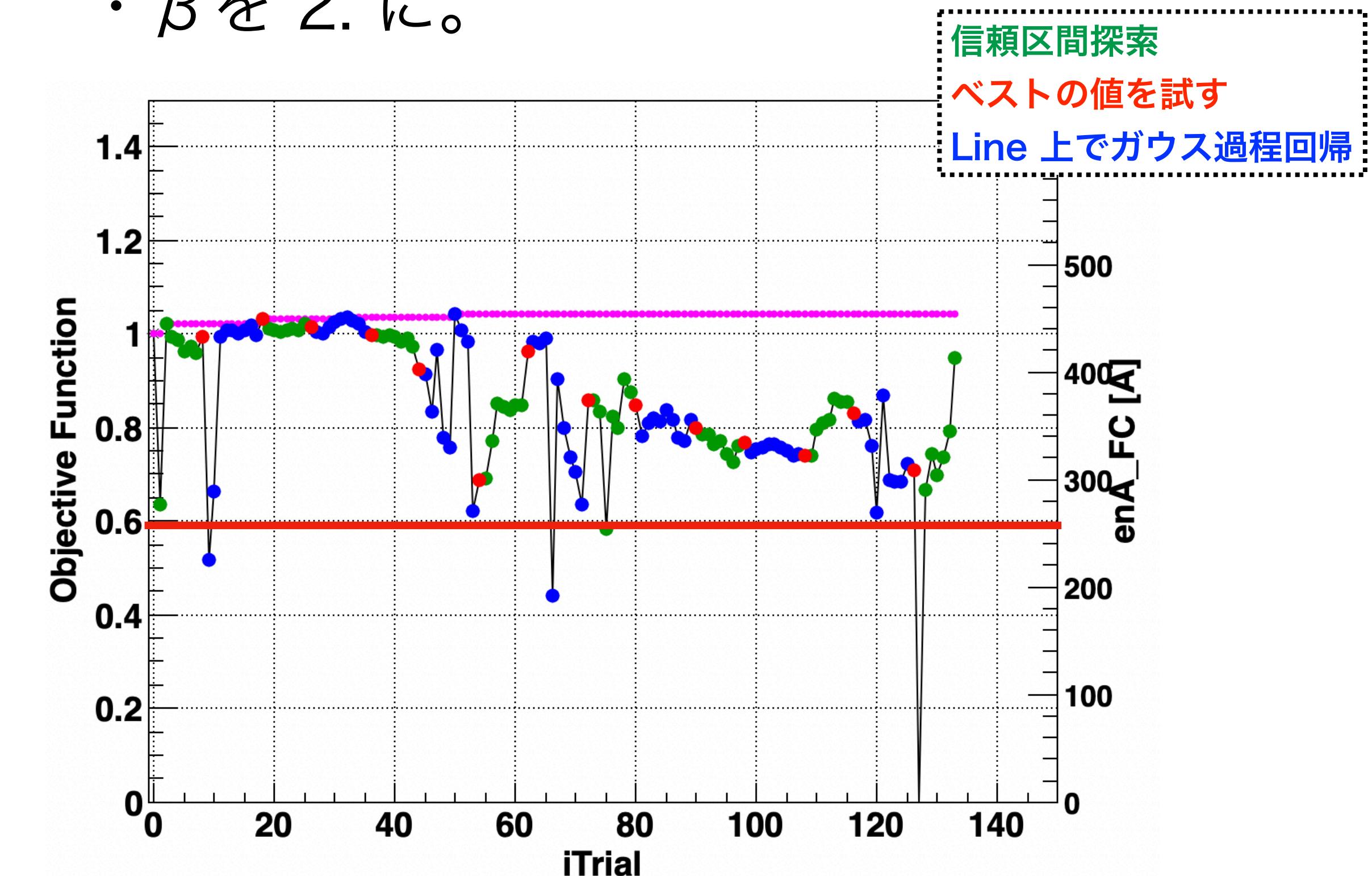
AVF ~ RRC における SafeLineBO テスト

イオン源から RRC 出口の FC まで。単純化のため、
今回は四重極磁石のみで 24 (23) パラメータ。

※ 今年 10 月に行った実験



その後色々とパラメータを調整
 • 信頼区間を狭くする。
 • FCの測定の間隔を空ける
 • β を 2. に。

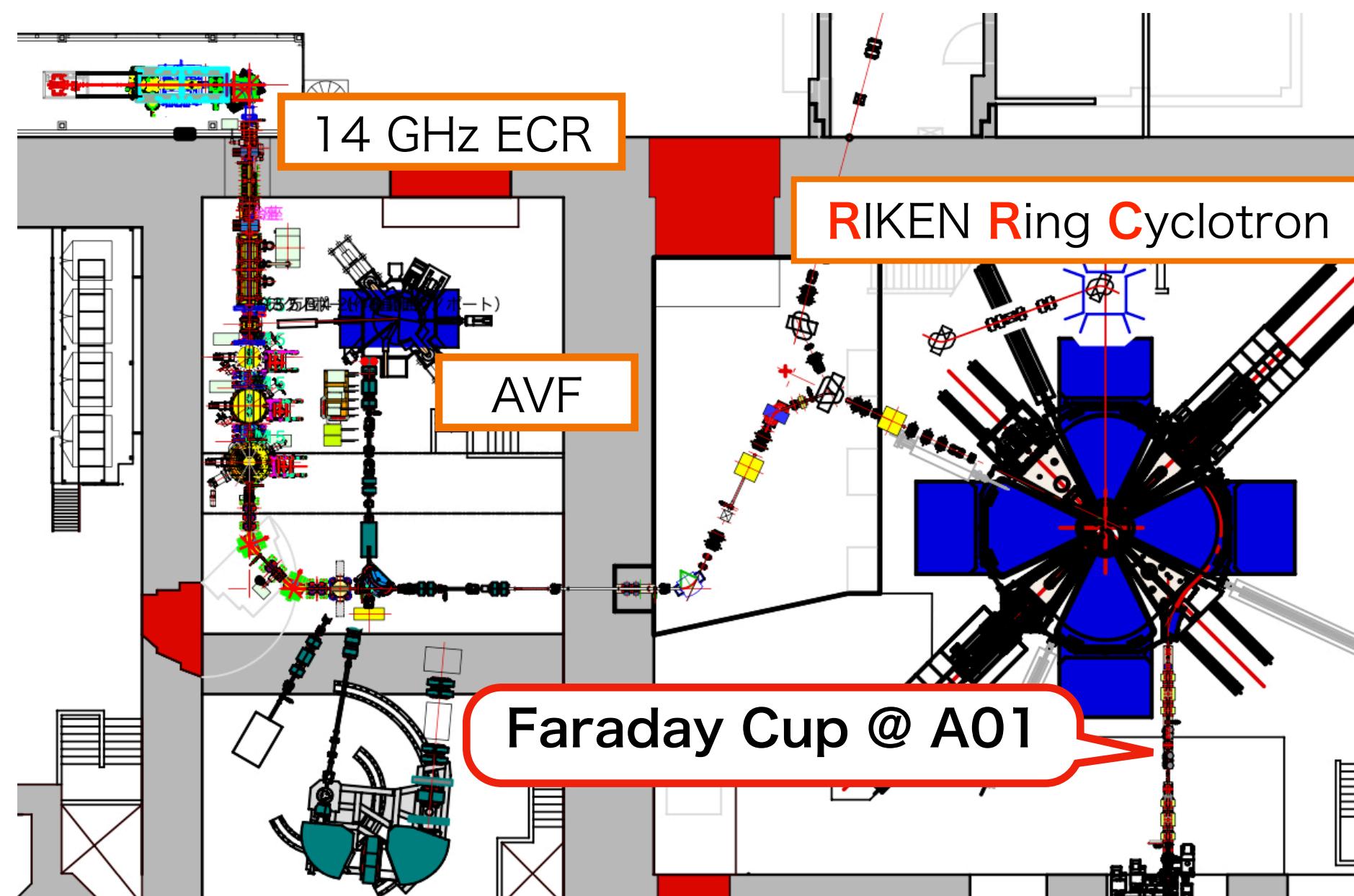


一応少しあは良い解を見つけたrun。ただしやはりおかしい。

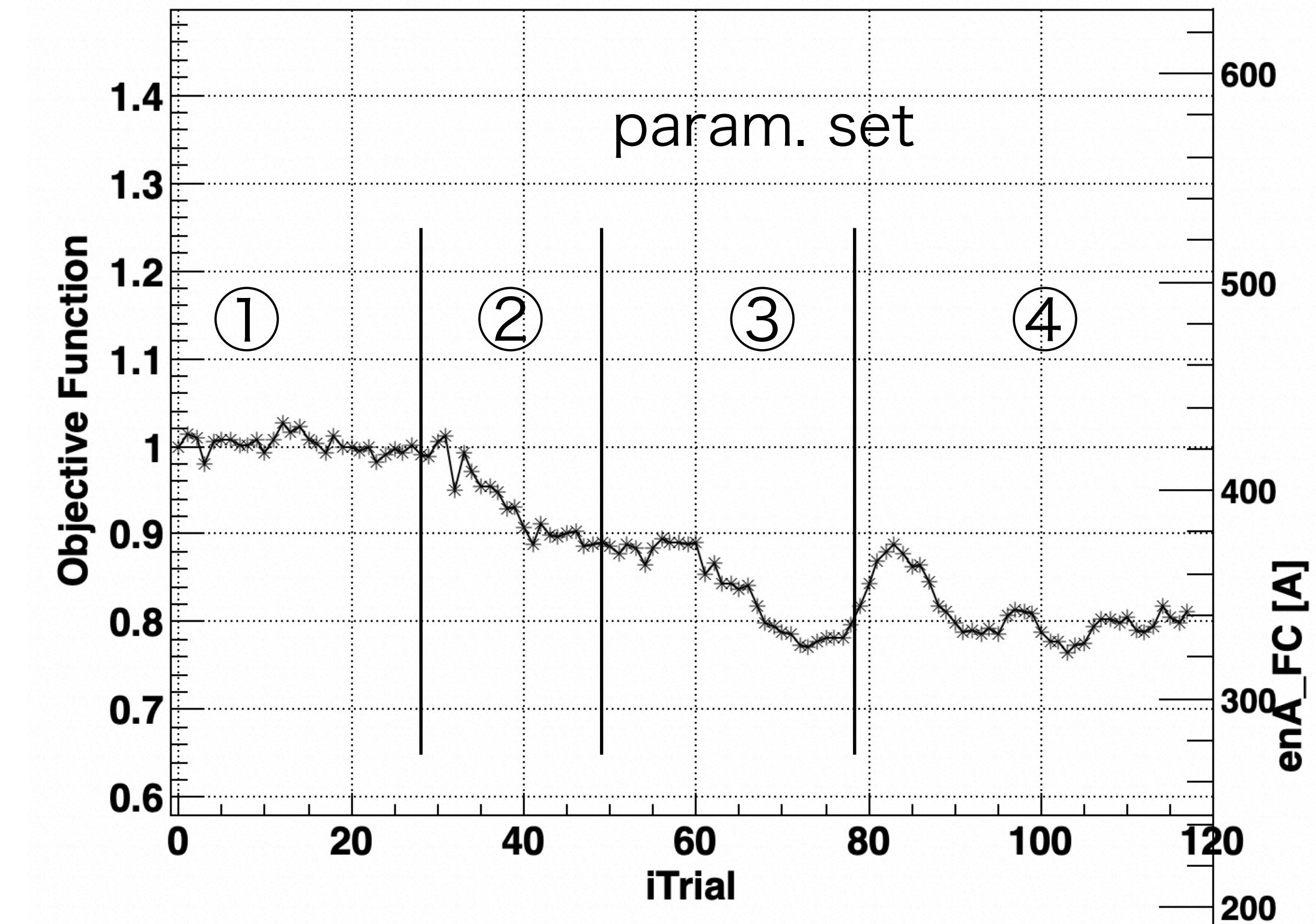
AVF ~ RRC における SafeLineBO テスト

イオン源から RRC 出口の FC まで。単純化のため、
今回は四重極磁石のみで 24 (23) パラメータ。

※ 今年 10 月に行った実験



本当に電流量が増加していくかを、
手動でパラメータを変えて検証

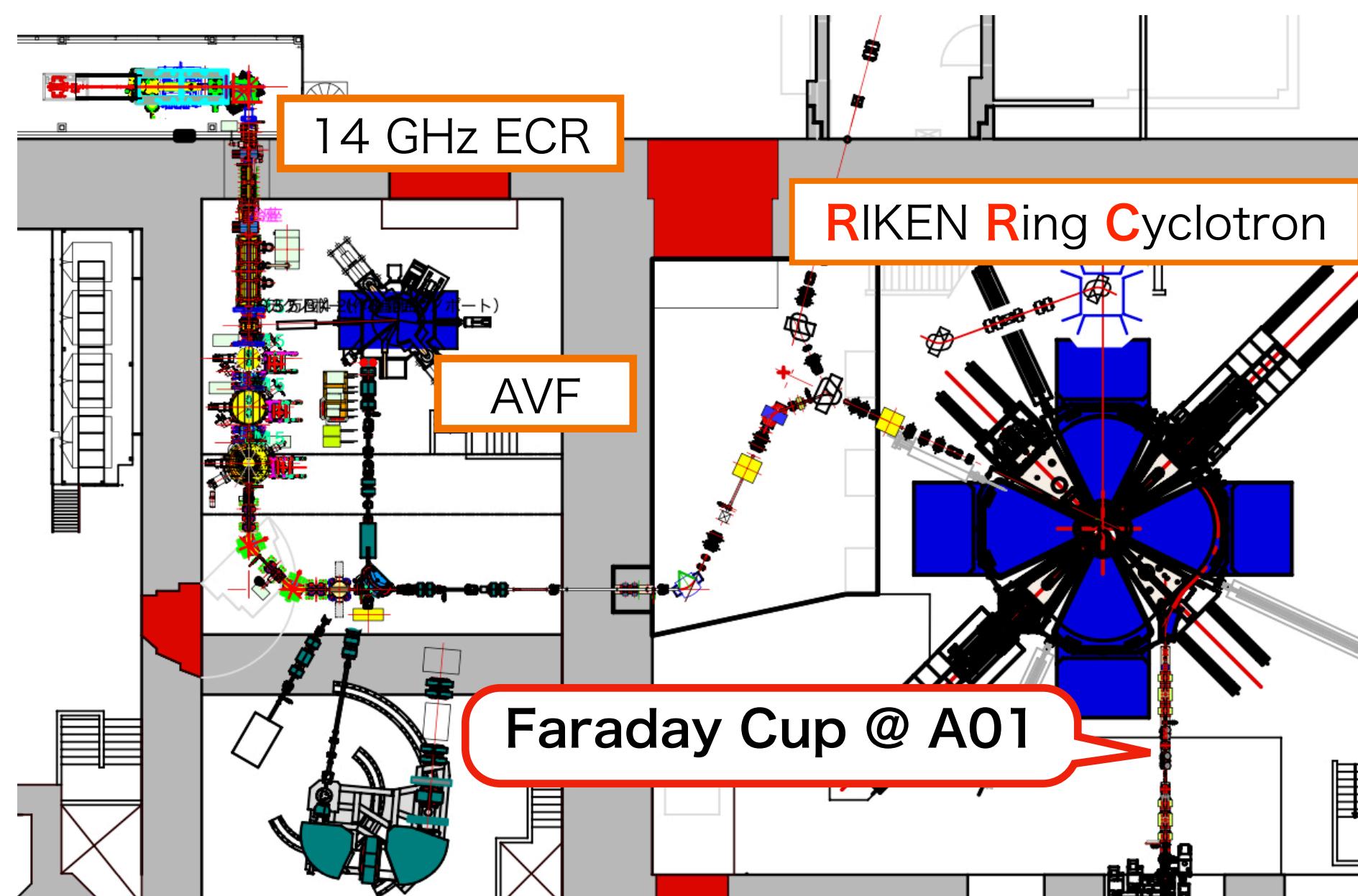


- ①: default
- ②: ML 初期値 (default を少し悪化)
- ③: ML best ver. 1
- ④: ML best ver. 2

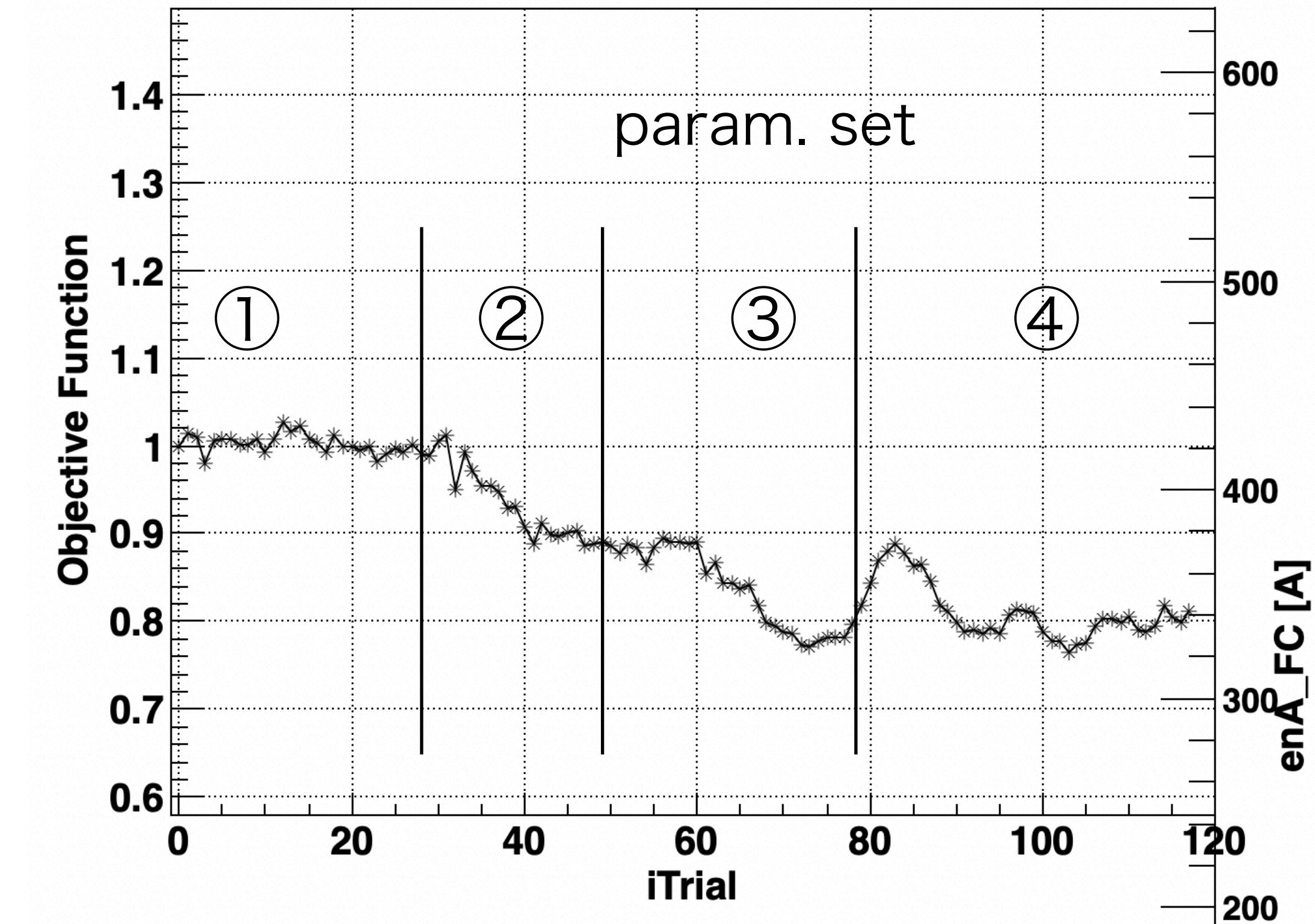
AVF ~ RRC における SafeLineBO テスト

イオン源から RRC 出口の FC まで。単純化のため、
今回は四重極磁石のみで 24 (23) パラメータ。

※ 今年 10 月に行った実験



本当に電流量が増加していくかを、
手動でパラメータを変えて検証

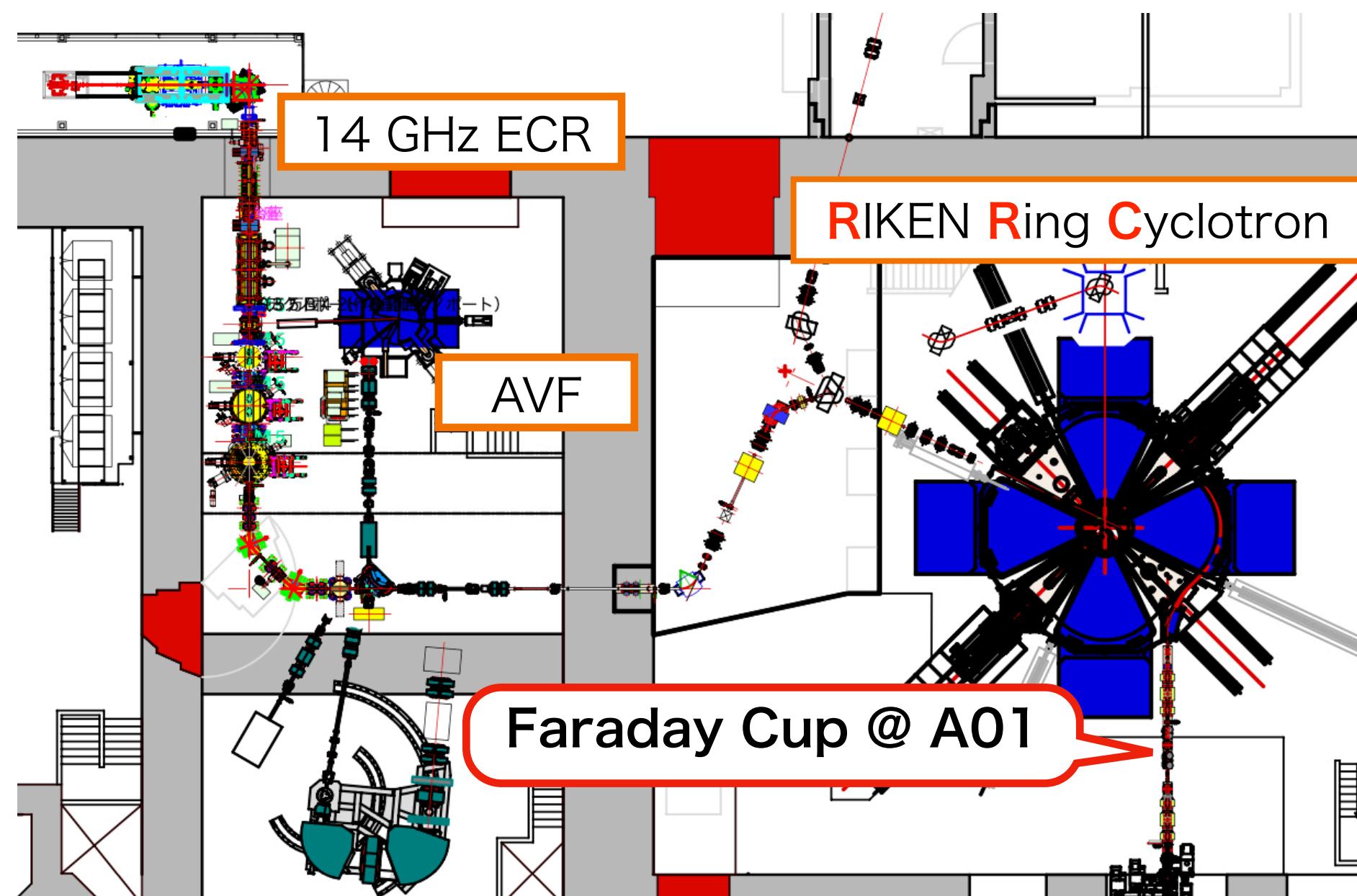


パラメータを変えた後もだらだら変わる！！
→ イオン源が不安定？

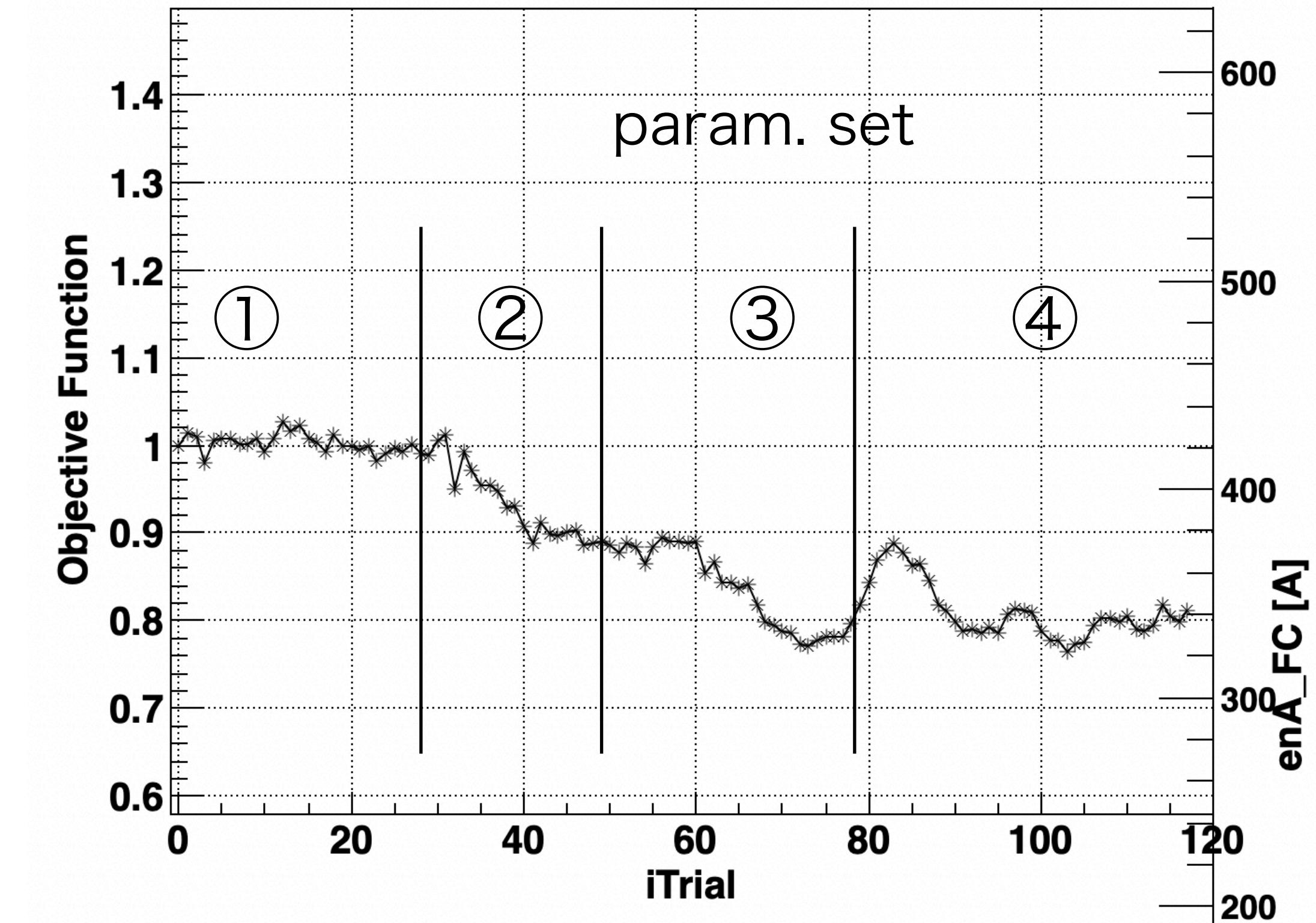
AVF ~ RRC における SafeLineBO テスト

イオン源から RRC 出口の FC まで。単純化のため、
今回は四重極磁石のみで 24 (23) パラメータ。

※ 今年 10 月に行った実験



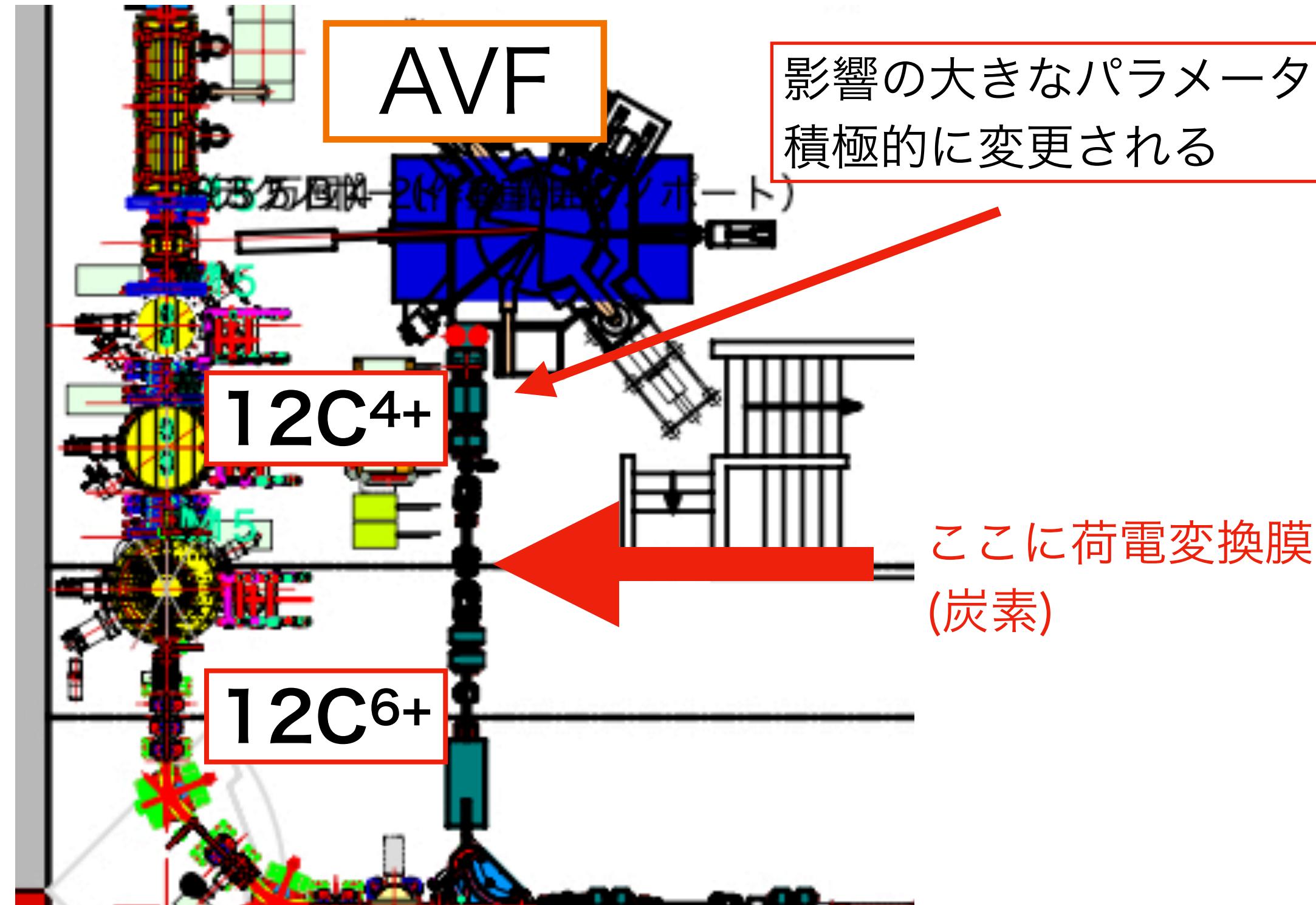
本当に電流量が増加していくかを、
手動でパラメータを変えて検証



パラメータを変えた後もだらだら変わる！！
→ ~~イオン源が不安定？~~ システムの応答遅延が原因

AVF ~ RRC における SafeLineBO テスト

イオン源から RRC 出口の FC まで。単純化のため、
今回は四重極磁石のみで 24 (23) パラメータ。



荷電変換膜である炭素の上流の TQ を触ることで、膜上のスポットが変わっていた。
→ 熱負荷でゆっくりとした応答をしていた可能性が高い
(運転員は基本的にここを触らない。)

そのため、目的関数 / 安全関数のモデル化がうまく行かず、
挙動がおかしかったのではないか。

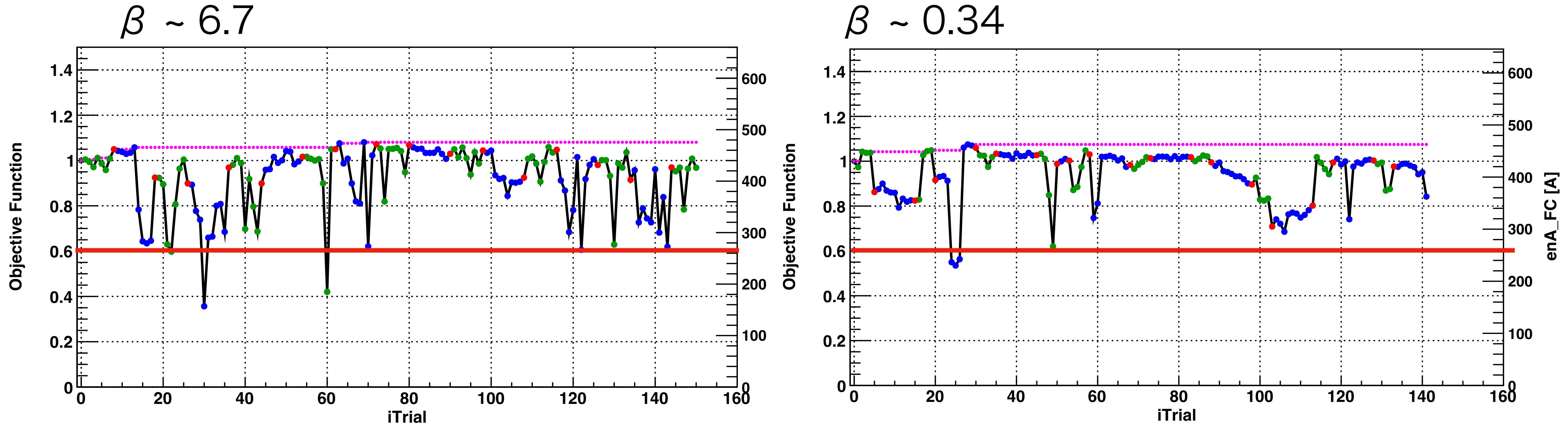
教訓

最適化の対象については事前に良く検討し、
普段の調整などを熟知してから行いましょう。

AVF ~ RRC における SafeLineBO テスト

テストの前提条件が一部崩れていた。とはいえ、いくつか知見は得られたので考えてみる。

(1) SafeLineBO の効果と β 依存性



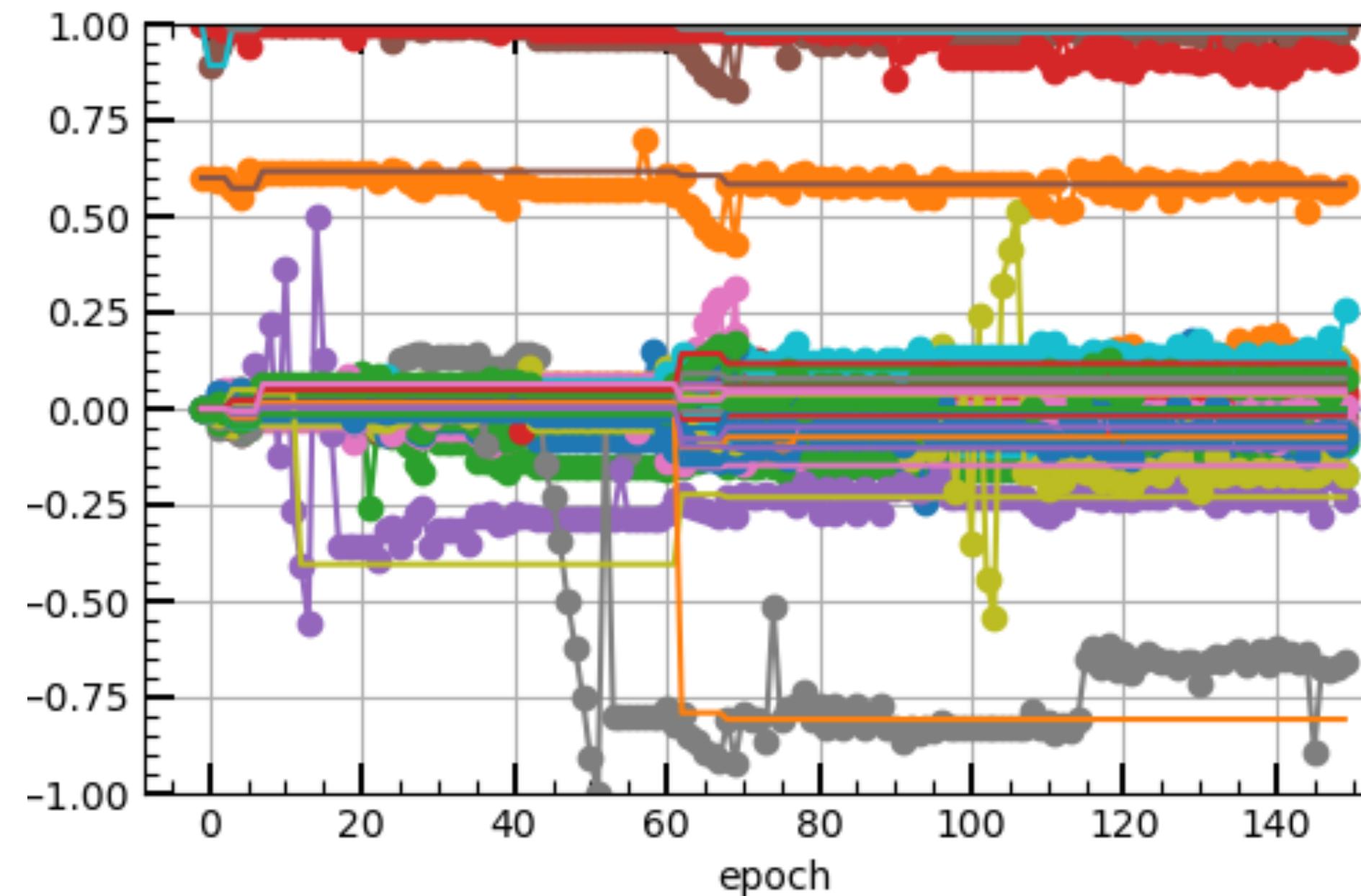
安全関数のモデル化が不完全だったものの、ある程度ハイパーパラメータを調整後は安全関数の閾値を"ある程度" 維持している。
 β に対する変化も期待通り。

AVF ~ RRC における SafeLineBO テスト

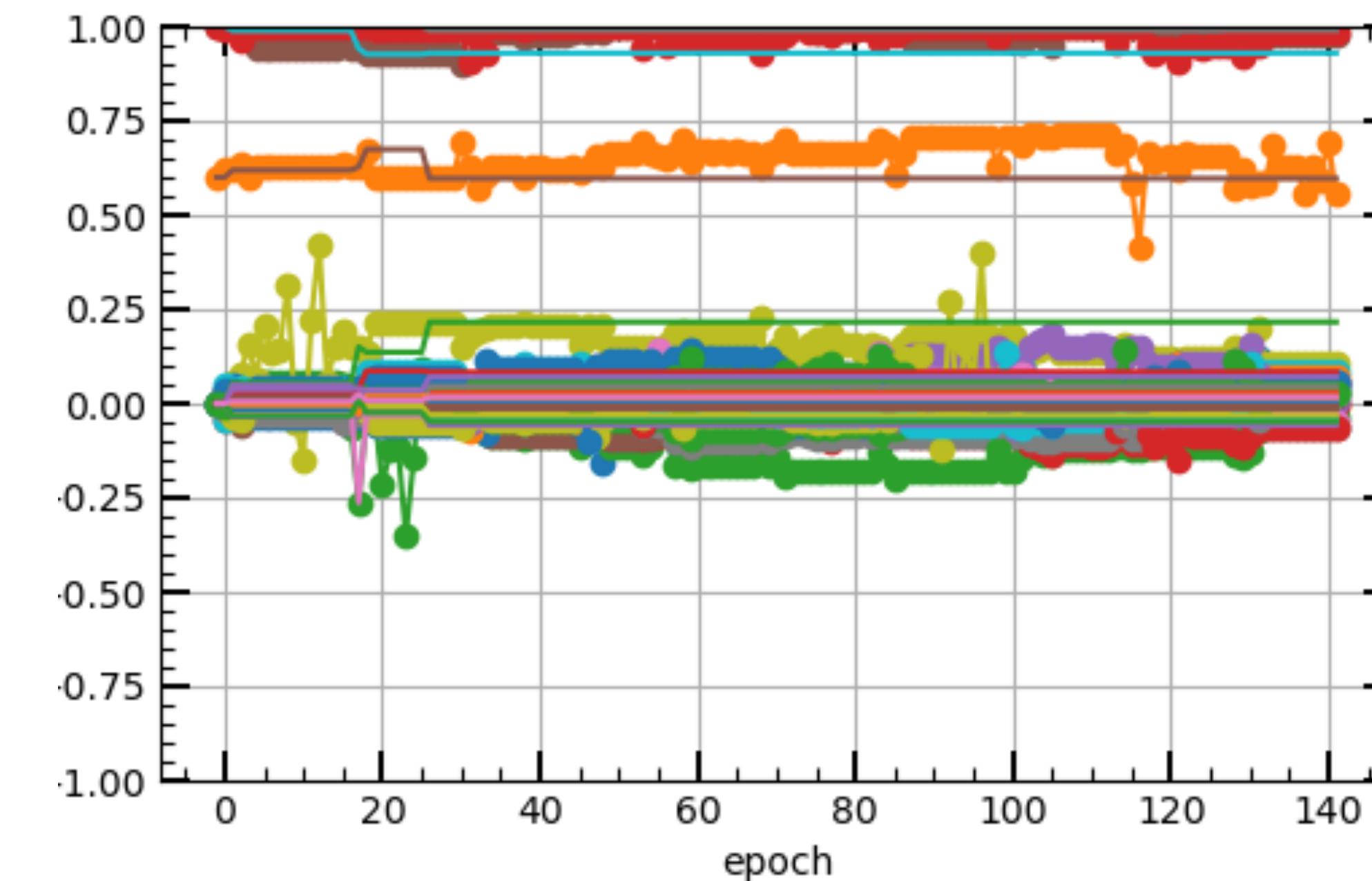
テストの前提条件が一部崩れていた。とはいえ、いくつか知見は得られたので考えてみる。

(1) SafeLineBO の効果と β 依存性

$\beta \sim 6.7$



$\beta \sim 0.34$

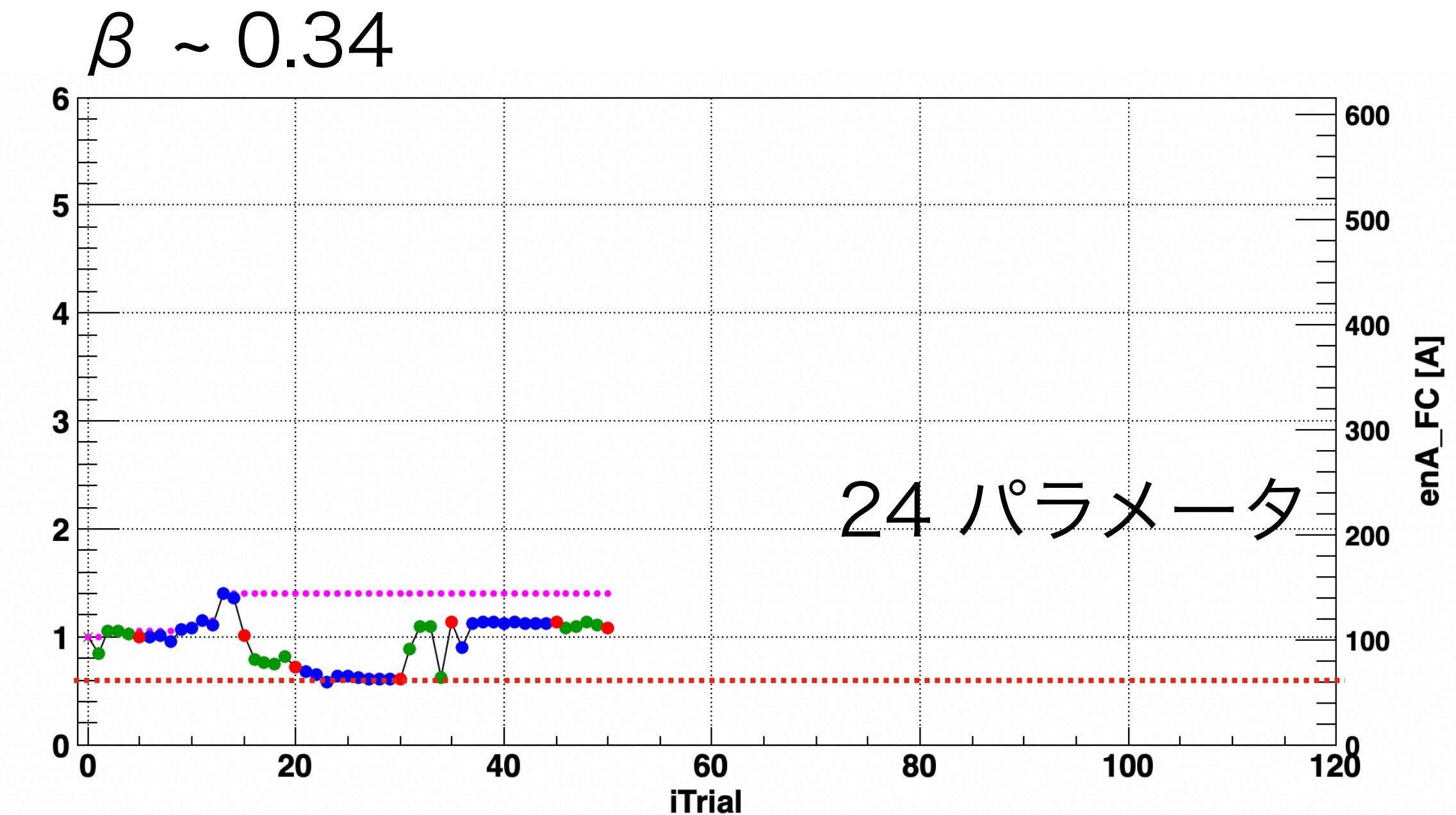
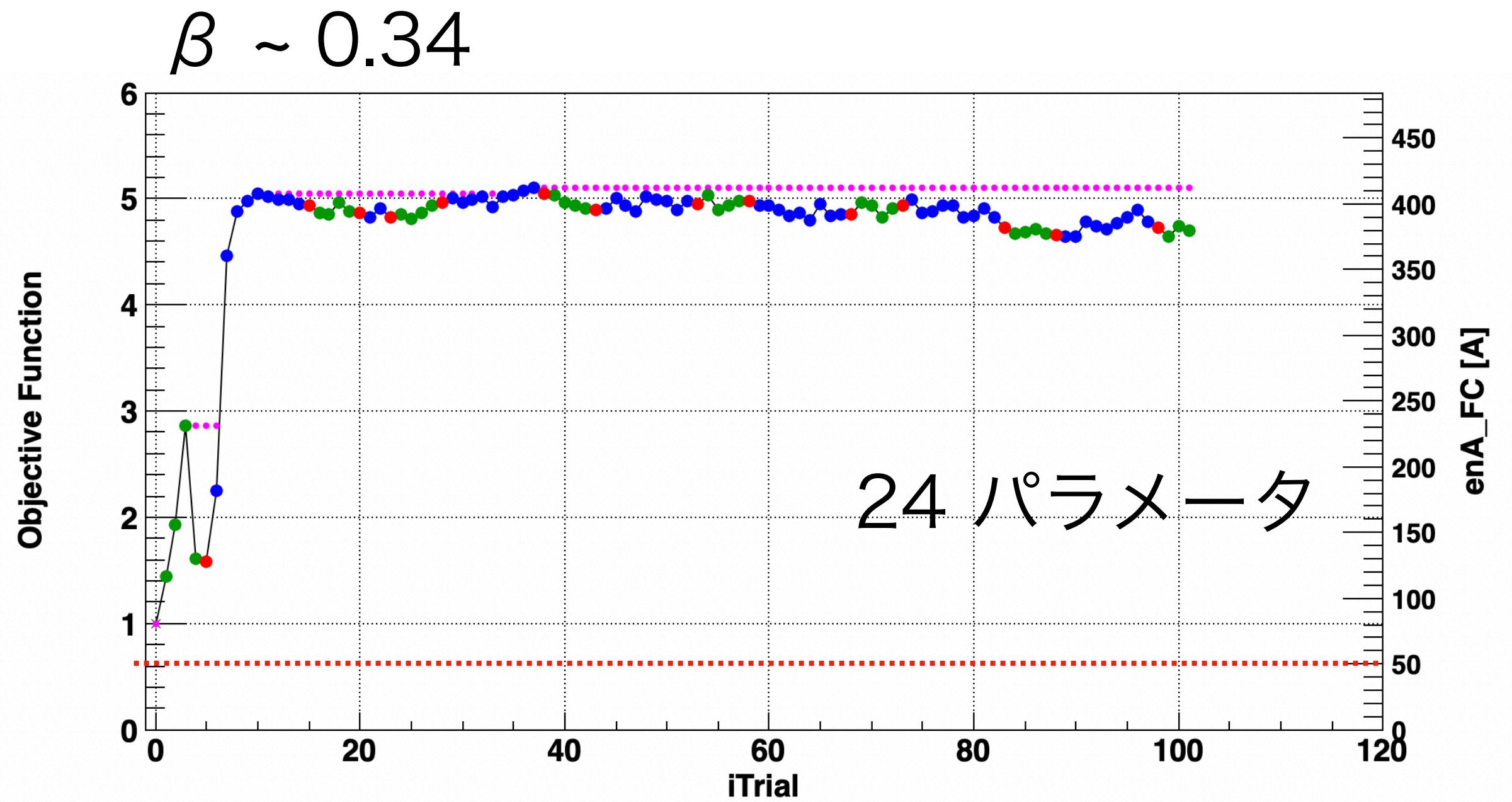


パラメータの変化についても β の値に対応している。
それなりに多くのパラメータを同時に動かしている。

AVF ~ RRC における SafeLineBO テスト

テストの前提条件が一部崩れていた。とはいえ、いくつか知見は得られたので考えてみる。

(2) 初期値を低い値から始めた時の SafeLineBO の最適化の不安定性



全く同じ条件でも、最適化が進む場合と進まない場合の落差が激しい。

→ 信頼探索で、全てのパラメータを触るわけではない。

運要素が大きい？

今後の SafeLineBO の改善点について

1. 信頼区間探索回数について

特に信頼区間探索の点数が少ないと運要素が大きくなる？

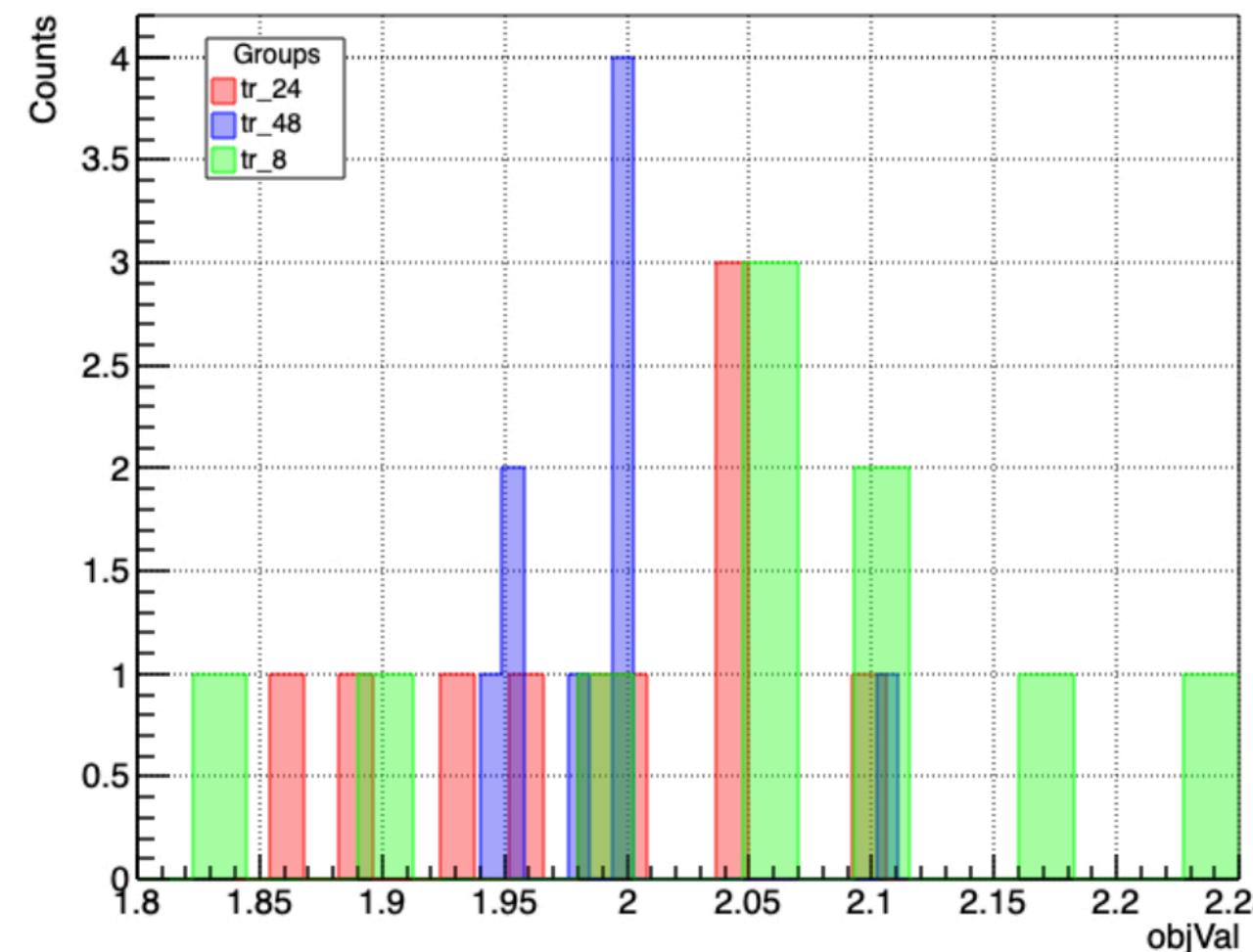
だが、信頼区間探索を増やすと同じ試行回数では LineBO の回数が減るジレンマ。

→ 江川氏によるシミュレーションでの検証。

信頼区間探索を 8, 24, 48 で試行。

やはり信頼区間探索を増やしたほうが結果は安定。

objVal Distribution by Group

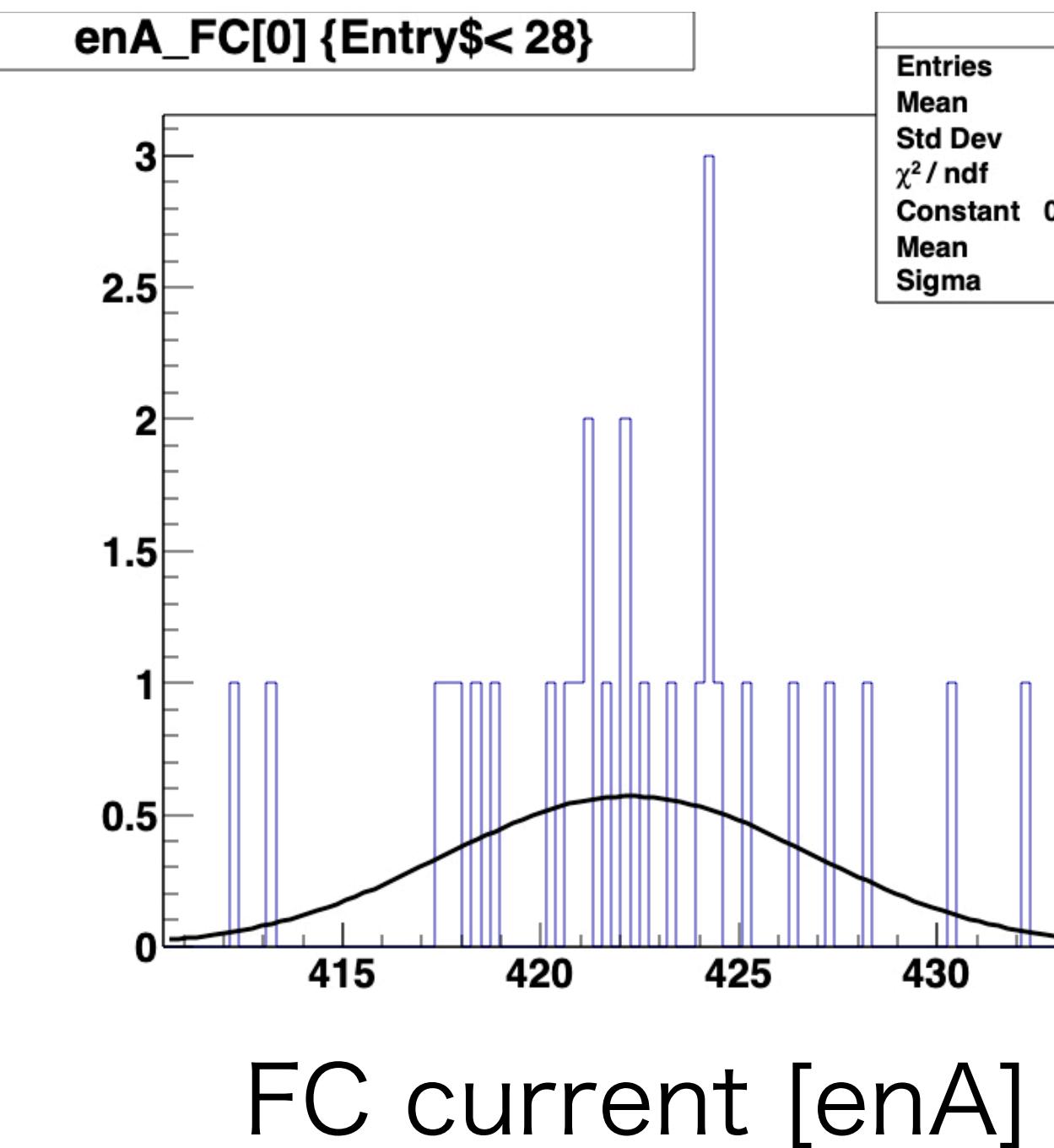


解決案

- (1) 信頼区間探索を $\text{parameter} * 2$ 程度にする
- (2) 一度 Coordinate Safe Line BO を行って一通り最適化。
→ そのデータを基に信頼区間探索少なめで
Ascent Safe Line BO に移る。
※ 信頼区間探索では過去の全データを基にモデル構築

今後の SafeLineBO の改善点について

2. So far best を試すと、過去のデータより結果が低くなりがち
 例えばデータで 412 enA の結果を見つけていても、最後そのパラメータを適応すると
 400 enA 程度にしかならないことがある。
 → 統計的に上振れしたデータが So far best になりやすいため？？



今回、FC は 0.1 秒ごとの測定 × 5 の平均値を取得
 その結果、統計的なゆらぎは ~ 5 enA 程度。
 → 2σ で上振れしたデータを取っていれば辻褷が合う。

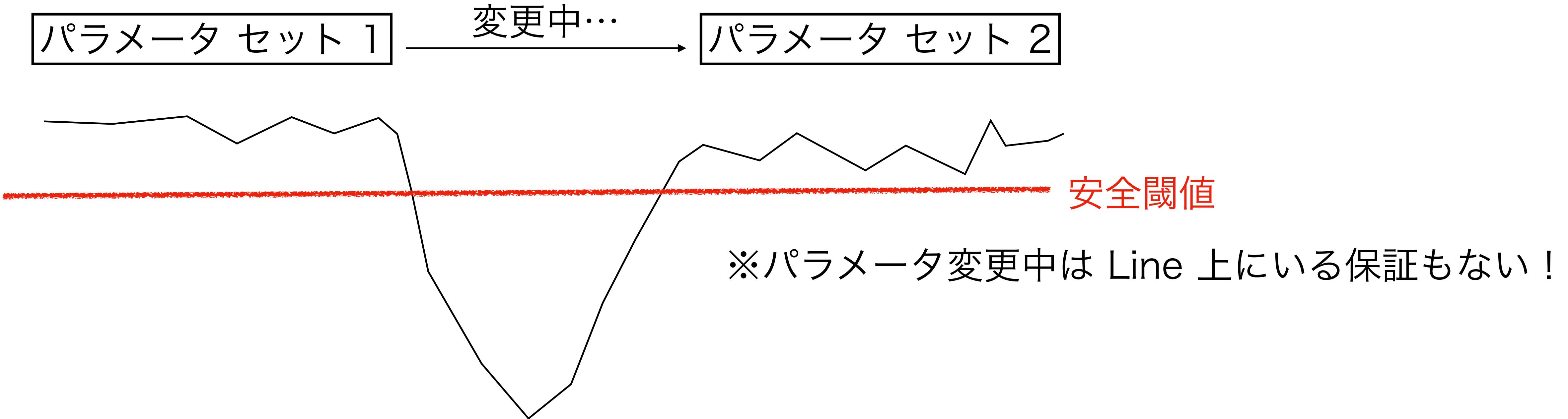
解決案

- 10 enA 程度 ($2 \sim 3\sigma$) は許容する。
- 測定データの精度を上げる？
 測定時間を長くする / intensity を上げる。

今後の SafeLineBO の改善点について

3. 安全システムについて (未来の話)

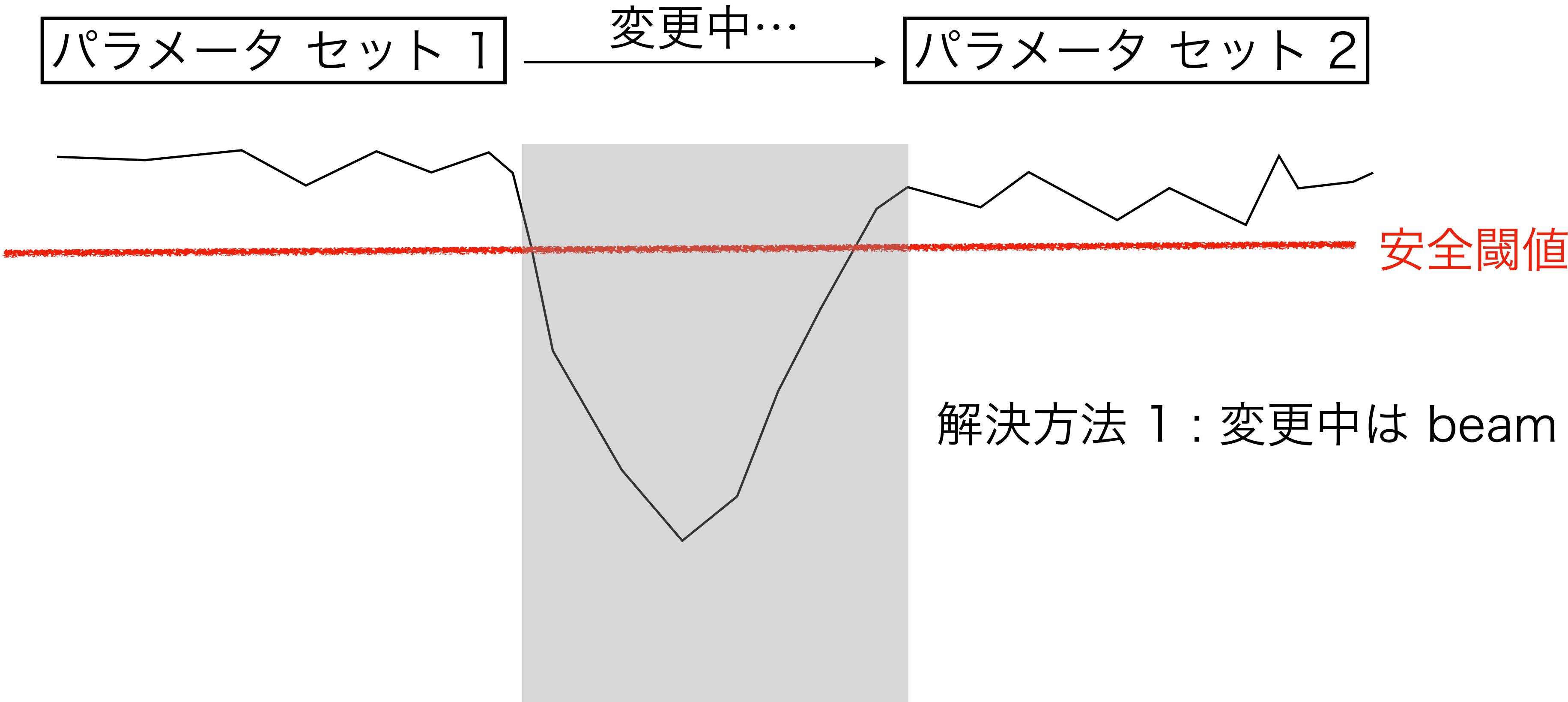
パラメータの変化中にビームロスが増加する事があるのでは？



今後の SafeLineBO の改善点について

3. 安全システムについて (未来の話)

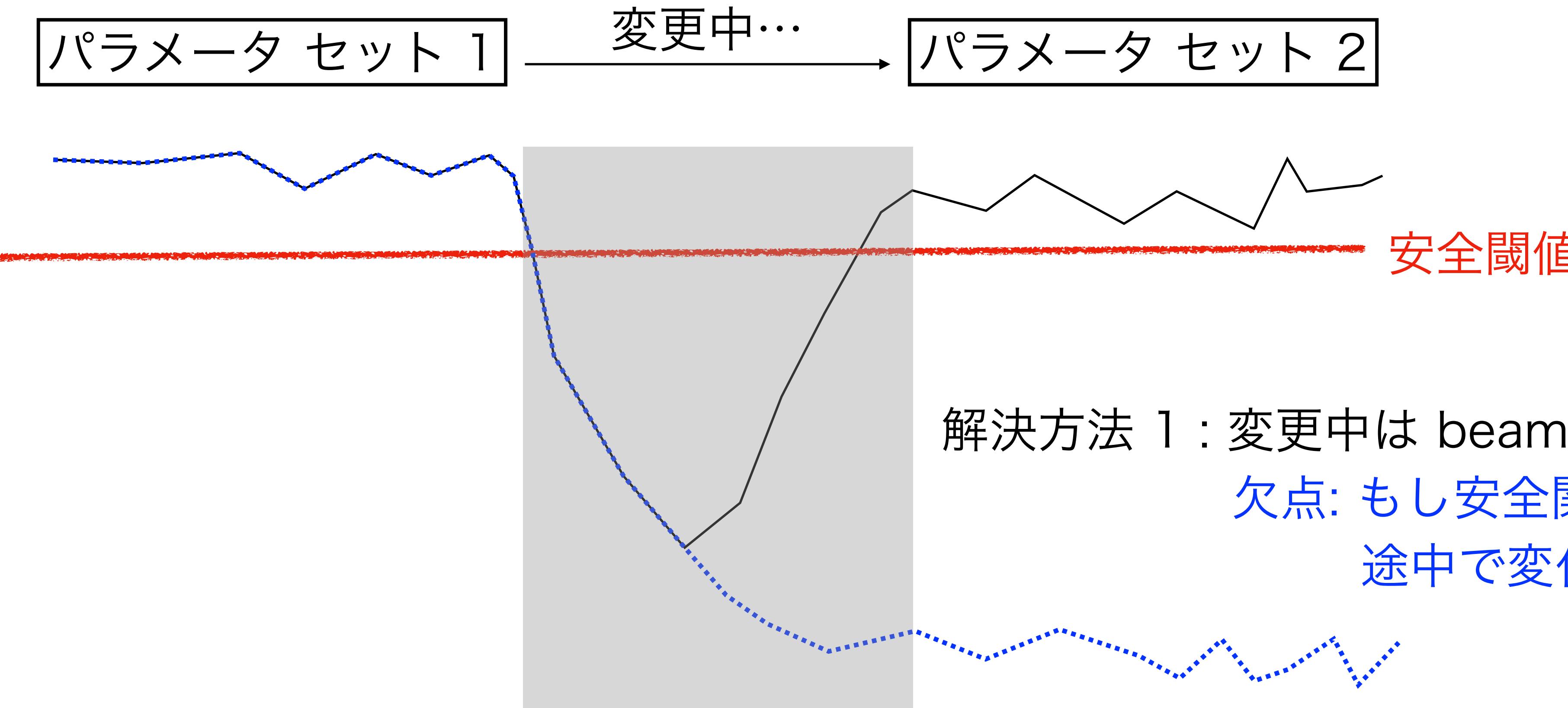
パラメータの変化中にビームロスが増加する事があるのでは？



今後の SafeLineBO の改善点について

3. 安全システムについて (未来の話)

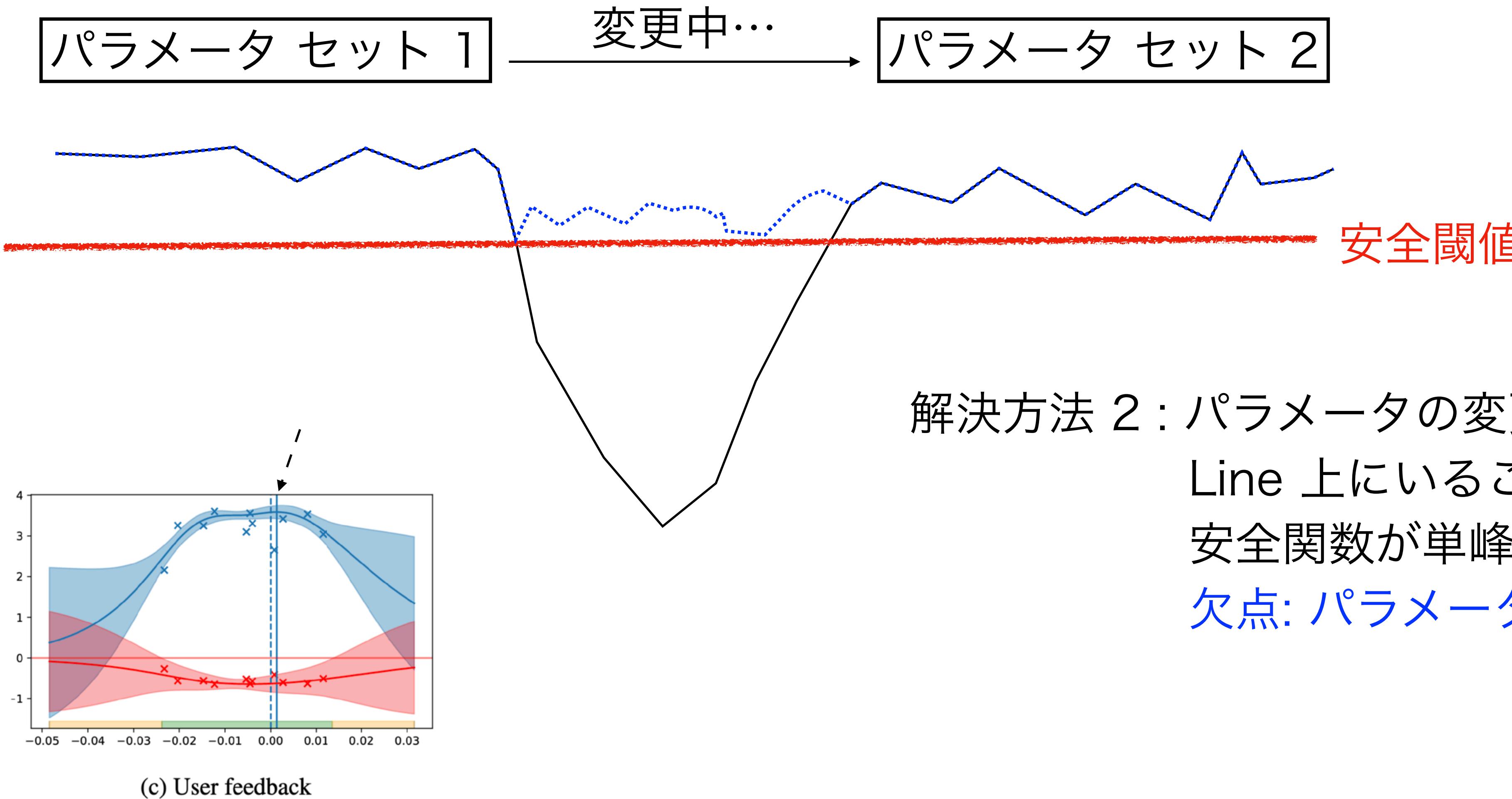
パラメータの変化中にビームロスが増加する事があるのでは？



今後の SafeLineBO の改善点について

3. 安全システムについて (未来の話)

パラメータの変化中にビームロスが増加することがあるのでは？



まとめ

- ・理研RIBFにおいては、SPring-8 SACLAにおいて開発されたGPR最適化プログラムを元に研究を進めている。GPR最適化プログラムは誤差付きの測定値を元にした多次元パラメータの最適化に強みをもつ。
- ・多数のパラメータを取り扱う方法として、グループ分けで局所最適化を試す。全70パラメータを複数のパートに分けて自動最適化通過率の10%向上 / 40分に成功。ただし安全性に難あり。
- ・SafeLineBOは安全性と多数のパラメータを取り扱う最適化として非常に有力。
- ・実機で試したものとの、適応するシステムへの理解が不十分でよいテストにはならなかつたが、課題はいくつか見つけることが出来た。
→今後も現場でのテストを繰り返し、実用性をあげていく予定。
- ・physics informed …??

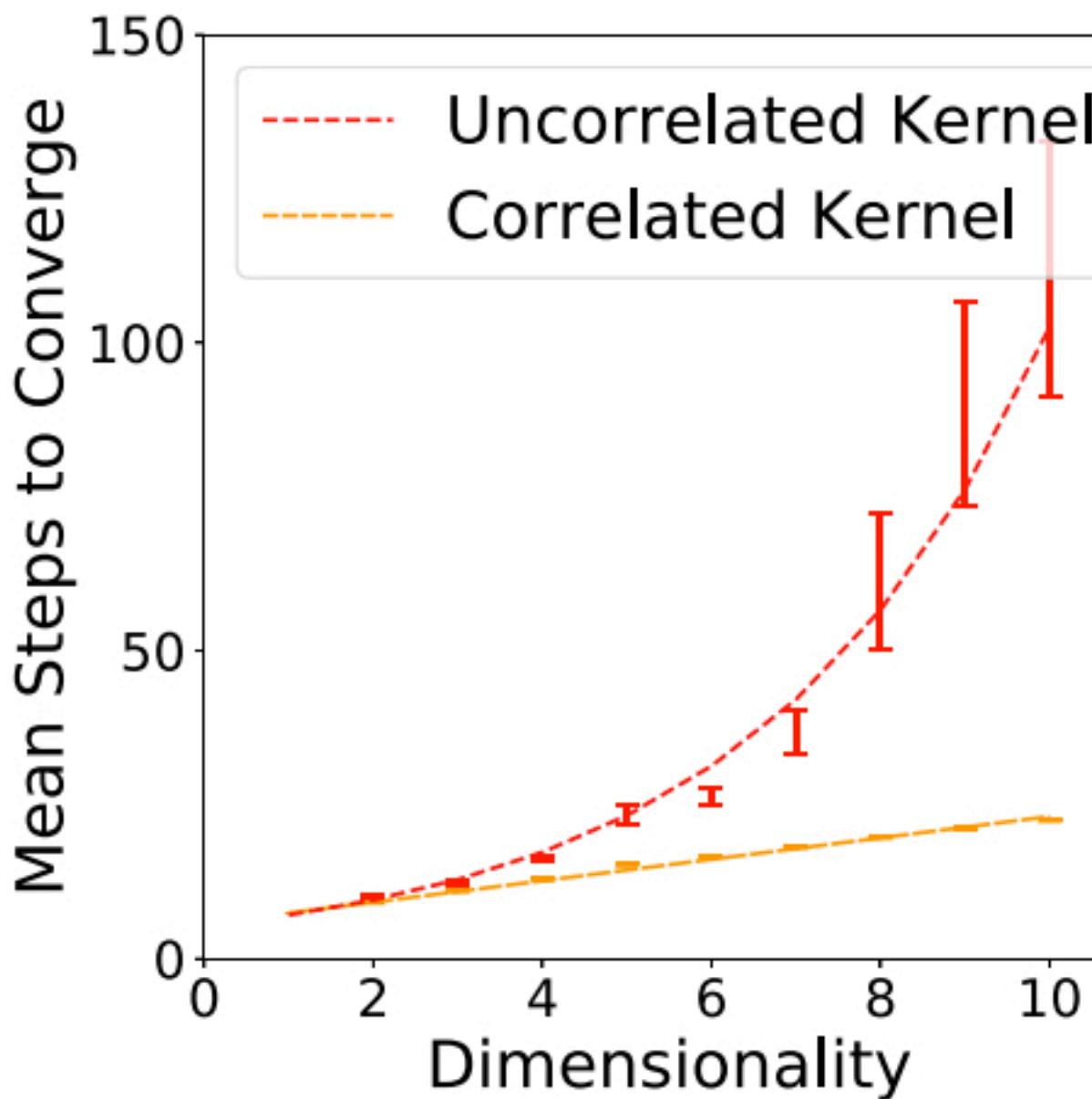
Back Up

その他課題を解決するためのいくつかの手法

b. Physics informed Gaussian Process

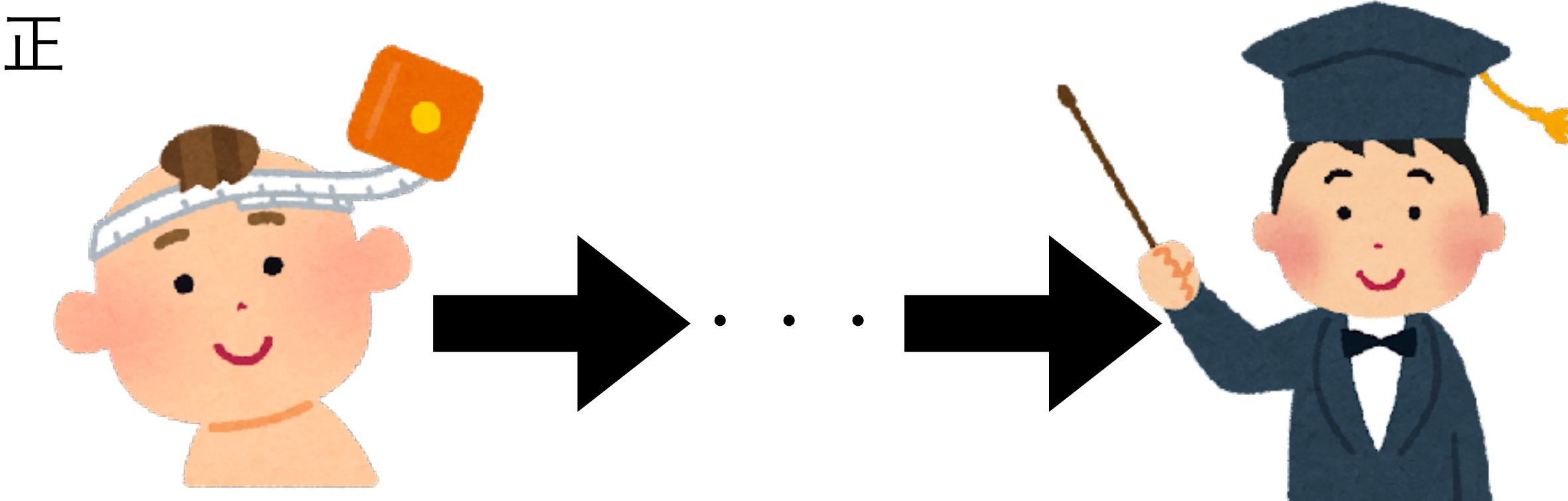
Normal GPR

事前にシミュレーションを行い、カーネル関数そのものを修正
→ 物理的な知識がある状態から最適化を始める

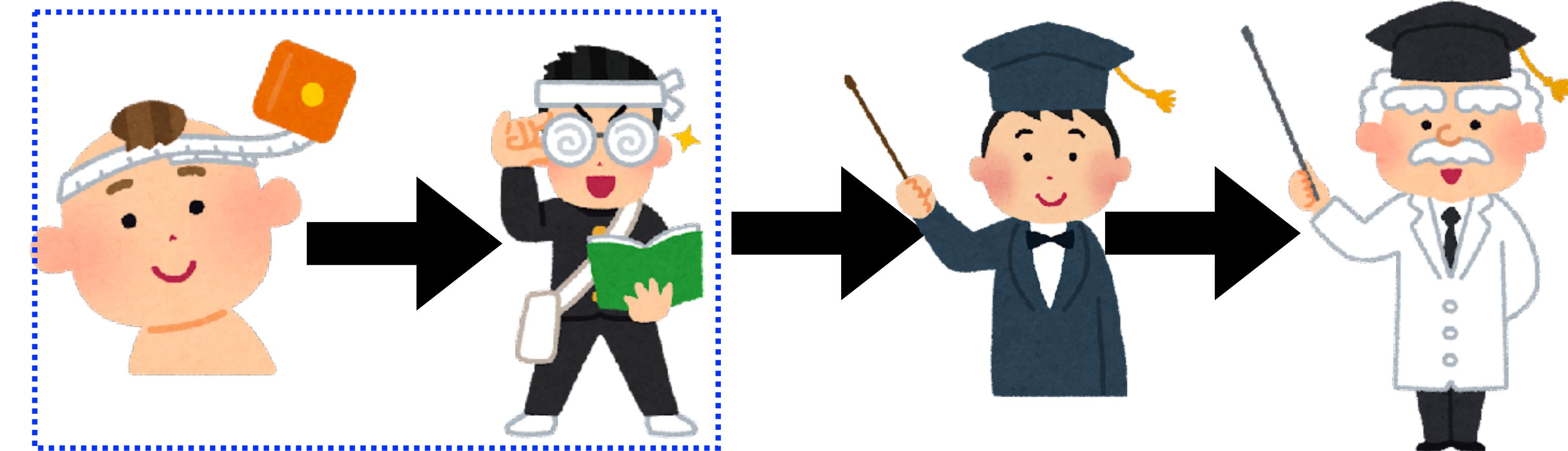


(d) Convergence tests

変数が増えてもより早く
解が収束する。



Physics Informed GPR



シミュレーション

GPR を賢くする

J. Duris, J. et al. Bayesian Optimization
of a Free-Electron Laser. Phys. Rev. Lett. 124, 124801 (2020).

その他課題を解決するためのいくつかの手法

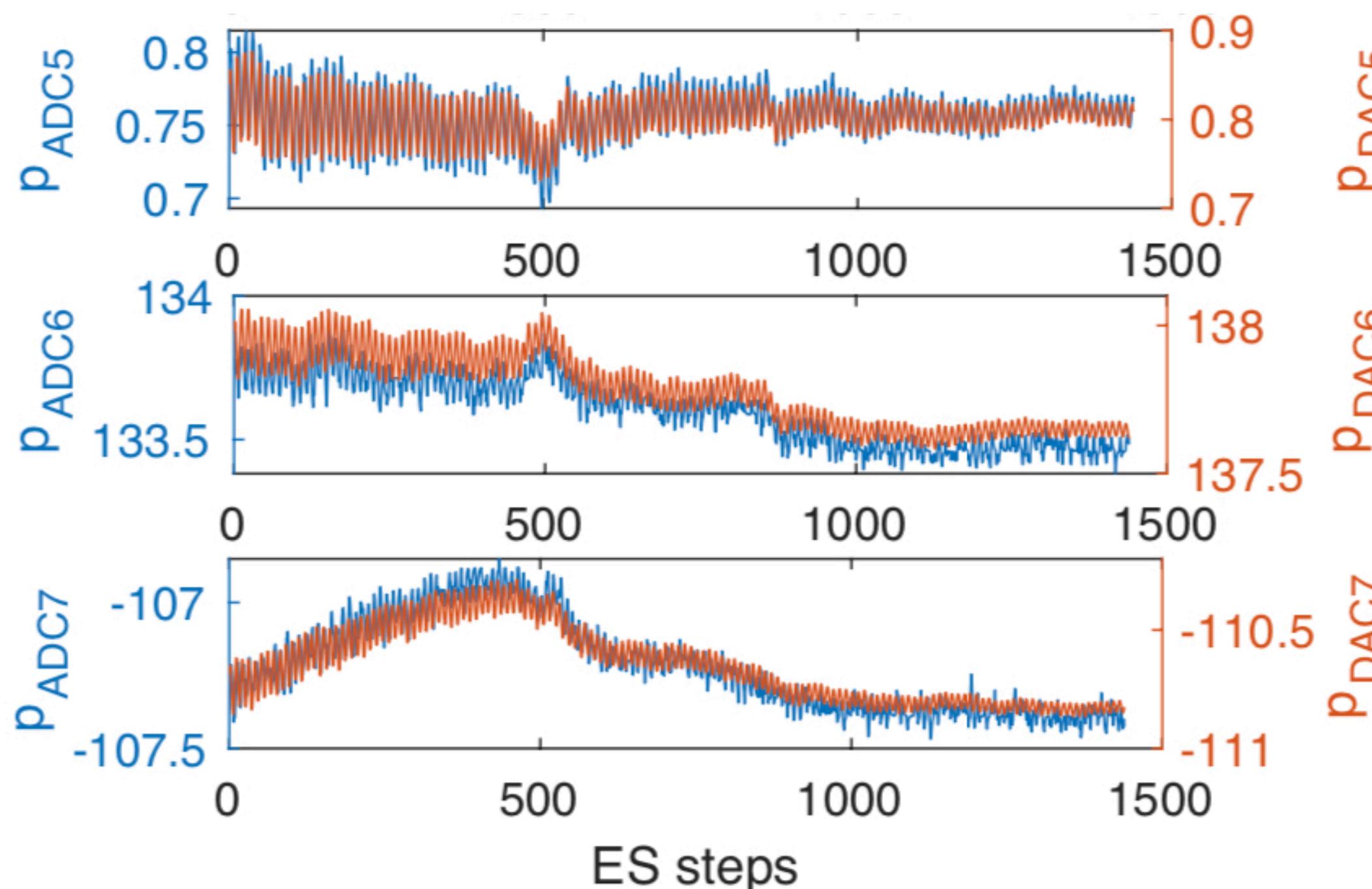
c. Extremum Seeking Control などによる適応制御

Extremum Seeking Control:

パラメータを常に微小に変化させて、目標関数の応答から最適化し続ける手法。

各パラメータの変化を異なる周期で変動させることにより応答を分離する。

→ ノイズとも分離できるため高精度で最適化が可能。



RIBFのあるビームラインでスリットとバッフルの
ビームロスを指標として ESC を適応したときのデータ

※ある程度うまくいったものの、非常に短い時間での
継続的な値の変化に対応するにはハードウェアの更新も
必要と思われる。