Exploring Reinforcement Learning for Optimal Bunch Merge in the AGS and related ML control

Georg Hoffstaetter de Torquat:
Cornell University and Brookhaven National Laboratory

3rd RL4AA Workshop
April 2-4, 2025, Hamburg, Germany

@BrookhavenLab

Desired result: higher proton polarization

- What high-impact operational challenge can be addressed by MI/AI? ➔ Polarized protons.
- From the source to high energy RHIC experiments, more than 20% polarization is lost.
- The EIC asks for 70% proton polarization, which is 5% higher than even a good RHIC run.
- Polarized luminosity for longitudinal collisions scales with P^4, i.e., a factor of 2 reduction!
- The proton polarization chain depends on many delicate accelerator settings form Linac to the Booster, the AGS, and the RHIC ramp.
- Even 5% more polarization would be a significant achievement.
- Approximately 2/3 of the polarization loss is in the injector chain.
- Accelerator time in RHIC is much less available than in the injector chain.

➔ Focus: polarization increase from the injector chain.

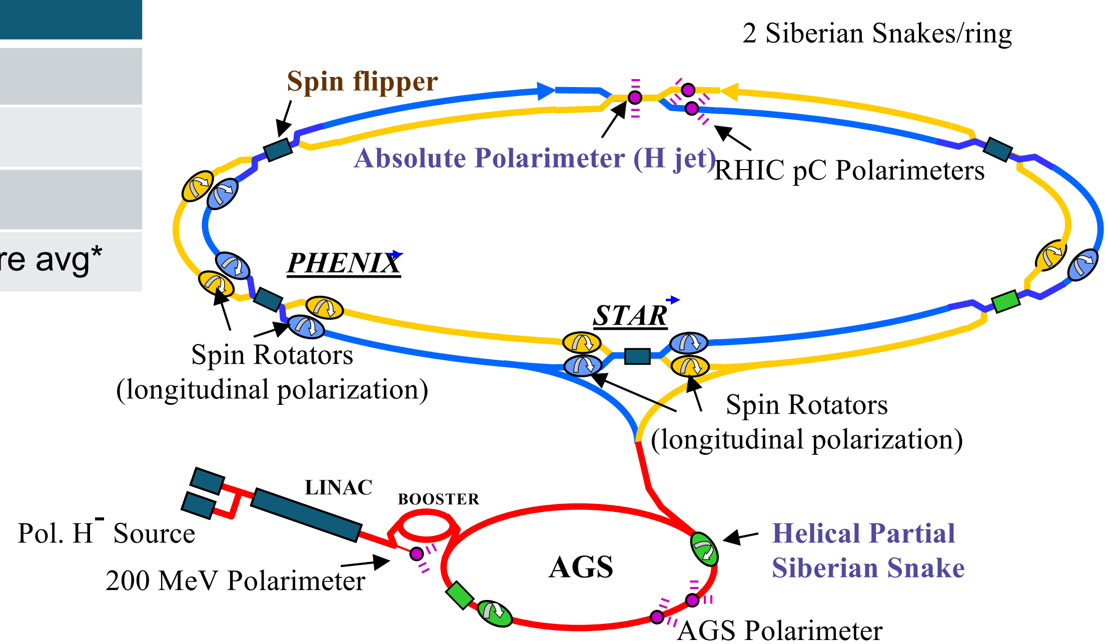
The polarized proton accelerator chain

Brookhaven National Laboratory

RHIC Polarized Beam Complex

	Max tot. Energy [GeV]	Pol. At Max Energy [%]	Polarimeter
Source+Linac	1.1	82-84	
Booster	2.5	~80-84	
AGS	23.8	67-70	p-Carbon
RHIC	255	55-60	Jet, full store avg*

* Includes both ramp loss and store decay



	Relative Ramp Polarization Loss (Run 17, full run avg)
AGS	17 %
RHIC	8 %

Topics that can improve polarization

- (1) Emittance reduction
- (2) More accurate timing of tune jumps
- (3) Reduction of resonance driving terms

Optimizers for different applications

less ← assumed knowledge of machine → more

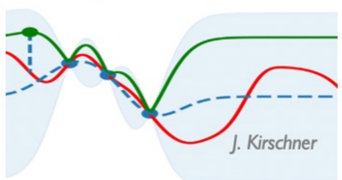
Model-Free Optimization

Observe performance change after a setting adjustment

→ estimate direction or apply heuristics toward improvement

gradient descent
simplex
ES

Model-guided Optimization

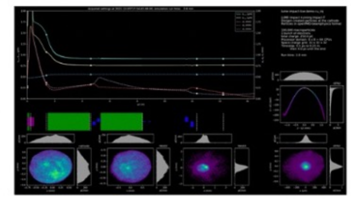


Update a model at each step

→ use model to help select the next point

Bayesian optimization
reinforcement learning

Global Modeling + Feed-forward Corrections



Make fast system model

→ provide initial guess (i.e. warm start) for settings or fast compensation

ML system models +
inverse models

Courtesy Auralee Edelen

Brookhaven National Laboratory

Characteristics of polarization optimizations

1. Optimal parameter settings are hard to find, and the optimum is difficult to maintain.

2. The data to optimize on has significant uncertainties.

3. Good, approximate models of the accelerator exist.

4. A history of much data is available.

Is this type of problem suitable for Machine Learning?

Why would ML be better suited than other optimizers and feedbacks?

Reinforcement Learning for accelerators

Extension of our initial goal of using physics-informed Bayesian Optimization: Can RL have advantages over BO for accelerator controls?

Operational experience: Many system parameters are measured in accelerators, where many are found to be implicitly related to the performance optimum, even if the accelerator model does not capture this relationship. ➔ Experience to golden hand operators.

+ RL uses more data about the accelerator (as state variables) even if relationship to the optimum is not known.

+ RL follows an optimum setting, even when the system changes ➔ accelerator control not only optimization.

-- RL requires millions of data points and may seem inapplicable to accelerators, but with an improved model may deliver many of these points, making RL feasible.

Reinforcement Learning empirically learns the relationships between system parameters and objectives, even if they are not closely related by accelerator models.

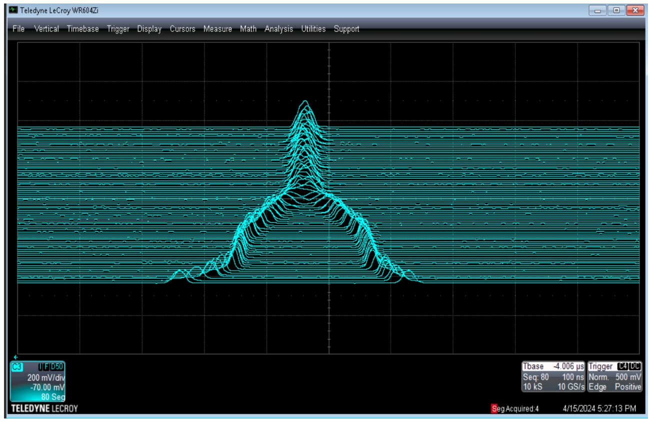
This relationship remains useful, even under small changes ➔ RL follows the optimum.

First RL application for more proton polarization at BNL: Bunch merging by RF manipulation, because an exceptionally good model exists.

Brookhaven National Laboratory

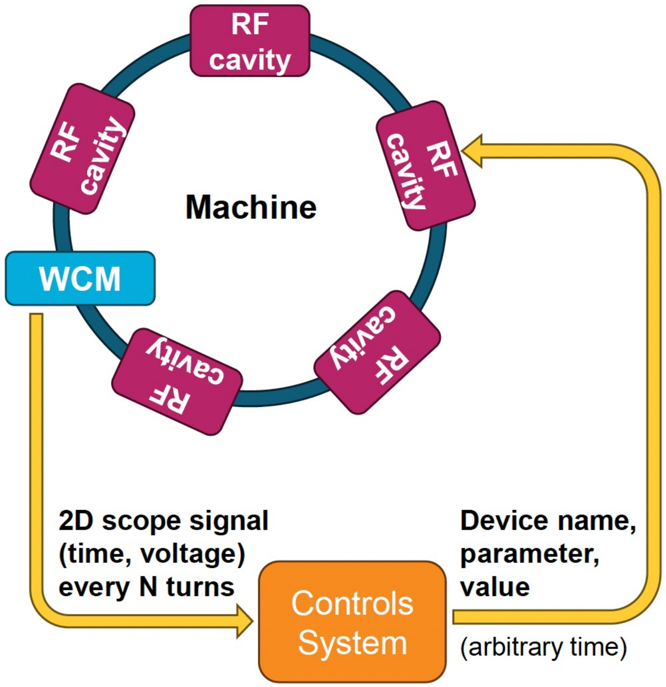
AGS Bunch Merging

- Before transferring to AGS, beam bunch is split into 2 longitudinally to reduce the space charge effect
  -> reduce emittance -> improve polarization
- Bunches are later merged before AGS extraction;
- Requires expert tuning of many parameters:
  - Prone to drift over time;
  - Time consuming;
- Controls: RF voltages, phases
- Goal: Obtain a "good" merged bunch profile:
  - Emittance preservation:
    - No particle lost;
    - Gaussian shape;
    - No "baby" bunches;
  - Stable final bunch profile:
    - Not shifting left to right;
    - Not bouncing up and down;
    - Merged in the center;



Real mountain range data showing 2-to-1 bunch merge in AGS

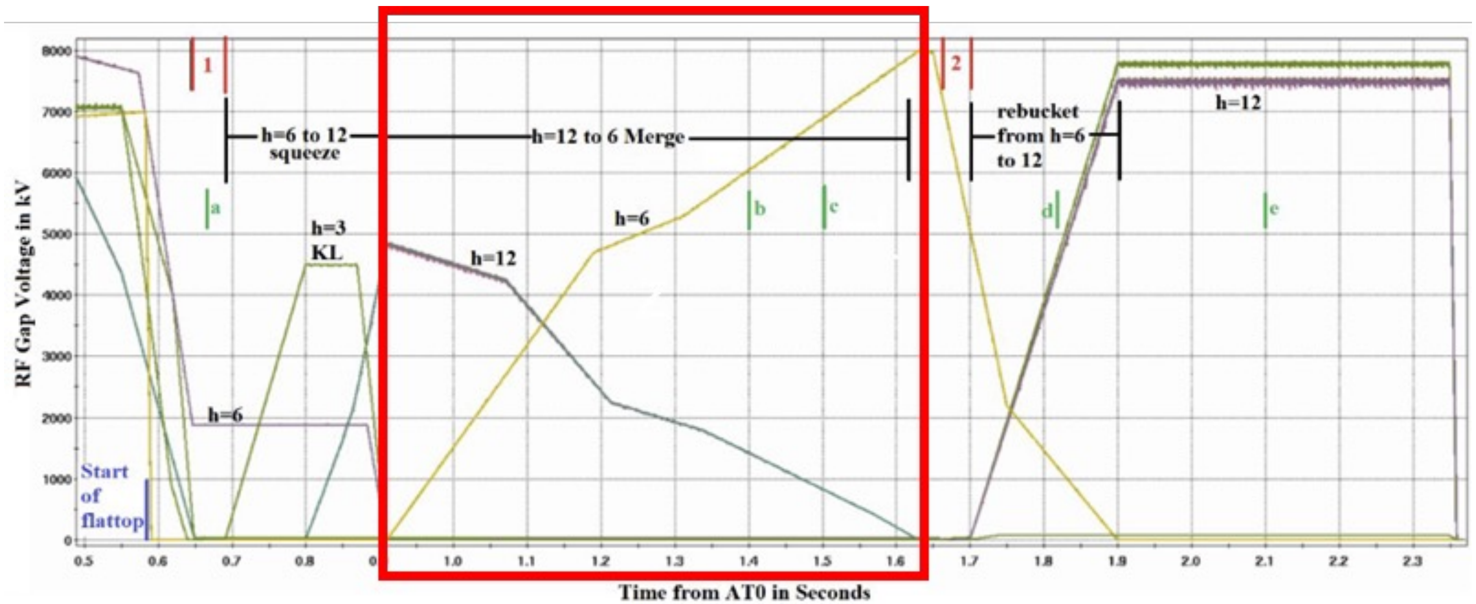
Wall current monitor (WCM) generates voltage vs time signal. Each separated in time by N turns (N accelerator periods)



Cartoon representation of accelerator with WCM, RF cavities (arbitrary number), and input/output

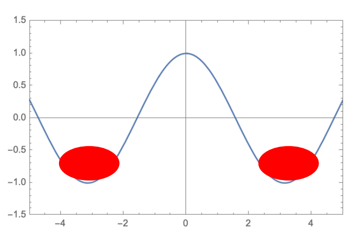
Bunch splitting in Booster / merging in AGS

Splitting in the Booster and merging after AGS accelerator reduces space charge and emittance growth ➔ more polarization

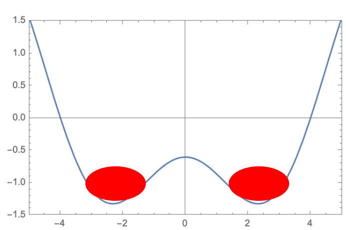


Three RF amplitudes (h=3, 6, 12) in the AGS during bucket manipulation and merging.

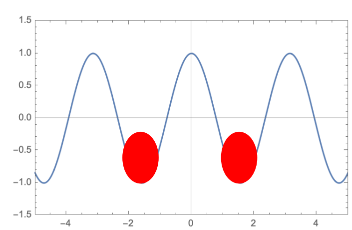
➔ We have set up Reinforcement Learning for the merging section.



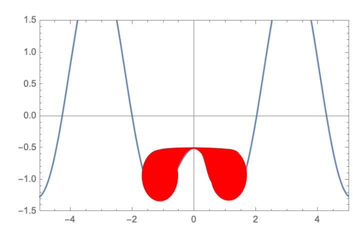
Accelerating RF h=6



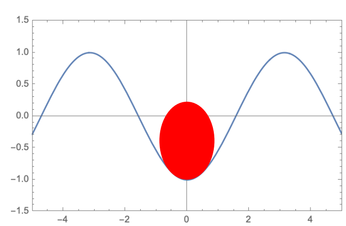
Attracting RF h=3



Close bucketing h=12

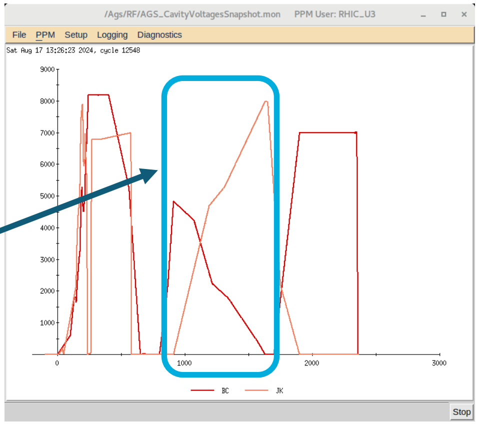


Combining h=6

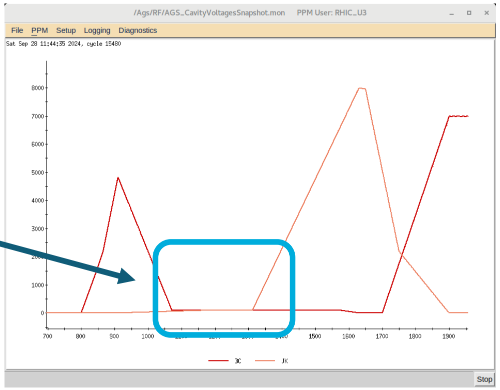


Final bucketing h=6

Machine Setup

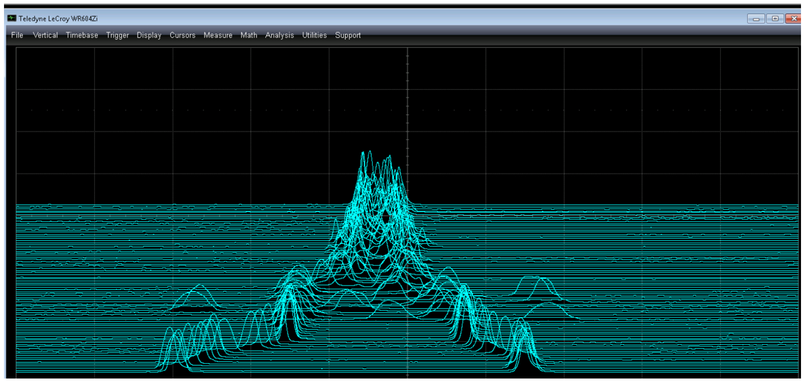


- Bunch merge uses cavity harmonic number 6 and 12:
  - Merge period ~ 900-1650 ms;
  - Three cavities in harmonic 12 (B, BC, C), two in harmonic 6 (JK, K);
  - The settings are initially set to close to 0, so the RFs cannot merge the bunches well;
  - RL agent is then applied to correct the RF settings;



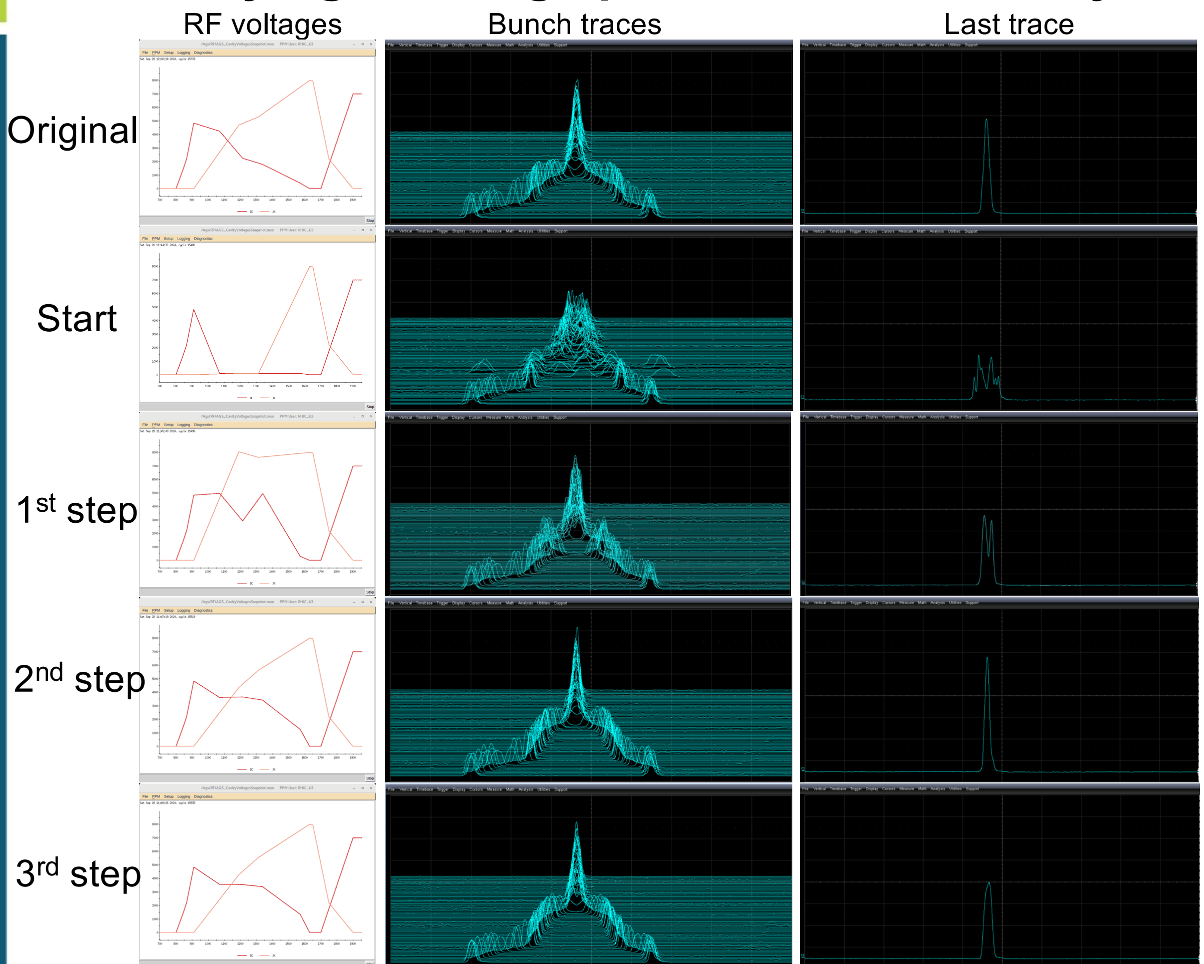
- Input: using system starting and ending voltages for each cavity;
  - Rf h12: 4 voltages;
  - Rf h6: 2 voltages;
  - Phase difference fixed to be pi / 2;

- Output:
  - **Final emittance; -> Used as the rewards**
  - Final bunch position;
  - Final bunch length;
  - Bunch position variation;
  - Bunch length variation;



WCM signals show the initial RFs cannot merge the bunches well

Reinforcement Learning Tuning
test - varying 6 voltage points for each RF system



Goal: minimize the longitudinal emittance after bunch merging

RF amplitudes as function of time have been optimized in experiments.

Automatic readout of longitudinal emittance not yet available, test used simulated bunch lengths as reward.

Plan: check whether Reinforcement Learning has advantages over BO.

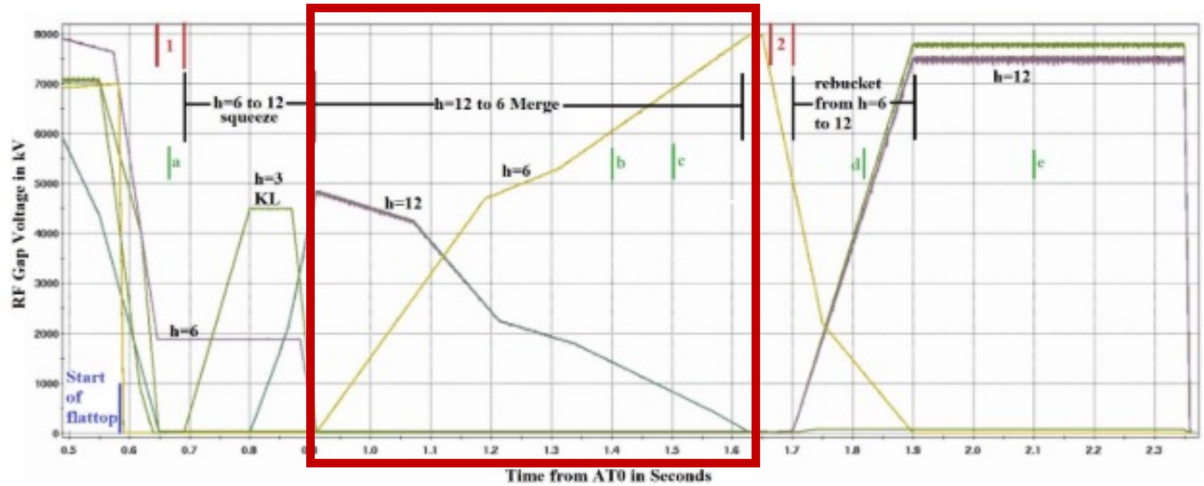
Plan: Include also RF phases as actors and coherent oscillations as state variables.

Determine useful state variables

- measurable
- related to the reward

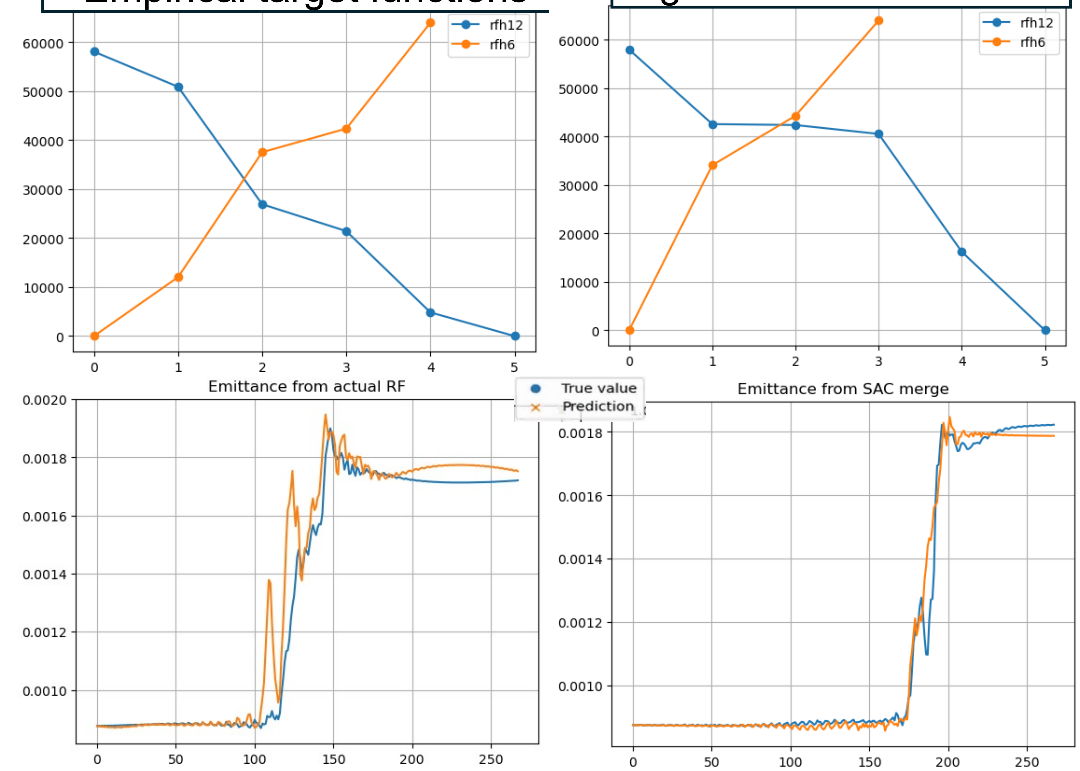
Simulation Results

- First simulator in Bmad: **Slow**;
- **Julia simulator;**
- SAC agent: 10,000 initial samples + 4,000 training steps;
- 3% emittance growth;



Empirical target functions

Agent learned functions



To heavy reliance on the model leads to 3% emittance growth

System Side Preparations

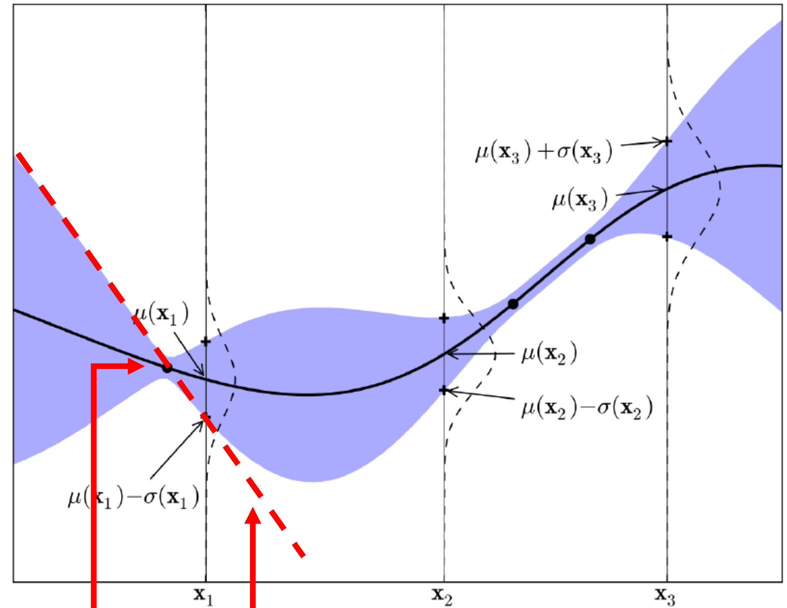
- A Python manager is developed to collect bunch information from live machine data; This application will be used by the RL in the next machine test;
- We can now acquire input signals from the wall current monitors (offloading oscilloscope data) and show results in a Jupyter notebook.
  - Evaluating multiple commercially available digitizer products for performance comparison to FPGA-based system;
  - Work on buffer memory implementation (to store multiple turns);
  - Work on the hardware configurations to match the actual system specifications;
- We have a Zynq Ultrascale FPGA evaluation board and an FMC expansion card to digitize the WCM signals:
  - 12-bit conversion at 1,000 Mega samples per second

Optimization with Gaussian Processes

- GP model built with scikit-learn library
- A probability distribution over possible functions that fit a set of points
- Mean function + Covariance function

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$$

- Kernel: covariance function $k(x_i, x_j)$ of the input variables
- Covariance matrix $\mathrm{K} = k(X, X) = \begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_t) \\ \vdots & \ddots & \vdots \\ k(x_t, x_1) & \cdots & k(x_t, x_t) \end{bmatrix}$
- At a sample point x_i, Gaussian process returns mean $\mu(x_i|X) = m(x_i) + k(x_i, X)K^{-1}\big(f(X) - m(X)\big)$ and variance $\sigma^2(x_i|X) = k(x_i, x_i) - k(x_i, X)K^{-1}k(X, x_i)$



2. The data to optimize on has significant uncertainties.
3. Models of the accelerator exist

Polarized collider performance vs. beam intensity

Collider luminosity, $\mathcal{L}$

$$\mathcal{L} \propto \frac{N^2}{\varepsilon}$$

N = intensity/ bunch
ε = tran. emittance

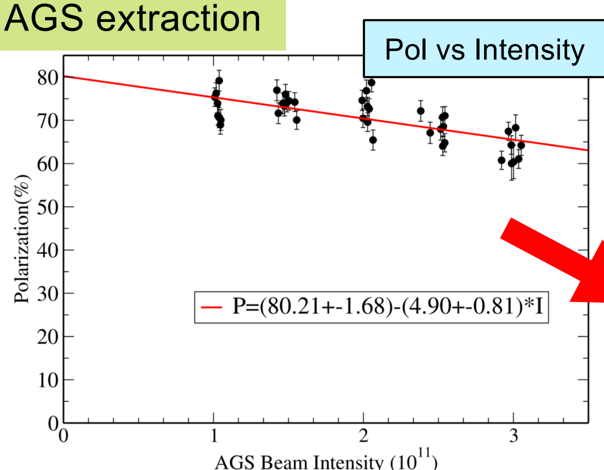
Polarized collider figure of merit (for polarization P):

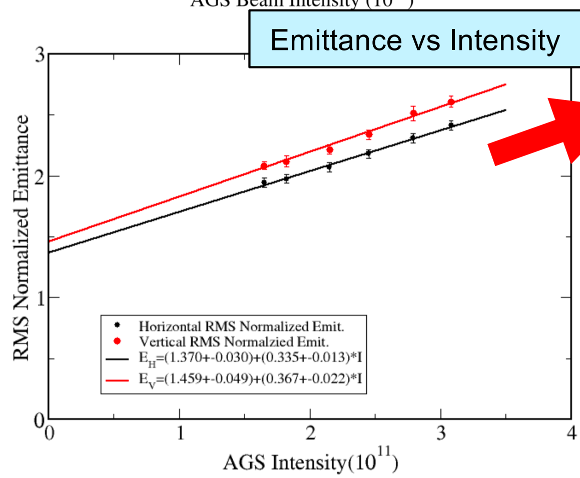
$$\text{FoM} = \begin{cases} \mathcal{L}\, P^2 & \text{transverse spin} \\ \mathcal{L}\, P^4 & \text{longitudinal spin} \end{cases}$$

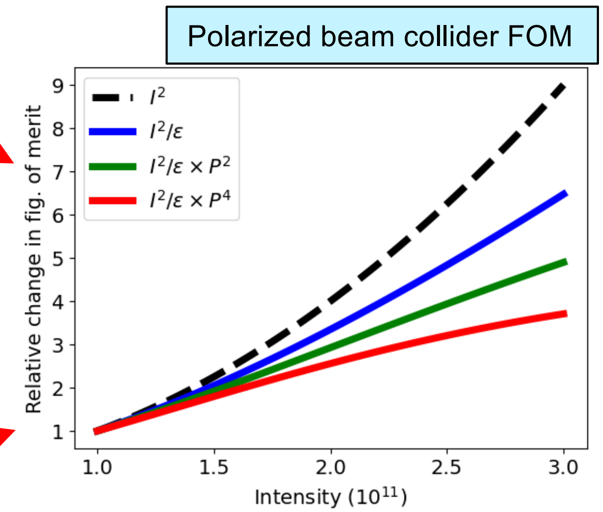
Since both emittance and polarization degrade with intensity figure of merit decreases rapidly

FoM dependence on intensity closer to linear in N than quadratic.

AGS extraction







Impact of intensity increase on FoM given emittance and polarization dependence at AGS extraction

Emittance reduction ➔ less depolarization

To reduce and maintain emittances we

- optimize Linac to Booster transfer
- optimize Booster to AGS transfer
- correct optics and orbit in Booster and AGS
- use orbit responses to calibrate models of Booster and AGS.
- split bunches in the Booster for space charge reduction and re-merge them at AGS top energy.

Xopt Overview

https://christophermayes.github.io/Xopt/

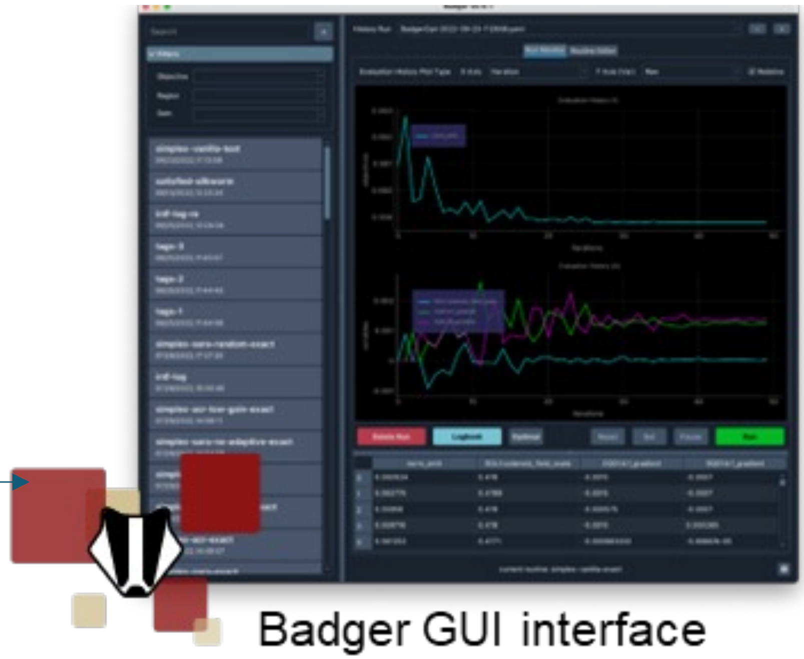
Text input

YAML file

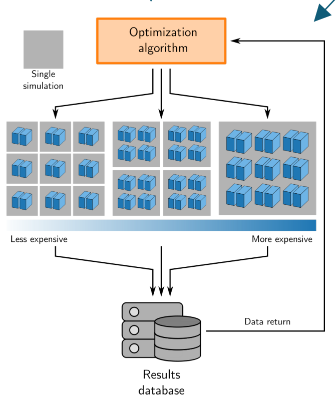
```yaml
xopt:
    max_evaluations: 6400

generator:
    name: cnsga
    population_size: 64
    population_file: test.csv
    output_path: .

evaluator:
    function: xopt.resources.test_functions.tnk.evaluate_TNK
    function_kwargs:
      raise_probability: 0.1

vocs:
    variables:
        x1: [0, 3.14159]
        x2: [0, 3.14159]
    objectives: {y1: MINIMIZE, y2: MINIMIZE}
    constraints:
        c1: [GREATER_THAN, 0]
        c2: [LESS_THAN, 0.5]
    linked_variables: {x9: x1}
    constants: {a: dummy_constant}
```

Python API

```python
# create Xopt object.
X = Xopt(YAML)

# take 10 steps and view data
for _ in range(10):
    X.step()

X.data
```

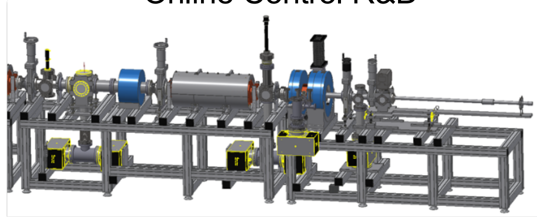


Badger GUI interface



Accelerator simulation

Online Control R&D



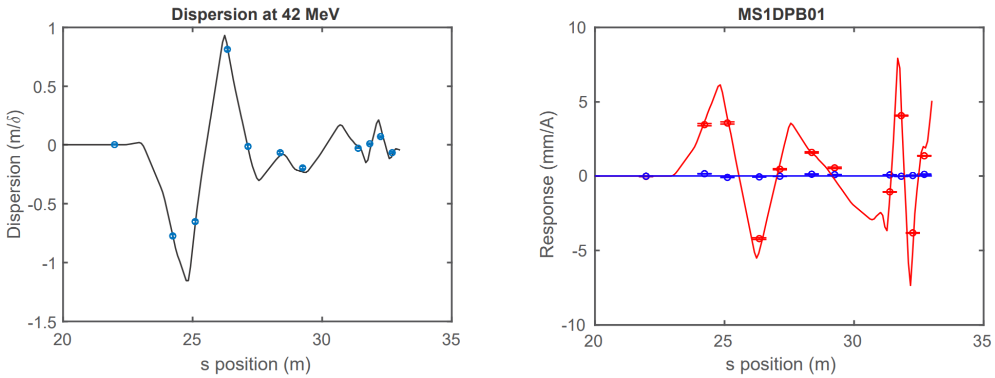
Experiment facility

Xopt implements a number of different algorithms:

- Various Bayesian optimizations:
  - Single/Multi-objective BO, Trust Region BO, Bayesian Algorithm Execution, custom model priors, etc.
- Genetic optimization (CNSGA), RCDS, Nelder-Mead Simplex, Extremum seeking;

Digital Twin for hadron injector sections

A Digital Twin is a bi-directional connection between an accelerator's physics model and its control system.

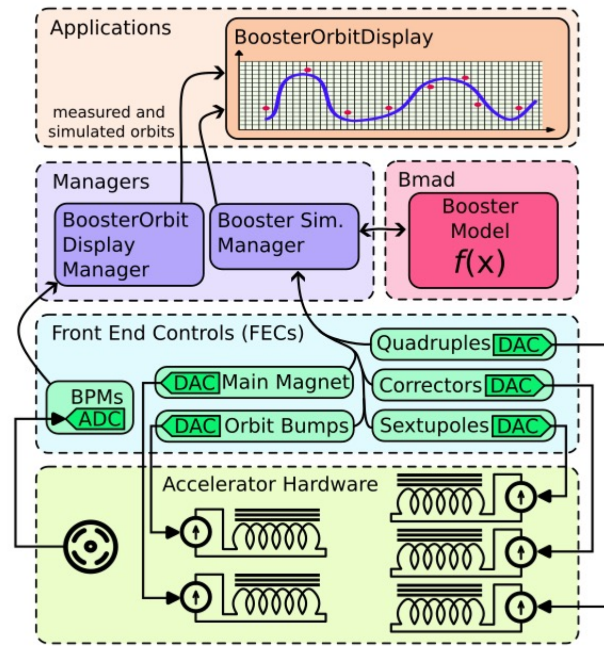


Example digital-twins for CBETA: combine Bmad with EPICS bidirectionally

- Bmad ➔ control system: DT results are displayed by the control system, just like measured accelerator data.
- Control system ➔ DT: Power supply settings automatically load into the physics model.

Great for continuous comparison of operations and model.
Great for offline development of operations procedures.
Great for virtual diagnostics.

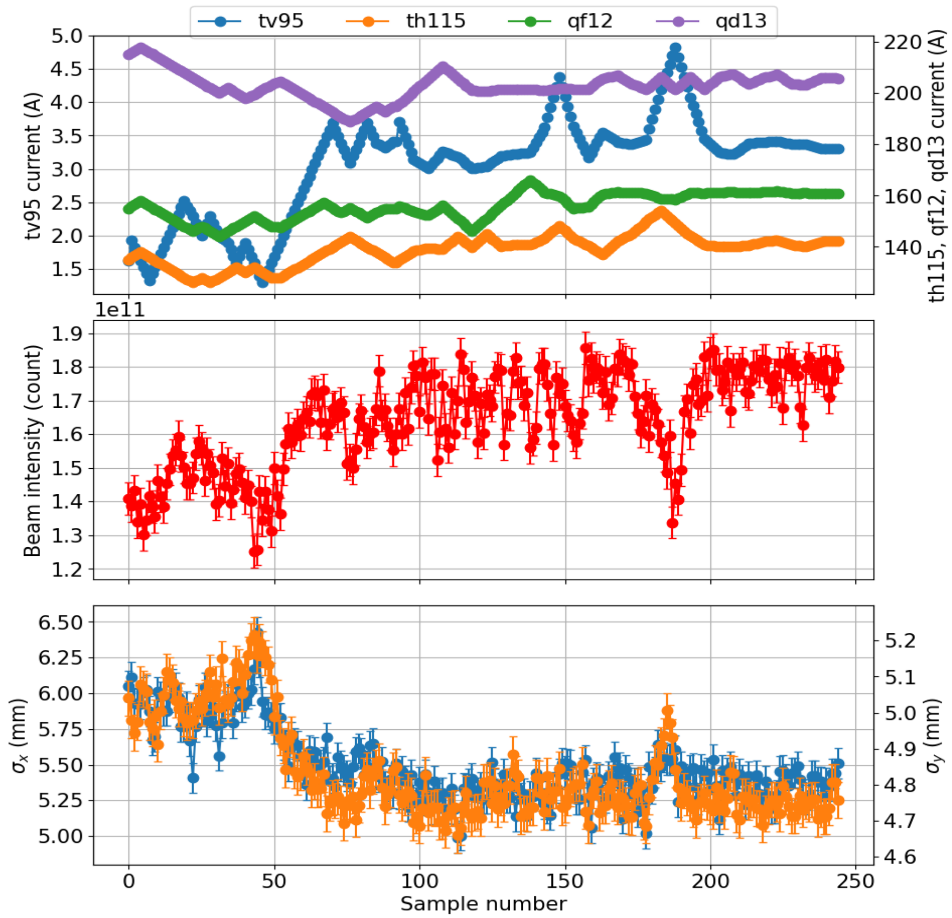
- ➢ Additional benefit: Neural network can be trained to predict slow to simulate beam behavior in operations time, e.g. space charge dynamics.
- ➢ ML control routines always have the up-to-date physics model available.



DT currently being prepared for the Booster.

Result: Automatic BO for Booster injection

- Controls: Power supply currents of two correctors and two quadrupoles at the end of the LtB line
- Beam size decrease in both planes in the BtA line in correspondence with intensity increase



Bayesian optimization of the Booster injection process.

Top: power supply currents of two correctors (tv95, th115) and two quadrupoles (qf12, qd13) in the LtB line.

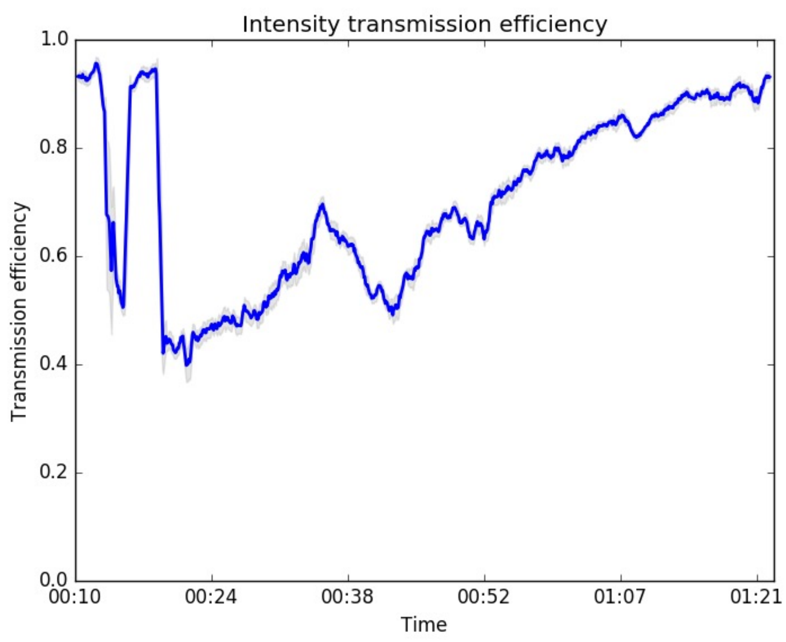
Middle: beam intensity after Booster injection, scaping, and acceleration.

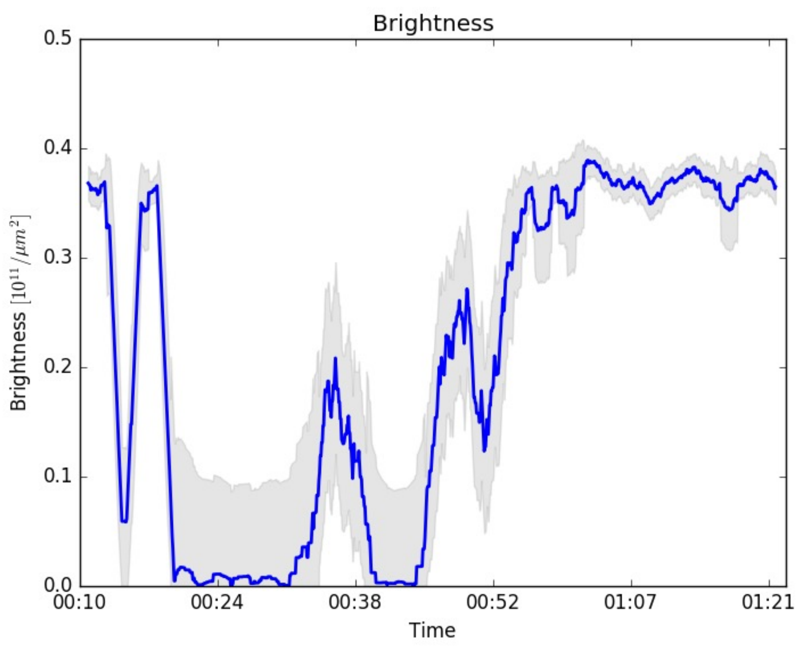
Bottom: Beam size measurements in the BtA line during Bayesian optimization.

Control system: This Bayesian Optimization is now available as a control system application to operators.

Result: Automatic BO for AGS injection

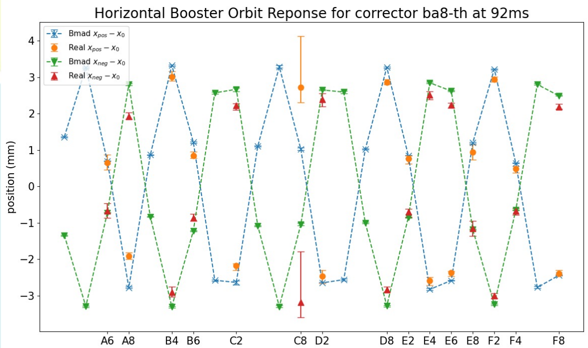
Algorithm efficiently found settings that were different, but at least as good as the previously optimized ones, automatically maintain the AGS injection at optimal performance without human intervention.

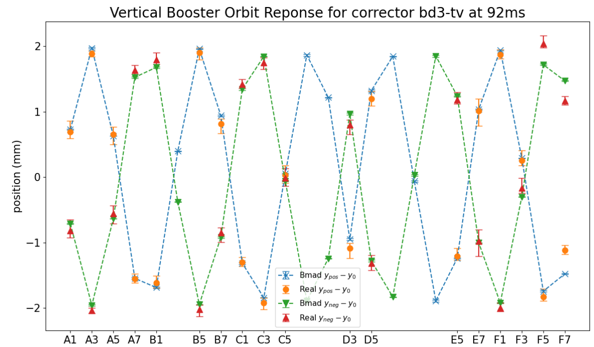




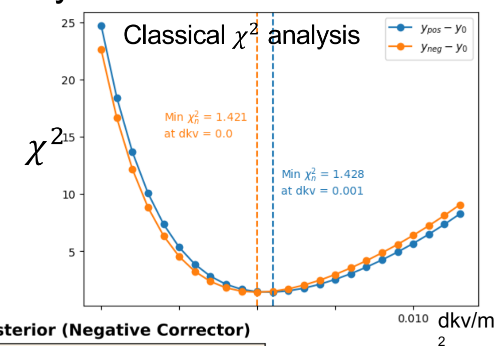
➔ Optimization of current while observing the brightness.

Uncertainty Quantification from orbit responses in the Booster





Orbit response data can be used to find and quantify unknown parameters (e.g., power supply scaling factors, magnet misalignment etc.) in real accelerators, by Lucy Lin, Nathan Urban, and Christopher Kelly.

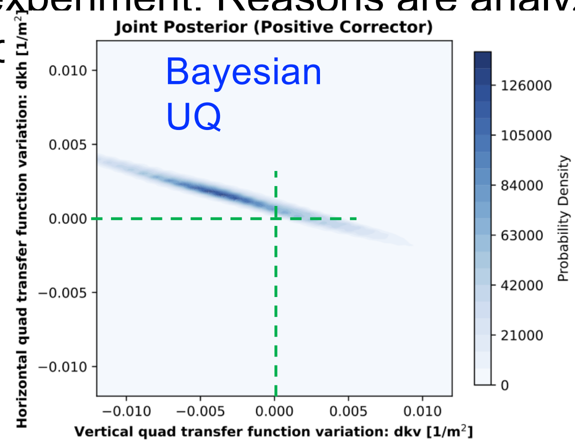


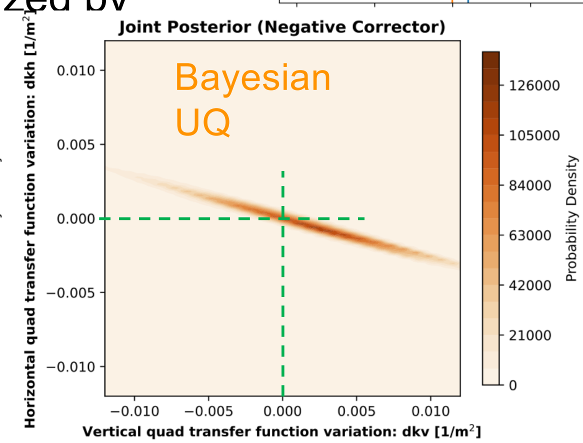
➔ Good agreements between Booster data and Bmad model are reached, with small discrepancies between model and measurement (within 1 mm)

➔ chi-squared/DF = 1.4 for model-experiment. Reasons are analyzed by
(a) Least square fitting to reduc
(b) Uncertainty Quantification.

➔ The main power supply transfer functions (a) do not reduce χ^2, (b) their UQ is consistent with 0

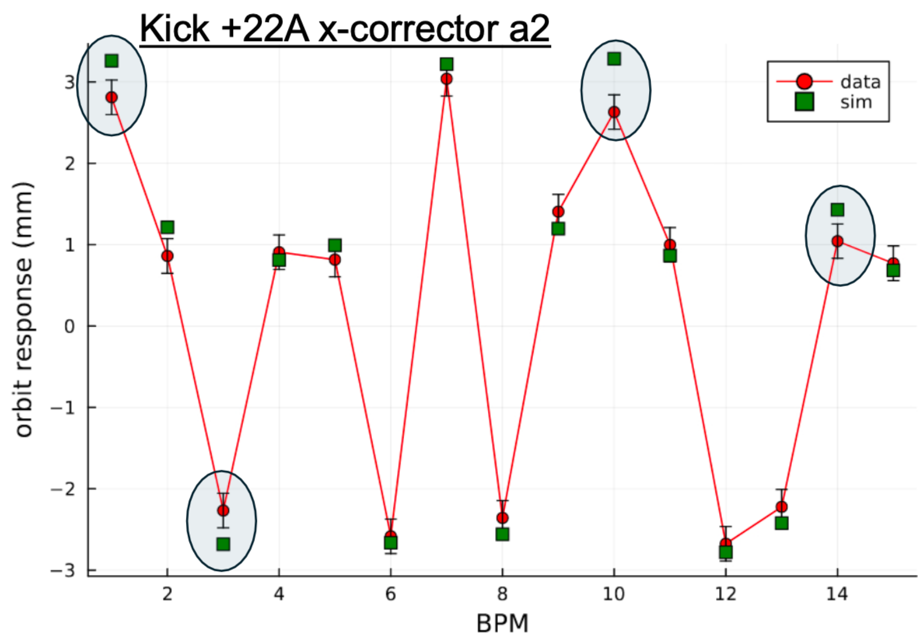
➔ Other error sources are being analyzed.

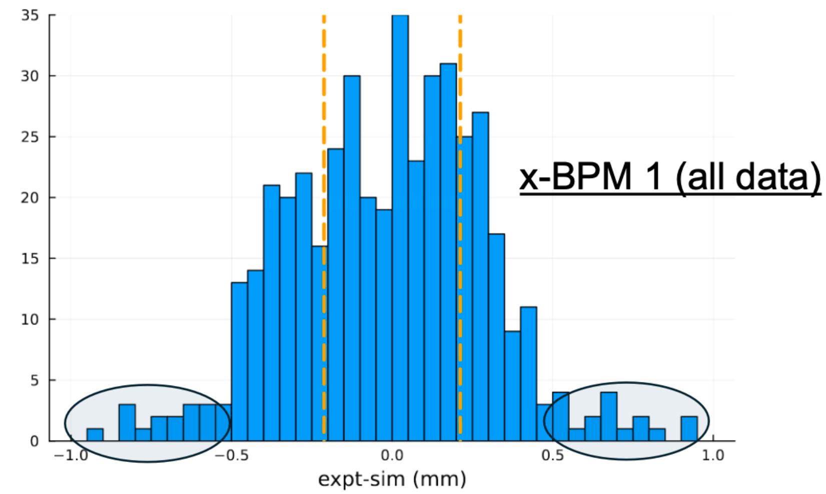


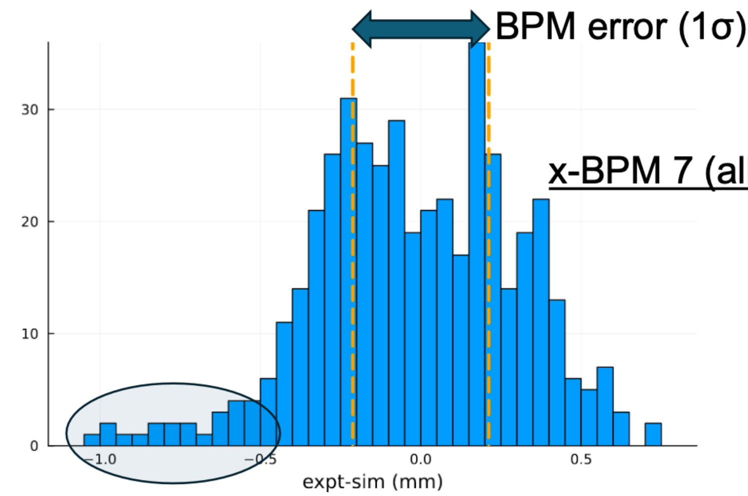


Simulation error

Sim. orbit responses can deviate from data far outside of BPM uncertainty for unknown reasons







Brookhaven

Bayesian UQ

- Bayesian UQ to probe and quantify sources of simulation error.
- Inputs are *probability distributions*:
  - "**Likelihood**": distribution of data given params
    normal centered on simulation

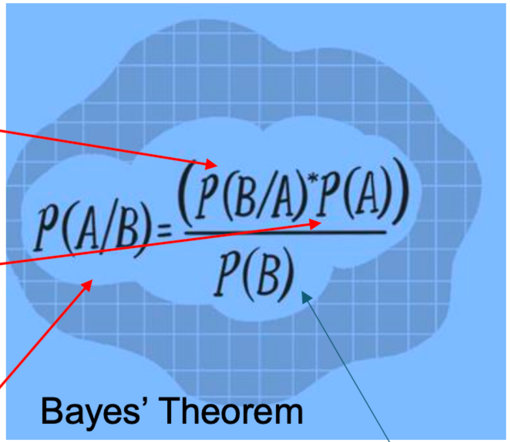
$$\mu = \mathrm{sim}(\mathrm{perturbed}; \mathrm{params}) - \mathrm{sim}(\mathrm{unperturbed}; \mathrm{params})$$
$$\sigma = \sqrt{2} \times \mathrm{bpm\ err.}$$

  - "**Priors**": expert knowledge of *parameters, e.g.*

Some additive parameter: $\mathrm{additive}_i \sim \mathrm{Normal}(\mu_i, \sigma_i)$

Some multiplicative parameter: $\mathrm{multiplicative}_i \sim \mathrm{LogNormal}(\mu_i, \sigma_i)$

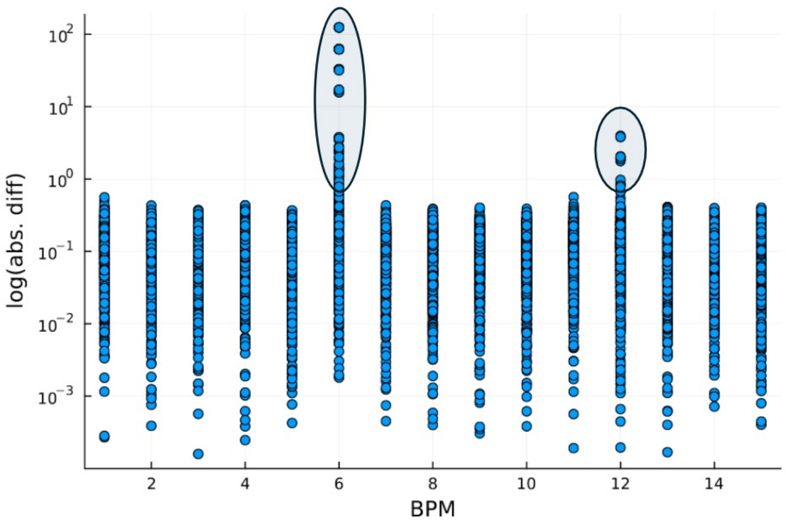
- Output is the "**posterior**": distribution of the parameters given the data
- Sample using Markov chain Monte Carlo methods via the Julia "Turing" package.



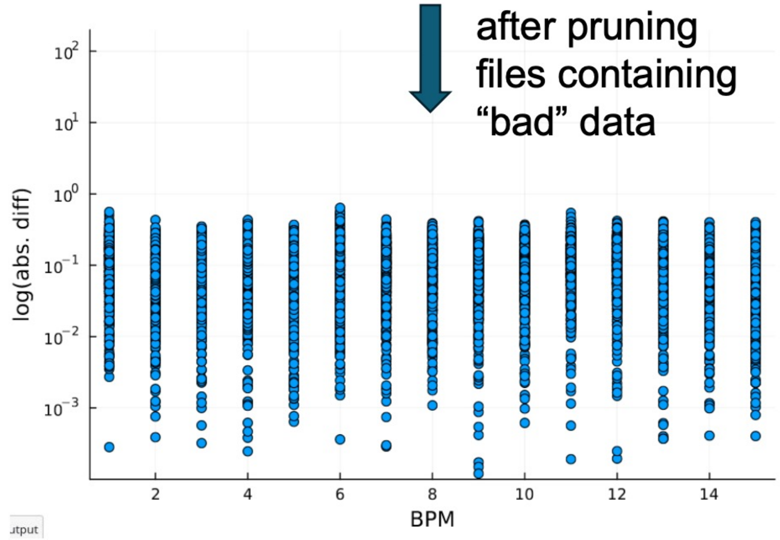
("Evidence" not needed for MCMC)

Dataset

- The “2022” dataset contains orbit measurements where each corrector in turn is set to -22A, 0A and +22A.
- 3-5 measurements per corrector setting.
- Focus on x-plane orbit. Build orbit responses
  - POSITIVE: +22A – 0A
  - NEGATIVE: -22A – 0A
- Studied fluctuation between meas. of each orbit response over all possible pairings.
  - Subtract simulation result to account for current fluctuations
  - Recenter about mean to focus on spread.
- Found a number of outliers in BPMs 6 and 12 that required pruning.
- (BPM 6 has known issues)



after pruning files containing “bad” data

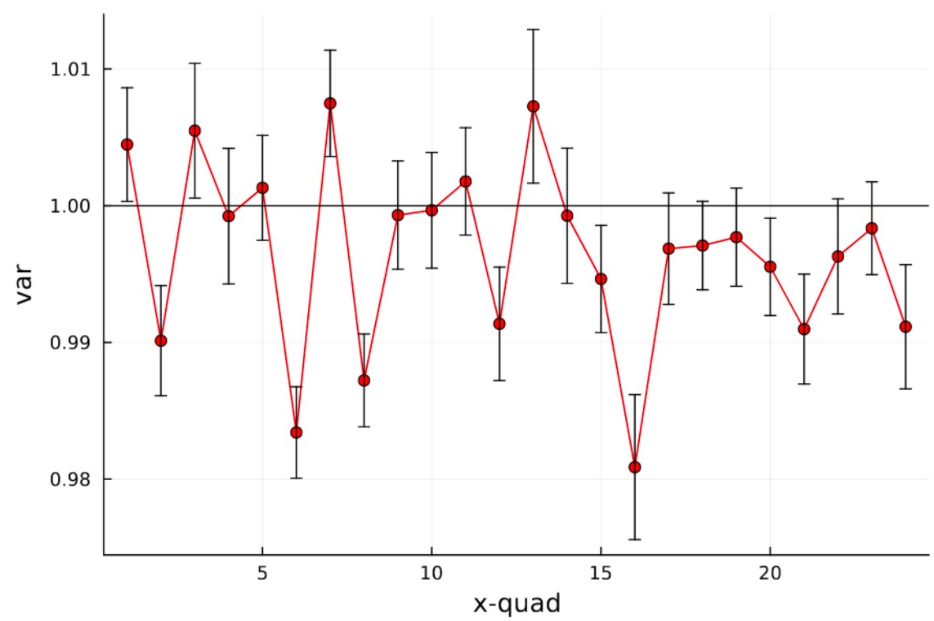


Brookhaven National Laboratory

Preliminary UQ result

- Performed UQ conditioned on 45 orbit response measurements
  - 1 pairing (after+before) for each perturbed corrector
- Probabilistic model includes prior/likelihood distributions for
  - ‘var’ values (LogNorm 1.0 ± 0.01)
  - Measurement errors (Normal, σ = sqrt(2) * BPM err ~ 0.21 mm)
- Does not account for
  - Measurement errors on readback currents on static magnets or correctors
  - Any uncertainty on machine characteristics or other internal params (e.g. coefficients in transfer func)
- Observe interesting pattern in output var values, some well distinguished from unity.
- Possible hint for origin of discrepancies?

‘var’ values in posterior



SciBmad a ML-oriented Toolkits (Libraries)

Advantages the toolkit:

Fully differentiable (reverse and forward)

➔ excellent for Neural Network optimizations

➔ Excellent for Bayesian optimization with slope information

- Cuts down on the *time* needed to develop programs.
- Cuts down on programming *errors* (via module reuse).
- Provides a simple mechanism for lattice function calculations from within control system programs.
- *Standardizes* sharing of lattice information between programs.
- Increased *safety*: Modular code provides a firewall. For example, a buggy module introduced into the toolkit will not affect programs that do not use it.

This project is

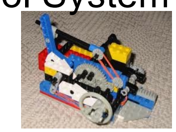
- funded by DOE-HEP
- has a growing list of collaborators
- has a weekly wise people meetings
- ➔ **is looking for collaborators**

Toolkit

Dynamic Aperture Program

Lattice Design Program

Control System Programs



IBS Simulation Programs

Etc.

Brookhaven National Laboratory

Acknowledgement

Brookhaven National Laboratory

- Kevin Brown, Michael Costanzo, Matt Cowan, Weinin Dai, Bahwin Dhital, Zhihua Dong, Yuan Gao, Levente Hajdu, Kiel Hock, Georg Hoffstaetter de Torquat, Bohang Huang, Natalie Isenberg, Lucy Lin, Chuyu Liu, John Morris, Linh Nguyen, Yihui Ren, Vincent Schoefer, Shruti Tajne, Nathan Urban, Shinjae Yoo, Keith Zeno

CLASSE – Cornell Laboratory for Accelerator-based ScienceS & Education

- Joseph Devlin, Eiad Hamwi, Georg Hoffstaetter de Torquat, Daria Kuzovkova, Lixing Li, David Sagan, Matthew Signorelli, Jonathan Unger, Sophia Yang

Rensselaer

- Yinan Wang

SLAC National Accelerator Laboratory

- Auralee Edelen, Ryan Roussel

Jefferson Lab – Thomas Jefferson National Accelerator Facility

- Armen Kasparian, Malachi Schram, Kishan Rajput

radiasoft

Nathan Cook, Jon Edelen, Chris Hall

Thank you and Questions?