

Bayesian optimization experiment for trajectory alignment at the low energy RHIC electron cooling system

Y. Gao^{1,*} W. Lin^{2,†} K. A. Brown^{1,3} X. Gu,¹ G. H. Hoffstaetter^{1,2,4}
J. Morris,¹ and S. Seletskiy¹

¹*Collider-Accelerator Department, Brookhaven National Laboratory, Upton, New York 11973, USA*

²*Cornell Laboratory for Accelerator Based Sciences and Education, Cornell University, Ithaca, New York 14853, USA*

³*Electrical and Computer Engineering Department, Stony Brook University, Stony Brook, New York 11794, USA*

⁴*Electron-Ion Collider Department, Brookhaven National Laboratory, Upton, New York 11973, USA*



(Received 2 September 2021; accepted 22 December 2021; published 7 January 2022)

The low energy RHIC electron cooling (LEReC) system is the world's first electron cooler utilizing radio frequency (rf) accelerated electron bunches, and a nonmagnetized electron beam. It is also the first electron cooler applied directly to colliding hadron beams. The unique approach to cooling makes beam dynamics in LEReC very different from the conventional electron coolers. Numerous LEReC parameters can affect the cooling rate. One of the most critical factors is the alignment of the electron and ion trajectories in the cooling section. In this work, we apply Bayesian optimization to check and if needed to optimize the trajectories' alignment. Experimental results are presented and it is demonstrated that machine learning (ML) methods can be applied to perform the control tasks effectively in the RHIC controls system.

DOI: [10.1103/PhysRevAccelBeams.25.014601](https://doi.org/10.1103/PhysRevAccelBeams.25.014601)

I. INTRODUCTION

To increase the collision rate [1] at the Relativistic Heavy Ion Collider (RHIC), the Collider-Accelerator Department (C-AD) at Brookhaven National Laboratory (BNL) developed, commissioned and operated the Low Energy RHIC electron Cooler (LEReC) [2,3]. LEReC was successfully used to increase the luminosity in 2020 and 2021 runs.

LEReC is the world's first electron cooler where electron bunches are accelerated with an rf linac. The layout is shown in Fig. 1. The electrons are generated from the 400 keV photo-gun and accelerated to the designed energy of 1.6–2 MeV (depending on the energy of the ions to be cooled) in the 704 MHz superconducting rf cavity. Through the transport line, the electrons are delivered first to the cooling section (CS) in the “Yellow” RHIC ring and then, passing the 180 degree bend, to the CS in the “Blue” RHIC ring, thus cooling the ion bunches in both rings of the collider. Finally, the electron beam is extracted from the Blue CS and dumped in the beam dump. The electron bunches are repeated with

704 MHz frequency and are “packed” into 9 MHz macrobunches (each contains 30–36 electron bunches), which match the frequency of the RHIC ion bunches. Hence, on its passage through the CS each ion bunch interacts with an electron macrobunch containing 30–36 electron bunches.

In the cooling section the ions experience a friction force from the co-propagating electrons. As a result of the electron-ion interaction both the momentum spread and the angular spread of the ion bunch is getting reduced and its phase-space density is getting increased [4,5].

The cooling force experienced by ions strongly depends on both the angular spread of the electron bunches and on the relative electron beam trajectory angles with respect to the ion beam.

The ion beam trajectory in the 20 m long LEReC cooling section is a straight line due to the high magnetic rigidity of the ions. The electron beam has low magnetic rigidity and is easily steered by not-perfectly aligned short CS solenoids located every 3 meters and by the transverse space charge of the ion beam. To keep electron beam trajectory angles small, each CS solenoid is combined with a pair of horizontal/vertical trajectory correctors. There is a Beam Position Monitor (BPM) installed downstream of each CS solenoid and it is capable of measuring both the ion and the electron beam positions.

The requirements for aligning electron-ion trajectories can be converted to requirements for matching electron and ion beam positions at each of the 8 CS BPMs with 100 μm

*ygao@bnl.gov

†wl674@cornell.edu

Published by the American Physical Society under the terms of the *Creative Commons Attribution 4.0 International* license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

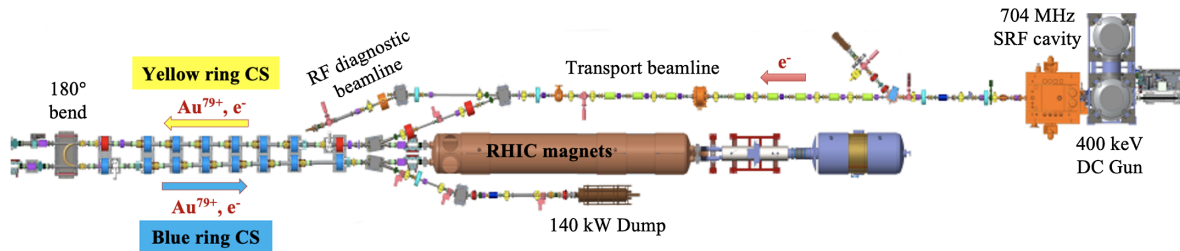


FIG. 1. LEReC system layout.

accuracy in both horizontal (x) and vertical (y) directions [6]. The ion and electron beam positions at the BPM locations are analyzed by separate measurement modules, filtering the incoming signal with 9 MHz frequency for the ions and 704 MHz frequency for the electrons. The procedure for relative calibration of the ion and electron position measurements is described in [6]. The required accuracy of electron-ion trajectory alignment was successfully achieved at the start of the 2020 RHIC run. This was an important step, which together with other LEReC optimizations allowed us to obtain operational cooling and a substantial increase in RHIC luminosity.

The main purpose of this work is two-fold.

First, it works as an independent way to optimize the cooling. There is always the possibility that due to some errors the BPM offsets are not corrected to the required level. Thus, it is worthwhile to check the electron-ion alignment with an independent method. This work provides a way to perform such a check by using the Bayesian approach. It determines an optimum trajectory which can be used as a reference for the control system's orbit correction routine. The results it generated can also be used to verify the validity of the traditional tuning routine.

Second, it demonstrates that machine learning (ML) methods can be applied in the LEReC system to perform traditional control tasks effectively, which opens up the possibility of trying different ML methods in the system.

As far as we know, this is the first work of applying Bayesian optimization on trajectory alignment in an electron cooling system.

In the following, Sec. II introduces the basics of the Bayesian optimization (BO) method and the algorithm parameters we used in this work, and summarizes related work of using BO in the accelerator field. Section III presents simulation results [7] conducted in a LEReC system simulator [8]. The purpose of the simulation is to prepare for the experiments on the live system. The experimental results are shown in Sec. IV, which demonstrates that optimum cooling can be achieved and maintained by using the BO approach. Finally, Sec. VI concludes this work.

II. BAYESIAN OPTIMIZATION

Bayesian optimization (BO) attempts to optimize an objective function f with as few samples as possible. It is

particularly useful when the explicit expression of f is unknown and evaluation of f is expensive, such as tuning the hyperparameters of a deep neural network.

A general procedure of BO is depicted in Fig. 2. As we can see, instead of sampling the original expensive objective f , BO builds a surrogate model of f and uses an acquisition function to guide the sampling procedure.

A Gaussian process (GP) is a very popular surrogate model used in BO [9]. A GP is a distribution over functions $f(x) \sim \mathcal{GP}(\mu(x), k(x, x'))$, which can be constructed on a set of N observed data points $\mathcal{D}_N = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$, where $y_n, n = 1, 2, \dots, N$ is usually a noisy response $f(x_n) + \epsilon$ with some normally distributed noise ϵ . A sample from a GP at a point x returns the predicted mean $\mu(x)$ and variance $\sigma(x)$ of a normal distribution over the possible function values at x .

For convenience, the mean function of a GP is usually assumed to be the zero function $\mu(x) = 0$. The covariance (kernel) function $k(x, x')$ describes how the objective varies (e.g., smoothness) in terms of the changes in the input space [10]. Common choices of kernels are squared exponential kernel, rational quadratic kernel, periodic kernel, etc. New kernels can also be created by combining other kernels via addition and multiplication. In this work, we use the Matérn kernel [Eq. (1)] with $\nu = 3/2$, which can be seen as a generalization of the Gaussian radial basis function [10].

$$C_\nu(d) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} \frac{d}{\rho} \right)^\nu K_\nu \left(\sqrt{2\nu} \frac{d}{\rho} \right) \quad (1)$$

where d is the distance between two points, Γ is the gamma function, K_ν is the modified Bessel function of the second kind. ρ and ν are positive parameters of the Matérn covariance. ρ is the length scale of the kernel, and is usually default to 1 at the beginning of the training process. ν controls the

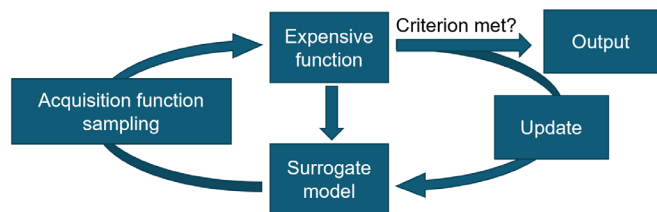


FIG. 2. Bayesian optimization process.

smoothness of the learned function. The smaller ν , the less smooth the approximated function is. Here, we use $\nu = 3/2$.

Another important component of BO is the acquisition function. The main role of an acquisition function is to guide the sampling process. In general, the acquisition function is designed so that high acquisition values correspond to potentially high objective values. Hence, the next sample point is selected by maximizing the acquisition function $\mathcal{A}(\cdot)$. That is, the process will next sample f at point $\operatorname{argmax}_x \mathcal{A}(x|\mathcal{D}_N)$. In this work, we optimize the acquisition function using DIRECT [11], a deterministic, derivative-free optimizer.

Some commonly used acquisition functions [9] are expected improvement (EI), upper confidence bound (UCB), and GP-UCB.

EI calculates the objective's expected improvement on a point x over the current best observation $f(x^+)$:

$$\begin{aligned} EI(x) &= \mathbb{E}(\max\{0, f(x) - f(x^+)\} | \mathcal{D}_N) \\ &= \begin{cases} (\mu(x) - f(x^+))\Phi(Z) + \sigma(x)\phi(Z) & \text{if } \sigma(x) > 0 \\ 0 & \text{if } \sigma(x) = 0 \end{cases} \\ Z &= \frac{\mu(x) - f(x^+)}{\sigma(x)} \end{aligned}$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ denote the cumulative density function (CDF) and probability density function (PDF) of the standard Gaussian distribution respectively.

UCB calculates a linear combination of mean and variance by a weight factor κ :

$$UCB(x) = \mu(x) + \kappa\sigma(x) \quad (2)$$

The parameter κ balances the trade off between exploitation (mean) and exploration (variance), with larger κ values favor exploration and vice versa.

Alternatively, work [12] proposed a no-regret scheme called GP-UCB:

$$GP - UCB(x) = \mu(x) + \sqrt{\beta_t}\sigma(x) \quad (3)$$

where $\beta_t = 2 \log(|\mathcal{D}|t^2\pi^2/6\delta)$, $|\mathcal{D}|$ is the dimension of the observation dataset, t is the algorithm's current running round, and $(1 - \delta)$ is the no-regret probability.

In this work, we use UCB as the acquisition function. The complete Bayesian optimization routine is summarized in Algorithm 1.

Algorithm 1. Bayesian Optimization.

Require: Objective function f , observation dataset \mathcal{D}_N , GP prior $M = \mathcal{GP}(\mu(x), k(x, x'))$, acquisition function $\mathcal{A}(\cdot)$.

- 1: **for** $t = 1, 2, \dots$ **do**
- 2: Decide a new sample point $x_{\text{new}} = \operatorname{argmax}_x \mathcal{A}(x|\mathcal{D}_N)$.
- 3: Query the objective $y_{\text{new}} = f(x_{\text{new}}) + \epsilon$.
- 4: Update \mathcal{D}_N and the GP model.
- 5: **end for**

Bayesian optimization has been applied in a broad variety of fields, such as engineering design, environment science, robotics and machine learning, financial market, etc. Some BO applications are summarized in [13].

As in the accelerator controls field, work [14,15] showed simulation results of using Bayesian optimization to tune quadrupole magnets settings in the LCLS Free-Electron Laser (FEL) at SLAC. Comparisons have been made with hand-tuning and with the existing Nelder-Mead optimization method. Moreover, work [15] showed that by leveraging knowledge of accelerator physics and building correlated kernels, the GP converges faster in simulation scenarios where the input dimensions are correlated.

In work [16], a variant of Bayesian optimization called SafeLineBO was proposed, which divides global problem into sequential subproblems that can be solved efficiently without violating safety constraints. Then the algorithm was compared with the simple parameter scanning and Nelder-Mead method to tune the FEL outputs of the SwissFEL with up to 40 parameters.

Work [17] proposed an alternative method to the classical Bayesian optimization that was trained on historical data. The new method incorporated a physics model of the system, and built a Gaussian process (GP) which uses a physics model-informed kernel. Then a simulation experiment was conducted of optimizing the electron beam loss rate on the SPEAR3 storage ring, and the results showed that the GP with a physics model-informed kernel converges faster than both the GP with a data-informed kernel and the Nelder-Mead simplex optimizer.

The technique used in this work has no fundamental differences in the assumptions with the ones mentioned above, but it is the first work of applying Bayesian optimization on beam trajectory alignment in an electron cooling system. It has practical value in the context of LEReC and the future Electron-Ion Collider (EIC) project.

III. SIMULATION RESULTS

In this section, we list some major simulation results acquired by using a LEReC system simulator [8]. A more comprehensive summary of the simulation results is presented in [7].

The main goal of the simulation is to validate the Bayesian technique's performance and prepare for the experiments in the actual LEReC system.

The simulation considers a realistic scenario in which the electrons travel in the same way as in the real system, as shown in Fig. 3. There are 8 beam position monitors (BPMs) along each of the "yellow" and "blue" cooling section. Each BPM reports 2D data points— x and y coordinates of the electron beam. Since the x and y planes are symmetric, in the simulation only the y plane is considered and has a range of $[-3, 3]$ mm. The ion beam is always in the center position of ($x = 0, y = 0$).

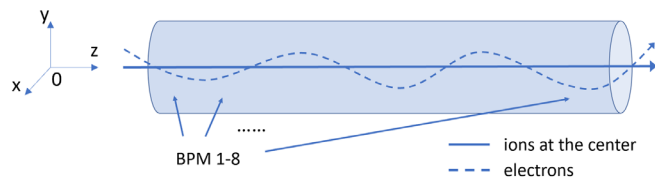


FIG. 3. The simulation considers a real-world scenario where electrons travel through the cooling sections and are monitored by 8 BPMs.

To avoid interrupting the normal operations of the live system, a LEReC simulator is used. It takes electron positions (BPMs readings) as the inputs and outputs the transverse cooling rate. The transverse cooling rate is defined as the decreasing speed of the transverse ion beam size. A more negative cooling rate means the beam size decreases faster, hence a better cooling result.

Due to the high computational complexity of the simulator, in the simulation we examine two features of the electron positions instead of using the 8 BPMs readings directly. Those two features are the root mean square (rms) and the standard deviation (std). The rms value ($\sqrt{\sum \mathbf{x}^2/n}$) measures on average how far away the electron trajectory is from the ions. The std value ($\sqrt{\sum (\mathbf{x} - \mu(\mathbf{x}))^2/n}$) measures how much the electron trajectory varies while cotraveling with the ions. Those two features are chosen because they can represent the BPMs' information well and are easy for the algorithm to learn.

The simulation results are summarized in Figs. 4 and 5. The Bayesian model was trained on 60 random samples (blue) and then used to make 15 more samples (red). Figure 4 shows the rms value distribution (top) and the std value distribution (bottom) of the BPM readings. As we can see, both the rms and std values of the Bayesian samples are closer to 0 compared with the random samples. It indicates that the algorithm learns a way to optimize the cooling rate, which is decreasing the rms and std values of the electron

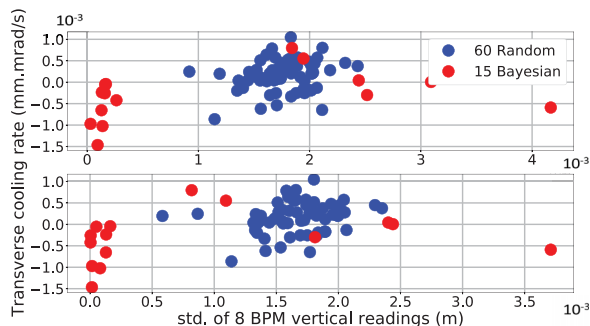


FIG. 4. Comparisons of rms and std values of BPM readings from the random samples (blue) and Bayesian samples (red). The Bayesian samples clearly have both rms and std values closer to 0 compared with the random samples. It indicates that the algorithm learns a way to optimize the cooling rate by reducing both rms and std values of electron trajectories, which matches our expectations.

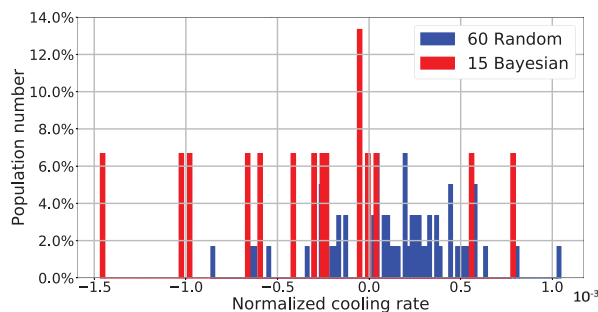


FIG. 5. The statistical distribution shows that the Bayesian samples (red) have a larger percentage of populations that possess a higher cooling rate compared with the random samples (blue).

positions. In other words, an electron trajectory that is closer to the center and has less variation tends to produce a better cooling rate, which matches our expectations. Figure 5 shows the statistical distribution of the cooling rates from all the samples. It quantitatively verifies that the Bayesian samples have more populations (higher percentage) that possess a faster cooling rate compared with the random samples.

IV. EXPERIMENTAL RESULTS

In this section, we demonstrate experimentally the effectiveness of the Bayesian method on optimizing the cooling rate in the live LEReC system.

A. Preliminaries

The LEReC system contains “yellow” and “blue” cooling sections, and along each of them there are 8 beam position monitors (BPMs). An orbit correction program is used to regulate the electrons' trajectories based on those BPMs readings. The program can continuously tune corrector magnets to keep electrons at some specific positions throughout the cooling sections.

The beam-based calibration and alignment of the CS BPMs is performed to zero the BPMs offsets. The ion trajectory is kept at the center of the cooling section. Hence, the present target BPM positions for electrons are ($x = 0, y = 0$). That should produce the best cooling rate, provided that the electron bunches' angular spread and momentum spread are optimized and that the relativistic γ -factors of both beams are well matched. However, the possibility of errors in BPM offset measurements always remains. Therefore, Bayesian optimization (BO) is used as an independent approach to discover an electron trajectory that optimizes the cooling rate, which also in turn validates the correctness of the beam-based alignment method.

Bayesian optimization as introduced in Sec. II takes the BPM readings as inputs and the cooling rates as outputs. It establishes a relationship between BPMs and cooling rates directly. Hence, the BPMs measurement errors are taken into account inherently. Moreover, BO can handle the

Gaussian random noise which exists in the system. Thus, the output trajectories from the BO should reflect the optimum BPM positions of the electrons.

The transverse cooling rate λ is defined as the decreasing speed of the transverse ion beam size δ , which is calculated as $\lambda = (1/\delta)(d\delta/dt)$. Therefore, a more negative λ indicates a faster cooling rate, and vice versa.

The goal of the experiment is to maximize $-\lambda$ by tuning the target BPM values using the orbit correction program. Due to the limited machine time, the experiment only considers the first 4 BPMs. In our studies the beam is moved in a range of $[-3, 3]$ mm in each of the BPMs. We take 40 initial samples to train the algorithm. At each step of the sampling process, the algorithm sets an anchor point and randomly uniformly samples around that point. Then, it modifies the anchor point by a step size. The anchor point is kept within the input range throughout the process.

The advantage of this sampling routine compared to the plain random BPM sampling is that it explores the entire input domain in an organized manner, which ensures that the training set contains various enough different input values and at the same time prevents dramatic changes in the machine settings. It also maintains certain randomness in the sampling process at each step. The detailed procedure is described in Algorithm 2. In the experiment, we take $r = 0.3$, $S = 0.6$, $t_s = 120$ s.

The initial samples and their corresponding objective values are shown in Fig. 6. We also tried sampling each input dimension independently, but it turns out this simultaneously sampling pattern gives the best optimization results.

From the top plot of Fig. 6 we can see that the inputs have iterated through the entire input domain, which is crucial for the fast convergence of the BO method. The objectives' pattern can also be revealed more evidently

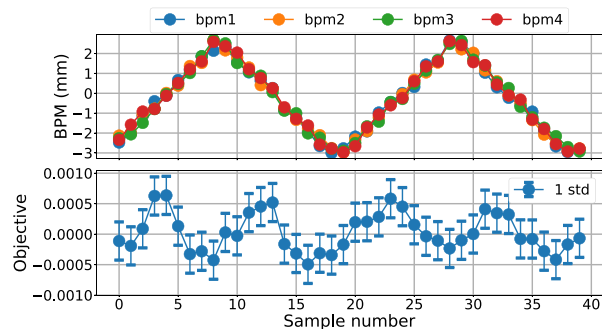


FIG. 6. 40 training points are sampled using Algorithm 2. The inputs step through the entire range (top) which helps to reveal a clear pattern of the outputs (bottom). As we can see, input positions around 0 potentially have higher objectives.

when the inputs are sampled systematically as shown in the bottom plot. We can see that input positions around 0 potentially generate higher cooling rates than others, which further validates the simulation results in Sec. III.

In the LEReC system, there are many factors that can affect the ion beam size (e.g., ion intensity, device measurement errors, etc.), which makes the beam size data noisy.¹ An example period of data during the experiment is shown² in Fig. 7. Due to the noise presented in δ , the original formula for cooling rate $\lambda = (1/\delta)(d\delta/dt)$ generates many misleading variations (large error bars in the bottom plot of Fig. 6) in the objective values. Those variations diminish the algorithm's performance significantly, and eventually stop the algorithm from converging. This issue will be demonstrated and addressed in the next section.

B. Objective function sensitivity

After training, the BO method is used to control the BPM positions. As noted above, the use of point δ values (beam sizes) in $\lambda = (1/\delta)(d\delta/dt)$ makes the objective function very unstable. Figure 8 illustrates the BO samples' trajectory, we can see that the objective (bottom plot) decreases too fast even when the algorithm converges to an optimum strategy (top plot, e.g., sample 7 to 11 and 13 to 17). Such instability confuses the algorithm and eventually makes it diverge as the algorithm does not have enough time to learn properly the correlation between the inputs and outputs.

Therefore, we modified the equation so that instead of using single beam size values it uses average beam sizes in an interval to calculate the cooling rate $\lambda = (1/\text{avg}(\delta))(\text{avg}(d\delta)/dt)$. It stabilizes the objective behaviors, thus giving the algorithm more time to learn. The length of the interval is defined as the “radius r ” of the

Algorithm 2. Initial Sampling Routine.

Require: Observation dataset $\mathcal{D}_{\mathcal{N}}$, random sampling function f_R , random sampling radius r , step size S , statistic period t_s .

- 1: Set $\mathcal{D}_{\mathcal{N}} = \emptyset$.
 - 2: Set an anchor point $x_{\text{ach}} = -3$.
 - 3: Set the initial operation $s_{\text{op}} = '+'$.
 - 4: **for** $t = 1, 2, \dots, 40$ **do**
 - 5: **if** x_{ach} is outside of the range $[-3, 3]$ **then**
 - 6: Reverse x_{ach} to the previous value, flip $s_{\text{op}} = -s_{\text{op}}$.
 - 7: **end if**
 - 8: Randomly uniformly sample around the anchor point, $x_{\text{new}} = f_R(x_{\text{ach}}, r)$.
 - 9: Set the BPMs at x_{new} .
 - 10: Collect transverse beam size data during t_s , and calculate the cooling rate as $y_{\text{new}} = (1/\delta)(d\delta/dt)$.
 - 11: Add $(x_{\text{new}}, y_{\text{new}})$ to the dataset $\mathcal{D}_{\mathcal{N}}$.
 - 12: According to the s_{op} , modify the anchor point by a step size $x_{\text{ach}} = x_{\text{ach}} + / - S$.
 - 13: **end for**
-

¹The beam size is measured by H-jet [18] and plotted using the rms value.

²By using the system tool “LogView,” which is designed for displaying logged data.

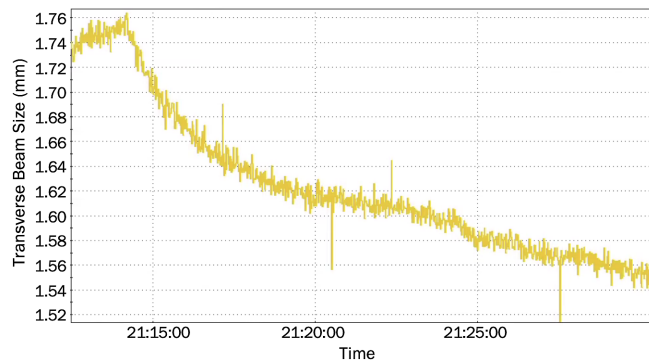


FIG. 7. An example period of transverse ion beam size data are fetched from the real system during the experiment. It shows the data are noisy.

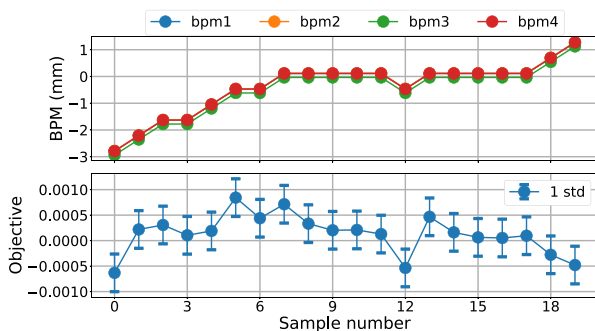


FIG. 8. Demonstration of the volatility of the original objective function. Due to the use of point δ values in $\lambda = (1/\delta)(d\delta/dt)$ and the noise in δ , the original objective varies too fast (bottom) even when the algorithm converges to an optimum strategy (top, e.g., from step 7 to 11 or 13 to 17). This eventually makes the algorithm diverge (after step 17) as the algorithm does not have enough time to learn properly the correlation between the inputs and outputs.

algorithm. By tuning r the objective function’s sensitivity can be controlled, which further affects how the algorithm behaves.

Figure 9 shows the algorithm’s behaviors under different r values. Generally, the larger the radius grows the less sensitivity the objective has. The right part of Fig. 9 shows an extreme case when $r = 30$, the radius is so large that the objective function becomes insensitive even to the inputs. The algorithm explores to the upper bound of the input domain to infer the model, but the output is unable to respond accordingly. However, a large radius can reduce the uncertainty greatly, as shown by the error bar plots. Thus, the radius should be set properly such that the objective function is not too sensitive to the noise but also can respond to the input changes reasonably well.

C. Results

After trying different values, we take $r = 15$ and the complete results are presented in Fig. 10. The plots show

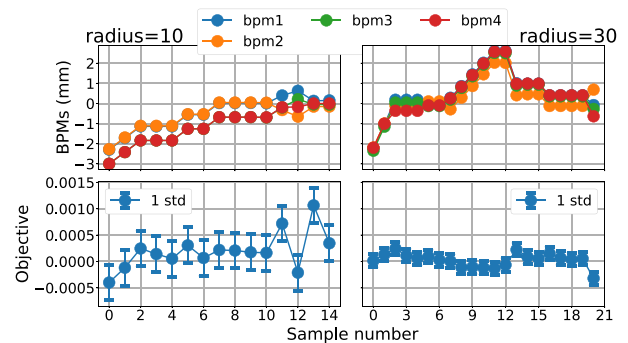


FIG. 9. Effects of different radius values on the algorithm’s behaviors. For a small radius $r = 10$ (left column), the objective function is still sensitive to changes (e.g., sample 10 to 14). But compared with the original objective (see Fig. 8), the process has become more stable (e.g., sample 7 to 10). For a large radius $r = 30$ (right column), the objective function becomes too insensitive. It is unable to change properly according to the inputs. As the inputs step through the entire range (top right), the objective values do not change with a proper scale (bottom right). Both cases should be avoided as they do not provide correct system information for the algorithm to learn.

that the BO method can converge to the optimum solution very quickly. One thing worth mentioning is that by using the radius parameter the Bayesian samples do not generate as high objective values as the initial 40 training samples, but it makes the process more stable. As shown in the bottom plot, the errors from the Bayesian samples are much smaller compared with the ones from the training samples. The purpose of the objective values is to train the algorithm correctly so it can effectively learn the behaviors of the system. The use of the radius parameter adds extra flexibility to the algorithm and brings more practical benefits.

The corresponding electron trajectories (displayed by the application “LogView”) in the live system are shown in

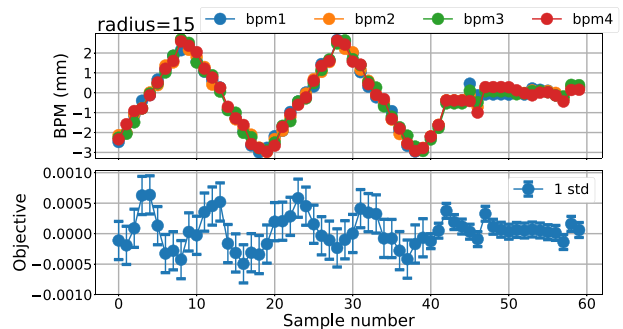


FIG. 10. The final experiment is performed by using a radius of 15. The outputs is not too sensitive to the noise but also sensitive enough to respond to the input changes properly. The first 40 points are the training samples. After training, the algorithm converges very quickly (already reaches a close neighborhood after the first 3 steps, top plot) to an optimum solution, which corresponds to the center position of $(x = 0, y = 0)$.

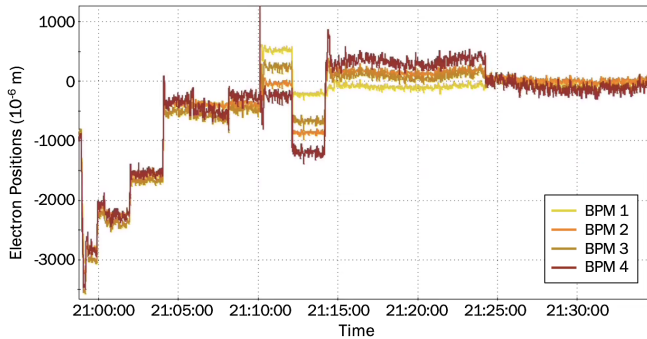


FIG. 11. After training, the algorithm is used to control the electron beam trajectories. As we can see, the algorithm tunes the electrons back to the center and maintains it there, which in turns verifies the correctness of the beam-based alignment method.

Fig. 11. It demonstrates that the BO method is capable of tuning the electrons from the farthest point (-3 mm) to the optimum position efficiently and can maintain that trajectory, which shows the effectiveness of the algorithm. Moreover, the trajectory returned by the BO method indicates that an electron trajectory around the center position ($x = 0, y = 0$) maximizes the cooling rate, which verifies the correctness of the traditional orbit correction program and also confirms that the BPMs in the system are calibrated reasonably well.

V. FUTURE WORK

There are several variants of Bayesian optimization that could bring extra performance benefits. Due to the limited machine time, we leave them to future work.

A. Physics model-informed BO

In the above experiment, we keep updating the model with the new data available, therefore the method is data-informed (DI).

In contrast to the DI method, work [17] introduces a physics model-informed (PMI) method. It gets the name from the way it constructs the GP kernel. In a DI method, the GP kernel is updated step by step. Whereas in a PMI method, the GP kernel is calculated directly by using physics data model. Specifically, the method calculates the kernel's precision matrix by expanding a Hessian matrix around the optimum point obtained from historical data. Then, the kernel is used directly in the optimization without any further updating. The Hessian matrix contains important information about the local curvature of the objective function around its optimum and can guide the BO to navigate through the input space to locate the true optimum. Hence, the physics model-informed method could potentially be more efficient.

To demonstrate the difference, a simulation is conducted where we have a Gaussian-like maximization objective with 4-dimensional inputs. Each input has a

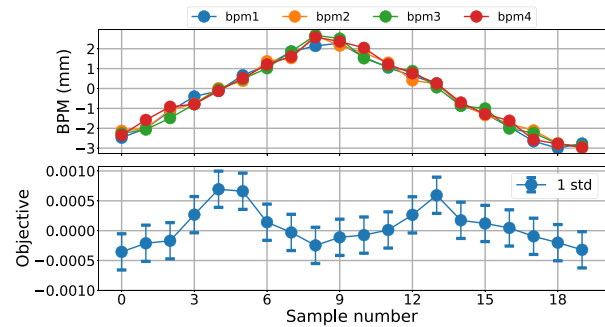


FIG. 12. 20 training points are sampled using Algorithm 2. Both the data-informed (DI) method and the physics model-informed (PMI) method use the same training data.

range of $[-3, 3]$. Figure 12 shows 20 initial training samples that are taken using Algorithm 2 within the input range.

Figure 13 shows the comparison results of the performance between the data-informed (left) and the physics model-informed BO (right). From the plots we can see that the physics model-informed BO produces a more steady optimum solution and a higher average objective value.

B. Contextual Gaussian process

Both the data-informed and physics model-informed methods can only work well in a static environment. However, a real-world scenario often involves some dynamic factors which can affect the optimum. For example [19], suppose we want to monitor the highest temperature in a building by a collection of sensors across the building. Due to battery limitations of the sensors, each time we can only select a small part of the sensors to activate and do the monitoring job. Due to the fact that the sun is moving relative to the building, the hottest area in the building shifts depending on the time of the day. The goal is

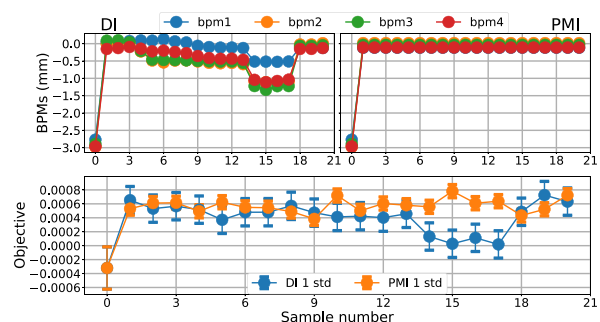


FIG. 13. Comparison of results from the optimization performed by the data-informed (DI, top left) and by the physics model-informed BO (PMI, top right). Both methods converge to the optimum solution very quickly. Moreover, the PMI method produces a more steady solution (top row), and hence a higher level of objective values (bottom row).

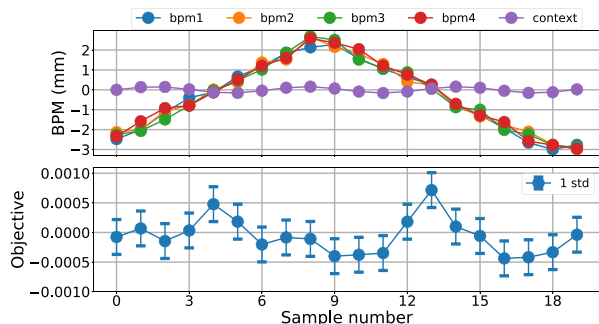


FIG. 14. 20 training samples with a sinusoidal context (purple line) added. The objective value is now dependent on both the inputs and the context. Compared with Fig. 12, the objective's pattern is more volatile.

to learn which sensors to activate at what time of the day. The time of the day is the dynamic factor (context) in this task that will affect the optimum solution. Traditional Bayesian optimization methods such as DI or PMI cannot handle such dynamic contexts, therefore cannot generate the optimum solution.

In work [19], a method called the contextual Gaussian process (CGP) was proposed to handle the environmental factors by using separate kernels to model the inputs and contexts. An overall kernel for the entire process is then constructed by either using addition or multiplication of those elemental kernels depending on the relationship between the inputs and the contexts.

With the same simulation settings in Sec. VA, a sinusoidal signal is added to the Gaussian objective as the changing context. This is a simplified dynamic scenario where the context only affects the objective value but not the optimum solution (still the center position). Figure 14 shows the initial training samples with a context (purple line) added. Note that now the objective is dependent on both the inputs and the context.

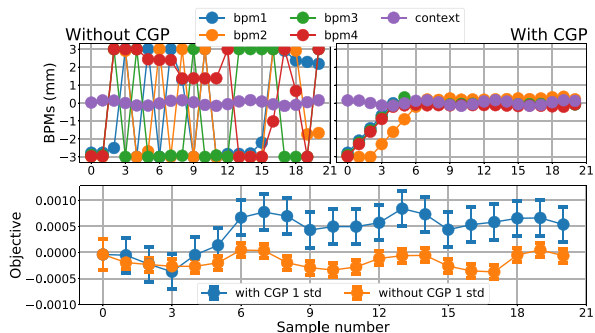


FIG. 15. Physics model-informed BO is used to optimize the contextual objective function. Without the CGP feature (top left), the BO is unable to converge as it cannot correctly model the relationship between the inputs and outputs. Whereas with the CGP modeling the context (top right), the BO successfully handles the dynamic environmental factors and is able to find the optimum solution.

Physics model-informed BO is used to optimize this contextual objective function. The results are shown in Fig. 15. We can see that without the CGP modeling the context (top left), the algorithm fails to converge as the objective now varies only partially related to the input changes. Whereas, adding the CGP (top right) enables the algorithm to precisely learn the correlations between the inputs and the objective and successfully converge to the optimum center position.

VI. CONCLUSIONS

In this work, we applied Bayesian optimization (BO) to maximize the cooling rate in the LEReC system. Both the simulation and experimental results show that the BO method is very effective in producing and maintaining good electron positions to optimize the cooling rate. It also verifies the correctness of the traditional orbit correction program and the BPM calibrations. It opens up many possibilities of trying different machine learning methods on optimizing performance for control tasks in the RHIC complex, as well as the future Electron-Ion Collider (EIC).

ACKNOWLEDGMENTS

We would like to thank Adi Hanuka who gave a comprehensive introduction to the physics model-informed Bayesian optimization. We would like to thank Ryan Roussel who provided information about the kernel functions used in the BO. We would like to thank Andrew Sampson who contributed to the preliminary testing of contextual Gaussian Process code. We would also like to thank all the department personnel in the Main Control Room (MCR) who coordinated the machine time for our experiments. One of the authors was supported in part by the U.S. National Science Foundation under Grant No. PHY-1549132, the Center for Bright Beams. This work was supported by Brookhaven Science Associates, LLC under Contract No. DE-SC0012704 with the U.S. Department of Energy.

- [1] K. M. Walsh, Electron bunches keep ions cool at rhic, <https://www.bnl.gov/newsroom/news.php?a=215585>, 2019.
- [2] A. V. Fedotov, Z. Altinbas, S. Belomestnykh *et al.*, Experimental Demonstration of Hadron Beam Cooling Using Radio-Frequency Accelerated Electron Bunches, *Phys. Rev. Lett.* **124**, 084801 (2020).
- [3] S. Seletskiy, M. Blaskiewicz, A. Drees *et al.*, Accurate setting of electron energy for demonstration of first hadron beam cooling with rf-accelerated electron bunches, *Phys. Rev. Accel. Beams* **22**, 111004 (2019).
- [4] G. I. Budker, An effective method of damping particle oscillations in proton and anti-proton storage rings, *Sov. At. Energy* **22**, 438 (1967).
- [5] Y. S. Derbenev, Theory of electron cooling, [arXiv:1703.09735](https://arxiv.org/abs/1703.09735).

- [6] S. Seletskiy, M. Blaskiewicz, A. Drees *et al.*, Obtaining transverse cooling with nonmagnetized electron beam, *Phys. Rev. Accel. Beams* **23**, 110101 (2020).
- [7] Y. Gao, K. A. Brown, P. Dyer *et al.*, Applying machine learning to optimization of cooling rate at low energy RHIC electron cooler, in *Proceedings of the 12th International Particle Accelerator Conference (IPAC)* (JACoW, Campinas, Brazil, 2021).
- [8] H. Zhao, M. Blaskiewicz, A. V. Fedotov *et al.*, Cooling simulation and experimental benchmarking for an rf-based electron cooler, *Phys. Rev. Accel. Beams* **23**, 074201 (2020).
- [9] E. Brochu, V. M. Cora, and N. de Freitas, A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning, [arXiv:1012.2599](https://arxiv.org/abs/1012.2599).
- [10] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning* (The MIT Press, Cambridge, MA, 2006).
- [11] D. R. Jones, C. D. Perttunen, and B. E. Stuckman, Lipschitzian optimization without the lipschitz constant, *J. Optimiz. Theory Appl.* **79**, 157 (1993).
- [12] N. Srinivas, A. Krause, S. Kakade *et al.*, Gaussian process optimization in the bandit setting: No regret and experimental design, in *Proceedings of the 27th International Conference on Machine Learning (ICML)* (Haifa, Israel, 2010).
- [13] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, Taking the human out of the loop: A review of bayesian optimization, *Proc. IEEE* **104**, 148 (2016).
- [14] M. McIntire, T. Cope, S. Ermon *et al.*, Bayesian Optimization of FEL Performance at LCLS, in *Proceedings of the 7th International Particle Accelerator Conference (IPAC), Busan, Korea* (JACoW, Geneva, Switzerland, 2016), p. WEPOW055.
- [15] J. Duris, D. Kennedy, A. Hanuka *et al.*, Bayesian Optimization of a Free-Electron Laser, *Phys. Rev. Lett.* **124**, 124801 (2020).
- [16] J. Kirschner, A. Adelman, N. Hiller *et al.*, Bayesian Optimisation for Fast and Safe Parameter Tuning of SwissFEL, in *Proceedings of the 39th International Free Electron Laser Conference (FEL)* (JACoW, Hamburg, Germany, 2019).
- [17] A. Hanuka, X. Huang, J. Shtalenkova, D. Kennedy, A. Edelen, Z. Zhang, V. R. Lalchand, D. Ratner, and J. Duris, Physics model-informed gaussian process for online optimization of particle accelerators, *Phys. Rev. Accel. Beams* **24**, 072802 (2021).
- [18] A. Zelenski, A. Bravar, D. Graham *et al.*, Absolute polarized H-jet polarimeter development, for RHIC, *Nucl. Instrum. Methods Phys. Res., Sect. A* **536**, 248 (2005).
- [19] A. Krause and C. Ong, Contextual gaussian process bandit optimization, in *Advances in Neural Information Processing Systems (NIPS)*, edited by J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger (Curran Associates, Inc., Granada, Spain, 2011), Vol. 24.