ISBN: 978-3-95450-261-5 ISSN: 2673-7000 doi: 10.18429/JACoW-NAPAC2025-MOP103

ONLINE OPTIMIZATIONS OF NSLS-II LINAC AND LINAC-TO-BOOSTER BEAM LINES USING MACHINE LEARNING METHODS

Minghao Song*, Guimei Wang, Yoshiteru Hidaka, Xi Yang
Brookhaven National Laboratory, Upton, NY, USA

Abstract

The NSLS-II is a cutting-edge 3 GeV storage ring light source in the world. The electron beam is initially accelerated in a linear accelerator to an energy of 170 MeV and subsequently accelerated in a booster synchrotron to a beam energy of 3 GeV. Therefore, the performance of the Linac and the Linac-to-Booster beam lines is imperative for good beam injection to the booster. Online optimization is an effective solution to improve accelerator performance when there is degradation. This paper presents the results of online optimization employing a machine learning method.

INTRODUCTION

Linear accelerators (Linacs), Linac-to-booster transport lines, upstream boosters, and storage rings are essential components in modern synchrotron radiation facilities. Achieving optimal injection performance from the Linac to the booster requires precise control of many machine parameters, including buncher phase, RF phases and amplitudes, and corrector magnet strengths. However, because of the lack of accurate physics models for the Linac and Linac-to-booster beamlines, along with the effects of drifts and environmental disturbances, it is often difficult for simulations to accurately reflect the real behavior of the machine. Therefore, a practical method for improving the performance of the actual machine is needed.

Online optimization has been increasingly used in the accelerator community to improve machine performance. These optimization algorithms adjust machine parameters in real time based on direct measurements from the accelerator, allowing performance improvements to be made directly and efficiently.

Traditional algorithms, such as robust simplex [1], robust conjugate direction search (RCDS) [2], non-dominated sorting genetic algorithm II (NSGA-II) [3], and particle swarm optimization (PSO) [4], have been widely used for online accelerator tuning [5, 6]. However, due to their stochastic nature, these algorithms are often inefficient, requiring a large number of objective function evaluations to reach the global optimum. This inefficiency arises because the trial solutions generated by these methods do not fully utilize the information contained in the previously sampled data from the parameter space.

In this study, we used the multi-generation Gaussian process optimizer (MG-GPO) [7] for online optimization of problems with complex parameter spaces. MG-GPO was originally developed for design optimization, which uses posterior Gaussian process regression models [8] to filter for good trial solutions in an iterative process similar to NSGA-II [3] or PSO [4]. It has been successfully applied to storage ring nonlinear lattice design optimization [9] and online accelerator optimization [10]. This method can significantly reduce the tuning time and improve operational reliability.

* msong1@bnl.gov

OPTIMIZATION METHODS

Particle Swarm Optimization

In particle swarm optimization, each solution is represented by a particle's position, $\mathbf{x}_i(t)$, which is updated by adding a velocity increment, $\mathbf{v}_i(t+1)$. The iterative update process is described in more detail by the following equations [4]:

$$\begin{aligned} \mathbf{x}_i(t+1) &= \mathbf{x}_i(t) + \mathbf{v}_i(t+1), & (1)\\ \mathbf{v}_i(t+1) &= \omega\, \mathbf{v}_i(t) + c_1 r_1 \left(\mathbf{pbest}_i(t) - \mathbf{x}_i(t)\right) \\ &\quad + c_2 r_2 \left(\mathbf{gbest}(t) - \mathbf{x}_i(t)\right), & (2) \end{aligned}$$

where $\mathbf{x}_i(t)$ and $\mathbf{x}_i(t+1)$ represent the positions of ith particle at generation t and $t+1$, respectively. The parameter $\mathbf{v}_i(t+1)$ is the velocity-based correction that determines the change in position between two successive generations. According to Eq. (2), the velocity update is influenced by three terms. The first term, known as the inertia influence, is scaled by the inertia weight ω, which determines the contribution rate from the previous velocity. The second term, referred to as the cognitive influence, depends on the distance between a particle's personal best position **"pbest"** and its current position. The third term, called the social influence, reflects the distance between the global best position **"gbest"** found by the swarm and the current position. The coefficients c_1 and c_2 are called the cognitive and the social learning factors, respectively. In addition to the hyper-parameters ω, c_1 and c_2, the random variables r_1 and r_2 also have an impact on the optimization results.

Multi-Generation Gaussian Process Optimizer

The multi-generation Gaussian process optimizer (MG-GPO) updates a population with a fixed number of solutions from generation to generation. Compared to traditional stochastic optimization algorithms such as NSGA-II [3] and PSO [4], MG-GPO enables the pre-selection of candidates from a large set of trial solutions using Gaussian process (GP) models, which is less expensive for evaluations and could guarantee the offspring solutions outperform the previous ones [7]. As a result, MG-GPO can be effectively used for online optimization problems where beam study time is limited and the objective evaluation can be time consuming.

ISBN: 978-3-95450-261-5 ISSN: 2673-7000 doi: 10.18429/JACoW-NAPAC2025-MOP103

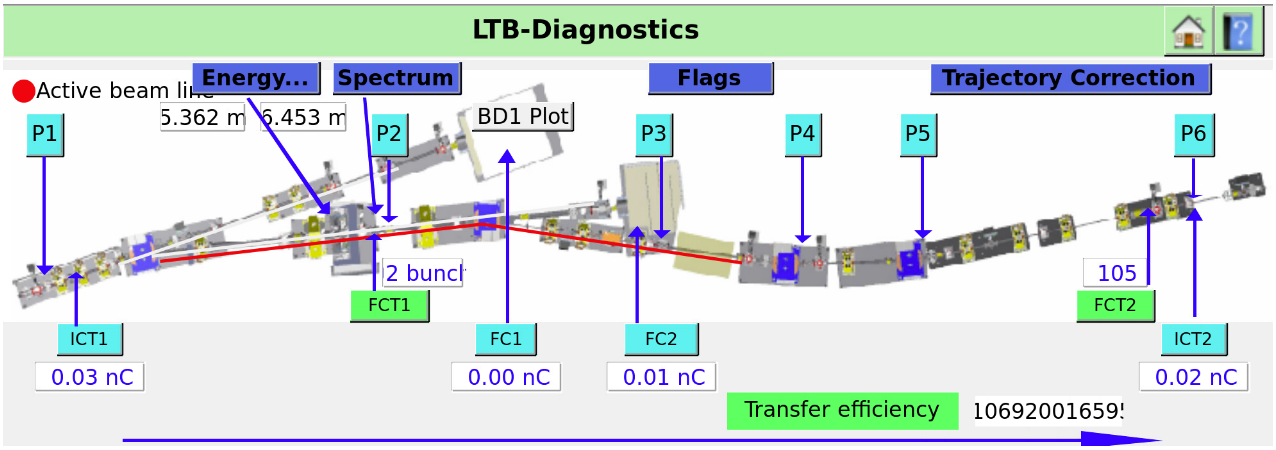

Figure 1: Layout of Linac and Linac-to-booster (LTB) beam lines.

The MG-GPO algorithm can be stated in detail as follows [7, 9, 10]:

(1) Randomly initialize a population of solutions with a fixed population size. These initial solutions are then evaluated using the actual system;

(2) Construct a prior Gaussian process model using the initially evaluated solutions;

(3) Generate a large number of trial solutions using operations such as polynomial mutation [11], simulated binary crossover (SBX) [12], and swarming [4]. It is worth noting that any combination of these methods, individually or together, can be used to produce new solutions;

(4) Make predictions for the trial solutions using the constructed Gaussian process (GP) model, which provides both the expected mean and standard variation for each solution. To effectively guide the search for promising candidates with large gains in the complex parameter space, an acquisition function is used as the figure of merit. In this study, the widely used GP-UCB (Gaussian process-upper confidence bound) [13] is selected, which is defined as: $\text{GP-UCB}(\mathbf{x}) = \mu(\mathbf{x}) - \kappa\sigma(\mathbf{x})$, where $\mu(\mathbf{x})$ and $\sigma(\mathbf{x})$ are the predicted mean and variance, respectively. The trade-off between exploration and exploitation is controlled by the balance parameter κ. In general, A smaller κ favors exploitation by focusing on areas with high predicted performance, but may not fully explore the whole parameter space. In contrast, a larger κ promotes exploration of uncertain regions, but may not make full use of the model. Therefore, it is crucial to select a proper κ. It can either be set as a fixed constant or dynamically adjusted during the optimization process;

(5) Perform another non-dominated sorting to select the best solutions based on the gain measured by the acquisition function;

(6) Evaluate the selected solutions by applying them to the actual system;

(7) Update the personal best solutions by comparing them with the newly evaluated solutions. Then, perform non-dominated sorting on the combined set of evaluated solutions and the global best solutions from the previous generation to update the global best solutions.

(8) Construct the posterior Gaussian process model using the evaluated solutions along with the global best solutions. It is worth noting that the evaluated solutions may come only from the current generation or may include a combination of solutions from previous generations.

(9) Repeat steps (3) through (8) until a stopping criterion is met. In this study, the maximum number of generations is used as the stopping condition for the optimization algorithm.

OPTIMIZATION RESULTS

Figure 1 shows the layout of the Linac and Linac-to-booster (LTB) beamlines. The optimization of injection efficiency from the Linac to the booster is carried out in two stages: first, optimizing the Linac to maximize injection efficiency from the Linac to the LTB beam line; and second, optimizing the LTB to maximize injection efficiency from the LTB to the booster.

During the experimental study, the Linac and LTB beamlines were loaded with their normal settings, and the booster beam was directed to a beam dump. To evaluate the effectiveness of the online optimization, the amplitude of "Klystron 2" and the strength of the corrector magnet "C4H" were deliberately adjusted to degrade the initial injection efficiency.

In the first stage of optimization (Linac to LTB), the optimization variables included the phases of the pre-buncher, buncher, RF cavity 1, and RF cavity 2, as well as the amplitude of RF cavity 2. The objective was to maximize the charge measured at "ICT2", while applying a constraint to keep the horizontal position on the monitor "P2" within the range of [-6 mm, -4 mm].

For this optimization problem, particle swarm optimization (PSO) was used with a population size of 50. The optimization process was terminated after approximately 600 evaluations, and the best solution was applied. As shown in Fig. 2, the injection efficiency from the Linac to the LTB exceeded 90%.

ISBN: 978-3-95450-261-5 ISSN: 2673-7000 doi: 10.18429/JACoW-NAPAC2025-MOP103

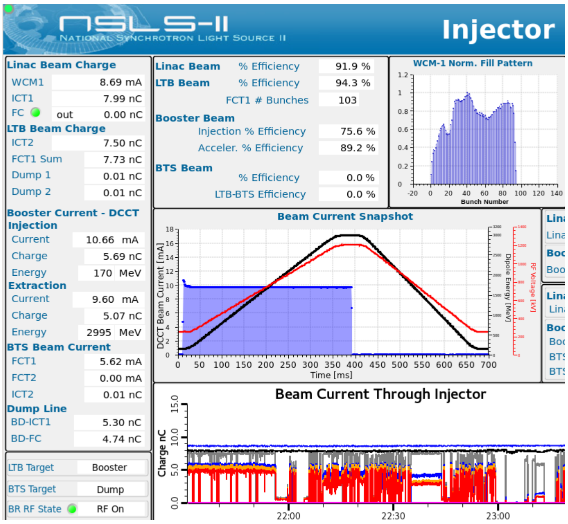

Figure 2: Interface of injector status after Linac optimization.

However, the injection efficiency into the booster remained below 80%. To address this, further optimization of the LTB beamline was carried out. Starting from the best solution found by the PSO optimization of the Linac, we continued to optimize the LTB beamline using the MG-GPO.

In our LTB optimization procedures, there were 6 tuning knobs including correctors at the LTB transport line. The objective was to maximize the booster extraction charge. The Gaussian process model used the squared exponential covariance function, with its hyperparameters optimized by maximizing the marginal likelihood [8]. A constant value of $\kappa = 0.5$ was used in the GP-UCB acquisition function. The population size was fixed at 50.

Figure 3 shows the history of all 551 evaluated solutions throughout the optimization process. As observed in the figure, MG-GPO quickly converged and found a good solution after approximately 100 evaluations. After applying the best settings for Linac and LTB, the overall injection efficiency reached approximately 95%, as shown in Fig. 4. These results demonstrate the effectiveness of the MG-GPO optimization approach.

CONCLUSION

In this study, we performed online optimization for a critical problem: maximizing the injection efficiency from the Linac to the booster. The traditional optimization method, Particle Swarm Optimization (PSO), was used to optimize the injection efficiency from the Linac to the Linac-to-booster (LTB) beamline. For the optimization from the LTB to the booster, we employed a machine learning–based method, the multi-generation Gaussian process optimizer (MG-GPO). Experimental results demonstrate that the machine learning approach effectively improves the machine's performance.

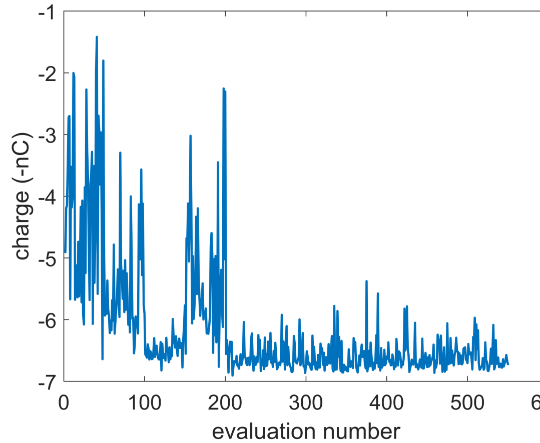

Figure 3: History of all evaluated solutions during the LTB optimization experiment with the MG-GPO.

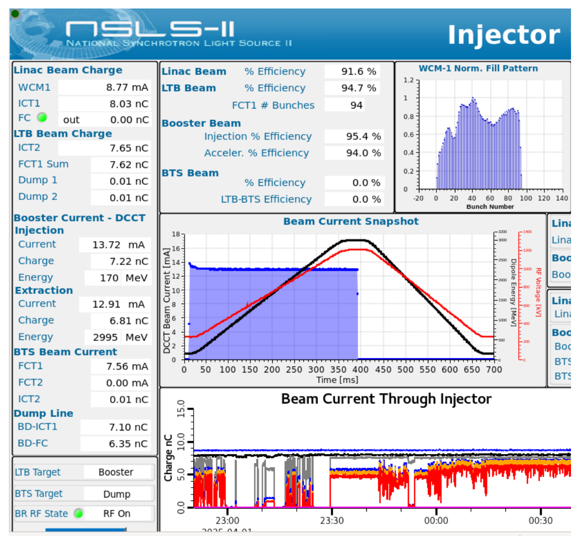

Figure 4: Interface of injector status after LTB optimization.

ACKNOWLEDGEMENTS

This work has been supported by the U.S. Department of Energy (DOE) under Contract No. DE-SC0012704. This information is controlled by the U.S. Department of Commerce Export Administration Regulations 15 CFR Parts 730-774 as EAR99 Technology.

REFERENCES

[1] X. Huang, "Robust simplex algorithm for online optimization", *Phys. Rev. Accel. Beams*, vol. 21, no. 10, p. 104 601, 2018. doi:10.1103/PhysRevAccelBeams.21.104601

[2] X. Huang, J. Corbett, J. Safranek, and J. Wu, "An algorithm for online optimization of accelerators", *Nucl. Instrum. Methods Phys. Res. A*, vol. 726, pp. 77–83, 2013. doi:10.1016/j.nima.2013.05.046

[3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II", *Trans. Evol. Comp.*, vol. 6, no. 2, pp. 182–197, 2002. doi:10.1109/4235.996017

ISBN: 978-3-95450-261-5 ISSN: 2673-7000 doi: 10.18429/JACoW-NAPAC2025-MOP103

[4] J. Kennedy and R. Eberhart, "Particle swarm optimization", in *Proc. ICNN'95*, Perth, WA, Australia, pp. 1942–1948, 1995. doi:10.1109/ICNN.1995.488968

[5] X. Huang and J. Safranek, "Online optimization of storage ring nonlinear beam dynamics", *Phys. Rev. Spec. Top. Accel. Beams*, vol. 18, no. 8, p. 084 001, 2015. doi:10.1103/PhysRevSTAB.18.084001

[6] K. Tian, J. Safranek, and Y. Yan, "Machine based optimization using genetic algorithms in a storage ring", *Phys. Rev. Spec. Top. Accel. Beams*, vol. 17, no. 2, p. 020 703, 2014. doi:10.1103/PhysRevSTAB.17.020703

[7] X. Huang, M. Song, and Z. Zhang, "Multi-objective multi-generation gaussian process optimizer for design optimization", *arXiv*, 2019. doi:10.48550/arXiv.1907.00250

[8] C. K. I. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*. MIT Press, 2006, vol. 2. doi:10.7551/mitpress/3206.001.0001

[9] M. Song, X. Huang, L. Spentzouris, and Z. Zhang, "Storage ring nonlinear dynamics optimization with multi-objective multi-generation gaussian process optimizer", *Nucl. Instrum. Methods Phys. Res. A*, vol. 976, p. 164 273, 2020. doi:10.1016/j.nima.2020.164273

[10] Z. Zhang, M. Song, and X. Huang, "Online accelerator optimization with a machine learning-based stochastic algorithm", *Mach. Learn.: Sci. Technol.*, vol. 2, no. 1, p. 015 014, 2020. doi:10.1088/2632-2153/abc81e

[11] K. Liagkouras and K. Metaxiotis, "An elitist polynomial mutation operator for improved performance of MOEAs in computer networks", in *Proc. ICCCN'13*, Nassau, Bahamas, pp. 1–5, 2013. doi:10.1109/ICCCN.2013.6614105

[12] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space", *Complex Syst.*, vol. 9, no. 2, pp. 115–148, 1995.

[13] P. Auer, "Using confidence bounds for exploitation-exploration trade-offs", *J. Mach. Learn. Res.*, vol. 3, pp. 397–422, 2002.