ISBN: 978-3-95450-261-5 ISSN: 2673-7000 doi: 10.18429/JACoW-NAPAC2025-MOP001

ACCELERATOR DRIFT COMPENSATION VIA A MODIFIED MG-GPO ALGORITHM*

R. W. Y. Yeung[1], X. Huang,† Z. Zhang, SLAC, Menlo Park, CA, USA
Y. Hao, FRIB, East Lansing, MI, USA
[1]also at FRIB

Abstract

Performance drift over long periods of operation due to changes in machines settings or the environment has been a longstanding problem for particle accelerators. Algorithms which are capable of tuning machine settings while keeping the performance within a desired threshold can be used to compensate for such drifts. We have developed a modified version of the Multi-Generation Gaussian Process Optimizer (MG-GPO) which is capable of tuning accelerator settings during user operation. The modified algorithm uses Gaussian Process regression to predict the performance of potential trial settings and removes ones with a high probability of giving too poor of a performance before selection for evaluation on the machine. The modified MG-GPO has been tested on analytic functions and applied to the SPEAR3 kicker-bump matching problem in both simulations and experiments as a proof of concept.

INTRODUCTION

Accelerator operation and commissioning benefit from online optimization algorithms that can be used to compensate for the differences between the accelerator design and the physical machine. Popular choices of optimization algorithms include the Nelder-Mead simplex method [1], the robust conjugate direction search [2], particle swarm optimization [3], and Bayesian optimization [4], which can efficiently provide optimal accelerator settings.

However, during the optimization process, these algorithms may sample accelerator settings which produce an unacceptably poor performance. This drawback is pertinent because initially-optimal settings in a particle accelerator often deteriorate in performance due to various changes in the environment, and the tendency of such algorithms to introduce poor performance in the process of finding the optimum will preclude their use during the accelerator's normal operation. The typical solution is to insert dedicated tuning periods in the schedule to re-optimize the accelerator after its performance deteriorates too much, leading to undesired interruptions to the user program. Therefore, developing online optimization algorithms that can be used to tune for optimal accelerator performance while keeping the performance within an acceptable threshold during the tuning process would be an impactful direction of research.

In this study, we have modified the Multi-Generation Gaussian Process Optimizer (MG-GPO) algorithm [5] to compensate for accelerator performance drift. The algorithm has been applied to the kicker-bump matching (KBM) problem in both simulation and experiment. In a simulated 4-dimensional KBM problem as well as an experimental 2-dimensional KBM problem, the algorithm shows promise in optimizing and maintaining the performance of the accelerator.

THE ALGORITHM

The algorithm developed in this study is based on modifications to the MG-GPO algorithm, a population-based evolutionary algorithm that incorporates Gaussian Process (GP) regression in seed selections [5]. The main modifications applied to the MG-GPO algorithm concern the modification of the GP regression algorithm to predict a time-varying input, the addition of a safety threshold, and the replacement of the candidate point generation mechanism. While MG-GPO is capable of optimizing multiple objectives simultaneously, our modified version currently addresss one objective at a time. Each candidate point represents the settings of the algorithm in the parameter space.

Candidate Point Generation

The original MG-GPO's candidate point generation schema presents several difficulties for maintaining optimized settings for a time-dependent objective function.

First, the seed points are taken from the best outputs saved from all previous generations. The positions of the points are saved according to their objective values at the time of their evaluation. With a drifting objective function, the objective value at the time of evaluation will differ from the objective value later. To address this, the seed points were changed to depend on the outputs of only the previous generation instead of on all the generations that have elapsed.

Assuming that the objective function is L-Lipschitz continuous, i.e.

$$|f(\vec{x}) - f(\vec{x}_0)| \leq L|\vec{x} - \vec{x}_0| \tag{1}$$

for objective function f, $\vec{x}, \vec{x}_0 \in \mathbf{D}$, where $\mathbf{D}$ is the domain of f, the minimum of each evaluation would be close to that of the previous generation. Given the continuity of the minimum over time, the minimum of a generation would necessarily be in the vicinity of the minimum of the previous. As such, a search near the previous minimum is sufficient to capture the new minimum, and the reliance on both MOGA and MOPSO as in the original MG-GPO is unnecessary for our purposes. Our acquisition function is modified accordingly

* Work supported by DOE Contract No. DE-AC02-76SF00515
† xiahuang@slac.stanford.edu

ISBN: 978-3-95450-261-5 ISSN: 2673-7000 doi: 10.18429/JACoW-NAPAC2025-MOP001

such that the candidate points are ordered and selected based solely on their predicted means, with no heed paid to the exploration of the parameter space.

Time-Varying Input

The GP regression in the original MG-GPO assumes a constant, non-varying model on the parameter space. However, since the problem we are addressing is the performance drift of accelerators, this assumption clearly does not hold. To improve the model accuracy, it is necessary to include time of evaluation as an additional parameter. This effectively increases the dimensionality of the problem by 1.

In each generation, MG-GPO generates $20N$ candidate points among which N points are selected for evaluation on the machine. Assuming the final point of the previous generation was evaluated at time t, the times of evaluation of this generation would range from $t+1$ to $t+N$, much less than the $t+20N$ that would be required to cover all the candidate points with one valid time each. Even more importantly, the predicted points will be sorted after the prediction prior to being passed onto the evaluation, such that, even if the N points are known beforehand, their order of evaluation will be scrambled relative to their positions as candidates. Therefore, we decided to treat all points as if they would be evaluated at the half-integer time $t + (N + 1)/2$, the mean time of the generation. As a result, predictions towards the beginning and end of a generation will be less accurate. Generation sizes, i.e. N, should be kept small relative to the drift rate such that this error within a generation should be small relative to the value of the safety threshold.

Safety Threshold

The safety threshold is the threshold of acceptable performance of the machine, to be specified by users. The goal of the algorithm is to keep the machine performance below this threshold. GP regression can predict a probability distribution p of the result for each candidate point:

$$p = \frac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{(y-\mu)^2}{2\sigma^2}} \tag{2}$$

given predicted mean μ and standard deviation σ. To obtain the probability of a certain predicted point being below the safety threshold, we integrate as follows:

$$P = \frac{1}{\sqrt{2\pi}\,\sigma} \int_{-\infty}^{y} e^{-\frac{(y'-\mu)^2}{2\sigma^2}} dy' \tag{3}$$

We arbitrarily choose a safety probability of $P = 0.8$, adjustable based on the certainty required by the problem, and remove all candidate points whose predicted results do not fulfill this condition prior to sorting and evaluation. This guarantees any points predicted to be unsafe will be removed prior to evaluation.

RESULTS

The algorithm has been applied in simulation and experiment to the SPEAR3 kicker-bump matching (KBM) problem.

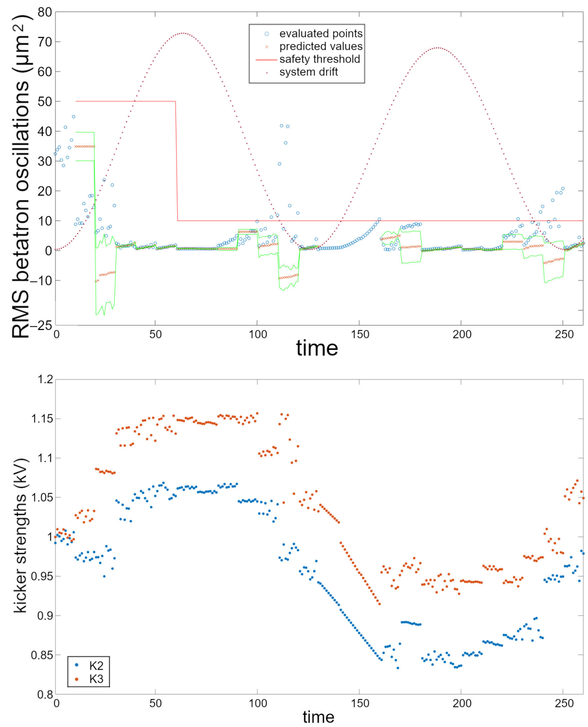

Figure 1: 2-D kicker-bump matching simulation of the SPEAR3 storage ring, with an initial safety threshold of $2\,\mu m^2$, reduced to $0.5\,\mu m^2$ after generation 5. The optimization knobs are the kicker strengths of kickers K2 and K3.

SPEAR3 has three horizontal injection kickers, K1, K2 and K3, which are used to make a closed kicker bump during injection. It is desired to maintain the closed bump condition to minimize perturbation to the stored beam after the kickers are fired. The kicker amplitudes, timing, and pulse widths are knobs available to optimize the kicker bump matching, while turn-by-turn BPM data are used to characterize the residual oscillation on the stored beam.

For the purpose of testing the algorithm, the kicker strength K1 is modulated sinusoidally to simulate drifting accelerator settings in the following pattern:

$$\nu(t) = \nu_0 + \sigma_d \sin\frac{2\pi t}{T} \tag{4}$$

with initial K1 value ν_0, K1 drift amplitude σ_d, and drift period T. Generation sizes are set to $N = 10$. The knobs tuned for the 2-dimensional case are the kicker strengths of the other two kickers, K2 and K3. In the 4-dimensional case, the kicker pulse widths of these two kickersare also adjusted. In Figs. 1- 3, the green lines show $\pm 1\sigma$ of the GP prediction around the mean.

Simulation

Both 2-D and 4-D simulations have been carried out at $T = 250$ and $\sigma_d = 0.1\,\text{kV}$, with an added Gaussian noise of $\varepsilon \sim N(0, \sigma_n = 0.01/3)$. σ_n is chosen such that the error will fall within ± 0.01 of the objective.

ISBN: 978-3-95450-261-5 ISSN: 2673-7000 doi: 10.18429/JACoW-NAPAC2025-MOP001

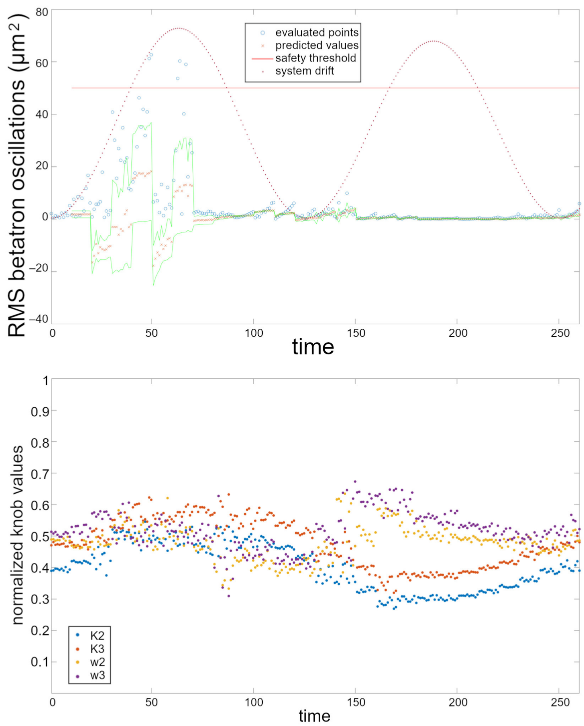

Figure 2: 4-D kicker-bump matching simulation of the SPEAR3 storage ring, with a safety threshold of 30 μm^2. The optimization knobs are the kicker strengths and pulse widths of kickers K2 and K3.

Fig. 1 is the result of the KBM simulation with 2 knobs, the kicker strengths of K2 and K3, being adjusted by the algorithm. The 4-dimensional simulation results are shown in Fig. 2.

The top plots of the figures show the evolution of the objective function as evaluated, the would-be objective value if no compensation is dialed in, the safety threshold, and the value predicted by the GP model. The $\pm\sigma$ confidence lines of the GP model are also plotted (in green lines). The bottom plot show the variation of the knobs during the tuning process.

For both the 2-D and 4-D cases, the algorithm is able to keep the objective function below the safety threshold most of the time while it is moving the knobs to compensate the otherwise large variations due to simulated drifts. However, improvement to the algorithm is needed as the objective function can still occasionally break the safety threshold requirement. Such occasions typically coincide with poor GP model predictions, indicating the importance of further improving GP model accuracy.

Experiment

The experiment was carried out as a 2-D problem, with $T = 261$ and $\sigma_d = 0.1$ kV, roughly corresponding to 0.1 mrad.

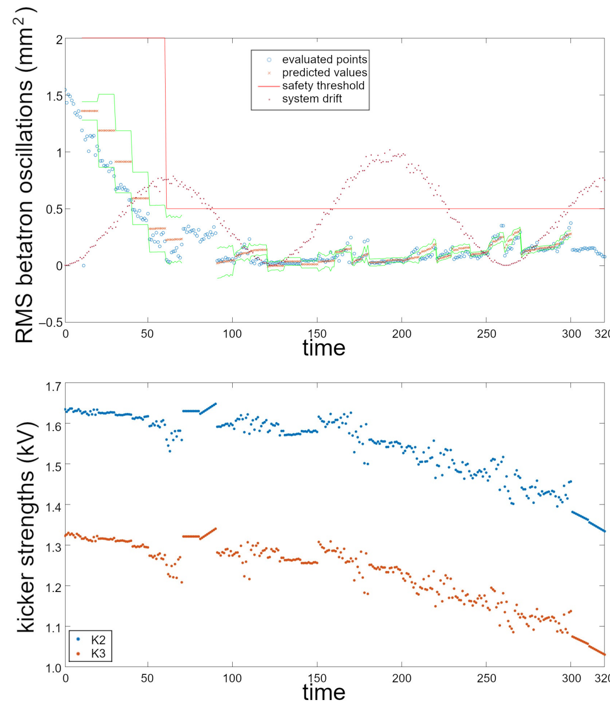

Figure 3: Kicker-bump matching experiment on the SPEAR3 storage ring, with an initial safety threshold of 2 mm^2, reduced to 0.5 mm^2 after generation 5.

In the experiment, the safety threshold was initially set to 2 mm^2, and reduced to the target of 0.5 mm^2 after 5 generations, after the objective function had fallen under said target. The results are shown in Fig. 3, where the performance of the algorithm is compared against that of the system drift.

The algorithm succeeded in keeping the evaluations within the safety region, although there are episodes where the objective function increases within the 10 points of the same generation.

CONCLUSION

We modified the MG-GPO algorithm to add safety features such that it can be used to tune accelerator performances online during user operation with the purpose of using it to compensate performance drift due to changes in the environment. Simulation and experimental tests with the kicker bump matching experiment for the SPEAR3 storage ring showed that the modified algorithm has been largely successful. It is capable of keeping the objective function below a pre-specified safety threshold, while tuning knobs to compensate the large drift introduced by intentionally changing the amplitude of one kicker. Further improvement is needed for better safety assurance and smaller perturbation to the machine.

REFERENCES

[1] J. A. Nelder and R. Mead, "A simplex function for function minimization," *The Computer Journal*, pp. 308–313, 1965. doi:10.1093/comjnl/7.4.308

ISBN: 978-3-95450-261-5 ISSN: 2673-7000 doi: 10.18429/JACoW-NAPAC2025-MOP001

[2] X. Huang, J. Corbett, J. Safranek, and J. Wu, “An algorithm for online optimization of accelerators,” *Nucl. Instrum. Methods*, vol. 726, no. 21, pp. 77–83, 2013. doi:10.1016/j.nima.2013.05.046

[3] J. Kennedy and R. Eberhart, “Particle swarm optimization,” *Proceedings of ICNN’95 - International Conference on Neural Networks*, Perth, WA, Australia, 1995, pp. 1942–1948. doi:10.1109/ICNN.1995.488968

[4] J. Duris *et al.*, “Bayesian optimization of a free-electron laser,” *Phys. Rev. Lett.*, p. 124801, Dec 2020. doi:10.1103/PhysRevLett.124.124801

[5] X. Huang, Z. Zhang, and M. Song, “Multi-objective multi-generation gaussian process optimizer,” in *Proc. 12th Int. Particle Acc. Conf.*, Campinas, SP, Brazil, May 2021, pp. 3383–3386. doi:10.18429/JACoW-IPAC2021-WEPAB304