ACCELERATOR PERFORMANCE DRIFT COMPENSATION WITH A MODIFIED MG-GPO ALGORITHM*

R. W.-Y. Yeung†, Michigan State University, East Lansing, MI, USA
X. Huang, SLAC National Accelerator Laboratory, Menlo Park, CA, USA
Y. Hao, Facility for Rare Isotope Beams, East Lansing, MI, USA

Abstract

Performance drift has been a longstanding problem for accelerators. A desirable solution is to tune the machine slowly and gently to compensate for such drift. Previously, we presented a version of the Multi-Generation Gaussian Process Optimizer which tunes accelerator settings during operation to maintain optimal performance. In this paper, we present an improved version of the algorithm and its application test examples, in which it corrects deviations from the ideal orbit caused by a drifting orbit corrector magnet and a drifting injection kicker magnet respectively. The modified algorithm takes measures to ensure the accuracy of the Gaussian process regression models and to improve the validity of the new trial solutions. We demonstrate that this is a promising development toward using safe, real-time tuning algorithms during accelerator programs to compensate for performance drift.

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 with unacceptably poor performances, making their use during the accelerator's normal operation ill-advised. This precludes their use in compensating for the deterioration of performance in particle accelerators, often caused by environmental factors.

The typical solution is to insert dedicated tuning periods in the schedule to re-optimize the accelerator after its performance deviates too much from the optimum, 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 profitable direction of research.

In a previous study, a modified version of the Multi-Generation Gaussian Process Optimizer (MG-GPO) algorithm [5] was presented, in which it was applied to the kicker-bump matching problem [6]. This version has had its candidate point generation and selection mechanisms modified to provide a more accurate picture of the changing objective function landscape for the Gaussian process, which brings it closer to the original MG-GPO algorithm.

IMPROVEMENTS

Between the version of the code presented in [6] and the current version, several changes have been made in order to improve the performance, including the modification of the candidate proposal mechanisms, the reintroduction of an exploration term in the objective function selection, and some bug fixes.

Reintroducing the candidate proposal mechanisms of MG-GPO, which involve particle swarm optimization (PSO) and simulated binary crossover (SBX), albeit in a modified form, has improved the performance of the algorithm. The primary difference involves the time-dependence of the PSO portion of the algorithm. MG-GPO implicitly assumes a static optimum which can be approached. Thus the search range of PSO is, in addition to the distance between the optima of two generations, also a decreasing function of time. As the assumption no longer holds, that is, the optimum of the objective function changes over time, the search range decreasing over time would simply impair the performance of the algorithm. Therefore, the time dependence has been removed, allowing the algorithm to propose candidate points in a larger region to compensate for the drifting optimum.

Similarly, the re-addition of an exploration term aids in providing a more thorough picture of the objective function landscape to the GP, leading to more accurate predictions. Similarly to the candidate generation, the original MG-GPO algorithm has an exploration term which decreases over time. This is preferable for a static optimum, as the overall trend of the optimizer would be to approach that optimum, but with a drifting optimum, it would be questionable as the optimum may not approach a static value over time. Thus having a constant-value exploration term would allow the algorithm to have a better picture of the changing objective function landscape.

The two changes mentioned above are spiritually continuations of one of the modifications in [6], which involve changing the record of optimal points such that only the best points in the last two generations are included, i.e. optima using settings from too long ago are not considered, as the knob settings represented therein are not guaranteed to be optimal or even near-optimal at later times.

One change which proved to not be as successful as desired was the attempt to maintain safe evaluations based

* Work supported under DOE Contract No. DE-AC02-76SF00515
† yeung@frib.msu.edu

on an extrapolation of safe points in previous generations. Earlier versions of the algorithm had experienced issues in finding safe candidates, at which point it ceases to function.

In the hope of extending the uptime of the algorithm, new evaluation points were proposed based on a linear extrapolation of the knob settings corresponding to safe points from previous generations. This was done in hopes of adding more safe points to the training data of the algorithm, allowing it to propose safe candidates once more and extend the lifetime of the algorithm. However, multiple tests have shown this almost never succeeds. As newer versions of the algorithm mostly succeed in finding candidates in the safety region without this change, the idea has not been revisited.

RESULTS

To show the efficacy of these changes, several experimental runs were made on SPEAR3. Previously the algorithm was tested on the kicker-bump matching problem [6], but we elected to use an orbit corrector matching problem this time due to its relative safety. This allows us to test changes to the algorithm with a lower risk of damage to machine components. The latter problem also allows us to increase the dimensionality of the problem such that all knobs are of the same physical variables, as opposed to the kicker-bump matching problem where increasing dimensionality necessarily implies variables beyond magnet strength.

Three separate set-ups were tested experimentally, guided by successful simulations, with success defined by reaching 2000 evaluations. The settings of the modified MG-GPO algorithm were the same between simulation and experiment. However, the parameters of the objective function differ between the two, namely in noise level and in period. In simulation, we could control the simulated noise level, but this level of control was impossible for the experiment.

While the period of the drifting corrector is set manually, the time it takes for the drift code to perform one evaluation and for the algorithm to do so differ by varying amounts. For example, in the time it takes MG-GPO to perform 250 evaluations, the simulated drift code may have performed 400 evaluations or 275 evaluations, and this number can differ in different runs.

Set-up 1 was a 2-dimensional problem in which corrector magnet 6.3 was set to drift in a sinusoidal fashion, while two other magnets, 6.1 and 6.4, were controlled by the algorithm to compensate for its drift. This result is shown in Fig. 1.

Set-ups 2 and 3 were 4-dimensional problems; in these cases, the algorithm was allowed to control four magnets. Set-up 2 has corrector magnets 6.3 and 11.3 set to drift sinusoidally, with magnets 6.1, 6.4, 11.1, and 11.4 set to compensate (Fig. 2). Set-up 3 has corrector magnets 6.3 and 6.4 set to drift sinusoidally, with magnets 5.4, 6.1, 7.1, and 7.3 compensating (Fig. 3). The results are shown in Figs. 2 and 3.

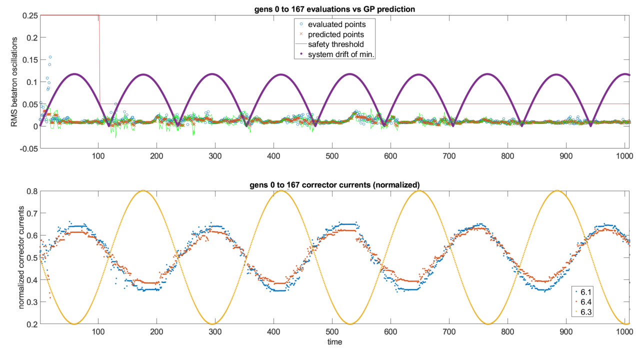

Figure 1: Results of the modified MG-GPO algorithm on the 2-dimensional performance drift compensation problem. Corrector magnet 6.3 is set to drift with some period while the algorithm is allowed to change the strengths of magnets 6.1 and 6.4 to compensate for the change (set-up 1).

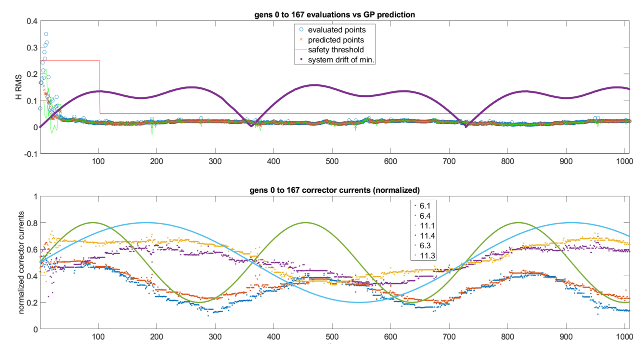

Figure 2: Results of the modified MG-GPO algorithm on the 4-dimensional performance drift compensation problem. Corrector magnets 6.3 and 11.3 are set to drift with some period while the algorithm is allowed to change the strengths of magnets 6.1, 6.4, 11.1, and 11.4 to compensate for the change (set-up 2).

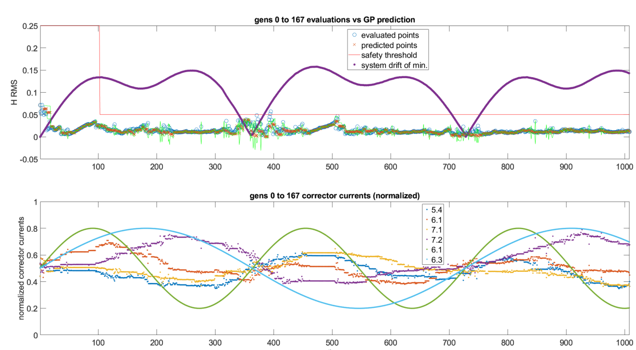

Figure 3: Results of the modified MG-GPO algorithm on the 4-dimensional performance drift compensation problem. Corrector magnets 6.3 and 6.4 are set to drift with some period while the algorithm is allowed to change the strengths of magnets 5.4, 6.1, 7.1, and 7.3 to compensate for the change (set-up 3).

Due to time constraints, an experimental run is considered successful once the algorithm reaches 1000 points evaluated, which takes approximately 30 minutes. As Figs. 1–3 show, all runs using the latest version of the algorithm succeeded by this metric.

CONCLUSION

We have further modified the MG-GPO algorithm to improve its ability to search the parameter space for the drifting optimum of the objective function, which improved its performance in compensating for said drift.

Simulation and experimental tests with the SPEAR3 corrector magnets have shown a marked improvement in the algorithm performance with respect to keeping objective function outputs below a pre-specified safety threshold, while tuning knobs to compensate for the large drift introduced by intentionally changing the magnetic field amplitude of corrector magnets.

From the results presented above, the modified MG-GPO algorithm succeeds in compensating for accelerator performance drift. In the future, we expect to apply this algorithm to different problems to prove its ability to be a general optimization algorithm for performance drift problems.

REFERENCES

[1] J. A. Nelder and R. Mead, "A simplex function for function minimization," *The Computer Journal*,vol. 7, no. 4, pp. 308–313, 1965. doi:10.1093/comjnl/7.4.308

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

[3] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. of ICNN'95*, 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.*, vol. 124, p. 124801, Mar. 2020. doi:10.1103/PhysRevLett.124.124801

[5] X. Huang, Z. Zhang, and M. Song, "Multi-objective multi-generation Gaussian process optimizer," in *Proc. IPAC'21*, Campinas, Brazil, May 2021, pp. 3383–3386. doi:10.18429/JACoW-IPAC2021-WEPAB304

[6] R. W.-Y. Yeung, X. Huang, Z. Zhang, and Y. Hao, "Accelerator drift compensation via a modified MG-GPO algorithm," in *Proc. NAPAC'25*, Sacramento, California, USA, Aug. 2025, pp. 34–38. doi:10.18429/JACoW-NAPAC2025-MOP001