ISBN: 978-3-95450-232-5 ISSN: 2673-7000 doi:10.18429/JACoW-NAPAC2022-THXD4

ONLINE ACCELERATOR TUNING WITH ADAPTIVE BAYESIAN OPTIMIZATION*

N. Kuklev†, M. Borland, G. Fystro, H. Shang, Y. Sun
Advanced Photon Source, Argonne National Laboratory, Lemont, IL, USA

Abstract

Particle accelerators require continuous adjustment to maintain beam quality. At the Advanced Photon Source (APS) this is accomplished using a mix of operator-controlled and automated tools. To improve the latter, we explored the use of machine learning (ML) at the APS injector complex. The core approach we chose was Bayesian optimization (BO), which is well suited for sparse data tasks. To enable long-term online use, we modified BO into adaptive Bayesian optimization (ABO) though auxiliary models of device drift, physics-informed quality and constraint weights, time-biased data subsampling, digital twin retraining, and other approaches. ABO allowed for compensation of changes in inputs and objectives without discarding previous data. Benchmarks showed better ABO performance in several simulated and experimental cases. To integrate ABO into the operational workflow, we developed a Python command line utility, pysddsoptimize, that is compatible with existing Tcl/Tk tools and the SDDS data format. This allowed for fast implementation, debugging, and benchmarking. Our results are an encouraging step for the wider adoption of ML at APS.

INTRODUCTION

Modern particle accelerators face increasing performance demands, resulting in tighter tolerances on accuracy and stability [1]. Due to cost, physical limits, and external factors, some amount of continuous parameter adjustment is constantly required. Historically, this tuning required expert guidance and intuition, with software tools only allowing for a partial automation. With the explosion of machine learning methods in the last decade, there is immense interest in making use of the newly available algorithms to improve reliability, reduce expert workload, and provide higher performance to the users.

A key application of ML for accelerators is in parameter optimization, whereby one or multiple objectives are tuned through an intelligent search of the parameter space. A number of conventional optimization methods are already in use, including simplex[2, 3], RCDS [4], genetic algorithms [5], extremum seeking [6], and several others. New ML methods include Bayesian optimization (BO) [7], reinforcement learning [8], and others. BO is of special interest since it allows efficient black-box function optimization with few samples, taking advantage of any prior physics model knowledge provided to the algorithm. This paper first reviews the basic BO process, and then discusses our contributions - a set of improvements that permits for continuous, robust, and adaptive BO use for optimizing time-varying systems.

* The work is supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, under Contract No. DE-AC02-06CH11357.
† nkuklev@anl.gov

ADAPTIVE BAYESIAN OPTIMIZATION

In standard BO process, system output is described by

$$\mathbf{y} = f(\mathbf{x}) + \varepsilon \tag{1}$$

where $f(\mathbf{x})$ is the black-box function of interest and $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$ the added noise. Vector $\mathbf{x}$ has dimension of n x d where d is the parameter space size and n the number of measurements. Using Gaussian Process (GP) a surrogate model for f can be parameterized as a multivariate normal distribution with a mean $m(\mathbf{x})$ and covariance kernel $k(\mathbf{x}, \mathbf{x}')$ as

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \tag{2}$$

The kernel is used to evaluate the similarity between values of f at $\mathbf{x}$ and $\mathbf{x}'$, and its' appropriate choice is critical for good GP convergence. Existing knowledge about the system can be encoded through prior distributions on kernel and mean, with the distribution parameters called *hyper-parameters*. During model fitting hyper-parameters are updated using Bayes' rule (conditioned on observed data) and posterior probability distribution $\mathrm{p}(\mathbf{f} \mid \mathbf{y}, \mathbf{x})$ can then be sampled to get model predictions [9]. BO evaluates a special 'acquisition' function over a fitted GP model so as to predict the best next location(s) to sample. A variety of analytic and Monte-Carlo acquisition functions exist, with one of simplest being the upper confidence bound (UCB)

$$\mathrm{UCB}(\mathbf{x}) = \mu(\mathbf{x}) + \sqrt{\beta} * \sigma(\mathbf{x}) \tag{3}$$

where mean μ and variance σ are provided by the GP model. The parameter β allows for trade-off between exploration (risk for high reward) and exploitation (use known good configuration).

Time-varying GP Models

The above discussion grouped all input parameters into vector $\mathbf{x}$, representing for example several magnet currents. A simple BO process would proceed by using standard isotropic kernels, such as Matérn and radial basis function [9], and only fit freshly collected data. To improve convergence speed, previous work has successfully used historic data to train covariance distributions [10]. Such pre-training works well when conditions are reproducible. However, some accelerators also have undesired and poorly modelled time-dependent drifts. ML methods dealing with

ISBN: 978-3-95450-232-5 ISSN: 2673-7000 doi:10.18429/JACoW-NAPAC2022-THXD4

dynamic systems are referred to as adaptive ML (AML). Several examples of their use in accelerators can be found in [1], including model-free extremum seeking and adaptive neural networks. Our work seeks instead to use an explicitly time-aware model as part of BO algorithm, which to our knowledge has not yet been demonstrated for accelerators. Formally, we extend f with explicit time dimension t, such that the system is now described by

$$\mathbf{y} = f(t, \mathbf{x}) + \varepsilon \tag{4}$$

Time effects have different expected correlation patterns, requiring careful composition of sub-kernels. Kernel multiplication and addition can be thought of as logical AND and OR operations along any shared dimensions. For example, composition of periodic and RBF kernels results in a 'locally periodic' correlation - a periodic (not necessarily sinusoidal) function that can slowly change shape:

$$k_{lp} = \sigma^2 \exp\left(\frac{2\sin^2\left(\pi|t-t'|/p\right)}{l^2}\right)\exp\left(\frac{-(t-t')^2}{2l^2}\right) \tag{5}$$

Hyper-parameters of this kernel are output variance σ, period p, and lengthscale l. Higher-dimensional kernels can be assembled as

$$k_{product} = k_x(x, x') \times k_t(t, t') \tag{6}$$

$$k_{add} = k_x(x, x') + k_t(t, t') \tag{7}$$

with AND/OR applying in each dimension. In experimental applications, some intuition about the underlying drift processes can be gained from historical data but precise values for parameters like number of periodic signals are difficult to specify. An example of experimental beam position data from the APS linac is shown in Fig. 1, demonstrating long-term linear drift as well as two distinct periodic signals. For unknown reasons, one of the periodic signal components sometimes disappears, demonstrating the need for not only time-aware but also time-adaptive models.

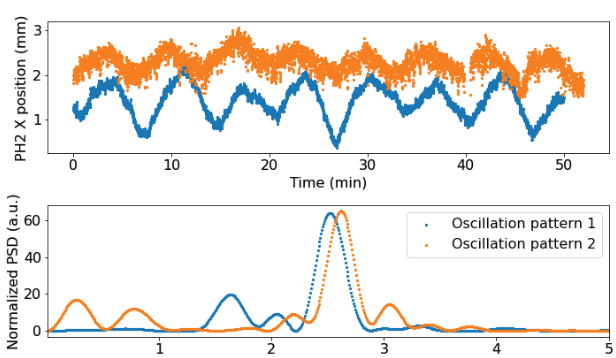

Figure 1: APS linac beam position drift and PSD showing two distinct types of signals observed on different days.

Automatic kernel selection and deep kernel learning (DKL) are an active area of research, with several promising results [11]. In this work we chose a standard spectral mixture (SM) kernel as a natural adaptive extension of the local periodicity concept [12]. SM is the basis for more complex DKL approaches - it can approximate any stationary kernel, including ones with multiple oscillatory and local correlations, and can be easily interpreted in terms of kernel spectral density. To account for non-stationary linear drifts, a linear kernel can be added when necessary. For spatial dimensions, we use a standard Matérn kernel ($\nu = 2.5$) with automatic relevance determination enabled. Thus, our final ABO model is given by

$$k(t, t', x, x') = (k_{SM}(t, t') + k_l(t, t')) \times \sigma^2 k_{Mt}(\mathbf{x}, \mathbf{x}') \tag{8}$$

History-aware Efficient Evaluation

An important GP/BO limitation is poor performance scaling with number of data points ($\mathcal{O}(n^3)$) and number of dimensions. With exact GP methods, computation time dominates past a few thousand data samples. For accelerator applications, this limit can be quickly exceeded. To optimize beam time usage, it is critical to generate new candidates at a speed comparable to sampling rate. Adaptive BO is particularly challenging since it requires an observational period of sufficiently long duration and sufficiently high sampling rate to capture all relevant system behaviour. We explored some of approximate and scalable GP algorithms available in GPyTorch [13], but nonetheless eventually encountered performance issues.

Due to the need to preserve long timescales, instead of a simple data cutoff or random subsampling, ABO uses novel time-biased importance bandwidth subsampling. The core idea is to adaptively discard old data that does not contribute to the overall model outside some specific time scale. Among such candidate points, subsampling is weighted by impact on current model fit/prediction. For example, with ideal periodic noiseless signals, data older than a complete period is allowed to be subsampled, and points in slowly varying regions/dimensions as reflected by small variance improvements will be discarded first. For fast signals, averaging is also performed on the main signal part so as to reduce noise. The exact cutoff is determined by the upper bandwidth threshold required from ABO, typically 0.05 Hz (20 s period). This strategy guarantees maximum possible preservation of spectral and correlation information.

Constraints and Priors

The final important element of our algorithm is the use of several tools to ensure ABO makes conservative parameter estimates and does not exceed safety constraints (i.e. beam losses) or operational limits (i.e., to avoid hysteresis). We implement GP feasibility models to predict constraints [14], and also limit slew rates and overall control bandwidth through use of proximal hard and soft cutoffs based on expected sampling rate and measurement delay [15]. For acquisition functions and parameter distributions, we use extremely conservative priors/parameters favouring exploitation (i.e. low β), ensuring sampling occurs in known good regions. These methods guarantee ABO is well behaved

ISBN: 978-3-95450-232-5 ISSN: 2673-7000 doi:10.18429/JACoW-NAPAC2022-THXD4

and robust, and can in worst case be quickly overridden by standard control tools.

RESULTS

We first tested ABO on several simulated problems, ranging from synthetic functions to linac particle tracking simulations. Figure 2 shows an example application to a simple case of sinusoidal corrector drift. After 30 initial points taken without any current changes, denoted by horizontal blue line, both methods were allowed to run with identical settings. After approximately a single period, ABO converges on the oscillation frequency and starts following the drift without any lag, demonstrating that it is in fact predicting the future location via the model. Simple BO meanwhile sees the drift as just large noise and oscillates around central input value, losing significant performance even after 4 periods.

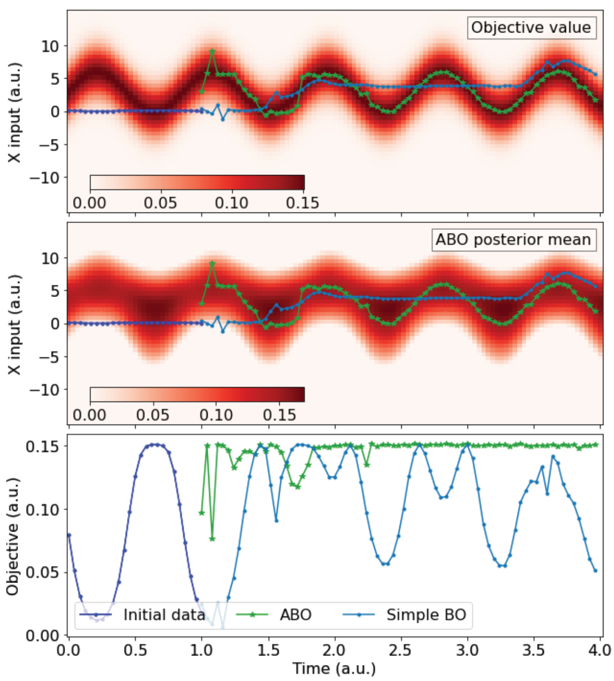

Figure 2: ABO and BO runs for sinusoidal corrector drift.

Experimental tests were done at the APS injector complex. It consists of a linac [16] and two rings [17, 18] that bring bunches to 7 GeV. To stabilize beam parameters, several proportional feedback controllers, called control laws [19], are used. They operate with pre-computed inverse response matrices based on experimental data. A recent analysis of beam parameters noted elevated high frequency jitter levels with control laws enabled, potentially caused by BPM noise or deviations from expected lattice parameters. With control laws off however, slower but larger amplitude oscillations were observed, shown in Fig. 1. While not large enough to impact overall injector efficiency, eliminating both the drifts and high frequency noise is desirable for experiments in the linac extension area. We tested ABO for that purpose by picking a suitable BPM and corrector pair at the end of the linac, and using as objective the mean squared trajectory error. Results of this test are given in Fig. 3.

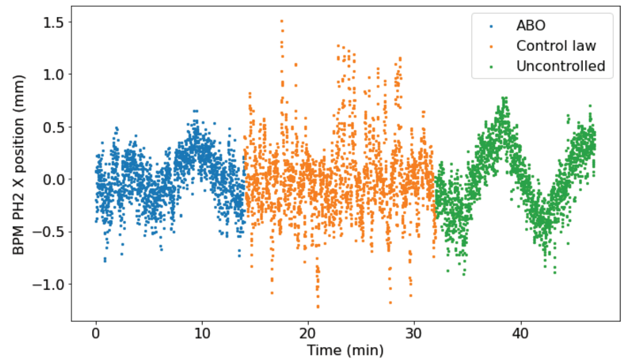

Figure 3: ABO and control law trajectory stability.

For the particular dataset shown, the linac exhibited the 'two frequency' mode described previously and ABO was constrained to 3 mixture components to speed up convergence. The previous 15 minutes of history were used as the training set for each run, subsampled to 100 points to improve BPM noise. Even with these limitations, ABO was clearly able to remove the main oscillatory signal at 2.5 mHz, with corresponding RMS jitter lowered to 0.23 mm as compared to 0.35 mm for control law and 0.33 mm for uncontrolled cases. Simulations indicate that both components can be fitted robustly with longer data collection time, but convergence speed is strongly dependent on noise levels.

As first step towards operational use, ABO was integrated with other optimization methods into a Python package *APSopt*. Its key features include full support for SDDS file format [20] (via PySDDS library [21]) and an SDDS-toolkit compatible command line interface [22–24]. This allows interchangeable BO/ABO algorithm use in place of existing simplex and RCDS ones. We are currently testing ABO long-term stability and robustness in a virtual environment which replays realistic data in an automated testing loop.

CONCLUSION

Improving robustness and adaptability of ML-based optimizers is a crucial step in making these tools useful in day-to-day accelerator operation. We have demonstrated that BO can be modified with an explicitly time-dependent adaptive SM kernel to fit a wide variety of experimentally-relevant drifts. Results on both simulated and experimental tasks, while not perfect, significantly improve on naive BO performance. Future work will focus on improving GP hot-start with pre-training on more varied historical data, and exploring DKL neural networks for further kernel model improvements.

ACKNOWLEDGMENTS

Authors acknowledge ANL Laboratory Computing Resource Center (LCRC) for providing the computational resources. We also thank APS operators for their help in late-night debugging and data collection.

ISBN: 978-3-95450-232-5 ISSN: 2673-7000 doi:10.18429/JACoW-NAPAC2022-THXD4

REFERENCES

[1] A. Scheinker, "Adaptive machine learning for time-varying systems: Low dimensional latent space tuning," *Journal of Instrumentation*, vol. 16, no. 10, p. P10008, 2021. doi:10.1088/1748-0221/16/10/P10008

[2] L. Emery, M. Borland, and H. Shang, "Use of a General-Purpose Optimization Module in Accelerator Control," in *Proc. PAC'03*, Portland, OR, USA, May 2003. https://jacow.org/p03/papers/WPPE002.pdf

[3] 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

[4] X. Huang, J. Corbett, J. Safranek, and J. Wu, "An algorithm for online optimization of accelerators," *Nuclear Instruments and Methods in Physics Research. Section A, Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 726, no. 21, pp. 77–83, 2013. doi:10.1016/j.nima.2013.05.046

[5] M. Borland, V. Sajaev, L. Emery, and A. Xiao, "Multi-objective direct optimization of dynamic acceptance and lifetime for potential upgrades of the Advanced Photon Source.," Tech. Rep. ANL/APS/LS-319, 986303, 2010. doi:10.2172/986303

[6] A. Scheinker, "Adaptive Machine Learning for Robust Diagnostics and Control of Time-Varying Particle Accelerator Components and Beams," *Information*, vol. 12, no. 4, p. 161, 4 2021. doi:10.3390/info12040161

[7] M. Balandat *et al.*, "BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 21 524–21 538. https://proceedings.neurips.cc/paper/2020/hash/f5b1b89d98b7286673128a5fb112cb9a-Abstract.html

[8] V. Francois-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, "An Introduction to Deep Reinforcement Learning," *Foundations and Trends® in Machine Learning*, vol. 11, no. 3-4, pp. 219–354, 2018. doi:10.1561/2200000071

[9] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2005. doi:10.7551/mitpress/3206.001.0001

[10] J. Duris *et al.*, "Bayesian Optimization of a Free-Electron Laser," *Physical Review Letters*, vol. 124, no. 12, p. 124 801, 2020. doi:10.1103/PhysRevLett.124.124801

[11] A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing, "Deep Kernel Learning," *arXiv preprint*, 2015. doi:10.48550/arXiv.1511.02222

[12] A. G. Wilson and R. P. Adams, "Gaussian Process Kernels for Pattern Discovery and Extrapolation," *arXiv preprint*, 2013. doi:10.48550/arXiv.1302.4245

[13] G. Pleiss, J. R. Gardner, K. Q. Weinberger, and A. G. Wilson, "Constant-Time Predictive Distributions for Gaussian Processes," *arXiv preprint*, 2018. http://arxiv.org/abs/1803.06058

[14] D. Eriksson and M. Poloczek, "Scalable Constrained Bayesian Optimization," *arXiv preprint*, 2021. doi:10.48550/arXiv.2002.08526

[15] R. Roussel *et al.*, "Turn-key constrained parameter space exploration for particle accelerators using Bayesian active learning," *Nature Communications*, vol. 12, no. 1, p. 5612, 1 2021. doi:10.1038/s41467-021-25757-3

[16] A. Nassiri, G. Mavrogenes, and W. Wesolowski, "The Linac Injector for the ANL 7 GeV Advanced Photon Source," in *Proc. LINAC'90*, Albuquerque, NM, USA, Sep. 1990, pp. 611–613. https://jacow.org/l90/papers/TH411.pdf

[17] M. Borland, "Commissioning of the Argonne Positron Accumulator Ring," in *Proc. PAC'95*, Dallas, TX, USA, May 1995.

[18] S. V. Milton, "The APS Booster Synchrotron: Commissioning and Operational Experience," in *Proc. PAC'95*, Dallas, TX, USA, May 1995.

[19] L. Emery, "Commissioning Software Tools at the Advanced Photon Source," in *Proc. PAC'95*, Dallas, TX, USA, May 1995.

[20] M. Borland, "A Self-Describing File Protocol for Simulation Integration and Shared Post-Processors," in *Proc. PAC'95*, Dallas, TX, USA, May 1995.

[21] N. Kuklev, "pySDDS: Efficient python SDDS reader and writer," https://github.com/nikitakuklev/pysdds

[22] M. Borland, "Applications Toolkit for Accelerator Control and Analysis," in *Proc. PAC'97*, Vancouver, Canada, May 1997.

[23] R. Soliday, M. Borland, L. Emery, and H. Shang, "New Features in the SDDS Toolkit," in *Proc. PAC'03*, Portland, OR, USA, May 2003. https://jacow.org/p03/papers/FPAG005.pdf

[24] H. Shang, M. Borland, L. Emery, and R. Soliday, "New Features in the SDDS-Compliant EPICS ToolkitMicrosoft Word - FPAG004.doc," in *Proc. PAC'03*, Portland, OR, USA, May 2003. https://jacow.org/p03/papers/FPAG004.pdf