ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TUPDP086

OPERATIONAL TOOL FOR AUTOMATIC SETUP OF CONTROLLED LONGITUDINAL EMITTANCE BLOW-UP IN THE CERN SPS

N. Bruchon*, I. Karpov, N. Madysa[1], G. Papotti, D. Quartullo , CERN, 1211 Geneva, Switzerland
[1]currently at GSI, 64291 Darmstadt, Germany

Abstract

The controlled longitudinal emittance blow-up is necessary to ensure the stability of high-intensity LHC-type beams in the CERN SPS. It consists of diffusing the particles in the bunch core by injecting a bandwidth-limited noise into the beam phase loop of the main 200 MHz RF system. Obtaining the correct amplitude and bandwidth of this noise signal is non-trivial, and it may be tedious and time-demanding if done manually. An automatic approach was developed to speed up the determination of optimal settings. The problem complexity is reduced by splitting the blow-up into multiple sub-intervals for which the noise parameters are optimized by observing the longitudinal profiles at the end of each sub-interval. The derived bunch lengths are used to determine the objective function which measures the error with respect to the requirements. The sub-intervals are tackled sequentially. The optimization moves to the next one only when the previous sub-interval is completed. The proposed tool is integrated into the CERN generic optimization framework that features pre-implemented optimization algorithms. Both single- and multi-bunch high-intensity beams are quickly and efficiently stabilized by the optimizer, used so far in high-intensity studies. A possible extension to Bayesian optimization is being investigated.

INTRODUCTION

Given the strict beam parameter constraints to be met before injecting into the LHC, maintaining longitudinal stability in the SPS is an essential task. Stability relies on the increased synchrotron frequency spread thanks to a double-harmonic RF system, as well as on the controlled longitudinal emittance blow-up. Both techniques increase the synchrotron frequency spread within the bunch, enhancing Landau damping [1, 2].

The controlled longitudinal emittance blow-up is based on the injection of bandwidth-limited phase noise into the beam phase loop which locks the main RF system operating at 200 MHz to the bunch phases. A noise signal with a bandwidth-limited excitation spectrum is needed. The spectrum, as well as the bandwidth, is defined by the cutoff frequencies that follow the variation of the small-amplitude synchrotron frequency, f_{s0}, during the acceleration ramp. The low and high cutoff frequencies are named f_{low}, and f_{high} respectively. By normalizing those values with respect to f_{s0}, the ratios called "margin low" $m_{low} = f_{low}/f_{s0}$ and "margin high" $m_{high} = f_{high}/f_{s0}$ are defined. Figure 1 illustrates the relation between f_{s0}, the frequency band, and the emittance.

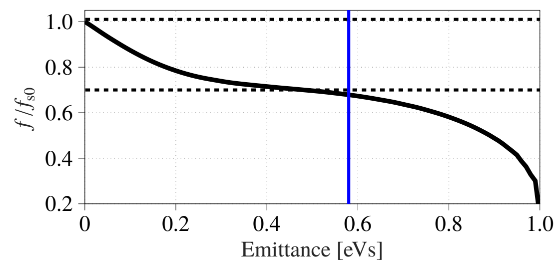


Figure 1: Example of a normalized synchrotron frequency distribution (black) in a double-harmonic RF system. The horizontal dashed lines indicate m_{low} and m_{high} respectively. The vertical blue line marks the longitudinal emittance.

These normalized settings are easier to manage for tuning the noise bandwidth since their value remains constant despite the changes of f_{s0} during acceleration. The aim of the blow-up is to impact the bunch core exclusively, without increasing the tail population of the particle distribution, which would risk generating losses. In addition, the effects on the longitudinal beam profiles are also dependent on the blow-up amplitude, a, defined in rms degrees of the 200 MHz RF system. A low-amplitude noise signal may be ineffective, while too high amplitude can negatively affect the bunch distribution. An additional parameter is the time interval during the cycle the RF manipulation is applied.

Fine-tuning the blow-up settings is challenging since multiple time-dependent settings are involved. The manual procedures are time-intensive and cannot guarantee optimal noise settings. Moreover, even optimal settings necessitate revaluation when parameter changes occur, e.g., higher bunch intensity or a different voltage program. To simplify the optimization, efforts have been recently dedicated to automating the setup of the noise for the controlled blow-up in the SPS. Initial studies in this direction were outlined in [3], where the automatic control of the blow-up was demonstrated for single-bunch beams. The software is integrated as part of the CERN Machine Learning (ML) platform [4], which provides pre-implemented generic optimization algorithms. The advantage of relying on this framework lies in its ambition to integrate numerical optimization, machine learning, and reinforcement learning into routine accelerator operation. The extension of the software for single- to multi-bunch beams has been straightforward; the management of the settings did not change, as well as obtaining the observations. However, a more careful reassessment was necessary for the cost function, which must take into consideration

* niky.bruchon@cern.ch

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TUPDP086

multiple bunches at the same time.

This paper presents the studies undertaken to develop and release an operational tool for the automatic setup of controlled longitudinal emittance blow-up for the SPS. The problem is introduced by highlighting the main aspects of the optimization loop required. The efficient approach of splitting the problem into sub-intervals and sequentially optimizing the blow-up settings is described. Some examples of optimization processes are shown thereafter. Moreover, the core of the operational tool was employed for some preliminary tests of Bayesian optimization.

OVERVIEW OF SPS AUTOMATIC BLOW-UP TUNING

The beam parameters at SPS extraction, e.g. the bunch lengths, are tightly specified for LHC-type beams. The quality of the extracted beam depends strongly on the blow-up settings, to achieve beam stability during acceleration and optimum bunch lengths for transfer to the LHC.

The effects of this RF manipulation, essentially the forced diffusion of particles with synchrotron frequencies within the bandwidth corresponding to the applied band-limited noise, are quantified through measurements of the longitudinal profile. One main parameter in the longitudinal plane to observe the blow-up effects is the bunch length. A black box approach is followed for the automatic setup of the controlled longitudinal emittance blow-up. The noise settings, i.e. a, m_{low}, and m_{high}, which are functions of the cycle time, are treated as inputs, while the resulting bunch length observations are the output.

The optimizer changes the noise settings by observing the bunch length at certain times in the cycle. The optimization loop is sketched in Fig. 2.

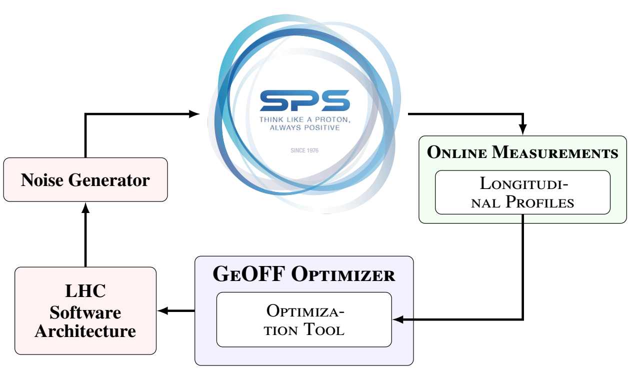


Figure 2: Optimization loop: the measured longitudinal beam profiles are used by the optimizer to calculate the bunch lengths, and subsequently adjust the blow-up by tuning the amplitude and frequency band via the LHC Software Architecture (LSA).

Noise Generation

The noise generation software package is detailed in [5]. The settings a, m_{low}, and m_{high} for the noise generator are stored in the LHC Software Architecture (LSA) settings database. It generates bandwidth-limited noise between m_{low} and m_{high}, with amplitude a, and a desired duration.

Online Measurements

The online longitudinal profiles are acquired throughout the acceleration cycle (generally every 20 ms). The length of the bunches is extracted by means of a Full-Width at Half-Maximum (FWHM) algorithm that serves as a metric to evaluate the effect of the applied blow-up noise, and to define the beam as stable by analyzing the bunch length spread among different bunches.

Optimization Tool

To achieve optimal noise settings, the code is developed relying on the Common Optimization Interfaces (COI) [6]. The Generic Optimisation Frontend and Framework (GeOFF [4], a graphical application) collects interfaces to the accelerators and simulations thereof, and numerical optimizers and reinforcement learners. Developed with COI requirements in mind, the operational tool interfaces with both input and output streams. On one side, it communicates with LSA, enabling the manipulation of the optimizable functions. On the other, it subscribes to the aforementioned real-time observations from the SPS to obtain the profiles.

KEY ASPECTS OF THE OPERATIONAL TOOL

Parameters for Operations

Parameters of the optimizer are easily accessible in LSA via the tool to check and quickly adjust them. A modular approach is followed: the functions to optimize are selectable (i.e. one can enable or disable a, and similarly m_{low} and m_{high}). Selected settings are scanned simultaneously within their ranges defined by minimum and maximum values. To simplify the optimization problem, the noise for the blow-up is injected when the voltage of the main 200 MHz RF system, the voltage ratio of the fourth harmonic RF system at 800 MHz, and the synchronous phase, are almost constant. Keeping a fixed bunch length along the selected interval, the stability and self-consistency of the problem are guaranteed [7, 8]. The initial guess for the frequency band (m_{low} and m_{high}) is provided based on the desired bunch length, while the amplitude of the noise is found by the optimizer. To facilitate the task of reaching and maintaining the target bunch length, the optimization can be distributed over multiple sub-intervals instead of considering the entire blow-up window. A more precise control of the bunch length is thus achieved without increasing the complexity of the problem.

Some degree of flexibility is achieved by allowing a tolerance around the target bunch length. The bunch length at the end of the injection plateau, before the blow-up starts, is verified to avoid optimizing bad-quality beams.

Objective Function

The operational tool for the blow-up optimization is designed for multi-bunch high-intensity LHC-type beams. Typ-

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TUPDP086

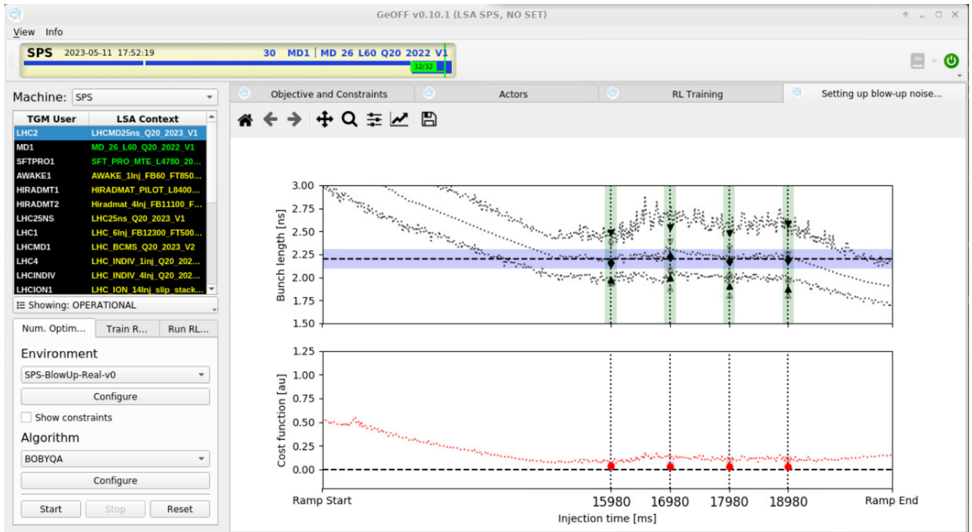


Figure 3: GeOFF application panel after a successful scan over four sub-intervals.

ically, these beams consist of up to four trains of 72 bunches with 25 ns spacing. Therefore, an objective function able to manage information from multiple bunches is required. The Root Mean Squared Error (RMSE) is taken as a representative cost function. The longitudinal profiles are acquired, and, for each bunch detected, four times the standard deviation of the equivalent Gaussian profile is calculated from the measured FWHM bunch length. Given n detected bunches, the bunch lengths collected are $= [\sigma_1, \sigma_2, ..., \sigma_n]$. The normalized error vector is $= /\sigma^* - 1$, where σ^* is the target bunch length. The RMSE is calculated according to Eq. 1, neglecting errors within the tolerance:

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{\varepsilon_i^2}{n}}\,. \tag{1}$$

Each time the optimizer tunes the settings for a certain sub-interval, the effects on the beam are detected by the cost function which considers the profile acquired at the end of the same sub-interval. The minimum of the cost function, zero, is achieved when the bunch length of all bunches satisfies the tolerance requirements.

Sequential Optimization

The optimization of the settings for a single sub-interval, leaving the others untouched, reduces the chance of beam instabilities, losses, and un-programmed beam dumps if settings are not optimal. This results in avoiding missing observations for the optimizer (flat beam profiles). The advantage provided by splitting the blow-up interval is even emphasized by the sequential optimization of the sub-intervals. After finding the optimal settings for a sub-interval, the optimizer moves to the next one. The optimized settings reduce the error between the measured bunch lengths and the target one, returning an almost constant bunch length along the entire interval. The disadvantage is a slower overall optimization. Thus, a compromise between the number of sub-intervals and the total optimization time must be found.

RESULTS IN OPERATIONS

The GeOFF application panel is shown in Fig. 3 after a test scan over four sub-intervals. On the left side, selectable items are available. These are the accelerator, the timing user, the problem to face, and the optimization algorithm. The list of optimizers consists in Bound Optimization BY Quadratic Approximation (BOBYQA, [9], default), Constrained Optimization BY Linear Approximation [10], Nelder Mead [11], and Powell's conjugate direct method [12]. The live plots are shown on the right to follow the optimization in real time. The evolution of the bunch length (upper plot in Fig. 3) and the cost (lower plot in Fig. 3) cover the energy ramp. Markers highlight the values at the end of each sub-interval.

A selection of two tests carried out to validate the tool release is presented in the following. A short list (not exhaustive) of the configurations used for the scans is reported in Table 1. While setting ranges are: $a \in [0.00, 1.00]$, $m_{\text{low}} \in [0.60, 0.70]$, and $m_{\text{high}} \in [0.95, 1.05]$.

Table 1: Setup for the optimization scans. The beam injection time is 0 s and the ramp extends from 11.091 s to 20.739 s.

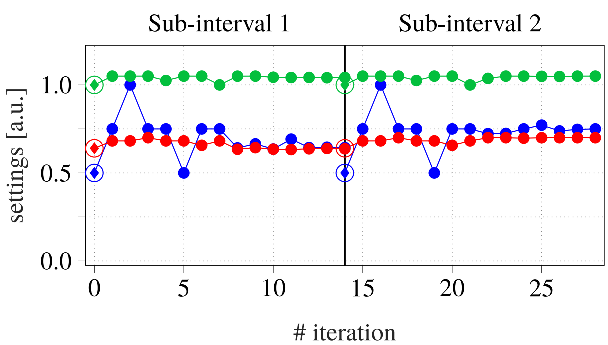
Configuration	**Value**
Start time	15 s from injection
Stop time	19 s from injection
Tolerance	5%
Intensity per bunch	$2.15 \cdot 10^{11}$ ppb
Number of expected bunches	$4 \times 72 = 288$

Scan of Three Settings on Two Sub-Intervals

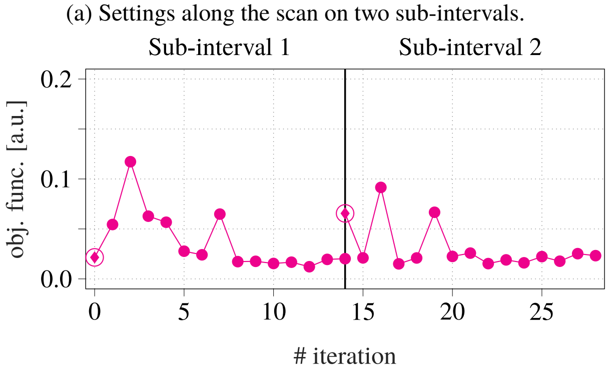
In the first test, the three parameters (a, m_{low}, and m_{high}) were optimized in two sub-intervals with a target bunch

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TUPDP086

length of 2.1 ns. The evolution of the settings and the cost function along the optimization is shown in Fig. 4. For each sub-interval, the initial settings, as well as the corresponding cost, are highlighted by the diamond marker inside a circle. The evolution followed during the optimization follows, according to the iteration number. The trend of the cost function shows a reduction of the distance between the calculated and the target bunch lengths in both sub-intervals. The total number of iterations required is 28, 14 for each sub-interval.



(a) Settings along the scan on two sub-intervals.



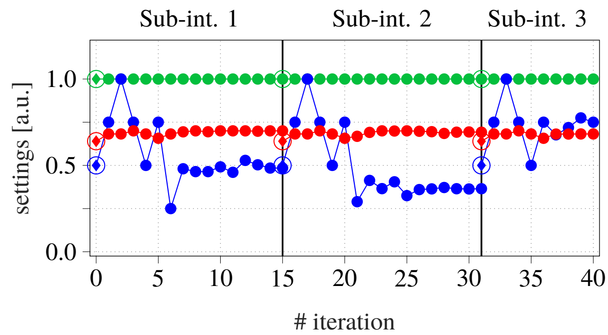
(b) Cost along the scan on two sub-intervals.

Figure 4: Automatic blow-up optimization for two sub-intervals and three functions: amplitude (blue), margin low (red), margin high (green), and objective function (magenta).

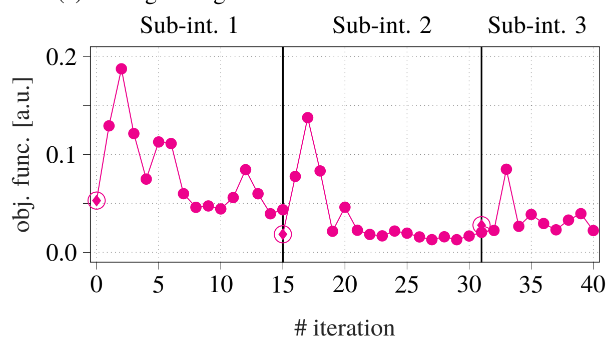
Scan of Two Settings on Three Sub-Intervals

The second scan features the optimization of two parameters (a and m_{low}) during three sub-intervals. This choice combines time resolution and efficiency. The target bunch length is set to 2.0 ns since a lower bunch length at extraction was required. Settings and costs collected throughout the optimization process are shown in Fig. 5, similarly to the previous example in Fig. 4. A significant improvement is visible in each sub-interval, even if the most visible one is achieved in the first sub-interval. The total number of steps for solving the problem is 40. The first sub-interval takes 15, the second 16, and the last only 9.

In this second example, it is clear that increasing the number of sub-intervals leads to a rise in the total number of iteration steps. However, more sub-intervals ensure a better result in keeping the desired bunch length flat during the



(a) Settings along the scan with three sub-intervals.



(b) Cost along the scan with three sub-intervals.

Figure 5: Automatic blow-up optimization for three sub-intervals and two functions: amplitude (blue), margin low (red), margin high (green), and objective function (magenta).

entire blow-up interval. The trade-off between the time required for tuning the noise settings and keeping the bunch length as flat as possible is necessary. With two and three sub-intervals the results are satisfactory, i.e. the time required is acceptable as well as the bunch length evolution along the blow-up interval. For a quicker scan, a single sub-interval can be considered if a constant bunch length during the blow-up is not required. Tests on four sub-intervals were also performed, but the longer optimization time was penalizing excessively, for a limited improvement in the bunch length trend.

ADVANCED STUDIES

By integrating the tool into the environment of the CERN ML platform, the extension of the study to learning-based algorithms has been straightforward. However, learning methods require a sufficient amount of data for the training. The available data is not sufficient to fully train the optimizer. Moreover, acquiring enough data to cover all settings ranges is very time-demanding. A similar problem is encountered in simulations: the amount of simulated data increases together with the number of sub-intervals. Therefore, a sample-efficient learning method is needed.

Preliminary Results with Bayesian Optimization

Bayesian optimization is a technique employed to optimize complex and expensive objective functions, often en-

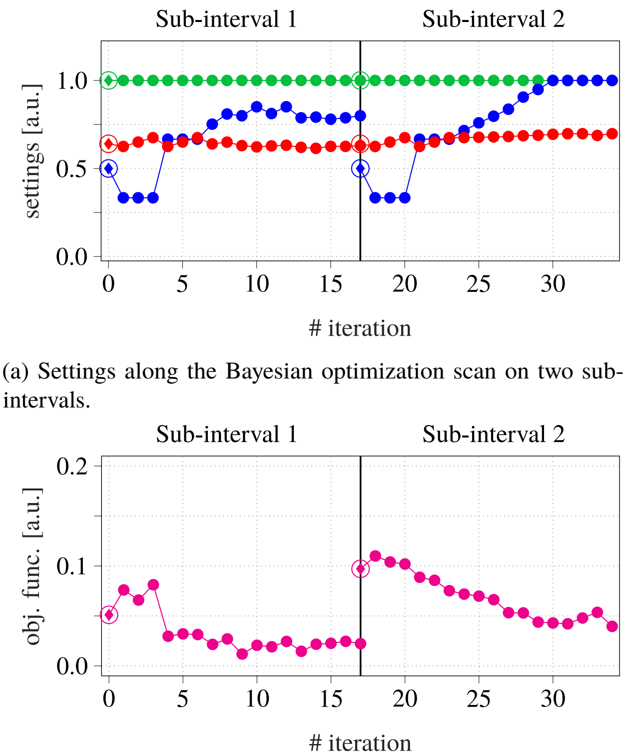
ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TUPDP086

countered in particle accelerators [13–15]. It combines probabilistic modeling with acquisition functions to efficiently search for the optimal solution while minimizing the number of function evaluations. BoTorch [16], a state-of-the-art library built on top of PyTorch, offers a comprehensive and well-benchmarked toolkit for Bayesian optimization. As with other learning approaches, the optimization requires an exploration-exploitation trade-off. Indeed, if the exploration phase aims to select samples that adequately cover the search space as much as possible, the exploitation phase focuses on selecting samples close to the best value encountered.

A two-parameter scan (m_{low} and a) was performed with a beam variant with 56-bunch trains and $1.9 \cdot 10^{11}$ protons per bunch (8b4e, mini-batches of eight bunches followed by gaps of four empty bunch positions). The approach followed in tuning the blow-up considers the total number of bunches. The empty bunch positions and the number of batches can be neglected. The start and the end of the blow-up interval, as well as the percentage tolerance, are as in Table 1. The target bunch length is 2.0 ns.

The exploration is performed over a set of six default settings common to both the sub-intervals. To ensure the coverage of the settings space, the training set is defined through a grid search within the settings range. In a fixed number of steps (ten) the model is exploited and improved to optimize the blow-up. The total number of iterations is therefore fixed to 34. The two additional iterations consist of putting the optimal settings at the end of the scan of each sub-interval. The results are summarized in Fig. 6.

The disadvantage of the Bayesian optimization relies on the larger number of steps required. This could be overcome by re-using data from previous optimizations to reduce the exploration phase and defining convergence criteria to stop the exploitation phase.



(a) Settings along the Bayesian optimization scan on two sub-intervals.

(b) Cost along the Bayesian optimization scan on two sub-intervals.

Figure 6: Bayesian optimization of the controlled blow-up for two sub-intervals and two functions: amplitude (blue), margin low (red), margin high (green), and objective function (magenta).

CONCLUSIONS

An operational tool to set up the controlled longitudinal emittance blow-up in the SPS was developed and successfully run during machine development (MD) studies with high-intensity multi-bunch LHC beams. The proposed study extends the previous investigation on single-bunch beams to multi-bunch beams by defining and implementing a simpler cost function and an *ad hoc* approach to the problem. A precise control along the acceleration ramp was obtained by maintaining a low problem complexity and repeating the scan for the desired sub-intervals thanks to the split and optimize approach. Moreover, the configurations are customizable via LSA and also easily adjustable directly from the tool, to adapt to different acceleration cycles and beam parameters. SPS operators can now profit from a robust and reliable application to automatically set up the controlled longitudinal emittance blow-up. Advanced studies were started to integrate learning-based approaches. Promising results were obtained by applying Bayesian optimization. Despite the slightly higher number of iterations required, the investigation of learning methods will continue to improve their efficiency.

REFERENCES

[1] T. Bohl, T. Linnecar, E. Shaposhnikova, and J. Tückmantel, “Study of Different Operating Modes of the 4th RF Harmonic Landau Damping System in the CERN SPS”, in *Proc. EPAC’98*, Stockholm, Sweden, Jun. 1998, paper THP09A, pp. 978–980.

[2] G. Papotti, T. Bohl, T. P. R. Linnecar, E. N. Shaposhnikova, and J. Tückmantel, “Study of Controlled Longitudinal Emittance Blow-up for High Intensity LHC Beams in the CERN SPS”, in *Proc. EPAC’08*, Genoa, Italy, Jun. 2008, paper TUPP059, pp. 1676–1678.

[3] N. Bruchon, I. Karpov, N. Madysa, G. Papotti, D. Quartullo, and C. Zisou, “Towards the Automatic Setup of Longitudinal Emittance Blow-Up in the CERN SPS”, in *Proc. IPAC’22*, Bangkok, Thailand, Jun. 2022, pp. 949–952. `doi:10.18429/JACoW-IPAC2022-TUPOST042`

[4] J-B. De Martel, and N. Madysa, “ML for Accelerator Controls: Machine Learning Platform and Generic Optimization Framework and Frontend”, 2022, `https://indico.cern.ch/event/1175862/`

[5] J. Tückmantel, “Digital Generation of Noise-signals with Arbitrary Constant or Time-varying Spectra”, in *Proc. EPAC’08*, Genoa, Italy, Jun. 2008, paper TUPC103, pp. 1299–1301.

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TUPDP086

[6] Common Optimization Interfaces, https://cernml-coi.docs.cern.ch/

[7] I. Karpov, "Longitudinal settings and stability", https://indico.cern.ch/event/1160070/contributions/4871893/subcontributions/380904/attachments/2469579/4246992/IPP_LIU_Ramp_Up_24062022_final.pdf

[8] D. Quartullo, L. Intelisano, I. Karpov, and G. Papotti, "Simulation Studies of Longitudinal Stability for High-Intensity LHC-Type Beams in the CERN SPS", in *Proc. IPAC'22*, Bangkok, Thailand, Jun. 2022, pp. 2249–2252. doi:10.18429/JACoW-IPAC2022-WEPOMS009

[9] M. J. D. Powell, "The BOBYQA algorithm for bound constrained optimization without derivatives", University of Cambridge, Cambridge, Rep. DAMTP 2009/NA06, 2009.

[10] M. J. D. Powell, "A direct search optimization method that models the objective and constraint functions by linear interpolation", in *Advances in Optimization and Numerical Analysis*, Dordrecht, Netherlands: Springer, 1994, pp. 51–67. doi:10.1007/978-94-015-8330-5_4

[11] J. A. Nelder, R. Mead, "A simplex method for function minimization", *Comput. J.*, vol. 7, no. 4, pp. 308–313, 1965. doi:10.1093/comjnl/7.4.308

[12] M. J. D. Powell, "Direct search algorithms for optimization calculations", *Acta Numer.*, vol. 7, pp. 287–336, 1998. doi:10.1017/S0962492900002841

[13] R. Roussel, A. Hanuka, and A. Edelen, "Multiobjective Bayesian optimization for online accelerator tuning", *Phys. Rev. Accel. Beams*, vol. 24, no. 6, p. 062801, 2021. doi:10.1103/PhysRevAccelBeams.24.062801

[14] J. Duris, D. Kennedy, A. Hanuka, *et al.*, "Bayesian optimization of a free-electron laser", *Phys. Rev. Lett.*, vol. 124, no. 12, p. 124801, 2020. doi:10.1103/PhysRevLett.124.124801

[15] R. J. Shalloo *et al.*, "Automation and control of laser wakefield accelerators using Bayesian optimization", *Nat. Commun.*, vol. 11, p. 6355, 2020. doi:10.1038/s41467-020-20245-6

[16] M. Balandat, B. Karrer, D. Jiang, S. Daulton, B. Letham, A.G. Wilson, and E. Bakshy, "BoTorch: A framework for efficient Monte-Carlo Bayesian optimization", in *Adv. Neural Inf. Process. Syst. 33 (NeurIPS 2020)*, 2020, pp. 21524–21538.