Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 56-2 (2023) 3079–3085

A Safe Bayesian Optimization Algorithm for Tuning the Optical Synchronization System at European XFEL ⋆

Jannis O. Lübsen * **Maximilian Schütte** **,* **Sebastian Schulz** **
Annika Eichler *,**

* *Hamburg University of Technology, Institute of Control Systems, Eißendorfer Straße 40 21073 Hamburg, Germany, (e-mail: jannis.luebsen@tuhh.de)*

** *Deutsches Elektronen-Synchrotron DESY, Notkestr. 85 22607 Hamburg, Germany (e-mail: maximilian.schuette@desy.de).*

Abstract: The European XFEL is one of the largest linear particle accelerator in the world used to generate extremely intense and ultra short X-Ray flashes to study ultra fast, time resolved chemical processes. In order to improve the quality of the observations, the laser-based optical synchronization system is optimized by tuning PI controllers using a safe Bayesian optimization approach. Since machine time on the European XFEL is very expensive, the algorithm needs to find the optimal parameters as fast as possible. In this contribution, we present a safe Bayesian optimization algorithm which, while guaranteeing safety, shows significantly improved convergence speed and noise robustness. Application and comparison results are presented in simulation for the optical synchronization system of the European XFEL and an experimental demonstration is performed for a laboratory synchronization system.

Keywords: Bayesian optimization; Machine learning in modelling, prediction, control and automation; Learning for control; European XFEL; Particle accelerator.

1. INTRODUCTION

Bayesian optimization (BO) is a learning based, gradient-free and global optimization method which gained attention in recent years. The method is very suitable for expensive to evaluate functions and robust to noisy observations. As schematically illustrated in Figure 1, BO is used to optimize systems with unknown dynamics represented by a black box. A limiting factor of BO is the dimension of the input space, as it is practically feasible up to 10 to 20 dimensions. Several works deal with the feasibility problem for high-dimensional inputs, see Kirschner et al. (2019) and Moriconi et al. (2020). Kirschner et al. (2019) propose to divide the overall problem into sub-problems to which BO is applied. Many real-world optimization problems require the consideration of constraints, as safety critical systems. The constraints are declared to avoid the evaluation of inputs which invoke undesirable effects, e. g., damaging the system. A widely known approach, which can guarantee safety constraints, is `SafeOpt` introduced by Sui et al. (2015). BO algorithms that are equipped with those constraints are denoted as safe BO. Recently, the algorithm was further developed in several publications, e. g., Sui et al. (2018) and Berkenkamp and Schoellig (2021). Safe BO algorithms were already successfully applied on tuning different safety critical systems, however, mostly in simulations as for Schillinger et al. (2017) and Khosravi et al. (2019).

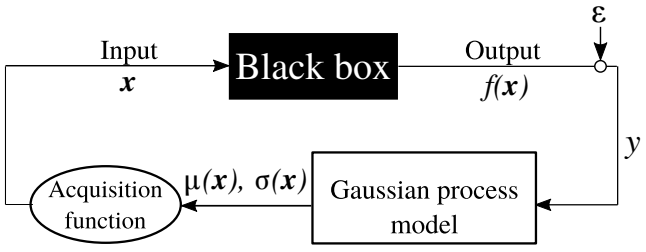


Fig. 1. Schematic illustration of Bayesian optimization.

An example for a safety critical system is the laser-based optical synchronization (LbSync) system at the European X-Ray Free-Electron Laser (XFEL) depict in Figure 2. The research at the European XFEL involves studies across many disciplines, e. g., watching ultra fast energy transfers or observing the behavior of electrons within molecules. In order to get high quality observations of these ultra fast time resolved processes, an excellent synchronization between the pump-probe laser (PPL) pulses and the X-Rays from the accelerator is required. Therefore, an LbSync system has been implemented at the European XFEL. With this system an ultra precise timing reference is distributed along 3.4 km via lasers and length stabilized links. Lasers and links are locally controlled via a PI controller. To reach the desirable timing sta-

⋆ The authors acknowledge support from Deutsches Elektronen-Synchrotron DESY Hamburg, Germany, a member of the Helmholtz Association HGF.

10.1016/j.ifacol.2023.10.1438

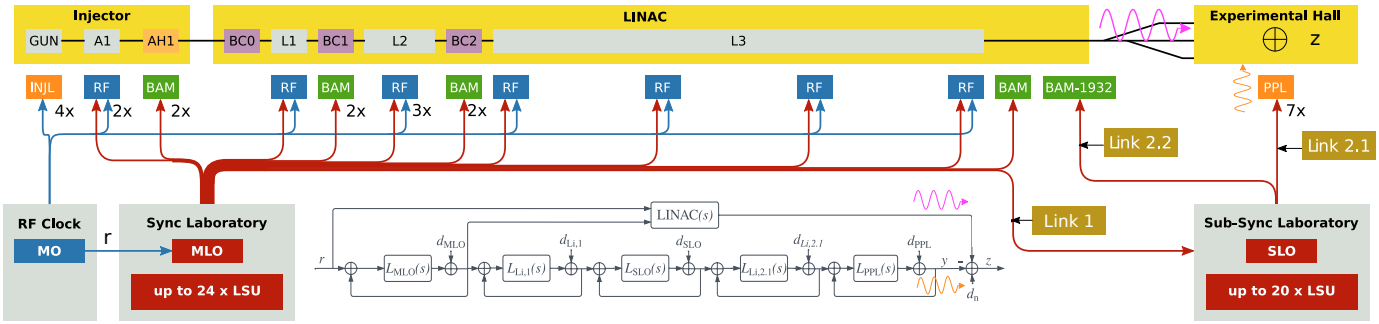


Fig. 2. Main building blocks of the laser-based optical synchronization system with the corresponding control scheme. $L_\bullet(s)$ denote the open-loop transfer functions of the particular subsystems.

bility in the femto-second range, the control parameter of the optical synchronization system need to be optimally tuned. In Heuer (2018) and Schütte et al. (2021), local and global white-box approaches have been presented for optimal controller tuning. As these methods can not easily adopt to system changes and system identification of such a complex system is very time consuming an alternative black-box approach using safe BO, denoted as modified safe optimization (`MoSaOpt`), is presented based on the formulations of Kirschner et al. (2019), Sui et al. (2015) and Berkenkamp et al. (2016). In contrast to these works, in `MoSaOpt`, we split exploration and exploitation phases and apply a new decision scheme to select the best observation. While phase splitting is also proposed in Sui et al. (2018), we use a computationally significantly cheaper implementation for exploration. Furthermore, we adapt the hyperparameters in exploitation phase. Our algorithm shows significantly improved convergence speed and noise robustness compared to existing algorithms as demonstrated in application, while guaranteeing safety. Thus, it is perfectly suited for the tuning of the European XFEL LbSync system as the tuning time is very costly and the signal-to-noise ratio is low. The performance of `MoSaOpt` is demonstrated in simulation for the European XFEL LbSync system as well as experimentally on a small-scale laboratory synchronization system, located at Deutsches Elektronen-Synchrotron (DESY) in comparison to `SafeOpt` and Nelder-Mead, respectively.

In Section 2, with BO and the LbSync system the preliminaries are introduced. The main contribution, the `MoSaOpt` algorithm, is presented in Section 3. In Section 4, `MoSaOpt` is compared to `SafeOpt` in simulation and experimentally to Nelder-Mead at a laboratory optical synchronization system.

2. FUNDAMENTALS

2.1 Optical Synchronization System

A schematic overview of the main building blocks of the LbSync system at the European XFEL is illustrated in Figure 2. In this work, we consider the optimization of the direct path from master laser oscillator (MLO) to PPL. The chain starts at the master timing reference oscillator (MO), which provides an RF signal to the MLO and is considered as the source of the reference signal. The MLO replicates the signal by emitting extremely short pulses of light in sync to the phase of the RF signal of the MO, which are distributed on optical fibers to the end-stations. This means the laser pulses of the MLO are distributed through the 3 km long *Link 1* to the subsidiary laser oscillator (SLO), where the signals are then forwarded through *Link 2.1* to the PPL. As such, there are in total two links and three lasers to be considered. Each device is exposed to disturbances as temperature changes, humidity and mechanical vibrations which cause length variations of the fibers and detune the laser cavities, resulting in shifted arrival times and a poor synchronization. In order to counteract the disturbances, all devices are stabilized by feedback loops, using piezoelectric actuators which are regulated by PI controllers. Therefore, there are in total ten tunable parameters to be jointly optimized towards a common goal, which is optimal replication of the reference timing at the end of the synchronization chain, i. e., minimization of the RMS error between the phase of the PPL oscillator and the X-rays generated by the accelerator. This RMS error is referred to as *integrated timing jitter*, denoted J and is computed over 100 ms intervals.

2.2 Bayesian Optimization

In Bayesian optimization an unknown system, represented by a black box, is optimized iteratively. The transition from inputs to outputs is described by an unknown objective function $f : \mathcal{X} \longrightarrow \mathbb{R}$, where $\mathcal{X} \subseteq \mathbb{R}^d$ describes a compact set of input parameters. A Gaussian process model (GPM) is used to model the objective function which is fully defined by a mean function $m(\boldsymbol{x})$ and a kernel $k(\boldsymbol{x}, \boldsymbol{x}')$. In order to ensure applicability of BO we need to impose an assumption on the objective function as shown in Williams and Rasmussen (2006) and Kanagawa et al. (2018).

Assumption 1. The function f is a member of a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ defined by a positive definite kernel $k : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}$, $k(\cdot, x) \in \mathcal{H}$ such that $f(\boldsymbol{x}) = \langle f(\cdot), k(\cdot, \boldsymbol{x}) \rangle_{\mathcal{H}}$ and $\|f\|_{\mathcal{H}} \leq B$.

This assumption is necessary to guarantee the capability of the GPM to model the objective. Depending on the kernel, the RKHS space can be extremely large including a wide range of functions which makes this assumption nonrestrictive. In general, f is a non-convex function which is learned online by evaluating the function at some inputs $\boldsymbol{x} \in \mathcal{X}$. The kernel determines the dependence between function values at different inputs $\mathrm{COV}(f(\boldsymbol{x}), f(\boldsymbol{x}')) = k(\boldsymbol{x}, \boldsymbol{x}')$. Commonly used kernels are the Matérn or the squared exponential kernel, where the latter is defined as

$$k_{\mathrm{SE}}(\boldsymbol{x},\boldsymbol{x}') = \sigma_f^2 \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}')^T\Delta^{-2}(\boldsymbol{x}-\boldsymbol{x}')\right) \quad (1)$$

with $\Delta = \mathrm{diag}[\boldsymbol{l}] = \mathrm{diag}[l_1,\dots,l_d]$. The function values themselves are not accessible, rather noisy observations are made by measuring the output. As illustrated in Figure 1, the resulting measurement uncertainty can be modeled by additive Gaussian noise $\epsilon \sim \mathcal{N}(0,\sigma_n^2)$, such that

$$y = f(\boldsymbol{x}) + \epsilon, \quad (2)$$

where σ_n^2 denotes the noise variance. The signal variance σ_f^2, the length scales $\boldsymbol{l}$ from (1) together with σ_n^2 are the so called hyperparameters $\boldsymbol{\theta}$ and can be used to adjust the covariance. The set of n observations is defined by $\mathcal{O} = \{(\boldsymbol{x}_k, y_k), k = 1\dots n\}$ and is composed of the evaluated inputs combined with the corresponding observations. The compact notation of all inputs of $\mathcal{O}$ is given by the matrix $X = [\boldsymbol{x}_1,\dots,\boldsymbol{x}_n]$ and of all observations by the vector $\boldsymbol{y} = [y_1,\dots,y_n]^T$. With the GPM, the posterior distribution $p(f_*|\mathcal{O},\boldsymbol{x}_*,\boldsymbol{\theta})$ for $f_* \widehat{=} f(\boldsymbol{x}_*)$ at test point $\boldsymbol{x}_* \in \mathcal{X}$ is determined. Defining the matrix with entries $[K(Z,Z')]_{(i,j)} = k(\boldsymbol{z}_i,\boldsymbol{z}'_j)$ where $Z \in \mathbb{R}^{d\times m}$, $Z' \in \mathbb{R}^{d\times k}$ and $\boldsymbol{z}_i,\boldsymbol{z}'_j \in \mathbb{R}^d$, and substitute $K = K(X,X)$, $K_* = K(\boldsymbol{x}_*,\boldsymbol{x}_*)$, then, the posterior mean is defined as

$$\begin{aligned}\mu(\boldsymbol{x}_*) &= \mathbb{E}\left[p(f_*|\mathcal{O},\boldsymbol{x}_*,\boldsymbol{\theta})\right] \\ &= K(\boldsymbol{x}_*,X)\left(K+\sigma_n^2 I\right)^{-1}\boldsymbol{y}\end{aligned} \quad (3)$$

and the posterior variance as

$$\begin{aligned}\sigma^2(\boldsymbol{x}_*) &= \mathrm{Var}\left[p(f_*|\mathcal{O},\boldsymbol{x}_*,\boldsymbol{\theta})\right] \\ &= K_* - K(\boldsymbol{x}_*,X)\left(K+\sigma_n^2 I\right)^{-1}K(X,\boldsymbol{x}_*).\end{aligned} \quad (4)$$

Finally, the loop in Figure 1 is closed by the acquisition function, which determines new promising inputs from the predictive distribution. A common choice is expected improvement (EI) as initially described in Jones et al. (1998). This acquisition function determines the expectation that the function at point $\boldsymbol{x}$ improves the so far best observation f_{opt}, $\mathrm{EI}(\boldsymbol{x}) := \mathbb{E}\left[\max(f_{\mathrm{opt}} - f(\boldsymbol{x}),0)\right]$. For numerical evaluation we need to rewrite the expression into integral form. Using integration by parts gives

$$\mathrm{EI}(\boldsymbol{x}) = (f_{\mathrm{opt}} - \mu(\boldsymbol{x}))\cdot\Phi_s(\lambda) + \sigma(\boldsymbol{x})\cdot\phi_s(\lambda), \quad (5)$$

where $\lambda = \left(\frac{f_{\mathrm{opt}}-\mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})}\right)$, ϕ_s denotes the standard normal probability density function and Φ_s represents the standard cumulative distribution function. Two other important functions are upper confidence bound (UCB) $u(\boldsymbol{x})$ and lower confidence bound (LCB) $l(\boldsymbol{x})$ given by

$$\begin{aligned}u(\boldsymbol{x}) &= \mu(\boldsymbol{x}) + \beta\sigma(\boldsymbol{x}), \\ l(\boldsymbol{x}) &= \mu(\boldsymbol{x}) - \beta\sigma(\boldsymbol{x}).\end{aligned} \quad (6)$$

The confidence β acts as trade-off between exploration and exploitation. Exploration means that points with high uncertainty are evaluated, while exploitation means that the function is optimized within a known space. UCB and LCB will be used later to define safe inputs as described in Section 3.

This naive BO implementation can be extended to consider constraints $g : \mathcal{X} \longrightarrow \mathbb{R}$ on the input space. We define a constrained optimization problem as

$$\min_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x}) \qquad \text{s. t.} \qquad g(\boldsymbol{x}) \leq 0.$$

One approach to include constraints is shown in Sui et al. (2015) and Berkenkamp et al. (2016). Based on these works, in Section 3 a new algorithm is introduced. For high-dimensional problems, i. e., $d >> 1$, we face several challenges. First, the current formulation of the constraints is numerically feasible up to $d = 3$, as the whole input space needs to be considered. This will become clear in the next section. Another major limitation is the search for new promising inputs with an acquisition function α. New inputs are defined by $\boldsymbol{x}_{\mathrm{new}} = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \alpha(\boldsymbol{x})$, which corresponds to a global optimization of a non-convex function across the whole input space. Moriconi et al. (2020) deems this operation is feasible up to problem dimension $\dim(\mathcal{X}) = 10,\dots,20$.

One approach to address both two issues is to divide the overall optimization problem into several sub-problems which are solved successively as shown in Kirschner et al. (2019). Initially, a direction oracle defines new directions $r_i \in \mathrm{span}\{\mathcal{X}\}$. Commonly, a random oracle is utilized. Then, the subspace $\mathcal{L} \subset \mathcal{X}$ is constructed and BO is applied on the subspace. The optimization of α can now be achieved via, e. g., grid-based methods as $\mathcal{L}$ is a low-dimensional space. In the following, we refer to `LineBO` when $\dim(\mathcal{L}) = 1$ and to `PlaneBO` when $\dim(\mathcal{L}) = 2$ which correspond to a line and plane search, respectively.

3. MODIFIED SAFE BAYESIAN OPTIMIZATION

In the following, the safe BO method introduced by Berkenkamp et al. (2016), `SafeOpt`, is formulated in terms of a minimization to present the differences to the proposed algorithm. In `SafeOpt`, the input domain is divided into three subsets, the safe set $\mathcal{S}$, the expander set $\mathcal{G}$ and the minimizer set $\mathcal{M}$. The safe set at iteration i is defined as

$$\mathcal{S}_i = \{\boldsymbol{x} \in \mathcal{X} | u(\boldsymbol{x}) \leq T\},$$

and depends on the UCB as well as on the safety threshold T that should never be exceeded. The confidence is given by

$$\beta_i = \left(2B + 300\gamma_i \log^3\left(\frac{i}{\delta}\right)\right)^{1/2}, \quad (7)$$

where B is a upper bound on the RKHS norm of f, γ_i the maximal mutual information of the prior GPM and observations, and δ the allowed failure probability. Note that β is updated every iteration; see Sui et al. (2015) for further details. Berkenkamp and Schoellig (2021) mention that this selection of the confidence may be too conservative. An alternative is to set β_i constant, $\beta_i = \beta, \forall\, i$, as done in Berkenkamp et al. (2016). Regarding the definition of the expander set, Berkenkamp et al. (2016) provide a formulation without using a Lipschitz constant. The indicator function given as

$$p(\boldsymbol{x}) = |\{\boldsymbol{x}' \in \mathcal{X} \setminus \mathcal{S} | u_{(\boldsymbol{x},l(\boldsymbol{x}))}(\boldsymbol{x}') \leq T\}|$$

determines the number of additional safe set members by adding an artificial pair to the observation set, $(\boldsymbol{x},l(\boldsymbol{x}))\bigcup\mathcal{O}$, assuming optimistically that the measured value lies on the LCB. Then, after updating the posterior distribution, it is checked whether the safe set could be enlarged. If there is an expansion, $\boldsymbol{x}$ is added to the expander set which is defined as $\mathcal{G} = \{\boldsymbol{x} \in \mathcal{S} | p(\boldsymbol{x}) > 0\}$. However, calculating the expander set with this formulation is expensive as the posterior distribution needs to be recalculated for every member of the safe set. As shown in (3) and (4) this involves $|\mathcal{S}|$ inversions of the matrix K.

3.1 Algorithm

Algorithm 1 presents the modified safe optimization algorithm (`MoSaOpt`). The procedure consists of two different states which are entered successively without alternation, i. e., if the algorithm once switches from exploration to exploitation it can never enter exploration again. Due to the fact that f is completely unknown, i. e., there exists no information about the location of the minimum. Thus, the reachable set, $\mathcal{R} = \{\boldsymbol{x} \in \mathcal{X} | f(\boldsymbol{x}) \leq T, \boldsymbol{x}_0 \in \mathcal{S}_0\}$, must be observed. The reachable set is a subset of $\mathcal{X}$ where the constraint is satisfied and depend on the start point $\boldsymbol{x}_0$. In the following, the constraint is assumed to impose an upper bound on the function value, such that

$$g(\boldsymbol{x}) = f(\boldsymbol{x}) - T.$$

If g would be an independent function, an additional GPM needs to be learned for $g(\boldsymbol{x})$ as in Schillinger et al. (2017). To initialize the algorithm, an initial safe set $\mathcal{S}_0$ is required which consists of at least one point that satisfy the constraint. The initial set of observations $\mathcal{O}_0$ can be empty, the confidence β is set constant, and together with $\boldsymbol{\theta}_0$ they are adjusted to fulfill Proposition 2. The outputs of the algorithm are the final set of observations and the optimal solution. The optimization runs until a termination condition is fulfilled. Such a condition could be satisfied if the maximal value of the acquisition function α falls below a threshold or if the maximal number of observations N is reached. Initially, the algorithm is in exploration state, where the safe set is defined in line **(5)** by the UCB. The set of expanders is given as

$$\mathcal{G}_i = \partial \mathcal{S}_i$$

and restricted to the boundary of the safe set, where the inputs with the greatest potential increase of $\mathcal{S}$ are located. Clearly, this requires a conservative selection of the hyperparameters to preserve safeness. The new definition of expanders reduces the computational complexity significantly compared to the formulation in Berkenkamp et al. (2016) as the boundary can be cheaply determined for one or two-dimensional spaces. Remind, that we assume the algorithm to be embedded into `LineBO` or `PlaneBO`. The exploration function in line **(7)**

$$w(\boldsymbol{x}) = u(\boldsymbol{x}) - l(\boldsymbol{x}) = 2\beta\sigma(\boldsymbol{x})$$

selects the expander with the highest uncertainty. Therefore, the expansion of $\mathcal{S}$ is maximized and $\mathcal{R}$ is observed within the least possible number of steps. These steps are repeated until the exploration function is smaller than an exploration threshold ν which indicates that the ν-Reachable set $\mathcal{R}_\nu \subset \mathcal{R}$ is observed.

If the uncertainty of the expanders falls below ν, the algorithm switches to exploitation state. The safe set is frozen and the hyperparameters are adjusted such that the marginal likelihood $p(\boldsymbol{y}|X, \boldsymbol{\theta})$ becomes maximal. In this operation we are searching for a model that can explain the acquired data best with the lowest complexity as described in Williams and Rasmussen (2006). The model complexity means how fast the target function is assumed to vary. The higher the complexity of the model the more functions can be modeled. Then, in line **(13)** the new promising input is selected from the optimized GPM by the acquisition function α. In our implementation we use expected improvement acquisition function given in (5).

Algorithm 1: Modified Safe Optimization (`MoSaOpt`)

Input: Initial safe set $\mathcal{S}_0$, acquisition function $\alpha(\boldsymbol{x})$, safety threshold T, confidence β, initial set of observations $\mathcal{O}_0$, initial hyperparameters $\boldsymbol{\theta}_0$
Output: $\boldsymbol{x}_{\text{opt}}$, $\mathcal{O}$

1 $\mathcal{G}_0 \leftarrow \partial \mathcal{S}_0$
2 $i \leftarrow 1$
3 **while** *termination condition not true* **do**
4 **if** $\nu \leq w_i(\boldsymbol{x}), \boldsymbol{x} \in \mathcal{G}_{i-1}$ **then** // exploration
5 $\mathcal{S}_i \leftarrow \mathcal{S}_{\text{i}-1} \bigcup \{\boldsymbol{x} \in \mathcal{X} | u(\boldsymbol{x}) \leq T\}$
6 $\mathcal{G}_i \leftarrow \partial \mathcal{S}_i$
7 $\boldsymbol{x}_i \leftarrow \arg\max_{\boldsymbol{x} \in \mathcal{G}_i} (w(\boldsymbol{x}))$
8 **else** // exploitation
9 $\boldsymbol{\theta}_i = \arg\max_{\boldsymbol{\theta}} \left(p(\boldsymbol{y}|X, \boldsymbol{\theta}_{i-1})\right)$
10 $\mu_i(\boldsymbol{x}_*) \leftarrow \mathbb{E}\left[p(f_*|\mathcal{O}_i, \boldsymbol{x}_*, \boldsymbol{\theta}_i)\right]$
11 $\sigma_i^2(\boldsymbol{x}_*) \leftarrow \text{Var}\left[p(f_*|\mathcal{O}_i, \boldsymbol{x}_*, \boldsymbol{\theta}_i)\right]$
12 $\mathcal{M}_i \leftarrow \{\boldsymbol{x} \in \mathcal{S}_i | l(\boldsymbol{x}) \leq \Phi(\boldsymbol{y}, X)\}$
13 $\boldsymbol{x}_i \leftarrow \arg\max_{\boldsymbol{x} \in \mathcal{M}_i} (\alpha(\boldsymbol{x}, \mu_i, \sigma_i))$
14 **end**
15 $y_i \leftarrow f(\boldsymbol{x}_i) + \epsilon$ // evaluate function
16 $\mathcal{O}_i \leftarrow \mathcal{O}_{i-1} \bigcup \{\boldsymbol{x}_i, y_i\}$ // update observation set
17 $i \leftarrow i + 1$
18 **end**
19 $\boldsymbol{x}_{\text{opt}} \leftarrow \arg \Phi(\boldsymbol{y}, X)$ // find optimal input

All potential inputs are collected in the minimizer set given as

$$\mathcal{M}_i = \{\boldsymbol{x} \in \mathcal{S}_i | l(\boldsymbol{x}) \leq \Phi(\boldsymbol{y}, X)\}.$$

Minimizers represent inputs that potentially improve the performance, here, indicated by a smaller LCB than the current best observation determined by the decision function $\Phi(\boldsymbol{y}, X)$. After that, f is evaluated at the obtained input in line **(15)** and the set of observations is updated. Finally, when the termination condition is fulfilled, the best solution of the observation set is identified by a decision scheme $\arg \Phi(\boldsymbol{y}, X)$ and returned. The safety critical part of the optimization occurs in exploration state where the ν-Reachable set is discovered. In order to guarantee safeness Proposition 2 needs to be fulfilled.

Proposition 2. Given Assumption 1, constant β and initial hyperparameters $\boldsymbol{\theta}_0$ as well as $k(\cdot, \cdot)$ selected such that $u(\boldsymbol{x}) \geq f(\boldsymbol{x}), \forall \boldsymbol{x} \in \mathcal{X}$, then safeness is guaranteed by Algorithm 1 in the sense that $f(\boldsymbol{x}_i) \leq T, \forall \boldsymbol{x}_i \in \mathcal{O}$.

Proof. Follows directly from the definition of the safe set in line **(5)** of Algorithm 1, evaluations are solely performed at $\boldsymbol{x} \in \mathcal{G} \bigcup \mathcal{M}$ where $(\mathcal{G} \bigcup \mathcal{M}) \subset \mathcal{S}$. □

This proposition requires advanced knowledge of the objective function, e. g., the Lipschitz constant which is for a black box optimization usually unknown. It is part of the future work to reduce the required information to guarantee safety. However, one can also use the safe set definition of Sui et al. (2015) with (7) to obtain safety guarantees with high probability during exploration.

The length scales $\boldsymbol{l}$ define how fast the covariance between neighbouring inputs decrease. The smaller the length

scales, the faster the function values can change. The signal variance σ_f^2 defines the range where the function values are expected. The larger σ_f^2 the higher the expected range of function values. Adjusting these two hyperparamters to realize the assumptions in Proposition 2 can, however, sacrifice the convergence rate. Due to the scheme presented here, we can first explore the ν-Reachable set while guaranteeing safety and iteratively construct a data set containing information about the function in the ν-Reachable set. Secondly, in exploitation safety is guaranteed since the safe set is no longer extended. We use the knowledge about the function given by the acquired data to perform a regression, i. e., optimizing the hyperparamters. In practice, we obtain good approximations of the function (due to the conservatively chosen hyperparameters during exploration) such that the optimization of the function can be efficiently performed with only a few additional observations.

3.2 Practical Considerations

The model complexity is adjusted by fitting the hyperparameters in line **(9)** via maximum likelihood estimation of the marginal likelihood $p(\boldsymbol{y}|X, \boldsymbol{\theta})$. Since the marginal likelihood has multiple optima, one could end up with a bad local optimum with a high complexity, i. e., misinterpreting noise as function varies, when using gradient based methods. In general, the higher the dimension of the problem, the more local optima exist. An intuitive approach is to apply gradient solvers from different, randomly selected starting points. In our implementation, we decided to use a quasi-Newton method with dense Hessian approximation which shows in practice good results. However, with increasing dimension of the input space, restricting the optimization to the hyperparameters of the subspace may lead to better results.

The performance of sequential subspace optimizations rely significantly on the obtained output as following subspaces are based on them. The selection is performed in line **(19)** in Algorithm 1 using a decision function $\Phi : \mathcal{O} \to \mathbb{R}$. A common approach is simply defining

$$\Phi(\boldsymbol{y}, X) = \min_{i=1,\dots,n} (y_i).$$

However, recall that the observations are noisy as shown in (2), the decisions made at the end of each subspace optimization would be prone to the measurement noise $\boldsymbol{\epsilon}$, which invokes the occurrence of outliers. Outliers could be misinterpreted as good solutions and lead to a poor optimization as non optimal solutions are propagated to subsequent subspace optimizations. Due to hyperparameter optimization, we decided to include the posterior mean of the fitted GPM into the decision scheme $\Phi(\boldsymbol{y}, X)$ given as

$$\Phi(\boldsymbol{y}, X) = \min_{\boldsymbol{x}_i, y_i \in \mathcal{O}} \big((1-\kappa) y_i + \kappa \mu(\boldsymbol{x}_i)\big), \tag{8}$$

where the posterior mean is weighted by a constant κ and the observations by $1-\kappa$. The posterior mean is for a noise robust decision scheme quite useful as it is the solution of a regularization problem, given by the equivalent kernel

$$J(f) = \frac{1}{2} \|f\|_{\mathcal{H}}^2 + \frac{1}{2\sigma_n^2} \sum_i^n (y_i - f(x_i))^2. \tag{9}$$

Minimizing the equivalent kernel with respect to f leads to the posterior mean as in (3); for further details see Williams and Rasmussen (2006). The expression consists of two parts, $\|f\|_{\mathcal{H}}^2$ represents the complexity and $\sum_i^n (y_i - f(x_i))^2$ represents the accuracy, which is divided by the noise variance. This means that the weight of the accuracy term increases as the reliability on the observations does. Note that the complexity term depends on the RKHS which is defined by the covariance function and its hyperparameters. Optimizing the hyperparameters via maximum likelihood estimation will favour the least complex model that agrees with the data. Regarding (9), this can be interpreted as increasing the weight on the complexity term. This leads to a posterior mean that may not agrees completely with every data point but rejects small, fast variations which are invoked by noise. Depending on the expected noise level, κ can be used to define the trade-off.

4. RESULTS

4.1 Simulation Results

The algorithm presented in Section 3 is demonstrated in a simulation environment of the XFEL optical synchronization system as illustrated in Figure 3. The system consists of $N = 5$ subsystems interconnected in a chain, each equipped with a PI controller, in other words, there are in total ten tuning knobs $\mathcal{X} \subseteq \mathbb{R}^{10}$. The LINAC block is neglected, i. e., $LINAC(s) = 1$, since there exist no model of the accelerator for now. The reference signal r is white Gaussian noise colorized with the filter F_r. The disturbances, $d_1, \dots, d_\mathrm{N}$, are colorized white Gaussian noise respectively. The performance quantity is the integrated timing jitter J given by the RMS norm of the performance channel z. Using the fact that the input signals are additive white Gaussian noise with power spectral density $S(\omega) = 1$, following Parseval's theorem, the optimization corresponds to an $\mathcal{H}_2$ minimization of the closed loop transfer function, $G_\mathrm{cl}(s)$, as shown in Heuer (2018). Since, the $\mathcal{H}_2$ norm undergoes hyperbolic growth when the poles of the system approaches the origin, closed-loop stability is guaranteed by the safe options as the $\mathcal{H}_2$ norm never exceeds the threshold. In the following, we compare the algorithm presented in this work, `MoSaOpt` in Algorithm 1, with `SafeOpt` from Berkenkamp et al. (2016). In addition, we want to investigate the impact of the dimension of the subspace on the performance. Therefore, we consider in total four BO implementations: (1) `LineBO` + `SafeOpt`, (2) `LineBO` + `MoSaOpt`, (3) `PlaneBO` + `SafeOpt`, (4) `PlaneBO` + `MoSaOpt`. We use the Matérn kernel with $\nu = 3/2$, all initial hyperparameters are chosen equivalently for each implementation. First, we consider a noise-free simulation, so the noise standard deviation is set to a very small value, $\sigma_n = 10^{-5}$. In all experiments we use the expected improvement acquisition function from (5) in exploitation. In addition, the maximum number of subspace iterations is restricted to $M = 40$ for `LineBO` and $M = 20$ for `PlaneBO`, because `PlaneBO` takes within one subspace optimization considerably more observations. As the set of observations increases over the subspace optimizations, we run into risk of prohibitively high computational demand. Furthermore, the combinations are restricted to the five PI-gains of each subsystem and the combinations of the P-

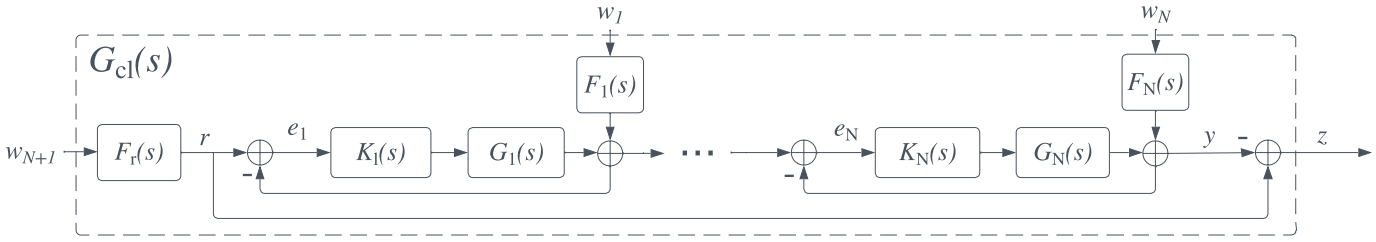


Fig. 3. Block diagram of the optical synchronization system for the simulation.

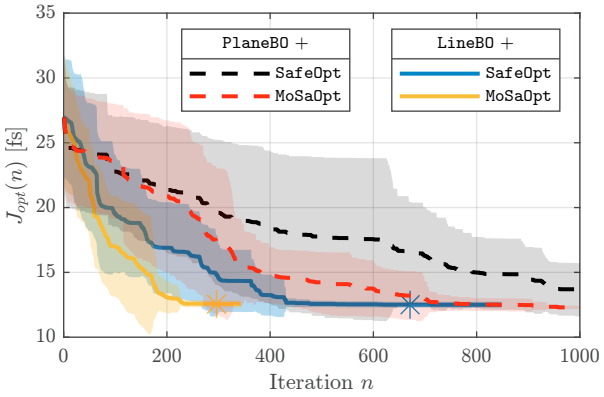


Fig. 4. Comparison of the BO methods based on the simulated system. The colored area corresponds to the single standard deviation and ✳ denotes the occurrence of the optimum.

gains of the laser subsystems. Hence, the set of directions is defined as

$$r_i \in \left\{ [P_1, P_3]; [P_1, P_5]; [P_3, P_5]; [P_1, I_1]; \dots; [P_5, I_5] \right\}.$$

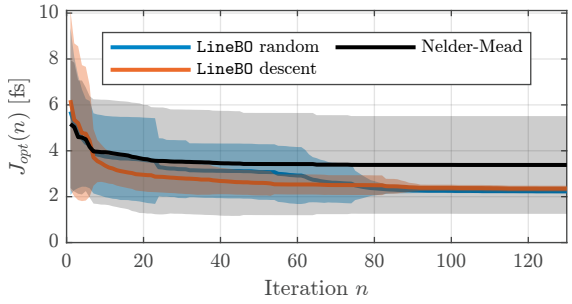
Recall that r_i is the direction selected by the direction oracle. Here, we use a random oracle which selects the direction based on a uniform probability distribution over all possible subspaces. In order to provide a statistical evident comparison, the optimization of each method is repeated ten times for different starting points. The code can be found in Lübsen et al. (2023).

Figure 4 shows the results of the BO comparison. It can be seen that line searches performs significantly better in terms of convergence rates, whereas `PlaneBO + MoSaOpt` provides a slightly improved solution quality. It is obvious that the BO methods with `MoSaOpt` clearly improves robustness to different starting points as indicated by the smaller, covered, areas in Figure 4. Furthermore, the new method improves the convergence rates for both implementations significantly. In case of `LineBO` with `MoSaOpt`, the variance could be reduced to its final value after only 230 iterations (function queries) and the optimum was found after 296 iterations. In contrast, `SafeOpt` requires 432 and 671 iterations respectively, which means that the number of evaluations is doubled. As expected, `PlaneBO` needs considerably more evaluations due to longer exploration. This results for the underlying system in a worse convergence rate; `PlaneBO + MoSaOpt` finds the optimum after 1365 evaluations as `PlaneBO + SafeOpt` requires 2284. The high iteration numbers are cut in Figure 4, due to illustrative reasons. However, Figure 4 shows, that `PlaneBO + MoSaOpt` converges faster.

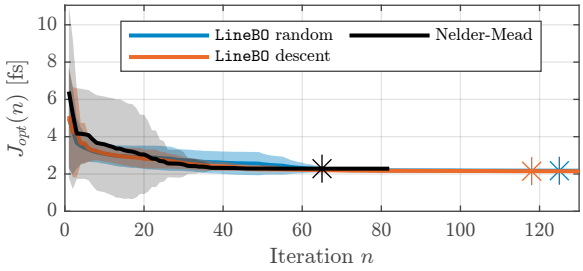
4.2 Experimental Results

In the following, we present experimental results from a laboratory optical synchronization system at DESY. The considered chain has $N = 2$ components. An additional laser provides the reference signal but is not included in the optimization. In this case, System 1 is the link and System 2 is the laser. The synchronization quantity is again the integrated timing jitter J between the reference laser and the laser, i. e., System 2, which is determined every 0.1 s by an out-of-loop optical cross-correlator (OXC) indicated by the last, right summation in Figure 3. The advantage of this instrument is its high resolution, so we decided to set $\kappa = 0.3$ in (8). In addition, the noise is further reduced by averaging over multiple jitter samples. In the first experiment, 10 samples are used, corresponding to one second of time per jitter generation. In the second experiment, the sample size is increased to 100. Each optimization is again repeated ten times for different starting points. Now we compare `LineBO + MoSaOpt` with the well known Nelder-Mead algorithm as described in Lagarias et al. (1998). Additionally, two different oracles are used, namely random and descent as described in Kirschner et al. (2019). The descent oracle increases the local convergence rate, whereas with a random oracle, also global guarantees are provided.

Figure 5 illustrates the mean optimal jitter for ten different starting points. The final, optimal performance of the `LineBO` implementations is equivalent, whereas using the descent oracle lead to a significantly improved convergence rate. As expected, Nelder-Mead shows a high convergence rate for the low dimensional problem, but is partially unable to find reasonable optima as shown in Figure 5a. This shows the low robustness of Nelder-Mead against noisy observations, even though the noise is small. In contrast, both BO implementations show robust solutions against the starting points and noise. Increasing the set of samples per jitter generation would clearly decrease the sample mean variance. As shown in Figure 5b, Nelder-Mead provides good solutions by averaging the jitter over 100 measurements, however, this increases the time consumption by a factor of ten. Comparing the optimization time of Nelder-Mead in Figure 5b and `LineBO` in Figure 5a, Nelder-Mead needs with approximately 400 s, five times longer than the `LineBO` implementations, which require 80 s. However, the solution qualities are equivalent in the sense that the obtained optima have the same performance. Considering time efficiency and solution quality `LineBO + MoSaOpt` clearly outperforms Nelder-Mead. Using the descent oracle significantly increases the convergence rate, while using the random oracle the optimal jitter is partially constant



(a) One second jitter generation.



(b) Ten seconds jitter generation.

Fig. 5. Performance comparison of `LineBO` + `MoSaOpt` with random and descent oracle, and Nelder-Mead based on a small-scale optical synchronization system consisting of a laser and a link.

as shown in Figure 5b. This indicates that a subspace does not affect the synchronization or is already optimized. The descent oracle does not select such none-effective subspaces, resulting in a better convergence rate. However, since the gradient is solely checked in the neighbourhood of the current point, the method is prone to converge to local minima.

5. CONCLUSION

We presented a practical formulation of safe Bayesian optimization by separating exploration and exploitation combined with hyperparameter optimization. With this we can show in practice a significantly improved convergence rate while guaranteeing safety. In addition, hyperparameter optimization opens the applicability of a new, noise robust decision scheme. This characteristic is especially decisive for sequential subspace optimization approaches, as wrong decisions would decrease the quality of the optimization process considerably. Furthermore, we demonstrated the performance of `MoSaOpt` on a simulated plant compared to an other safe BO scheme. Experimental results on a physical plant showed the noise robustness in a comparison to a classical optimization algorithm like Nelder-Mead. The future work will be focused on a new definition of the set which involves less knowledge about the objective. One approach is to use multiple inputs, e. g., a simulator of the real plant and learn the dependency between both functions. Depending on the fidelity between the real plant and the simulator samples, the objective of the real plant can be predicted for unobserved inputs given the samples of the simulator which will also lead to a more efficient solution finding.

REFERENCES

Berkenkamp, F., Schoellig, A.P., and Krause, A. (2016). Safe controller optimization for quadrotors with Gaussian processes. In *IEEE Int. Conf. Robot. Autom (ICRA)*, 491–496.

Berkenkamp, Felix, K.A. and Schoellig, A.P. (2021). Bayesian optimization with safety constraints: safe and automatic parameter tuning in robotics. *Machine Learning*.

Heuer, M. (2018). *Identification and control of the laser-based synchronization system for the European X-ray Free Electron Laser*. Doctoral dissertation, Technische Universität Hamburg-Harburg.

Jones, D.R., Schonlau, M., and Welch, W.J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4), 455–492.

Kanagawa, M., Hennig, P., Sejdinovic, D., and Sriperumbudur, B.K. (2018). Gaussian processes and kernel methods: A review on connections and equivalences.

Khosravi, M., Eichler, A., Schmid, N., Smith, R.S., and Heer, P. (2019). Controller tuning by Bayesian optimization an application to a heat pump. In *18th Eur. Control Conf. (ECC)*, 1467–1472.

Kirschner, J., Mutny, M., Hiller, N., Ischebeck, R., and Krause, A. (2019). Adaptive and safe bayesian optimization in high dimensions via one-dimensional subspaces. In *36th Int. Conf. Mach. Learn. (ICML)*, 3429–3438.

Lagarias, J.C., Reeds, J.A., Wright, M.H., and Wright, P.E. (1998). Convergence properties of the nelder–mead simplex method in low dimensions. *SIAM Journal on Optimization*, 9(1), 112–147.

Lübsen, J.O., Schütte, M., Schulz, S., and Eichler, A. (2023). Code for paper: A Safe Bayesian Optimization Algorithm for Tuning the Optical Synchronization System at European XFEL. doi:10.5281/zenodo.7810494. URL `https://doi.org/10.5281/zenodo.7810494`.

Moriconi, R., Deisenroth, M.P., and Sesh Kumar, K.S. (2020). High-dimensional Bayesian optimization using low-dimensional feature spaces. *Machine Learning*, 109(9), 1925–1943.

Schillinger, M., Hartmann, B., Skalecki, P., Meister, M., Nguyen-Tuong, D., and Nelles, O. (2017). Safe active learning and safe Bayesian optimization for tuning a pi-controller. In *20th World Congr. Int. Fed. Autom. Control (IFAC)*, 5967–5972.

Schütte, M., Eichler, A., Schlarb, H., Lichtenberg, G., and Werner, H. (2021). Decentralized output feedback control using sparsity invariance with application to synchronization at European XFEL. In *60th IEEE Conf. Decision Control (CDC)*, 5723–5728.

Sui, Y., Gotovos, A., Burdick, J., and Krause, A. (2015). Safe exploration for optimization with Gaussian processes. In *32nd Int. Conf. Mach. Learn. (ICML)*, volume 37 of *Proceedings of Machine Learning Research*, 997–1005. PMLR, Lille, France.

Sui, Y., Zhuang, V., Burdick, J.W., and Yue, Y. (2018). Stagewise safe Bayesian optimization with Gaussian processes. In *35th Int. Conf. Mach. Learn. (ICML)*.

Williams, C.K. and Rasmussen, C.E. (2006). *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA.