2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA) | 978-1-7281-8470-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICMLA51294.2020.00044

SASBO: Self-Adapting Safe Bayesian Optimization

Stefano De Blasi
Bosch Rexroth AG
Lohr am Main, Germany
Stefano.DeBlasi@boschrexroth.de

Alexander Gepperth
UAS Fulda
Fulda, Germany
Alexander.Gepperth@informatik.hs-fulda.de

***Abstract*—Optimizing an unknown objective function under uncertainty requires a balance between exploration (learn more about the objective) and exploitation (find the global optimum). Safe optimization aims to guarantee that safety requirements of the next observation to be performed are fulfilled before performing it. Common approaches are based on Gaussian process regression, also known as Kriging, as a surrogate model iteratively estimating the safety and selecting next observations by Bayesian optimization methods. The hyper-parameter setup usually requires a lot of domain-specific knowledge (if no data is available) or prior data to optimize the hyper-parameters. But it is precisely the lack of these two factors that is the main reason when safe optimization becomes interesting: If the system is unknown and random experiments to generate data are not allowed due to restrictions. We present a novel method for safe Bayesian optimization with self-adapting hyper-parameters, which requires only one safe initial observation and easily selectable initial hyper-parameters. By safely self-adapting the parameters, it is possible to find the global optimum with a reliability regarding safety requirements. Thus, the method can be used even with limited domain-specific expertise and covers a wide range of applications with a minimum of customization.**

***Index Terms*—Safety constrains, global optimization, Kriging, Bayesian optimization, hyper-parameter adaption**

I. Introduction

Minimizing uncertainty in machine learning is a research topic with growing attention especially for environments with safety requirements, e.g. applications in industrial plants are not allowed to lead to downtime by a wrong decision or the physical security of humans has to be ensured (autonomous driving). The definition of safe optimization is to guarantee that the safety requirements of the next observation to be performed are fulfilled before performing it. This makes safe optimization especially interesting for *interacting with real world* applications, e.g. recommendation systems [1], quadrotor vehicles [2] or in industrial context [3]–[5]. Common approaches are based on Gaussian process (GP) regression, originally Kriging in geostatistics [6], because it provides predictions of distributions instead of a point in contrast to common regression models. These predicted distributions can be used as a surrogate model iteratively estimating the safety by predicting the uncertainty of potential next observations [2] and then selecting the next observations with Bayesian optimization methods [7] by optimizing an acquisition function [8]. Even though there have already been applications in various areas, safe global optimization is still scientifically further investigated and developed from an algorithmic point of view [9].

Here, the requirements regarding prior domain knowledge about the objective function are problematic. Depending on the domain, experts are not always able to provide abstract mathematical information about the problem to be solved. Unfavorably, exactly such abstract information is needed to set up GP regression for safe optimization, including the selection of hyper-parameters. A common approach is to optimize the hyper-parameters of the GP regression by prior data, e.g [10], [11], which is not always available for the given problem. This is especially true in the context of safe optimization, since it is not tolerated for such problems to perform prior experiments with random outcome. To solve such problems, we propose a method that allows a safe Bayesian optimization with self-adapting hyper-parameters, which requires only one safe initial observation and easily selectable initial hyper-parameters. Thus, self-adaptive safe Bayesian optimization (SASBO) can be used even with limited expertise, and is ultimately a tool that is as general as possible and covers a wide range of applications with a minimum of customization.

A. Own contributions

We introduce modifications of safe Bayesian optimization methods automatically handling the majority of hyper-parameters by running an optimization algorithm on some parameters (length scales of the kernel and noise variance of the GP) while iterative scaling the observations leads to a good fit of the remaining and fixed properties (variances of the kernel and the mean of the GP). Our evaluation demonstrate reliability regarding the safety constraints and the resulting optimized hyper-parameters.

II. Background and methods

We formulate the optimization problem of a d dimensional nonlinear objective function $f(x)$ as a maximization. Safe optimization is defined by solving a general maximization problem with constraints regarding tolerated observation values during the optimization. For example, no experiments are allowed to lead to $f(x) < f_{\min}$, where $f_{\min}$ is an user defined threshold, not to be confused with the global minimum of the $f(x)$. The optimization goal is then:

$$\max_{x \in \mathcal{X}^d} f(x) \quad \text{s.t.} \quad f(x) \geq f_{\min}. \tag{1}$$

Restrictions based on other functions are possible, however, for simplicity we continue with (1). We assume that $f(x)$ excluding the observation noise is at least quasi-continuous

978-1-7281-8470-8/20/$31.00 ©2020 IEEE
DOI 10.1109/ICMLA51294.2020.00044

and thus continuous within the mentioned limitations of the optimization space $\mathcal{X}^d$. Otherwise any significant discontinuity, even with cautious exploration, could lead to an unexpected observation and thus make it impossible to ensure safe exploration.

A. Gaussian process regression

We assume that each $f(x)$ with different x is a random variable and that a finite number N leads to a joint Gaussian distribution [12]:

$$[f(x_1), .., f(x_N)] \sim \mathcal{N}\left(m(x_1), .., m(x_N), \sigma(x_1), .., \sigma(x_N)\right). \tag{2}$$

Such a joint distribution of all random variables is a distribution of a GP. For regression, the GP is obtained by multiple generated sample functions, each fitting the prior measurement points, so-called observations, of $f(x)$. Thus, GP regression returns a mean and variance of the Gaussian normal distribution for each x of the approximated unknown function:

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')). \tag{3}$$

By setting a prior mean function $m(x)$ for the distribution, one can adjust the average value over the generated sample functions. Different kernels $k(x, x')$, also called covariance functions, can be selected to define the covariance of any two function values $f(x)$ and $f(x')$ with specific smoothness and periodicity modeling properties. In this way, GP regression, originally Kriging in geostatistics [6], can be used across disciplines, e.g. for geographic terrains [13], sensor networks [14], or battery conditions [15]. We focus on the radial basis function (RBF) kernel, also known as squared-exponential kernel, which is widely used because of its infinite derivatives and its property of universal integration against most functions [16] (not only for GPs, but also for support vector machines):

$$k(x, x') = \sigma^2 \exp\left(-\frac{(x - x')^2}{2\ell^2}\right). \tag{4}$$

The RBF kernel is parametrized by a scale parameter ℓ (larger values for smoother functions) and a variance σ^2 (larger values for larger average distance from function to GP mean). Since the selection of other kernels might be more appropriate for modeling an unknown function [12], the optimal kernel selection remains a domain specific problem.

B. Bayesian optimization

Bayesian optimization [7], [11] is a method that iteratively explores the unknown objective function $f(x)$ in pursuit of the global maximum by determining new query points of x. After each iteration, $f(x)$ is modeled with multiple sample functions based on previous observations. Therefore we use GP regressions with the RBF kernel, which is the most common probabilistic model used in Bayesian optimization [12]. The model returns predicted distributions for each x, providing mean values $\mu(x)$ and the variance $\sigma^2(x)$ which indicates the uncertainty of the models prediction at x. This can be used to determine the most informative observation point for the next iteration by constructing the acquisition function [17]. Such an acquisition function indicates the expected information gain of evaluating the objective function with specific x to minimize the uncertainty at this point. For example, the upper confidence bound (UCB) acquisition function for iteration n is defined by:

$$u_i(x) = \mu_i(x) + \beta\sigma_i(x), \tag{5}$$

where β is the constant defining the scaling of the confidence interval. The combination of exploitation ($\mu_i(x)$) and exploration ($\sigma_i(x)$) is chosen to minimize the number of experiments required to approximate $f(x)$. The observation points for the next iteration is obtained by maximizing the chosen acquisition function, e.g. UCB:

$$x_{i+1} = \arg\max_{x \in \mathcal{X}^d} u_i(x), \tag{6}$$

C. Safe Bayesian optimization

Safe Bayesian optimization [1] is a constrained Bayesian optimization guaranteeing only observations within a predefined safety restriction and therefore only partly global. Here, each observation should be above the safety threshold f_{min}, which limits the optimization space to the safe set $\mathcal{S} = \{x \in \mathcal{X}^d | f(x) \geq f_{\text{min}}\}$. In this way, the algorithm minimizes the risk of negative experiments during optimization [2] by checking the safety requirements of the next observation before performing it. Since the objective function $f(x)$ is unknown, the true safe set $\mathcal{S}$ can only be estimated. For this estimation, the lower confidence interval can be used:

$$l_i(x) = \mu_i(x) - \beta\sigma_i(x). \tag{7}$$

Whenever the calculated lower limit of the interval results in values above the defined safety threshold f_{min}, the points are assumed to be safe:

$$\mathcal{S}_i = \{(x) \in \mathcal{X}^d | l_i(x) \geq f_{\text{min}}\}. \tag{8}$$

The estimation is made for each iteration i and the safe set is extended during the exploration. Since the optimization no longer considers the entire space $\mathcal{X}^d$, but the safe set $\mathcal{S}_i$, the formula changes to:

$$x_{i+1} = \arg\max_{x \in \mathcal{S}_i} u_i(x). \tag{9}$$

The most promising maximizers $\mathcal{M}_i$ (points with increased probability where the global maximum could be located) of the safe set are obtained by looking for all x where the upper bound is larger than the largest lower bound:

$$\mathcal{M}_i = \{x \in \mathcal{S}_i | u_i(x) \geq \max_{x' \in \mathcal{X}^d} l_i(x')\}. \tag{10}$$

To obtain possible expanders $\mathcal{E}$ (points with increased probability where the safe set could be extended) we implemented an efficient method by interpreting $\mathcal{S}$ as d dimensional function:

$$S_i(x) = \begin{cases} 1, & x \in \mathcal{S}_i \\ 0, & x \notin \mathcal{S}_i \end{cases}. \tag{11}$$

The second derivatives of $S_i(x)$ is approximated by d dimensional Laplace filtering (12) leading to our expander set with simple logical comparisons:

$$\mathcal{E}_i = \{x \in \mathcal{S}_i | \sum_{r=1}^{d} \frac{\partial^2 S_i(x)}{\partial^2 x^{(r)2}} < 0\}. \tag{12}$$

These potential observation points in $\mathcal{E}$ are a subset of $\mathcal{S}$ and have a high expectation to extend the knowledge about the objective function. Each experiment is a trade off between finding the maximum and minimizing the uncertainty. Thus, the observation points are selected within the union of both calculated sets:

$$x_i = \underset{x \in \mathcal{E}_i \cup \mathcal{M}_i}{\arg\max} (u_i(x) - l_i(x)). \tag{13}$$

III. Self-Adaptation of Safe Bayesian Optimization

The hyper-parameter setup of the RBF kernel based GP regression includes the length scale parameters $\ell \in \mathcal{L}^d$ and the variances $\sigma^2 \in \mathcal{O}^d$, as well as the noise variance σ^2_{noise}. Additionally, one can choose the mean function $m(x)$ of the GP regression. The selection of these $2d + 2$ parameters requires usually prior domain knowledge to predict the rough characteristics of the objective function. Alternatively, the hyper-parameters can be optimized by a large set of observation points. In both ways, prior knowledge about the objective function has to be provided before running a GP regression based safe optimization.

To reduce the necessary prior knowledge for safe optimization, we extend SafeOpt [1] (section II-C) by two steps, which are described in this section. Both modifications are equally applicable to other safe Bayesian optimization methods like StageOpt [18]. We provide pseudo-code (Algorithm 1) for comprehensibility with references to the formulas and information specified in this paper. While the exploration iterations are very conservative at the beginning, the updates of GP hyper-parameters minimize the required iterations to find the safe optimum, see toy example in Fig. 1.

A. Scaling the objective function

Since safe optimization does not consider the full objective function, the unsafe regions can be interpreted as irrelevant for the optimization and are not explored. By setting the mean value over all sample functions of the GP regression at each x to $m(x) < f_{\text{min}}$, it is more likely to predict values excluded from $\mathcal{S}$ in the absence of knowledge to the contrary, e.g. proximity to prior observations significantly above f_{min}. We ensure this advantageous effect by scaling the safety threshold to $f'_{\text{min}} = 1$ and setting $m(x) = 0$. The second scaling point for our iterative adaptive linear scaling transformation is the currently known maximum $f'_i(x^*) = 2$, which leads to the other scaled observation data:

$$\mathbf{y}' = \frac{\mathbf{y} - f_{\text{min}}}{\max(\mathbf{y}) - f_{\text{min}}} + 1, \tag{14}$$

where $\mathbf{y}$ is the vector of all observation results. Whenever the known maximum $f_i(x^*) = \max(\mathbf{y})$ exceeds the prior one, the scaling denominator of (14) is changed and the whole set of observation points is transformed again. The relative large distance of the GP regression mean to f'_{min} causes conservative safety estimations for unknown regions of $f(x)$. The scaling transformation affects finally the results of the acquisition function (13). To ensure a good optimization, a few iterations with a greedy acquisition function like (5) within the subset of the safe region can be helpful after the safe exploration phase [18].

B. Constrained optimization of hyper-parameters

As the variances σ^2 are only a scaling parameter that depends on the absolute values of the objective function, we can fix the variances with regard to the iterative scaling. The remaining length scale kernel and noise variance hyper-parameters $\theta = (\ell, \sigma^2_{\text{noise}})$ of the GP regression can be optimized by maximizing the log marginal likelihood [12]:

$$\theta^* = \arg\max_{\theta} \log p(\mathbf{y}'|X, \theta), \tag{15}$$

where X is the data of all previous observation inputs. The optimization leads to an trade-off between model fit and model complexity. Evolutionary or conjugate-gradient algorithms are able to analytically solve (15) because of the GP predictions. We use the truncated Newton algorithm, also called Newton conjugate-gradient method [19], to optimize every k iterations. Information about our constrained optimization of $d+1$ hyper-parameters is listed in Table I. These constraints avoid rapid and unsafe modeling changes by setting length scale dependent limits.

TABLE I
Constraints of GP adaptation optimization

Parameters	Comment	Constraints
σ^2	Variances of kernel	fixed to 1.0
ℓ	Length scales of kernel	$(0.8 \cdot \ell_i, 1.4 \cdot \ell_i)$
σ^2_{noise}	Noise variance of GP	$(0.001, 0.05)$

Since too large length scales would cause uncertain exploration, the parameters are reduced by 10% after optimization, so that the resulting values are rather too small than too large. This also reduces possible negative effects of optimization variations which could endanger safety.

C. Remaining initialization efforts

Our hyper-parameter selection of the GP regression includes only the initialization of length scales $\ell_0 \in \mathcal{L}^d$ and noise variance $\sigma^2_{0\ \text{noise}}$. These parameters are easy to choose compared to the standard method because low values of ℓ and high values of σ^2_{noise} generally lead to safer but at the beginning slower optimization and so only little prior knowledge of the objective function is required, e.g. the knowledge of one single safe starting point x_0. If the objective function value $f(x_0)$ is relatively close to f_{min}, as compared to the maximum $f(x^*)$, the length scale initialization should be smaller. Alternatively, one wants to provide an assumption for the expected maximum f_{max} for the scaling step, ideally close to the global

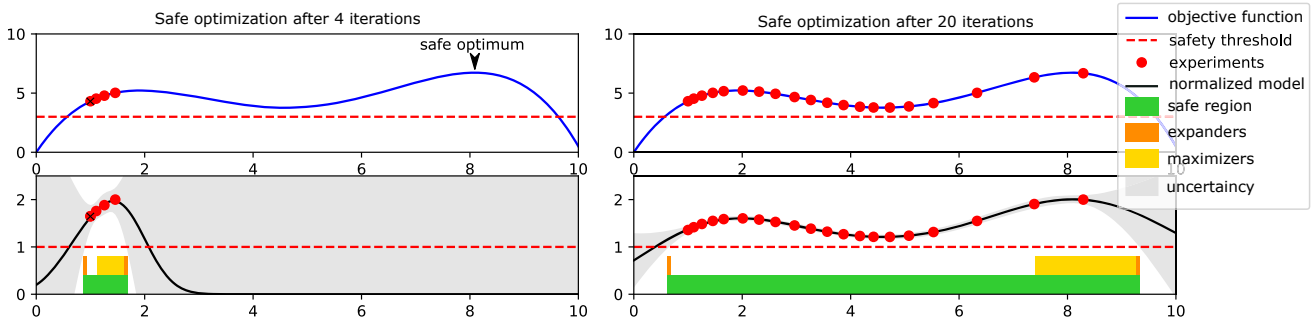

Fig. 1. Toy example for safe optimization: The true objective function (blue line) is normalized (black line) after each iteration while the estimations of safe region $\mathcal{S}$, expanders $\mathcal{E}$ and maximizers $\mathcal{M}$ are updated. During the safe optimization, no experiments lead to objective result values below the threshold (red dashed line). Learning more about the unknown objective function initialize updates of GP hyper-parameters which minimize the required iterations to find the safe optimum.

Algorithm 1 SASBO

Input: $f_{\min}$, $f_{\max}$, ℓ_0, $\sigma_{0\ \text{noise}}^2$, x_0

Output: $\mathcal{GP}$

```
Run experiment and get f(x_0)
initialize GP with ℓ_0, σ²_{0 noise}, σ²_{1..d} = 1.0 and f(x_0)
for i = 1 to N do
    Estimate S_i by (8)
    Calculate M_i by (10)
    Calculate E_i by (12)
    Calculate x_i by (13)
    Run experiment and get f(x_i)
    Scale all observations by (14)
    Update GP
    if i mod k = 0 then
        Update constraints according to Table 3
        Optimize ℓ_i and σ²_{i noise} by (15)
        Decrease ℓ_i by 10%
        Update GP with new hyper-parameters
    end if
end for
return GP
```

maximum $f_{\max} \approx f(x^*)$. However, every provided value with $f(x_0) < f_{\max} < f(x^*)$ is helpful to reduce this effect.

For the safe optimization, further hyper-parameters have to be selected: β, $f_{\min}$, a threshold for expander regulation [2] and sometimes Lipschitz parameters [1], [9] to estimate the safe set. With our modifications, the static selection of $\beta = 3.0$ is recommended throughout, while the latter two parameters are omitted. This simplifies the application when the objective function is unknown. Instead of choosing the optimal length scale by system knowledge (e.g. prior experiments or expertise), one simply chooses length scales that are rather too small at the beginning (smaller initialization values mean slower, but safer optimization). For the batch size, k should be selected rather too large, because especially in the beginning of the optimization enough knowledge has to be generated before a safe adaption can be possible. Therefore, the optimal k is related to the multi-dimensionality of the objection function, since information should be existing about each dimension. In Table II the recommendations of hyper-parameter selection are summarized.

TABLE II
HYPER-PARAMETERS OF SAFE OPTIMIZATION

Parameters	Comment	Selection
$\ell_0 \in \mathcal{L}^d$	Init. length scales of kernel	rather too small
$\sigma_{0\ \text{noise}}^2$	Init. noise variance of GP	0.01
β	Multiplier for confidence interval	3.0
k	Batch size	increases with d, rather too large
$f_{\max}$	Assumption for global maximum	optional

IV. Evaluation

For the evaluation, the inverted Styblinski-Tank function is a good optimization problem because it can be resized to arbitrary dimensions d. To evaluate also the reliability of the length scale adaption with 16 different objective functions, we extend the Styblinski-Tank function by a transformation for independently scaling the dimensions with a vector h:

$$f(x) = \frac{-1}{2} \sum_{r=1}^{d} \left(\frac{x^{(r)}}{h^{(r)}} \right)^4 - 16 \left(\frac{x^{(r)}}{h^{(r)}} \right)^2 + 5 \left(\frac{x^{(r)}}{h^{(r)}} \right). \quad (16)$$

There is one global maximum $x^* \approx (-2.9, ..., -2.9) \cdot h$ within the optimization space $x^{(r)} \in [-5 \cdot h^{(r)}, 5 \cdot h^{(r)}]$. The modification allows us to evaluate the iterative optimization of the length scale parameters $\ell_{1..d}$. We randomly initialize the optimization with x_0 where $f(x_0) \geq 15$, see green regions in Fig. 2. The safety constraint is defined by $f_{\min} = 10.0$ leading to unsafe regions indicated in gray in Fig. 2.

We initialize the length scale parameters $\ell_{1..d} = 0.5$, although lower values would be fine, but require more iterations to safely adjust the these parameters, which slows down the optimization. No optional assumption for global maximum was not necessary because we ensured a relatively large difference between $f(x_0)$ and $f_{\min}$. We found that $k = 20$ is a good batch size, although larger values would be just as

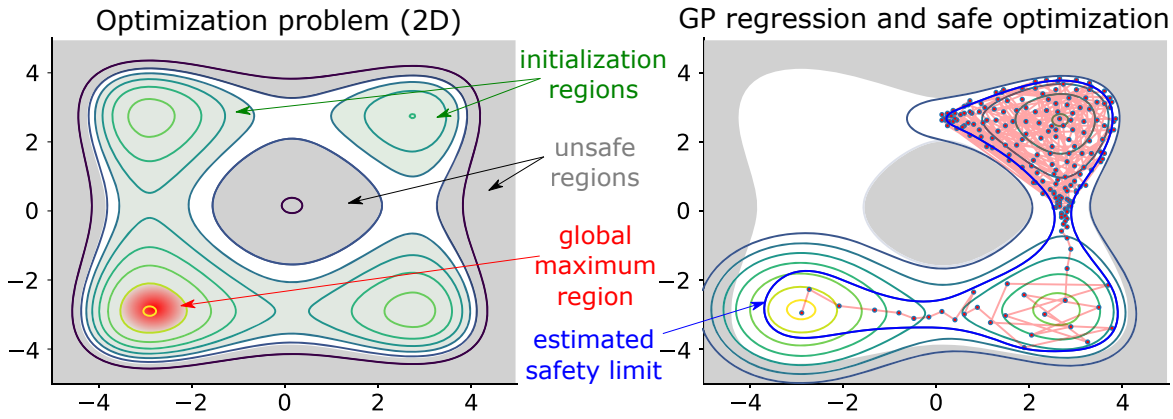

Fig. 2. Exemplary optimization of the inverted Styblinski-Tank function with $d = 2$ and $h = (1, 1)$: Gray regions indicate the true unsafe regions (values below 10.0). The initialization point is randomly selected for a value above 15.0 (green regions in left plot). The exemplary safe optimization on the right contour plot is initialized at $x_0 = (2.5, 2.5)$ and starts with tiny steps to adapt its GP hyper-parameters and then approaches the optimum in larger steps. After each iteration, the estimated safety limit (blue line) is updated to ensure that the next observation leads to results above $f_{\min}$.

okay as initialization length scales that are rather too small. The remaining parameters are set according Table II.

16 different combinations of the scaling transformation h between $(0,5, 0,5)$ to $(2, 2)$ vary the sensitivity of the objective ($n = 10$ times each). In this way the reliability can be evaluated for a wide range of different situations. The exploration and optimization ran (all in all 160 times) over 300 iterations each, and no violation of the safety restriction was observed, while the region of the global maximum was always successfully detected. The right contour plot in Fig. 2 illustrates an exemplary optimization run with $h = (1, 1)$ initialized at $x_0 = (2.5, 2.5) \cdot h$, which is one of the most distant safe points from the optimum x^*. By tiny steps starting from x_0 at the beginning, information is carefully collected and thus the GP hyper-parameters are adjusted. After that one approaches the global optimum in larger and more determined steps. The safety limit is re-estimated after each iteration to exclude potentially uncertain observations from the acquisition function for the next iteration. Some former observations are outside the estimated safety limit because of estimated noise.

If h is varied, the iterative optimization of the length scale parameters $\ell_{1..d}$ leads to different values. To investigate the correlation between the objective function scaling transformation values of h and the optimized length scale parameters at the end of the 300 iterations and $\frac{300}{20} = 25$ optimization runs, the classical statistics of Pearson correlation coefficients (values close to 1 indicate a positive linear correlation, 0 is random and close to -1 indicates a negative linear correlation) are calculated. We found clear linear correlations for the x and y dimension with coefficients of 0.994 and 0.993 ($n = 160$). The variation of one dimension did not affect the optimization of the other, which can also be seen in Fig.3. Furthermore, the identified linear correlation is indicated as black dashed line.

Finally, we evaluated the self-adapting safe Bayesian optimization using the Styblinski-Tank function with $d = 3$ and $d = 4$ in different setups. For example, the initialization point was set to $x_0 = (2.5, ..., 2.5) \cdot h$. For $d = 3$, the safe

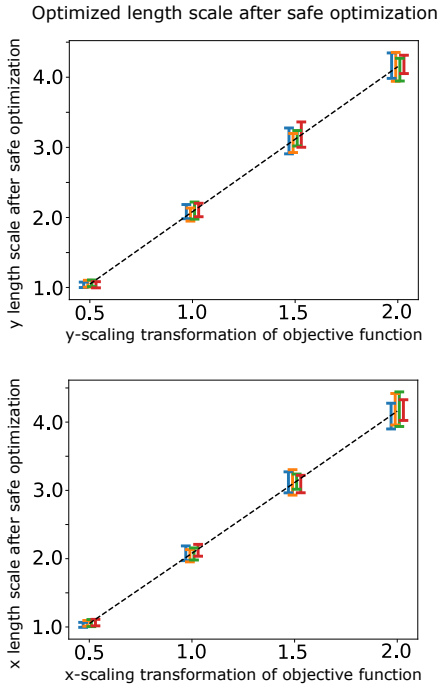

Fig. 3. Evaluation of length scale optimization: For different scaling transformations of the objective function, different length scale parameters are adapted during safe optimization. Standard deviation for all 16 transformation combination (each $n = 10$) is illustrated and no outliers were observed. Different bar colors represent different scaling transformations of the other dimension.

optimization required less than 160 iterations to safely find the maximum region starting from one of the most distant safe points. Safely optimizing the $d = 4$ setup successfully required about 190 iterations to reach the region. All the tested optimizations did not lead to failure and ensured the safety criteria. The adapted length scale parameter results of Fig.3 are also applied to the higher-dimensional experiments.

V. Conclusion

The hyper-parameter setup of GP regression usually requires a significant amount of domain-specific knowledge (if no data is available) or prior data to optimize the hyper-parameters. But it is precisely the lack of these two factors that is the main reason when safe optimization becomes interesting: If the system is unknown and random experiments to generate data are not allowed due to restrictions. We presented a novel method for safe Bayesian optimization with self-adapting hyper-parameters, which requires only one safe initial observation and easily selectable initial hyper-parameters. By safely self-adapting the parameters, it was possible to find the global optimum within an acceptable number of experiments and with a reliability regarding safety requirements. Our modifications demonstrated with SafeOpt [1] can be used equivalently for other safe Bayesian optimization methods like StageOpt [18].

Iterative learning and automatic tuning control methods are not only a popular research topic in theory, but also relevant for applications, especially for industrial process control [20]. Ensuring safety during learning is essential here, as industry best practice requires proof of safety [21]. With our modifications, e.g., Bayesian optimization with unknown hyper-parameters of welding [22] could be applied to optimize the process without producing bad-quality results during learning. As advances in automation technology enable the inclusion of time critical active learning in industrial process control [23], we continue our research with applications that optimize such processes with minimal manual adjustments addressing the expense problem of machine learning projects in industry.

A. Final remarks

Even if the results are promising, there are remaining aspects which should be noted before applying SASBO:

1) RBF kernel selection: We fixed the kernel selection to the most popular one. For some objective functions, however, other kernels are more appropriate, so extending the functionality of our approach with secure automatic kernel selection would further improve its applicability. The combination of minimizing complexity and maximizing data customization should be considered [24] so that the efficiency of the proposed method is maintained and suitable for active learning.

2) Only observations near the safety threshold: Whenever the current known optimum is not significantly larger than the safety threshold, the next iteration is risky. Especially for the first iterations this remains an issue. We recommend providing a guess for the expected optimum to remedy this risk.

3) High-dimensional objective functions: Even though we used a batch size of $k = 20$ for all experiments presented, it is quite logical that k must rise with larger d. This should also be examined before applying this approach to more complex problems, e.g. $d > 4$. Either way, k should be generously selected.

References

[1] Y. Sui, A. Gotovos, J. Burdick, and A. Krause, "Safe exploration for optimization with gaussian processes," in *International Conference on Machine Learning*, 2015, pp. 997–1005.

[2] F. Berkenkamp, A. P. Schoellig, and A. Krause, "Safe controller optimization for quadrotors with gaussian processes," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 491–496.

[3] M. Khosravi, A. Eichler, N. Schmid, R. S. Smith, and P. Heer, "Controller tuning by bayesian optimization an application to a heat pump," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 1467–1472.

[4] M. Schillinger, B. Hartmann, P. Skalecki, M. Meister, D. Nguyen-Tuong, and O. Nelles, "Safe active learning and safe bayesian optimization for tuning a pi-controller," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 5967–5972, 2017.

[5] S. De Blasi, "Active learning approach for safe process parameter tuning," in *Machine Learning, Optimization, and Data Science*. Cham: Springer International Publishing, 2019, pp. 689–699.

[6] D. G. Krige, "A statistical approach to some basic mine valuation problems on the witwatersrand," *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 52, no. 6, pp. 119–139, 1951.

[7] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2015.

[8] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[9] Y. D. Sergeyev, A. Candelieri, D. E. Kvasov, and R. Perego, "Safe global optimization of expensive noisy black-box functions in the δ-lipschitz framework," *Soft Computing*, pp. 1–21, 2020.

[10] R. Marchant and F. Ramos, "Bayesian optimisation for intelligent environmental monitoring," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 2242–2249.

[11] J. Mockus, *Bayesian approach to global optimization: theory and applications*. Springer Science & Business Media, 2012, vol. 37.

[12] C. E. Rasmussen and C. Williams, "Gaussian processes for machine learning, vol. 1," *MIT press*, vol. 39, pp. 40–43, 2006.

[13] S. Vasudevan, F. Ramos, E. Nettleton, and H. Durrant-Whyte, "Gaussian process modeling of large-scale terrain," *Journal of Field Robotics*, vol. 26, no. 10, pp. 812–840, 2009.

[14] K. Yang, S. Keat Gan, and S. Sukkarieh, "A gaussian process-based rrt planner for the exploration of an unknown and cluttered environment with a uav," *Advanced Robotics*, vol. 27, no. 6, pp. 431–443, 2013.

[15] R. R. Richardson, M. A. Osborne, and D. A. Howey, "Gaussian process regression for forecasting battery state of health," *Journal of Power Sources*, vol. 357, pp. 209–219, 2017.

[16] C. A. Micchelli, Y. Xu, and H. Zhang, "Universal kernels," *Journal of Machine Learning Research*, vol. 7, no. Dec, pp. 2651–2667, 2006.

[17] N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: No regret and experimental design," 2010.

[18] Y. Sui, J. Burdick, Y. Yue *et al.*, "Stagewise safe bayesian optimization with gaussian processes," in *International Conference on Machine Learning*, 2018, pp. 4781–4789.

[19] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.

[20] S. Yin, X. Li, H. Gao, and O. Kaynak, "Data-based techniques focused on modern industry: An overview," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 1, pp. 657–667, 2014.

[21] N. Fulton and A. Platzer, "Safe reinforcement learning via formal methods: Toward safe control through proof and learning," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[22] T. Sterling and H. Chen, "Robotic welding parameter optimization based on weld quality evaluation," in *2016 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*. IEEE, 2016, pp. 216–221.

[23] S. De Blasi and E. Engels, "Next generation control units simplifying industrial machine learning," in *2020 IEEE 29th International Symposium on Industrial Electronics (ISIE)*. IEEE, in press.

[24] T. Chugh, A. Rahat, and P. S. Palar, "Trading-off data fit and complexity in training gaussian processes with multiple kernels," in *International Conference on Machine Learning, Optimization, and Data Science*. Springer, 2019, pp. 579–591.