

Article

Bayesian Optimization of Non-Invariant Systems with Constraints Developed for Application to the ECR Ion Source VENUS

Victor Watson , Christopher M. Campbell , Heather L. Crawford , Yue Shi Lai , Marco Salathe 
and Damon Todd 

Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA; cmccampbell@lbl.gov (C.M.C.);
hlcrawford@lbl.gov (H.L.C.); ylai@lbl.gov (Y.S.L.); msalathe@lbl.gov (M.S.); dstodd@lbl.gov (D.T.)

* Correspondence: viktorwatson@gmail.com

Abstract

In this work, we consider the optimization of non-invariant systems with both safety and control constraints. We present a new approach based on Bayesian optimization for the dynamic, safe and controlled optimization of such systems. Although there are other possible use cases, we focus on the application to the electron cyclotron resonance ion source VENUS. From experimental data, we have observed that VENUS behaves to first order as a non-invariant dynamic system with moving areas of instability. Our novel approach aims at providing a tool that can maintain system optimization in a safe way. This is accomplished by making sure the objective function, the beam current in the case of VENUS, does not fall under an operational minimum, while simultaneously requiring the optimization to avoid areas where VENUS is unstable. We compare the result of our approach on synthetic data modeled to mimic the behavior of VENUS with two methods from the literature, a standard Bayesian optimizer and a safe Bayesian optimizer, both adapted to deal with dynamic systems. A cross Student *T*-test is conducted to show the significance of the improvement given by the new method we introduce here, regarding the two preexisting methods we compared to. The results of the tests conducted on synthetic data show that the proposed method succeeds at maintaining the system optimized and obeys the predefined constraints better than the literature methods explored.

Keywords: Bayesian optimization; dynamic systems; non-invariant; multiple constraints; ECR ion source

MSC: 49N99



Academic Editors: Yanqing Zhang
and Puying Zhao

Received: 29 November 2025

Revised: 26 January 2026

Accepted: 28 January 2026

Published: 31 January 2026

Copyright: © 2026 by the authors.
Licensee MDPI, Basel, Switzerland.
This article is an open access article
distributed under the terms and
conditions of the [Creative Commons
Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Electron cyclotron resonance (ECR) ion sources are often used at large accelerator facilities to produce high-current, highly-charged ion beams [1]. Part of the popularity of these sources is owed to the fact that ion beams can be produced from any material that can be injected into the source plasma without destroying the plasma. Ion beams are easily produced from materials injected as gases, and from metals and other room temperature solids by heating those materials to temperatures where they can outgas into the plasma.

An ECR ion source has at its heart a magnetically confined plasma. Confinement along the beam extraction axis is typically provided by solenoids at each end of the plasma, while a magnetic multipole (usually sextupole) is used for radial confinement. The resulting

field has a minimum at the source center, and it increases in magnitude in all directions away from that center. Injected microwaves are used to raise the energy of confined plasma electrons through resonant heating on closed surfaces of constant magnetic field magnitude surrounding the plasma center.

The fully-superconducting ECR ion source VENUS at Lawrence Berkeley National Laboratory (LBNL) is one of the world's most advanced and highest-performing ECR sources [2]. The magnetic fields used to confine VENUS' plasmas are as high as 4 T axially and 2.1 T radially, and injected microwaves of 28 and 18 GHz are used to energize plasma electrons. VENUS's success at producing high current beams (e.g., 20 mA He^+ , 4.7 mA O^{6+} , 395 μA U^{34+} , etc.) made it the prototype ion source for the United States' Facility for Rare Ion Beams (FRIB), where a near-copy is in service and two others are being constructed.

The VENUS ion source is the primary injector for LBNL's 88-Inch Cyclotron, and ECR ion sources are in service for many ion accelerators around the world. It is typically required that these sources not only deliver a requested quantity of ion beam to an accelerator to meet experimental or programmatic needs, but that the beam delivered is stable. The quality factor for stability is often standard deviation, and accelerators typically require relative stability in the low single-digit percent range.

Optimization of extracted beam current and the maintenance of stability of an ion source like VENUS requires the manipulation of approximately 10–15 control parameters, including magnets, injected microwave power and electrodes. The physics governing the plasma and its relation to the control parameters is complicated and constructing a reliable analytical or numerical model, able to make precise predictions, has not been achieved to our knowledge. With no means of modeling its behavior, the source is considered to be a black box for further discussion. Analysis of logged data over the past few years revealed that the system is not only non-linear, but also non-invariant, e.g., the optimal settings for production of a given beam species varies over time. This non-invariant behavior can be explained by many factors that perturb the system and are not easily observable. For example, material from previous ion source runs can be adsorbed on the plasma chamber walls and later desorbed into the plasma at some decaying rate. Additionally, small temperature changes can affect desorption and flow rates for gas valves.

This paper provides the community with a Bayesian optimizer that can be used on a non-invariant dynamic system with control constraints. We will see that although constrained Bayesian optimization has been well studied, non-invariant systems create an objective function that is modified through time in an unpredictable manner. The control constraint is also unpredictable and evolves through time. The new method that we present is specifically designed to track the evolution of these functions and maintain system optimization through time.

This paper is organized as follows: in Section 2, we contextualize the study by exploring related work, in Section 3, we provide an overview of Bayesian optimization, the use of Gaussian processes and acquisition functions. In Section 4, we present how Bayesian optimization can be adapted to deal with moving maxima in a dynamic system and we develop a novel approach to deal with both functioning and control constraints in these cases. In Section 5, we present a comparative study evaluating the performance of the method with other methods adapted from the literature. Section 6 will discuss the results and the first steps to apply the method to ECR ion source control.

2. Related Work

In the general case of a system optimization, one usually wants to find the input parameters that optimize a certain variable that can be described by a function called the objective function. In the case of the VENUS ECR ion source, the objective function is

the output beam current of the species of interest, which we aim to maximize, while the control function is a measure of stability derived from the beam statistics that must be maintained under a certain threshold. When the derivative of the objective function is analytically known throughout the operation space, the extrema can be found where this derivative is zero. However, when there is no analytic knowledge of the derivative of the function, the search for the optimum relies on evaluating the function in several places and attempting to move toward the optimum. Even when there is no analytical solution, if the problem is convex, there is only one optimum and a gradient descent method [3] performs well as it locally approximates the derivative of the function and moves toward the unique optimum where the gradient becomes null. When the system is not convex, there can be several solutions where the gradient becomes null. In this case, optimization aims at finding the global optimum and cannot be satisfied by simply finding a null gradient. Common solutions involve methods using several initiations of optimizers that will each find an optimum [4], and then select the optimum. Markov chains are another family of solutions that are guaranteed to find the global optimum asymptotically [5]. These different methods, when applied to a non-convex problem, need to evaluate the objective function many times before finding the optimum, and this can be an issue when the evaluation of the function is costly with respect to time, system availability, and/or actual cost.

Bayesian optimization is a powerful approach whose tools are used in many different fields, including pharmaceutical development, deep-learning model development and aerospace engineering [6–10]. In these cases, Bayesian optimization is a good option as it can find a global optimum with a relatively low number of function evaluations. This optimization process leads to the parameters that will optimize the function, but sometimes the optimized function itself can unpredictably evolve with time and the optimum solution can change, as in the case of the aerodynamics of rotary wings [11]. This would be the case of a non-invariant system and would require it to be frequently re-optimized. This is similarly the case for the operation of the VENUS ion source. In the case of a non-invariant system, analytical solutions for linear and convex systems have been demonstrated [12,13]. In the case of the search for the time-varying global optimum of a non-convex system, it has been shown that it is possible to use a Bayesian optimizer to track the optimum as it moves with time [14].

In this work, with the application case of the VENUS ECR ion source in mind, we use a Bayesian optimization technique for dynamic systems, but we add functioning and control constraints. The functioning constraint is based on the fact that the system would no longer be functional (the ion beam not usable by the accelerator) if minimum beam currents are not met for too long. In this case, when navigating the operation space to maximize output, settings that produce currents under the prescribed threshold should be avoided. In a static context, methods have been developed based on an optimal lower confidence bound to safely find the optimum of a function with a threshold constraint [15–17]. This type of method has been applied in other areas, including robotics [18] and brain simulations [19]. In [20], the authors develop a method using two different Gaussian processes for both the objective function and the constraint from which they derive an expected constrained improvement as the acquisition function. However, the extension of this work to a dynamic system problem is not straightforward. In our case, we are interested in a dynamic system, so we developed a new method to track a moving optimum of a time-variant system, maximizing its efficiency while also adding a control constraint to the system. For the ion source, not only must the output beam current be kept above some minimum, but that current needs to remain stable. Practically, the stability of the ion beam is measured by the relative standard deviation of the measured beam once the system is in a settled state. Avoiding regions of instability is necessary for both the beam user and for the safety of

the ion source, since instabilities can lead to kilowatts of injected microwave power being absorbed by sensitive parts. This control variable is statistically independent from the objective function we try to optimize but is necessary for the system to function properly. We use a variation of the lower confidence bound to determine the safe area, using the combination of two Gaussian processes to determine this space, and then maximize the acquisition function in the restricted safe area.

3. Bayesian Optimization

When optimizing an unknown objective function $f(x)$ with $x \in \mathbb{R}^D$ (assuming a D -dimensional parameter space), we are looking for a solution to:

$$x^* = \operatorname{argmax}_{x \in \mathbb{R}^D} (f(x)) \quad (1)$$

Bayesian optimization is a class of methods that builds a surrogate model for an unknown $f(\cdot)$ based on previous data and a prior on the function auto-correlation. Using such a surrogate model allows us to conduct our optimization despite the limited access or cost to directly evaluate the objective function $f(\cdot)$. In essence, the Bayesian optimizer describes the function at any coordinate by a probabilistic distribution built on known data. The surrogate model is most often made by a Gaussian process [21]. The Gaussian process describes the value of the objective function $f(\cdot)$ at any coordinates by a Gaussian distribution. Hence, at each iteration j the value of the objective function at every coordinate x_i is entirely described by the mean of the distribution estimator $\widehat{f(x_i^j)}$ and its standard deviation $\sigma_{f(x_i^j)}$.

In order to provide the statistics of $f(x_i^j) \forall i, j$, the Gaussian process uses a function to define the auto-correlation between every pair of coordinates. This function, also called a kernel function, determines how precisely the Gaussian process can predict $f(\cdot)$ at specific coordinates given the values already known.

The most used kernel functions are stationary and isotropic, meaning that the correlation between two values of $f(x_1^j)$ and $f(x_2^j)$ only depends on the distance between x_1^j and x_2^j . This distance is often called the radius, and this class of kernels is known as radius basis functions, noted $k(r)$. As a correlation function, $0 \leq k(r) \leq 1 \forall r$ and $k(0) = 1$. We also need a function that describes the increasing uncertainty as the distance grows from the coordinates where $f(\cdot)$ has actually been evaluated. Common functions used are the squared exponential and the Matérn kernel.

Even if these two functions (squared exponential and Matérn) have similar behavior, the Matérn kernel parametrization allows to separate tuning of the smoothness of the function and the maximum distance over which values remain highly correlated near a known point, thanks to the additional parameter ν . The Matérn kernel has the flexibility to reduce the predictability close to previous evaluations, hence reducing the confidence in the information yield by the measure locally, while at the same time conserving a comparable correlation as a square exponential function at a greater distance.

Once the Gaussian process provides an initial statistical model of the objective function $f(\cdot)$, the Bayesian optimizer uses an acquisition function $a(x)$ to identify the most beneficial point for the next evaluation. The parameters of this acquisition function are entirely determined by the current model. Because for VENUS both the Gaussian process and the acquisition function are computationally simple to evaluate compared to the evaluation of $f(\cdot)$, users can afford to choose robust optimizers that require a high number of acquisition function evaluations to ensure the best possible next step is taken. The acquisition function balances the need to evaluate the function near the expected

maximum (exploitation) against the need to explore the parameter space in less-covered areas (exploration). The common acquisition functions ($a(x)$) used in literature are the expected improvement (EI), the probability of improvement (PI) and the lower confidence bound (LB) [22,23]. These are defined below.

•

$$EI : \sigma_{f(x)}(u\Phi(u) + \phi(u)) \quad (2)$$

with $u = \frac{f_{\min} - \widehat{f(x)} - v}{\sigma_{f(x)}}$ and v is the exploration bias and $\sigma_{f(x)}$ the standard deviation of the Gaussian distribution given by the Gaussian process regressor for $f(\cdot)$ at x .

•

$$PI : \Phi\left(\frac{T - \widehat{f(x)}}{\sigma_{f(x)}}\right) \quad (3)$$

where T is the target beyond which we wish to improve the current optimum

•

$$LB : \widehat{f(x)} - \kappa\sigma_{f(x)} \quad (4)$$

where κ is a tuning parameter determining the lower confidence bound.

$\Phi(\cdot)$ is the cumulative density function and $\phi(\cdot)$ is the probability density function of a variable $\sim \mathcal{N}(0, 1)$. The expected improvement is particularly interesting because of the v parameter that allows the user to adjust the balance between exploration and exploitation.

At every iteration of the optimization process, the Gaussian process provides statistics about the objective function given the previous evaluations and the chosen kernel. The acquisition function is then optimized and the objective function evaluated. The Gaussian process is updated and the optimizer iterates. Bayesian optimization allows the user to find the optimum of the objective function in fewer iterations/evaluations than other optimization techniques [21].

4. Non-Invariant Dynamic Systems with Constraints

ECR ion sources are examples of non-invariant dynamic systems, which may need to be re-optimized continuously. They also need to operate under certain constraints. The first constraint considered, which we call the functioning constraint, is a lower boundary threshold for the objective function. In the case of VENUS, the output beam current. The second constraint, which we call the control constraint, is a constraint of stability. For VENUS, this is taken as the relative standard deviation of the objective function (beam current). In this work, we will consider the control constraint as independent of the objective function, as the empirical evidence from observations on the source itself does not show mutual information between the beam current and its standard deviation. Bayesian optimization has been used to track the optimum of a time-varying objective function [14] and the consideration of a functioning constraint in a static context has been well studied in the literature and optimization methods respecting such a constraint have been developed [15,16]. The specificity of our particular problem requires a method that can both track the optimum of a time-varying function while taking into account the functioning constraint, but also to take into account the additional independent control constraint to keep the optimization safe.

4.1. Non-Invariant Dynamic Bayesian Optimization

In the dynamic case, because we need to find the optimum at each time sample $t = t_c$, the problem becomes:

$$(x|t = t_c)^* = \underset{x \in \mathbb{R}^{D-1}}{\operatorname{argmax}}(f(x, t|t = t_c)) \quad (5)$$

The problem is set in a space containing all $D - 1$ dimensions of x and one additional dimension for time, but at every step, the goal is to optimize only the current time slice at $t = t_c$ where t_c is the current time.

In their work, Ref. [14], a Gaussian process with a time dimension and a dedicated kernel function for time is used. The idea is to consider the time when the previous evaluations have been made and transform this time into a distance which, with the attenuation of the kernel, will weigh more heavily recent evaluations to give more accurate information about the present distribution. At every step of the way, the Bayesian optimizer tries to determine the optimum for the next sample. The optimization process is restricted to the next sample, hence the present $t = t_c$. For causality reasons, it is impossible to evaluate $f(x, t)$ for $t \neq t_c$ and only the previously evaluated values are accessible.

4.2. Lower Confidence Bounds and Constraints

Existing safe (safety in this context refers to limiting evaluations of the objective function that do not meet the constraint requirements) optimization methods [15,16] are considering a functioning constraint and build a set of previously allowed evaluations and then determine a permissible space based on the lower confidence bound and the Lipschitz-continuity principle to gradually extend the allowed exploration space. The optimized lower-bound acquisition function is:

$$x_n = \underset{x \in S_n}{\operatorname{argmax}} \left(\widehat{f_{n-1}}(x) + \beta_n^{1/2} \sigma_{f_{n-1}(x)} \right) \quad (6)$$

where the subset S_n is the allowed area determined by the ensemble of coordinates that are sufficiently close to a known evaluation point that is above the threshold h . $\beta_n^{1/2}$, as shown in [16], is an optimized κ calculated taking the functioning risk (of evaluating the system under the threshold) into account. In [16], β_n is defined as:

$$\beta_n = 2B + 300\gamma_n \log^3(n/\delta) \quad (7)$$

where δ is the allowed risk to fall under the functioning threshold, B is a bound on the Reproducing Kernel Hilbert Space norm of the objective function [24] and γ_n is the degree of freedom associated with the kernel function [16].

The subset S_n is defined mathematically as:

$$S_n = \bigcup_{x \in S_{n-1}} \{x' \in D \mid l_n(x) - Ld(x, x') \geq h\} \quad (8)$$

where $l_n(x)$ is the confidence bound at x , L is the Lipschitz constant and $d(\cdot)$ is a distance function.

This approach gives very good results in a static system, but in a dynamic system, $f(x, t_i) \neq f(x, t_j)$ and a coordinate x evaluated at an instant t_i may not be above the functioning threshold anymore at a later time t_j .

4.3. Proposed Method for Dynamic Optimization with Constraints

The alternative that we propose is to determine the safe area where the time interval between iterations becomes important due to the time-varying behavior of the objective function, based on the statistics given by two dynamic Gaussian processes at the current time t_c , one for the functioning and one for the control constraint, by using the time kernel. Both the objective function and the constraint function are considered black boxes. We use the expected improvement in this new subspace at every iteration to determine the next sample. We can determine h , the threshold under which the objective function is considered

undesirable in terms of the functioning constraint. In this case, we can write the associated probability as:

$$P(f(x, t_c) > h) = \int_{y=\alpha}^{+\infty} \frac{1}{\sqrt{2\pi}\sigma_{f(x, t_c)}} \exp\left(-\frac{1}{2}\left(\frac{y - \widehat{f(x, t_c)}}{\sigma_{f(x, t_c)}}\right)^2\right) dy, \quad (9)$$

where $\widehat{f(x, t_c)}$ and $\sigma_{f(x, t_c)}$ are the statistics given by the Gaussian process associated with $f(\cdot)$ at the coordinates x and time $t = t_c$.

We can proceed analogously with the control constraint, in our case the stability measure:

$$P(g(x, t_c) < \alpha_c) = \int_{y=-\infty}^{\alpha_c} \frac{1}{\sqrt{2\pi}\sigma_{g(x, t_c)}} \exp\left(-\frac{1}{2}\left(\frac{y - \widehat{g(x, t_c)}}{\sigma_{g(x, t_c)}}\right)^2\right) dy, \quad (10)$$

with $g(\cdot)$ being the independent control function that does not need to be optimized but is required to stay below a certain threshold, α_c .

Having both $P(f(x, t) > h)$ and $P(g(x, t) < \alpha_c)$, we define γ and γ_c as the associated maximum acceptable risks of sampling in an undesirable or unsafe area.

With these probabilities defined, the safe-mode acquisition function $a(x)$ becomes:

$$a_{safe}(x) = a(x) \times \mathbb{1}_{P(f(x) > h) > (1-\gamma)} \times \mathbb{1}_{P(g(x) < \alpha_c) > (1-\gamma_c)}, \quad (11)$$

which with the expected improvement acquisition function $a(x)$ gives:

$$a_{safe}(x) = \sigma_{f(x)}(u\Phi(u) + \phi(u)) \times \mathbb{1}_{P(f(x) > h) > (1-\gamma)} \times \mathbb{1}_{P(g(x) < \alpha_c) > (1-\gamma_c)} \quad (12)$$

This new approach is what we call the Non-Invariant Dynamic Systems Optimizer (NIDSO) from here forward. A pseudo-code of NIDSO is available Appendix A. If the acceptable risks are too restrictive, there will be more cases where no solutions are found and where the method switches back to a standard Bayesian optimizer and optimizes a instead of a_{safe} for the current iteration. The search for the maximum of the acquisition function, in addition to avoiding local minima, has to deal with the discontinuities and null areas introduced by the indicator. Because the acquisition function is light to compute, this can be addressed by a large number of initializations. The system itself (VENUS) transitions very slowly in comparison to the computation time required for the acquisition function.

4.4. Adaptation of the Optimized Lower Confidence Bound to a Dynamic System

In order to be able to compare NIDSO to the Lower confidence bound method presented in [15], we adapted it so it would only consider the added time dimension that is not included in the original version. S_n is then modified as follows:

$$S_n(t_c) = \bigcup_{x \in S_{n-1:n-T-1}} \{x' \in D \mid l_n(x) - Ld(x, x') \geq h\} \quad (13)$$

where previous safe samples are removed from the set after a period T .

5. Application to Synthetic Data

In this section we will compare the performances for three methods: first, a standard Bayesian optimizer operating on a dynamic system [14] (Std-BO), second, the adaptation of the Lower-Bound method to dynamic systems with an additional control constraint [15] (LB-BO), and, third, the method we developed based on the modification of the acquisition function by two indicator functions determining where queries are safe, NIDSO. In order

to compare the methods, four criteria are considered: the efficiency, the global failure rate, the optimization failure rate, and the control failure rate. The efficiency is determined by comparing the actual yield to the yield along the optimum path $\sum_t Eff(t) = \sum_t \frac{f(x,t)}{f(x_t^*,t)}$ where x_t^* are the coordinates of the stable optimum at time t . The global failure rate is the percentage of evaluations of the function that either are under the functioning threshold ($f(x,t) < h$) or, at which the control function is over the stability threshold ($g(x,t) > \alpha_c$), or both. The global failure rate is divided into the optimization failure rate and the control failure rate. When $f(x,t) < \alpha$ or $g(x,t) > \alpha_c$, we set $Eff(t) = 0$.

The simulated data is generated to mimic the behavior that we observe based on the data collected from the VENUS ion source. With a moving optimum and the possibility of local maxima and very low values of the beam current (objective function), $f(x,t)$ is defined as a series of randomly generated positive and negative 2D Gaussians. Regions of instability, $g(x,t)$, are also modeled and generated by a series of positive 2D Gaussians.

5.1. Example of the Methods Behavior

In the following, one example case is taken to demonstrate the behavior of the three models under consideration. In this scenario, each method tries to find the x that maximizes $f(x,t)$ for every value of t while only having access to the previous evaluations of the function. The results are shown in Figure 1. The surface shows the location of the maxima as well as the areas that are forbidden either because the objective function values are under the functioning threshold, or control function values which exceed the control threshold.

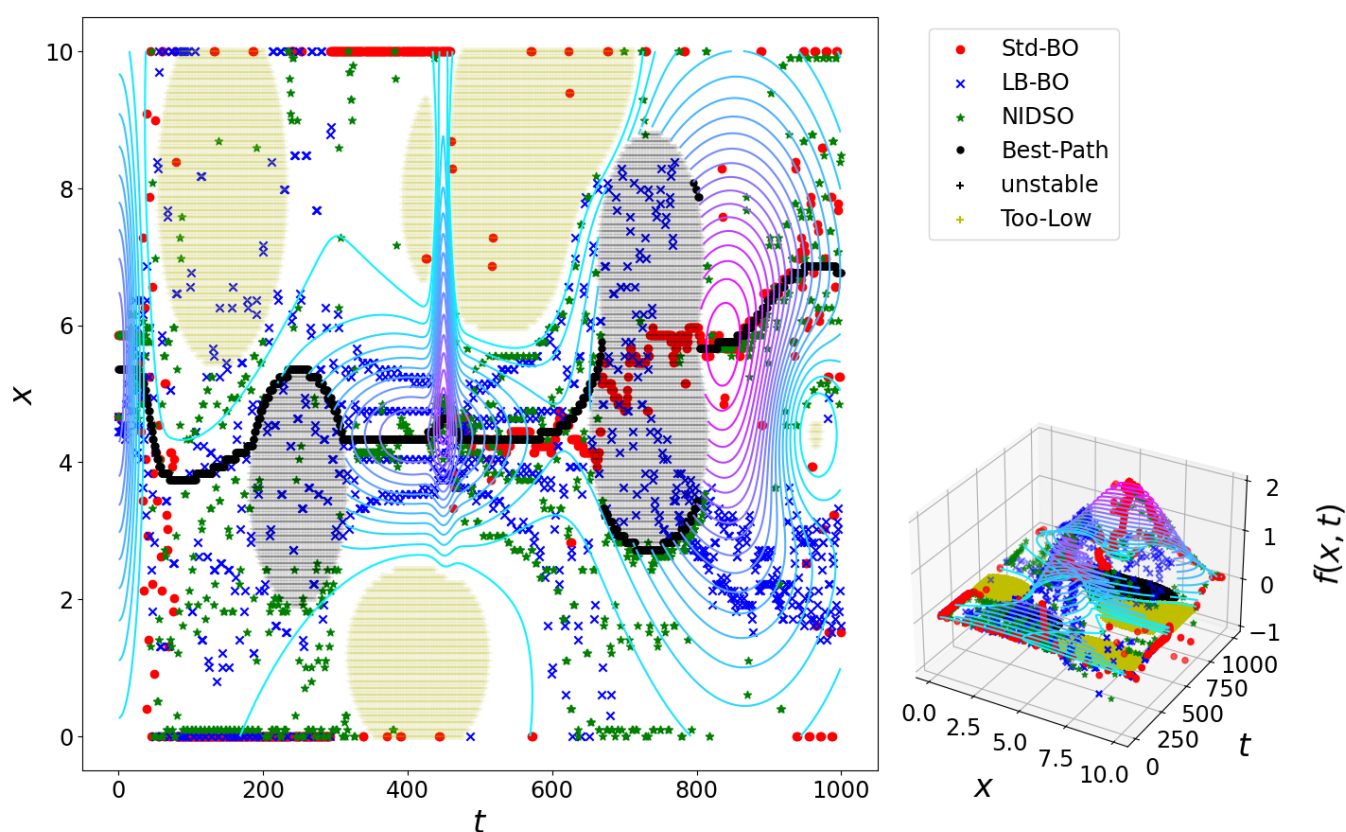


Figure 1. Optimizations of a single parameter, x , of the objective function $f(x,t)$. The objective function is represented by surface contours that vary with time t , while unstable areas are indicated in black and regions below the functioning threshold are yellow (labeled too low). The bottom right corner shows a 3D representation of $f(x,t)$ as a visual aid of the exploration space.

In Figure 1, the three methods begin their optimizations at $t = 0$ and for each of the 1000 time steps all three optimizers work to stay as close as possible to the maximum (thick black markers labeled Best Path) by optimizing following x at every given t $x^* = \underset{x}{\operatorname{argmax}}(f(x, t = t_c))$. At every time, the optimizers only know their own past evaluations (lower values of t). At time $t = 0$, all the methods initialize close to a local optimum for x , but the function value drops at this location as time advances, the methods start exploring to find a new optimum. Around $t = 200$ there is an area where the high values of x (~ 7) correspond to $f(x, t)$ under the functioning threshold, but all three methods manage to limit evaluations in that area. The area where the control function exceeds the control threshold around $t = 300$ and $x = 3$ is also avoided quite well. While the standard Bayesian optimizer takes time to recover and find the new optimum around $t = 600$, LB-BO and NIDSO find it as soon as it is no longer in the out-of-control area. When the second out-of-control area appears between $t = 800$ and $t = 900$, both the standard Bayesian optimized and the LB-BO have trouble avoiding it. Indeed, the mitigation of the control function in the case of LB-BO is outweighed by the high value of the objective function. There is some exploration to find other solutions at the beginning of this region when the objective function is not as high, but LB-BO rapidly provides solutions that are in the out-of-control area. While it is possible to give more weight to the control function in the penalization of LB-BO, which would result in a better avoidance of out-of-control areas, this would ultimately make it harder for it to find an optimum that is close to an unstable region. NIDSO, on the other hand, despite a few queries in the out-of-control area, maintains its solution almost exclusively on the edge of the acceptable area.

Figure 2 shows the values taken by the objective function as a function of time compared to the optimal path. We can observe that LB-BO and NIDSO follow a path closer to the optimum than the standard Bayesian optimizer. LB-BO and Std-BO show very few values between $t = 600$ and $t = 800$ because they are tracking too closely the optimum of the objective function, despite the fact that this region is an out-of-control area. NIDSO, on the other hand, does a good job avoiding this area and staying as close as possible to the best solution.

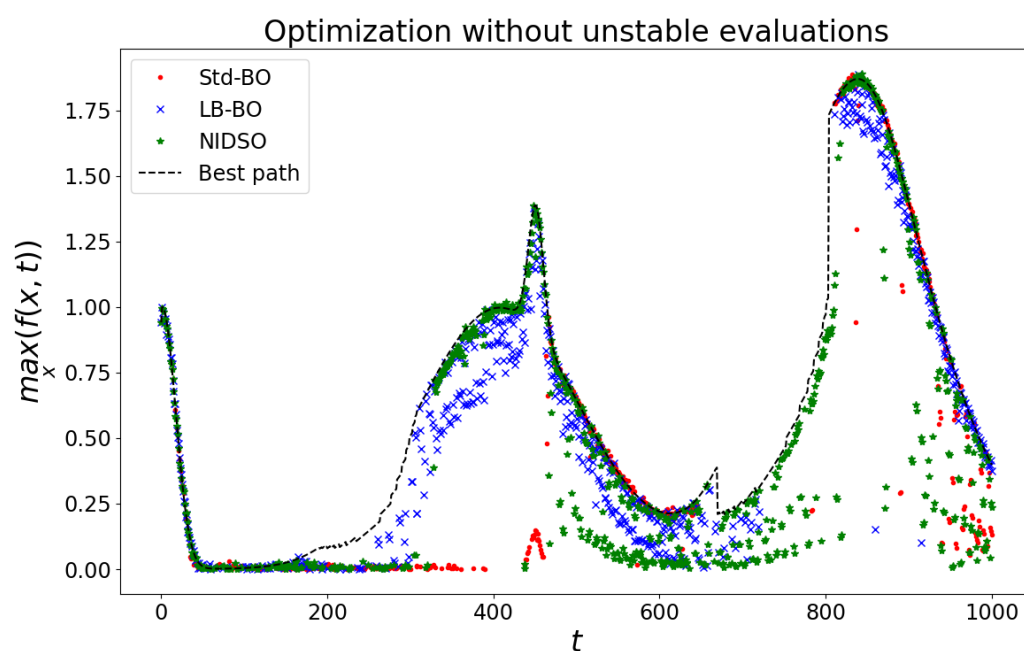


Figure 2. Evaluations of the objective function of the different methods along time on the problem presented in Figure 1.

Figure 3 shows how the different methods managed to stay out of the areas considered as unstable. In the first area from $t = 150$ to $t = 350$, both LB-BO and NIDSO sample in the unstable region, while the Std-BO is so far from the optimal area that it is not impacted. This first unstable area is difficult to avoid because the values of the objective function in stable regions are quite low. The second unstable area between $t = 600$ and $t = 800$ is right where the optimum of the objective function is, and it makes it almost impossible to avoid for Std-BO and LB-BO methods, while NIDSO finds acceptable solutions at the edge of the unstable area almost the entire time.

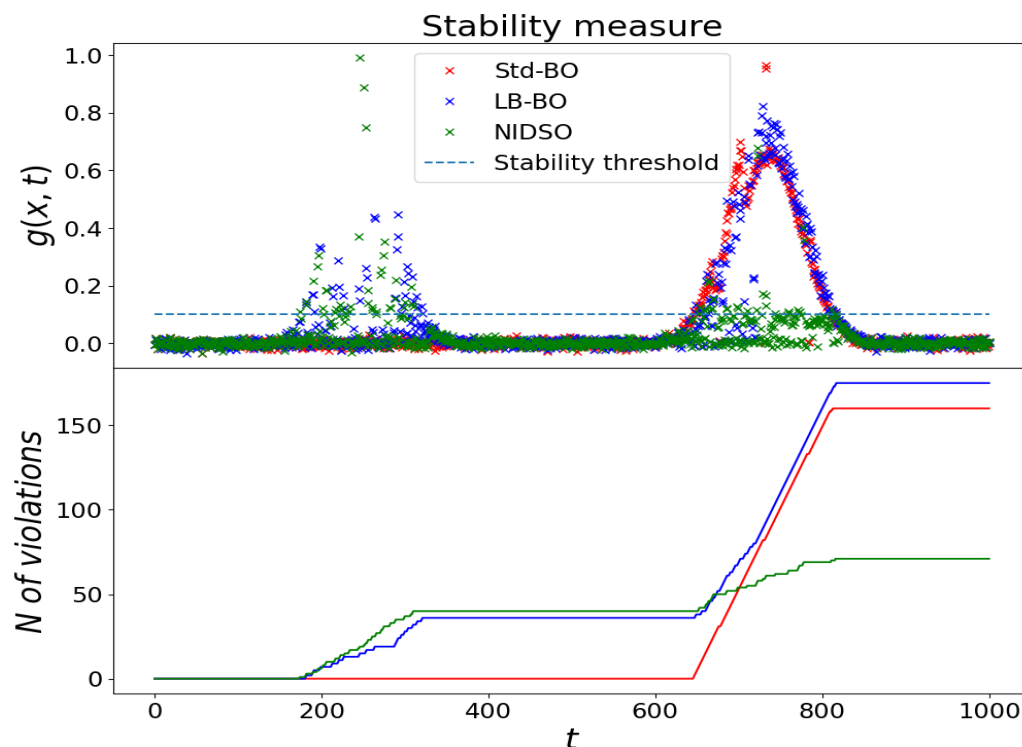


Figure 3. Evaluations of the control function of the different methods along time on the problem presented in Figure 1.

5.2. Quantitative Tests

Expanding on Section 5.1 we are interested in global statistics evaluating the performances of the different methods. Their tuning is optimized to give similar control failure rates and then tested on a different series of random scenarios similar to the one presented in Section 5.1. The NIDSO kernel parameters for these tests are (length = 0.1, smoothness = 1.5), and the exploration exploitation trade-off is $\nu = 0.5$. For LB-BO, the kernel parameters are the same and β_n is adjusted in each case to have a similar control failure rate as NIDSO.

Figures 4 and 5 show the results in terms of efficiency and failure rates of the three methods on 100 random generations of scenarios per point, where each scenario contains 1000 time samples. In the scenarios summarized in Figure 4, we generated 5 positive Gaussians with random locations and random widths to set the maxima and a varying number of negative Gaussian distributions with random location and width which depresses the objective function under the functioning threshold. The control function in these cases is set by 3 Gaussians with random locations and widths that determine the area considered as unstable.

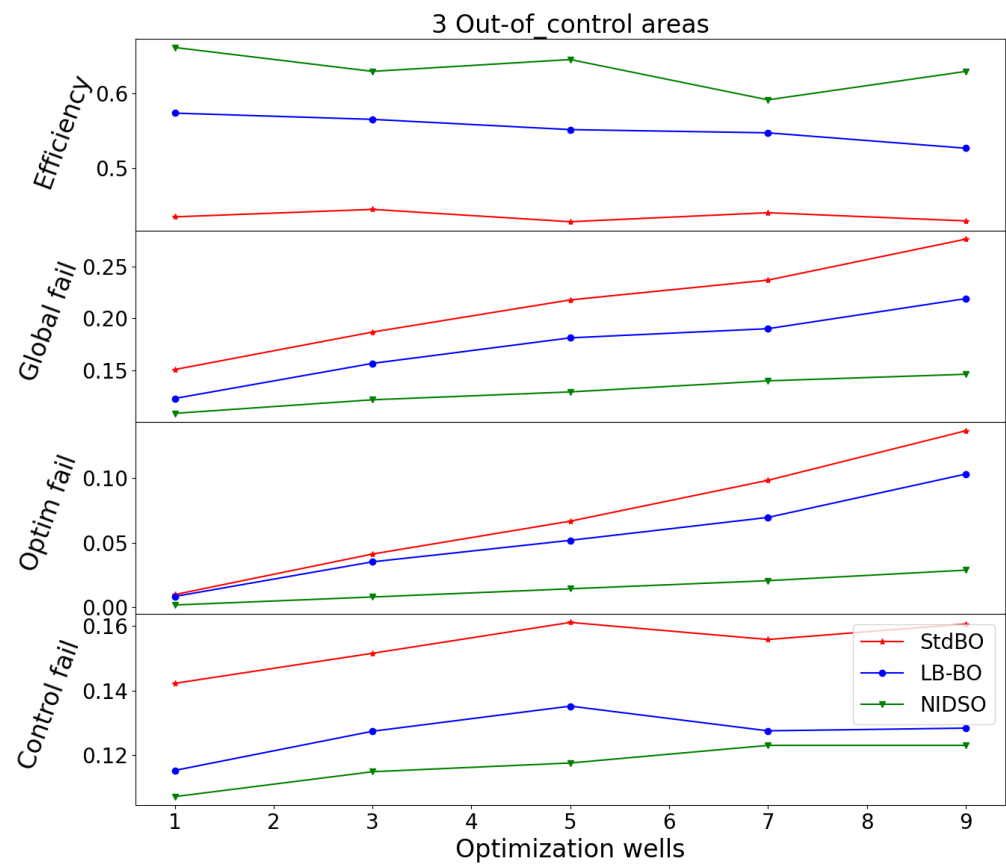


Figure 4. Results with 3 areas out-of-control and a varying number of negative Gaussians in the objective function.

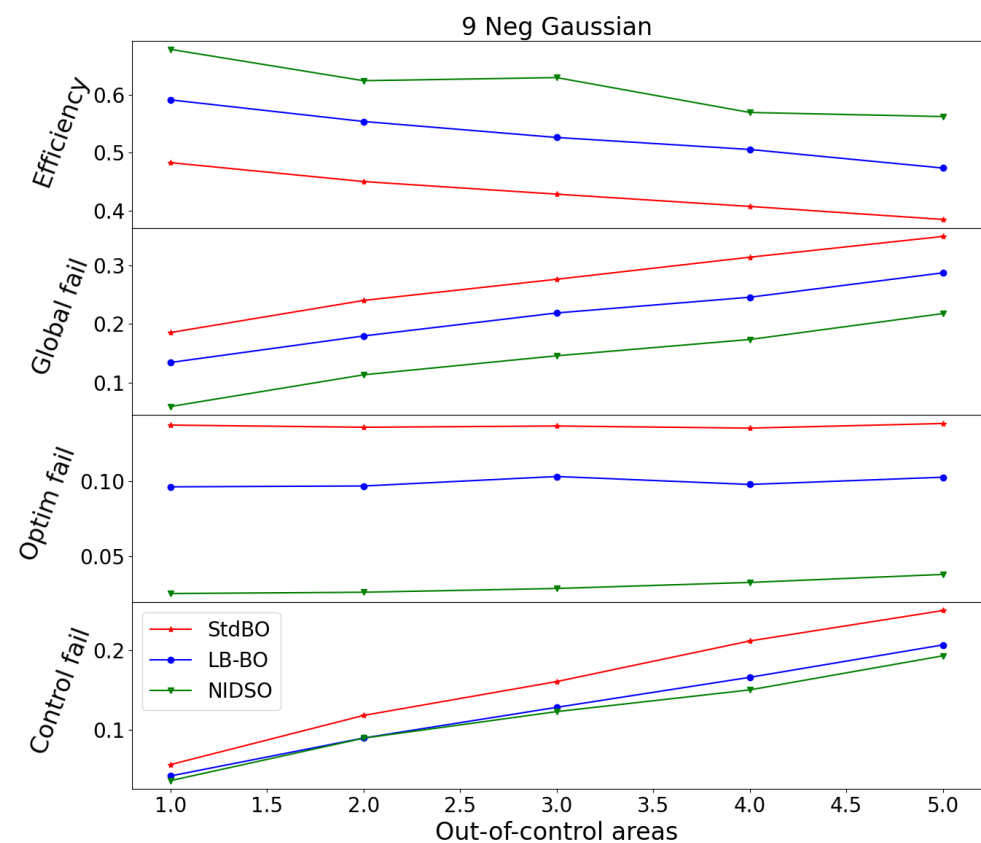


Figure 5. Results with 9 Negative Gaussians and a varying number of areas out-of-control.

Both LB-BO and NIDSO presented here improve on the results provided by a standard Bayesian optimizer that does not take into account the functioning threshold and control function. The results of LB-BO and NIDSO show higher efficiency with lower failure rates. The results presented in Figure 4 are summarized in Table 1. LB-BO and NIDSO are tuned to have similar (less than 1.5% difference) control failure rates (bottom plot in Figure 4). This is confirmed by the Student test results presented in Table 2 where the mean of LB-BO and NIDSO control failure are almost identical. From these results, it is evident that the efficiency of NIDSO is always higher, while the optimization failure rate increases more slowly with the increase in the number of negative Gaussians in the objective function. This shows a more robust performance relative to the number of low-value areas in the objective function.

Table 1. Efficiency and failure rates of the compared methods with a various number of optimization wells and three areas out-of-control. In each case, the best performances are in bold text and the worst are in shaded cells.

Efficiency	StdBO	LB-BO	NIDSO	Global Fail	StdBO	LB-BO	NIDSO
1 Opt Well	0.43	0.57	0.66		0.15	0.12	0.11
3 Opt Well	0.44	0.57	0.63		0.19	0.16	0.12
5 Opt Well	0.43	0.55	0.65		0.22	0.18	0.13
7 Opt Well	0.43	0.55	0.59		0.24	0.19	0.14
9 Opt Well	0.43	0.53	0.63		0.27	0.22	0.15
Optim Fail	StdBO	LB-BO	NIDSO	Control Fail	StdBO	LB-BO	NIDSO
1 Opt Well	0.01	0.009	0.001		0.14	0.12	0.11
3 Opt Well	0.04	0.04	0.008		0.15	0.13	0.11
5 Opt Well	0.07	0.05	0.01		0.16	0.14	0.12
7 Opt Well	0.10	0.07	0.02		0.15	0.13	0.12
9 Opt Well	0.14	0.10	0.03		0.16	0.13	0.12

In the scenarios considered for Figure 5, the number of negative Gaussians in the objective function is fixed to 9, while the number of positive Gaussians in the objective function is still 5. The difference in performance between the methods increases with the number of negative Gaussians. In these scenarios, the number of Gaussians of the control function is varied. As before, the LB-BO and NIDSO methods outperform the Std-BO. We can observe that the optimization failure is approximately constant this time as the number of negative Gaussians to avoid is always the same, but the optimization failure rate of NIDSO is around 3%, where LB-BO has an optimization failure rate of approximately 9%—this difference explains the gap in performance on the global failure rate. The efficiency of NIDSO is also approximately 10% better than what LB-BO achieves. It is important to note that LB-BO improves on the Std-BO even though it was not meant to solve this type of problem and that we had to adapt it with Equation (13) to face the new challenges of a dynamic system and an external constraint. The results presented in Figure 5 are summarized in Table 3.

Student *T*-tests on the significance of the Table 1 results are presented in Table 2.

Student *T*-tests on the significance of the Table 3 results are presented in Table 4.

Table 2. *p* value of student *T*-test on sample used to calculate the Efficiency and failure rates of the compared methods with a various number of optimization wells and three areas out-of-control.

Efficiency	StdBO = NIDSO	LB-BO = StdBO	NIDSO = LB-BO	Glob Fail	StdBO = NIDSO	LB-BO = StdBO	NIDSO = LB-BO
1 Opt Well	1×10^{-18}	1×10^{-12}	2×10^{-5}		4×10^{-4}	0.01	0.1
3 Opt Well	3×10^{-13}	2×10^{-12}	2×10^{-3}		4×10^{-8}	5×10^{-4}	7×10^{-5}
5 Opt Well	3×10^{-19}	1×10^{-13}	3×10^{-6}		3×10^{-14}	9×10^{-5}	2×10^{-7}
7 Opt Well	3×10^{-9}	3×10^{-12}	5×10^{-3}		4×10^{-15}	1×10^{-5}	9×10^{-6}
9 Opt Well	1×10^{-19}	2×10^{-12}	2×10^{-9}		6×10^{-24}	4×10^{-7}	2×10^{-9}
Optim Fail	StdBO = NIDSO	LB-BO = StdBO	NIDSO = LB-BO	Cont Fail	StdBO = NIDSO	LB-BO = StdBO	NIDSO = LB-BO
1 Opt Well	7×10^{-8}	0.5	2×10^{-4}		2×10^{-3}	0.01	0.37
3 Opt Well	7×10^{-22}	0.3	2×10^{-8}		7×10^{-4}	0.01	0.17
5 Opt Well	4×10^{-23}	0.01	3×10^{-13}		1×10^{-5}	4×10^{-3}	0.04
7 Opt Well	3×10^{-33}	9×10^{-6}	4×10^{-18}		1×10^{-3}	1×10^{-3}	0.64
9 Opt Well	2×10^{-41}	4×10^{-5}	4×10^{-20}		1×10^{-4}	1×10^{-4}	0.58

Table 3. Efficiency and failure rates of the compared methods with a various number of out-of-control areas and nine optimization wells. In each case, the best performances are in bold text and the worst are in shaded cells.

Efficiency	StdBO	LB-BO	NIDSO	Global Fail	StdBO	LB-BO	NIDSO
1 OC	0.48	0.59	0.68		0.19	0.13	0.06
2 OC	0.45	0.55	0.62		0.24	0.18	0.11
3 OC	0.43	0.52	0.63		0.28	0.22	0.15
4 OC	0.41	0.51	0.57		0.31	0.25	0.17
5 OC	0.39	0.47	0.56		0.35	0.29	0.22
Optim Fail	StdBO	LB-BO	NIDSO	Control Fail	StdBO	LB-BO	NIDSO
1 OC	0.14	0.1	0.03		0.06	0.04	0.04
2 OC	0.14	0.1	0.03		0.11	0.09	0.09
3 OC	0.14	0.1	0.03		0.16	0.13	0.12
4 OC	0.14	0.1	0.03		0.21	0.17	0.15
5 OC	0.14	0.1	0.04		0.25	0.21	0.19

Table 4. *p* value of Student *T*-test on sample used to calculate the Efficiency and failure rates of the compared methods with nine optimization wells. Bold values mark where the T-test fails.

Efficiency	StdBO = NIDSO	LB-BO = StdBO	NIDSO = LB-BO	Glob Fail	StdBO = NIDSO	LB-BO = StdBO	NIDSO = LB-BO
1 OC	6×10^{-15}	2×10^{-15}	1×10^{-4}		3×10^{-31}	6×10^{-7}	1×10^{-14}
2 OC	3×10^{-13}	1×10^{-14}	1×10^{-3}		4×10^{-24}	1×10^{-8}	2×10^{-9}
3 OC	1×10^{-19}	2×10^{-12}	2×10^{-7}		6×10^{-24}	4×10^{-7}	2×10^{-9}
4 OC	5×10^{-13}	1×10^{-11}	1×10^{-3}		1×10^{-20}	4×10^{-8}	2×10^{-8}
5 OC	2×10^{-15}	9×10^{-11}	1×10^{-5}		2×10^{-20}	2×10^{-8}	2×10^{-7}
Optim Fail	StdBO = NIDSO	LB-BO = StdBO	NIDSO = LB-BO	Cont Fail	StdBO = NIDSO	LB-BO = StdBO	NIDSO = LB-BO
1 OC	8×10^{-43}	1×10^{-7}	1×10^{-22}		1×10^{-3}	0.01	0.30
2 OC	1×10^{-41}	9×10^{-7}	4×10^{-21}		2×10^{-3}	7×10^{-5}	0.98
3 OC	2×10^{-41}	5×10^{-5}	4×10^{-20}		2×10^{-4}	1×10^{-4}	0.58
4 OC	4×10^{-38}	9×10^{-7}	2×10^{-19}		2×10^{-6}	1×10^{-5}	0.17
5 OC	9×10^{-39}	1×10^{-7}	2×10^{-18}		4×10^{-6}	2×10^{-5}	0.26

Tables 2 and 4 show that the samples used on the comparative test in this study have significantly different mean values under a Student *t*-test with a risk of 5%. Some exceptions can be noted in Table 2: The global fail rate of NIDSO and LB-BO are not differentiable with 1 optimization well. This is due to the fact that the Optim fail rate is very low compared to the control fail rate in this case, and the control fail rate of both methods has been calibrated to match in mean value. Another exception is for 1 or 3 optimization, where it is impossible to differentiate the performances of LB-BO and StdBO in terms of optimization failure, and this is consistent with the results observed in Figure 4. In both tests, control rates of NIDSO and LB-BO are not differentiable because the two methods have been tuned to have similar control failure rates, allowing an easier comparison of the base of efficiency results.

6. Summary and Conclusions

Non-invariant systems with control constraints can be challenging to optimize while achieving optimal performance and maintaining good control at the same time. In this paper, we presented a novel method based on Bayesian optimization for dynamic systems under control constraints. For this class of problems, which include the operation of ECR ion sources, we have proposed the Non-Invariant Dynamic System Optimizer (NIDSO) which determines the acceptable solution space as a priority and shows the best results in terms of low failure rate both for control and optimization, but also in terms of high efficiency overall. The conducted performance and significance tests show that NIDSO significantly outperforms the preexisting methods it is compared to. NIDSO is also easy to tune, since aside from the meta parameters of the Bayesian optimizer, one only has to set up the thresholds and the acceptable risk limits. NIDSO has been used for the optimization of VENUS and showed improvement both in the speed to reach the maximum of the beam and in avoiding instabilities. The tests so far have been conducted on a time period of a few hours, which is too short to observe the drift due to the non-invariance of the source. Future tests will be able to quantify the improvement of the system maintained stable and optimized over a few days. These tools should provide new possibilities of automated control of non-invariant black box systems such as ECR ion sources, but can also provide interesting solutions in other automated control applications for fields like robotics.

Author Contributions: Conceptualization, V.W., C.M.C., H.L.C., Y.S.L., M.S. and D.T.; methodology, V.W.; software, V.W.; validation, V.W.; analysis, V.W.; writing, V.W.; review, V.W., C.M.C., H.L.C., Y.S.L., M.S. and D.T.; supervision, H.L.C. and D.T.; application, V.W. and D.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the U.S. Department of Energy, Office of Science, Office of Nuclear Physics under Contract No. DE-AC02-05CH11231.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ECR	Electron Cyclotron Resonance
Std-BO	Standard Bayesian Optimizer
LB-BO	optimized Lower-Bound Bayesian Optimizer
NIDSO	Non-Invariant Dynamic Systems Optimizer
LBNL	Lawrence Berkeley National Laboratory
FRIB	Facility for Rare Ion Beams

EI	Expected Improvement
PI	Probability of Improvement
LB	Lower Confidence Bound

Appendix A. NIDSO Pseudo-Code

Algorithm A1 NIDSO Pseudo-code

```

1:  $x_0 \leftarrow U_{[0;1]}^D$ 
2: Output:  $x_0$ 
3: Input:  $f_0 \leftarrow f(x_0, t_0)$ 
4: Input:  $g_0 \leftarrow g(x_0, t_0)$ 
5:  $n = 0$ 
6:  $Klen = 0.1$  Kernel Length
7:  $s = 1.5$  Kernel smoothness
8:  $\nu = 0.5$  Exploration exploitation trade off
9: while System runs do
10:    $GPO \leftarrow$  Gaussian Process Regressor of  $f$  over  $x$  and  $t$ 
11:    $GPC \leftarrow$  Gaussian Process Regressor of  $g$  over  $x$  and  $t$ 
12:   Optimize  $a_{safe}$  on  $x$  at  $t_c$  current time
13:   if  $argmax_{x,t=t_c}(a_{safe}(x))$  is Defined then
14:      $x_{n+1} \leftarrow argmax_{x,t=t_c}(a_{safe}(x))$ 
15:   else
16:      $x_{n+1} \leftarrow argmax_{x,t=t_c}(a(x))$ 
17:   end if
18:   Output:  $x_{n+1}$ 
19:    $t_{n+1} \leftarrow t_c$ 
20:   Input:  $f_{n+1} \leftarrow f(x_{n+1}, t_{n+1})$ 
21:   Input:  $g_{n+1} \leftarrow g(x_{n+1}, t_{n+1})$ 
22:    $n \leftarrow n + 1$ 
23: end while

```

References

1. Sun, L.; Zhao, H.; Lu, W.; Guo, J.; Yang, Y.; Jia, H.; Lu, L.; Wu, W. Review of highly charged ion production with ECR ion source and the future opportunities for HCI physics. *X-Ray Spectrom.* **2019**, *49*, 47–53. <https://doi.org/10.1002/xrs.3049>.
2. Lyneis, C.; Leitner, D.; Leitner, M.; Taylor, C.; Abbott, S. The third generation superconducting 28 GHz electron cyclotron resonance ion source VENUS (invited). *Rev. Sci. Instrum.* **2010**, *81*, 02A201. <https://doi.org/10.1063/1.3271135>.
3. Adby, P. *Introduction to Optimization Methods*; Springer Science & Business Media: Dordrecht, The Netherlands, 2013.
4. Boese, K.D.; Kahng, A.B.; Muddu, S. A new adaptive multi-start technique for combinatorial global optimizations. *Oper. Res. Lett.* **1994**, *16*, 101–113. [https://doi.org/10.1016/0167-6377\(94\)90065-5](https://doi.org/10.1016/0167-6377(94)90065-5).
5. Robert, C.P.; Casella, G. The Metropolis—Hastings Algorithm. In *Monte Carlo Statistical Methods*; Springer: New York, NY, USA, 2004; pp. 267–320. https://doi.org/10.1007/978-1-4757-4145-2_7.
6. Frazier, P.I. Bayesian Optimization. In *Recent Advances in Optimization and Modeling of Contemporary Problems*; INFORMS: Catonsville, MD, USA, 2018; Chapter 11, pp. 255–278. <https://doi.org/10.1287/educ.2018.0188>.
7. Sano, S.; Kadowaki, T.; Tsuda, K.; Kimura, S. Application of Bayesian Optimization for Pharmaceutical Product Development. *J. Pharmaceutical Innov.* **2020**, *15*, 333–343. <https://doi.org/10.1007/s12247-019-09382-8>.
8. Wang, K.; Dowling, A.W. Bayesian optimization for chemical products and functional materials. *Curr. Opin. Chem. Eng.* **2022**, *36*, 100728. <https://doi.org/10.1016/j.coche.2021.100728>.
9. Snoek, J.; Larochelle, H.; Adams, R.P. Practical Bayesian Optimization of Machine Learning Algorithms. In *Proceedings of the Advances in Neural Information Processing Systems*; Pereira, F., Burges, C., Bottou, L., Weinberger, K., Eds.; Curran Associates, Inc.: New York, NY, USA, 2012; Volume 25.

10. Lam, R.; Poloczek, M.; Frazier, P.; Willcox, K.E. Advances in Bayesian Optimization with Applications in Aerospace Engineering. In Proceedings of the 2018 AIAA Non-Deterministic Approaches Conference, Kissimmee, FL, USA, 8–12 January 2018. <https://doi.org/10.2514/6.2018-1656>.
11. Quero, D. Application of Time-Variant Systems Theory to the Unsteady Aerodynamics of Rotary Wings. *AIAA J.* **2024**, *62*, 3728–3748. <https://doi.org/10.2514/1.J064078>.
12. Martin-Sanchez, J.M. Adaptive Control for Time-Variant Processes. *Int. J. Control* **1986**, *44*, 315–329. <https://doi.org/10.1080/00207178608933601>.
13. Freeman, E.A. On the optimization of linear, time-variant, multivariable control systems using the contraction mapping principle. *J. Optim. Theory Appl.* **1969**, *3*, 416–443. <https://doi.org/10.1007/BF00929357>.
14. Nyikosa, F.M.; Osborne, M.A.; Roberts, S.J. Bayesian Optimization for Dynamic Problems. *arXiv* **2018**, arXiv:1803.03432. <https://doi.org/10.48550/arXiv.1803.03432>.
15. Han, G.; Jeong, J.; Kim, J.H. Adaptive Bayesian Optimization for Fast Exploration Under Safety Constraints. *IEEE Access* **2023**, *11*, 42949–42969.
16. Sui, Y.; Gotovos, A.; Burdick, J.; Krause, A. Safe exploration for optimization with Gaussian processes. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 997–1005.
17. Sui, Y.; Zhuang, V.; Burdick, J.; Yue, Y. Stagewise Safe Bayesian Optimization with Gaussian Processes. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; Volume 80, pp. 4781–4789.
18. Berkenkamp, F.; Krause, A.; Schoellig, A.P. Bayesian optimization with safety constraints: Safe and automatic parameter tuning in robotics. *Mach. Learn.* **2020**, *112*, 3713–3747. <https://doi.org/10.1007/s10994-021-06019-1>.
19. Cole, E.R.; Connolly, M.J.; Ghetiya, M.; Sendi, M.E.S.; Kashlan, A.; Eggers, T.E.; Gross, R.E. SAFE-OPT: A Bayesian optimization algorithm for learning optimal deep brain stimulation parameters with safety constraints. *J. Neural Eng.* **2024**, *21*, 046054. <https://doi.org/10.1088/1741-2552/ad6cf3>.
20. Gardner, J.; Kusner, M.; Xu, E.; Weinberger, K.; Cunningham, J. Bayesian Optimization with Inequality Constraints. In Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 21–26 June 2014; Volume 32.
21. Rasmussen, C.E.; Williams, C.K.I. *Gaussian Processes for Machine Learning*; Adaptive Computation and Machine Learning; MIT Press: Cambridge, MA, USA, 2006; pp. I–XVIII, 1–248.
22. Jones, D.; Schonlau, M.; Welch, W. Efficient Global Optimization of Expensive Black-Box Functions. *J. Glob. Optim.* **1998**, *13*, 455–492. <https://doi.org/10.1023/A:1008306431147>.
23. Jones, D.R. A taxonomy of global optimization methods based on response surfaces. *J. Glob. Optim.* **2001**, *21*, 345–383.
24. Schölkopf, B.; Smola, A.J. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*; The MIT Press: Cambridge, MA, USA, 2001. <https://doi.org/10.7551/mitpress/4175.001.0001>.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.