VABO: Violation-Aware Bayesian Optimization for Closed-Loop Control Performance Optimization with Unmodeled Constraints

Wenjie Xu[1,2], Colin N Jones[1], Bratislav Svetozarevic[2], Christopher R. Laughman[3], Ankush Chakrabarty[3,‡]

***Abstract*—We study the problem of performance optimization of closed-loop control systems with unmodeled dynamics. Bayesian optimization (BO) has been demonstrated effective for improving closed-loop performance by automatically tuning controller gains or reference setpoints in a model-free manner. However, BO methods have rarely been tested on dynamical systems with unmodeled constraints. In this paper, we propose a violation-aware BO algorithm (VABO) that optimizes closed-loop performance while simultaneously learning constraint-feasible solutions. Unlike classical constrained BO methods which allow an unlimited constraint violations, or 'safe' BO algorithms that are conservative and try to operate with near-zero violations, we allow budgeted constraint violations to improve constraint learning and accelerate optimization. We demonstrate the effectiveness of our proposed VABO method for energy minimization of industrial vapor compression systems.**

I. INTRODUCTION

Closed-loop systems can often be optimized after deployment by altering controller gains or reference inputs guided by the performance observed through operational data. Manually tuning these control parameters often requires care and effort along with considerable task-specific expertise. Algorithms that can automatically adjust these control parameters to achieve optimal performance are therefore invaluable for saving manual effort, time, and expenditure.

Optimal performance of a control system is generally defined via domain-specific performance functions whose arguments are outputs that can be measured from the closed-loop system. While the map from measurements to performance may be clearly defined, the map from control parameters (that can actually be tuned) to performance is often unmodeled or unknown, since closed-form system dynamics may not be available during tuning [1]. It is thus common to treat the control parameters-to-performance map as a black-box, and design the tuning algorithm to perform in a data-driven manner, where data is collected by performing experiments or simulating a model. However, since both experimentation and high-fidelity software simulations are expensive, tuning algorithms must be designed to assign a near-optimal set of control parameters with as few experiments/simulations (equivalently, performance function evaluations) as possible.

It is precisely for this reason that Bayesian optimization (BO) has received widespread attention in the context of closed-loop performance optimization. BO is a sample-efficient derivative-free global optimization method [2], [3] that utilizes probabilistic machine learning to intelligently search through high-dimensional parameter spaces. In recent work, BO has demonstrated potential in controller gain tuning [4]–[7], MPC tuning [8]–[10] and in various real-world control applications, such as wind energy systems [11], [12], engines [13] and space cooling [1].

A challenge that has garnered recent interest is that of *safe* Bayesian optimization; that is, BO in the presence of safety-critical system constraints. These constraints may also be unmodeled ('black-box'), as a mathematical representation of the constraint with respect to the control parameters is not always known or straightforward to represent. To handle these constraints, a variety of so-called 'safe BO' techniques have been recently proposed; these methods either operate on the principle of minimizing constraint violations during optimization [14]–[16], or leverage partial model knowledge to ensure safety via Lyapunov arguments [17]. In either case, safe BO achieves the goal of learning feasible optima without violating unmodeled constraints, or risking their violation with predefined small probability. Often, this conservativeness results in obtaining local minima, slow convergence speeds, and reduced data-efficiency. Conversely, generic constrained optimization with BO learns constraints without paying heed to the amount of constraint violation during the exploration phase. These methods [18], [19], are mostly agnostic to the deleterious consequences of constraint violation, such as long-term damage to expensive hardware caused by large violation, rendering them impractical for many industrial applications. A more violation-aware direction of BO research proposes the use of budgets on the cost of constraint violation [20]–[23]. However, the budget considered (e.g., neural network training time [20]) is usually related to the effort or failure risk for performance function evaluation, and does not provide a way to manage the magnitude of constraint violations.

For many industrial systems, small constraint violations over a short period are acceptable if that exploration results in speedup of the optimization procedure, but large violations are heavily discouraged. Concretely, we study vapor compression systems (VCSs) where it is imperative that constraint violation on system variables such as compressor discharge

[1] Laboratoire d'Automatique, École polytechnique fédérale de Lausanne, Lausanne, Switzerland. ✉ {wenjie.xu, colin.jones}@epfl.ch.

[2] Swiss Federal Laboratories for Materials Science and Technology (EMPA), Switzerland. ✉ bratislav.svetozarevic@empa.ch

[3] Mitsubishi Electric Research Laboratories, Cambridge, MA, USA. ✉ laughman@merl.com

[‡] Corresponding author. ✉ achakrabarty@ieee.org. ☎ +1 (617) 758-6175. 📍 201 Broadway, 8th Floor, Cambridge, MA 02139, USA.

This work has received support from the Swiss National Science Foundation under the NCCR Automation project, grant agreement 51NF40_180545.

temperature are limited to short time periods, and therefore, the interplay of informativeness and safety can be systematically adjusted through budgeting. In other words, for VCS, the benefits of accelerated global convergence significantly outweigh the cost of short-term constraint violations.

In this paper, we propose a novel violation-aware Bayesian optimization (VABO) that exhibits accelerated convergence compared to safe BO algorithms, while ensuring the violation cost is within a prescribed budget. We demonstrate that our proposed VABO algorithm is less conservative than 'safe BO' algorithms that tend to have sample-inefficient exploration phases and can get stuck in a local minimum because they cannot allow any constraint violation. The VABO algorithm is also more cautious than constrained Bayesian optimization, which is agnostic to constraint violations and thus, is likely to incur significant violation costs. Our VABO algorithm is based on the principle of encouraging performance function evaluation at combinations of control parameters that greatly assist the optimization process, as long as it does not incur high constraint violations likely to result in system failure or irreversible damage.

Our **contributions** include:

1) we propose a variant of constrained BO methods for control parameter tuning that improves global convergence rates within a prescribed amount of constraint violation with guaranteed high probability;
2) we propose a simple and tractable constrained auxiliary acquisition function optimization problem for trading off performance improvement and constraint violation; and,
3) we validate our algorithms on a set-point optimization problem using a high-fidelity VCS that has been calibrated on an industrial HVAC system.

II. Preliminaries

A. Problem Statement

We consider closed-loop systems of the form

$$\xi_+ = F(\xi, \theta), \tag{1}$$

where $\xi, \xi_+ \in \mathbb{R}^{n_\xi}$ denote the system state and update (respectively), $\theta \in \Theta \subset \mathbb{R}^{n_\theta}$ the control parameters (e.g., set-points) to be tuned, and $F(\cdot,\cdot)$ the closed-loop dynamics with initial condition ξ_0. We assume that the closed-loop system (1) is designed such that it is exponentially stable to a control parameter dependent equilibrium state $\xi^\infty(\theta)$ for every $\theta \in \Theta$. We further assume that $\xi^\infty(\cdot)$ is a continuous map on Θ. For simplicity, we also assume Θ is a box (product of intervals) in $\mathbb{R}^{n_\theta}$.

To determine the system performance, we define a continuous cost function $\ell(\theta) : \mathbb{R}^{n_\theta} \mapsto \mathbb{R}$ to be minimized, which is an unknown/unmodeled function of the parameters θ. This is not unusual: while ℓ may be well-defined in terms of system outputs, it is often the case that the map from control parameters to cost remains unmodeled; in fact, ℓ may not even admit a closed-form representation on Θ, c.f. [1], [24].

We also define N unmodeled constraints on the system outputs that require caution during tuning. The i-th such constraint is given by $g_i(\theta) : \mathbb{R}^{n_\theta} \mapsto \mathbb{R}, i \in [N]$, where the notation $[N] \overset{\text{def}}{=} \{i \in \mathbb{N}, 1 \le i \le N\}$; we assume each $g_i(\cdot)$ is continuous on Θ. We assume that the cost function $l(\theta)$ and every constraint $g_i(\theta)$, $i \in [N]$ can be ascertained, either by measurement or estimation, during the hardware/simulation experiment. We introduce the brief notation $g(\cdot) \le 0 \triangleq g_i(\cdot) \le 0$, $i \in [N]$ and assume that an initial feasible set of solutions is available at design time.

Assumption 1. *The designer has access to a non-empty safe set Θ_0 such that for any $\theta \in \Theta_0$, no constraint is violated; that is, $g(\theta) \le 0$ for every $\theta \in \Theta_0$.*

While such an initial set Θ_0 can be derived using domain expertise, it is likely that Θ_0 contains only a few feasible θ, and at worst could even be a singleton set.

We cast the control parameter tuning problem as a black-box constrained optimization problem, formally described by

$$\min_{\theta \in \Theta} \quad \ell(\theta), \tag{2a}$$

$$\text{subject to:} \quad g_i(\theta) \le 0, \quad \forall i \in [N]. \tag{2b}$$

Our **objective** is to solve the constrained optimization problem (2) with limited constraint violations during the optimization process. Since the constraints are assumed to be unmodeled and a limited set of feasible solutions is known at design time, we do not expect a guarantee of zero constraint violations. The tolerable amount and duration of constraint violations are problem-dependent. In some applications, such as vapor compression systems, small constraint violations over a short-term are acceptable, while large constraint violations are strongly discouraged. In such cases, instead of being overly cautious and ending up with suboptimal solutions, we allow small constraint violations *as long as the resulting knowledge gathered by evaluating an infeasible (in terms of constraint violation) θ accelerates the optimization process or helps avoid local minima.*

Remark 1. *Our formulation* (2) *can also optimize batch processes over finite time-horizons, say T_h. This would involve defining the objective and constraints over a batch trajectory with stage loss $\ell(\theta) := \frac{1}{T_h}\int_0^{T_h} l(\tau,\theta)\,\mathrm{d}\tau$.*

B. Proposed Solution

We propose a modified Bayesian optimization framework to solve the problem (2) that is violation-aware: the algorithm automatically updates the degree of risk-taking in the current iteration based on the severity of constraint violations in prior iterations. Concretely, for an infeasible θ, the constraint violation cost is given by

$$\bar{c}_i(\theta) \triangleq c_i\left([g_i(\theta)]^+\right), \quad i \in [N] \tag{3}$$

where $[g_i(\cdot)]^+ := \max\{g_i(\cdot), 0\}$ and $c_i : \mathbb{R}_{\ge 0} \mapsto \mathbb{R}_{\ge 0}$. Note that g_i corresponds to physically meaningful system outputs that we can measure, e.g., temperature. This violation cost function c_i is user-defined as a means to explicitly weight the severity of 'small' versus 'large' constraint violations. While the function c_i is at the discretion of the designer, it needs to satisfy the following mild assumptions in order to achieve desirable theoretical properties; see §3.

Assumption 2. *The violation function c_i satisfies:*
(A1) $c_i(0) = 0$,
(A2) $c_i(s_1) \geq c_i(s_2)$*, if* $s_1 > s_2 \geq 0$,
(A3) c_i *is left continuous on* $\mathbb{R}_{\geq 0}$.

Assumption 2 captures some intuitive properties required of the violation function. According to (A1), there is no cost associated with no violation. From (A2), we ensure that the violation cost is monotonically non-decreasing with increased violations. Finally (A3) ensures that this monotonic increase is smooth and does not exhibit discontinuous jumps.

To adapt the degree of risk-taking based on prior data obtained, we define a violation budget over a horizon of $T \in \mathbb{N}$ optimization iterations. Our proposed VABO algorithm is designed to sequentially search over T iterations $\{\theta_t\}_{t=1}^{T}$ while using a prescribed budget of constraint violations in order to obtain a constraint-optimal set of parameters

$$\theta_T^\star := \operatorname*{arg\,min}_{\substack{t\in[T];\\ g(\theta_t)\leq 0}} \ell(\theta_t) \text{ subject to: } \textstyle\sum_{t=1}^{T} \bar{c}_i(\theta_t) \leq B_i,$$

where B_i denotes a budget allowed for the i-th violation cost.

Note that this formulation is a generalization of well-known constrained Bayesian optimization formulations proposed in the literature. For instance, if we set $B_i = 0, \forall i \in [N]$, then our formulation is reminiscent of safe Bayesian optimization [14], [15]. Alternatively, setting $B_i \equiv \infty$ reduces our problem to constrained Bayesian optimization problem without violation cost consideration [18], [19].

III. Violation-Aware Bayesian Optimization (VABO)

A. Bayesian Optimization Preliminary

For Bayesian optimization, one models $\ell(x)$ and $g(x)$ as functions sampled from independent Gaussian processes. At iteration t, conditioned on previous function evaluation data $\mathscr{D} := \{\theta_{1:t}, \ell(\theta_{1:t})\}$, the posterior mean and standard deviation of ℓ is given by $\mu_\ell(\theta|\mathscr{D}) = k_\ell^\top(\theta, \theta_{\mathscr{D}})K_\ell^{-1}\Delta y_\ell + \mu_{\ell,0}(\theta)$ and $\sigma_\ell^2(\theta|\mathscr{D}) = k_\ell(\theta,\theta) - k_\ell^\top(x,\theta_{\mathscr{D}})K_\ell^{-1}k_\ell(\theta_{\mathscr{D}},\theta)$, where $\theta_{\mathscr{D}} = \theta_{1:t}$ is the set of control parameters with which previous experiments/simulations have been performed. Here,

$$k_\ell(\theta,\theta_{\mathscr{D}}) \triangleq [k_\ell(\theta,\theta_i)]_{\theta_i\in\theta_{\mathscr{D}}},\; k_\ell(\theta_{\mathscr{D}},\theta) \triangleq [k_\ell(\theta_i,\theta)]_{\theta_i\in\theta_{\mathscr{D}}},$$
$$K_\ell \triangleq \left(k_\ell(\theta_i,\theta_j)\right)_{\theta_i,\theta_j\in\theta_{\mathscr{D}}},\; \Delta y_\ell \triangleq [\ell(\theta_i) - \mu_{\ell,0}(\theta_i)]_{\theta_i\in\theta_{\mathscr{D}}},$$

and $k_\ell(\cdot,\cdot)$ is a user-defined kernel function and $\mu_{\ell,0}$ is the prior mean function, both associated with ℓ; see [3] for more details on kernel and prior mean selection. The above quantities are all assumed to be column vectors, except K_ℓ, which is a positive-definite matrix. For the constraint functions g, similar expressions for the posterior mean $\mu_{g_i}(\theta|\mathscr{D})$ and standard deviation $\sigma_{g_i}(\theta|\mathscr{D})$ can be obtained.

The kernelized functions above provide tractable approximations of the performance cost of the closed-loop system, along with the constraint functions, both of which were hitherto unmodeled/unknown. Classical BO methods operate by using the statistical information embedded within these approximations to intelligently explore the search space Θ via acquisition functions. A specific instance of an acquisition function commonly used in constrained BO is the constrained expected improved (CEI) function [18], given by

$$\mathsf{CEI}(\theta|\mathscr{D}) = \mathbb{E}\left(\prod_{i\in[N]} \mathbf{1}_{g_i(\theta)\leq 0}\, I(\theta)|\mathscr{D}\right). \tag{4}$$

where $\mathbf{1}$ denotes the indicator function, $\mathbb{E}$ denotes the expectation operator, and $I(\theta) = \max\{0, f(\theta_t^{\min}) - f(\theta)\}$ is the improvement of θ over the incumbent solution over t iterations. Here, the incumbent best solution is given by

$$\theta_t^{\min} = \operatorname*{arg\,min}_{\{\theta_\tau|\tau\in[t-1],g_i(\theta_\tau)\leq 0,\forall i\in[N]\}\cup\Theta_0} \ell(\theta), \tag{5}$$

with Θ_0 being the initial safe set of feasible points.

As $g_i(\theta), \forall i \in [N]$ and $\ell(\theta)$ are independent, we deduce

$$\mathsf{CEI}(\theta|\mathscr{D}) = \prod_{i\in[N]} \mathbb{P}(g_i(\theta)\leq 0|\mathscr{D})\,\mathbb{E}\left(I(\theta)|\mathscr{D}\right).$$

This acquisition function is therefore modulated by the feasibility probability

$$\mathbb{P}(g_i(\theta)\leq 0|\mathscr{D}) = \Phi\left(\frac{-\mu_{g_i}(\theta|\mathscr{D})}{\sigma_{g_i}(\theta|\mathscr{D})}\right), \tag{6a}$$

where $\Phi(\cdot)$ is the cumulative distribution function of standard normal distribution. The closed-form expression of unconstrained expected improvement is derived in [2],

$$\mathbb{E}\left(I(\theta)|\mathscr{D}\right) = \Big(\ell(\theta_t^{\min}) - \mu_\ell(\theta|\mathscr{D})\Big)\Phi(z) + \sigma_\ell(\theta|\mathscr{D})\phi(z), \tag{6b}$$

where $\phi(\cdot)$ is the standard normal probability density function and $z = \frac{\ell(\theta_t^{\min}) - \mu_\ell(\theta|\mathscr{D})}{\sigma_\ell(\theta|\mathscr{D})}$. Combining (6a) and (6b) yields a tractable expression for computing CEI in (7).

B. The VABO Algorithm

Our VABO algorithm proposes an auxiliary optimization problem that leverages the constrained expected improvement acquisition function to select feasible good candidates while ensuring (with high probability) that the violation cost of evaluating this candidate will remain within a prescribed budget.

Let the remaining budget at the t-th VABO iteration be $B_{i,t} \triangleq B_i - \sum_{\tau=1}^{t-1} \bar{c}_i(\theta_\tau)$. At this iteration, we solve the following *auxiliary* problem

$$\theta_t^\star := \operatorname*{arg\,max}_{\theta\in\Theta} \mathsf{CEI}(\theta|\mathscr{D}), \tag{7a}$$
$$\text{subject to:} \quad \prod_{i\in[N]} \mathbb{P}(\bar{c}_i(\theta) \leq \beta_{i,t}B_{i,t}) \geq 1 - \varepsilon_t, \tag{7b}$$

to compute the next control parameter candidate $\theta_t^\star$, where $\beta_{i,t} \in [0,1]$ is a user-defined weighting scalar that determines how much of the remaining budget can be used up, and $0 < \varepsilon_t \ll 1$ determines the probability of large constraint violation. Note that (7a) involves maximizing the constrained expected improvement objective, which is common to cBO algorithms; c.f. [18]. Our modification using the budget, as written in (7b), enforces that the next sampled point will not use up more than a $\beta_{i,t}$ fraction of the remaining violation cost budget $B_{i,t}$ for all constraints with a probability of at least $1-\varepsilon_t$, conditioned on the data seen so far. This modification allows us to trade a prescribed level of violation risk for more aggresive exploration, leading to faster convergence. Although we are using the commonly used constrained expected improvement acquisition function here, we can easily generalize to other acquisition functions by simply replacing CEI in the objective (7a).

We now discuss how to efficiently solve the auxiliary problem (7). Recall from Assumption 2 that c_i is non-decreasing on $\mathbb{R}_{\geq 0}$ for every $i \in [N]$. Therefore, we can define an inverse violation function $c_i^{-1}(s) = \sup\{r \in \mathbb{R}_{\geq 0} \mid c_i(r) \leq s\}$ for any $s \in \mathbb{R}_{\geq 0}$. Therefore, we can write

$$\begin{aligned}\mathbb{P}(\bar{c}_i(\theta) \leq \beta_{i,t}B_{i,t}|\mathscr{D}) &= \mathbb{P}([g_i(\theta)]^+ \leq c_i^{-1}(\beta_{i,t}B_{i,t})|\mathscr{D}) \\ &= \mathbb{P}(g_i(\theta) \leq c_i^{-1}(\beta_{i,t}B_{i,t})|\mathscr{D}).\end{aligned}$$

Since $g_i(\theta)$ follows a Gaussian distribution with mean $\mu_{g_i}(\theta|\mathscr{D})$ and variance $\sigma^2_{g_i}(\theta|\mathscr{D})$, we get

$$\mathbb{P}(\bar{c}_i(\theta) \leq \beta_{i,t}B_{i,t}|\mathscr{D}) = \Phi\left(\frac{c_i^{-1}(\beta_{i,t}B_{i,t}) - \mu_{g_i}(\theta|\mathscr{D})}{\sigma_{g_i}(\theta|\mathscr{D})}\right).$$

When the number of control parameters n_θ is small (< 6), we can place a grid on Θ and evaluate the cost and constraints of (7) at all the grid nodes. The maximum feasible solution can then be used as the solution to the auxiliary problem. When $n_\theta > 6$, we can use gradient-based methods with multiple starting points to solve problem (7), since evaluating the learned GPs approximating ℓ and g require very little computational time or effort when T is not large (empirically, < 2000). The VABO algorithm is terminated either when T iterations of the algorithm have been reached, or the cumulative violation cost for constraint i exceeds B_i for any $i \in [N]$. We provide pseudocode for implementation in Algorithm 1.

The following proposition provides a probabilistic guarantee of using up the given number of samples while keeping the violation cost below the given budget. It highlights the "violation-awareness" property exhibited by VABO.

Proposition 1. *Fix $\delta \in (0,1)$ and $T \in \mathbb{N}$. For any $\beta_{i,t} \in (0,1]$, if ε_t are chosen for every $t \in [T]$ such that $\delta = 1 - \prod_{t=1}^{T}(1-\varepsilon_t)$, then the VABO algorithm described in Algorithm 1 satisfies the probability of*

$$\{T \text{ iters are used}\} \cap \left\{\sum_{t=1}^{T} \bar{c}_i(\theta_t) \leq B_i, \forall i \in [N]\right\}$$

is at least $1-\delta$.

Proof. Let

$$\mathscr{E}_t := \{t \text{ iters are used}\} \cap \left\{\sum_{\tau=1}^{t} \bar{c}_i(\theta_\tau) \leq B_i,\ \forall i \in [N]\right\},$$

where $t \in [T]$. We have

$$\begin{aligned}\mathbb{P}(\mathscr{E}_T) &= \mathbb{P}(\mathscr{E}_{T-1})\,\mathbb{P}(\mathscr{E}_T \mid \mathscr{E}_{T-1}) \\ &= \mathbb{P}(\mathscr{E}_{T-1})\,\mathbb{P}\left(\bar{c}_i(\theta_T)) \leq B_{i,T}, \forall i \in [N] \mid \mathscr{E}_{T-1}\right) \\ &\geq \mathbb{P}(\mathscr{E}_{T-1})\,\mathbb{P}\left(\bar{c}_i(\theta) \leq \beta_{i,T}B_{i,T}, \forall i \in [N] \mid \mathscr{E}_{T-1}\right) \\ &\geq \mathbb{P}(\mathscr{E}_{T-1})\,(1-\varepsilon_T).\end{aligned}$$

By recursion, we have $\mathbb{P}(\mathscr{E}_T) \geq \mathbb{P}(\mathscr{E}_1)\prod_{t=2}^{T}(1-\varepsilon_t) \geq \prod_{t=1}^{T}(1-\varepsilon_t) = 1-\delta$, which concludes the proof. □

Algorithm 1 Violation-Aware Bayesian Optimization

```
Require: Max VABO iterations $T$, violation budget $B_i, \forall i \in [N]$ and an initial safe set of points $\Theta_0$
Evaluate $\ell(\theta)$, $g_i(\theta), \forall i \in [N]$ for $\theta \in \Theta_0$ by performing experiments or simulation
Calculate incumbent solution using (5)
Initialize dataset $\mathscr{D} = \{\theta_t, \ell(\theta_t), g(\theta_t)\}\ \forall \theta_t \in \Theta_0$
for $t \in [T]$ do
    $\Theta_t = \left\{\prod_{i \in [N]} \mathbb{P}(\bar{c}_i(\theta) \leq \beta_{i,t}B_{i,t}|\mathscr{D}) \geq 1-\varepsilon_t | \theta \in \Theta\right\}$
    $\theta_t^\star = \arg\max_{\theta \in \Theta_t} \mathsf{CEI}(\theta|\mathscr{D})$          ▷ Solving (7)
    $\ell(\theta_t^\star), g(\theta_t^\star) \leftarrow$ perform experiment with $\theta_t^\star$
    $B_{i,t+1} = B_{i,t} - \bar{c}_i(\theta_t), \forall i \in [N]$
    if $\exists i \in [N]$, such that $B_{i,t+1} < 0$ then
        return $\theta_{t+1}^{\min}$
    Update dataset, $\mathscr{D} \leftarrow \mathscr{D} \cup \{\theta_t^\star, \ell(\theta_t^\star), g(\theta_t^\star)\}$
    Update Gaussian process posterior
return VABO solution: $\theta_{T+1}^{\min}$
```

IV. Case Study: VCS Optimization with Temperature Constraints

In this section, we consider the problem of safely tuning set-points of a vapor compression system (VCS). As shown in Fig. 1, a VCS typically consists of a compressor, a condenser, an expansion valve, and an evaporator. While physics-based models of these systems can be formulated as large sets of nonlinear differential algebraic equations to predict electrical power consumption, there are a variety of challenges in developing and calibrating these models. This motivates interest in directly using measurements of the VCS power under different operating conditions, and assigning optimal set-points to the VCS actuators using data-driven, black-box optimization methods such as BO, which minimize the power consumption.

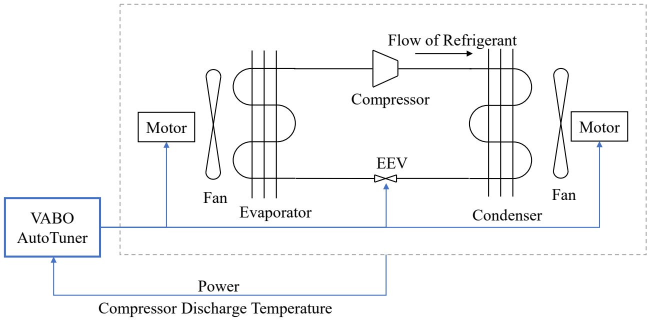

Fig. 1. Schematic diagram of vapor compression system with our proposed VABO controlling the EEV (electronic expansion valve) and the two fans' speed. For simplicity, we do not show other measurements and controls.

Reckless alterations of the set-points can drive the system into operating modes that reduce the reliability or lifespan of the system. To avoid these deleterious effects, one can add several constraints during the tuning process. One such constraint, considered here, is on the temperature of the refrigerant leaving the compressor, also referred to as the 'discharge temperature'. The discharge temperature must be managed because compressors are designed to operate within specific temperature ranges; excessively high temperatures can result in the breakdown of refrigerant oils and increase wear and tear. In addition, high temperatures are often correlated with high pressures, which can cause metal fatigue and compromise the integrity of the pressurized refrigerant pipes in extreme cases. While managing the constraints mentioned above in the long run is critical, we also observe that small violations over short periods of time have limited harmful effects. Indeed, it may be beneficial to take the risk of short-period limited violation to accelerate the tuning process.

We cast the VCS optimization problem in the same form as (2), with ℓ denoting the steady-state power of the VCS with set points θ. The constraint $g \leq 0$ is given by $T_d(\theta) - \hat{T}_d \leq 0$, where $T_d(\theta)$ is the steady-state discharge temperature with set points θ and $\hat{T}_d$ is a safe upper bound. We close a feedback loop from compressor frequency to room temperature, leaving the set of 3 tunable set points θ as the expansion valve position and the indoor/outdoor fan speeds. The effect of these set points on power and discharge temperature are not easy to model, and no simple closed-form representation exists. In practice, we assign a setpoint θ, wait for an adequate amount of time until the power signal resides within a 95% settling zone, and use that power value as $\ell(\theta)$ and the corresponding discharge temperature as $T_d(\theta)$.

Implementation Details: For this paper, we use a high-fidelity model of the dynamics of a prototype VCS[1] written in the `Modelica` language [25] to collect data and optimize the set-points on-the-fly. A complete description of the model is available in [1]. The model was first developed in the `Dymola` [26] environment, and then exported as a functional mockup unit (FMU) [27], and its current version comprises 12,114 differential equations. Bayesian optimization is implemented in `GPy` [28].

We define our set-point search space $\Theta := [200,300] \times [300,400] \times [500,800]$, in expansion valve counts, indoor fan rpm, and outdoor fan rpm, respectively. We aim to keep the discharge temperature below $\hat{T}_d = 331$ K; these constraints are set according to domain knowledge [24]. We initialize the simulator at an expansion valve position of 280 counts, an indoor fan speed of 380 rpm, and an outdoor fan speed of 700 rpm, which is known to be a feasible set-point based on experience. Constraint violations are penalized with the function $c_i(s) = s^2, s \in \mathbb{R}^+$. The quadratic nature of the violation cost implies that minor violations are not as heavily penalized as larger ones. The reason for this is that small violations over a small period of time are unlikely to prove deleterious to the long-term health of the VCS, whereas large violations could have more significant effects, even over short periods of time; for instance, damage to motor winding insulation or exceeding mechanical limits on the pressure vessel of the compressor. These constraints have been incorporated into the selection of the thresholds $\hat{T}_d$. Of course, the threshold values and a parameterized violation cost could be considered to be hyperparameters, and hence could be optimized via further experimentation. We choose the RBF kernel for our problem, which is commonly used in Bayesian optimization [3], and compare our method to safe BO [29] and generic cBO [19].

We showcase the effect of varying B by selecting $B = 0$, 10, and 20, and set $\beta_t = \max\{\beta_0, \frac{1}{T-t+1}\}$. This choice allows the algorithm to use an increasing fraction of the budget when approaching the sampling limit T. A violation of the cost budget of 10 works well in this case study, and additional violations of the cost budget brings little convergence benefit. We also observe that the impact of β_0 is insignificant for this example, and therefore we set $\beta_0 = 1.0$. We use "VABO B" to indicate violation aware BO with budget B.

Results and Discussion: Figure 2 illustrates the experimental results. We observe that the best feasible solution's power of our VABO with $B = 10$ decreases slightly faster than generic constrained BO and significantly faster than safe BO. At the same time, our method manages the violation cost well under the violation cost budget. In comparison, generic constrained BO incurs very large costs in the last few steps, far exceeding the violation cost budget 10 by more than an order of magnitude. The improvements from VABO are due to its recognition that the discharge temperature is sensitive to the expansion valve position, but that this position needs to be adjusted in coordination with the fan speeds. cBO causes large discharge temperatures after step 12 because it makes large adjustments to the expansion valve position while maintaining the indoor fan speed at a low value, a combination that is not penalized during exploration. VABO with $B = 10$ reduces the power by about 9% compared to the most power-efficient initial safe set-points after 20 steps. We also observe that the T_d constraint is often violated during exploration, and that large violations are entirely possible without violation awareness, as demonstrated by generic cBO. Even without any violation cost budget ($B_i = 0$), our violation-aware BO finds better solutions faster than safe BO. One explanation for this is that safe BO tends to waste a lot of evaluations to enlarge the safe set, which may lead to slow convergence. Conversely, VABO implicitly encodes the safe set exploration into the acquisition optimization problem (7) and only enlarges the safe set when necessary for optimization, while keeping the violation risk under control.

V. Conclusions

In this paper, we design a sample-efficient and violation-aware Bayesian optimization (VABO) algorithm to solve the closed-loop control performance optimization problem with unmodeled constraints, by leveraging the fact that small violations over a short period only incur limited costs during the optimization process in many applications such as for vapor-compression cycles. We strategically trade the bud-

[1] Note that while the behavior of this model have been validated against a real VCS, the numerical values and/or performance presented in this work is not representative of any product.

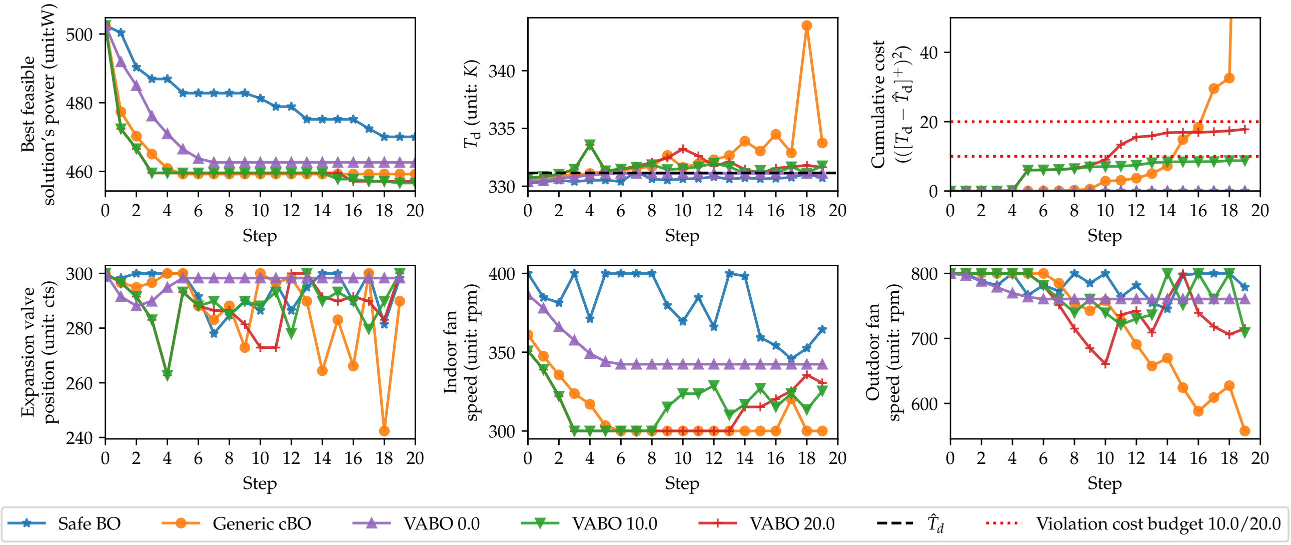

Fig. 2. Best feasible solution's power, discharge temperature, cumulative violation cost and the three set-points' evolution.

geted violations for faster convergence by solving a tractable auxiliary problem with probabilistic budget constraints at each step. Our experiments on a VCS show that, as compared to existing safe BO and generic constrained BO, our method simultaneously exhibits accelerated global convergence and manages the violation cost.

References

[1] A. Chakrabarty, C. Danielson, S. A. Bortoff, and C. R. Laughman, "Accelerating self-optimization control of refrigerant cycles with Bayesian optimization and adaptive moment estimation," *Appl. Therm. Eng.*, vol. 197, p. 117335, 2021.

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

[3] P. I. Frazier, "A tutorial on Bayesian optimization," *arXiv preprint arXiv:1807.02811*, 2018.

[4] A. Lederer, A. Capone, and S. Hirche, "Parameter optimization for learning-based control of control-affine systems," in *Learning for Dynamics and Control*. PMLR, 2020, pp. 465–475.

[5] R. R. Duivenvoorden, F. Berkenkamp, N. Carion, A. Krause, and A. P. Schoellig, "Constrained Bayesian optimization with particle swarms for safe adaptive controller tuning," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 11 800–11 807, 2017.

[6] 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 Eur. Control Conf. (ECC)*. IEEE, 2019, pp. 1467–1472.

[7] C. König, M. Khosravi, M. Maier, R. S. Smith, A. Rupenyan, and J. Lygeros, "Safety-aware cascade controller tuning using constrained Bayesian optimization," *arXiv preprint arXiv:2010.15211*, 2020.

[8] S. Bansal, R. Calandra, T. Xiao, S. Levine, and C. J. Tomlin, "Goal-driven dynamics learning via bayesian optimization," in *2017 IEEE 56th Annu. Conf. on Decis. and Control (CDC)*. IEEE, 2017, pp. 5168–5173.

[9] D. Piga, M. Forgione, S. Formentin, and A. Bemporad, "Performance-oriented model learning for data-driven mpc design," *IEEE Contr. Syst. Lett.*, vol. 3, no. 3, pp. 577–582, 2019.

[10] J. A. Paulson and A. Mesbah, "Data-driven scenario optimization for automated controller tuning with probabilistic performance guarantees," *IEEE Contr. Syst. Lett.*, vol. 5, no. 4, pp. 1477–1482, 2020.

[11] A. Baheri and C. Vermillion, "Altitude optimization of airborne wind energy systems: A bayesian optimization approach," in *2017 Amer. Control Conf. (ACC)*. IEEE, 2017, pp. 1365–1370.

[12] ——, "Waypoint optimization using Bayesian optimization: A case study in airborne wind energy systems," in *2020 Amer. Control Conf. (ACC)*. IEEE, 2020, pp. 5102–5017.

[13] A. Pal, L. Zhu, Y. Wang, and G. G. Zhu, "Multi-objective stochastic Bayesian optimization for iterative engine calibration," in *Proc. of the Amer. Control Conf.* IEEE, 2020, pp. 4893–4898.

[14] Y. Sui, A. Gotovos, J. Burdick, and A. Krause, "Safe exploration for optimization with Gaussian processes," in *Proc. of the Int. Conf. on Mach. Learn.*, 2015, pp. 997–1005.

[15] Y. Sui, J. Burdick, Y. Yue, *et al.*, "Stage-wise safe Bayesian optimization with Gaussian processes," in *Proc. of the Int. Conf. on Mach. Learn.*, 2018, pp. 4781–4789.

[16] M. Turchetta, F. Berkenkamp, and A. Krause, "Safe exploration for interactive machine learning," *Advances in Neural Inf. Process. Syst. 32*, vol. 4, pp. 2868–2878, 2020.

[17] A. Chakrabarty and M. Benosman, "Safe learning-based observers for unknown nonlinear systems using Bayesian optimization," *Automatica*, vol. 133, p. 109860, 2021.

[18] J. R. Gardner, M. J. Kusner, Z. E. Xu, K. Q. Weinberger, and J. P. Cunningham, "Bayesian optimization with inequality constraints." in *ICML*, vol. 2014, 2014, pp. 937–945.

[19] M. A. Gelbart, J. Snoek, and R. P. Adams, "Bayesian optimization with unknown constraints," in *Proc. of the 30th Conf. on Uncertainty in Artif. Intell.*, ser. UAI'14. Arlington, Virginia, USA: AUAI Press, 2014, p. 250–259.

[20] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," *Advances in Neural Inf. Process. Syst.*, vol. 25, 2012.

[21] E. H. Lee, V. Perrone, C. Archambeau, and M. Seeger, "Cost-aware bayesian optimization," *arXiv preprint arXiv:2003.10870*, 2020.

[22] R. Lam, K. Willcox, and D. H. Wolpert, "Bayesian optimization with a finite budget: An approximate dynamic programming approach," *Advances in Neural Inf. Process. Syst.*, vol. 29, pp. 883–891, 2016.

[23] A. Marco, A. von Rohr, D. Baumann, J. M. Hernández-Lobato, and S. Trimpe, "Excursion search for constrained bayesian optimization under a limited budget of failures," *arXiv preprint arXiv:2005.07443*, 2020.

[24] D. J. Burns, C. R. Laughman, and M. Guay, "Proportional-integral extremum seeking for vapor compression systems," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 2, pp. 403–412, 2018.

[25] Modelica Association, "Modelica specification, Version 3.4," 2017. [Online]. Available: www.modelica.org

[26] Dassault Systemes, "Dymola 2020," 2019.

[27] Modelica Association, "Functional Mockup Interface for Model Exchange and Co-Simulation, Version 2.0.1," 2019. [Online]. Available: www.fmi-standard.org

[28] GPy, "GPy: A gaussian process framework in python," http://github.com/SheffieldML/GPy, since 2012.

[29] F. Berkenkamp, A. P. Schoellig, and A. Krause, "Safe controller optimization for quadrotors with gaussian processes," in *2016 IEEE Int. Conf. on Robot. and Automat. (ICRA)*. IEEE, 2016, pp. 491–496.