A Data-Driven Automatic Tuning Method for MPC under Uncertainty using Constrained Bayesian Optimization ⋆

Farshud Sorourifar [*,**] **Georgios Makrygirgos** [**] **Ali Mesbah** [**]
Joel A. Paulson [*]

[*] *Department of Chemical and Biomolecular Engineering, The Ohio State University, Columbus, OH 43210, USA*
[**] *Department of Chemical and Biomolecular Engineering, University of California, Berkeley, CA 94720, USA*

Abstract: The closed-loop performance of model predictive controllers (MPCs) is highly dependent on the choice of prediction models, controller formulation, and tuning parameters. However, prediction models are typically optimized for prediction accuracy, instead of performance, and MPC tuning is typically done manually to satisfy (probabilistic) constraints. In this work, we demonstrate a general approach for automating the tuning of MPC under uncertainty. In particular, we formulate the automated tuning problem as a constrained black-box optimization problem that can be tackled with derivative-free optimization. We rely on a constrained variant of Bayesian optimization to solve the MPC tuning problem that can directly handle noisy and expensive-to-evaluate functions. The benefits of the proposed automated tuning approach are demonstrated on a benchmark continuously stirred tank reactor case study.

Keywords: Model predictive control; Constrained Bayesian optimization; Model learning

1. INTRODUCTION

Model predictive control (MPC) is one of the most widely used methods for the control of constrained multivariable systems (Rawlings and Mayne, 2009). The closed-loop performance of MPC strongly depends on (i) the quality of its underlying process model; (ii) the formulation of the objective and constraints; and (iii) the choice of several tuning parameters (e.g., prediction horizon, weights in cost function, and constraint backoff terms) (Garriga and Soroush, 2010; Paulson and Mesbah, 2018). A major challenge in MPC tuning arises from the non-trivial relationships between the tuning parameters and the closed-loop control performance and constraint satisfaction (Lu et al., 2020). As such, MPC tuning via trial-and-error or other heuristic strategies may require a significant number of closed-loop simulations, which can quickly become prohibitive especially when system uncertainties are considered.

Recently, there has been a growing interest in automated strategies for controller tuning using Bayesian optimization (BO); e.g., see Berkenkamp et al. (2016); Bansal et al. (2017); Neumann-Brosig et al. (2019); Forgione et al. (2019); Khosravi et al. (2020); Lu et al. (2020); Paulson and Mesbah (2020). BO has emerged as a powerful derivative-free method for optimizing black-box functions in various applications (Shahriari et al., 2015); most notably for hyperparameter selection of machine learning algorithms (Snoek et al., 2012), including in the context of approximate MPC design (Bonzanini et al., 2021). BO is deemed particularly useful for solving MPC tuning problems since it can accommodate a mixture of continuous and discrete decision variables and also overcome the limitations of alternative derivative-free optimization methods such as genetic algorithms and particle swarm optimization (Garriga and Soroush, 2010).

In any model-based control approach, the chosen model plays a pivotal role in the design of the controller; however, identification of the model has been traditionally separated from controller design. An alternative idea that has been gaining popularity in recent years is to treat the identification process as a "tuning parameter" that should take into account the intended control application. The notion of identification for control (I4C) has been heavily studied in the context of fixed-order controllers for linear time-invariant systems (Gevers, 2005). Recently, the I4C was extended to MPC in (Piga et al., 2019), wherein BO is used to search for the best (parametrized) prediction model for MPC by directly optimizing closed-loop performance determined from plant (or high-fidelity simulation) data. We take a similar perspective in this work and show how the framework is applicable to more general MPC formulations.

The main focus of this paper is on integrated performance-driven model learning and MPC tuning under uncertainty. The explicit incorporation of uncertainty into the formulation of the automated tuning optimization problem (through both the objective and constraints) is one of the main contributions of this work. In particular, we

⋆ GM and AM acknowledge the support by the National Aeronautics and Space Administration (NASA) under grant number NNX17AJ31G . Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NASA .

show how independent uncertainty samples can be used to obtain effectively noisy measurements of objective and constraint, which can be accounted for in the BO through proper estimation of the noise variance. The second contribution is to leverage a *constrained* variant of BO (Gardner et al., 2014; Hernández-Lobato et al., 2016) to directly handle output constraints. Constrained BO has been shown to overcome challenges with barrier methods, which penalize constraint violation in the objective, since the constraints are separately modeled and thus provide an independent representation of the feasibility region.

The remainder of the paper is structured as follows. In Section 2, we define the problem of interest. In Section 3, a general overview of the parametrized MPC policy is provided. Section 4 presents our performance-driven MPC tuning algorithm including the details of the constrained BO method used. The effectiveness of the proposed method is demonstrated on a nonlinear continuously stirred tank reactor case study in Section 5 and conclusions are provided in Section 6.

2. PROBLEM FORMULATION

Consider an uncertain, time-invariant nonlinear system of the form

$$x_{k+1} = f_{\text{plant}}(x_k, u_k, w_k), \tag{1a}$$
$$y_k = h_{\text{plant}}(x_k, v_k), \tag{1b}$$

where $k \in \mathbb{N}_0$ denotes the discrete time index, $x_k \in \mathbb{R}^{n_x}$ denotes the system state, $u_k \in \mathbb{R}^{n_u}$ denotes the control input, $y_k \in \mathbb{R}^{n_y}$ denotes the measured output, $w_k \in \mathbb{R}^{n_w}$ denotes the process noise, and $v_k \in \mathbb{R}^{n_v}$ denotes the measurement noise. The initial state x_0, the process noise sequence $w_{0:T-1}$, and the measurement noise sequence $v_{0:T}$ are assumed to be random variables for some finite horizon $T \in \mathbb{N}$. Control performance $J(y_{1:T}, u_{1:T})$ is assumed to be some known function of the output and input sequences. Since the output depends on the realization of the uncertainty $\delta = (x_0, w_{0:T-1}, v_{0:T})$, our actual objective is to minimize the expected performance cost

$$\mathbb{E}_\delta\{J(y_{1:T}, u_{1:T})\} = \int_\Delta J(y_{1:T}, u_{1:T}) p(\delta) d\delta, \tag{2}$$

where $p(\delta)$ denotes the probability density function and Δ is the support of the random vector δ. The controlled system should also satisfy hard input constraints

$$u_{0:T} \in \mathbb{U}_{0:T} = \mathbb{U}_0 \times \cdots \times \mathbb{U}_T, \tag{3}$$

and joint chance constraints on the output

$$\mathbb{P}_\delta\{y_{1:T} \in \mathbb{Y}_{1:T} = \mathbb{Y}_1 \times \cdots \times \mathbb{Y}_T\} \geq 1 - \epsilon, \tag{4}$$

where $\epsilon \in [0, 1]$ denotes the allowed violation probability. The ideal controller design problem can be formulated in terms of the following stochastic optimization problem

$$\min_{\pi \in \Pi} \ \mathbb{E}_\delta\{J(y_{1:T}, u_{1:T})\}, \tag{5a}$$
$$\text{s.t.} \ \ x_{k+1} = f_{\text{plant}}(x_k, u_k, w_k), \quad \forall k \in \mathbb{N}_0^{T-1}, \tag{5b}$$
$$y_k = h_{\text{plant}}(x_k, v_k), \quad \forall k \in \mathbb{N}_0^{T}, \tag{5c}$$
$$u_k = \pi_k(y_{0:k}) \in \mathbb{U}_k, \quad \forall k \in \mathbb{N}_0^{T}, \tag{5d}$$
$$\mathbb{P}_\delta\{y_{1:T} \in \mathbb{Y}_{1:T}\} \geq 1 - \epsilon, \tag{5e}$$

where $\pi = \{\pi_0, \pi_1, \ldots, \pi_T\}$ denotes the control *policy* that is composed of a sequence of control laws $\pi_k : \mathbb{R}^{n_y(k+1)} \to \mathbb{R}^{n_u}$, which are arbitrary functions of the *history* of measured outputs and Π denotes the set of possible control policy structures. This problem is extremely challenging to solve for several reasons including the set of functions $\pi \in \Pi$ is infinite dimensional and the probabilistic operators $\mathbb{E}_\delta\{\cdot\}$ and $\mathbb{P}_\delta\{\cdot\}$ cannot be computed exactly for general nonlinear systems. Another key complication considered in this work is that functions $f_{\text{plant}}(\cdot)$ and $h_{\text{plant}}(\cdot)$ and uncertainty distribution $\delta \sim p(\delta)$ may not be explicitly known. Thus, (5) belongs to the challenging class of constrained black-box optimization problems under uncertainty.

Since the optimization problem (5) cannot be solved directly, one must resort to a heuristic controller design strategy such as model predictive control (MPC) (Rawlings and Mayne, 2009). The standard MPC design strategy involves: (i) identification of a control-relevant model of the system (1) using first-principles or system identification techniques; (ii) specification of the MPC controller using this control-relevant model to make internal predictions about the future system behavior to select the optimal control inputs; and (iii) trial-and-error experimentation to select the remaining MPC tuning parameters (e.g., prediction horizon, input and output weight matrices, and constraint backoffs) that can have a strong influence on closed-loop performance. Not only are the tuning parameters selected in an ad-hoc way in this approach, it is difficult to know how the quality of the predictions of the identified model will translate into control performance.

To address these challenges, inspired by the notion of I4C, we take a *performance-driven* perspective in this work in which the *model* of (1) is treated as a degree of freedom that can be used to minimize the closed-loop performance cost. The proposed MPC parametrization is summarized in the next section, which is followed by a description of an automated procedure for simultaneously selecting the internal model and other relevant MPC tuning parameters.

3. POLICY APPROXIMATION: PARAMETRIZED MODEL PREDICTIVE CONTROL

A critical component of MPC is the model used to predict the outputs given a sequence of inputs. Let the control-relevant model be defined as

$$s_{k+1} = f_{\text{mod}}(s_k, u_k; \theta_m), \tag{6a}$$
$$y_k = h_{\text{mod}}(s_k; \theta_m), \tag{6b}$$

where $s_k \in \mathbb{R}^{n_s}$ is the state of the control-relevant model at time step $k \in \mathbb{N}_0$ and $\theta_m \in \mathbb{R}^{n_{\theta,m}}$ is the set of model parameters. Although written in state-space form, (6) could be a realization of an input-output black-box model such as a nonlinear auto-regressive model with exogenous variables (NARX) in which case θ_m would represent the coefficients of the selected basis functions.

Given the approximate system model (6), the MPC controller solves the following finite-horizon optimal control problem at each sampling time $k \in \mathbb{N}_0^T$:

$$\min_{\mathbf{U}_k} \ \textstyle\sum_{i=1}^{N_p} \tilde{\ell}(y_{i|k}, u_{i|k}; \theta_o), \tag{7a}$$
$$\text{s.t.} \ \ s_{i+1|k} = f_{\text{mod}}(s_{i|k}, u_{i|k}; \theta_m), \quad \forall i \in \mathbb{N}_0^{N_p - 1}, \tag{7b}$$
$$y_{i|k} = h_{\text{mod}}(s_{i|k}; \theta_m), \quad \forall i \in \mathbb{N}_1^{N_p}, \tag{7c}$$
$$(y_{i|k} + \theta_b, u_{i|k}) \in \mathbb{Y}_{k+i} \times \mathbb{U}_{k+i}, \quad \forall i \in \mathbb{N}_1^{N_p}, \tag{7d}$$
$$s_{0|k} = \hat{s}_k, \tag{7e}$$

where N_p denotes the prediction horizon; $s_{i|k}$ and $u_{i|k}$ are, respectively, the predicted state and control input at time $k+i$ given information at time k; $\mathbf{U}_k = \{u_{0|k}, \dots, u_{N-1|k}\}$ is the control input sequence; $\hat{s}_k \in \mathbb{R}^{n_s}$ is the current state estimate; $\tilde{\ell} : \mathbb{R}^{n_y} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_{\theta,o}} \to \mathbb{R}$ is the stage cost function parametrized by $\theta_o \in \mathbb{R}^{n_{\theta,o}}$; and $\theta_b \in \mathbb{R}^{n_{\theta,b}}$ are the constraint backoff parameters that can be used to improve the robustness properties of the controller, as discussed in, e.g., (Paulson and Mesbah, 2018). The closed-loop performance index $J(y_{1:T}, u_{1:T})$ is not necessarily the same as the MPC cost function (7a). In most cases $J(y_{1:T}, u_{1:T})$ will be some function that reflects engineering and economic goals, while $\tilde{\ell}(\cdot)$ can be substantially simplified to facilitate online optimization.

Let $\{u^\star_{i|k}(\hat{s}_k; \theta_m, \theta_o, \theta_b, N_p)\}_{i=0}^{N_p-1}$ denote the optimal solution to (7) for a given state estimate and a set of tuning parameters. The receding-horizon control law, implicitly defined in terms of the solution to (7), is given by

$$\kappa_c(\hat{s}_k; \theta_m, \theta_o, \theta_b, N_p) = u^\star_{0|k}(\hat{s}_k; \theta_m, \theta_o, \theta_b, N_p). \quad (8)$$

Since the control-relevant state may not be directly measured, we assume a parametrized state estimator of the following form exists

$$\hat{s}_{k+1} = \kappa_e(\hat{s}_k, u_k, y_{k+1}; \theta_e), \quad (9)$$

where $\theta_e \in \mathbb{R}^{n_{\theta,e}}$ denotes any free parameters in the estimator. For example, θ_e could represent the process and measurement noise matrices in an extended Kalman filter (Hoshiya and Saito, 1984). By combining the specified controller (8) and estimator (9) with the system dynamics (1), we can represent the closed-loop system in terms of the augmented state $z_k = (x_k, \hat{s}_k)$

$$z_{k+1} = f_{\text{cl}}(z_k, w_k; \theta), \quad (10a)$$
$$u_k = \kappa_{\text{cl}}(z_k; \theta), \quad (10b)$$
$$y_k = h_{\text{cl}}(z_k, v_k), \quad (10c)$$

where $f_{\text{cl}} : \mathbb{R}^{n_z} \times \mathbb{R}^{n_w} \times \mathbb{R}^{n_\theta} \to \mathbb{R}^{n_z}$ defines the autonomous closed-loop dynamics, $\kappa_{\text{cl}} : \mathbb{R}^{n_z} \times \mathbb{R}^{n_\theta} \to \mathbb{R}^{n_u}$ is the control function in the augmented space, $h_{\text{cl}} : \mathbb{R}^{n_z} \times \mathbb{R}^{n_v} \to \mathbb{R}^{n_y}$ is the measurement function in the augmented space, $\theta = (\theta_m, \theta_o, \theta_b, N_p, \theta_e)$ denotes the complete set of tuning parameters appearing in the control structure, and $n_z = n_x + n_s$ and $n_\theta = n_{\theta,m} + n_{\theta,o} + n_{\theta,b} + n_{\theta,e} + 1$ are the augmented state and tuning parameter dimensions, respectively. Under this restricted class of control policies, we can construct an approximation to (5) as follows

$$\min_{\theta \in \Theta} \ \mathbb{E}_\delta\{J(y_{1:T}, u_{1:T})\}, \quad (11a)$$
$$\text{s.t.} \ \ (10), \ \mathbb{P}_\delta\{y_{1:T} \in \mathbb{Y}_{1:T}\} \geq 1 - \epsilon. \quad (11b)$$

Note that the control input constraints (3) are not included in this formulation as they are directly enforced by the MPC law (7). The key difference between (11) and (5) is that the proposed approximation (11) optimizes over a finite dimensional space $\Theta \subset \mathbb{R}^{n_\theta}$. The problem is also a single stage one in which a full simulation can be carried out once the tuning parameters θ are fixed. As such, we are not concerned about adapting the control policy over the time steps $k = 1, \dots, T$, as this is implicitly done through the MPC law defined above. However, this problem is still not immediately solvable due to the presence of the probabilistic operators. We attempt to address this challenge next using a simulation optimization paradigm (Amaran et al., 2016).

Remark 1. It is important to note that the number of model parameters $n_{\theta,m}$ in (6) can be quite large in generic black-box models, which inherently increases the complexity of the problem. As such, it is advised to encode as much prior knowledge as possible in a particular problem at hand, which will typically result in a gray-box model defined in a reduced parametric space.

Remark 2. Although we focused on a nominal MPC formulation above, this could straightforwardly be replaced with robust or stochastic MPC methods that directly account for uncertainty within the predictions. However, this will come at the cost of more expensive closed-loop simulations, as well as the need to develop an uncertainty description. Again, one can imagine that some parameters of the uncertainty distribution θ_u could also be treated as controller tuning parameters.

Remark 3. Whenever the state of the model (6) is measurable, we can ignore the state estimator (9). It is worthwhile noting, however, that the estimator could still be useful whenever it is desired to adapt some of the model parameters online due to unexpected drifts or faults.

4. PROPOSED CLOSED-LOOP SIMULATION OPTIMIZATION METHOD

4.1 Formulation as simulation optimization problem

Let $f(\theta) = \mathbb{E}_\delta\{J(y_{1:T}, u_{1:T})\}$ denote the expected performance and $c(\theta) = \mathbb{P}_\delta\{y_{1:T} \in \mathbb{Y}_{1:T}\} - 1 + \epsilon$ be a shorthand for the chance constraint. We can now restate (11) in the following compact manner

$$\min_{\theta \in \Theta} \ f(\theta) \ \ \text{s.t.} \ \ c(\theta) \geq 0. \quad (12)$$

As discussed in (Paulson and Mesbah, 2018), the uncertainty propagation steps needed to evaluate $f(\theta)$ and $c(\theta)$ can be performed using a variety of different techniques. Since the number of uncertainties $n_\delta = n_x + n_w T + n_v(T+1)$ grows with the number of time steps T, Monte Carlo (MC) sampling is generally a good choice since its convergence rate is independent of the n_δ and instead only depends on the number of samples. Using MC sampling, the objective and constraints in (12) can then be approximated as (Kleywegt et al., 2002)

$$f(\theta) \approx M^{-1} \textstyle\sum_{j=1}^{M} J(y^j_{1:T}, u^j_{1:T}), \quad (13a)$$
$$c(\theta) \approx M^{-1} \textstyle\sum_{j=1}^{M} \left(\mathbf{1}_{\mathbb{Y}_{1:T}}(y^j_{1:T}) - 1 + \epsilon\right), \quad (13b)$$

where M denotes the number of samples, $\mathbf{1}_A(x)$ denotes the indicator function over the set A (1 when $x \in A$ and 0 otherwise), $\{\delta^1, \dots, \delta^M\}$ are independent and identically distributed (i.i.d.) samples of $\delta \sim p(\delta)$, and $(y^j_{1:T}, u^j_{1:T})$ are the simulated closed-loop output and input sequences given the jth uncertainty sample δ^j. It is worth noting that the only assumption about the system thus far is that we have the ability to execute (10) for any choice of tuning parameters $\theta \in \Theta$, and every simulation is performed under an i.i.d. sample of δ.

Since a new set of uncertainty samples is drawn every time we run the closed-loop "simulator" (10), the sample average approximations in (13) produce stochastic/noisy observations of the objective and the constraints

$$y^f = f(\theta) + \varepsilon^f, \quad y^c = c(\theta) + \varepsilon^c, \quad (14)$$

where ε^f and ε^c represent the observation noise for $f(\theta)$ and $c(\theta)$, respectively. For any M, these estimators are unbiased, i.e., $\mathbb{E}_{\delta^1,\ldots,\delta^M}\{y^f \mid \theta\} = f(\theta)$ and $\mathbb{E}_{\delta^1,\ldots,\delta^M}\{y^c \mid \theta\} = c(\theta)$; however, these observations may have relatively high variance unless M is large. In general, it may not be possible to select a large M since the closed-loop simulation or experiment can be prohibitively costly. Instead these functions can only be evaluated on the order of 100 times or less, meaning M cannot be large enough to ignore noise in the observations. As such, we want to utilize an algorithm that can systematically explore the tuning parameter space (relative to random or grid-based search methods) and can also accommodate noisy objective and constraint observations.

4.2 Overview of Bayesian optimization framework

BO is a family of algorithms that can solve black-box optimization problems in the presence of noisy observations. The basic idea in BO is to construct a surrogate model of the objective function $f(\theta)$ using a set of initial n observations denoted by $\mathcal{D}_n = \{(\theta_{1:n}, y^f_{1:n})\}$. The statistical surrogate model provides a posterior distribution of the objective function that can be combined with an acquisition function to decide where to sample next. The acquisition function is defined in a way to tradeoff between uncertainty (related to variance) and performance (related to mean) at unexplored points $\theta \in \Theta$. The new observation is added to the dataset, i.e., $\mathcal{D}_{n+1} = \{\mathcal{D}_n, (\theta_{n+1}, y^f_{n+1})\}$, which is then used to update the surrogate model. This process continues until the surrogate model converges to the global solution, or the maximum number of iterations is reached. See, e.g., Shahriari et al. (2015) for a recent review of BO.

First, we discuss the traditional BO strategy for (12) that neglects constraints $c(\theta) \geq 0$ and then describe an extension that modifies the acquisition function to account for the probability that constraints are satisfied. In this work, we exclusively use Gaussian process (GP) surrogate models (Rasmussen, 2003). In particular, we assume that the objective function has a GP prior of the form $f(\theta) \sim \mathcal{GP}(\mu_0, k)$, where $\mu_0 : \Theta \to \mathbb{R}$ is the prior mean and $k : \Theta \times \Theta \to \mathbb{R}$ is the prior covariance function. There are many possible choices for the covariance function. In this work, we selected the squared exponential function

$$k(\theta, \theta') = \exp\left(-\frac{1}{2l^2}\|\theta - \theta'\|^2\right), \tag{15}$$

where l is a length-scale parameter. Under the GP prior and assumed i.i.d. Gaussian noise $\varepsilon^f \sim \mathcal{N}(0, \sigma^2)$, the function evaluations $y^f_{1:n}$ are jointly Gaussian

$$y^f_{1:n} \sim \mathcal{N}(\mathbf{m}, \mathbf{K} + \sigma^2 I_n), \tag{16}$$

where $[\mathbf{m}]_i = \mu_0(\theta_i)$ are the elements of the mean vector and $\mathbf{K} \in \mathbb{R}^{n\times n}$ is a symmetric covariance matrix with elements $[\mathbf{K}]_{i,j} = k(\theta_i, \theta_j)$. This implies the corresponding function value $f(\theta)$ at any test point θ must be jointly Gaussian with $y^f_{1:n}$, i.e.,

$$\begin{bmatrix} y^f_{1:n} \\ f(\theta) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{m} \\ \mu_0(\theta) \end{bmatrix}, \begin{bmatrix} \mathbf{K} + \sigma^2 I_n & \mathbf{k}(\theta) \\ \mathbf{k}(\theta)^\top & k(\theta,\theta) \end{bmatrix}\right), \tag{17}$$

where $\mathbf{k}(\theta) \in \mathbb{R}^n$ is a vector of covariance terms between $\theta_{1:n}$ and θ, i.e., $[\mathbf{k}(\theta)]_i = k(\theta_i, \theta)$. Due to the properties of joint Gaussian random variables, we find that the posterior distribution $p(f(\theta)|y^f_{1:n}, \theta_{1:n}, \theta)$ of the objective, given all available noisy observations, is Gaussian with the following mean and covariance (Rasmussen, 2003)

$$\mu_n(\theta) = \mu_0(\theta) + \mathbf{k}(\theta)^\top(\mathbf{K} + \sigma^2 I_n)(y^f_{1:n} - \mathbf{m}), \tag{18a}$$

$$\sigma^2_n(\theta) = k(\theta,\theta) - \mathbf{k}(\theta)^\top(\mathbf{K} + \sigma^2 I_n)^{-1}\mathbf{k}(\theta). \tag{18b}$$

Given this posterior distribution, we still need to optimize over an acquisition function $\alpha(\theta; \mathcal{D}_n)$ to search for the next optimal sampling point θ_{n+1}. Here, we focus on the expected improvement (EI) criteria that measures the expected amount by which the current objective will be improved over some incumbent (or current best) solution. The EI acquisition function has been shown to be analytically computable for a GP model

$$\alpha_{\mathrm{EI}}(\theta; \mathcal{D}_n) = \sigma_n(\theta)(z_n(\theta)\Phi(z_n(\theta)) + \phi(z_n(\theta)), \tag{19}$$

where $z_n(\theta) = (\eta - \mu_n(\theta))/\sigma_n(\theta)$, Φ and ϕ are, respectively, the standard Gaussian cumulative density and probability density functions, and η is the incumbent solution. The "best" choice of η depends on the context; for deterministic objective evaluations, it is often set to the best observed value $\eta = \max_{i\in\{1,\ldots,n\}} y^f_i$. However, as discussed in (Wang and de Freitas, 2014), this choice can be quite fragile in the stochastic setting and thus we use the best mean value $\eta = \min_{\theta\in\Theta} \mu_n(\theta)$ as a more reasonable alternative.

4.3 Including constraints in expected improvement

To handle the black-box constraints, we now modify the acquisition function to show improvement only when $c(\theta) \geq 0$ holds. Similarly to the objective, we model the constraint function $c(\theta)$ with a GP prior whose evaluations are corrupted with Gaussian noise. We must then weight the original EI in (19) by the probability of the constraints being satisfied. This results in the expected improvement with constraints (EIC) that can be analytically computed as follows (Gardner et al., 2014)

$$\begin{aligned}\alpha_{\mathrm{EIC}}(\theta; \mathcal{D}_n) &= \alpha_{\mathrm{EI}}(\theta; \mathcal{D}_n)\mathbb{P}\{c(\theta) \geq 0 | \mathcal{D}^c_n\}, \\ &= \alpha_{\mathrm{EI}}(\theta; \mathcal{D}_n)\Phi\left(\frac{\mu^c_n(\theta)}{\sigma^c_n(\theta)}\right),\end{aligned} \tag{20}$$

where $\mathcal{D}^c_n = \{(\theta_{1:n}, y^c_{1:n})\}$ is the set of n noisy constraint observations and $\mu^c_n(\theta)$ and $\sigma^c_n(\theta)$ are, respectively, the posterior predictive mean and standard deviation of the GP surrogate model for $c(\theta)$ – similarly defined to (18) for the objective. Due to the presence of constraints, we must modify the definition of the incumbent value η to be the best mean value such that the constraints are satisfied. Since we cannot guarantee that constraints are satisfied given only noisy observations, constraint satisfaction can be enforced in a probabilistic sense

$$\eta = \min_{\theta\in\Theta} \ \mu_n(\theta) \ \text{ s.t. } \ \Phi\left(\frac{\mu^c_n(\theta)}{\sigma^c_n(\theta)}\right) \geq 1 - \beta, \tag{21}$$

for a relatively small $\beta \in (0, 1)$. However, we note that the existence of a feasible solution to (21) cannot be guaranteed due to, e.g., limited data available for the constraints. To overcome this challenge, the factor $\alpha_{\mathrm{EI}}(\theta; \mathcal{D}_n)$ in (20) can be ignored to search only for a feasible point by maximizing the probability of constraint satisfaction (Gelbart et al., 2014).

In our proposed strategy, the next sampling point θ_{n+1} is obtained by solving the optimization problem

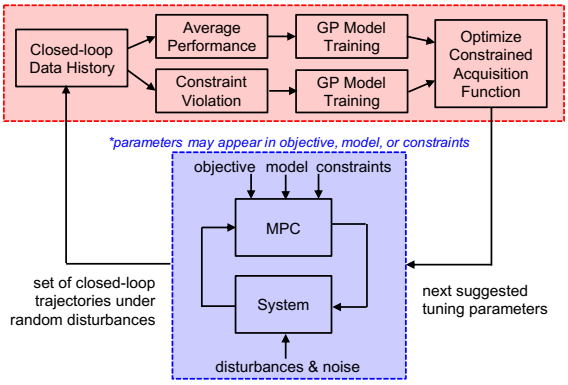


Fig. 1. Illustration of how constrained BO method can be used for data-driven automatic MPC tuning under constraints. The blue box denotes the closed-loop experiment or simulation that must be performed 1 or more times for fixed tuning parameters. The red box shows the constrained BO steps, which must be executed to decide the next best tuning parameter values to use for closed-loop data collection.

$$\theta_{n+1} = \underset{\theta\in\Theta}{\operatorname{argmax}}\ \alpha_{\mathrm{EIC}}(\theta;\mathcal{D}_n). \tag{22}$$

Since the acquisition function is cheap to evaluate (relative to f and c), its maximization can be carried out more efficiently. An overview of various solution methods for (22) that can avoid local solutions is provided in (Gardner et al., 2014). An illustration of the proposed automated MPC tuning strategy under uncertainty using constrained BO is shown in Figure 1.

Remark 4. There are several hyperparameters in the GP regression models that may appear in the mean function, covariance kernel (e.g., l), or the observation model (e.g., noise level σ^2). There are two main ways for handling these hyperparameters. The most involved way is to marginalize out the uncertainty in these hyperparameters by determining their posterior distribution given the data $\mathcal{D}_n$ using Bayes' rule. The acquisition function must then be integrated over this posterior, which can be approximated using some sampling strategy. A simpler approach is to approximate this integral in terms of a Dirac delta measure for the posterior at the maximum likelihood or maximum *a posteriori* point estimate. These point estimates can be found by either maximizing the marginal likelihood or the unnormalized posterior for which standard optimization methods are readily available.

Remark 5. The assumed Gaussian noise model with constant variance for the objective and constraint evaluations in (14) is an approximation made for simplicity. In reality, the noise level may depend on the specific value of θ, which should be accounted for to obtain a more accurate GP surrogate model. One way to do this is to model the noise with a second GP model, as discussed in (Goldberg et al., 1998). Future work is needed to best incorporate input-dependent noise models into constrained BO.

5. NUMERICAL EXAMPLE

5.1 Plant model description and control objectives

We demonstrate the proposed approach on a nonlinear continuously stirred tank reactor (CSTR) problem adapted from (Paulson and Mesbah, 2018). The following set of reactions are assumed to take place in the CSTR

$$\mathrm{A} \to \mathrm{B} \to \mathrm{C}, \quad 2\mathrm{A} \to \mathrm{D}.$$

The plant model (1) is assumed to be a discretized version of the following set of ordinary differential equations

$$\dot{c}_{\mathrm{A}} = F(c_{\mathrm{A0}} - c_{\mathrm{A}}) - k_1 c_{\mathrm{A}} - k_3 c_{\mathrm{A}}^2, \tag{23a}$$
$$\dot{c}_{\mathrm{B}} = -F c_{\mathrm{B}} + k_1 c_{\mathrm{A}} - k_2 c_{\mathrm{B}}, \tag{23b}$$
$$\dot{T}_{\mathrm{R}} = F(T_{\mathrm{in}} - T_{\mathrm{R}}) + \frac{k_{\mathrm{W}} A}{\rho c_{\mathrm{p}} V_{\mathrm{R}}}(T_{\mathrm{K}} - T_{\mathrm{R}}) \tag{23c}$$
$$- \frac{k_1 c_{\mathrm{A}} \Delta H_{\mathrm{AB}} + k_2 c_{\mathrm{B}} \Delta H_{\mathrm{BC}} + k_3 c_{\mathrm{A}}^2 \Delta H_{\mathrm{AD}}}{\rho c_{\mathrm{p}}},$$
$$\dot{T}_{\mathrm{K}} = \frac{1}{m_{\mathrm{K}} c_{\mathrm{pK}}}(\dot{Q}_{\mathrm{K}} + k_{\mathrm{W}} A (T_{\mathrm{R}} - T_{\mathrm{K}})), \tag{23d}$$

where c_{A} and c_{B} denote the concentrations of species A and B, respectively, T_{R} is the reactor temperature, and T_{K} is the coolant temperature. The control input is the feed flowrate $F = \dot{V}_{in}/V_R$ and the reaction kinetics exhibit the Arrhenius dependence

$$k_i = k_{0,i} e^{\frac{-E_{A,i}}{R(T_R+273.15)}}, \quad \forall i \in \{1,2,3\}. \tag{24}$$

The sampling time was chosen to be $\delta t = 0.005$ h. We assume that only species B and the reactor temperature are measurable online such that $y_k = (c_{\mathrm{B},k}, T_{\mathrm{R},k}) + v_k$, where $v_k \sim \mathcal{N}(0, \Sigma_v)$ is noise with $\Sigma_v = \mathrm{diag}(\sigma_B^2, \sigma_R^2)$. For simplicity, we did not consider process noise. The model parameters can be found in Table 1 and the initial conditions and relevant constraints are summarized in Table 2. Since only noisy temperature measurements are available, the bounds on T_{R} are enforced as a chance constraint. Instead of the single constraint (4), we enforce constraints separately at each time point, i.e., $\mathbb{P}_\delta\{y_k \in \mathbb{Y}_k\} \geq 1 - \epsilon_k$ where $\epsilon_k = 0.05$ for all $k = 1, \ldots, T$. This does not change the derivations shown in Section 4 because we can always define $c(\theta) = \min_{k\in\{1,\ldots,T\}}\{\mathbb{P}_\delta\{y_k \in \mathbb{Y}_k\} - 1 + \epsilon_k\} \geq 0$ in terms of the smallest probability over time. We assume that our closed-loop objective is to maximize the production of component B

$$n_{B,Total} = \int_0^{t_f} \dot{n}_B(t)dt \approx \sum_{k=0}^{T-1} V_{\mathrm{in}} c_{\mathrm{B},k} F_k \delta t, \tag{25}$$

such that the performance cost is $J(y_{1:T}, u_{1:T}) = -n_{B,Total}$, where $t_f = 0.2$ h is the final time of the experiment and $T = 40$ sampling time intervals.

5.2 Parametrized formulation of MPC policy

We emphasize the fact that the plant model (23)–(24) is a black-box in this work and, thus, cannot be used in model-based control design. Instead, we look to learn a control-relevant model of the form (6). In particular, we focus on an NARX model structure

$$y_k = f_{\mathrm{NARX}}(y_{k-1}, , \ldots, y_{k-L_y}, u_{k-1}, \ldots, u_{k-L_u}), \tag{26}$$

where L_y and L_u denote the number of output and input lags, respectively. The function f_{NARX} can be any nonlinear function such as a polynomial or a neural network. In this work, we selected a second-order polynomial that ignores the interaction terms. Although in principle L_y and L_u could be treated as model parameters θ_m, we decided to fix them to $L_y = L_u = 1$ to ensure the tuning

Table 1. Parameter values for CSTR case study.

Parameter	Value	Unit
$k_{0,1}$	1.287 10^{12}	h^{-1}
$k_{0,2}$	1.287 10^{12}	h^{-1}
$k_{0,3}$	9.043 10^{9}	L mol^{-1} h^{-1}
$E_{A,1}/R$	9758.3	K
$E_{A,2}/R$	9758.3	K
$E_{A,3}/R$	7704.0	K
ΔH_{AB}	4.2	kJ mol^{-1}
ΔH_{BC}	−11.0	kJ mol^{-1}
ΔH_{AD}	−41.85	kJ mol^{-1}
ρ	0.9342	kg L^{-1}
c_p	3.01	kJ kg^{-1} K^{-1}
c_{pK}	2.0	kJ kg^{-1} K^{-1}
A	0.215	m^2
V_R	10.01	L
m_k	5.0	kg
T_{in}	130.0	°C
k_W	4032	kJ h^{-1} m^{-2} K^{-1}
$\dot{Q}_K$	−4500	kJ h^{-1}

Table 2. Initial conditions and constraints for CSTR case study.

Variable	Init. cond.	Min.	Max.	Unit
c_A	1.0	–	–	mol/L
c_B	1.0	–	–	mol/L
T_R	100.0	100.0	150.0	°C
T_K	100.0	–	–	°C
F	–	5.0	35.0	h^{-1}

parameter space was not too large. This implies that θ_m is composed of 14 coefficients – 7 for each $n_y = 2$ output that multiplies the basis set $\{1, [y_k]_1, [y_k]_2, u_k, [y_k]_1^2, [y_k]_2^2, u_k^2\}$. We fixed the MPC prediction horizon to be $N_p = 10$, as performance was found to be relatively insensitive to this choice. We selected an economic MPC objective function related to the moles of B produced, i.e., $\tilde{\ell}(y_{i|k}, u_{i|k}) = V_{in} c_{B,i|k} F_{i|k} \delta t$. The reactor temperature constraint is active in this problem such that it can easily be violated whenever the prediction model is not very accurate. To address this, we include a backoff parameter θ_b that we wish to select using the proposed constrained BO method, which results in a total of $n_\theta = 14 + 1 = 15$ tuning parameters to be optimized.

5.3 Automated MPC tuning using constrained BO and comparison to sequential model identification

Given the black-box plant description and the parametrized MPC policy, we can now solve (12) using the constrained BO method summarized in Section 4. The `bayesopt` function in Matlab was used to handle the GP construction of the objective and constraints, as well as specify the EIC criteria using the "coupled constraint" option. We explicitly set the objective and constraint evaluations to be stochastic such that the variance is included as a hyperparameter in the corresponding GP models. We budgeted a total of $N_{iter} = 40$ iterations, with the first 5 being randomly chosen within the assumed parameter space $\Theta = [-2, 2]^{14} \times [0, 0.2]$, i.e., all NARX parameters are bounded by $[-2.2]$ and the backoff $\theta_b \in [0, 0.2]$ [1] .

[1] The NARX model is constructed using scaled data, so that all inputs and outputs are bounded within $[0, 1]$.

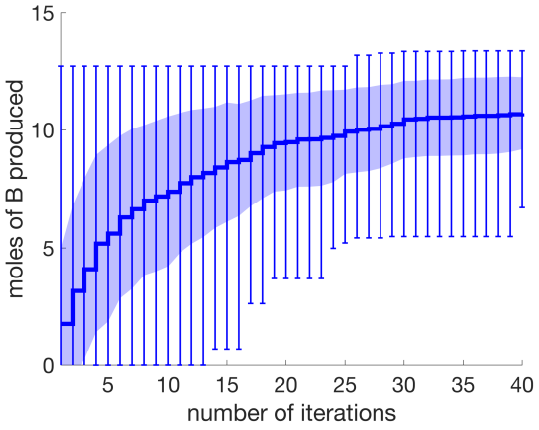
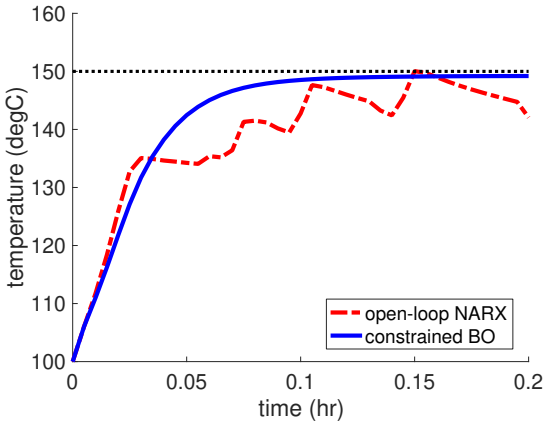


Fig. 2. The maximum performance (moles of B) versus number of BO iterations for 100 replicate runs. The dark blue line represents the mean value, the shaded region is +/- one standard deviation, and the error bars show the minimum and maximum values.

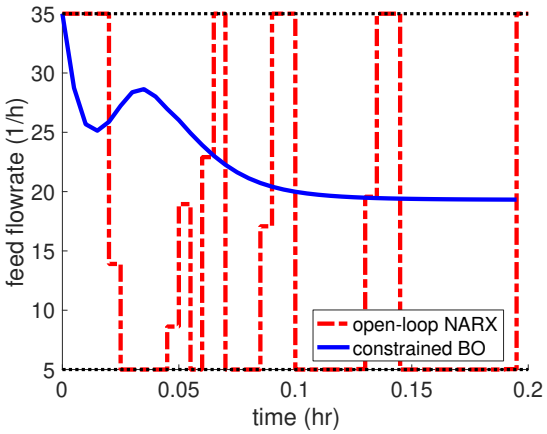
To focus on constraint handling first, we initially set the measurement noise variances to zero. The performance evolution, measured in terms of total moles of B produced, over the number of iterations is shown in Figure 2 for 100 independent runs of the constrained BO algorithm. We see a consistent improvement in performance up until around 30 iterations – we also see that the variance in the final result reduces as number of iterations increases. The resulting closed-loop temperature and feed flowrate profiles are shown in Figure 3, which tightly satisfy constraints. To better contextualize these results, we compare this automated model learning procedure to the traditional open-loop NARX identification paradigm (see red dash-dotted lines in Figure 3). In particular, we performed pseudo random binary step tests on the plant every 100 time steps to collect a total of 3000 input-output data points. This open-loop dataset was used to train an NARX model (26) using Matlab's System Identification Toolbox, which achieved high accuracy on a holdout dataset, i.e., prediction accuracy of 91.5%.

Even though the prediction accuracy is high, the resulting closed-loop performance is poor in the sense that we observe significant variability in the control input profile and minor temperature constraint violations. In fact, the moles of B produced was 5.62, which is less than half of the 12.17 obtained with the constrained BO method. This is likely due to the fact that the input-output data were not sufficiently informative in the regime of high-performance control. As such, we can interpret the proposed strategy as a way to select control-oriented prediction models that are suited to the task at hand – this is important because the desired set of closed-loop trajectories usually represents a much smaller slice of the input-output space than that used in open-loop identification techniques.

Lastly, to demonstrate that the approach is capable of operating in the presence of uncertainty, we repeated the analysis above with measurement noise variances $\sigma_B = 0.2$ mol and $\sigma_R = 10$°C. To keep a small experimental budget, we selected $M = 1$. Due to space limitations, we only show the resulting closed-loop temperature profiles for the open-loop NARX and constrained BO identified models



(a) temperature profile



(b) feed flowrate profile

Fig. 3. Closed-loop (a) output and (b) input profiles for best solution obtained with BO compared to an open-loop identified NARX model for the noise-free case.

in Figure 4. It is interesting to note that the open-loop identified NARX model appears to be quite susceptible to overfitting, which leads to much higher variability and constraint violations than in the noise-free case. On the other hand, the constrained BO method is able to select NARX coefficients that produce temperature profiles with significantly lower variance and, through proper tuning of the backoff, completely avoids constraint violations. We also emphasize that these results were obtained using the same number of runs as in the deterministic case (only one experiment performed for a random noise sequence). This highlights the value of using a GP model that can explicitly account for noisy objective and constraint violations, as it helps guide the search in a way that is not overly optimistic about any one observation.

6. CONCLUSIONS AND FUTURE WORK

We presented an automated performance-driven MPC tuning strategy for black-box systems in the presence of uncertainty. Since the MPC tuning parameters generally affect the closed-loop performance and constraints in non-convex and non-smooth ways, the automated tuning strategy relies on derivative-free simulation optimization methods. In particular, we used a constrained variant of Bayesian optimization that utilizes state-of-the-art surrogate models for the objective and constraint functions constructed from noisy observations. The effectiveness of the proposed tuning strategy was demonstrated for control-relevant model learning on a nonlinear continuously stirred tank reactor case study. We observed a more than two-fold improvement in performance compared to the traditional open-loop model identification methods. There are several interesting directions for future work such as development of novel constrained Bayesian optimization methods that can more readily scale to high-dimensional problems and attempting to account for the noise levels in the objective and constraint evaluations that can vary with the tuning parameter values.

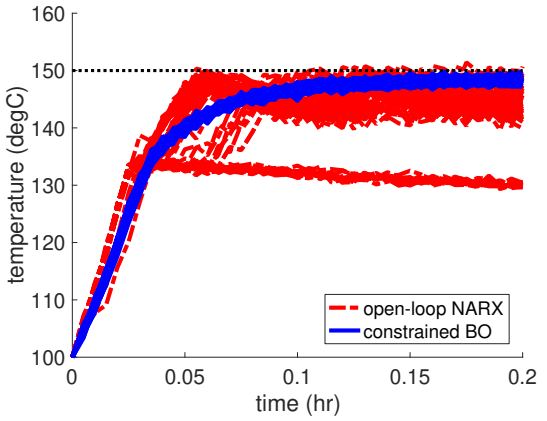


Fig. 4. Closed-loop temperature profile obtained using the proposed constrained BO method compared to an open-loop identified NARX model in the presence of measurement noise for 100 Monte Carlo realizations.

REFERENCES

Amaran, S., Sahinidis, N.V., Sharda, B., and Bury, S.J. (2016). Simulation optimization: A review of algorithms and applications. *Annals of Operations Research*, 240, 351–380.

Bansal, S., Calandra, R., Xiao, T., Levine, S., and Tomiin, C.J. (2017). Goal-driven dynamics learning via Bayesian optimization. In *Proceedings of the 56th IEEE Conference on Decision and Control*, 5168–5173. Melbourne.

Berkenkamp, F., Schoellig, A.P., and Krause, A. (2016). Safe controller optimization for quadrotors with Gaussian processes. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 491–496. Stockholm.

Bonzanini, A.D., Paulson, J.A., Makrygiorgos, G., and Mesbah, A. (2021). Fast approximate learning-based multistage nonlinear model predictive control using gaussian processes and deep neural networks. *Computers & Chemical Engineering*, 145, 107174.

Forgione, M., Piga, D., and Bemporad, A. (2019). Efficient calibration of embedded MPC. *arXiv preprint arXiv:1911.13021*.

Gardner, J.R., Kusner, M.J., Xu, Z.E., Weinberger, K.Q., and Cunningham, J.P. (2014). Bayesian optimization with inequality constraints. In *Proceedings of the International Conference on Machine Learning*, 937–945. Beijing.

Garriga, J.L. and Soroush, M. (2010). Model predictive control tuning methods: A review. *Industrial & Engi-*

neering Chemistry Research, 49(8), 3505–3515.

Gelbart, M.A., Snoek, J., and Adams, R.P. (2014). Bayesian optimization with unknown constraints. In *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence*, 250–259. Quebec.

Gevers, M. (2005). Identification for control: From the early achievements to the revival of experiment design. *European Journal of Control*, 11, 335–352.

Goldberg, P.W., Williams, C.K., and Bishop, C.M. (1998). Regression with input-dependent noise: A Gaussian process treatment. In *Advances in Neural Information Processing Systems*, 493–499.

Hernández-Lobato, J.M., Gelbart, M.A., Adams, R.P., Hoffman, M.W., and Ghahramani, Z. (2016). A general framework for constrained Bayesian optimization using information-based search. *Journal of Machine Learning Research*, 17(160), 1–53.

Hoshiya, M. and Saito, E. (1984). Structural identification by extended Kalman filter. *Journal of Engineering Mechanics*, 110, 1757–1770.

Khosravi, M., Behrunani, V., Myszkorowski, P., Smith, R.S., Rupenyan, A., and Lygeros, J. (2020). Performance-driven cascade controller tuning with Bayesian optimization. *arXiv preprint arXiv:2007.12536*.

Kleywegt, A.J., Shapiro, A., and Homem-de Mello, T. (2002). The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12, 479–502.

Lu, Q., Kumar, R., and Zavala, V.M. (2020). MPC controller tuning using Bayesian optimization techniques. *arXiv preprint arXiv:2009.14175*.

Neumann-Brosig, M., Marco, A., Schwarzmann, D., and Trimpe, S. (2019). Data-efficient autotuning with Bayesian optimization: An industrial control study. *IEEE Transactions on Control Systems Technology*, 28(3), 730–740.

Paulson, J.A. and Mesbah, A. (2018). Nonlinear model predictive control with explicit backoffs for stochastic systems under arbitrary uncertainty. *IFAC-PapersOnLine*, 51(20), 523–534.

Paulson, J.A. and Mesbah, A. (2020). Data-driven scenario optimization for automated controller tuning with probabilistic performance guarantees. *IEEE Control Systems Letters*, 5(4), 1477–1482.

Piga, D., Forgione, M., Formentin, S., and Bemporad, A. (2019). Performance-oriented model learning for data-driven MPC design. *IEEE Control Systems Letters*, 3(3), 577–582.

Rasmussen, C.E. (2003). Gaussian processes in machine learning. In *Summer School on Machine Learning*, 63–71. Springer.

Rawlings, J.B. and Mayne, D.Q. (2009). *Model predictive control: Theory and design*. Nob Hill Publishing, Madison, Wisconsin.

Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., and De Freitas, N. (2015). Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1), 148–175.

Snoek, J., Larochelle, H., and Adams, R.P. (2012). Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, 2951–2959.

Wang, Z. and de Freitas, N. (2014). Theoretical analysis of Bayesian optimisation with unknown Gaussian process hyperparameters. *arXiv preprint arXiv:1406.7758*.