

Chance-constrained state feedback control law design for nonlinear systems under uncertainties

Yu Yang, Nguyen Cam Thuy Dam

Abstract—A chance-constrained full-state feedback control law is designed to regulate nonlinear systems under uncertainties. The proposed scheme utilizes Monte Carlo sampling to generate multiple scenarios, formulates the optimal control problem as a scenario-based nonlinear optimization, and develops a sequential algorithm to obtain probabilistic feasible solutions. The resulting controller offers three advantages: First, the optimization-based design minimizes the tracking error across considered scenarios. Second, the sampling complexity is determined adaptively and the chance of constraint violation is bounded with a guaranteed confidence interval. Third, the sequential algorithm can reach a probabilistic feasible solution faster than directly using a state-of-the-art solver for the full-scenario optimization problem. Two case studies, including a CSTR and fermentor, are presented to demonstrate the effectiveness of the proposed method.

I. INTRODUCTION

The model-based optimal controller design has been studied and applied in various domains. Although it outperforms simple PID controller via model prediction and explicitly handling constraints, its sensitivity to uncertainty may degrade the control performance. To address that issue, the robust model-based controller attracts considerable attention and has progressed significantly in the last few decades. The essential idea is to minimize the worst-case tracking error through minimax optimization [1] and enforce constraint satisfaction under all possible scenarios. For robust nonlinear control, the polytopic uncertainty set can be employed to model the system, and the resulting convex control problem is solved by a semidefinite program to generate a conservative solution [2]. Compared with the traditional robust controller, a chance-constrained controller allows a certain probability of constraint violation, denoted as the risk level ϵ , to provide more flexibilities, and does not sacrifice performance too much. Following this research thrust, several methodologies have been proposed.

For a linear system with normally distributed uncertain parameters, the chance-constrained optimization can be cast as a second-order cone program and solved efficiently [3]. The works of [4], [5], [6] took this advantage to develop robust model predictive control under chance constraints. In [7], the controller for linear systems with stationary additive disturbances was synthesized to meet chance constraints by solving a semi-definite program. Ref. [8] studied both state and output measurement noise in the linear system and

developed a hybrid method to combine the advantage of analytical and sampling approaches for chance-constrained control. The scenario-based approximation was shown to be a robust and distribution-free approach for chance-constrained control of convex systems [9]. Several works [10], [11], [12] have been proposed subsequently to reduce the required number of scenarios and improve the quality of solutions.

The above literature focuses on linear or convex systems. However, nonlinear, non-convex dynamics are more general in real applications, and thus deserve more study. In [13], the chance-constrained nonlinear control problem was solved through policy gradient and constraint tightening. The backoffs were adjusted via Bayesian optimization and added into critical constraints to ensure the probabilistic feasibility that allows only ϵ chance of constraint violation. Such an approach needs to learn both controller and backoff parameters through a number of iterations, but may not sufficiently make use of the model structure. In [14], stochastic nonlinearity was considered, and an H_∞ output-feedback controller was designed, but it assumed the first and second moments of nonlinear terms are known and used expectation to overly approximate the chance constraints. The polynomial chaos expansion has been applied to ensure the expectation constrain or chance constraint in nonlinear control [15]. The scenario-based optimization control was also extended to non-convex systems via posterior evaluation that determines the reliability of a solution and necessary sampling complexity [16].

In our previous research, authors have developed a scenario-based chance-constrained optimization algorithm and determined the sampling complexity to ensure the probabilistic feasibility for nonlinear systems that can be convexified [17]. In this paper, we further propose a full-state feedback controller design method through scenario-based optimization and use posterior evaluation to determine the sampling complexity for general nonlinear systems. Our contributions lie in three aspects: First, posterior evaluation in an independent validation scenario set is used to adjust the sampling complexity of the training set; second, the expected tracking error is minimized in addition to construct a probabilistic terminal set within a finite horizon; third, the sequential optimization algorithm obtains a feasible solution by choosing a few support scenarios from the full set to improve computational efficiency.

The rest of this paper is organized as follows: The problem formulation is introduced in Section II. The scenario-based controller design and sampling complexity determination are discussed in Section III. The controllers for a continuously

This work was supported by the National Science Foundation (NSF) with award number 2151497.

Yu Yang (yu.yang@csulb.edu) and Nguyen Cam Thuy Dam are with the Chemical Engineering Department of California State University, Long Beach, CA 90840, USA.

stirred-tank reactor (CSTR) and a fermentor under bounded uncertainties are developed, respectively, in Section IV. Finally, conclusions are drawn in Section V.

Notation. Throughout this paper, vectors and matrices are denoted by boldface letters. Underlined variables denote lower bounds (used for N , N_1, N_2 , and S).

II. PROBLEM FORMULATION

A general non-convex discrete system under parametric uncertainties shown below is studied in this paper:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k; \boldsymbol{\theta}_k) \quad (1)$$

where $\mathbf{x}_{k+1} \in \mathbb{R}^n$ is the system state at time instant $k+1$; $\mathbf{u}_k \in \mathbb{R}^m$ is the input; $\boldsymbol{\theta}_k \in \mathbb{R}^r$ is the system uncertain parameter vector that may satisfy any distribution; $\mathbf{f}$ represents system dynamics.

A full-state feedback control law, defined as $\mathbf{u}_k = \mathbf{K}(\mathbf{x}_k - \bar{\mathbf{x}}) + \mathbf{W}$, is designed by solving a chance-constrained program (CCP) offline:

$$\begin{aligned} \min_{\mathbf{K}, \mathbf{W}} \quad & J = \mathbb{E} \left(\sum_{k=0}^L (\mathbf{x}_k - \bar{\mathbf{x}})^2 \right) \quad (\text{CCP}) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k; \boldsymbol{\theta}_k), \\ & \mathcal{P}(\mathbf{G}(\mathbf{x}_k, \boldsymbol{\omega}_k) \in \mathcal{X}, \forall k \in \{0, 1, \dots, H\}) \geq 1 - \epsilon, \end{aligned}$$

where $\mathbf{K}$ and $\mathbf{W}$ are decision variables, representing controller parameters; $\bar{\mathbf{x}}$ is the setpoint state; J is the expected squared tracking error over a finite horizon L ; $\mathbb{E}$ is the expectation operator and $\mathcal{P}$ is the probability; $\mathcal{X}$ is the admissible region of constraint $\mathbf{G}(\mathbf{x}, \boldsymbol{\omega})$; ϵ is a small positive number representing the allowable risk level; $\boldsymbol{\omega}_k$ represents measurement uncertainties. In fact, $\mathbf{G}(\mathbf{x}) \in \mathcal{X}$ can represent any type and number of constraints on $\mathbf{x}$. Thus, (CCP) can be a joint chance-constrained optimization problem, which is non-convex and hard to solve analytically.

(CCP) is similar to the well-known model prediction controller (MPC) in that they all minimize the tracking error over a finite horizon. However, (CCP) is solved offline to generate a state feedback control law instead of the online optimization and receding horizon used in MPC. This is due to the complexity of chance-constrained controller for a non-convex stochastic system. Searching for its optimal solution can be computationally intractable, and thus online optimization should be avoided, especially for large-scale fast dynamic systems.

III. METHODS FOR SOLVING CHANCE-CONSTRAINED OPTIMIZATION

Totally N scenarios can be generated by drawing samples $\{\boldsymbol{\theta}_k^{(1)}, \boldsymbol{\omega}_k^{(1)}; \boldsymbol{\theta}_k^{(2)}, \boldsymbol{\omega}_k^{(2)}; \dots; \boldsymbol{\theta}_k^{(N)}, \boldsymbol{\omega}_k^{(N)}\}$, $\forall k \in \{0, 1, 2, \dots, H\}$ from the distribution of $\boldsymbol{\theta}$ and $\boldsymbol{\omega}$, along the horizon H . Then, the problem (CCP) can be approximated by a deterministic scenario-based problem (SP):

$$\begin{aligned} \min_{\mathbf{K}, \mathbf{W}} \quad & J = \frac{1}{N} \sum_{i=1}^N J^{(i)} = \frac{1}{N} \sum_{i=1}^N \left(\sum_{k=0}^L (\mathbf{x}_k^{(i)} - \bar{\mathbf{x}})^2 \right) \quad (\text{SP}) \\ \text{s.t.} \quad & \mathbf{x}_{k+1}^{(i)} = \mathbf{f}(\mathbf{x}_k^{(i)}, \mathbf{u}_k^{(i)}; \boldsymbol{\theta}_k^{(i)}) \end{aligned}$$

$$\begin{aligned} \mathbf{G}(\mathbf{x}_k^{(i)}, \boldsymbol{\omega}_k^{(i)}) &\in \mathcal{X}, \\ \forall k &\in \{0, 1, \dots, L\}, \forall i \in \{1, 2, \dots, N\}, \end{aligned} \quad (2)$$

where $J^{(i)}$ is the objective value of sampled scenario i . In (SP), the expectation operator in objective function is replaced by the average of tracking error across all sampled scenarios. The chance constraint is approximated by N scenario-based constraints. Even though (SP) is still non-convex, it does not contain any stochastic component and thus can be solved by a deterministic nonlinear programming (NLP) solver under mild N . One may further consider allowing the solution to violate constraint $\mathbf{G}(\mathbf{x}_k^{(i)}, \boldsymbol{\omega}_k^{(i)}) \in \mathcal{X}$ in $\lceil N\epsilon \rceil$ scenarios. However, that scheme requires a much larger number of N and has to introduce N binary variables [18], rendering (SP) more difficult to solve.

The key contributions of this paper include:

- Introducing the probabilistic terminal constraint with prolonged horizon.
- Adjusting the sampling complexity N adaptively.
- Solving the (SP) more efficiently under large N scenarios.

A. Probabilistic Terminal Constraint with Prolonged Horizon

For prediction-based control, because only finite horizon L can be applied, there should be a terminal state constraint in (SP) for all scenarios:

$$\mathbf{G}_{\text{terminal}}(\mathbf{x}_L^{(i)}, \boldsymbol{\omega}_L^{(i)}) \in \mathcal{X}_{\text{terminal}}, \quad \forall i \in \{1, 2, \dots, N\} \quad (3)$$

One may assume that the resulting terminal set is a safe region in which the controller can always drive the system approaching to the setpoint. However, that assumption does not hold if uncertainty significantly disturbs system dynamics. Namely, the reliability of a terminal set cannot be guaranteed if only point-wise constraint is enforced. To resolve this issue, we suggest terminal constraint with a prolonged horizon $L' > L$:

$$\begin{aligned} \mathbf{G}_{\text{terminal}}(\mathbf{x}_k^{(i)}, \boldsymbol{\omega}_k^{(i)}) &\in \mathcal{X}_{\text{terminal}}, \\ \forall k &\in \{L, L+1, \dots, L'\}, \forall i \in \{1, 2, \dots, N\} \end{aligned} \quad (4)$$

As L' is large enough, satisfying (4) for all N scenarios means that the system can stay within the terminal set for a long period with high probability. However, the prolonged horizon may increase the computational burden of (CCP), which should be addressed in the proposed algorithm.

B. Determine and Adjust Sampling Complexity N

The scenarios number N considerably impacts the performance of solution in (SP). Because the expectation is approximated by the average, N should be large enough to incorporate all representative scenarios. However, it may incur two issues. First, a large number of scenarios introduce more variables in (SP) and thus render it more difficult to solve. Second, because N scenarios are required to meet constraints in (CCP) simultaneously, overly large N may restrict the feasible region and lead to a conservative solution.

The resulting constraint violation chance will be far less than the allowable level ϵ .

As mentioned in [16], for non-convex systems, a posterior evaluation could be a less conservative approach to determine N . Note that $G(\mathbf{x}_k, \omega_k) \in \mathcal{X}$ is a Binomial random variable, Lemma 1 is presented based on the Clopper-Pearson interval [19]:

Lemma 1: For a Binomial distributed random variable, if the observed success among N trials is S , then the $1 - \alpha$ confidence interval of success probability is:

$$[\text{Binv}(1 - \frac{\alpha}{2}; S, N - S + 1), \text{Binv}(1 - \frac{\alpha}{2}; S + 1, N - S)], \quad (5)$$

where Binv represents the quantile of beta distribution.

According to [19], the Clopper-Pearson interval is relatively conservative, but strictly adheres to the prescription of confidence interval $1 - \alpha$. Two scenario set are generated based on the Clopper-Pearson interval for training and validation, respectively. The training set has N_1 scenarios to build the controller by solving (SP), and can be further adjusted. The second set is for validation with much larger cardinality N_2 . The controller should guarantee state trajectories to satisfy constraints with high probability under the validation scenario set.

To determine N_1 , we specify the confidence parameter α , let $S = N_1$, and solve Eq. (6) derived from the lower bound in Eq. (5):

$$\text{Binv}(1 - \frac{\alpha}{2}; S, N_1 - S + 1) \geq 1 - \epsilon \quad (6)$$

Here $S = N_1$ is enforced because the solution of (SP) is required to meet constraints under all sampled scenarios. If we let $\alpha = 0.1\%$, and $\epsilon = 5\%$, then solving (6) will yield an integer lower bound of N_1 , denoted as $\underline{N}_1 = 149$. However, it is worthwhile to note that using $N_1 = \underline{N}$ in (SP) does not guarantee the probabilistic feasibility of the solution to be greater than $1 - \epsilon$, because Lemma 1 is only applicable for the posterior evaluation. Hence, we use $\underline{N}_1$ as an initial guess of N_1 and tune the number of scenarios according to the validation results.

There should be $N_2 \gg N_1$ because the validation set is employed to estimate the expected objective value and probabilistic feasibility. Given N_2 , we can use Lemma 1 and replace N_1 by N_2 in Eq. (6) to determine the lower bound on the number of successful scenarios S . For example, if $N_2 = 1000$, $\alpha = 0.1\%$, and $\epsilon = 5\%$, then the lower bound of number S is $\underline{S} = 972$. Here $N_2 = 1000$ is chosen to balance the computational burden of validation and representative. Note that scenarios in the validation set are unseen by the optimization solver, and thus Lemma 1 can be used for posterior validation and find reliable confidence interval. If the solution $\{K^*, W^*\}$ does not pass the validation, it implies that N_1 is not sufficiently large and has to be increased. In such a case, we need to continuously sample new scenarios and attach them to the training set until a scenario with constraint violation under the current controller is found. The new training set then will enable (SP) to yield

a more reliable controller. This process is repeated until the designed regulator passes the validation.

C. Solve Scenario-based Optimization

Once N_1 is determined, sampled scenarios can be generated and resulting (SP) should be solved for controller design. However, a large number N_1 of scenarios with non-linear dynamics render the computation of (SP) challenging and finally become intractable. Hence, a sequential algorithm is developed to find a sub-optimal solution of (SP). The rationale include: (i) only a few constraints are active in the optimal solution. If the scenarios incorporating such active constraints can be pre-selected in (SP), then the resulting optimization is easier to solve; (ii) Terminal constraint with prolonged horizon may significantly increase computational demands. Thus, Eq. (3) is applied for most of scenarios, whereas constraint (4) is enforced only on selected ones.

Let us define the initial support set $\Pi := \{1\}$, a prolonged horizon $L' > L$, and a batch size B . Algorithm 1 shown below will return a pool of feasible solutions for (SP):

- Step 1: Solve (SP) on the scenario set Π and obtain the solution $\{K^*, W^*\}$.
- Step 2: Evaluate the controller performance on $\forall i \in \{1, 2, \dots, N_1\}$ scenarios to calculate $J^{(i)} = (\sum_{k=0}^H (\mathbf{x}_k^{(i)} - \bar{\mathbf{x}})^2)$ and construct a set Ξ incorporating all scenarios with constraint violation.
- Step 3: If $\Xi = \emptyset$, go to Step 4; Otherwise, enumerate Ξ and select B scenarios such that: $i = \arg \max_{i \in \Xi} J^{(i)}$, $\Xi \leftarrow \Xi \setminus i$, $\Pi \leftarrow \Pi \cup \{i\}$. Go back to Step 1.
- Step 4: Evaluate the constraint violation and objective values on the validation set. If a probabilistic feasible solution is not found, re-sample scenarios with larger N_1 and go back to Step 2. If a probabilistic feasible solution is found and its objective value is improved, update J^* and go to Step 5. If a probabilistic feasible solution is found but the objective value is not improved, go to Step 5.
- Step 5: If the maximum number of iteration is reached, terminate Algorithm 1; otherwise, enumerate N_1 scenarios and select B of them such that: $i = \arg \max_{i \in \{1, 2, \dots, N_1\} \setminus \Pi} J^{(i)}$, $\Pi \leftarrow \Pi \cup \{i\}$. Go back to Step 1.

Several comments about Algorithm 1 are presented. First, (SP) is generally solved to a sub-optimal solution in Step 1 because nonlinear, non-convex dynamics are involved in that optimization. Second, Step 3 aims to find scenarios with top B maximum objective values among the violation set Ξ if it is not empty, and then all these scenarios will be integrated into Π . Third, in Step 4, if a feasible solution from the training set cannot make at least $\underline{S}$ scenarios in validation set feasible, then more samples should be generated and attached to the training set. Fourth, when the terminal set constraint is violated in scenario i , Eq. (4) is introduced into (SP) with prolonged length L' for scenario i . Fifth, if there is no constraint violation, we can still attach scenarios with B largest objective values to Π for solution

improvement, as shown in Step 5. Sixth, a balance between computational burden and optimality should be achieved by carefully choosing the maximum iteration number. As more scenarios are incorporated into Π , the objective value is expected to be improved, but the resulting (SP) is more difficult to solve. Finally, a flowchart of Algorithm 1 with N_1 adjustment is shown in Fig. 1.

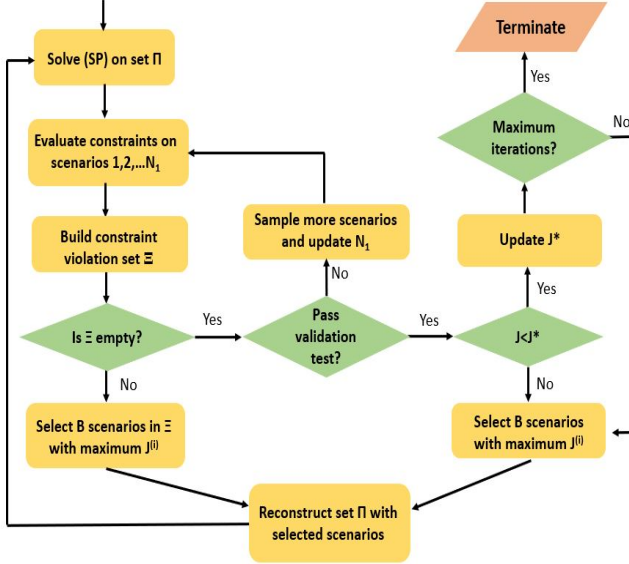


Fig. 1. Algorithm 1 flowchart.

IV. CASE STUDIES

Two case studies about nonlinear stochastic process controller design are presented using scenario-based optimization under the risk level $\epsilon = 5\%$. The problem (SP) is solved through GAMS 32.2.0 with an optimization solver: BARON [20], which enables the multi-start local search and global optimization. The hardware is i7-7500U CPU 2.70 GHZ with 8 GB memory.

A. Continuous Stirred-Tank Reactor (CSTR) Control

In this example, a full-state feedback controller for a CSTR subject to additive uncertainties is developed. The process model is shown below:

$$\dot{C}_A = \frac{F}{V}(C_{A0} - C_A) - k_0 e^{-E/RT_R} C_A + \theta_1, \quad (7)$$

$$\dot{T}_R = \frac{F}{V}(T_{A0} - T_R) - \frac{\Delta H}{\rho C_p} k_0 e^{-E/RT_R} C_A + \frac{Q_\sigma}{\rho C_p V} + \theta_2, \quad (8)$$

where process state is $\mathbf{x} = [x_1, x_2]^T = [C_A, T_R]^T$ in which C_A is concentration and T_R is temperature. Two control inputs are $0 \leq C_{A0} \leq 2 \text{ kmol/m}^3$ and $-250 \leq Q_\sigma \leq 250 \text{ kJ/min}$. Uncertainties θ_1 and θ_2 are uniformly distributed within ranges $[-0.2, 0.2]$ and $[-1, 1]$, respectively. Other model parameters are shown in Table I. Here $C_{A,SP}$ and $T_{R,SP}$ are desired state setpoint. Besides θ_1 and θ_2 , uniformly distributed uncertainties on the initial condition are introduced, such that $x_{1,0} \in [0.75, 0.85]$ and $x_{2,0} \in$

TABLE I
PARAMETERS OF THE CSTR MODEL

$V = 0.1 \text{ m}^3$	$R = 8.314 \text{ kJ/kmolK}$
$C_{A,SP} = 0.5709 \text{ kmol/m}^3$	$T_{R,SP} = 395.4047 \text{ K}$
$\Delta H = -4.78 \times 10^4 \text{ kJ/kmol}$	$k_0 = 72 \times 10^9 \text{ min}^{-1}$
$E = 8.314 \times 10^4 \text{ kJ/kmol}$	$C_p = 0.239 \text{ kJ/kgK}$
$\rho = 1000 \text{ kg/m}^3$	$F = 0.1 \text{ m}^3/\text{min}$

[386, 388]. The model is discretized with the sampling time interval 0.1 minute.

For the controller design, the horizon for tracking error minimization is $L = 15$ and the prolonged horizon is $L' = 20$. Note that concentration and temperature are on the different magnitude. A weighting parameter should be introduced to rescale the objective function. The resulting optimization with two constraints is:

$$\begin{aligned} \min_{\mathbf{K}, \mathbf{W}} J &= \frac{1}{N_1} \sum_{i=1}^{N_1} J^{(i)} = \frac{1}{N_1} \sum_{i=1}^{N_1} \sum_{k=1}^L 100(x_{1,k}^{(i)} - x_{1,SP})^2 \\ &\quad + (x_{2,k}^{(i)} - x_{2,SP})^2 \\ \text{s.t. } (7), (8), \\ &\quad x_{1,k}^{(i)} \leq 1.05, \forall k \in \{1, 2, \dots, L'\}, \\ &\quad (\mathbf{x}_k^{(i)} - \mathbf{x}_{SP}) \Phi (\mathbf{x}_k^{(i)} - \mathbf{x}_{SP})^T \leq 0.75, \\ &\quad \forall k \in \{12, 13, \dots, L \text{ or } L'\}, \forall i \in \{1, 2, \dots, N_1\}, \end{aligned} \quad (9)$$

where $\mathbf{x}_{SP} = [C_{A,SP}, T_{R,SP}]$; the matrix Φ is derived from [22] to represent a region of attraction (ROA) with level 0.75:

$$\Phi = \begin{bmatrix} 51.6785 & 2.3359 \\ 2.3359 & 0.1245 \end{bmatrix}$$

Here two scenario-based constraints are introduced. The concentration is limited by 1.05 and the state is enforced to enter the ROA after $k = 12$ and stay inside.

As mentioned before, the number of scenarios in training and validation set are $N_1 = 149$ and $N_2 = 1000$, respectively. Because CSTR has relatively short settling time and only two states, the full scenario-based controller design problem (9) can be solved directly by using BARON on the training set. It thus provides a good benchmark to evaluate the optimality of Algorithm 1. The results are shown in Table II. Only 1 scenario violation in the validation set ($N_2 = 1000$) implies the probabilistic feasibility with risk level $\epsilon = 5\%$. It is not surprised to see that solving N_1 scenarios simultaneously reaches the optimal solution. However, the proposed Algorithm 1 finds a solution whose objective value is only 0.5% worse. In addition, only one scenario has constraint violation, which implies that $N_1 = 149$ in (9) is sufficient to yield a highly reliable controller for this CSTR. Note that CSTR is a small-scale problem for demonstration purpose. When a large-scale nonlinear process is studied, the computational burden of solving N_1 scenarios simultaneously may become unmanageable by BARON.

The evolution of objective value and violation rate on training set is plotted in Fig. 2 to show the performance

TABLE II
RESULTS OF SCENARIO OPTIMIZATION, CASE 1
($N_1 = 149, N_2 = 1000$)

K^*	Algorithm 1 [5.815 0.254 -32.098 9.707]	Full scenarios [6.357 0.280 -14.948 3.909]
W^*	[0.707, 147.491]	[0.565, 208.232]
Violation on training set	0	0
Violation on validation set	1	1
J on training set	374.285	372.512
J on validation set	384.183	383.085

of Algorithm 1. Here the maximum iteration is set as 15. At the third iteration, a feasible solution is found. As more scenarios are incorporated into set II, the objective value may fluctuate and the optimal solution is selected from the pool of searched feasible solutions. In Fig. 3, the regulated state trajectories under validation scenarios are shown. The region of attraction is also plotted and it demonstrates that constraint violation only happens in one scenario.

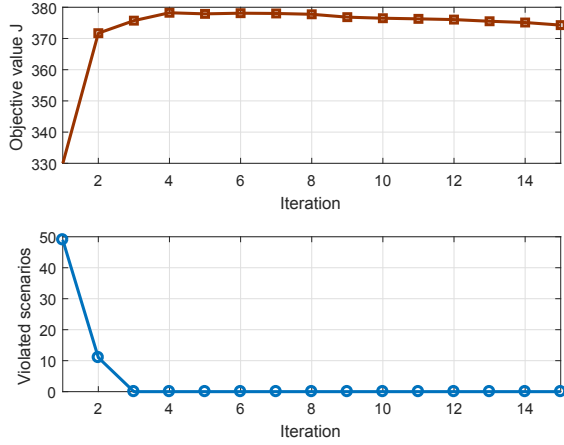


Fig. 2. The objective value and number of constraint violated scenarios on training set case 1.

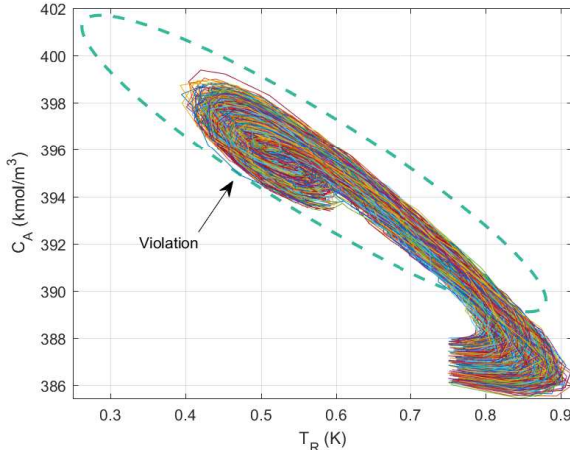


Fig. 3. The state trajectory of 1000 scenarios in case 1 (validation set).

TABLE III
PARAMETERS OF THE FERMENTER MODEL

Parameter	Value
$Y_{x/s}$	0.4 g/g
μ_m	0.48 h ⁻¹
P_m	50 g/L
K_m	1.2 g/L
K_i	22 g/L

B. Fermentor Control

A continuous fermentation process is studied in this subsection. The process model is shown in (10)-(12):

$$\dot{X} = -DX + \mu X, \quad (10)$$

$$\dot{S}_o = D(S_f - S_o) - \frac{1}{Y_{x/s}}\mu X, \quad (11)$$

$$\dot{P} = -DP + (\alpha\mu + \beta)X, \quad (12)$$

where X is the effluent cell-mass or biomass concentration; S_o is the substrate concentration; P is the product concentration; μ is the specific growth rate; $Y_{x/s}$ is the cell-mass yield; α and β are kinetic parameters, subject to bounded uncertainties. The specific growth rate can be described in a few different ways based on the current stage of fermentation. The chosen specific growth rate is shown in Eq. (13).

$$\mu = \frac{\mu_m(1 - \frac{P}{P_m})S_o}{K_m + S_o + \frac{S_o^2}{K_i}}, \quad (13)$$

where μ_m is the maximum specific growth rate; P_m is the product saturation constant; K_m is the substrate saturation constant, and K_i is the substrate inhibition constant. The process state is defined as $\mathbf{x} = [x_1, x_2, x_3] = [X, S_o, P]$ with desired setpoint [7.2925, 5.1687, 24.9369]. The dilution rate D and feed substrate concentration S_f are denoted as inputs u_1 and u_2 , respectively. The model parameter values are from [21], shown in Table III. The sampling time is set as 1 hour. Please note that the proposed method designs a full-state feedback controller offline, and thus the sampling rate does not impact the implementation of the controller.

For the controller design, the horizon for tracking error minimization is $L = 25$ and the prolonged horizon is $L' = 50$ for state constraint. The weighting parameters for X, S_o, P are 2, 2, 1, respectively. The resulting scenario-based optimization is presented in (14):

$$\min_{K, W} J = \frac{1}{N_1} \sum_{i=1}^{N_1} J^{(i)} = \frac{1}{N_1} \sum_{i=1}^{N_1} \sum_{k=1}^L 2(x_{1,k}^{(i)} - x_{1,SP})^2 \quad (14)$$

$$+ 2(x_{2,k}^{(i)} - x_{2,SP})^2 + (x_{3,k}^{(i)} - x_{3,SP})^2$$

$$\text{s.t. (10) - (13),}$$

$$0 \leq u_1 \leq 1, 0 \leq u_2 \leq 30,$$

$$x_{1,k}^{(i)} \geq 5, \forall k \in \{1, 2, \dots, L'\}, \quad (15)$$

$$7.0 \leq x_{1,k}^{(i)} \leq 7.5, 5.0 \leq x_{2,k}^{(i)} \leq 5.4, 24.4 \leq x_{3,k}^{(i)} \leq 25.4, \quad (16)$$

$$\forall k \in \{L, L+1, \dots, L'\}, \forall i \in \{1, 2, \dots, N_1\}.$$

Here the state constraint (15) and terminal constraint (16) are sampled with N_1 scenarios.

The number of validation scenarios is $N_2 = 1000$ and the cardinality of training set starts from $N_1 = 149$. Solving (14) for this three-state slow system requires high computational demand, and thus we terminate the algorithm once a feasible solution on validation set is found. The objective value and the number of scenarios with constraint violation are shown in Fig. 4. Even though the objective values in early iterations are smaller, they are not feasible solutions. A feasible solution on training set is found in 8th iteration, which renders more than $N_2 - \underline{S}$ scenarios on the validation set violating the constraint. Hence, extra scenarios are continuously generated and attached to the training set until $N_1 = 153$ in which an infeasible scenario is obtained. After 10th iteration, a feasible solution on training set is found and it enables more than $N_2 - \underline{S}$ scenarios feasible on the validation set. The entire solution time of Algorithm 1 in case 2 is two-hour, whereas BARON cannot find any feasible solution of the N_1 -scenario problem within three hours. This is due to the long prediction horizon and more state equations in the optimization. The results are summarized in Table IV. We find that the objective value on training and validation set are only with 0.06% difference. In addition, the number of violation scenarios on the validation set is 18, and thus the number of satisfactory scenarios is greater than $\underline{S} = 972$, which implying the probabilistic feasibility.

TABLE IV
RESULTS OF SCENARIO OPTIMIZATION, CASE 2
($N_1 = 153, N_2 = 1000$)

	Algorithm 1
K^*	[-0.004 -0.013 -0.00055 -2.996 2.041 0.429]
W^*	[0.163, 23.814]
Violation on training set	0
Violation on validation set	18
J on training set	385.941
J on validation set	386.188

In Fig. 5, the three-state trajectories in the scenarios of validation set are shown. Even though the system is subject to uncertainties, the proposed controller still can drive the states to a small region around the setpoint defined in Eq. (16) with high chance.

V. CONCLUSION

The controller design for nonlinear stochastic systems is formulated as a scenario-based offline optimization problem. The lower bound on sampling complexity is determined for training and validation set based on the exact Clopper-Pearson interval for Binomial distribution. In addition, the number of scenarios in the training set can be adjusted based on the constraint violation rate in the validation set. The resulting scenario-based large-scale optimization program is efficiently solved by the proposed algorithm through a sequential scheme. Namely, the support scenarios

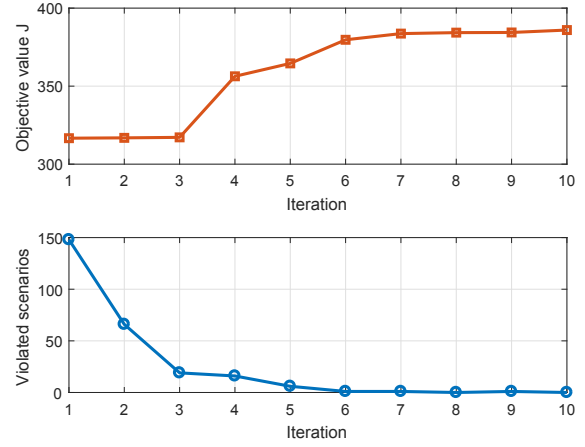


Fig. 4. The objective value and number of constraint violated scenarios on training set case 2.

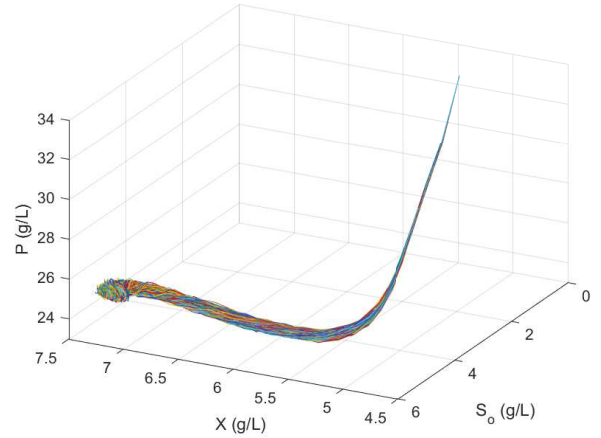


Fig. 5. The state trajectory of 1000 scenarios in case 2 (validation set).

are identified iteratively to form a small-scale optimization, which yields a feasible solution within relatively shorter time than the original full-scenario problem. We test and compare the proposed algorithm with BARON in the CSTR and fermentor controller design. For CSTR, the objective value obtained by our method is only 0.5% worse than the true optimal solution. For the fermentor, our method generates a feasible solution within two hours, whereas BARON cannot find any feasible solution. The designed full-state feedback controllers are evaluated on the validation set, incorporating 1000 scenarios to verify and demonstrate their probabilistic feasibility. Future work will be scenario reduction on the training set to improve the solution and solving time.

REFERENCES

- [1] P. J. Campo and M. Morari, "Robust Model Predictive Control," *Proceedings of American Control Conference*, 1987.
- [2] M. V. Kothare, V. Balakrishnan, and M. Morari, "Robust constrained model predictive control using linear matrix inequalities," *Automatica*, vol. 32, pp. 1361-1379, 1996.
- [3] A. Prékoba, *Stochastic programming*; Netherlands: Kluwer Academic Publishers, 1995.

- [4] P. Li, M. Wendt, and G. Wozny, "Robust model predictive control under chance constraints," *Computers and Chemical Engineering*, vol. 24, pp. 829-834, 2000.
- [5] P. Li, M. Wendt, and G. Wozny, "A probabilistically constrained model predictive controller," *Automatica*, vol. 38, pp. 1171-1176, 2002.
- [6] A. T. Schwarm and M. Nikolaou, "Chance-constrained model predictive control," *AIChE Journal*, vol. 45, pp. 1743-1752, 1999.
- [7] G. Schildbach, P. Goulart and M. Morari, "Linear controller design for chance constrained systems," *Automatica*, vol. 51, pp. 278-284, 2015.
- [8] M. P. Vitus, Z. Zhou, and C. J. Tomlin, "Stochastic control with uncertain parameters via chance constrained control," *IEEE Transactions on Automatic Control*, vol. 61, pp. 2892-2905, 2016.
- [9] G. C. Calafiore and M. C. Campi, "The scenario approach to robust control design," *IEEE Transactions on Automatic Control*, vol. 51, no. 5, pp. 742-753, 2006.
- [10] M. Campi and S. Garatti, "A sampling-and-discarding approach to chance-constrained optimization: Feasibility and optimality," *J. Optimiz. Theory Appl.*, vol. 148, pp. 257-280, 2011.
- [11] G. Calafiore and L. Fagiano, "Robust model predictive control via scenario optimization," *IEEE Trans. Autom. Control*, vol. 58, pp. 219-224, 2013.
- [12] T. Alamo, R. Tempo, A. Luque, D. R. Ramirez, "Randomized methods for design of uncertain systems: Sample complexity and sequential algorithms," *Automatica*, 5, 160-172, 2015.
- [13] P. Petsagkourakis, I. O. Sandoval, E. Bradford, F. Galvanin, D. Zhang and E. A. del Rio-Chanona, "Chance constrained policy optimization for process control and optimization," *Journal of Process Control*, vol. 111, pp. 35-45, 2022.
- [14] E. Tian, Z. Wang, L. Zou and D. Yue, "Chance-constrained H_∞ control for a class of time-varying systems with stochastic nonlinearities: The finite-horizon case," *Automatica*, vol. 107, pp. 296-305, 2019.
- [15] L. Fagiano and M. Khammash, "Nonlinear stochastic model predictive control via regularized polynomial chaos expansions," *Proceedings of IEEE Conference on Decision and Control*, Maui, Hawaii, USA, 142-2728, 2012.
- [16] M. C. Campi, S. Garatti, and F. A. Ramponi, "A general scenario theory for non-convex optimization and decision making," *IEEE Trans. Autom. Control*, vol. 63, pp. 4067-4078, 2018.
- [17] Y. Yang and C. Sutanto, "Chance-constrained optimization for non-convex programs using scenario-based methods," *ISA Transactions*, vol. 90, pp. 157-168, 2019.
- [18] J. Luedtke, "A branch-and-cut decomposition algorithm for solving chance-constrained mathematical programs with finite support," *Mathematical Programming*, 146, 219-244, 2014.
- [19] L. D. Brown, T. T. Cai, and A. DasGupta, "Interval estimation for a binomial proportion," *Statistical Science*, vol. 16, pp. 101-117, 2001.
- [20] N. V. Sahinidis, "BARON: A general purpose global optimization software package," *Journal of Global Optimization*, vol. 8, pp. 201-205, 1996.
- [21] M. Henson and D. Seborg, "Nonlinear control strategies for continuous fermenters," *Proceedings of American Control Conference*, San Diego, CA, USA, 2723-2728, 1990, May.
- [22] Y. Yang and J. M. Lee, "Design of robust control Lyapunov function for non-linear affine systems with uncertainty," *IET Control Theory and Applications*, vol. 6, pp. 2248-2256, 2012.