Korean J. Chem. Eng., 40(7), 1563-1574 (2023)
DOI: 10.1007/s11814-023-1446-7

pISSN: 0256-1115
eISSN: 1975-7220

Sigma-point and stochastic gradient descent approach to solving global self-optimizing controlled variables

Xie Ma*, Hongwei Guan, and Lingjian Ye***,†**

*Ningbo University of Finance & Economics, Ningbo 315175, China
**Zhejiang Business Technology Institute, Ningbo 315012, China
***Huzhou Key Laboratory of Intelligent Sensing and Optimal Control for Industrial Systems, School of Engineering, Huzhou University, Huzhou 313000, China

(*Received 25 April 2022 • Revised 10 January 2023 • Accepted 9 February 2023*)

Abstract−Direct numerical optimization for the global self-optimizing control (gSOC) problem has been recently attempted in the rigorous nonlinear programming (NLP) framework. Compared with the previous *perturbation-based* SOC approaches, the global scheme is of potential to obtain solutions with better performances, as the economics are evaluated via the rigorous nonlinear process model, rather than approximations using the Taylor expansion. The main obstacles for solving the NLP are, however, difficulties for the statistical computations for the cost and constrained variables. In this paper, we firstly introduce the sigma-point approach, which generates less and more efficient sampling points with linear complexity with respect to the uncertain variables, such that the computational load is eased. Furthermore, we incorporate the stochastic gradient descent algorithm to accelerate the search of optimal combination matrix, which can be carried out upon evaluations of only a few, rather than all, sampling points. The scheme, therefore, makes it possible to deal with problems that have high dimensional uncertain parameters and/or when a single evaluation of the cost is time-consuming. A batch reactor and a batch distillation column are investigated to show the usefulness of the presented ideas.

Keywords: Real-time Optimization, Self-optimizing Control, Sigma-point, Stochastic Gradient Descent

INTRODUCTION

Real-time optimization (RTO) is of paramount importance for the modern chemical industry, which aims to restore the process optimality damaged by uncertainties. Among various RTO approaches developed in literature, self-optimizing control (SOC) is a feedback control based strategy, by means of regulatory control of appropriate controlled variables (CVs) [1]. In the past two decades, the CVs have been extended to the form of combinations of the output measurements [2,3], $\mathbf{c}=\mathbf{H}\mathbf{y}$, where $\mathbf{y}$, $\mathbf{H}$ and $\mathbf{c}$ denote the output measurements, combination matrix and the CVs, respectively. Using the measurement combinations can absorb more information than using single measurements as the CVs, thus enhancing the RTO performance. However, solving the optimal combination matrix, $\mathbf{H}$, is challenging, especially when the numbers of measurements and uncertain variables are large.

In the previous studies, several metrics were introduced to simplify seeking the optimal $\mathbf{H}$. In the well-developed local SOC approaches [2,4-7], the economic loss is firstly evaluated through the second-order Taylor expansion of the cost function around the nominal point, and then a linear model is adopted to approximate the input-output relationships. This framework is referred as the *exact local method* [2] and later studied by a number of researchers. Especially, explicit closed-form solutions for the optimal $\mathbf{H}$ have been derived [6,7] based on the exact local method. The local analysis, however, leads to a globally suboptimal $\mathbf{H}$ due to the restrictions of Taylor expansion and linearization. As most chemical plants are inherently nonlinear and often drift away from the nominal point, the RTO performance may degrade using the local SOC approaches.

Later, to enlarge the effective window of SOC, the so-called global SOC (gSOC) approaches have been proposed, aiming to minimize the average economic loss in the entire variation space spanned by the uncertainties, which include the disturbances and measurement noise [8-11]. In [11], the loss function is evaluated via, again, the second-order Taylor expansion; however, at a number of independent disturbance scenarios via Monte Carlo sampling, rather than the single nominal point, then the globally approximated average loss is minimized with respect to $\mathbf{H}$. Zhao et al. [12] proposed a data-based gSOC approach to minimize the power consumption in a chiller plant. Su et al. [13] presented an intelligent gSOC approach to deal with constrained problems. Optimal pressure measurement location for buffering control is handled based on the SOC approach [14]. Real-time optimization compensation method was proposed and used in gold hydrometallurgy processes [15], where the gSOC methodology plays an important role. Normally, the SOC is developed on the basis of process models with parametric uncertainties, then efficient solutions can be promoted using off-line optimization for the CVs. Furthermore, the methodology can be hopefully extended to hybrid models where the first principle and data driven approaches are integrated [16-18].

[†]To whom correspondence should be addressed.
E-mail: lingjian.ye@zjhu.edu.cn

To solve the gSOC problem rigorously, the main obstacle is the computation of the average economic loss for a given **H** [8], which calls for expectation of the loss over nonlinear process models. In most existing SOC approaches, the computations are simplified using the Taylor expansion for the cost function, which can be interpreted as a class of *perturbation-based* schemes. The average economic loss turns out to be analytically derivable in the perturbation framework. However, it is also understandable that such computations are accurate only when the loss is sufficiently small. In many cases, this condition may not be satisfied, for example, when the system is highly nonlinear and/or with limited measurements, then one may end up with biased solutions.

Recently, direct numerical optimization for the optimal **H** has been attempted in the general framework of the gSOC problem [19]. The economic objective function at a single operating scenario, as well as the constrained variables, are computed via the nonlinear process model rather than Taylor approximations as was previously done in perturbation-based schemes. In the presence of stochastic uncertainties, the polynomial chaos expansion (PCE) was used for the computation of reliable function statistics. Furthermore, the sparse grid sampling method has been incorporated to ease the *curse of dimensionality* for constructing the PCEs of the cost function and constraints. It turns out that the new gSOC approach has the potential to find the true optimal global self-optimizing controlled variables. Meanwhile, the active-set change problem, which is challenging in the development of SOC approaches [20-22], can be additionally handled by formulating chance constraints. Using sparse grid sampling, the number of sampling nodes using sparse-grid collocation grows polynomially with the number of uncertain parameters. For large-scale problems, nonetheless, the computations may still be intensive. Hence, the problem of identifying a simple yet accurate gSOC approach is still open.

In this paper, we firstly incorporate the sigma-point approach as an alternative to generating sampling points for the gSOC problem. The sigma-point approach is an efficient sampling technique usually adopted for the unscented transform, where only a few deterministic sigma points are selected to estimate the statistics of nonlinear functions. The number of sampling points grows linearly with the dimension of uncertain stochastic parameters, thus is computationally efficient. The most prevalent application of sigma-point approach is the unscented Kalman filter (UKF) [23], which has achieved tremendous success and been widely applied. The main motivation in this paper is thus the employment of the state-of-art sigma-point approach to speed up solving the global self-optimizing controlled variables. In addition, we further propose a stochastic gradient descent (SGD) approach to solving the gSOC problem. Compared with the sequential solution strategy in [19], the SGD approach is designed such to update the decision variables without evaluations for all, but only a few, sampling points to identify desired updating directions. Therefore, the process of searching an optimum is significantly accelerated, which is especially necessary when the dimension of uncertain variables is large and/or evaluation for a single point is time-consuming.

The rest of this paper is structured as follows. In Section 2, the general formulation in [19] for solving the gSOC problem is outlined, and then the PCE-based solution method is briefly reviewed. Then, we introduce the principle of sigma-point approach in Section 3, which is proposed as an alternative to the PCE-based approach. In Section 4, we propose the SGD solution strategy to solve the gSOC problem, where the Adam algorithm [24] is adopted as a realization of the SGD method. A batch reactor and a batch column example are studied in the subsequent section to illustrate the contributions. Finally, Section 6 concludes this paper.

THE GLOBAL SELF-OPTIMIZING CONTROL PROBLEM

1. Problem Statement

Consider the following static optimization problem for chemical processes:

$$\begin{array}{lll} \min\limits_{u} & J(\mathbf{u}, \mathbf{d}) & \\ \text{s.t.} & \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{d})=0 & \quad (1) \\ & \mathbf{y}=\mathbf{f}_y(\mathbf{x}) & \\ & \mathbf{g}(\mathbf{u}, \mathbf{d})\leq 0 & \end{array}$$

where J is the cost function, $\mathbf{u}\in\Re^{n_u}$, $\mathbf{d}\in\Re^{n_d}$, $\mathbf{x}\in\Re^{n_x}$, and $\mathbf{y}\in\Re^{n_y}$ are the manipulated variables, disturbances, state variables and output measurements, respectively; $\mathbf{f}$: $\Re^{n_x\times n_u\times n_d}\rightarrow\Re^{n_x}$, $\mathbf{f}_y$: $\Re^{n_x}\rightarrow\Re^{n_y}$ and $\mathbf{g}$: $\Re^{n_u\times n_d}\rightarrow\Re^{n_g}$ are the state equations, measurement mapping and operational constraints, respectively. **d** are assumed independent and identically distributed (*i.i.d.*), with probability densities $\mathbf{d}\sim\rho(\mathbf{d})$.

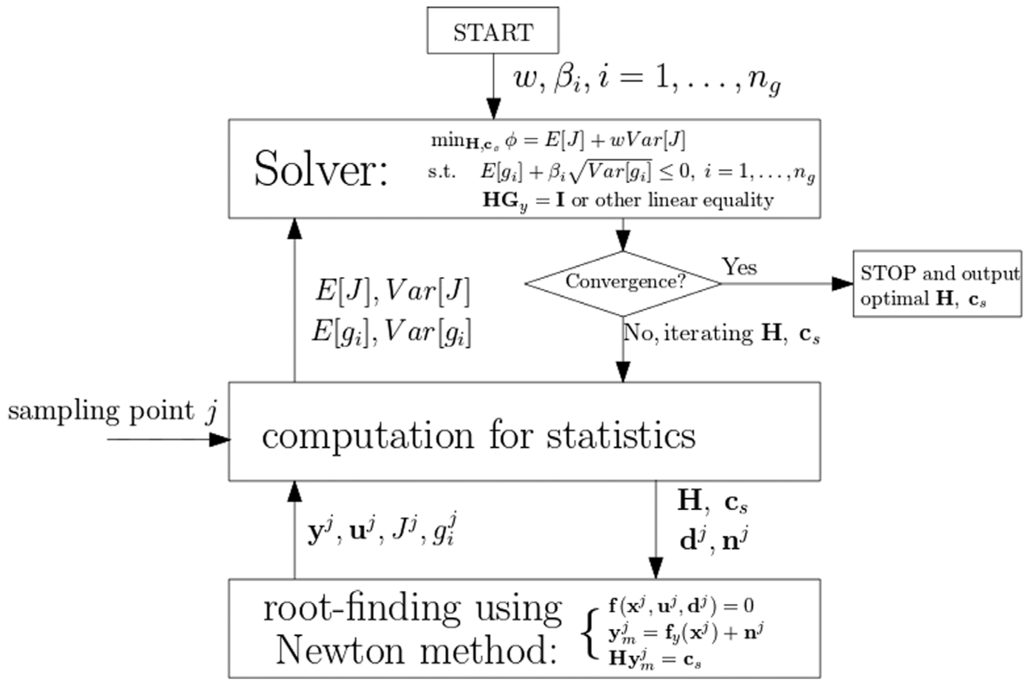
Remark 1 In this paper, we assume *i.i.d.* disturbances such that they are treated independently; however, in many cases it is possible to describe them in a lower dimensional space while their effects are still captured [25-27]. In this case, these methodologies can be applied in prior to simplify the problem.

Self-optimizing control aims to identify linear combinations of measurements as the controlled variables (CVs), **c**=**Hy** ($\mathbf{H}\in\Re^{n_u\times n_y}$ is the combination matrix), such that when c are controlled at constant set-points, say $\mathbf{c}_s$, the economic cost is automatically minimized in the face of disturbances. In the context of global SOC (gSOC), we want to minimize some statistics of the economic cost in the full distribution of stochastic uncertainties, which can be stated as

$$\begin{array}{lll} \min\limits_{H, c_s} & \phi(J) & \\ \text{s.t.} & \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{d})=0 & \quad (2) \\ & \mathbf{y}=\mathbf{f}_y(\mathbf{x}) & \\ & \mathbf{g}(\mathbf{u}, \mathbf{d})\leq 0 & \\ & \mathbf{H}\mathbf{y}_m=\mathbf{c}_s,\ \mathbf{y}_m=\mathbf{y}+\mathbf{n} & \\ & \mathbf{d}\sim\rho(\mathbf{d}),\ \mathbf{n}\sim\rho(\mathbf{n}) & \end{array}$$

where ϕ is a chosen statistic for the economic index J; in the following, we select ϕ: =E[J]+wVar[J], namely, the average loss weighted by its variance; $\mathbf{y}_m\in\Re^{n_y}$ are the noisy measurements corrupted by **n**. It is also assumed that the noise n which is of density $\rho(\mathbf{n})$; The equality constraints, $\mathbf{H}\mathbf{y}_m=\mathbf{c}_s$, represent the effect of controlling the CVs via feedback controllers.

Chance constraints have been incorporated in [19,20] to cover the active-set change problem. It suggests to satisfy g in the probability sense, $\Pr(g_i\leq 0)\geq\alpha_i$, i=1, …, n_g, with $0<\alpha_i\leq 1$ the desired prob-



START
$w, \beta_i, i = 1, \ldots, n_g$
Solver:
$\min_{\mathbf{H},\mathbf{c}_s} \phi = E[J] + wVar[J]$
s.t. $E[g_i] + \beta_i\sqrt{Var[g_i]} \leq 0,\ i = 1, \ldots, n_g$
$\mathbf{HG}_y = \mathbf{I}$ or other linear equality
Convergence?
Yes
STOP and output optimal $\mathbf{H}$, $\mathbf{c}_s$
No, iterating $\mathbf{H}$, $\mathbf{c}_s$
$E[J], Var[J]$
$E[g_i], Var[g_i]$
sampling point j
computation for statistics
$\mathbf{y}^j, \mathbf{u}^j, J^j, g_i^j$
$\mathbf{H}$, $\mathbf{c}_s$
$\mathbf{d}^j, \mathbf{n}^j$
root-finding using Newton method:
$\mathbf{f}(\mathbf{x}^j, \mathbf{u}^j, \mathbf{d}^j) = 0$
$\mathbf{y}_m^j = \mathbf{f}_y(\mathbf{x}^j) + \mathbf{n}^j$
$\mathbf{Hy}_m^j = \mathbf{c}_s$

Fig. 1. The structure of the gSOC formulation.

ability level for each constraint. In practice, the chance constraints are handled by considering

$$E[g_i]+\beta_i\sqrt{Var[g_i]}\leq 0,\ i=1, \ldots, n_g \tag{3}$$

where parameters β_i control the probability levels. The final formulation to be solved therefore reads

$$\begin{aligned} \min_{\mathbf{H}, \mathbf{c}_s} \quad & \phi(J):=E[J]+wVar[J] \\ \text{s.t.} \quad & \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{d})=0 \\ & \mathbf{y}=\mathbf{f}_y(\mathbf{x}) \\ & E[g_i]+\beta_i\sqrt{Var[g_i]}\leq 0,\ i=1, \ldots, n_g \\ & \mathbf{Hy}_m=\mathbf{c}_s,\ \mathbf{y}_m=\mathbf{y}+\mathbf{n} \\ & \mathbf{d}\sim\rho(\mathbf{d}),\ \mathbf{n}\sim\rho(\mathbf{n}) \end{aligned} \tag{4}$$

2. The PCE-based Approach

The structure of nonlinear programming (NLP) formulation (4) is illustrated in Fig. 1. The main difficulty solving (4) is the computation of the statistics for the cost function and constraints (E[J], Var[J], E[g_i], Var[g_i]), for the given decision variables, **H** and $\mathbf{c}_s$, iterated by the numerical optimizer. In the recent work [19], polynomial chaos expansion (PCE) has been proposed as the solution method, which is computationally more efficient and accurate than the common Monte Carlo sampling method for estimating statistics of nonlinear mapping functions. The PCE approach is briefly reviewed as follows.

Let ξ be a random variable defined over a measure space (Ξ, $\mathcal{M}$, $\mathcal{F}$), where Ξ is a nonempty space, $\mathcal{M}$ is the σ-algebra and $\mathcal{F}$ is a probability measure on $\mathcal{M}$. Orthogonal polynomials, $\{\Phi_n(\xi)\}$, for $\xi\in\mathcal{M}$, are defined with their mutual inner products satisfy

$$\langle\Phi_n, \Phi_m\rangle \triangleq \int\Phi_n(\xi)\Phi_m(\xi)d\mathcal{F}(\xi)=\gamma_n\delta_{mn} \tag{5}$$

where δ_{mn} is the Kronecker delta function: δ_{mn}=1 if m=n, and δ_{mn}= 0 otherwise; γ_n is the normalization constant, computed by

$$\gamma_n=\langle\Phi_n^2\rangle=\int\Phi_n^2(\xi)d\mathcal{F}(\xi) \tag{6}$$

Based on the orthogonal relation (5), random variables with different distributions have different basis polynomial functions. For example, Hermite polynomials for Gaussian distribution, Legendre polynomials for uniform distribution, and so on. Common basis orthogonal functions for different distributions are available in the literature.

Let $\psi(\xi)$ be a dependent function with finite variance, then $\psi(\xi)$ can be expanded by PCE as

$$\psi(\xi)=\sum_{i=0}^{\infty}\alpha_i\Phi_i(\xi)=\alpha_0+\alpha_1\Phi_1(\xi)+\alpha_2\Phi_2(\xi)+\alpha_3\Phi_3(\xi)+\cdots \tag{7}$$

where Φ_n and α_n are the n-order PC and its associated coefficient, respectively. The polynomial coefficients are determined by

$$\alpha_n=\frac{\langle\psi, \Phi_n\rangle}{\langle\Phi_n^2\rangle}=\frac{1}{\gamma_n}\int\psi\Phi_n(\xi)d\mathcal{F}(\xi),\ n=0, 1, \ldots \tag{8}$$

For practical implementations, (8) is truncated at a finite order, say N_p, to represent a suitable approximation. This yields

$$\psi(\xi)\approx\hat{\psi}(\xi)=\sum_{i=0}^{N_p}\alpha_i\Phi_i(\xi)=\alpha_0+\alpha_1\Phi_1(\xi)+\ldots+\alpha_{N_p}\Phi_{N_p}(\xi) \tag{9}$$

where $\hat{\psi}$ approximates ψ, and the truncated order N_p is selected per the trade-off between the approximation accuracy and computation cost. Based on the finite order PCE, the statistics of ψ can be analytically computed. The first two order moments are

$$E[\psi]=\alpha_0,\ Var[\psi]=\sum_{i=1}^{N_p}\gamma_i\alpha_i^2 \tag{10}$$

The above presentations are for the univariate case, and extension to the multivariable case is quite straightforward [19]. Therefore, we do not repeatedly present here for the brevity. One notes that the determination of PCE coefficients in (8) involves the computation of integrands. In classical numerical computations, for example the Gaussian quadrature, the number of sampling nodes grows exponentially with respect to the number of uncertain variables. In the context of gSOC problems, since the number of uncertain variables is often large (which contain both the disturbances and measurement noise), it is not realistic to follow the exponential sampling method.

To ease the difficulty, the sparse grid sampling technique is further adopted in [19]. In general, the number of sampling nodes using the sparse-grid collocation grows polynomially in terms of the number of uncertain stochastic variables [28] (of complexity $N \sim (2n_\xi)^k/k!$), which makes it computationally easier to implement in practice than the tensor product collocation.

Sequential solution strategy. An available numerical optimizer, such as the state-of-art IPOPT, is invoked performing iterative search for the optimal H. In each iteration, PCE of both the cost function and constraints are constructed in terms of the random uncertain variables, then the statistics of these quantities will be computed via (10), which requires N (number of sampling points) evaluations of the involved functions. This information is then passed up to the optimizer to determine the next iteration of H and c s. One notes that this procedure is still computationally intensive, which may be intractable for many large-scale and complicated cases.

SIGMA-POINT SAMPLING

The objective of this paper is to further provide a simple and yet relatively accurate gSOC approach, by firstly using the sigma-point sampling as an alternative to the PCE-based approach. The sigma-point sampling method has been widely applied to address various practical problems, typically, in the unscented Kalman filter [23]. the literature, there are a number of variants of sigma-point methods as summarized in [23]. They differ in how the sampling points are selected and their weights to compute the statistics of the unscented transformed nonlinear functions. In the following, the one in [29] is adopted as an application in this paper.

Let $\psi(\xi)$: $\Re^{n_\xi} \rightarrow \Re^1$ be a nonlinear real-valued function, defined over stochastic variables, ξ. The means and covariance of ξ are $\bar{\xi}$ and $\mathbf{P}_\xi$, respectively. The $2n_\xi+1$ sigma points (with associated weights) are defined as

$$\xi_0 = \bar{\xi} \tag{11}$$

$$\xi_i = \bar{\xi} + (\sqrt{(n_\xi+\lambda)\mathbf{P}_\xi})_i,\ i=1, \ldots, n_\xi \tag{12}$$

$$\xi_i = \bar{\xi} - (\sqrt{(n_\xi+\lambda)\mathbf{P}_\xi})_{i-n_\xi},\ i=n_\xi+1, \ldots, 2n_\xi \tag{13}$$

$$W_0^m = \frac{\lambda}{n_\xi+\lambda} \tag{14}$$

$$W_0^c = \frac{\lambda}{n_\xi+\lambda} + (1-a^2+b) \tag{15}$$

$$W_i^m = W_i^c = \frac{1}{2(n_\xi+\lambda)},\ i=1, \ldots, 2n_\xi \tag{16}$$

where $\lambda=(a^2-1)n_\xi$, a is a scaling parameter determining the spread of sigma points around the mean, which is often set as a small value (e.g. 0.001); b can be set based on the prior knowledge of distribution of ξ, b=2 is optimal for Gaussian variables; $(\cdot)_i$ denotes the ith row of a matrix. In the literature, there are different versions of sampling methods and parameter settings; however, in most cases they do not render substantial differences [23].

The mean and covariance of the nonlinear function, ψ, can be computed as the summation of weighted values evaluated at the deterministic sigma points:

$$E[\psi] = \sum_{i=0}^{2n_\xi} W_i^m \psi(\xi_i) \tag{17}$$

$$Var[\psi] = \sum_{i=0}^{2n_\xi} W_i^c (\psi(\xi_i) - E[\psi])^2 \tag{18}$$

The most attractive advantage here is that it requires only $(2n_\xi+1)$ function evaluations, which is obviously of linear complexity against the dimension of uncertainties. This substantially reduces the computation burden, compared to the classical Monte Carlo sampling, as well as the sparse-grid collocation in the PCE approach. On the other hand, it was proven that the sigma-point approach is of third-order accuracy for Gaussian ξ for any nonlinearity and at least to the second order accuracy for non-Gaussian inputs with any nonlinearity [29]. The accuracy can be further improved by the choice of scaling parameters, a and b.

The application of the sigma-point approach to solve the gSOC problem is straightforward. That is, in each iteration in the numerical search of the optimal **H**, $(2n_\xi+1)$ sigma points, distributed over the spans of the disturbances **d** and measurement noise **n**, are selected based on their prior distributions. Then, in each iteration in the numerical optimizer, the economic objective function J and constraints g_i are evaluated accordingly, and the desired E[J], Var[J], E[g_i], Var[g_i] are computed as weighted functions corresponding to the selected $(2n_\xi+1)$ sigma points. These values are fed to the optimizer to determine the next iteration.

The main benefit here is, therefore, the accelerated computation of required statistics by using reduced sampling points (usually a substantial reduction), which makes the overall optimization fast (Problem (4)). In our view, this simple and efficient solution method deserves to be reported within the scope of SOC.

STOCHASTIC GRADIENT DESCENT APPROACH

1. Algorithm Description

In addition to the sampling factor, the sequential solution strategy in [19] is somehow computationally expensive. However, in the following we show that in the special case of w=0 and when there are no active-set changes, the computationally more efficient stochastic gradient descent (SGD) approach can be adopted to solve the global optimal CVs. In this case, the overall objective function is E[J], which is computed as a weighted summation of the costs sampled at independent deterministic points.

Let $\bar{\mathbf{h}}$ be the vector of decision variables, the updating iteration following the gradient descent is given by

$$\bar{\mathbf{h}}_{k+1}=\bar{\mathbf{h}}_k-\eta\nabla \mathrm{J}_k(\bar{\mathbf{h}}_k) \tag{19}$$

where $\nabla \mathrm{J}_k(\bar{\mathbf{h}}_k)$ is the gradient of J with respect to $\bar{\mathbf{h}}_k$ at iteration k, η is referred as the learning rate, which controls the step size of updating $\bar{\mathbf{h}}$. One observes that in SGD, the update of $\bar{\mathbf{h}}$ is performed for every evaluation at a single sampling point, which is computationally cheap to implement.

Regarding the basic form of (19), there are a number of SGD developments in publications to enhance the optimization performance. For example, instead of at every single point, the iteration in the "mini-batch" method is carried out via evaluation of the (average) gradients at more than one point. The algorithms can therefore be smoother and, thus, more robust against randomness. Similar averaging idea can also be imposed on the iterated variables for the same purpose. The learning rate, η, is proposed to be adaptive or implicitly determined to accelerate the optimization [30]. Note that, a locally optimal η would be the second-order sensitivity (Hessian), which is, however, not in general recommended because of the computation complexity.

Remark 2 The case of w=0 with no active-set changes is the most prevalent case in the SOC problem, which covers many industrial scenarios. In this case, the cost function is decomposed into elements associated with separated samples. Otherwise, one needs to estimate the variances of J and/or constraint g_i, whose computations are coupled with other samples, thus cannot be directly handled in the framework of SGD. The potential extension to dealing with active-set changing problems would be transforming the constraints as barrier/penalty terms in the cost function. In this paper, however, we will restrict the contribution to the sigma-point method in the face of such problems.

2. The Adam Algorithm for SGD

In this paper, we employ the Adam algorithm [24] as a special realization of the SGD, which is well-known for its efficiency and has achieved a great success in the field of machine learning. Basically, it was extended from the AdaGrad [31] and RMSProp [32], by absorbing advantages of the two. In general, it is based on adaptive estimates of lower-order moments, which is computationally efficient, has little memory requirement, is invariant to diagonal rescaling of the gradients, and is suitable for large-scale problems.

The implementation pseudo-code of the Adam algorithm is illustrated in Algorithm 1. There are four parameters to specify: η is the aforementioned learning rate, which limits the maximal step size of decision variables; β_1, β_2 control the exponential decay rates of moving averages of the gradients and squared gradients; ε is a small number to avoid division by zero. The last three parameters are often set as $\beta_1=0.9$, $\beta_2=0.999$ and $\varepsilon=10^{-8}$.

Remark 3 The Adam is compatible with the mini-batch SGD approach, simply by using average gradients of more than one sampling points to perform iterations. The size of mini-batch can be adjusted by the designer, to balance the smoothness and convergence speed.

Remark 4 The convergence property of SGD (19) to global or local optimum has been investigated [33,34], that is, the algorithm can be almost surely convergent to the optimum due to the Robbins-Siegmund theorem. Convergence analysis for the Adam-based Algorithm 1 can be found in [24] for more details. Basically, the practical convergence rate is affected by several parameters, for example, the learning rate η and the size of mini-batch m. Their effects will be investigated in the batch column example (Section 5.2).

Algorithm 1. The Adam algorithm [24] for SGD optimization

```
Require: $\eta$: learning rate (stepsize)
Require: $\beta_1, \beta_2 \in [0,1)$: Exponential decay rates for the moment estimates
Require: $\mathrm{J}(\bar{\mathbf{h}})$: cost function for $\bar{\mathbf{h}}$ (vectorized form of $\mathbf{H}$)
1: $\mathrm{m}_0 \leftarrow 0$, $\mathrm{v}_0 \leftarrow 0$, $\mathrm{k} \leftarrow 0$, $\bar{\mathbf{h}}_k \leftarrow \bar{\mathbf{h}}_0$ (initial decision variables)
2: while $\bar{\mathbf{h}}_k$ not converged do
3:     $\mathrm{k} \leftarrow \mathrm{k}+1$
4:     $\mathbf{g}_k \leftarrow \nabla \mathrm{J}_k(\bar{\mathbf{h}}_k)$
5:     $\mathrm{m}_k \leftarrow \beta_1 \mathrm{m}_{t-1}+(1-\beta_1)\mathbf{g}_k$
6:     $\mathrm{v}_k \leftarrow \beta_2 \mathrm{m}_{t-1}+(1-\beta_2)\mathbf{g}_k^2$
7:     $\hat{\mathrm{m}}_k \leftarrow \mathrm{m}_k/(1-\beta_1^k)$
8:     $\hat{\mathrm{v}}_k \leftarrow \mathrm{v}_k/(1-\beta_2^k)$
9:     $\bar{\mathbf{h}}_k \leftarrow \bar{\mathbf{h}}_{k-1}-\eta\hat{\mathrm{m}}_k(\sqrt{\hat{\mathrm{v}}_k}+\varepsilon)$
10: end while
11: return $\bar{\mathbf{h}}_k$
```

3. Derivation of the Gradients

To solve the gSOC problem using the Adam algorithm, the iterated vector h is simply the vectorized form of $\bar{\mathbf{h}}$: =vec([$\mathbf{Hc}_s$]). Then, the core element to implement Adam is the evaluation of the gradient term $\mathrm{J}_k(\bar{\mathbf{h}}_k)$ at a given sampling point, which is derived as follows.

The calculation of the cost J is related to two parts: the system model (including the state and output models) and the feedback control effect. Firstly, the variational form of state model is written as

$$\mathbf{f}_x\partial\mathbf{x}+\mathbf{f}_u\partial\mathbf{u}+\mathbf{f}_d\partial\mathbf{d}=0 \tag{20}$$

where we follow the convention that a_b: $=(\partial a/\partial b)$.

Since we are interested in an independent realization of **d**, the third term in (20) vanishes. Then,

$$\mathbf{f}_x\partial\mathbf{x}+\mathbf{f}_u\partial\mathbf{u}=0 \tag{21}$$

The variational form of the feedback control effect, $\mathbf{Hy}=\mathbf{c}_s$ is given as

$$\begin{aligned}&\mathbf{H}\partial\mathbf{y}+[\mathbf{y}^T]_{diag}\partial\bar{\mathbf{h}}=0 \\ &\rightleftharpoons \mathbf{H}\mathbf{f}_x^y\partial\mathbf{x}+[\mathbf{y}^T]_{diag}\partial\bar{\mathbf{h}}=0\end{aligned} \tag{22}$$

where $[\mathbf{y}^T]_{diag}:=\begin{bmatrix}\mathbf{y}^T & & \\ & \ddots & \\ & & \mathbf{y}^T\end{bmatrix}$ is the block diagonalized matrix composed by vector *y*.

Combining (21) and (22) yields

$$\begin{cases}\mathbf{f}_x\partial\mathbf{x}+\mathbf{f}_u\partial\mathbf{u}+\mathbf{0}_{n_x\times(n_u n_y)}\partial\bar{\mathbf{h}}=0 \\ \mathbf{H}\mathbf{f}_x^y\partial\mathbf{x}+\mathbf{0}_{n_u\times n_u}\partial\mathbf{u}+[\mathbf{y}^T]_{diag}\partial\bar{\mathbf{h}}=0\end{cases}$$

$$\Rightarrow\begin{bmatrix}\partial\mathbf{x}\\ \partial\mathbf{u}\end{bmatrix}=-\begin{bmatrix}\mathbf{f}_x & \mathbf{f}_u\\ \mathbf{H}\mathbf{f}_x^y & \mathbf{0}_{n_u\times n_u}\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{0}_{n_x\times(n_u n_y)}\\ [\mathbf{y}^T]_{diag}\end{bmatrix}\partial\bar{\mathbf{h}} \tag{23}$$

$$=\begin{bmatrix}\times & -\mathbf{f}_x^{-1}\mathbf{f}_u(\mathbf{H}\mathbf{f}_x^y\mathbf{f}_x^{-1}\mathbf{f}_u)^{-1}\\ \times & (\mathbf{H}\mathbf{f}_x^y\mathbf{f}_x^{-1}\mathbf{f}_u)^{-1}\end{bmatrix}\begin{bmatrix}\mathbf{0}_{n_x\times(n_u n_y)}\\ [\mathbf{y}^T]_{diag}\end{bmatrix}\partial\bar{\mathbf{h}}$$

$$=\begin{bmatrix}-\mathbf{f}_x^{-1}\mathbf{f}_u(\mathbf{H}\mathbf{f}_x^y\mathbf{f}_x^{-1}\mathbf{f}_u)^{-1}[\mathbf{y}^T]_{diag}\\ (\mathbf{H}\mathbf{f}_x^y\mathbf{f}_x^{-1}\mathbf{f}_u)^{-1}[\mathbf{y}^T]_{diag}\end{bmatrix}\partial\bar{\mathbf{h}}$$

where we have used the equation of inverse of a partitioned matrix [35], × denote nonzero elements which do not affect the results (because they are corresponding to multiplications by zeros).

Then, the total derivative of the cost due to a change in $\partial\bar{\mathbf{h}}$ is computed as

$$\begin{aligned}dJ&=J_x\partial\mathbf{x}+J_u\partial\mathbf{u}\\&=\underbrace{(-J_x\mathbf{f}_x^{-1}\mathbf{f}_u+J_u)(\mathbf{H}\mathbf{f}_x^y\mathbf{f}_x^{-1}\mathbf{f}_u)^{-1}[\mathbf{y}^T]_{diag}}_{\nabla J(\bar{h})}\,d\bar{\mathbf{h}}\end{aligned}\tag{24}$$

which gives the exact expressions of gradients $\nabla J(\bar{\mathbf{h}})$. The required sensitivities in the above equation for computing $\nabla J(\bar{\mathbf{h}})$ during every iteration can be available via the symbolical framework using automatic differentiation [36]. Based on our experiments and experiences, computation via (24) can often save more than 90% CPU time compared with the finite difference by formulating closed-loop systems.

CASE STUDIES

We next studied two simulated batch chemical examples. The first, a fed-batch reactor, involves an active-set change for the terminal constraint. Therefore, the sigma-point sampling is mainly investigated to explore the reliability of using a smaller size of sampling points. The second, a batch distillation column, has a number of uncertain parameters and output measurements which cannot be handled by the sequential solution method. Both the sigma-point and SGD strategies are explored for this challenging problem. Note that the proposed approach is also applicable to continuous processes, although two batch examples are shown in the following.

1. Batch Reactor

1-1. Process Description

Consider batch-to-batch self-optimizing control of a fed-batch reactor [37], where two reactions occur: A+B→C and 2B→D, A and B are the reactants, and C and D are the product and byproduct, respectively. A is fed once at t=0, while B is fed continuously along the reaction, the feed rate u(t) is the input variable and constrained within $0\le u(t)\le 0.001\ \mathrm{l\cdot min^{-1}}$.

The model equations are as follows:

$$\dot{c}_A=-k_1c_Ac_B-c_A\frac{c_Au}{V},\ c_A(0)=c_{A0}\tag{25}$$

$$\dot{c}_B=-k_1c_Ac_B-2k_2c_B^2-\frac{(c_B-c_B^{in})u}{V},\ c_B(0)=c_{B0}\tag{26}$$

$$\dot{V}=u,\ V(0)=V_0\tag{27}$$

$$\dot{c}_C=k_1c_Ac_B-\frac{c_Cu}{V},\ c_C(0)=c_{C0}\tag{28}$$

$$\dot{c}_D=k_2c_B^2-\frac{c_Du}{V},\ c_D(0)=c_{D0}\tag{29}$$

Table 1. Parameter values for the reactor

Variable	Description	Value
k_1	Kinetic coefficient (main)	$0.053\ l\ \mathrm{mol}\ \mathrm{min}^{-1}$
k_2	Kinetic coefficient (side)	$0.128\ l\ \mathrm{mol}^{-1}\mathrm{min}^{-1}$
c^{in}_B	Inlet concentration of B	$5\ \mathrm{mol}\ l^{-1}$
c_{A0}	Initial concentration (A)	$0.72\ \mathrm{mol}\ l^{-1}$
c_{B0}	Initial concentration (B)	$0.05\ \mathrm{mol}\ l^{-1}$
c_{C0}	Initial concentration (C)	$0\ \mathrm{mol}\ l^{-1}$
c_{D0}	Initial concentration (D)	$0\ \mathrm{mol}\ l^{-1}$
V_0	Initial holdup	$1\ l$
t_f	Batch during time	250 min
ω	Penalty coefficient	$2{,}500\ l^2\ \mathrm{min}^{-1}\ \mathrm{mol}^{-1}$

where c_X and c_{X0} are the concentration of material X in the reactor and its initial value, V is the reactor holdup initially being V_0. And k_1 and k_2 are the kinetic coefficients of two reactions, respectively; c_B^{in} is the concentration of B in the feed. The nominal process variables are listed in Table 1, where three model parameters are uncertain: k_1, k_2 and c_B^{in}, which are Gaussian distributed, with zero means and 25% magnitudes of their nominal values.

The operational objective is to minimize the following cost function:

$$J=-c_C(t_f)V(t_f)+\omega\int_0^{t_f}u(t)^2dt\tag{30}$$

where the first term represents the negative profit of product C, $\omega=2{,}500\ \mathrm{l^2\cdot min^{-1}\cdot mol^{-1}}$ is the penalty weight for material B. Besides the input constraints, the maximum end quality of B and D are not allowed to be greater than $0.025\ \mathrm{mol\cdot l^{-1}}$ and $0.15\ \mathrm{mol\cdot l^{-1}}$, respectively.

The nominal optimal input trajectory using control vector parametrization (CVP) [38] is shown in Fig. 2, where u(t) maintains moderate at the early stage and drops dramatically afterwards. Therefore, an alternative to simplifying the operation is implementing a two-stage strategy, by keeping u(t) constant in the first stage and switches to u(t)=0 in the second [37]. In this case, we have only two decision variables, $u(t)\rightarrow[F_s, t_s]$, where F_s is the constant feed-rate in the first stage and t_s is the switching time to apply u(t)=0. Nominally, the optimal decision variables with re-parametrization are obtained as [0.00040, 224.1]. The added economic cost compared to implementing the full optimal input arc is less than 0.002.

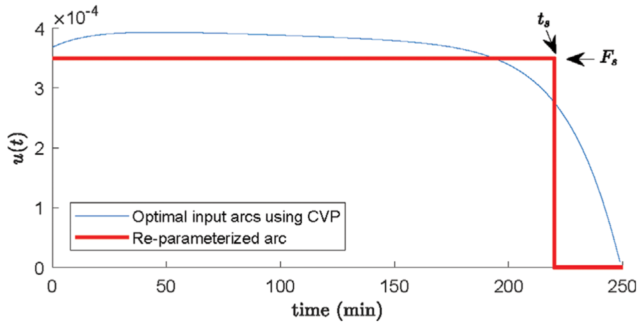


Fig. 2. Optimal input arc and re-parametrization.

With two parameterized degrees of freedom, two controlled variables are required for batch-to-batch self-optimizing control of the reactor. Furthermore, we note that although the constraint $c_B(t_f)\leq 0.025$ is nominally active, it may vary between active and inactive as the disturbances change. The other constraint, $c_D(t_f)\leq 0.15$, remains inactive in the operating region of interest.

The five state variables are measured at the batch end, which are corrupted by Gaussian noise, with 5% magnitudes of their nominal values. Together with the manipulated variables, the full measurement set is

$$\mathbf{y}=[c_{Af}\ c_{Bf}\ V_f\ c_{Cf}\ c_{Df}\ F_s\ t_s]^T \tag{31}$$

where F_s and t_s are also assumed with Gaussian noise to account for implementation errors, with magnitudes of 0.02 ml·min^{-1} and 1 min, respectively.

1-2. Preliminary Tests for the Sigma-point Sampling

First, consider two control schemes: the first is the nominal operation by keeping F_s and t_s constants, the second is the solution applying the local SOC approach by ignoring the constraint of c_{Bf}. The two cases adopt the following CVs:

$$c_1=\begin{bmatrix} F_s \\ t_s \end{bmatrix},\ \text{setpoints:}\begin{bmatrix} 0.00040 \\ 224.1 \end{bmatrix} \tag{32}$$

$$\mathbf{c}_2=\begin{bmatrix} 0.343 & 0.102 & 0.014 & -0.561 & 0.710 & 0.160 & 0.011 \\ 0.115 & 0.109 & 0.002 & -0.143 & 0.014 & 0.018 & 0.039 \end{bmatrix}\mathbf{y},$$

$$\text{setpoints:}\begin{bmatrix} 0.898 \\ 0.949 \end{bmatrix} \tag{33}$$

We compare the computed statistics of J and c_{Bf} using the sigma-point approach against the PCE-based approach, the results are summarized in Table 2. For $\mathbf{c}_1$ and $\mathbf{c}_2$, the number of uncertain variables is 5 and 10, respectively. To construct the PCEs, the results are given for parameters set as k=3 and k=6, n_p=2, where k is the number of collocation points along each uncertain variable, and n_p is the order of PCEs. Both the tensor-product and sparse-grid collocations have been used to sample the uncertainty space. Compared with the tensor-product collocation, the sparse-grid approach is able to greatly reduce the number of sampling points, with almost the identical results. For example, in the case of $\mathbf{c}_1$, it reduces N from 243 and 7,776 to 243 and 993, respectively, for k=3 and k=6. The tensor-product collocation is computationally prohibitive for $\mathbf{c}_2$, due to the exponentially growing N (5.9e4 and 6.05e7), while the sparse-grid collocation requires 201 and 19,485 points to evaluate.

On the other hand, the sigma-point approach requires very few nodes in both cases (11 and 21), which are computationally efficient to implement. In terms of accuracy, define the level of errors as

$$\kappa := \frac{\|\Xi-\Xi_0\|}{\Xi_0}\times 100\% \tag{34}$$

where Ξ and Ξ_0 are the quantity estimated using sigma-point approach and benchmark values, respectively. The benchmark value is taken as the PCE approach (k=6, n_p=2) using sparse-grid collocation (last column in Table 2). Observed from Table 2, the sigma-point approach is quite accurate, especially for means of J and c_{Bf}, where the error levels κ are less than 1%. The worst performance is the standard deviation of c_{Bf}, where the error κ is 7.95% and 11.84% for $\mathbf{c}_1$ and $\mathbf{c}_2$, respectively. However, considering the fact that the number of sampling points in the sigma-approach is less than the PCE approach in several magnitudes, the obtained results are considered to be satisfactory. Note that, in the case of $\mathbf{c}_2$, the ratio of sampling points is about 21 : 19,485≈1 : 1,000.

1-3. Optimization Efficiency

In the following, we test the optimization efficiency using the sigma-point approach. The simulations were carried out in a PC with Intel Core i7 @3.7GHz CPU and 16 GB RAM. For software, the state-of-art IPOPT package was adopted as the numerical optimizer [39], which is invoked by the CasADi toolbox [36] implemented in the MATLAB environment. The batch duration of the reactor was discretized into 31 grids, where 3-point collocation was performed within each grid to obtain polynomial approximations. The dynamic model was transformed into a number of equality constraints, where all state variables were considered as the decision variables. With the collocation method, there were 628 equalities being enforced in every iteration in the NLP (4).

The consumed CPU times for optimizing the controlled variables were compared between the sigma-point and PCE approach. For the PCE approach, the level of accuracy and polynomial order

Table 2. Comparisons of the computed statistics of J and c_{Bf}

		Sigma-point	Error level (κ)	PCE (k=3, n_p=2)		PCE (k=6, n_p=2)	
				Tensor product	Sparse grid	Tensor product	Sparse grid
$\mathbf{c}_1$	$\bar{J}$	−0.22974	0.11%	−0.22949	−0.22949	−0.22948	−0.22948
	σ(J)	0.0619	2.37%	0.0632	0.0632	0.0634	0.0634
	$\bar{c}_B$	0.025672	0.69%	0.02584	0.02584	0.02585	0.02585
	$\sigma(c^B)$ (10^{-3})	7.64	7.95%	8.25	8.25	8.30	8.30
	Number of nodes	11		3^5=243	51	6^5=7776	993
$\mathbf{c}_2$	$\bar{J}$	−0.23171	0.32%	-	−0.23096	-	−0.23096
	σ(J)	0.06246	0.60%	-	0.06289	-	0.06284
	$\bar{c}_B$	0.026012	0.98%	-	0.026242	-	0.026270
	$\sigma(c^B)$ (10^{-3})	9.43	11.84%	-	10.491	-	10.697
	Number of nodes	21		$3^{10}\approx$5.9e4	201	$6^{10}\approx$6.05e7	19485

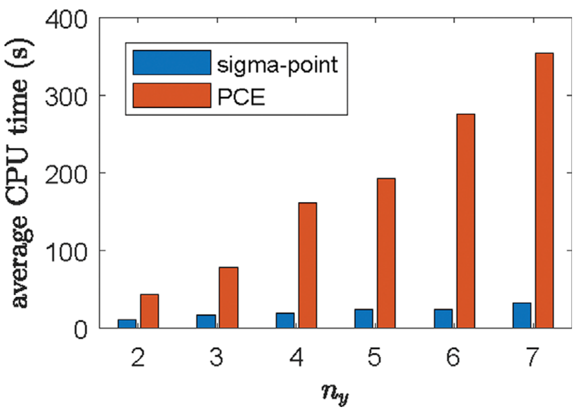


Fig. 3. Average CPU time for solving the gSOC problem.

were selected as k=3 and n_p=2, which is computationally feasible for this reactor example. To compare, we investigated scenarios where the number of measurement ranged from n_y=2 to 7, which have different dimensions of uncertainty. In these cases, F_s and t_s were always included, and new measurements listed in (31) were added one by one as n_y increased. The local SOC approach was first applied and the solutions were fed into the optimizer as the initial values. In addition, parameters w and β were set randomly within the range [0, 50] and [2, 3], respectively. With default IPOPT settings, the average CPU times until convergence of the optimizer were computed over 20 trials. The results are summarized in Fig. 3. It is evident that the sigma-point approach is far more efficient than the PCE approach, especially when the number of measurements is large.

1-4. Closed-loop Validations

Given a measurement subset and specified parameters, w and β, we are able to successfully solve the optimal controlled variables satisfying chance constraints. In the following, we conducted and compared closed-loop performances, using the subset [F_s t_s] and the full measurement set y, respectively. Note that in the first case, it amounts to determining the optimal F_s and t_s, equivalent to the classical robust optimization problem. Therefore, we refer them as the robust operation and gSOC operation, respectively.

Monte Carlo simulations were performed for closed-loop validation, where 10^4 groups of random uncertain scenarios are investigated. Three sets of optimization parameters are as follows:

(1) w=0, β=3;

(2) w=50, β=3;

(3) w=0, β=2.

The distribution of J and c_{Bf} via nonlinear model computations is illustrated in Figs. 4, 5 and 6, respectively. From these results, we have the following general observations:

• Comparing Fig. 4 and Fig. 5, tuning a larger w effectively compensates the variation of J, but at the expense of the average economics.

• Comparing Fig. 4 and Fig. 6, a larger parameter β implies tighter control for the constraint (high probability of satisfaction); however, the economics is sacrificed, and vice versa.

• In general, the SOC operation achieves much better performance than the robust operation, in terms of both economic index and constraint. For example, from Fig. 4(b) and Fig. 6(b), it is clear

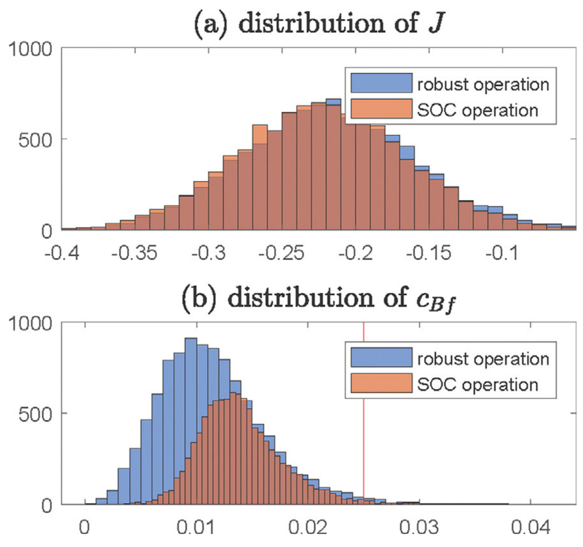


Fig. 4. Monte Carlo simulations of closed-loop validations (w=0, β=3). Robust operation: $\bar{J}$=−0.2181, Pr(c_{Bf}<0.025)=98.90%; SOC operation: $\bar{J}$=−0.2232, Pr(c_{Bf}<0.025)=98.71%.

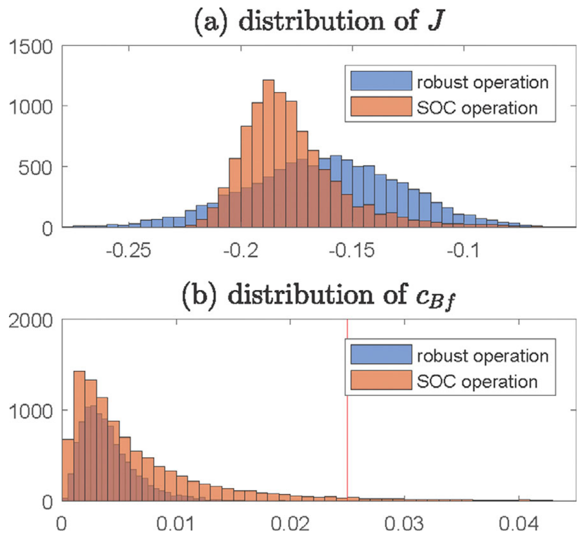


Fig. 5. Monte Carlo simulations of closed-loop validations (w=50, β=3). Robust operation: $\bar{J}$=–0.1626, σ(J)=0.0355, Pr(c_{Bf}< 0.025)=99.97%; SOC operation: $\bar{J}$=−0.1763, σ(J)=0.0272, Pr(c_{Bf}<0.025)=95.86%.

that c_{Bf} can be maintained by the SOC operation closer to the boundary, 0.025, which is helpful to improve the economics. However, Fig. 4(b) shows that the constraint by SOC operation is violated much more frequently than the robust operation. This may be caused by the biased estimation for the variance of c_{Bf}, as investigated in Table 2. In addition, in this case the distributed c_{Bf} shapes are quite different, which means that the same β value corresponds to different probabilities of chance constraints.

2. Batch Distillation Column

2-1. Process Description

Now consider a batch distillation column [40,41], which is operated to separate the light valuable product from the top of column.

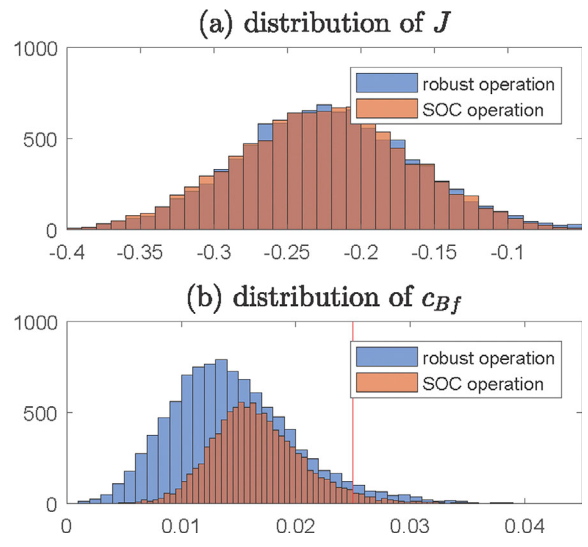


Fig. 6. Monte Carlo simulations of closed-loop validations (w=0, β=2). Robust operation: $\bar{J}$=−0.2236, Pr(c_{Bf}<0.025)=95.27%; SOC operation: $\bar{J}$=−0.2259, Pr(c_{Bf}<0.025)=96.05%.

The process model, composed by material balance, is as follows:

$$\text{Reboiler: } \frac{dM_1}{dt} = -uV \tag{35}$$

$$\frac{dx_1}{dt} = \frac{V}{M_1}(x_1 - y_1 + (1-u)x_2) \tag{36}$$

$$\text{Stages: } \frac{dx_i}{dt} = \frac{V}{M_i}(y_{i-1} - y_i + (1-u)(x_{i+1} - x_i)) \tag{37}$$

$$\text{Condenser: } \frac{dx_c}{dt} = \frac{V}{M_c}(y_p - x_c) \tag{38}$$

with i=2, …, p. p: number of stages; M_i: liquid holdup on stage i (counting from the bottom and stage 1 is the re-boiler); x_i and y_i: molar fraction in liquid and vapor on stage i; x_c: liquid molar fraction in condenser; V: vapor flow; u: distillate ratio u=D/V, 0≤u≤1. With a total condenser, $x_c=y_p$. The vapor-liquid equilibrium relationships on all stages are:

$$y_i = \frac{\alpha x_i}{1 + (\alpha - 1)x_i} \tag{39}$$

where α is the relative volatility. The accumulated distillate, M_d, and its composition, x_d, are calculated as

$$M_d(t) = \int_0^t uV dt = M_1(0) - M_1(t) \tag{40}$$

$$x_d(t) = \frac{\sum_{i=1}^{p} x_i(0)M_i(0) - x_i(t)M_i(t)}{M_1(0) - M_1(t)} \tag{41}$$

The operational objective is to maximize the qualified distillate product M_d for a fixed batch duration [0, t_f], defined as:

$$\begin{aligned} \max_{u(t)} J \quad & = M_d(t_f)x_d(t_f) \\ \text{s.t.} \quad & x_d(t_f) \geq x_d^{des} \\ & 0 \leq u(t) \leq 1 \\ & \text{dynamic process model: (35)-(41)} \end{aligned} \tag{42}$$

Table 3. Parameter values of batch distillation column

Variable	Value	Unit	Variable	Value	Unit
p	10		V	15±2	kmol/h
t_f	10	h	x_d^{des}	0.9	
α	1.5±0.1		$M_1(0)$	100	kmol
M_i	0.2	kmol	$x_i(0)$	0.5±0.1	
M_c	2	kmol	$x_c(0)$	0.5	

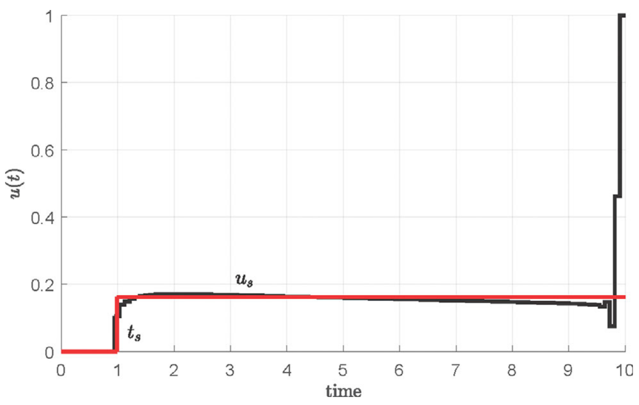


Fig. 7. Optimal input trajectory of the batch column with input parametrization.

where x_d^{des} is the minimal allowable quality. The nominal model parameters are given in Table 3, where α and V are the uncertain disturbances, whose variation ranges are 1.4≤α≤1.6 and 13≤V≤17 kmol/h, respectively. We additionally consider the 11 initial states, $x_i(0)$, are uncertain, with variation ranges 0.4≤$x_i(0)$≤0.6.

The optimal trajectory of u(t) is composed of three sub-arcs, as shown in Fig. 7. The first and the third lie in the lower and upper limits of u(t), respectively. The third arc is very short and can be absorbed into the second [42], which is a sensitivity-seeking arc. We further parametrize the whole trajectory as two decision variables, t_s and u_s, where u(t)=0 for t≤t_s and otherwise u(t)=u_s (a constant level). This input parametrization is a simplified scheme, but captures the main feature of the optimal operation, hence is likely to ease the online implementation of batch-to-batch SOC. Moreover, the terminal quality constraint, $x_d(t_f) \geq x_d^{des}$, turns out to be active in the whole disturbance range. This leaves us with one degree of freedom for SOC. Without loss of generality, t_s is considered for the further analysis, while u_s is assumed for constraint control.

The output measurements are taken as the terminal compositions of light material associated with all stages, **x**: $=[x_i(t_f)]^T$, i=0, …, p, each with random noise within ranges of ±0.02. These measurements are, however, highly correlated due to physical conditions. To maximize the condensed information, the principal component analysis (PCA) is first performed to extract independent variables based on the simulated data. The results suggest that the first three principal components, say **v**: $=[v_1, v_2, v_3]^T$, explain over 99.9% variance of the 11 variables, where the principal components, **v**, are linear transformations of **x**:

$$\mathbf{v} = \mathbf{P}_{3\times 11}\mathbf{x} \tag{43}$$

where $\mathbf{P}_{3\times 11}$ is a 3×11-dimensional loading matrix from the PCA. In addition to t_s, which incorporates some open-loop effect, the CV is composed of the following four variables:

$$c=\mathbf{H}[\mathbf{v}^T\ t_s]^T \tag{44}$$

where **H** is to be solved by the gSOC criterion. The setpoint of **c** can be, without loss of generality, set as 1 to eliminate redundant decision variables.

2-2. Solving the gSOC Problem Using SGD

In this problem, there are 13 independent disturbances and 12 dimensional measurement noise; hence there are 25 uncertain parameters influencing the SOC performance. Even with the sparse grid collocation used in the PCE method, the sampling nodes are about 1300 for merely 3 collocation points in each dimension. However, the sigma-point approach gives only 51 nodes, which is approximately 4% of the sparse grid collocation.

Secondly, we further experiment that simulation of one deterministic scenario of the batch column consumes over 5s CPU time (with the multiple-shooting method, 35 discrete grids over the batch), in the same software and hardware environments as the previous case study. This means that the computation time for evaluation of the average cost function (for a given **H**) needs 2 hours and 4 minutes for the sparse grid based PCE method and sigma-point approach, respectively. Since iterating **H** in an NLP optimizer typically requires hundreds or thousands cycles, it is therefore computationally intractable to perform the previous sequential solution strategy.

In this context, the Adam-based SGD algorithm is applied to circumvent this difficulty. Since there are only 51 sigma points, the algorithm may not necessarily converge for one epoch of these fixed points. In this case, cycles will be repeated with shuffled data to continue optimization of **H**.

Fig. 8 and Fig. 9 show the influences of two main parameters, the learning rate η in (19), and the size of "mini-batch" m, respectively. In all cases, the initial starting point is taken as **H**=[0.1 0.1 0.1 0.5]. In Fig. 8, where η is set as 0.005, 0.01, 0.02 (all with m=4), we observe different convergence rates in terms of η. Basically, a larger η leads to faster convergence for searching the optimal solution, because larger step adaptations are allowed (see subfigure (c) the norm of increments of H); however, it also increases the risk of numerical instability, even the failure of the SGD algorithm. Actually, we have tested that for even larger learning rates, for example η=0.03, or when the starting point of **H** is poorly chosen, the problem could be infeasible due to unreasonable settings, namely, the CV cannot be maintained at the constant setpoint in the operating ranges.

The results in Fig. 9 indicate that using a smaller size of "mini-batch" m accelerates the rate of convergence, due to the more frequent updating of **H** in the same computation load against a larger m; thus, more quickly approaching the optimum. However, it is also expected that in the neighborhood of convergence, a smaller m results in more evident fluctuations due to "less filtered" signals, see the zoomed figure in Fig. 9(c). Anyway, the fluctuations in this case are not very significant; then one is allowed to choose a smaller m for the sake of a quicker convergence.

Closed-loop validations: Lastly, Fig. 10 visualizes the improve-

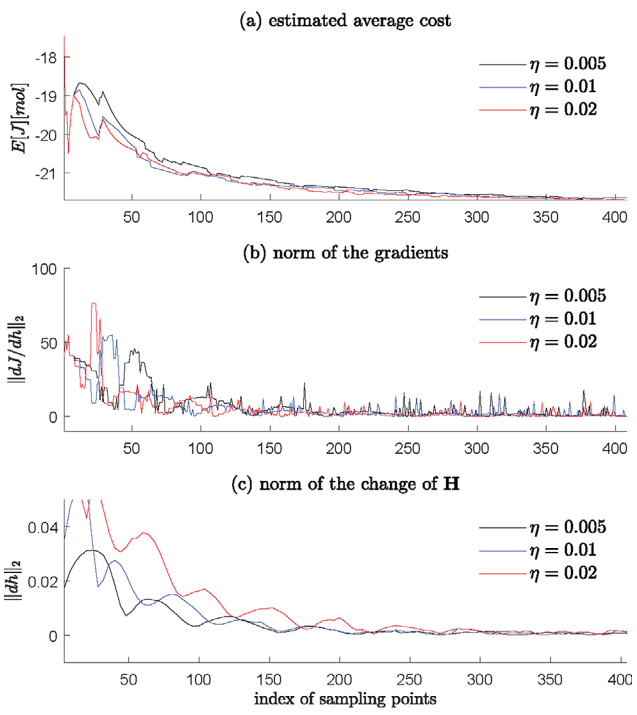


Fig. 8. Adam-based stochastic gradient descent optimization with respect to different learning rates η, (m=4). (a) estimated average cost during training; (b) the norm of gradients; (c) the norm of increments of H.

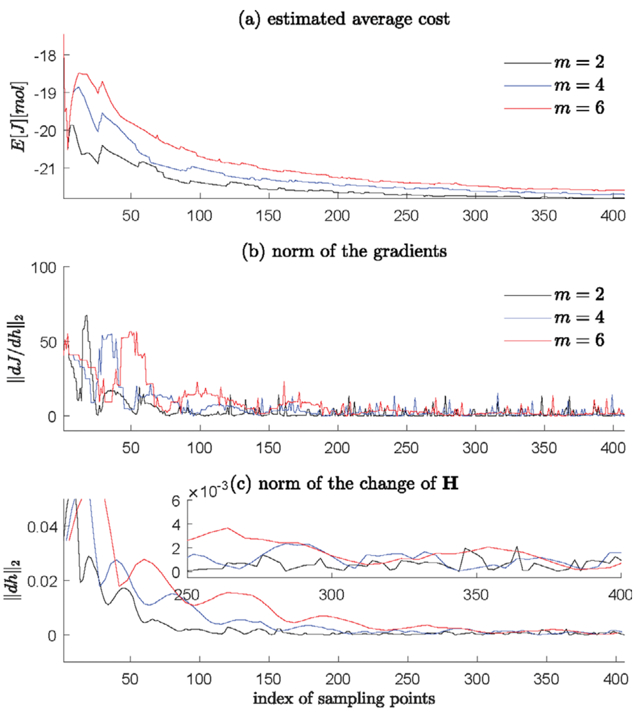


Fig. 9. Adam-based stochastic gradient descent optimization with respect to different size of "mini-batch" m, (η=0.01). (a) estimated average cost during training; (b) the norm of gradients; (c) the norm of increments of H.

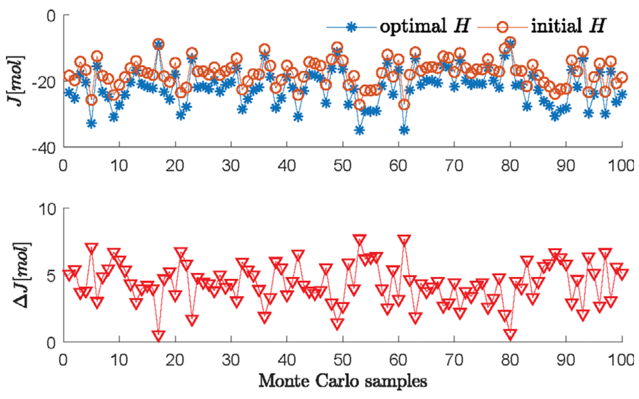


Fig. 10. Improvements of the economic objective function for 100 Monte Carlo points. Upper: comparison of the cost function; lower: their differences (positiveness indicate an improvement).

ments of the objective function evaluated over 100 Monte Carlo sampled points. Although these are sparse in the 23-dimensional space of uncertain parameters, the improvements are evident (averagely, 4.4 mol more collected products), which benefit from the SGD-based optimization.

Remark 5 In both the above two case studies we have considered parametric variations of chosen disturbance variables, then the self-optimizing performances are valid in the specified regions. However, if the sampling is extended to more broader spaces, in terms of either the members of uncertain parameters or their variation magnitudes, then more general self-optimizing performances can be expected as implied by the gSOC methodology. A case study on the plant-wide Tennessee Eastman process was presented in [43], which involves a multi-mode optimal control problem with large disturbance variations.

CONCLUSIONS

We made efforts to accelerate solving the global self-optimizing controlled variables in the NLP framework [19]. Compared with the former PCE based approach, the sigma-point sampling method reduces the number of sampling nodes in the space of uncertain parameters, thus alleviating the computation cost for optimization, while maintaining an acceptable accuracy as studied in the batch reactor example. On the other hand, the Adam-based SGD solution strategy is able to quickly search the (near) optimal H without evaluating all sampling points for the computation of the average cost, which is, however, required in the case of sequential solution strategy in [19], therefore rendering further acceleration.

The SGD algorithm combined with the sigma-point approach was successfully applied to a batch distillation column, which turns out that cannot be easily handled using previous methods. Aided by the two design tools, the application of gSOC can be effectively extended to broader classes of complex chemical processes. An interesting extension of the proposed approach would be developing distributed algorithms, such that large-scale problems can be decomposed into small and computationally tractable ones, which is a future research direction.

ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of China (61673349), Basic Public Welfare research Plan of Zhejiang Province (LGG19F030006) and Huzhou Key Laboratory of Intelligent Sensing and Optimal Control for Industrial Systems (2022-17).

REFERENCES

1. S. Skogestad, *J. Process Control*, **10**, 487 (2000).
2. I. J. Halvorsen, S. Skogestad, J. C. Morud and V. Alstad, *Ind. Eng. Chem. Res.*, **42**, 3273 (2003).
3. J. Jäschke, Y. Cao and V. Kariwala, *Annu. Rev. Control*, **43**, 199 (2017).
4. V. Alstad and S. Skogestad, *Ind. Eng. Chem. Res.*, **46**, 846 (2007).
5. V. Kariwala, *Ind. Eng. Chem. Res.*, **46**, 3629 (2007).
6. V. Kariwala, Y. Cao and S. Janardhanan, *Ind. Eng. Chem. Res.*, **47**, 1150 (2008).
7. V. Alstad, S. Skogestad and E. S. Hori, *J. Process Control*, **19**, 138 (2009).
8. L. Ye and Y. Cao, UKACC International Conference on Control, 136 (2012).
9. L. Ye, Y. Cao, Y. Li and Z. Song, *Ind. Eng. Chem. Res.*, **52**, 798 (2013).
10. L. Ye, Y. Cao, X. Ma and Z. Song, *Ind. Eng. Chem. Res.*, **53**, 14695 (2014).
11. L. Ye, Y. Cao and Y. Xiao, *Ind. Eng. Chem. Res.*, **54**, 12040 (2015).
12. Z. Zhao, Y. Li, T. I. Salsbury and J. M. House, *J. Dyn. Syst., Meas., Control*, **144**, 021008 (2022).
13. H. Su, C. Zhou, Y. Cao, S.-H. Yang and Z. Ji, *Syst. Sci. Control Eng.*, **10**, 65 (2022).
14. A. J. Wiid, J. D. le Roux and I. K. Craig, *Comput.Chem. Eng.*, **145**, 107178 (2021).
15. Y. Liu, Y. Chang, F. Wang, D. Niu and L. Zhao, *Chem. Eng. Res. Des.*, **177**, 136 (2022).
16. P. Shah, M. Z. Sheriff, M. S. F. Bangi, C. Kravaris, J. S.-I. Kwon, C. Botre and J. Hirota, *Chem. Eng. J.*, **441**, 135643 (2022).
17. S. Kurz, H. De Gersem, A. Galetzka, A. Klaedtke, M. Liebsch, D. Loukrezis, S. Russenschuck and M. Schmidt, *J. Math. Ind.*, **12**, 1 (2022).
18. D. Lee, A. Jayaraman and J. S. Kwon, PLoS *Comput. Biol.*, **16**, e1008472 (2020).
19. L. Ye, Y. Cao and S. Yang, *Comput. Chem. Eng.*, **159**, 107662 (2022).
20. L. Ye, Y. Cao and S. Skogestad, The 20th World Congress of the International Federation of Automatic Control, Toulouse, France (2017).
21. H. Manum and S. Skogestad, *J. Process Control*, **22**, 873 (2012).
22. D. Krishnamoorthy and S. Skogestad, *Ind. Eng. Chem. Res.*, **58**, 13555 (2019).
23. H. M. Menegaz, J. Y. Ishihara, G. A. Borges and A. N. Vargas, *IEEE Trans. Autom. Control*, **60**, 2583 (2015).
24. D. P. Kingma and J. Ba, arXiv preprint arXiv:1412.6980 (2014).
25. J. Hasenauer, S. Waldherr, M. Doszczak, N. Radde, P. Scheurich and F. Allgöwer, *BMC Bioinformatics*, **12**, 1 (2011).
26. D. Lee, A. Jayaraman and J. S.-I. Kwon, *AIChE J.*, **66**, e16925 (2020).
27. A. Narasingam, P. Siddhamshetty and J. S.-I. Kwon, *Ind. Eng. Chem. Res.*, **57**, 3977 (2018).

28. D. Xiu, *Numerical methods for stochastic computations: A spectral method approach*, Princeton University Press (2010).
29. E. A. Wan and R. Van Der Merwe, *Proceedings of the IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium*, 153 (2000).
30. P. Toulis and E. M. Airoldi, *The Ann. Statistics*, **45**, 1694 (2017).
31. J. Duchi, E. Hazan and Y. Singer, *J. Machine Learning Res.*, **12**, 2121 (2011).
32. T. Tieleman and G. Hinton, RMSProp, COURSERA: Neural Networks for Machine Learning, Technical report (2012).
33. D. Saad, *Online Learning*, **5**, 6 (1998).
34. K. C. Kiwiel, *Math. Programming*, **90**, 1 (2001).
35. H. Lutkepohl, Handbook of Matrices, John Wiley & Sons, New York (1997).
36. J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings and M. Diehl, *Mat. Programming Comput.*, **11**, 1 (2019).
37. B. Chachuat, B. Srinivasan and D. Bonvin, *Comput. Chem. Eng.*, **33**, 1557 (2009).
38. B. Srinivasan, S. Palanki and D. Bonvin, *Comput. Chem. Eng.*, **27**, 1 (2003).
39. L. T. Biegler and V. M. Zavala, *Comput. Chem. Eng.*, **33**, 575 (2009).
40. L. Ye and S. Skogestad, *Comput. Chem. Eng.*, **117**, 451 (2018).
41. L. Ye, F. Shen and H. Guan, *J. Process Control*, **113**, 1 (2022).
42. C. Welz, B. Srinivasan and D. Bonvin, Barcelona, Spain, 1586 (2004).
43. L. Ye, Y. Cao, X. Yuan and Z. Song, *IEEE Trans. Ind. Electron.*, **64**, 4662 (2017).