Design Space Description through Adaptive Sampling and Symbolic Computation

Fei Zhao, Ignacio E. Grossmann*

Center for Advanced Process Decision-Making, Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213

Salvador García Muñoz, Stephen D. Stamatis

Synthetic Molecule Design and Development, Lilly Research Laboratories, Indianapolis, IN 46285

Abstract

Design space definition is one of the key parts in pharmaceutical research and development. In this article, we propose a novel solution strategy to explicitly describe the design space without recourse decisions. First, to smooth the boundary, the Kreisselmeier-Steinhauser (KS) function is applied to aggregate all inequality constraints. Next, for creating a surrogate polynomial model of the KS function, we focus on finding sampling points on the boundary of KS space. After performing Latin hypercube sampling (LHS), two methods are presented to efficiently expand the boundary points, i.e., line projection to the boundary through any two feasible LHS points and perturbation around the adaptive sampling points. Finally, a symbolic computation method, cylindrical algebraic decomposition, is applied to transform the surrogate model into a series of explicit and triangular subsystems, which can be converted to describe the KS space. Two case studies show the efficiency of the proposed algorithm.

Keywords: process design; design space; adaptive sampling; symbolic computation; cylindrical algebraic decomposition

*Correspondence concerning this article should be addressed to grossmann@cmu.edu.

1. Introduction

In the pharmaceutical industry, process parameters[1] correspond to degrees of freedom or variables that can be manipulated in the operation of a manufacturing process, and which can be measured and set within the controller tolerance for a desired value. Design space is defined as "multidimensional combination and interaction of input variables and process parameters that have been demonstrated to provide assurance of quality"[2]. In other words, Product quality is maintained as long as the process parameters are controlled within the design space.

Early approaches to identify the design space were solely based on experiments and empirical functions[3]. By performing extensive experiments, the relationships of process parameters and critical quality attributes (CQAs) can be built through regression and the process parameters that have medium/high impacts on the CQAs can be determined. The design space is visualized by response surface modeling and further verified by additional experiments[4]. This method requires extensive experiments, and it is very time-consuming and expensive. To lower the cost of developing design spaces, mechanistic models that contain relationships of process parameters and CQAs can be formulated in advance and parametrized with less data. Goyal and Ierapetritou[5] proposed an approach based on outer-approximation to identify the operating envelopes where process operation is feasible, safe, and profitable. In addition, in order to address the computationally expensive models, the surrogate-based methods are then proposed. Rogers and Ierapetritou[6,7] applied Kriging as the surrogate models to approximate the original functions and identify the design spaces with limited samplings. Compared with the kriging surrogate models, Wang and Ierapetritou[8] used RBF surrogate models. Metta et al.[9] proposed to use an artificial neural network to create the surrogate models for addressing problems that are computationally expensive or do not have constraints in closed form.

Moreover, optimization approaches based on mechanistic models have been extensively studied to describe the design space[10,11]. Characterizing a design space for a process design

model is analogous to the flexibility index problem in the chemical industry[12,13,14]. The flexibility index is used to describe an operational range, which represents a maximum scaled departure of all process parameters from the given nominal conditions. It is worth stating that "design space" and "feasible space" are interchangeably used in the pharmaceutical industry. Generally, it is not easy to accurately describe the boundary of the real design space because of the high nonlinearity. The flexibility index can approximate the design space by a largest inscribed subspace with a specific shape, which may be a rectangle, ellipse, or other shapes. Because we are only concerned with this subspace, for simplicity, this subspace is denoted as "design space" in this paper. When approximating it as a rectangle, the vertex direction search method[13] can be employed to find the flexibility index, which is rigorous for convex regions. To avoid the convexity assumption, Grossmann and Floudas[15] developed an active constraint strategy, where the two-level optimization formulation for the flexibility index problem can be reformulated as a mixed-integer linear or nonlinear programming model by applying the KKT conditions to the inner optimization problem. In addition, Pulsipher and Zavala[16] proposed to use an ellipsoidal set to capture correlations of process parameters, as well as a mixed-integer conic programming formulation to compute the flexibility index. A number of approaches are proposed to quantify system flexibility, and an extensive review is provided by Grossmann et al. [17] If the nominal conditions of the process parameters are unknown, the flexibility index problem can be extended to the design centering problem[18], which focuses on determining the nominal conditions that maximize the size of the design space. From a mathematical view, the design centering problem is a generalized semi-infinite programming problem[19,20]. Flexibility index and design centering are two complementary ways for estimating a candidate design space, which have been widely studied in recent decades; however, both methods need to specify the shape of design space in advance, which is quite hard to reflect the reality of the feasible region, especially for nonconvex cases.

In the absence of models capable of efficiently probing the fullest extent of the feasible region, design spaces were obligated to be defined with an assumption of shape to be probed experimentally. With a robust model, there is no reason to make the shape assumption because we can effectively probe the boundaries to the extent of the feasible region. The challenge is thus computing and defining that boundary. Zhao and Chen[21] first proposed representing the design model as an existential quantifier formula and applied the cylindrical algebraic decomposition (CAD) method[22] to accurately describe the design space and explicitly express the relationships between uncertain parameters. The CAD method can provide a complete description of the design space (in this case, the design space is identical to the feasible region), and the triangular structure makes possible the explicit algebraic representation of the bounds of each process parameter. The method is suitable for convex and nonconvex systems described by polynomials. Zhao et al.[23] proposed a space projection method based on the CAD method to deal with flexibility index problems. Due to the heavy computational burden of the CAD method, the above methods are only applicable to relatively small-scale problems. For high-dimensional systems consisting of a large number of equalities and limited inequalities, Zheng et al.[24] proposed to build a surrogate model to correlate the inequality constraints based on an initial sample set. The design space is explicitly expressed via the CAD method, and the boundary can be checked to iteratively refine the CAD results. However, if the design space has a severely irregular shape, the computational burden of the CAD method will be very high. Moreover, since it is not appropriate to reduce process parameters, the number of inequality constraints becomes another key factor of the computational complexity.

In this work, we propose a novel design space description method based on efficient adaptive sampling and symbolic computation, and in which no recourse is considered for the realization of the parameters. The proposed method not only can eliminate all the equality constraints and state variables, but also significantly reduce the sampling burden, and decrease the number of

inequality constraints to one. In addition, there is a tradeoff between the complexity of the CAD method and the accuracy of design space description. The rest of this article is organized as follows. Sections 2 provides problem statements including the research framework. Section 3 defines the design space based on the Kreisselmeier-Steinhauser (KS) function. Section 4 illustrates the main steps of the CAD method. Section 5 provides an adaptive sampling strategy to find the explicit expression of the design space. Two case studies are provided in Section 6 to illustrate the proposed methods. Section 7 discusses the key characteristics of the proposed method. Section 8 concludes the paper.

2. Problem statement

An ultimate goal of the design space definition problem is to accurately and explicitly describe the design space, regardless of whether it is a convex and nonconvex space. For a given design model with no recourse[12], the feasibility of the model can be described as:

$$\forall \boldsymbol{\theta} \in DS(\boldsymbol{\theta})\{\forall j \in J[g_j(\boldsymbol{\theta},\boldsymbol{x}) \leq 0], \forall i \in I[h_i(\boldsymbol{\theta},\boldsymbol{x}) = 0]\} \quad (1)$$

where $\boldsymbol{\theta}$ and $\boldsymbol{x}$ are process parameters and state variables, respectively. Equation (1) states that for any possible realization of process parameters in the design space, denoted as $DS(\boldsymbol{\theta})$, all the individual constraints should be satisfied. In other words, $DS(\boldsymbol{\theta})$ can also be defined as

$$DS(\boldsymbol{\theta}) := \left\{ \boldsymbol{\theta} \in \mathbb{R}^P \middle| \begin{bmatrix} g_j(\boldsymbol{\theta},\boldsymbol{x}) \leq 0, & \forall j \in J \wedge \\ h_i(\boldsymbol{\theta},\boldsymbol{x}) = 0, & \forall i \in I \wedge \\ \boldsymbol{\theta}^L \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}^U & \end{bmatrix} \right\} \quad (2)$$

$DS(\boldsymbol{\theta})$ represents the entire feasible region of process parameters. It is generally difficult to describe it analytically because of the nonlinearities of the design model. Based on previous work[21], the CAD method can equivalently transform an inequality system to a triangular system, and the upper and lower bounds of each process parameter can be expressed explicitly. For high-dimensional cases, the equations and state variables can be eliminated through surrogate models for the inequality constraints[24], which can reduce the computational burden of the CAD method. Since it needs to sample points over the whole design space, this method must sample

enough points and take many iterations to accurately capture the profile of the design space.

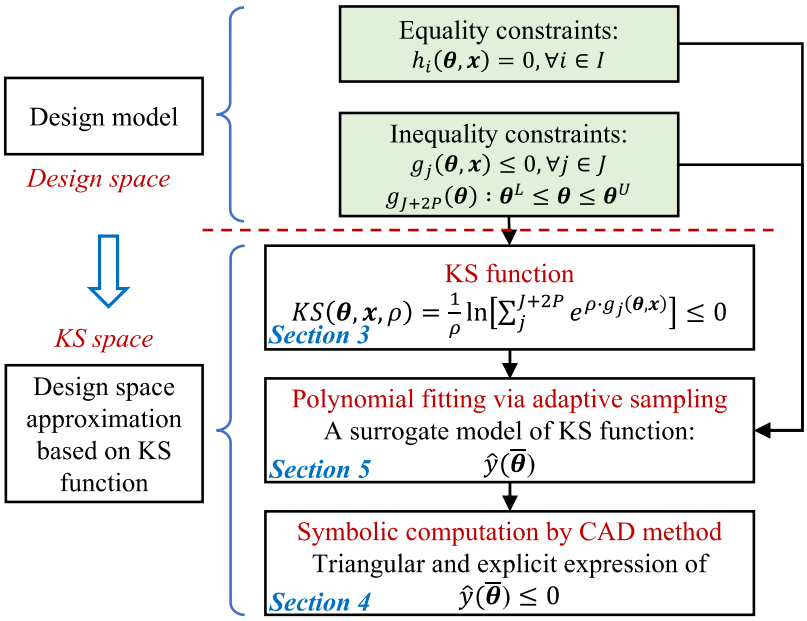


Figure 1. Conceptual computational framework.

In order to further ease the computational burden of the CAD method and improve the efficiency of adaptive sampling and surrogate modeling, a novel solution strategy is proposed in this work to explicitly describe the design space. The research framework is shown in Figure 1, which can be interpreted as follows.

(1) For a given design model, the KS function can aggregate all inequality constraints, and then an underestimate of the desired design space, denoted as *KS space*, can be described by a single inequality constraint. The boundary of the KS space is continuously differentiable.

(2) The KS function is a transcendental function. To be able to process the KS function with the CAD method, an adaptive sampling strategy is proposed to create a polynomial surrogate model of the KS function. Two methods are presented to efficiently expand the boundary points of the KS space. i.e., line projection points through any two feasible LHS points, and perturbation boundary points around the adaptive sampling points. The cross-validation method is applied to evaluate the stopping criteria of the adaptive sampling.

(3) The CAD method is applied to triangulate the polynomial surrogate model. A simple

checking rule is applied to evaluate the feasibility of the CAD result.

Based on the above conceptual computational framework, we can point out two motivations:

(1) The complexity of the original CAD method[25] is formulated as

$$O\left(d_1{}^{2^{2n+8}} m^{2^{n+6}}\right) \tag{3}$$

which indicates that the complexity grows doubly exponentially with the number of variables n; d_1 is the maximum degree in any one variable in the original model; m is the number of polynomials. After applying the KS function to aggregate all inequality constraints and create its polynomial surrogate model, the complexity can be reduced to Equation (4),

$$O\left(d_2{}^{2^{2n+8}}\right) \tag{4}$$

where m has been reduced to one; d_2 is the maximum degree in any one variable in the surrogate model. While still doubly exponential in the number of variables, if the fitted degree, d_2, is not too large, the complexity is acceptable; thus, there is a tradeoff between the complexity of the CAD method and the accuracy of design space description.

(2) According to Equation (2), the equalities are used to represent the process model, and the state variables have the same dimension as equality constraints, i.e., $\dim(\boldsymbol{x}) = |I|$. Once the value of $\boldsymbol{\theta}$ is specified, we can run the simulation and obtain the results of $\boldsymbol{x}$. The inequality constraints define the quality requirements of the process design problems. If we sample the points in the space of $\boldsymbol{\theta}$, the results of $\boldsymbol{x}$ in the equalities can be used to evaluate the inequality constraints. Thus, in this work, all the equality constraints and state variables can be eliminated when generating the surrogate model of the KS function. Moreover, since the KS space must be contained within the original design space, we only need to focus on finding the points on the boundary of the KS space. The intention is to locate as many boundary points as possible, which can significantly reduce the sampling burden.

In summary, approximating the design space by the KS space, the proposed method can ease the sampling burden of surrogate modeling, reduce the computational expense of the CAD

method, and the assumptions for the shape of design space are not required.

3. Design space approximation through KS function

The KS function was first proposed by G. Kreisselmeier and R. Steinhauser[26], which was initially presented for controller design. In the last two decades, the KS function has been widely used in constraint aggregation methods for gradient-based optimization, e.g., chemical process design[27], as well as problems involving local stress constraints, e.g., aircraft design[28,29]. The KS function shown in Equation (5) can aggregate a set of inequality constraints into a single function, and it only contains one parameter ρ.

$$KS(\boldsymbol{\theta},\rho) = \frac{1}{\rho}\ln\left[\sum_{j}^{J} e^{\rho \cdot g_j(\boldsymbol{\theta})}\right] \leq 0 \tag{5}$$

where $g_j(\boldsymbol{\theta}) \leq 0$ are inequality constraints, $j \in J$. The KS function produces an envelope surface that is continuous and represents a conservative estimate of the feasible region for a set of constraints. $\rho > 0$ is an aggregation parameter defined by the user, which can control how close the envelope is to the original constraints. In the following nonlinear and nonconvex example, the KS function as a constraint aggregation method, and the effect of increasing ρ for inequality constraints, can be visualized.

An illustrative example

Consider the inequalities,

$$\begin{aligned} &g_1{:}(\theta_2-2)^2+(\theta_1-2)^3+(\theta_2-2)(\theta_1-2)-0.5\leq 0\\ &g_2{:}(\theta_2-2)^2+(\theta_1-2)^2-2\leq 0\\ &g_3{:}\theta_1-4\leq 0\\ &g_4{:}-\theta_1\leq 0\\ &g_5{:}\theta_2-4\leq 0\\ &g_6{:}-\theta_2\leq 0 \end{aligned} \tag{6}$$

For the design space definition problem, θ_1 and θ_2 are regarded as the process parameters. The feasible region of these inequality constraints, which can be denoted as a complete design space, is depicted by the yellow region shown in Figure 2(a). According to Equation (5), the KS

function of $g_1,\ldots,g_6$ is formulated as

$$KS(\boldsymbol{\theta},\rho) = \frac{1}{\rho}\ln\left[e^{\rho \cdot g_1} + e^{\rho \cdot g_2} + e^{\rho \cdot g_3} + e^{\rho \cdot g_4} + e^{\rho \cdot g_5} + e^{\rho \cdot g_6}\right] \tag{7}$$

which generates an underestimate profile of the design space after fixing ρ. As ρ increases, the profile depicted by the KS function can approach in the limit the true profile of the feasible region. As shown in Figure 2(b), when $\rho = 5$, the profile of the KS function is continuous and almost coincides with the real boundary of the feasible region. However, the profile for $\rho = 2$ is smoother, although less accurate.

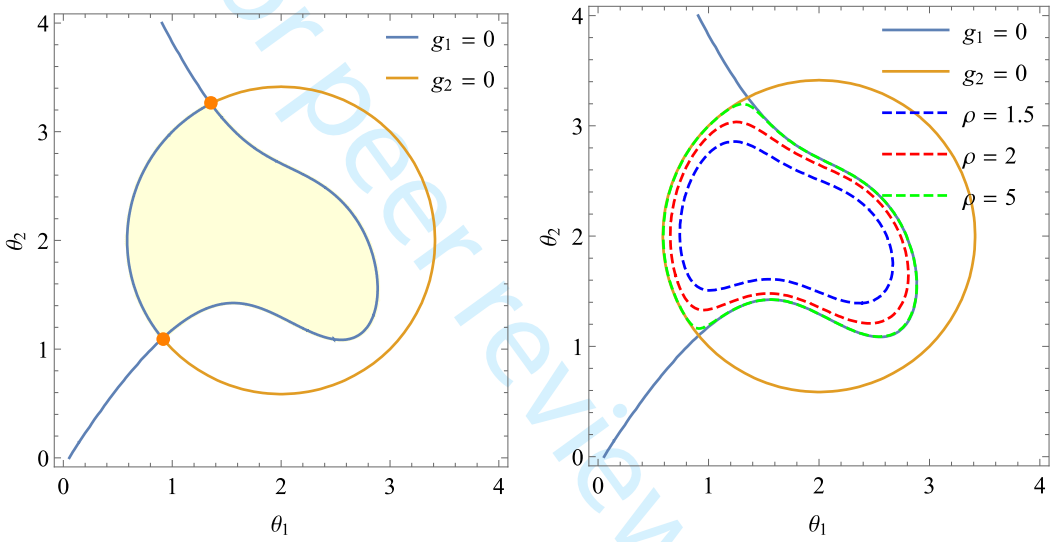


Figure 2. Feasible region of the example and profiles of KS function with increasing ρ.

The KS function can be applied to approximate the design space. Hence, the problem of approximately describing a design space can be transformed into describing the corresponding KS space. The major motivations for using the KS function can be stated as follows:

(1) The KS function can aggregate multiple inequality constraints, and the space can be depicted by a single inequality constraint. Since general constraints can be handled, the KS function is also applicable to non-convex design spaces.

(2) The KS function only involves one parameter $\rho > 0$, which is a scaling factor of the space.

As ρ increases, the KS function can provide a closer approximation to the design space, i.e.,

$$\{\boldsymbol{\theta} \in \mathbb{R}^n | KS(\boldsymbol{\theta},\rho_1) \leq 0\} \subset \{\boldsymbol{\theta} \in \mathbb{R}^n | KS(\boldsymbol{\theta},\rho_2) \leq 0\}, \quad 0 < \rho_1 < \rho_2 \tag{8}$$

In the limit, when ρ tends to infinity, the KS function can exactly represent the design space, i.e.,

$$\left\{\boldsymbol{\theta} \in \mathbb{R}^n | \lim_{\rho \to \infty} KS(\boldsymbol{\theta},\rho) \leq 0\right\} = \{\boldsymbol{\theta} \in \mathbb{R}^n | \forall j \in J[g_j(\boldsymbol{\theta}) \leq 0]\} \tag{9}$$

(3) The KS function is continuously differentiable. The KS function can smooth the profile and reduce most irregular regions, e.g., removing the intersection points in the original design space. As shown in Figure 2(a), the design space contains two orange intersection points. However, the smooth profiles of the KS function in Figure 2(b) have no intersection points. For $\rho_2 = 5$ and $\rho_1 = 2$, the profile of $KS(\boldsymbol{\theta},\rho_1)$ is smoother than the profile of $KS(\boldsymbol{\theta},\rho_2)$.

According to the KS function, the design model in Equation (2) can be reformulated as follows,

$$\begin{aligned} &KS(\boldsymbol{\theta},\boldsymbol{x},\rho) = \frac{1}{\rho}\ln\left[\sum_j^J e^{\rho \cdot g_j(\boldsymbol{\theta},\boldsymbol{x})} + \sum_p^P \left(e^{\rho \cdot (\theta_p^L - \theta_p)} + e^{\rho \cdot (\theta_p - \theta_p^U)}\right)\right] \leq 0 \\ &h_i(\boldsymbol{\theta},\boldsymbol{x}) = 0, \quad \forall i \in I \end{aligned} \tag{10}$$

where $KS(\boldsymbol{\theta},\boldsymbol{x}) \leq 0$ represents the KS space for a fixed value of ρ.

4. Explicit expression of design space through symbolic computation

Generally, the design space is a bounded and closed space. In previous work[21], if a design space is formulated by a polynomial system, the CAD method can transform this polynomial system into a series of triangular subsystems, where each subsystem corresponds to a subspace. All the subspaces define the entire design space. For the design model shown in Equation (2), the formulation of each triangular subsystem is as follows.

$$\begin{gathered} k_1 \le \theta_1 \le q_1 \\ k_2(\theta_1) \le \theta_2 \le q_2(\theta_1) \\ \vdots \\ k_P(\theta_1,\ldots,\theta_{P-1}) \le \theta_P \le q_P(\theta_1,\ldots,\theta_{P-1}) \\ k_{P+1}(\theta_1,\ldots,\theta_{P-1},\theta_P) \le x_1 \le q_{P+1}(\theta_1,\ldots,\theta_{P-1},\theta_P) \\ \vdots \\ k_{P+N}(\theta_1,\ldots,\theta_P,x_1,\ldots,x_{N-1}) \le x_N \le q_{P+N}(\theta_1,\ldots,\theta_P,x_1,\ldots,x_{N-1}) \end{gathered} \tag{11}$$

All the lower and upper bounds of the process parameters are explicit expressions. For a given $(\theta_1,\ldots,\theta_p)$, the lower and upper bound of θ_{p+1}, i.e., $k_{p+1}(\theta_1,\ldots,\theta_p)$ and $q_{p+1}(\theta_1,\ldots,\theta_p)$, become constants.

The CAD method mainly contains two stages: projection and lifting. In the projection stage, the key point is to calculate discriminants and resultants[30] of polynomials, which can find the tangency and intersection points in each dimension. In this way, the space in each dimension can be decomposed by using these points. For the example in Equation (6), as shown in Figure 3(a), if the triangular structure is set as $\theta_1 \prec \theta_2$, the green tangency points and orange intersection points can be calculated by

$$\begin{aligned} &\text{discriminant}(g_1,\theta_2) = 38 - 52\theta_1 + 25\theta_1^2 - 4\theta_1^3, \\ &\text{discriminant}(g_2,\theta_2) = -4(2 - 4\theta_1 + \theta_1^2), \\ &\text{resultant}(g_1,g_2,\theta_2) = 0.25 * (473 - 1440\theta_1 + 1700\theta_1^2 - 1012\theta_1^3 + \\ &\qquad 328\theta_1^4 - 56\theta_1^5 + 4\theta_1^6) \end{aligned} \tag{12}$$

Equation (12) are univariate polynomials of θ_1, so that θ_2 is eliminated. By performing the real root isolation algorithm[31], five distinct real roots of θ_1 for these polynomials can be obtained.

$$0.585786 < 0.910159 < 1.35389 < 2.88639 < 3.41421$$

Figure 3(a) illustrates that the tangency and intersection points can be projected onto the θ_1-axis, i.e., A_1: (0.585786, 0), A_2: (0.910159, 0), A_3: (1.35389, 0), A_4: (2.88639, 0), A_5: (3.41421, 0). The boundary between any two adjacent projection points is continuously differentiable. Based on these projection points, the entire two-dimensional (2D) space can be decomposed into six cylindrical 2D subspaces, i.e., [0, A_1], [A_1, A_2], [A_2, A_3], [A_3, A_4], [A_4, A_5], [A_5, 4]. In the lifting stage, these cylindrical 2D subspaces should be checked successively. Taking the

subspace on [A_1, A_2] as an example, in Figure 3(b), the steps of the lifting stage are as follows.

(1) C_1 is the midpoint of A_1 and A_2. After substituting C_1, g_1 and g_2 can be converted to

$$\begin{matrix}\overline{g}_1\text{:} -2.46264 - 1.25203 * (-2+\theta_2) + (-2+\theta_2)^2 \\ \overline{g}_2\text{:} -0.4324275990814064 + (-2+\theta_2)^2\end{matrix} \quad (13)$$

Thus, the intersections with g_1 and g_2 on the vertical line can be solved, i.e., B_1: (0.747973, 0.936478), B_2: (0.747973, 1.34241), B_3: (0.747973, 2.65759).

(2) Based on B_1, B_2, B_3 and the corresponding curves, the 2D cylindrical subspace on [A_1, A_2] can be decomposed into four parts, which are marked by different colors in Figure 3(b).

(3) On the vertical line, four sampling points in four parts can be determined, i.e., C_1, C_2, C_3 and C_4, where C_2, C_3 and C_4 are the midpoints of [B_1, B_2], [B_2, B_3] and [B_3, 4], respectively.

(4) C_1, C_2, C_3 and C_4 are used to check the feasibility of all the subspaces. When substituting C_1 to g_1 and g_2, we can get ($g_1 > 0$, $g_2 > 0$), which means that the subspace including C_1 is infeasible, because they cannot meet the original signs, $g_1 \leq 0$, $g_2 \leq 0$. Similarly, for C_2, C_3 and C_4, we can obtain ($g_1 < 0$, $g_2 > 0$), ($g_1 < 0$, $g_2 < 0$) and ($g_1 < 0$, $g_2 > 0$), respectively, and only the yellow subspace including C_3 is feasible. Therefore, it is a part of the entire feasible region.

(5) As shown in Figure 3(c), on [A_1, A_2], the upper boundary of the feasible subspace is (D_1, D_3) and the lower boundary is (D_1, D_2). The coordinate of B_3 is (0.747973, 2.65759). In Equation (13), 2.65759 is the second root of $\overline{g}_2 = 0$; thus, for any value of $\overline{\theta}_1$ in [A_1, A_2], the corresponding values of θ_2 on (D_1, D_3) are the second roots of $g_2(\overline{\theta}_1,\theta_2) = 0$, denoted as $Root(g_2\&, 2)$. Similarly, the values of θ_2 on lower boundary (D_1, D_2) are the first roots of $g_2(\overline{\theta}_1,\theta_2) = 0$, i.e., $Root(g_2\&, 1)$. Therefore, this subregion can be expressed as

$$\begin{cases}0.585786 \leq \theta_1 \leq 0.910159 \\ Root(g_2\&, 1) \leq \theta_2 \leq Root(g_2\&, 2)\end{cases} \quad (14)$$

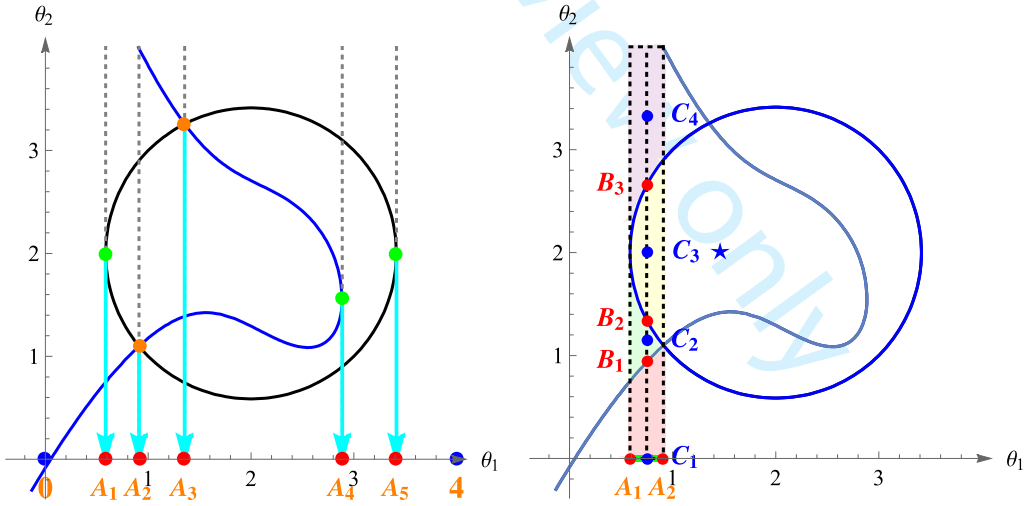
which is a triangular and explicit expression of this feasible subspace. Moreover, since g_2

is a bivariate polynomial, θ_2 can be solved explicitly, and Equation (14) is reformulated as,

$$\begin{cases} 0.585786 \le \theta_1 \le 0.910159 \\ 2 - \sqrt{-2 + 4\theta_1 - \theta_1^2} \le \theta_2 \le 2 + \sqrt{-2 + 4\theta_1 - \theta_1^2} \end{cases} \tag{15}$$

After checking all the cylindrical subspaces, as shown in Figure 3(d) and Figure 3(e), another two feasible subspaces can be found. Finally, we know that the feasible region consists of three subregions, as shown in Figure 3(f). Equation (16) shows the complete triangular and explicit expressions of the feasible region. The command *CylindricalDecomposition* in Mathematica[32] can perform the CAD method.

$$\begin{aligned} 1.&\begin{cases} 0.585786 \le \theta_1 \le 0.910159 \\ 2 - \sqrt{-2 + 4\theta_1 - \theta_1^2} \le \theta_2 \le 2 + \sqrt{-2 + 4\theta_1 - \theta_1^2} \end{cases} \\ 2.&\begin{cases} 0.910159 < \theta_1 \le 1.35389 \\ 0.5(6 - \theta_1) - 0.5\sqrt{38 - 52\theta_1 + 25\theta_1^2 - 4\theta_1^3} \le \theta_2 \le \\ 2 + \sqrt{-2 + 4\theta_1 - \theta_1^2} \end{cases} \\ 3.&\begin{cases} 1.35389 < \theta_1 \le 2.88639 \\ 0.5(6 - \theta_1) - 0.5\sqrt{38 - 52\theta_1 + 25\theta_1^2 - 4\theta_1^3} \le \theta_2 \le \\ 0.5(6 - \theta_1) + 0.5\sqrt{38 - 52\theta_1 + 25\theta_1^2 - 4\theta_1^3} \end{cases} \end{aligned} \tag{16}$$



(a) (b)

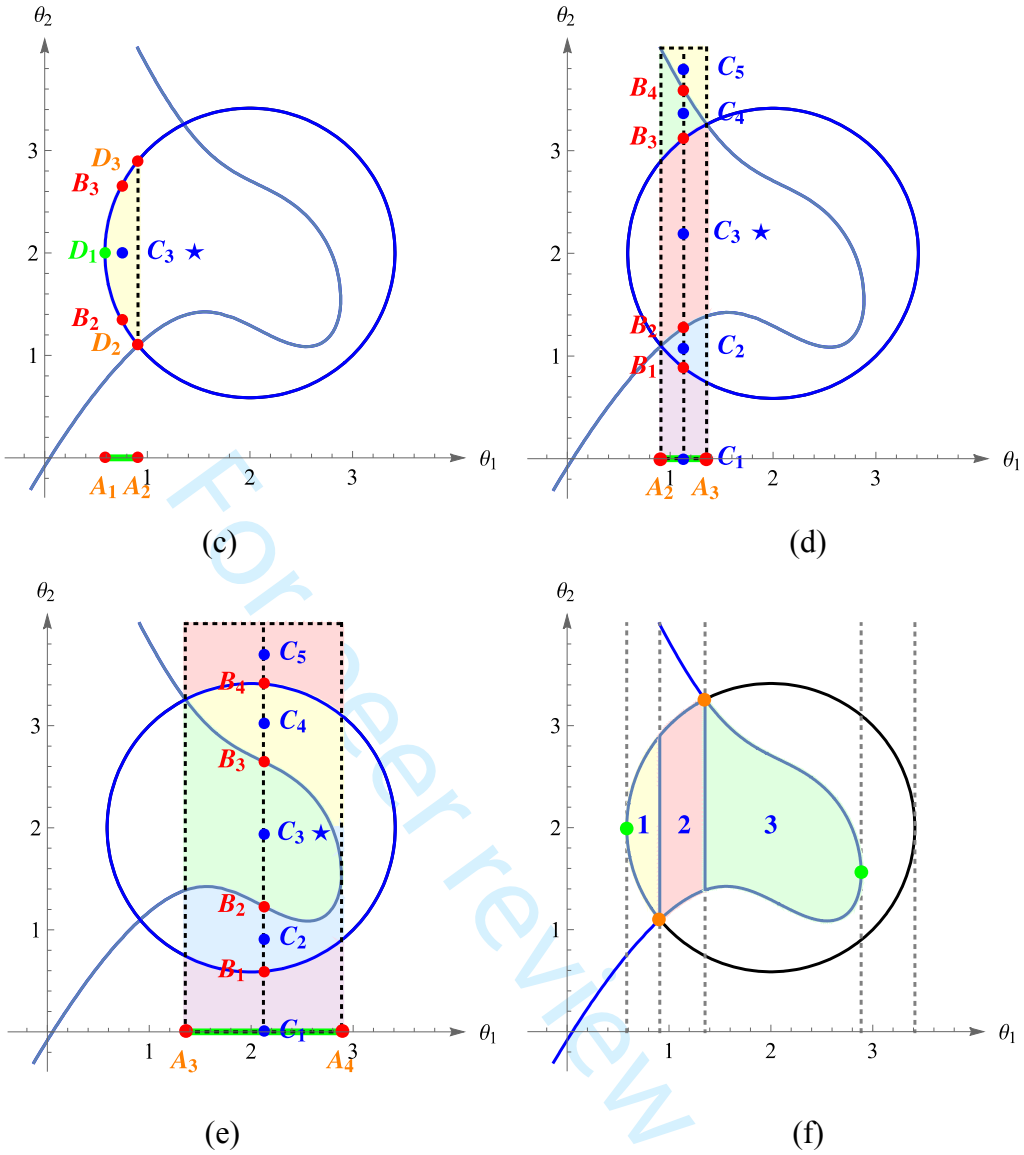


Figure 3. Solution process of the CAD method for the example.

According to the KS function, for a fixed ρ, the design space can be approximated by

$$y := KS(\boldsymbol{\theta},\boldsymbol{x}) \leq 0 \tag{17}$$

However, Equation (17) is an implicit and transcendental function, which cannot be directly used for symbolic computation. A common way is to create its polynomial surrogate model. As shown in Figure 4, $KS(\boldsymbol{\theta},\boldsymbol{x})$ aggregates all the inequality constraints. The surrogate model $\hat{y}(\overline{\boldsymbol{\theta}})$ can then be created through polynomial fitting, while the state variables $\boldsymbol{x}$ are eliminated. Thus, the CAD result of $\hat{y}(\overline{\boldsymbol{\theta}}) \leq 0$ contains a series of subspaces, each of which is a triangular system of $\overline{\theta}_1,\ldots,\overline{\theta}_P$.

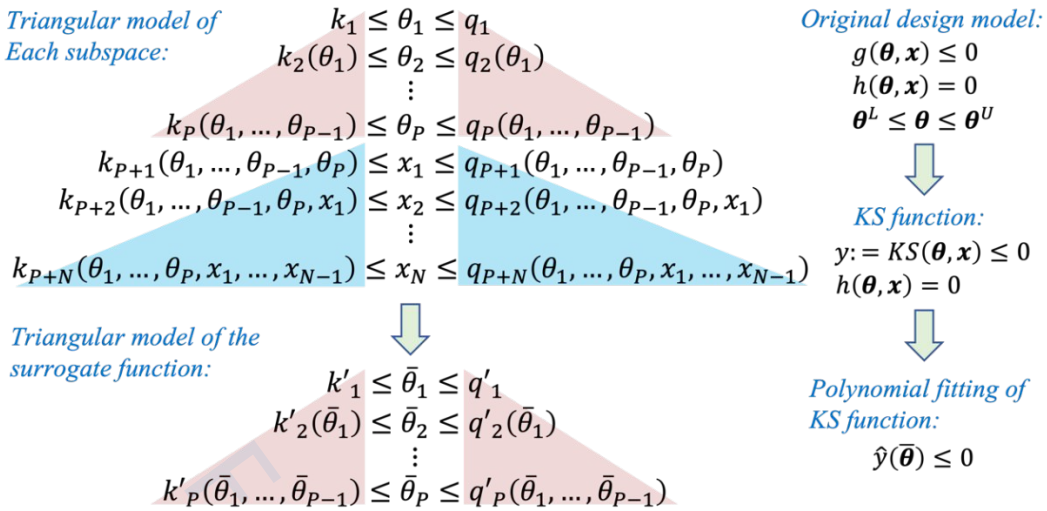


Figure 4. Triangular structure obtained by the CAD method.

Once the polynomial surrogate model $\hat{y}(\bar{\boldsymbol{\theta}})$ is obtained, it is convenient to analyze the design space, because the space can be described by a single constraint. Moreover, as shown in Figure 2(b), the smooth profile of the KS function has no intersection points, and we do not need to calculate the resultant for a single KS function.

5. Polynomial fitting of KS function through adaptive sampling

To create a polynomial surrogate model of the KS function, an adaptive sampling strategy is proposed. According to Equation (17), a point on the boundary of the KS space is denoted as

$$\{(\theta_1,\dots,\theta_P) \in \mathbb{R}^P | KS(\boldsymbol{\theta},\boldsymbol{x}) = 0\} \tag{18}$$

which can be solved by the following system of equations,

$$\begin{aligned} &KS(\boldsymbol{\theta},\boldsymbol{x}) = 0 \\ &h_i(\boldsymbol{\theta},\boldsymbol{x}) = 0, \quad \forall i \in I \end{aligned} \tag{19}$$

Note that, since the bounds of process parameters are also used for formulating the KS function, the KS space must be a closed space. If the design space is described by a set of inequality constraints, the traditional methods to build a surrogate model need to sample enough points over the whole space. However, in this work, a single equation, $KS(\boldsymbol{\theta},\boldsymbol{x}) = 0$, can describe the entire boundary of the KS space. Thus, we only focus on exploring enough sampling points on the boundary to create the surrogate model. The solution strategy of the proposed adaptive

sampling method is shown in Figure 5, which contains four steps:

(1) *Initial LHS sampling*. For a given number of sampling points, perform Latin hypercube sampling (LHS) over the space of process parameters. Those feasible sampled points in the feasible region forms a set **A**;

(2) *Expanded boundary points*. Through each pair of points in **A**, a line can be generated. Two intersection points of the line and the boundary can be obtained. All the intersection points on the boundary form an initial set **B**;

(3) *Adaptive sampling*. Based on the KS function, a simple DFO model of adaptive sampling is proposed. K-fold cross validation is applied to evaluate the stopping criteria. For each adaptive sampling point in set **C**, P points around the current adaptive point can be found through perturbation, which can form set **D**. All four sets are used for polynomial fitting.

(4) *Explicit description*. Through the CAD method, the fitted polynomial model $\hat{y}(\overline{\boldsymbol{\theta}})$ can be used to deduce the explicit expression of the KS space.

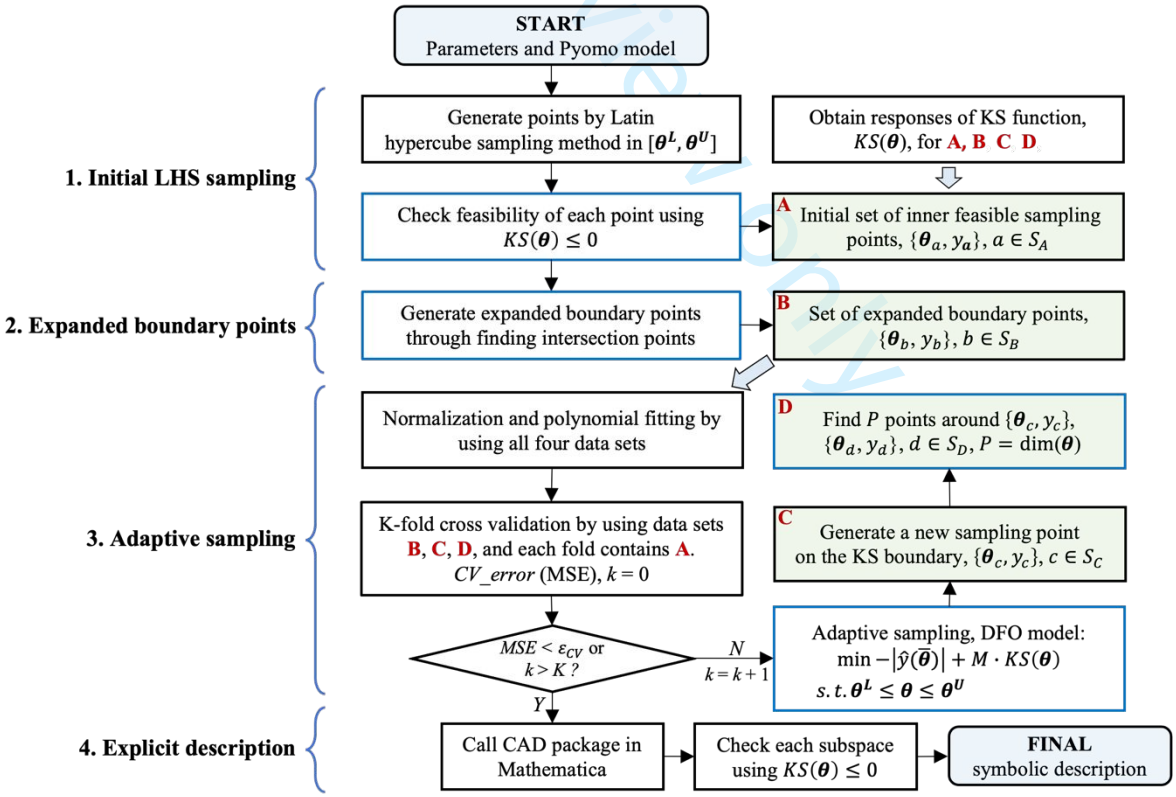


Figure 5. Solution strategy of the adaptive sampling method.

5.1. Initial LHS sampling

The LHS strategy[33] is applied to generate a given number of sampling points, denoted as num, in the process parameter space, where upper and lower bounds are required. Then, a feasibility check needs to be performed by evaluating $KS(\boldsymbol{\theta},\boldsymbol{x})$ at each LHS point. The presence of the state variables implies that the KS space is described by a set of multivariate functions of $\boldsymbol{\theta}$ and $\boldsymbol{x}$. Thus, the feasibility can be checked by solving the following NLP model,

$$\begin{aligned} & u_a = \min u \\ & s.t.\ KS(\boldsymbol{\theta},\boldsymbol{x}) \le u \\ & \quad h_i(\boldsymbol{\theta},\boldsymbol{x}) = 0,\ \ \forall i \in I \end{aligned} \tag{20}$$

where $u_a \le 0$ indicates that the point is feasible. All the feasible LHS sampling points and the corresponding output of $KS(\boldsymbol{\theta},\boldsymbol{x})$ form set $\mathbf{A} := \{(\boldsymbol{\theta}_a, y_a), \forall a \in S_A\}$. Note that the number of LHS sampling points needs to be specified in advance. The more sampling points specified, the larger set **A** obtained, and the easier polynomial fitting will be.

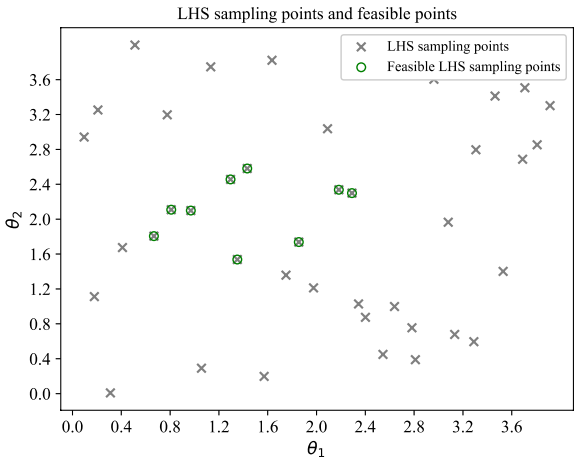


Figure 6. Initial LHS sampling of the example. ($\rho = 2, num = 40$)

For the above-mentioned example with $\rho = 2$, the sampling ranges are set to [0, 4] and [0, 4] for θ_1 and θ_2. As shown in Figure 6, 40 points are sampled, and 9 of them are feasible.

5.2. Expanded boundary points

To build the surrogate model, we need to find enough sampling points on the boundary of the

KS space. In this subsection, based on the properties of the KS function and the obtained initial set **A**, an approach to efficiently locate many points on the boundary is proposed. For a pair of points in the set **A**, e.g., $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$, there are two directions, $\boldsymbol{\theta}_1 \rightarrow \boldsymbol{\theta}_2$ and $\boldsymbol{\theta}_2 \rightarrow \boldsymbol{\theta}_1$, which can be formulated as follows,

$$\begin{aligned} &\boldsymbol{\theta}_1 \rightarrow \boldsymbol{\theta}_2: \ \ \boldsymbol{\theta} = \boldsymbol{\theta}_1 + \delta \cdot (\boldsymbol{\theta}_2 - \boldsymbol{\theta}_1) \\ &\boldsymbol{\theta}_2 \rightarrow \boldsymbol{\theta}_1: \ \ \boldsymbol{\theta} = \boldsymbol{\theta}_2 + \delta \cdot (\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2) \end{aligned} \tag{21}$$

Along both directions, we can find two intersection points on the KS boundary by solving the following two optimization problems,

$$\begin{aligned} &\max \delta \\ &s.t. \ \ KS(\boldsymbol{\theta},\boldsymbol{x}) = 0 \\ &\quad\ h_i(\boldsymbol{\theta},\boldsymbol{x}) = 0, \ \ \forall i \in I \\ &\quad\ \boldsymbol{\theta} = \boldsymbol{\theta}_1 + \delta \cdot (\boldsymbol{\theta}_2 - \boldsymbol{\theta}_1) \end{aligned} \tag{22}$$

$$\begin{aligned} &\max \delta \\ &s.t. \ \ KS(\boldsymbol{\theta},\boldsymbol{x}) = 0 \\ &\quad\ h_i(\boldsymbol{\theta},\boldsymbol{x}) = 0, \ \ \forall i \in I \\ &\quad\ \boldsymbol{\theta} = \boldsymbol{\theta}_2 + \delta \cdot (\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2) \end{aligned} \tag{23}$$

If the set **A** contains S_A points, the total number of directions for all the pairs of points can be calculated from Equation (24), which is a permutation problem. The obtained boundary points and the output of $KS(\boldsymbol{\theta},\boldsymbol{x})$ then define the set of initial boundary points, $\mathbf{B} := \{(\boldsymbol{\theta}_b, y_b), \forall b \in S_B\}$. For example, in Figure 7, 4 points can generate $4 \times 3 = 12$ boundary points.

$$S_B = A_{S_A}^2 = S_A \cdot (S_A - 1) \tag{24}$$

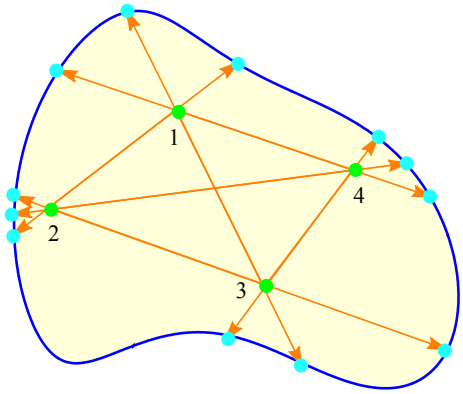


Figure 7. Illustration of expanded boundary points via line projection.

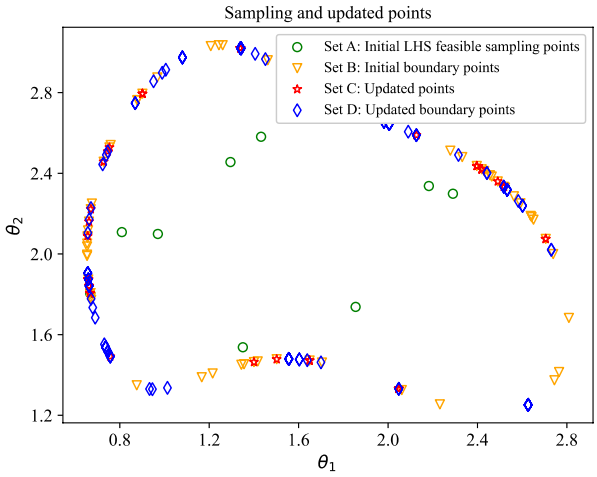


Figure 8. Sampling points of the example. ($\rho = 2, num = 40$).

For this example, 9 × 8=72 initial boundary points can be obtained by solving Equations (22) and (23), which are marked as yellow points in Figure 8. Note that, even though we only have 9 feasible LHS feasible points, 72 boundary points can be directly found in this step.

5.3. Adaptive sampling

Based on the set of feasible LHS points **A** and the initial set of expanded boundary points **B**, a common multivariable polynomial fitting method can be executed to build the surrogate model between the process parameters $\boldsymbol{\theta} = (\theta_1,...,\theta_P)$ and the response y, denoted as $\hat{y}(\boldsymbol{\theta})$,

$$\hat{y}(\boldsymbol{\theta}) = \sum_{k=0}^{K} c_k \phi_k(\boldsymbol{\theta}) \quad where\ \phi_k(\boldsymbol{\theta}) = \theta_1^{k_1} \cdots \theta_P^{k_P} \tag{25}$$

where $c_k \in \mathbb{R}$ and $c_K \neq 0$. The sum is taken over all indices (nonnegative integer vectors) with $k_P \leq deg$. For example, a bi-variable polynomial with $deg = 3$, is formulated as follows,

$$\begin{aligned}\hat{y}(\theta_1,\theta_2) = c_0 + c_1\theta_2 + c_2\theta_2^2 + c_3\theta_2^3 + c_4\theta_1 + c_5\theta_1\theta_2 + \\ c_6\theta_1\theta_2^2 + c_7\theta_1\theta_2^3 + c_8\theta_1^2 + c_9\theta_1^2\theta_2 + c_{10}\theta_1^2\theta_2^2 + \\ c_{11}\theta_1^2\theta_2^3 + c_{12}\theta_1^3 + c_{13}\theta_1^3\theta_2 + c_{14}\theta_1^3\theta_2^2 + c_{15}\theta_1^3\theta_2^3\end{aligned}$$

The K-fold cross-validation (CV) method[34] is used to evaluate the result of polynomial fitting. The number of folds, k_f, should be specified in advance. ε_{CV} is given as the stop criteria of CV. The set of **B**, **C** and **D** is split into k_f folds for cross validation, and the set **A** is used for each

fold. The maximum MSE (mean squared error) for all folds is used for comparison with ε_{CV}. Moreover, the data set of process parameters should be normalized before polynomial fitting, denoted as $\overline{\boldsymbol{\theta}}$. Then, a new point $\boldsymbol{\theta}$ can be roughly scaled by the current stored minimization $\boldsymbol{\theta}^{min}$ and maximization $\boldsymbol{\theta}^{max}$, and the corresponding output of the fitted model is $\hat{y}(\overline{\boldsymbol{\theta}})$,

$$\overline{\boldsymbol{\theta}} = \frac{\boldsymbol{\theta} - \boldsymbol{\theta}^{min}}{\boldsymbol{\theta}^{max} - \boldsymbol{\theta}^{min}} \tag{26}$$

The purpose of adaptive sampling is to search the design space for areas of model inconsistency or model mismatch and to find points that maximize the model error. Since only the points on the boundary are considered, the adaptive sampling model is proposed as,

$$\begin{aligned} &\max |\hat{y}(\overline{\boldsymbol{\theta}})| \\ &s.t.\ KS(\boldsymbol{\theta},\boldsymbol{x}) = 0 \\ &\quad h_i(\boldsymbol{\theta},\boldsymbol{x}) = 0,\ \ \forall i \in I \\ &\quad \boldsymbol{\theta}^{\boldsymbol{L}} \le \boldsymbol{\theta} \le \boldsymbol{\theta}^{\boldsymbol{U}} \end{aligned} \tag{27}$$

After relaxing the objective function $|\hat{y}(\overline{\boldsymbol{\theta}})|$ into two constraints, $\hat{y}(\overline{\boldsymbol{\theta}}) \le Z$ and $-\hat{y}(\overline{\boldsymbol{\theta}}) \le Z$, we can apply an NLP solver to solve Equation (27); however, most adaptive sampling points are likely to fall into the same area that has the maximum model error and cannot escape to explore other areas, which is not conducive to polynomial fitting. Therefore, in order to explore the areas that contain more local information, we transform Equation (27) into the following derivative-free optimization (DFO) model,

$$\begin{aligned} &\min\ -|\hat{y}(\overline{\boldsymbol{\theta}})| + M \cdot KS(\boldsymbol{\theta}) \\ &s.t. \boldsymbol{\theta}^{\boldsymbol{L}} \le \boldsymbol{\theta} \le \boldsymbol{\theta}^{\boldsymbol{U}} \end{aligned} \tag{28}$$

where M is a penalty coefficient. Compared with Equation (27), the equality system, i.e., $h_i(\boldsymbol{\theta},\boldsymbol{x}) = 0, \forall i \in I$, can be viewed as a black box. This black-box model is used to calculate the state variables $\boldsymbol{x}$ for each iteration of adaptive sampling, and then to evaluate the objective function of Equation (28).

A DFO solver, Py-BOBYQA, which is a Python implementation of the BOBYQA (Bound Optimization BY Quadratic Approximation) Fortran solver by Powell[35], is employed to solve

the adaptive sampling model. In this work, the purpose of adaptive sampling is to explore local areas leading to model mismatch as many as possible, rather than finding global optima in each iteration. Py-BOBYQA is based on the trust-region method, and it has superior performance in finding local solutions of nonlinear and nonconvex problems. Thus, we use this DFO solver to address the adaptive sampling problem. In addition, the initial values are of great importance to the results of the DFO solver. To take a more complete search, a reproducible random sequence of the data set is generated as the initial values, and more local optima can be found. The obtained adaptive sampling points form set $\mathbf{C} := \{(\boldsymbol{\theta}_c, y_c), \forall c \in S_C\}$. To further expand the points on the boundary, for a point in $\mathbf{C}$, $\boldsymbol{\theta}_c := (\theta_{c_1},...,\theta_{c_i},...,\theta_{c_P})$, a perturbation method is proposed to find P points around $\boldsymbol{\theta}_c$, where P is the dimension of $\boldsymbol{\theta}$,

$$\boldsymbol{\theta}_c^i = (\theta_{c_1}^i,...\theta_{c_i}^i,...,\theta_{c_P}^i), \quad \forall i \in P \tag{29}$$

First, we can find a point $\boldsymbol{\theta}_a := (\theta_{a_1},...,\theta_{a_i},...,\theta_{a_P})$ in set $\mathbf{A}$, which has the longest Euclidean distance to $\boldsymbol{\theta}_c$. Then, assuming that the perturbation factor is α, if $\boldsymbol{\theta}_a$ is located on the left side of $\boldsymbol{\theta}_c$ on the coordinate θ_i, i.e., $\theta_{a_i} < \theta_{c_i}$, $\theta_{c_i}^i$ can be calculated by,

$$\theta_{c_i}^i = \theta_{c_i} \cdot (1 + \alpha), \quad i \in \{1,2,...,P\} \tag{30}$$

If $\boldsymbol{\theta}_a$ is located on the right side of $\boldsymbol{\theta}_c$, i.e., $\theta_{a_i} > \theta_{c_i}$, $\theta_{c_i}^i$ can be calculated by,

$$\theta_{c_i}^i = \theta_{c_i} \cdot (1 - \alpha), \quad i \in \{1,2,...,P\} \tag{31}$$

If $\theta_{a_i} = \theta_{c_i}$, $\theta_{c_i}^i = \theta_{c_i}$. For each point $\boldsymbol{\theta}_c^i$, the direction $\boldsymbol{\theta}_a \rightarrow \boldsymbol{\theta}_c^i$, can be formulated as,

$$\boldsymbol{\theta} = \boldsymbol{\theta}_a + \delta \cdot (\boldsymbol{\theta}_c^i - \boldsymbol{\theta}_a) \tag{32}$$

Through Equation (22), the intersection point on the KS boundary can be solved. All such intersection points can form set $\mathbf{D} := \{(\boldsymbol{\theta}_d, y_d), \forall d \in S_D\}$.

To illustrate the above method, Figure 9 shows geometric interpretation of a bivariate case. In Figure 9(a), $\boldsymbol{\theta}_c = (\theta_{c_1}, \theta_{c_2})$ is the current adaptive sampling point, and the blue point $\boldsymbol{\theta}_a = (\theta_{a_1}, \theta_{a_2})$ is a point in set $\mathbf{A}$ which is the farthest point from $\boldsymbol{\theta}_c$. Since the dimension P is 2 and

$\boldsymbol{\theta}_a$ is on the left of $\boldsymbol{\theta}_c$, two perturbation points around $\boldsymbol{\theta}_c$ can be expressed as follows,

$$\boldsymbol{\theta}_c^1 = (\theta_{c_1} \cdot (1+\alpha), \theta_{c_2})$$

$$\boldsymbol{\theta}_c^2 = (\theta_{c_1}, \theta_{c_2} \cdot (1+\alpha))$$

Thus, the following two direction formulations can be used to find two intersection points, $\boldsymbol{\theta}_d^1$ and $\boldsymbol{\theta}_d^2$, on the boundary,

$$\boldsymbol{\theta} = \boldsymbol{\theta}_a + \delta \cdot (\boldsymbol{\theta}_c^1 - \boldsymbol{\theta}_a)$$

$$\boldsymbol{\theta} = \boldsymbol{\theta}_a + \delta \cdot (\boldsymbol{\theta}_c^2 - \boldsymbol{\theta}_a)$$

Similarly, as shown in Figure 9(b), $\boldsymbol{\theta}_a$ is on the right of $\boldsymbol{\theta}_c$, thus,

$$\boldsymbol{\theta}_c^1 = (\theta_{c_1} \cdot (1-\alpha), \theta_{c_2})$$

$$\boldsymbol{\theta}_c^2 = (\theta_{c_1}, \theta_{c_2} \cdot (1-\alpha))$$

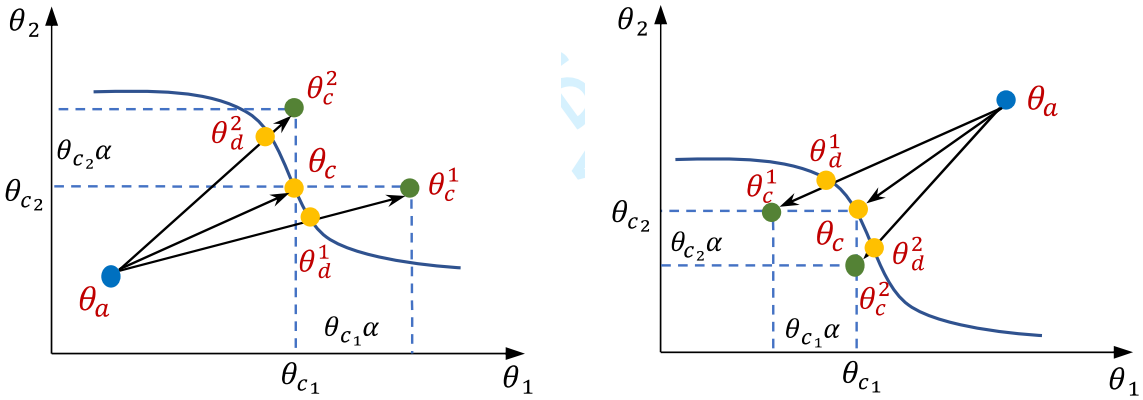
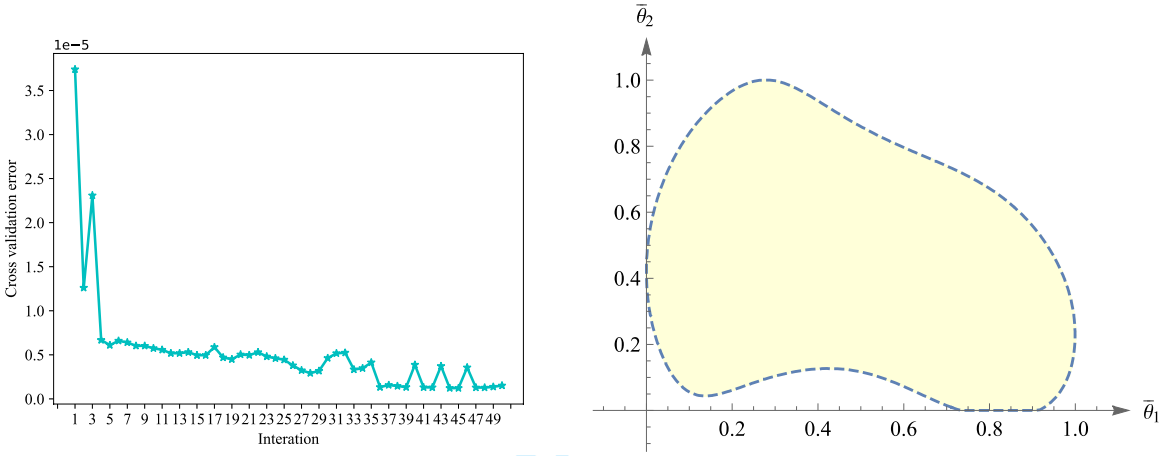


Figure 9. Perturbation points for an adaptive sampling point.

For the example, the maximum number of iterations for adaptive sampling is set to 50. Figure 8 shows the results of four different sets of points. All the points are used for polynomial fitting. Figure 10(a) shows the MSE values of 4-fold CV. The final surrogate model with 4 degrees is

$$\begin{aligned}\hat{y}_0(\bar{\theta}_1,\bar{\theta}_2) = {} & 0.332558 - 4.924684\bar{\theta}_1 + 27.677621\bar{\theta}_1^{\,2} - 48.130965\bar{\theta}_1^{\,3} + 25.497110\bar{\theta}_1^{\,4} \\ & - 1.355176\bar{\theta}_2 + 5.423242\bar{\theta}_1\bar{\theta}_2 - 54.471859\bar{\theta}_1^{\,2}\bar{\theta}_2 + 89.127179\bar{\theta}_1^{\,3}\bar{\theta}_2 \\ & - 43.856384\bar{\theta}_1^{\,4}\bar{\theta}_2 + 1.129803\bar{\theta}_2^{\,2} - 18.827193\bar{\theta}_1\bar{\theta}_2^{\,2} + 60.655279\bar{\theta}_1^{\,2}\bar{\theta}_2^{\,2} \\ & + 21.796085\bar{\theta}_1^{\,3}\bar{\theta}_2^{\,2} - 44.339153\bar{\theta}_1^{\,4}\bar{\theta}_2^{\,2} - 0.2916\bar{\theta}_2^{\,3} + 14.54047\bar{\theta}_1\bar{\theta}_2^{\,3} \\ & + 54.884484\bar{\theta}_1^{\,2}\bar{\theta}_2^{\,3} - 326.654775\bar{\theta}_1^{\,3}\bar{\theta}_2^{\,3} + 223.200627\bar{\theta}_1^{\,4}\bar{\theta}_2^{\,3} + 1.887766\bar{\theta}_2^{\,4} \\ & - 11.284009\bar{\theta}_1\bar{\theta}_2^{\,4} - 44.465319\bar{\theta}_2^{\,4} + 212.964235\bar{\theta}_1^{\,3}\bar{\theta}_2^{\,4} - 134.362005\bar{\theta}_1^{\,4}\bar{\theta}_2^{\,4}\end{aligned}$$



(a) (b)

Figure 10. Surrogate model of the example ($\rho = 2, num = 40$).

5.4. Explicit description

Since the sampling data has been normalized before performing polynomial fitting, the profile of the KS space, $\hat{y}(\bar{\boldsymbol{\theta}}) \le 0$, should be limited within a square with [0,1] sides. Considering the polynomial fitting error, the profile may be slightly outside of the square. Hence, the KS space based on the surrogate model is the intersection of $\hat{y}(\bar{\boldsymbol{\theta}}) \le 0$ and $0 \le \bar{\boldsymbol{\theta}} \le 1$, i.e.,

$$\{\bar{\boldsymbol{\theta}} \in \mathbb{R}^P | \hat{y}(\bar{\boldsymbol{\theta}}) \le 0 \wedge 0 \le \bar{\boldsymbol{\theta}} \le 1\} \tag{33}$$

The CAD method can transform Equation (33) into a triangular system, which consists of a series of subspaces, and each subspace is formulated as,

$$\begin{gathered} k_1' \le \bar{\theta}_1 \le q_1' \\ k_2'(\bar{\theta}_1) \le \bar{\theta}_2 \le q_2'(\bar{\theta}_1) \\ \vdots \\ k_P'(\bar{\theta}_1,\ldots,\bar{\theta}_{P-1}) \le \bar{\theta}_P \le q_P'(\bar{\theta}_1,\ldots,\bar{\theta}_{P-1}) \end{gathered} \tag{34}$$

For the example, the corresponding triangular system with $\overline{\theta}_1 \prec \overline{\theta}_2$ is as follows:

$$1:\begin{cases}0 \le \overline{\theta}_1 < 0.0964731 \\ Root(\Gamma_0(\overline{\theta}_1, \#1)\&,1) \le \overline{\theta}_2 \le Root(\Gamma_0(\overline{\theta}_1, \#1)\&, 2)\end{cases}$$

$$2:\begin{cases}0.0964731 \le \overline{\theta}_1 < 0.137589 \\ Root(\Gamma_0(\overline{\theta}_1, \#1)\&,3) \le \overline{\theta}_2 \le Root(\Gamma_0(\overline{\theta}_1, \#1)\&,4)\end{cases}$$

$$3:\begin{cases}0.137589 \le \overline{\theta}_1 \le 0.266024 \\ Root(\Gamma_0(\overline{\theta}_1, \#1)\&,2) \le \overline{\theta}_2 \le Root(\Gamma_0(\overline{\theta}_1, \#1)\&, 3)\end{cases}$$

$$4:\begin{cases}0.266024 < \overline{\theta}_1 \le 0.290682 \\ Root(\Gamma_0(\overline{\theta}_1, \#1)\&,2) \le \overline{\theta}_2 \le 1\end{cases}$$

$$5:\begin{cases}0.290682 < \overline{\theta}_1 \le 0.376366 \\ Root(\Gamma_0(\overline{\theta}_1, \#1)\&,2) \le \overline{\theta}_2 \le Root(\Gamma_0(\overline{\theta}_1, \#1)\&, 3)\end{cases}$$

$$6:\begin{cases}0.376366 < \overline{\theta}_1 \le 0.732135 \\ Root(\Gamma_0(\overline{\theta}_1, \#1)\&,1) \le \overline{\theta}_2 \le Root(\Gamma_0(\overline{\theta}_1, \#1)\&,2)\end{cases}$$

$$7:\begin{cases}0.732135 < \overline{\theta}_1 \le 0.91514 \\ 0 \le \overline{\theta}_2 \le Root(\Gamma_0(\overline{\theta}_1, \#1)\&, 2)\end{cases}$$

$$8:\begin{cases}0.91514 < \overline{\theta}_1 \le 1 \\ Root(\Gamma_0(\overline{\theta}_1, \#1)\&,1) \le \overline{\theta}_2 \le Root(\Gamma_0(\overline{\theta}_1, \#1)\&,2)\end{cases}$$

where $\Gamma_0(\overline{\theta}_1, \#1)$ is a pure function with the parameter $\overline{\theta}_1$, which is provided in Supporting Information. $Root(\Gamma_0(\overline{\theta}_1, \#1)\&,n)$ means nth root of $\Gamma_0(\overline{\theta}_1, \#1) = 0$ for a given $\overline{\theta}_1$. The result indicates that the bounds of $\overline{\theta}_2$ are explicit expressions of $\overline{\theta}_1$. To further test the feasibility of the subspaces, a random point $\overline{\boldsymbol{\theta}}_s$ can be converted into the KS space by Equation (35), denoted as $\boldsymbol{\theta}_s$. $KS(\boldsymbol{\theta}_s) \le 0$ means that the current subspace is feasible,

$$\boldsymbol{\theta}_s = \overline{\boldsymbol{\theta}}_s \cdot (\boldsymbol{\theta}^{max} - \boldsymbol{\theta}^{min}) + \boldsymbol{\theta}^{min} \tag{35}$$

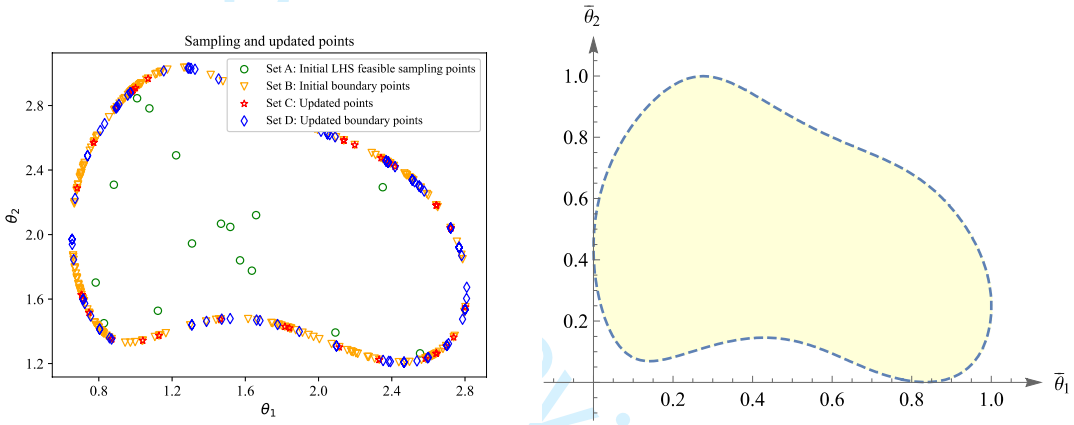
In the original data set, the maximum values of θ_1 and θ_2 are 2.809134, 3.03503, and the minimum values are 0.653351, 1.251803, respectively. Taking the first subspace as an example, a chosen point is $\overline{\boldsymbol{\theta}}_s$ = (0.05, 0.4) and the corresponding $\boldsymbol{\theta}_s$ is (0.761141, 1.965094). $KS(\boldsymbol{\theta}_s) = -0.207169 < 0$ means that the first subspace is feasible. Similarly, we can find that the other subspaces are also feasible. The final KS space is shown in Figure 10(b).

Note that through Equations (26), (34) and (35), the CAD result can be used to evaluate the KS space. Each point satisfying $\hat{y}(\overline{\boldsymbol{\theta}}) \le 0$ can be transformed into the KS space. The triangular

formulation of KS space is formulated as follows,

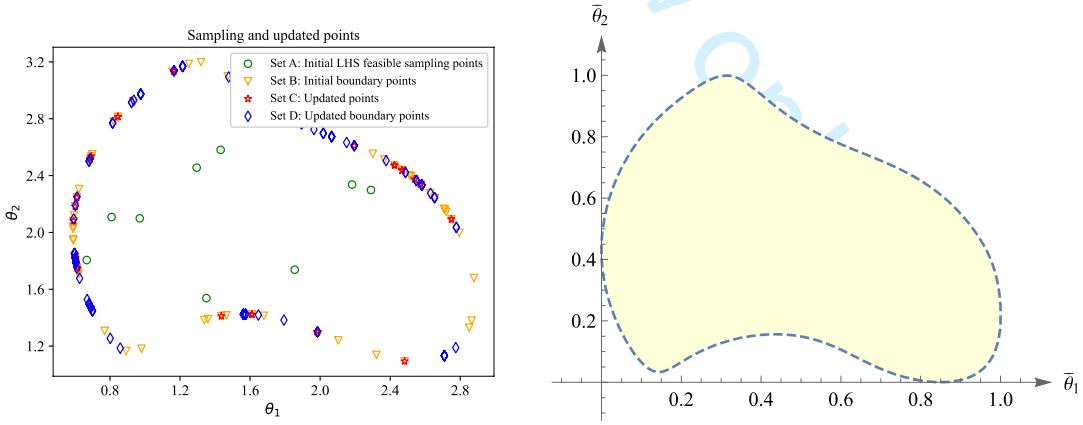
$$\begin{aligned}
&\boldsymbol{\theta}_1^{min} + \left(\boldsymbol{\theta}_1^{max} - \boldsymbol{\theta}_1^{min}\right) \cdot k_1' \le \theta_1 \le q_1' \cdot \left(\boldsymbol{\theta}_1^{max} - \boldsymbol{\theta}_1^{min}\right) + \boldsymbol{\theta}_1^{min} \\
&\boldsymbol{\theta}_2^{min} + \left(\boldsymbol{\theta}_2^{max} - \boldsymbol{\theta}_2^{min}\right) \cdot k_2'\left(\frac{\theta_1 - \boldsymbol{\theta}_1^{min}}{\boldsymbol{\theta}_1^{max} - \boldsymbol{\theta}_1^{min}}\right) \le \theta_2 \le q_2'\left(\frac{\theta_1 - \boldsymbol{\theta}_1^{min}}{\boldsymbol{\theta}_1^{max} - \boldsymbol{\theta}_1^{min}}\right) \cdot \left(\boldsymbol{\theta}_2^{max} - \boldsymbol{\theta}_2^{min}\right) + \boldsymbol{\theta}_2^{min} \\
&\vdots
\end{aligned} \tag{36}$$

Moreover, another critical parameter, the number of initial LHS points, num, is also discussed. First, compared with the result shown in Figure 8, we apply the algorithm by setting $num = 80$. In this case, 16 feasible LHS points can be found, and more expanded boundary points can be located. In Figure 11, the denser boundary points can generate a more precise surrogate model.



(a) (b)

Figure 11. Sampling points and surrogate model of the example. ($\rho = 2, num = 80$).



(a) (b)

Figure 12. Sampling points and surrogate model of the example. ($\rho = 5, num = 40$).

Then, another key parameter is ρ. Compared with $\rho = 2$, we test the KS function with $\rho = 5$. All sampling points are shown in Figure 12(a). The KS space is illustrated in Figure 12(b). We

can see that the shape of the KS space with $\rho = 5$ more completely covers the feasible region than Figure 11(b). Therefore, we can conclude that more LHS points are more conducive to the polynomial fitting, and larger ρ can make the profile closer to the original feasible region.

6. Case studies

Two case studies are presented to illustrate the proposed design space description method. Pyomo[36] is applied to define the models. The GAMS global solver, BARON, is used to solve the NLP models through the interface of Pyomo and GAMS, and Py-BOBYQA is applied to solve the DFO models. For both cases, the parameters in Table 1 are set to the same values. The difference is the parameter ρ, which will be specified at different values.

Table 1. The specifications of the parameters.

Initial LHS points, *num*	Degree of polynomial, *deg*	Perturbation, α	Maximum Iteration	MSE of CV, ε_{CV}	K-fold, k_f
100	4	0.2	50	10^{-10}	4

6.1. CSTR reaction

This case study deals with a 2-step reaction with the following mechanism[10,37],

$$A + B \xrightarrow{k_1} C,\ r_1 = k_1 \cdot c_A \cdot c_b$$
$$C \xrightarrow{k_2} D + E,\ r_2 = k_2 \cdot c_c$$

where r_j are the reaction rates. Two process parameters correspond to the residence time, θ_1, and the ratio of the concentration of B to A, θ_2. k_j correspond to the model parameters, which are fixed at their mean values $\{0.31051, 0.026650\}$. The feasible ranges of θ_1 and θ_2 are given as follows.

$$0 \le \theta_1 \le 550$$
$$0 \le \theta_2 \le 6$$

The mass balance of the CSTR reaction is given by the following set of equations,

$$c_A^0 - c_A + \theta_1 \cdot (-r_1) = 0$$
$$c_B^0 - c_B + \theta_1 \cdot (-r_1) = 0$$
$$c_C^0 - c_C + \theta_1 \cdot (r_1 - r_2) = 0$$
$$c_D^0 - c_D + \theta_1 \cdot r_2 = 0$$
$$c_E^0 - c_E + \theta_1 \cdot r_2 = 0$$

where c_i^0 are initial concentrations $\{c_A^0 = 0.53, c_B^0 = 0.53 \cdot \theta_2, c_C^0 = 0, c_D^0 = 0, c_E^0 = 0\}$. The quality constraints are the minimum yield of *D* and the minimum ratio of *D* to unreacted species.

$$\frac{c_D}{c_A^0 - c_A} \geq 0.9$$
$$\frac{c_D}{c_A + c_B + c_C} \geq 0.2$$

In this case, ρ is fixed to 20. Figure 13(a) shows the feasible LHS points and updated points. All the updated points are located at the boundary of the KS space, and we can create the following 4-degree polynomial surrogate model with this small amount of sampling points.

$$\begin{aligned}\hat{y}_1(\bar{\theta}_1,\bar{\theta}_2) = {} & 0.00003937 + 0.001384\bar{\theta}_1 - 0.000118\bar{\theta}_1^2 - 0.003772\bar{\theta}_1^3 + 0.002389\bar{\theta}_1^4 \\ & - 0.000678\bar{\theta}_2 - 0.165238\bar{\theta}_1\bar{\theta}_2 - 0.563174\bar{\theta}_1^2\bar{\theta}_2 + 0.265011\bar{\theta}_1^3\bar{\theta}_2 \\ & + 0.464374\bar{\theta}_1^4\bar{\theta}_2 + 0.003796\bar{\theta}_2^2 - 0.790496\bar{\theta}_1\bar{\theta}_2^2 + 10.757368\bar{\theta}_1^2\bar{\theta}_2^2 \\ & - 15.685365\bar{\theta}_1^3\bar{\theta}_2^2 + 5.714148\bar{\theta}_1^4\bar{\theta}_2^2 - 0.007253\bar{\theta}_2^3 + 2.002811\bar{\theta}_1\bar{\theta}_2^3 \\ & - 21.109944\bar{\theta}_1^2\bar{\theta}_2^3 + 31.940187\bar{\theta}_1^3\bar{\theta}_2^3 - 12.825599\bar{\theta}_1^4\bar{\theta}_2^3 + 0.004422\bar{\theta}_2^4 \\ & - 1.035763\bar{\theta}_1\bar{\theta}_2^4 + 10.935095\bar{\theta}_1^2\bar{\theta}_2^4 - 16.595443\bar{\theta}_1^3\bar{\theta}_2^4 + 6.691871\bar{\theta}_1^4\bar{\theta}_2^4\end{aligned}$$

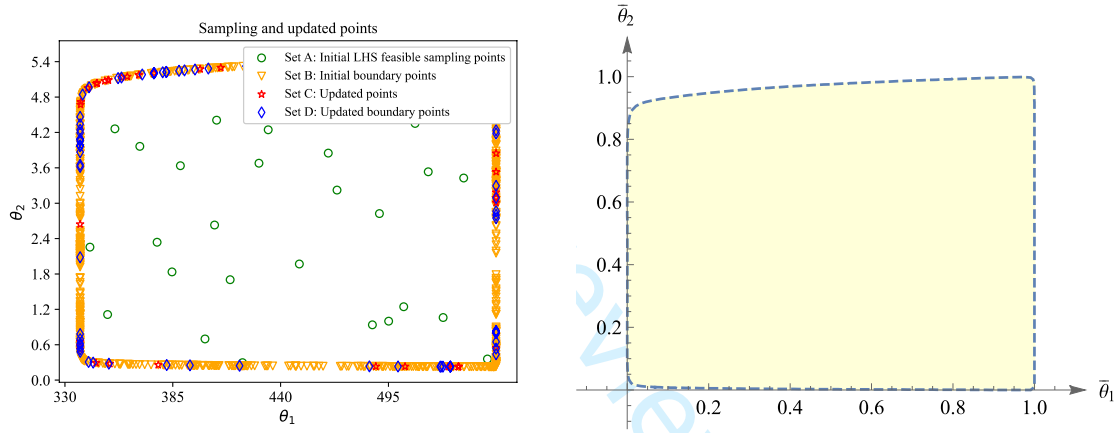
The profile of $\hat{y}_1(\bar{\theta}_1,\bar{\theta}_2) = 0$ is shown in Figure 13(b). Because of normalization, the profile is limited within a square with [0,1] sides. $\hat{y}_1(\bar{\theta}_1,\bar{\theta}_2) \leq 0$ can be used to describe the KS space. Through the CAD method, the equivalent triangular system can be obtained, which consists of 3 explicit 2D subspaces.

$$1:\begin{cases}0 \leq \bar{\theta}_1 \leq 0.107855 \\ Root(\Gamma_1(\bar{\theta}_1, \#1)\&, 2) \leq \bar{\theta}_2 \leq Root(\Gamma_1(\bar{\theta}_1, \#1)\&, 3)\end{cases}$$

$$2:\begin{cases}0.107855 < \bar{\theta}_1 < 0.90871 \\ Root(\Gamma_1(\bar{\theta}_1, \#1)\&, 1) \leq \bar{\theta}_2 \leq Root(\Gamma_1(\bar{\theta}_1, \#1)\&, 2)\end{cases}$$

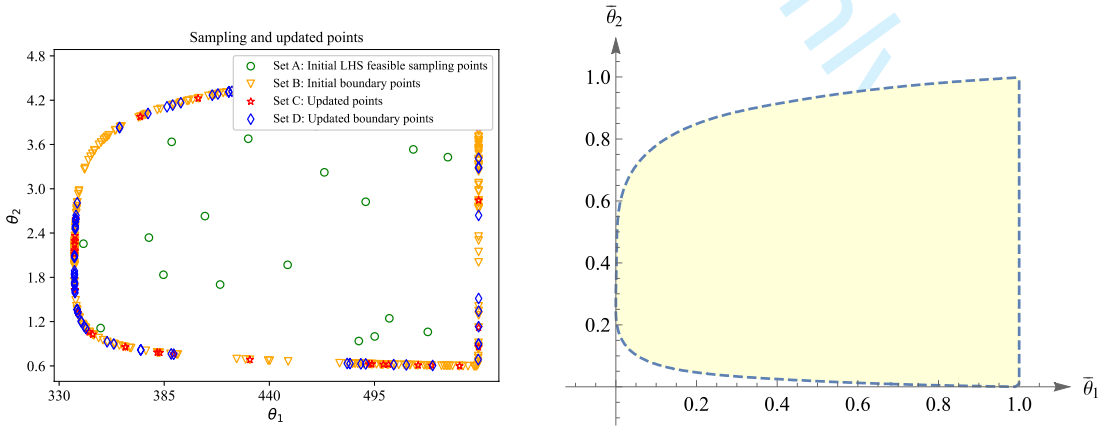
$$3:\begin{cases}0.90871 \leq \bar{\theta}_1 \leq 1 \\ 0 \leq \bar{\theta}_2 \leq Root(\Gamma_1(\bar{\theta}_1, \#1)\&, 2)\end{cases}$$

where $\Gamma_1(\overline{\theta}_1, \#1)$ is provided in Supporting Information. Due to the error of polynomial fitting, the feasibility of each subspace must be tested. In the stored data set, the maximum values of θ_1 and θ_2 are 549.94261, 5.491875; the minimum values are 337.704, 0.226248. Taking the second subspace as an example, a sampling point is chosen as $\overline{\boldsymbol{\theta}}_s$ = (0.5, 0.8). $\boldsymbol{\theta}_s$ = (443.823, 4.43875) is calculated by Equation (35). $KS(\boldsymbol{\theta}_s) = -0.011638 < 0$ indicates that the second subspace is feasible. Similarly, the other subspaces can be tested for feasibility Moreover, to show the performance of ρ, Figure 14 illustrates the results with $\rho = 5$. Since ρ is set smaller, the KS space is smaller than $\rho = 20$. The required CPU/Wall times are reported in Table 2.



(a) (b)

Figure 13. Results of design space description for Case 1. ($\rho = 20$, BARON)



(a) (b)

Figure 14. Results of design space description for Case 1. ($\rho = 5$, BARON)

6.2. Michael Addition Reaction

This case study deals with the Michael Addition Reaction with kinetics[10] described by the following equations,

$$AH + B \overset{k_1}{\rightarrow} A^- + BH^+,\ r_1 = k_1 \cdot c_{AH} \cdot c_B$$

$$A^- + C \overset{k_2}{\rightarrow} AC^-,\ r_2 = k_2 \cdot c_{A^-} \cdot c_C$$

$$AC^- \overset{k_3}{\rightarrow} A^- + C,\ r_3 = k_3 \cdot c_{AC^-}$$

$$AC^- + AH \overset{k_4}{\rightarrow} A^- + P,\ r_4 = k_4 \cdot c_{AC^-} \cdot c_{AH}$$

$$AC^- + BH^+ \overset{k_5}{\rightarrow} P + B,\ r_5 = k_5 \cdot c_{AC^-} \cdot c_{BH^+}$$

where r_i are reaction rates. The rate constants k_i correspond to the model parameters, fixed at their mean values: {49.7796, 8.9316, 1.3177, 0.3109, 3.8781}. The mass balance is as follows,

$$c_{AH}^0 - c_{AH} + \theta_1 \cdot (-r_1 - r_4) = 0$$

$$c_B^0 - c_B + \theta_1 \cdot (-r_1 + r_5) = 0$$

$$c_C^0 - c_C + \theta_1 \cdot (-r_2 + r_3) = 0$$

$$c_{A^-}^0 - c_{A^-} + \theta_1 \cdot (r_1 - r_2 + r_3 + r_4) = 0$$

$$c_{AC^-}^0 - c_{AC^-} + \theta_1 \cdot (r_2 - r_3 - r_4 - r_5) = 0$$

$$c_{BH^+}^0 - c_{BH^+} + \theta_1 \cdot (r_1 - r_5) = 0$$

$$c_P^0 - c_P + \theta_1 \cdot (r_4 + r_5) = 0$$

Two quality constraints are specifying that the conversion of C must be greater than 90%, and that the concentration of AC^- in the outlet must be less than 0.002,

$$\frac{c_C^0 - c_C - c_{AC^-}}{c_C^0} \geq 0.9$$

$$c_{AC^-} \leq 0.002$$

The initial concentrations $\{c_{AH}^0, c_B^0, c_C^0, c_{A^-}^0, c_{AC^-}^0, c_{BH^+}^0, c_P^0\}$ are set to $\{0.3955, 0.3955/\theta_2, 0.25, 0, 0, 0, 0\}$. The process parameters are the residence time θ_1 and the molar ratio θ_2.

$$400 \leq \theta_1 \leq 1400$$
$$10 \leq \theta_2 \leq 30$$

The KS parameter ρ is fixed to 10. Figure 15(a) shows all the feasible LHS points and updated boundary points, which can demonstrate the boundary of the KS space. Based on all the points,

the polynomial surrogate model with four degrees can be created.

$$\hat{y}_2(\bar{\theta}_1,\bar{\theta}_2) = 0.008668 - 0.070106\bar{\theta}_1 + 0.180528\bar{\theta}_1^2 - 0.193306\bar{\theta}_1^3 + 0.071808\bar{\theta}_1^4 - 1.627739\bar{\theta}_2 - 49.395899\bar{\theta}_1\bar{\theta}_2 + 212.180206\bar{\theta}_1^2\bar{\theta}_2 - 391.591254\bar{\theta}_1^3\bar{\theta}_2 + 230.359832\bar{\theta}_1^4\bar{\theta}_2 + 75.211453\bar{\theta}_2^2 - 364.868356\bar{\theta}_1\bar{\theta}_2^2 + 858.685883\bar{\theta}_1^2\bar{\theta}_2^2 - 532.802744\bar{\theta}_1^3\bar{\theta}_2^2 - 35.709792\bar{\theta}_1^4\bar{\theta}_2^2 - 38.784014\bar{\theta}_2^3 + 191.959125\bar{\theta}_1\bar{\theta}_2^3 - 902.219004\bar{\theta}_1^2\bar{\theta}_2^3 + 903.265210\bar{\theta}_1^3\bar{\theta}_2^3 - 155.198075\bar{\theta}_1^4\bar{\theta}_2^3 - 17.521365\bar{\theta}_2^4 + 153.291539\bar{\theta}_1\bar{\theta}_2^4 - 38.510764\bar{\theta}_1^2\bar{\theta}_2^4 - 103.476741\bar{\theta}_1^3\bar{\theta}_2^4 + 6.773545\bar{\theta}_1^4\bar{\theta}_2^4$$

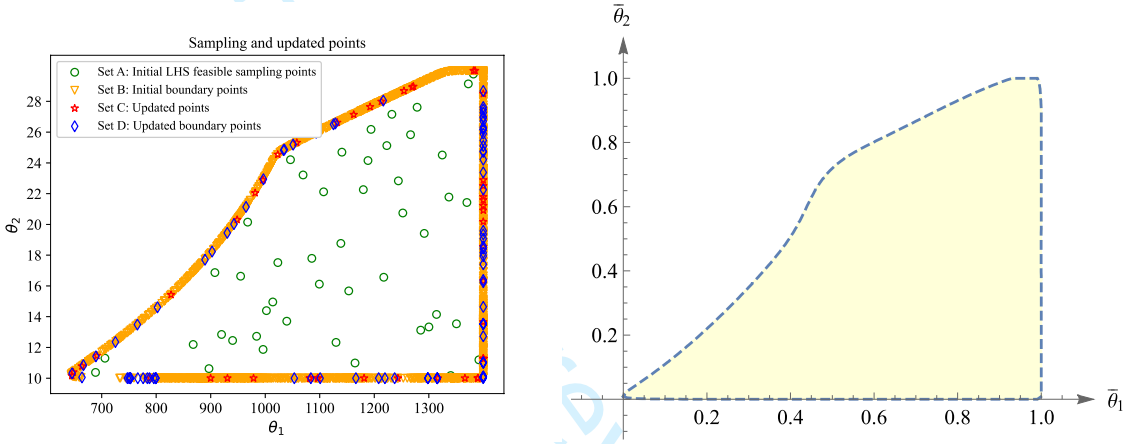


Figure 15. Results of design space description for Case 2. ($\rho = 10$, BARON)

The profile of the surrogate model $\hat{y}_2(\bar{\theta}_1,\bar{\theta}_2) = 0$ is shown in Figure 15(b). Because of the polynomial fitted error, the profile is slightly outside of the square. According to Equation (33), the CAD method can generate a triangular and explicit system as follows, where $\Gamma_2(\bar{\theta}_1, \#1)$ is provided in Supporting Information.

$$1:\begin{cases} 0 \le \bar{\theta}_1 \le 0.118986 \\ Root(\Gamma_2(\bar{\theta}_1, \#1)\&, 2) \le \bar{\theta}_2 \le Root(\Gamma_2(\bar{\theta}_1, \#1)\&, 3) \end{cases}$$

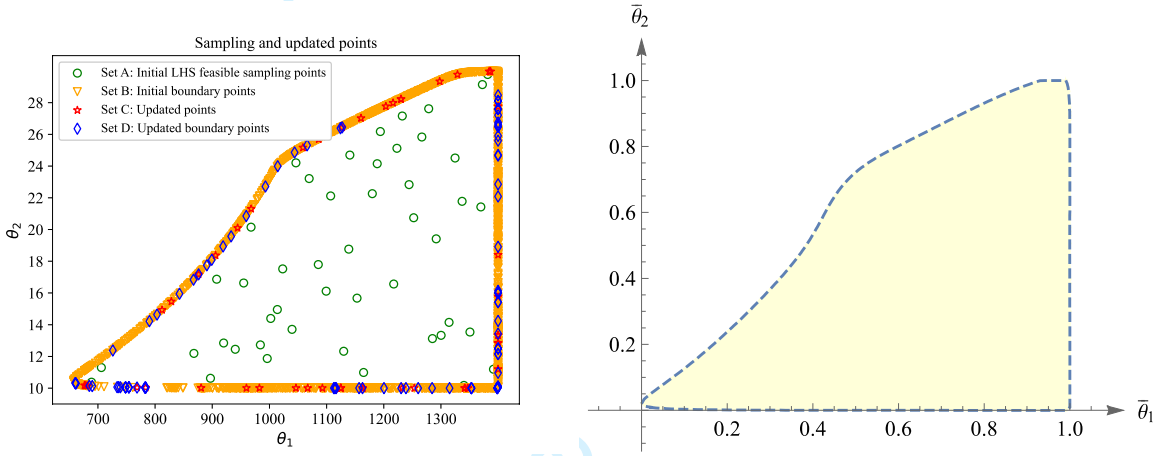
$$2:\begin{cases} 0.118986 < \bar{\theta}_1 < 0.228252 \\ Root(\Gamma_2(\bar{\theta}_1, \#1)\&,1) \le \bar{\theta}_2 \le Root(\Gamma_2(\bar{\theta}_1, \#1)\&,2) \end{cases}$$

$$3:\begin{cases} 0.228252 \le \bar{\theta}_1 \le 0.930122 \\ 0 \le \bar{\theta}_2 \le Root(\Gamma_2(\bar{\theta}_1, \#1)\&, 2) \end{cases}$$

$$4:\begin{cases} 0.930122 < \bar{\theta}_1 \le 0.990867 \\ 0 \le \bar{\theta}_2 \le 1 \end{cases}$$

$$5:\begin{cases} 0.990867 < \bar{\theta}_1 \le 1 \\ 0 \le \bar{\theta}_2 \le Root(\Gamma_2(\bar{\theta}_1, \#1)\&, 2) \end{cases}$$

In the data set of polynomial fitting, the maximum values of θ_1 and θ_2 are 1400 and 29.9977, and the minimum values are 644.084 and 10, respectively. Taking the third subspace as an example, a sampling point is chosen as $\bar{\boldsymbol{\theta}}_s$ = (0.6, 0.5) and $\boldsymbol{\theta}_s$ is (1097.63, 19.9989). $KS(\boldsymbol{\theta}_s) = -0.250481 < 0$ means that the subspace is feasible. Moreover, Figure 16 illustrates the results with $\rho = 5$. For this case, Figure 15(b) and Figure 16(b) are almost the same, which indicates that $\rho = 10$ can provide a good approximation of the feasible region.



(a) (b)

Figure 16. Results of design space description for Case 2. ($\rho = 5$, BARON)

Table 2. Summary of model information and computational expense.

	Model information		Proposed method			Use CAD method only
	Number of eqs / ineqs	Number of variables	ρ	Part 1: Adaptive sampling and surrogate modeling	Part 2: Explicit description	
Case 1	8 / 6	10	20	154.29 / 687.98 (BARON)	0.23 / 0.24	0.35 / 0.36
			5	89.81 / 578.12 (BARON)	0.26 / 0.29	
Case 2	13 / 6	15	10	352.68 / 2279.72 (BARON)	0.25 / 0.27	>7200 / >7200*
				251.68 / 476.16 (CONOPT4)	0.22 / 0.22	
			5	380.53/ 2701.63 (BARON)	0.25 / 0.25	
				265.71 / 498.40 (CONOPT4)	0.20 / 0.21	

*Cannot solve within 7200 seconds.

The computational time includes CPU time (s) / Wall time (s)

In Table 2, the model information of both cases and the computational times are summarized. Case 1 includes 8 equalities and 10 variables; Case 2 has a larger scale, involving 13 equalities

and 15 variables. The adaptive sampling and surrogate modeling are executed in Python, and the explicit description based on the CAD method is implemented in Mathematica. The CAD method transforms the surrogate model of KS function very fast. In addition, we also compare the computational time with using the CAD method only. The CAD method can directly solve Case 1 in 0.36s; however, Case 2 cannot be handled within 7200s, because of the larger scale. The proposed method can solve Case 2 in around 2500s when using the BARON solver.

7. Discussion

The proposed method consists of adaptive sampling and symbolic computation. There are some key points that can affect the results. To further clarify the performance of the proposed method, some discussion is provided in this section.

7.1. Values of critical parameters

The critical parameters mainly contain the initial LHS points, *num* and the parameter of the KS function, ρ. As shown in Figure 8 and Figure 11a, it is obvious that more initial LHS points are specified, the denser the boundary points can be found, and the more accurate the surrogate model will be. In addition, note that there may be empty spaces on the boundary, e.g., Figure 8. The main reasons are lack of feasible LHS points to generate the line projections on the boundary and/or lack of exploration during DFO search. The empty spaces can affect the polynomial fitting accuracy. In particular, if the empty area contains critical nonlinear features, the fitted polynomial will have a serious distortion. Thus, to reduce the empty areas, we can set a larger initial number of LHS points and a more stringent stopping criteria of adaptive sampling, including a larger iteration limit and a smaller termination error of K-fold cross validation.

Moreover, there are two main limitations for the KS function, i.e, the exponential functions can make the KS function strongly nonlinear; it is difficult to determine an appropriate value of ρ

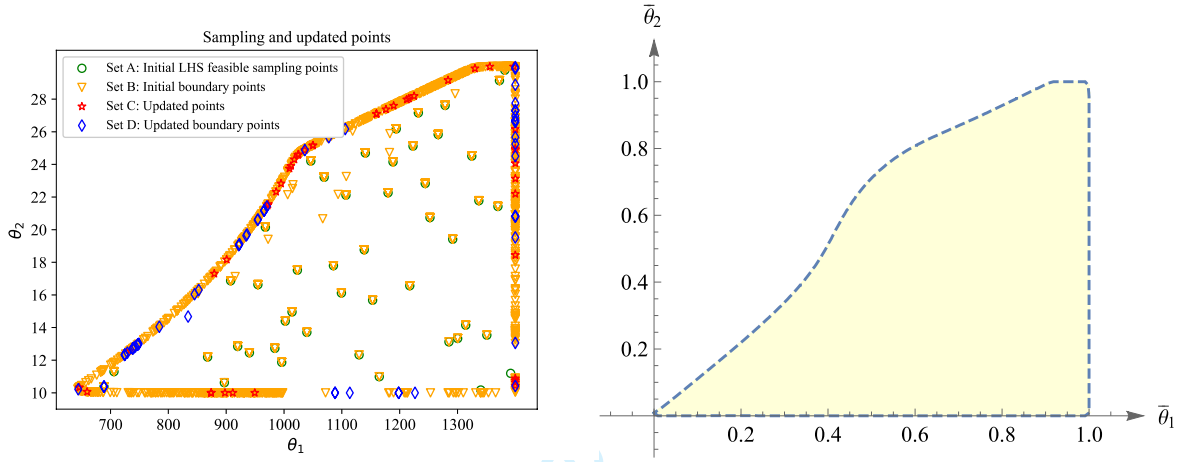
for a specific problem. The different values of ρ have been compared in the example and cases. A small value of ρ may lead to an overly conservative KS function, while too large a value of ρ may make the KS function ill-conditioned and cause unstable convergence and making it difficult to solve. Therefore, there is a tradeoff between accuracy of the KS function and ease of finding its surrogate model. A larger value of ρ can make the envelope closer to the real constrains, but possibly capture the undesired complex nonlinearity as well. It is not easy to create an accurate surrogate model for the KS function with complex or nonlinear structures. For simplicity, in this work, the value of ρ needs to be specified in advance.

7.2. Selection of NLP solvers

In the procedure of adaptive sampling, we propose two methods to expand the boundary points, as shown in Figure 7 and Figure 9. Because there must be two intersection points for a line and the boundary. The target is to locate both points; thus, the formula of the line should be defined by a parametric form, as shown in Equation (21). Both methods of expanding boundary points involve solving NLP models.

The real intersection points refer to the global optimal solutions, and common NLP solvers cannot guarantee finding the global optima. In the above cases, the global NLP solver, BARON, is adopted, and the results show that we can locate the boundary points effectively. Moreover, it is worth noting that, as the value of ρ and the number of inequalities increases, the KS function will be very complex and highly nonlinear, which will cause unstable convergence. Therefore, BARON is often time consuming. To ease the computational burden, it is necessary to compare the performance with a local solver. The local NLP solver in GAMS, CONOPT4, is selected to deal with Case 1. The results shown in Figure 17 indicate that this local NLP solver is also acceptable, and the design space can be described accurately. However, compared with Figure 15(a), Figure 17(a) shows that some points in set B overlap on the points in set A and some are located inside. The main reason is that the optimization model converges to the

local solutions. Nonetheless, we also can use these points to build the surrogate model, because these points are located within the design space. The CPU/Wall time for using CONOPT4 is 251.68s/476.16s, which is significantly less than using BARON. In addition, for an extremely complex and nonlinear model, BARON may require long computational times, and may not guarantee finding the global optimum for a fixed time limit. Therefore, there is also a tradeoff between the selection of solvers and the computational efficiency.



(a) (b)

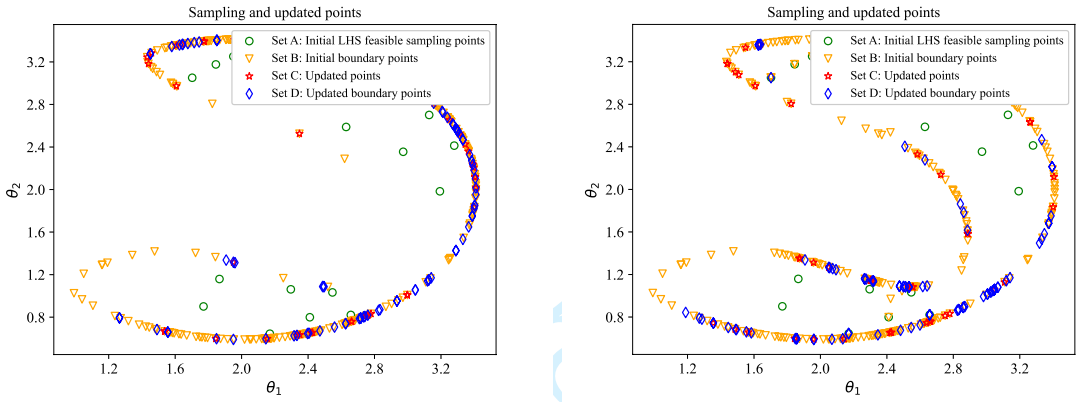
Figure 17. Results of design space description for Case 2. ($\rho = 10$, CONOPT4)

7.3. Effect of nonconvexity

The above numerical example and the Michael addition reaction case involve non-convex feasible spaces. Figure 2 indicates that the non-convex boundary can be described more accurately as the value of ρ increases. To further show the characteristics and performances of the proposed method, the example shown in Equation (6) is modified as Equation (37), which has a more nonconvex feasible region. The parameters are set to $\rho = 5, num = 80$.

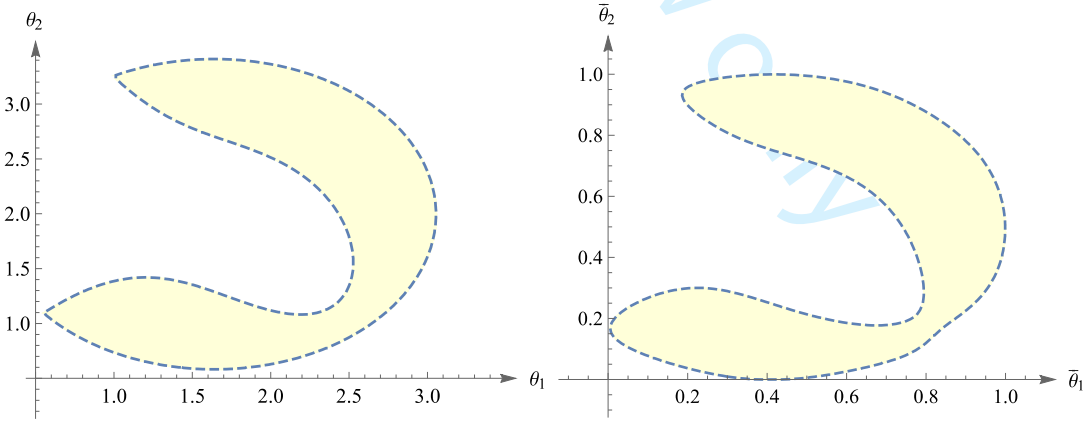
$$\begin{aligned} &g_1: -\left((\theta_2-2)^2+(\theta_1-2)^3+(\theta_2-2)(\theta_1-2)-0.5\right)\le 0\\ &g_2:(\theta_2-2)^2+(\theta_1-2)^2-2\le 0\\ &g_3:\theta_1-4\le 0\\ &g_4:-\theta_1\le 0\\ &g_5:\theta_2-4\le 0\\ &g_6:-\theta_2\le 0 \end{aligned} \tag{37}$$

In this case, BARON and CONOPT4 are compared. Figure 18(a) indicates that BARON cannot capture the inner nonconvex areas, because the solutions are always converged to the global maxima, i.e., the outermost boundary points. Thus, a large empty area on the boundary can be generated. By comparison, the local solver CONOPT4 can converge to local solutions, and the inner nonconvex areas can be described more completely, as shown in Figure 18(b). Based on these points, Figure 19 shows that the surrogate model can be created effectively. Therefore, the local solvers are more suitable for the problems with strongly nonconvex feasible regions.



(a) (b)

Figure 18. Sampling points of the modified example by using (a) BARON; (b) CONOPT4.



(a) (b)

Figure 19. Surrogate model of the modified example. (a) Real space; (b) KS space obtained by CONOPT4

8. Conclusions

In this paper, we propose a novel design space description method based on adaptive sampling and symbolic computation. The KS function is applied to aggregate all the inequality constraints, and the KS space can approximate the design space with a single constraint. Thus, we only need to focus on finding the sampling points on the boundary of the KS space. Based on the feasible LHS points and adaptive sampling points, two methods have been presented to effectively expand the set of boundary points. i.e., line projection to the boundary through any two feasible LHS points, and perturbation around the adaptive sampling points. The obtained polynomial surrogate model can be transformed into an equivalent triangular model through the CAD method, which can be further used to describe the KS space explicitly.

The case studies show that the proposed method is applicable to both convex and nonconvex feasible regions. Moreover, it is worth noting that, the CAD method is originally limited to polynomial functions, but the proposed method can address the inequality constraints with transcendental terms. This is because before performing the CAD method, the single KS function should be fitted as a multivariate polynomial by using the proposed adaptive sampling method. In addition, theoretically, the proposed method is more applicable to the cases with a small number of process parameters, because it is not easy to generate a single surrogate polynomial model in a high-dimensional space.

Acknowledgements

The authors gratefully acknowledge the financial support from Eli Lilly and Company, and the Center for Advanced Process Decision-making (CAPD) from Carnegie Mellon University.

Literature Cited

1 Hakemeyer C, McKnight N, John RS, Meier S, Trexler-Schmidt M, Kelley B, Zettl F, Puskeiler R, Kleinjans A, Lim F, Wurth, C. Process characterization and design space definition. *Biologicals*. 2016;44(5):306-318.

2 Kusumo KP, Gomoescu L, Paulen R, García-Muñoz S, Pantelides CC, Shah N, Chachuat B. Bayesian approach to probabilistic design space characterization: a nested sampling strategy. *Ind Eng Chem Res*. 2019;59(6):2396-2408.

3 Huang J, Kaul G, Cai C, Chatlapalli R, Hernandez-Abad P, Ghosh K, Nagi A. Quality by design case study: an integrated multivariate approach to drug product and process development. *Int J Pharm*. 2009;382(1-2):23-32.

4 Garcia-Munoz S, Luciani CV, Vaidyaraman S, Seibert KD. Definition of design spaces using mechanistic models and geometric projections of probability maps. *Org Process Res Dev*. 2015;19(8):1012-1023.

5 Goyal V, Ierapetritou MG. Determination of operability limits using simplicial approximation. *AIChE J*. 2002;48(12):2902-2909.

6 Rogers A, Ierapetritou MG. Feasibility and flexibility analysis of blackbox processes part 1: surrogate-based feasibility analysis. *Chem Eng Sci*. 2015;137:986-1004.

7 Rogers A, Ierapetritou MG. Feasibility and flexibility analysis of blackbox processes part 2: surrogate-based flexibility analysis. *Chem Eng Sci*. 2015;137:1005-1013.

8 Wang Z, Ierapetritou M. A novel feasibility analysis method for blackbox processes using a radial basis function adaptive sampling approach. *AIChE J*. 2017;63(2):532-550.

9 Metta N, Ramachandran R, Ierapetritou M. A novel adaptive sampling based methodology for feasible region identification of compute intensive models using artificial neural network. *AIChE J*. 2021;67(2):e17095.

10 Laky D, Xu S, Rodriguez JS, Vaidyaraman S, García-Muñoz S, Laird C. An optimization-based framework to define the probabilistic design space of pharmaceutical processes with model uncertainty. *Processes*, 2019;7(2):96.

11 Zhang Q, Grossmann IE, Lima RM. On the relation between flexibility analysis and robust optimization for linear systems. *AIChE J*. 2016;62(9):3109-3123.

12 Halemane KP, Grossmann IE. Optimal process design under uncertainty. *AIChE J*. 1983;29(3):425-433.

13 Swaney RE, Grossmann IE. An index for operational flexibility in chemical process design. Part I: Formulation and theory. *AIChE J*. 1985;31(4):621-630.

14 Swaney RE, Grossmann IE. An index for operational flexibility in chemical process design. Part II: Computational algorithms. *AIChE J*. 1985;31(4):631-641

15 Grossmann IE, Floudas CA. Active constraint strategy for flexibility analysis in chemical processes. *Comput Chem Eng*. 1987;11(6): 675-693.

16 Pulsipher JL, Zavala VM. A mixed-integer conic programming formulation for computing the flexibility index under multivariate gaussian uncertainty. *Comput Chem Eng*. 2018;119: 302-308.

17 Grossmann IE, Calfa BA, Garcia-Herreros P. Evolution of concepts and models for quantifying resiliency and flexibility of chemical processes. *Comput Chem Eng*. 2014;70:22-34.

18 Harwood SM, Barton PI. How to solve a design centering problem. *Math Methods Oper Res*. 2017;86(1):215-254.

19 Still G. Generalized semi-infinite programming: Theory and methods. *Eur J Oper Res*. 1999;119(2):301-313.

20 Vázquez FG, Rückmann JJ, Stein O, Still G. Generalized semi-infinite programming: a tutorial. *J Comput Appl Math*. 2008;217(2):394-419.

21 Zhao F, Chen X. Analytical and triangular solutions to operational flexibility analysis using quantifier elimination. *AIChE J*. 2018;64(11):3894-3911.

22 Collins GE. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. *Lect Notes Comput Sci*. 1975;33:134-183.

23 Zhao F, Zheng C, Zhang S, Zhu L, Chen X. Quantification of process flexibility via space projection. *AIChE J.* 2019;65(10):e16706.

24 Zheng C, Zhao F, Zhu L, Chen X. Operational flexibility analysis of high-dimensional systems via cylindrical algebraic decomposition. *Ind Eng Chem Res*. 2020;59(10):4670-4687.

25 Collins GE. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. *Lec Notes Comp Sci*. 1975;33:134-183.

26 Kreisselmeier G, Steinhauser R. Systematic control design by optimizing a vector performance index. Proc. *IFAC Symp. On Computer-Aided Design of Control Systems*. Zurich, Switzerland, 1979.

27 Rooney WC, Biegler LT. Optimal process design with model parameter uncertainty and process variability. *AIChE J.* 2003;49(2):438-449.

28 Martins JRRA, Alonso JJ, Reuther JJ. High-fidelity aerostructural design optimization of a supersonic business jet. *J Aircr*. 2004;41(3):523-530.

29 Brooks TR, Martins JRRA, Kennedy GJ. High-fidelity aerostructural optimization of tow-steered composite wings. *J Fluids Struct*. 2019;88:122-147.

30 Gelfand IM, Kapranov MM, Zelevinsky AV. Discriminants, resultants, and multidimensional determinants. Birkhäuser: Theory & Applications, 1994.

31 Sagraloof M, Mehlhorn K. Computing real roots of real polynomials. *J Symb Comput*. 2016; 73:46-86.

32 Wolfram Research Inc. Mathematica Version 12.2. Champaign, IL. 2020.

33 Deutsch JL, Deutsch CV. Latin hypercube sampling with multidimensional uniformity. *J Stat Plan Inference*. 2012;142(3):763-772.

34 Zhang P. Model selection via multifold cross validation. *Ann stat*. 1993;21(1):299-313.

35 Powell MJD. The BOBYQA Algorithm for Bound Constrained Optimization Without Derivatives. Technical report, Department of Applied Mathematics and Theoretical Physics,

University of Cambridge. 2009.

36 Hart WE, Watson JP, Woodruff DL. Pyomo: modeling and solving mathematical programs in Python. *Math Program Comput.* 2011;3(3):219-260.

37 Chen W, Biegler LT, García-Muñoz S. An approach for simultaneous estimation of reaction kinetics and curve resolution from process and spectral data. *J Chemom*. 2016;30(9):506-522.

Design Space Description through Adaptive Sampling and Symbolic Computation

Fei Zhao, Ignacio E. Grossmann*

Center for Advanced Process Decision-Making, Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213

Salvador García Muñoz, Stephen D. Stamatis

Synthetic Molecule Design and Development, Lilly Research Laboratories, Indianapolis, IN 46285

Abstract

Design space definition is one of the key parts in pharmaceutical research and development. In this article, we propose a novel solution strategy to explicitly describe the design space without recourse decisions. First, to smooth the boundary, the Kreisselmeier-Steinhauser (KS) function is applied to aggregate all inequality constraints. Next, for creating a surrogate polynomial model of the KS function, we focus on finding sampling points on the boundary of KS space. After performing Latin hypercube sampling (LHS), two methods are presented to efficiently expand the boundary points, i.e., line projection to the boundary through any two feasible LHS points and perturbation around the adaptive sampling points. Finally, a symbolic computation method, cylindrical algebraic decomposition, is applied to transform the surrogate model into a series of explicit and triangular subsystems, which can be converted to describe the KS space. Two case studies show the efficiency of the proposed algorithm.

Keywords: process design; design space; adaptive sampling; symbolic computation; cylindrical algebraic decomposition

*Correspondence concerning this article should be addressed to grossmann@cmu.edu.

1. Introduction

In the pharmaceutical industry, process parameters[1] correspond to degrees of freedom or variables that can be manipulated in the operation of a manufacturing process, and which can be measured and set within the controller tolerance for a desired value. Design space is defined as "multidimensional combination and interaction of input variables and process parameters that have been demonstrated to provide assurance of quality"[2]. In other words, Product quality is maintained as long as the process parameters are controlled within the design space.

Early approaches to identify the design space were solely based on experiments and empirical functions[3]. By performing extensive experiments, the relationships of process parameters and critical quality attributes (CQAs) can be built through regression and the process parameters that have medium/high impacts on the CQAs can be determined. The design space is visualized by response surface modeling and further verified by additional experiments[4]. This method requires extensive experiments, and it is very time-consuming and expensive. To lower the cost of developing design spaces, mechanistic models that contain relationships of process parameters and CQAs can be formulated in advance and parametrized with less data. Goyal and Ierapetritou[5] proposed an approach based on outer-approximation to identify the operating envelopes where process operation is feasible, safe, and profitable. In addition, in order to address the computationally expensive models, the surrogate-based methods are then proposed. Rogers and Ierapetritou[6,7] applied Kriging as the surrogate models to approximate the original functions and identify the design spaces with limited samplings. Compared with the kriging surrogate models, Wang and Ierapetritou[8] used RBF surrogate models. Metta et al.[9] proposed to use an artificial neural network to create the surrogate models for addressing problems that are computationally expensive or do not have constraints in closed form.

Moreover, optimization approaches based on mechanistic models have been extensively studied to describe the design space[10,11]. Characterizing a design space for a process design

model is analogous to the flexibility index problem in the chemical industry[12,13,14]. The flexibility index is used to describe an operational range, which represents a maximum scaled departure of all process parameters from the given nominal conditions. It is worth stating that "design space" and "feasible space" are interchangeably used in the pharmaceutical industry. Generally, it is not easy to accurately describe the boundary of the real design space because of the high nonlinearity. The flexibility index can approximate the design space by a largest inscribed subspace with a specific shape, which may be a rectangle, ellipse, or other shapes. Because we are only concerned with this subspace, for simplicity, this subspace is denoted as "design space" in this paper. When approximating it as a rectangle, the vertex direction search method[13] can be employed to find the flexibility index, which is rigorous for convex regions. To avoid the convexity assumption, Grossmann and Floudas[15] developed an active constraint strategy, where the two-level optimization formulation for the flexibility index problem can be reformulated as a mixed-integer linear or nonlinear programming model by applying the KKT conditions to the inner optimization problem. In addition, Pulsipher and Zavala[16] proposed to use an ellipsoidal set to capture correlations of process parameters, as well as a mixed-integer conic programming formulation to compute the flexibility index. A number of approaches are proposed to quantify system flexibility, and an extensive review is provided by Grossmann et al. [17] If the nominal conditions of the process parameters are unknown, the flexibility index problem can be extended to the design centering problem[18], which focuses on determining the nominal conditions that maximize the size of the design space. From a mathematical view, the design centering problem is a generalized semi-infinite programming problem[19,20]. Flexibility index and design centering are two complementary ways for estimating a candidate design space, which have been widely studied in recent decades; however, both methods need to specify the shape of design space in advance, which is quite hard to reflect the reality of the feasible region, especially for nonconvex cases.

In the absence of models capable of efficiently probing the fullest extent of the feasible region, design spaces were obligated to be defined with an assumption of shape to be probed experimentally. With a robust model, there is no reason to make the shape assumption because we can effectively probe the boundaries to the extent of the feasible region. The challenge is thus computing and defining that boundary. Zhao and Chen[21] first proposed representing the design model as an existential quantifier formula and applied the cylindrical algebraic decomposition (CAD) method[22] to accurately describe the design space and explicitly express the relationships between uncertain parameters. The CAD method can provide a complete description of the design space (in this case, the design space is identical to the feasible region), and the triangular structure makes possible the explicit algebraic representation of the bounds of each process parameter. The method is suitable for convex and nonconvex systems described by polynomials. Zhao et al.[23] proposed a space projection method based on the CAD method to deal with flexibility index problems. Due to the heavy computational burden of the CAD method, the above methods are only applicable to relatively small-scale problems. For high-dimensional systems consisting of a large number of equalities and limited inequalities, Zheng et al.[24] proposed to build a surrogate model to correlate the inequality constraints based on an initial sample set. The design space is explicitly expressed via the CAD method, and the boundary can be checked to iteratively refine the CAD results. However, if the design space has a severely irregular shape, the computational burden of the CAD method will be very high. Moreover, since it is not appropriate to reduce process parameters, the number of inequality constraints becomes another key factor of the computational complexity.

In this work, we propose a novel design space description method based on efficient adaptive sampling and symbolic computation, and in which no recourse is considered for the realization of the parameters. The proposed method not only can eliminate all the equality constraints and state variables, but also significantly reduce the sampling burden, and decrease the number of

inequality constraints to one. In addition, there is a tradeoff between the complexity of the CAD method and the accuracy of design space description. The rest of this article is organized as follows. Sections 2 provides problem statements including the research framework. Section 3 defines the design space based on the Kreisselmeier-Steinhauser (KS) function. Section 4 illustrates the main steps of the CAD method. Section 5 provides an adaptive sampling strategy to find the explicit expression of the design space. Two case studies are provided in Section 6 to illustrate the proposed methods. Section 7 discusses the key characteristics of the proposed method. Section 8 concludes the paper.

2. Problem statement

An ultimate goal of the design space definition problem is to accurately and explicitly describe the design space, regardless of whether it is a convex and nonconvex space. For a given design model with no recourse[12], the feasibility of the model can be described as:

$$\forall \boldsymbol{\theta} \in DS(\boldsymbol{\theta})\{\forall j \in J[g_j(\boldsymbol{\theta},\boldsymbol{x}) \leq 0], \forall i \in I[h_i(\boldsymbol{\theta},\boldsymbol{x}) = 0]\} \quad (1)$$

where $\boldsymbol{\theta}$ and $\boldsymbol{x}$ are process parameters and state variables, respectively. Equation (1) states that for any possible realization of process parameters in the design space, denoted as $DS(\boldsymbol{\theta})$, all the individual constraints should be satisfied. In other words, $DS(\boldsymbol{\theta})$ can also be defined as

$$DS(\boldsymbol{\theta}) := \left\{ \boldsymbol{\theta} \in \mathbb{R}^P \middle| \begin{bmatrix} g_j(\boldsymbol{\theta},\boldsymbol{x}) \leq 0, & \forall j \in J \wedge \\ h_i(\boldsymbol{\theta},\boldsymbol{x}) = 0, & \forall i \in I \wedge \\ \boldsymbol{\theta}^L \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}^U & \end{bmatrix} \right\} \quad (2)$$

$DS(\boldsymbol{\theta})$ represents the entire feasible region of process parameters. It is generally difficult to describe it analytically because of the nonlinearities of the design model. Based on previous work[21], the CAD method can equivalently transform an inequality system to a triangular system, and the upper and lower bounds of each process parameter can be expressed explicitly. For high-dimensional cases, the equations and state variables can be eliminated through surrogate models for the inequality constraints[24], which can reduce the computational burden of the CAD method. Since it needs to sample points over the whole design space, this method must sample

enough points and take many iterations to accurately capture the profile of the design space.

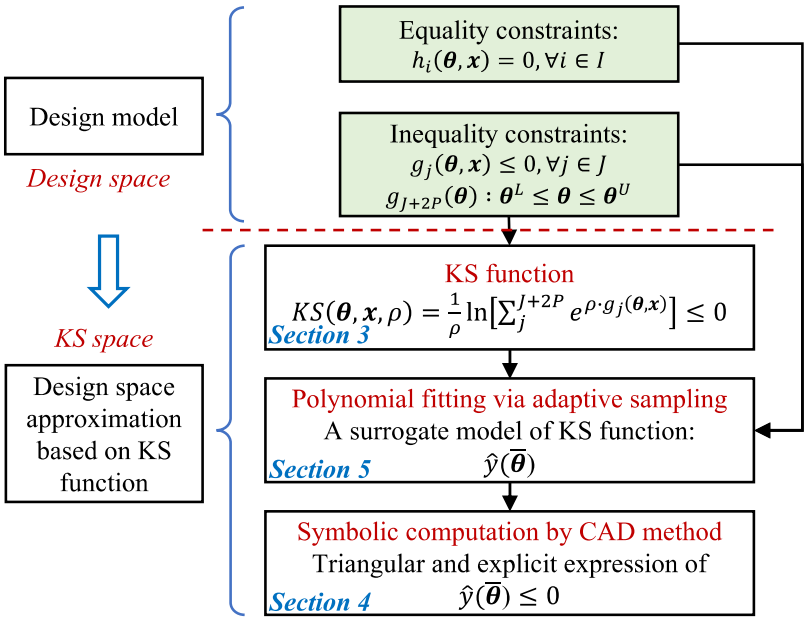


Figure 1. Conceptual computational framework.

In order to further ease the computational burden of the CAD method and improve the efficiency of adaptive sampling and surrogate modeling, a novel solution strategy is proposed in this work to explicitly describe the design space. The research framework is shown in Figure 1, which can be interpreted as follows.

(1) For a given design model, the KS function can aggregate all inequality constraints, and then an underestimate of the desired design space, denoted as *KS space*, can be described by a single inequality constraint. The boundary of the KS space is continuously differentiable.

(2) The KS function is a transcendental function. To be able to process the KS function with the CAD method, an adaptive sampling strategy is proposed to create a polynomial surrogate model of the KS function. Two methods are presented to efficiently expand the boundary points of the KS space. i.e., line projection points through any two feasible LHS points, and perturbation boundary points around the adaptive sampling points. The cross-validation method is applied to evaluate the stopping criteria of the adaptive sampling.

(3) The CAD method is applied to triangulate the polynomial surrogate model. A simple

checking rule is applied to evaluate the feasibility of the CAD result.

Based on the above conceptual computational framework, we can point out two motivations:

(1) The complexity of the original CAD method[25] is formulated as

$$O\left(d_1{}^{2^{2n+8}} m^{2^{n+6}}\right) \tag{3}$$

which indicates that the complexity grows doubly exponentially with the number of variables n; d_1 is the maximum degree in any one variable in the original model; m is the number of polynomials. After applying the KS function to aggregate all inequality constraints and create its polynomial surrogate model, the complexity can be reduced to Equation (4),

$$O\left(d_2{}^{2^{2n+8}}\right) \tag{4}$$

where m has been reduced to one; d_2 is the maximum degree in any one variable in the surrogate model. While still doubly exponential in the number of variables, if the fitted degree, d_2, is not too large, the complexity is acceptable; thus, there is a tradeoff between the complexity of the CAD method and the accuracy of design space description.

(2) According to Equation (2), the equalities are used to represent the process model, and the state variables have the same dimension as equality constraints, i.e., $\dim(\boldsymbol{x}) = |I|$. Once the value of $\boldsymbol{\theta}$ is specified, we can run the simulation and obtain the results of $\boldsymbol{x}$. The inequality constraints define the quality requirements of the process design problems. If we sample the points in the space of $\boldsymbol{\theta}$, the results of $\boldsymbol{x}$ in the equalities can be used to evaluate the inequality constraints. Thus, in this work, all the equality constraints and state variables can be eliminated when generating the surrogate model of the KS function. Moreover, since the KS space must be contained within the original design space, we only need to focus on finding the points on the boundary of the KS space. The intention is to locate as many boundary points as possible, which can significantly reduce the sampling burden.

In summary, approximating the design space by the KS space, the proposed method can ease the sampling burden of surrogate modeling, reduce the computational expense of the CAD

method, and the assumptions for the shape of design space are not required.

3. Design space approximation through KS function

The KS function was first proposed by G. Kreisselmeier and R. Steinhauser[26], which was initially presented for controller design. In the last two decades, the KS function has been widely used in constraint aggregation methods for gradient-based optimization, e.g., chemical process design[27], as well as problems involving local stress constraints, e.g., aircraft design[28,29]. The KS function shown in Equation (5) can aggregate a set of inequality constraints into a single function, and it only contains one parameter ρ.

$$KS(\boldsymbol{\theta},\rho) = \frac{1}{\rho}\ln\left[\sum_{j}^{J} e^{\rho \cdot g_j(\boldsymbol{\theta})}\right] \leq 0 \tag{5}$$

where $g_j(\boldsymbol{\theta}) \leq 0$ are inequality constraints, $j \in J$. The KS function produces an envelope surface that is continuous and represents a conservative estimate of the feasible region for a set of constraints. $\rho > 0$ is an aggregation parameter defined by the user, which can control how close the envelope is to the original constraints. In the following nonlinear and nonconvex example, the KS function as a constraint aggregation method, and the effect of increasing ρ for inequality constraints, can be visualized.

An illustrative example

Consider the inequalities,

$$\begin{aligned} &g_1{:}(\theta_2-2)^2+(\theta_1-2)^3+(\theta_2-2)(\theta_1-2)-0.5\leq 0\\ &g_2{:}(\theta_2-2)^2+(\theta_1-2)^2-2\leq 0\\ &g_3{:}\theta_1-4\leq 0\\ &g_4{:}-\theta_1\leq 0\\ &g_5{:}\theta_2-4\leq 0\\ &g_6{:}-\theta_2\leq 0 \end{aligned} \tag{6}$$

For the design space definition problem, θ_1 and θ_2 are regarded as the process parameters. The feasible region of these inequality constraints, which can be denoted as a complete design space, is depicted by the yellow region shown in Figure 2(a). According to Equation (5), the KS

function of $g_1,\ldots,g_6$ is formulated as

$$KS(\boldsymbol{\theta},\rho) = \frac{1}{\rho}\ln\left[e^{\rho \cdot g_1} + e^{\rho \cdot g_2} + e^{\rho \cdot g_3} + e^{\rho \cdot g_4} + e^{\rho \cdot g_5} + e^{\rho \cdot g_6}\right] \tag{7}$$

which generates an underestimate profile of the design space after fixing ρ. As ρ increases, the profile depicted by the KS function can approach in the limit the true profile of the feasible region. As shown in Figure 2(b), when $\rho = 5$, the profile of the KS function is continuous and almost coincides with the real boundary of the feasible region. However, the profile for $\rho = 2$ is smoother, although less accurate.

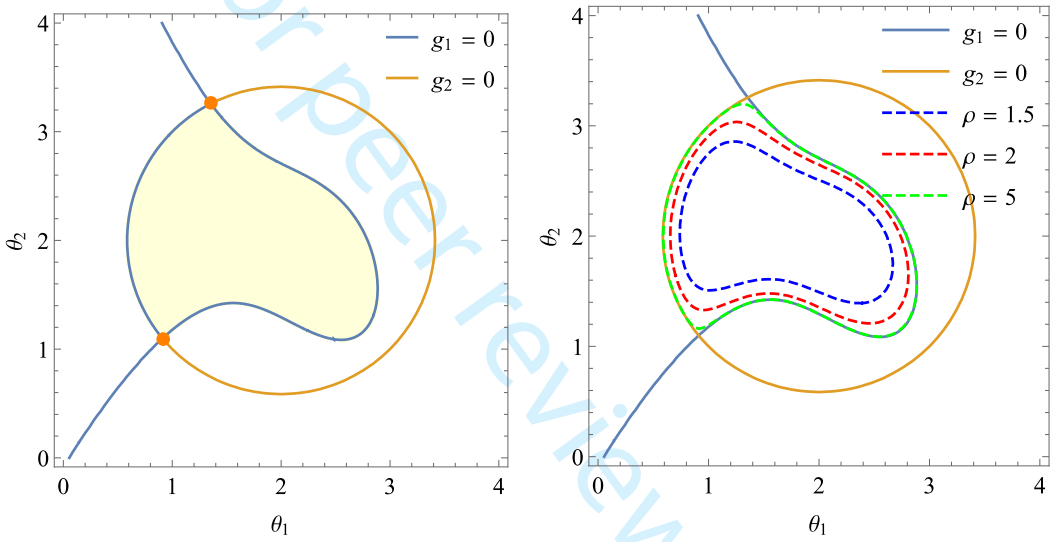


Figure 2. Feasible region of the example and profiles of KS function with increasing ρ.

The KS function can be applied to approximate the design space. Hence, the problem of approximately describing a design space can be transformed into describing the corresponding KS space. The major motivations for using the KS function can be stated as follows:

(1) The KS function can aggregate multiple inequality constraints, and the space can be depicted by a single inequality constraint. Since general constraints can be handled, the KS function is also applicable to non-convex design spaces.

(2) The KS function only involves one parameter $\rho > 0$, which is a scaling factor of the space.

As ρ increases, the KS function can provide a closer approximation to the design space, i.e.,

$$\{\boldsymbol{\theta} \in \mathbb{R}^n | KS(\boldsymbol{\theta},\rho_1) \leq 0\} \subset \{\boldsymbol{\theta} \in \mathbb{R}^n | KS(\boldsymbol{\theta},\rho_2) \leq 0\}, \quad 0 < \rho_1 < \rho_2 \tag{8}$$

In the limit, when ρ tends to infinity, the KS function can exactly represent the design space, i.e.,

$$\left\{\boldsymbol{\theta} \in \mathbb{R}^n | \lim_{\rho \to \infty} KS(\boldsymbol{\theta},\rho) \leq 0\right\} = \{\boldsymbol{\theta} \in \mathbb{R}^n | \forall j \in J[g_j(\boldsymbol{\theta}) \leq 0]\} \tag{9}$$

(3) The KS function is continuously differentiable. The KS function can smooth the profile and reduce most irregular regions, e.g., removing the intersection points in the original design space. As shown in Figure 2(a), the design space contains two orange intersection points. However, the smooth profiles of the KS function in Figure 2(b) have no intersection points. For $\rho_2 = 5$ and $\rho_1 = 2$, the profile of $KS(\boldsymbol{\theta},\rho_1)$ is smoother than the profile of $KS(\boldsymbol{\theta},\rho_2)$.

According to the KS function, the design model in Equation (2) can be reformulated as follows,

$$\begin{aligned} &KS(\boldsymbol{\theta},\boldsymbol{x},\rho) = \frac{1}{\rho}\ln\left[\sum_j^J e^{\rho \cdot g_j(\boldsymbol{\theta},\boldsymbol{x})} + \sum_p^P \left(e^{\rho \cdot (\theta_p^L - \theta_p)} + e^{\rho \cdot (\theta_p - \theta_p^U)}\right)\right] \leq 0 \\ &h_i(\boldsymbol{\theta},\boldsymbol{x}) = 0, \quad \forall i \in I \end{aligned} \tag{10}$$

where $KS(\boldsymbol{\theta},\boldsymbol{x}) \leq 0$ represents the KS space for a fixed value of ρ.

4. Explicit expression of design space through symbolic computation

Generally, the design space is a bounded and closed space. In previous work[21], if a design space is formulated by a polynomial system, the CAD method can transform this polynomial system into a series of triangular subsystems, where each subsystem corresponds to a subspace. All the subspaces define the entire design space. For the design model shown in Equation (2), the formulation of each triangular subsystem is as follows.

$$\begin{gathered} k_1 \le \theta_1 \le q_1 \\ k_2(\theta_1) \le \theta_2 \le q_2(\theta_1) \\ \vdots \\ k_P(\theta_1,\ldots,\theta_{P-1}) \le \theta_P \le q_P(\theta_1,\ldots,\theta_{P-1}) \\ k_{P+1}(\theta_1,\ldots,\theta_{P-1},\theta_P) \le x_1 \le q_{P+1}(\theta_1,\ldots,\theta_{P-1},\theta_P) \\ \vdots \\ k_{P+N}(\theta_1,\ldots,\theta_P,x_1,\ldots,x_{N-1}) \le x_N \le q_{P+N}(\theta_1,\ldots,\theta_P,x_1,\ldots,x_{N-1}) \end{gathered} \tag{11}$$

All the lower and upper bounds of the process parameters are explicit expressions. For a given $(\theta_1,\ldots,\theta_p)$, the lower and upper bound of θ_{p+1}, i.e., $k_{p+1}(\theta_1,\ldots,\theta_p)$ and $q_{p+1}(\theta_1,\ldots,\theta_p)$, become constants.

The CAD method mainly contains two stages: projection and lifting. In the projection stage, the key point is to calculate discriminants and resultants[30] of polynomials, which can find the tangency and intersection points in each dimension. In this way, the space in each dimension can be decomposed by using these points. For the example in Equation (6), as shown in Figure 3(a), if the triangular structure is set as $\theta_1 \prec \theta_2$, the green tangency points and orange intersection points can be calculated by

$$\begin{aligned} &\text{discriminant}(g_1,\theta_2) = 38 - 52\theta_1 + 25\theta_1^2 - 4\theta_1^3, \\ &\text{discriminant}(g_2,\theta_2) = -4(2 - 4\theta_1 + \theta_1^2), \\ &\text{resultant}(g_1,g_2,\theta_2) = 0.25 * (473 - 1440\theta_1 + 1700\theta_1^2 - 1012\theta_1^3 + \\ &\qquad 328\theta_1^4 - 56\theta_1^5 + 4\theta_1^6) \end{aligned} \tag{12}$$

Equation (12) are univariate polynomials of θ_1, so that θ_2 is eliminated. By performing the real root isolation algorithm[31], five distinct real roots of θ_1 for these polynomials can be obtained.

$$0.585786 < 0.910159 < 1.35389 < 2.88639 < 3.41421$$

Figure 3(a) illustrates that the tangency and intersection points can be projected onto the θ_1-axis, i.e., A_1: (0.585786, 0), A_2: (0.910159, 0), A_3: (1.35389, 0), A_4: (2.88639, 0), A_5: (3.41421, 0). The boundary between any two adjacent projection points is continuously differentiable. Based on these projection points, the entire two-dimensional (2D) space can be decomposed into six cylindrical 2D subspaces, i.e., [0, A_1], [A_1, A_2], [A_2, A_3], [A_3, A_4], [A_4, A_5], [A_5, 4]. In the lifting stage, these cylindrical 2D subspaces should be checked successively. Taking the

subspace on [A_1, A_2] as an example, in Figure 3(b), the steps of the lifting stage are as follows.

(1) C_1 is the midpoint of A_1 and A_2. After substituting C_1, g_1 and g_2 can be converted to

$$\begin{matrix}\overline{g}_1\text{:} -2.46264 - 1.25203 * (-2+\theta_2) + (-2+\theta_2)^2 \\ \overline{g}_2\text{:} -0.4324275990814064 + (-2+\theta_2)^2\end{matrix} \quad (13)$$

Thus, the intersections with g_1 and g_2 on the vertical line can be solved, i.e., B_1: (0.747973, 0.936478), B_2: (0.747973, 1.34241), B_3: (0.747973, 2.65759).

(2) Based on B_1, B_2, B_3 and the corresponding curves, the 2D cylindrical subspace on [A_1, A_2] can be decomposed into four parts, which are marked by different colors in Figure 3(b).

(3) On the vertical line, four sampling points in four parts can be determined, i.e., C_1, C_2, C_3 and C_4, where C_2, C_3 and C_4 are the midpoints of [B_1, B_2], [B_2, B_3] and [B_3, 4], respectively.

(4) C_1, C_2, C_3 and C_4 are used to check the feasibility of all the subspaces. When substituting C_1 to g_1 and g_2, we can get ($g_1 > 0$, $g_2 > 0$), which means that the subspace including C_1 is infeasible, because they cannot meet the original signs, $g_1 \leq 0$, $g_2 \leq 0$. Similarly, for C_2, C_3 and C_4, we can obtain ($g_1 < 0$, $g_2 > 0$), ($g_1 < 0$, $g_2 < 0$) and ($g_1 < 0$, $g_2 > 0$), respectively, and only the yellow subspace including C_3 is feasible. Therefore, it is a part of the entire feasible region.

(5) As shown in Figure 3(c), on [A_1, A_2], the upper boundary of the feasible subspace is (D_1, D_3) and the lower boundary is (D_1, D_2). The coordinate of B_3 is (0.747973, 2.65759). In Equation (13), 2.65759 is the second root of $\overline{g}_2 = 0$; thus, for any value of $\overline{\theta}_1$ in [A_1, A_2], the corresponding values of θ_2 on (D_1, D_3) are the second roots of $g_2(\overline{\theta}_1,\theta_2) = 0$, denoted as $Root(g_2\&, 2)$. Similarly, the values of θ_2 on lower boundary (D_1, D_2) are the first roots of $g_2(\overline{\theta}_1,\theta_2) = 0$, i.e., $Root(g_2\&, 1)$. Therefore, this subregion can be expressed as

$$\begin{cases}0.585786 \leq \theta_1 \leq 0.910159 \\ Root(g_2\&, 1) \leq \theta_2 \leq Root(g_2\&, 2)\end{cases} \quad (14)$$

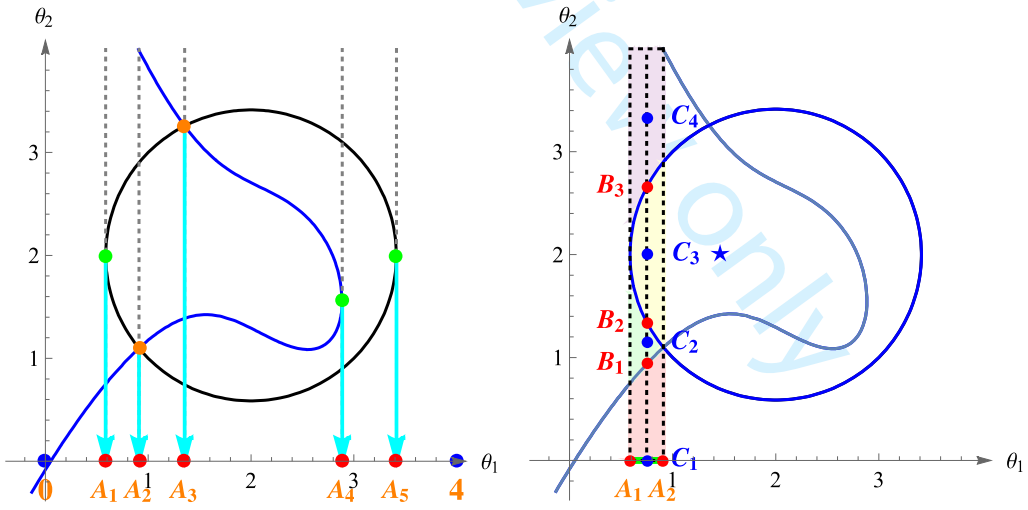
which is a triangular and explicit expression of this feasible subspace. Moreover, since g_2

is a bivariate polynomial, θ_2 can be solved explicitly, and Equation (14) is reformulated as,

$$\begin{cases} 0.585786 \le \theta_1 \le 0.910159 \\ 2 - \sqrt{-2 + 4\theta_1 - \theta_1^2} \le \theta_2 \le 2 + \sqrt{-2 + 4\theta_1 - \theta_1^2} \end{cases} \tag{15}$$

After checking all the cylindrical subspaces, as shown in Figure 3(d) and Figure 3(e), another two feasible subspaces can be found. Finally, we know that the feasible region consists of three subregions, as shown in Figure 3(f). Equation (16) shows the complete triangular and explicit expressions of the feasible region. The command *CylindricalDecomposition* in Mathematica[32] can perform the CAD method.

$$\begin{aligned} 1.&\begin{cases} 0.585786 \le \theta_1 \le 0.910159 \\ 2 - \sqrt{-2 + 4\theta_1 - \theta_1^2} \le \theta_2 \le 2 + \sqrt{-2 + 4\theta_1 - \theta_1^2} \end{cases} \\ 2.&\begin{cases} 0.910159 < \theta_1 \le 1.35389 \\ 0.5(6 - \theta_1) - 0.5\sqrt{38 - 52\theta_1 + 25\theta_1^2 - 4\theta_1^3} \le \theta_2 \le \\ 2 + \sqrt{-2 + 4\theta_1 - \theta_1^2} \end{cases} \\ 3.&\begin{cases} 1.35389 < \theta_1 \le 2.88639 \\ 0.5(6 - \theta_1) - 0.5\sqrt{38 - 52\theta_1 + 25\theta_1^2 - 4\theta_1^3} \le \theta_2 \le \\ 0.5(6 - \theta_1) + 0.5\sqrt{38 - 52\theta_1 + 25\theta_1^2 - 4\theta_1^3} \end{cases} \end{aligned} \tag{16}$$



(a) (b)

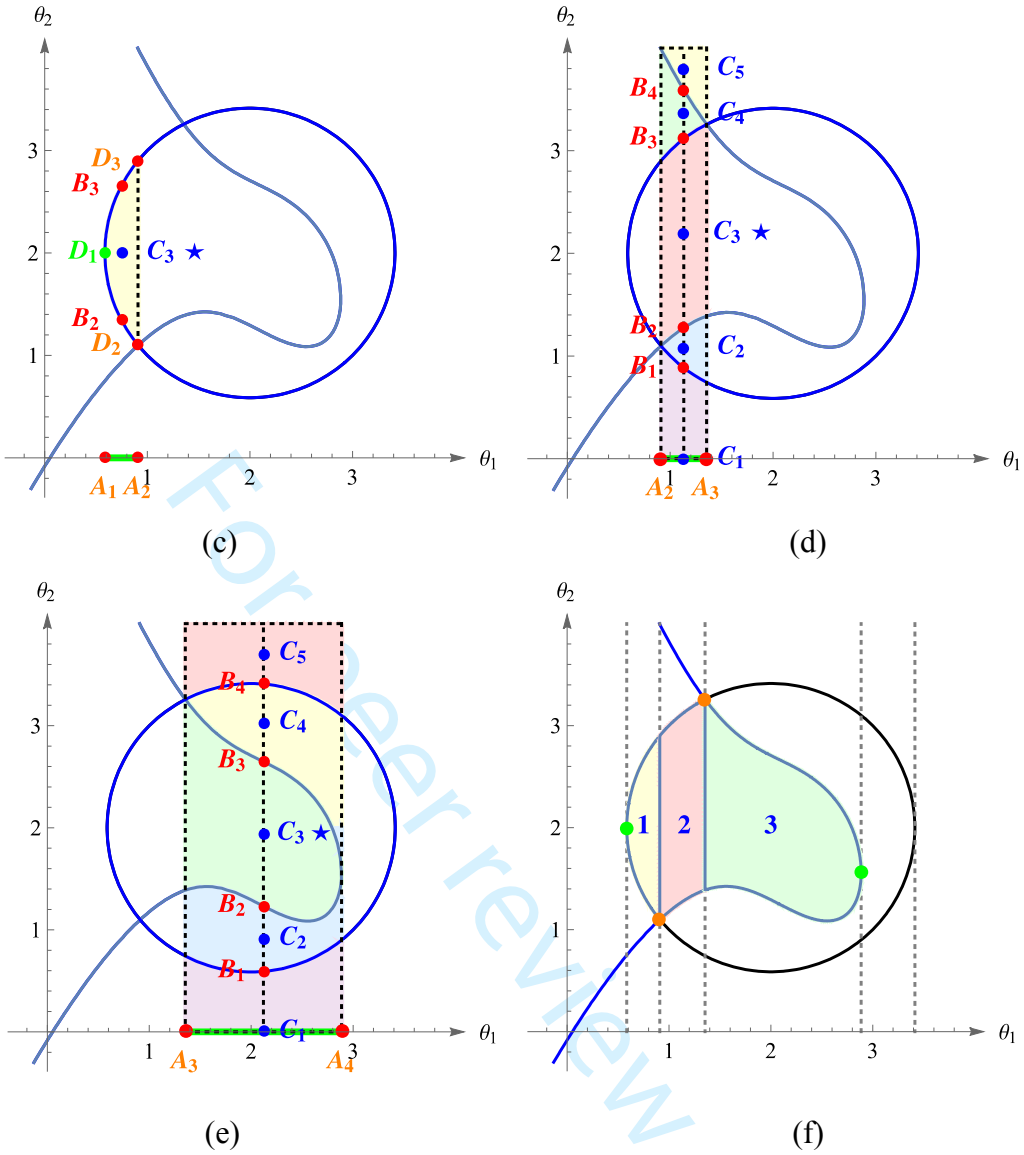


Figure 3. Solution process of the CAD method for the example.

According to the KS function, for a fixed ρ, the design space can be approximated by

$$y := KS(\boldsymbol{\theta},\boldsymbol{x}) \leq 0 \tag{17}$$

However, Equation (17) is an implicit and transcendental function, which cannot be directly used for symbolic computation. A common way is to create its polynomial surrogate model. As shown in Figure 4, $KS(\boldsymbol{\theta},\boldsymbol{x})$ aggregates all the inequality constraints. The surrogate model $\hat{y}(\overline{\boldsymbol{\theta}})$ can then be created through polynomial fitting, while the state variables $\boldsymbol{x}$ are eliminated. Thus, the CAD result of $\hat{y}(\overline{\boldsymbol{\theta}}) \leq 0$ contains a series of subspaces, each of which is a triangular system of $\overline{\theta}_1,\ldots,\overline{\theta}_P$.

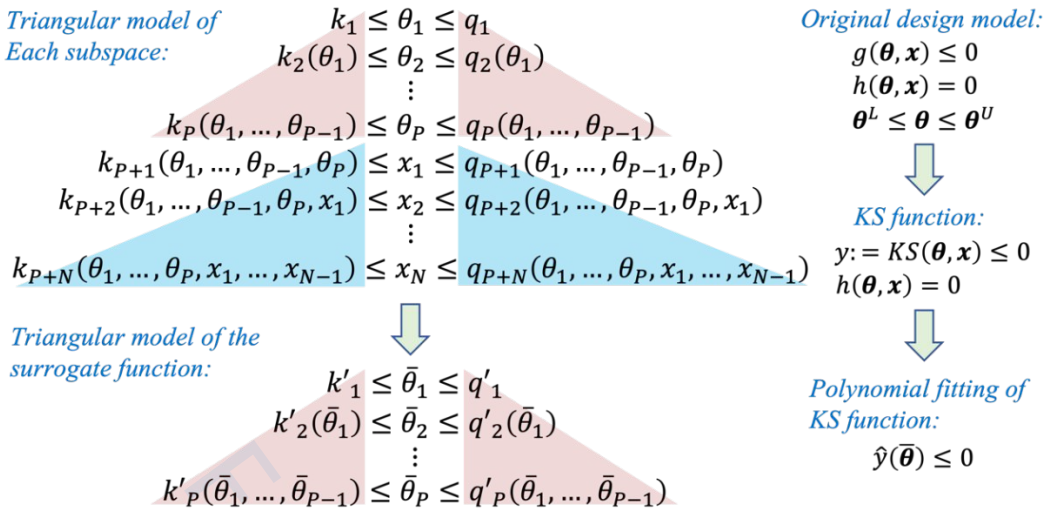


Figure 4. Triangular structure obtained by the CAD method.

Once the polynomial surrogate model $\hat{y}(\bar{\boldsymbol{\theta}})$ is obtained, it is convenient to analyze the design space, because the space can be described by a single constraint. Moreover, as shown in Figure 2(b), the smooth profile of the KS function has no intersection points, and we do not need to calculate the resultant for a single KS function.

5. Polynomial fitting of KS function through adaptive sampling

To create a polynomial surrogate model of the KS function, an adaptive sampling strategy is proposed. According to Equation (17), a point on the boundary of the KS space is denoted as

$$\{(\theta_1,\dots,\theta_P) \in \mathbb{R}^P | KS(\boldsymbol{\theta},\boldsymbol{x}) = 0\} \tag{18}$$

which can be solved by the following system of equations,

$$\begin{aligned} &KS(\boldsymbol{\theta},\boldsymbol{x}) = 0 \\ &h_i(\boldsymbol{\theta},\boldsymbol{x}) = 0, \quad \forall i \in I \end{aligned} \tag{19}$$

Note that, since the bounds of process parameters are also used for formulating the KS function, the KS space must be a closed space. If the design space is described by a set of inequality constraints, the traditional methods to build a surrogate model need to sample enough points over the whole space. However, in this work, a single equation, $KS(\boldsymbol{\theta},\boldsymbol{x}) = 0$, can describe the entire boundary of the KS space. Thus, we only focus on exploring enough sampling points on the boundary to create the surrogate model. The solution strategy of the proposed adaptive

sampling method is shown in Figure 5, which contains four steps:

(1) *Initial LHS sampling*. For a given number of sampling points, perform Latin hypercube sampling (LHS) over the space of process parameters. Those feasible sampled points in the feasible region forms a set **A**;

(2) *Expanded boundary points*. Through each pair of points in **A**, a line can be generated. Two intersection points of the line and the boundary can be obtained. All the intersection points on the boundary form an initial set **B**;

(3) *Adaptive sampling*. Based on the KS function, a simple DFO model of adaptive sampling is proposed. K-fold cross validation is applied to evaluate the stopping criteria. For each adaptive sampling point in set **C**, P points around the current adaptive point can be found through perturbation, which can form set **D**. All four sets are used for polynomial fitting.

(4) *Explicit description*. Through the CAD method, the fitted polynomial model $\hat{y}(\overline{\boldsymbol{\theta}})$ can be used to deduce the explicit expression of the KS space.

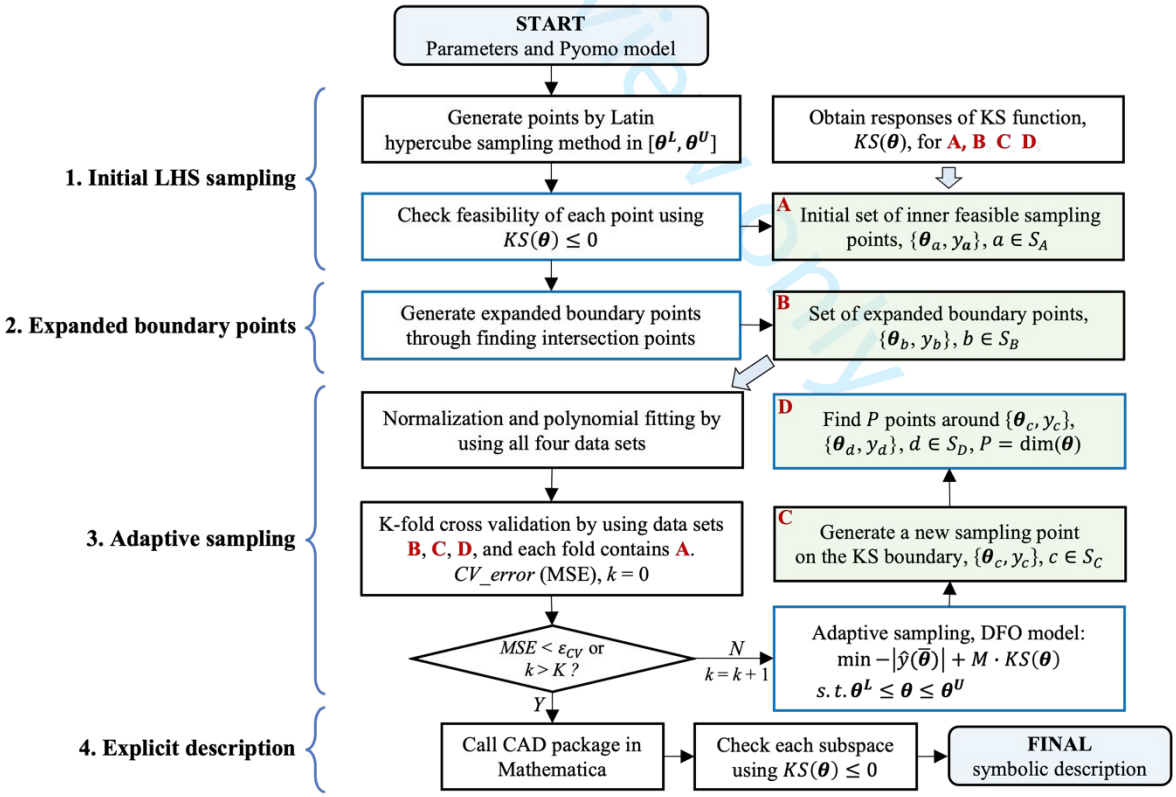


Figure 5. Solution strategy of the adaptive sampling method.

5.1. Initial LHS sampling

The LHS strategy[33] is applied to generate a given number of sampling points, denoted as num, in the process parameter space, where upper and lower bounds are required. Then, a feasibility check needs to be performed by evaluating $KS(\boldsymbol{\theta},\boldsymbol{x})$ at each LHS point. The presence of the state variables implies that the KS space is described by a set of multivariate functions of $\boldsymbol{\theta}$ and $\boldsymbol{x}$. Thus, the feasibility can be checked by solving the following NLP model,

$$\begin{aligned} & u_a = \min u \\ & s.t.\ KS(\boldsymbol{\theta},\boldsymbol{x}) \le u \\ & \quad h_i(\boldsymbol{\theta},\boldsymbol{x}) = 0, \quad \forall i \in I \end{aligned} \tag{20}$$

where $u_a \le 0$ indicates that the point is feasible. All the feasible LHS sampling points and the corresponding output of $KS(\boldsymbol{\theta},\boldsymbol{x})$ form set $\mathbf{A} := \{(\boldsymbol{\theta}_a, y_a), \forall a \in S_A\}$. Note that the number of LHS sampling points needs to be specified in advance. The more sampling points specified, the larger set **A** obtained, and the easier polynomial fitting will be.

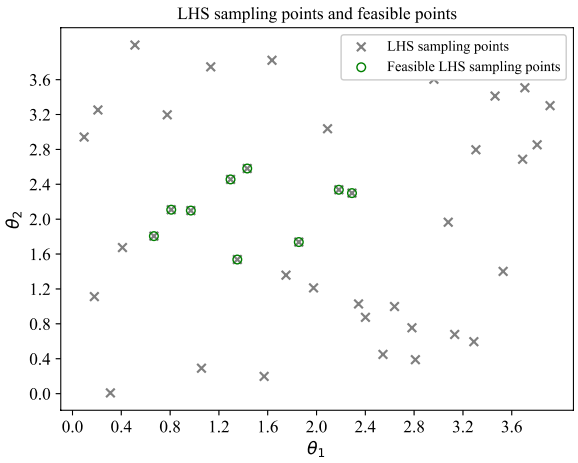


Figure 6. Initial LHS sampling of the example. ($\rho = 2, num = 40$)

For the above-mentioned example with $\rho = 2$, the sampling ranges are set to [0, 4] and [0, 4] for θ_1 and θ_2. As shown in Figure 6, 40 points are sampled, and 9 of them are feasible.

5.2. Expanded boundary points

To build the surrogate model, we need to find enough sampling points on the boundary of the

KS space. In this subsection, based on the properties of the KS function and the obtained initial set **A**, an approach to efficiently locate many points on the boundary is proposed. For a pair of points in the set **A**, e.g., $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$, there are two directions, $\boldsymbol{\theta}_1 \rightarrow \boldsymbol{\theta}_2$ and $\boldsymbol{\theta}_2 \rightarrow \boldsymbol{\theta}_1$, which can be formulated as follows,

$$\begin{aligned} &\boldsymbol{\theta}_1 \rightarrow \boldsymbol{\theta}_2: \ \ \boldsymbol{\theta} = \boldsymbol{\theta}_1 + \delta \cdot (\boldsymbol{\theta}_2 - \boldsymbol{\theta}_1) \\ &\boldsymbol{\theta}_2 \rightarrow \boldsymbol{\theta}_1: \ \ \boldsymbol{\theta} = \boldsymbol{\theta}_2 + \delta \cdot (\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2) \end{aligned} \tag{21}$$

Along both directions, we can find two intersection points on the KS boundary by solving the following two optimization problems,

$$\begin{aligned} &\max \delta \\ &s.t. \ \ KS(\boldsymbol{\theta},\boldsymbol{x}) = 0 \\ &\quad\ h_i(\boldsymbol{\theta},\boldsymbol{x}) = 0, \ \ \forall i \in I \\ &\quad\ \boldsymbol{\theta} = \boldsymbol{\theta}_1 + \delta \cdot (\boldsymbol{\theta}_2 - \boldsymbol{\theta}_1) \end{aligned} \tag{22}$$

$$\begin{aligned} &\max \delta \\ &s.t. \ \ KS(\boldsymbol{\theta},\boldsymbol{x}) = 0 \\ &\quad\ h_i(\boldsymbol{\theta},\boldsymbol{x}) = 0, \ \ \forall i \in I \\ &\quad\ \boldsymbol{\theta} = \boldsymbol{\theta}_2 + \delta \cdot (\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2) \end{aligned} \tag{23}$$

If the set **A** contains S_A points, the total number of directions for all the pairs of points can be calculated from Equation (24), which is a permutation problem. The obtained boundary points and the output of $KS(\boldsymbol{\theta},\boldsymbol{x})$ then define the set of initial boundary points, $\mathbf{B} := \{(\boldsymbol{\theta}_b, y_b), \forall b \in S_B\}$. For example, in Figure 7, 4 points can generate $4 \times 3 = 12$ boundary points.

$$S_B = A_{S_A}^2 = S_A \cdot (S_A - 1) \tag{24}$$

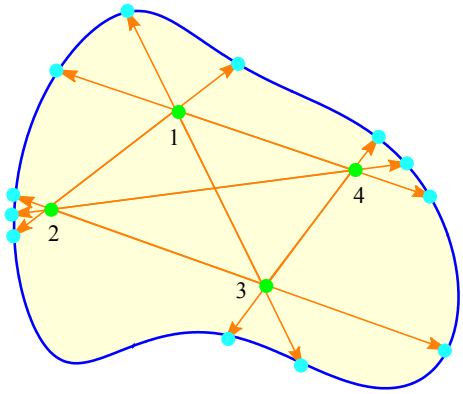


Figure 7. Illustration of expanded boundary points via line projection.

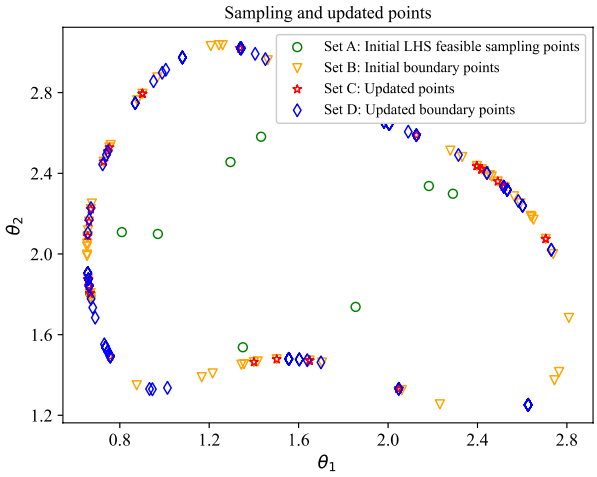


Figure 8. Sampling points of the example. ($\rho = 2, num = 40$).

For this example, 9 × 8=72 initial boundary points can be obtained by solving Equations (22) and (23), which are marked as yellow points in Figure 8. Note that, even though we only have 9 feasible LHS feasible points, 72 boundary points can be directly found in this step.

5.3. Adaptive sampling

Based on the set of feasible LHS points **A** and the initial set of expanded boundary points **B**, a common multivariable polynomial fitting method can be executed to build the surrogate model between the process parameters $\boldsymbol{\theta} = (\theta_1,...,\theta_P)$ and the response y, denoted as $\hat{y}(\boldsymbol{\theta})$,

$$\hat{y}(\boldsymbol{\theta}) = \sum_{k=0}^{K} c_k \phi_k(\boldsymbol{\theta}) \quad where\ \phi_k(\boldsymbol{\theta}) = \theta_1^{k_1} \cdots \theta_P^{k_P} \tag{25}$$

where $c_k \in \mathbb{R}$ and $c_K \neq 0$. The sum is taken over all indices (nonnegative integer vectors) with $k_P \leq deg$. For example, a bi-variable polynomial with $deg = 3$, is formulated as follows,

$$\begin{aligned}\hat{y}(\theta_1,\theta_2) = c_0 + c_1\theta_2 + c_2\theta_2^2 + c_3\theta_2^3 + c_4\theta_1 + c_5\theta_1\theta_2 + \\ c_6\theta_1\theta_2^2 + c_7\theta_1\theta_2^3 + c_8\theta_1^2 + c_9\theta_1^2\theta_2 + c_{10}\theta_1^2\theta_2^2 + \\ c_{11}\theta_1^2\theta_2^3 + c_{12}\theta_1^3 + c_{13}\theta_1^3\theta_2 + c_{14}\theta_1^3\theta_2^2 + c_{15}\theta_1^3\theta_2^3\end{aligned}$$

The K-fold cross-validation (CV) method[34] is used to evaluate the result of polynomial fitting. The number of folds, k_f, should be specified in advance. ε_{CV} is given as the stop criteria of CV. The set of **B**, **C** and **D** is split into k_f folds for cross validation, and the set **A** is used for each

fold. The maximum MSE (mean squared error) for all folds is used for comparison with ε_{CV}. Moreover, the data set of process parameters should be normalized before polynomial fitting, denoted as $\overline{\boldsymbol{\theta}}$. Then, a new point $\boldsymbol{\theta}$ can be roughly scaled by the current stored minimization $\boldsymbol{\theta}^{min}$ and maximization $\boldsymbol{\theta}^{max}$, and the corresponding output of the fitted model is $\hat{y}(\overline{\boldsymbol{\theta}})$,

$$\overline{\boldsymbol{\theta}} = \frac{\boldsymbol{\theta} - \boldsymbol{\theta}^{min}}{\boldsymbol{\theta}^{max} - \boldsymbol{\theta}^{min}} \tag{26}$$

The purpose of adaptive sampling is to search the design space for areas of model inconsistency or model mismatch and to find points that maximize the model error. Since only the points on the boundary are considered, the adaptive sampling model is proposed as,

$$\begin{aligned} &\max |\hat{y}(\overline{\boldsymbol{\theta}})| \\ &s.t.\ KS(\boldsymbol{\theta},\boldsymbol{x}) = 0 \\ &\quad h_i(\boldsymbol{\theta},\boldsymbol{x}) = 0,\ \ \forall i \in I \\ &\quad \boldsymbol{\theta}^{\boldsymbol{L}} \le \boldsymbol{\theta} \le \boldsymbol{\theta}^{\boldsymbol{U}} \end{aligned} \tag{27}$$

After relaxing the objective function $|\hat{y}(\overline{\boldsymbol{\theta}})|$ into two constraints, $\hat{y}(\overline{\boldsymbol{\theta}}) \le Z$ and $-\hat{y}(\overline{\boldsymbol{\theta}}) \le Z$, we can apply an NLP solver to solve Equation (27); however, most adaptive sampling points are likely to fall into the same area that has the maximum model error and cannot escape to explore other areas, which is not conducive to polynomial fitting. Therefore, in order to explore the areas that contain more local information, we transform Equation (27) into the following derivative-free optimization (DFO) model,

$$\begin{aligned} &\min\ -|\hat{y}(\overline{\boldsymbol{\theta}})| + M \cdot KS(\boldsymbol{\theta}) \\ &s.t. \boldsymbol{\theta}^{\boldsymbol{L}} \le \boldsymbol{\theta} \le \boldsymbol{\theta}^{\boldsymbol{U}} \end{aligned} \tag{28}$$

where M is a penalty coefficient. Compared with Equation (27), the equality system, i.e., $h_i(\boldsymbol{\theta},\boldsymbol{x}) = 0, \forall i \in I$, can be viewed as a black box. This black-box model is used to calculate the state variables $\boldsymbol{x}$ for each iteration of adaptive sampling, and then to evaluate the objective function of Equation (28).

A DFO solver, Py-BOBYQA, which is a Python implementation of the BOBYQA (Bound Optimization BY Quadratic Approximation) Fortran solver by Powell[35], is employed to solve

the adaptive sampling model. In this work, the purpose of adaptive sampling is to explore local areas leading to model mismatch as many as possible, rather than finding global optima in each iteration. Py-BOBYQA is based on the trust-region method, and it has superior performance in finding local solutions of nonlinear and nonconvex problems. Thus, we use this DFO solver to address the adaptive sampling problem. In addition, the initial values are of great importance to the results of the DFO solver. To take a more complete search, a reproducible random sequence of the data set is generated as the initial values, and more local optima can be found. The obtained adaptive sampling points form set $\mathbf{C} := \{(\boldsymbol{\theta}_c, y_c), \forall c \in S_C\}$. To further expand the points on the boundary, for a point in $\mathbf{C}$, $\boldsymbol{\theta}_c := (\theta_{c_1},...,\theta_{c_i},...,\theta_{c_P})$, a perturbation method is proposed to find P points around $\boldsymbol{\theta}_c$, where P is the dimension of $\boldsymbol{\theta}$,

$$\boldsymbol{\theta}_c^i = (\theta_{c_1}^i,...\theta_{c_i}^i,...,\theta_{c_P}^i), \quad \forall i \in P \tag{29}$$

First, we can find a point $\boldsymbol{\theta}_a := (\theta_{a_1},...,\theta_{a_i},...,\theta_{a_P})$ in set $\mathbf{A}$, which has the longest Euclidean distance to $\boldsymbol{\theta}_c$. Then, assuming that the perturbation factor is α, if $\boldsymbol{\theta}_a$ is located on the left side of $\boldsymbol{\theta}_c$ on the coordinate θ_i, i.e., $\theta_{a_i} < \theta_{c_i}$, $\theta_{c_i}^i$ can be calculated by,

$$\theta_{c_i}^i = \theta_{c_i} \cdot (1 + \alpha), \quad i \in \{1,2,...,P\} \tag{30}$$

If $\boldsymbol{\theta}_a$ is located on the right side of $\boldsymbol{\theta}_c$, i.e., $\theta_{a_i} > \theta_{c_i}$, $\theta_{c_i}^i$ can be calculated by,

$$\theta_{c_i}^i = \theta_{c_i} \cdot (1 - \alpha), \quad i \in \{1,2,...,P\} \tag{31}$$

If $\theta_{a_i} = \theta_{c_i}$, $\theta_{c_i}^i = \theta_{c_i}$. For each point $\boldsymbol{\theta}_c^i$, the direction $\boldsymbol{\theta}_a \rightarrow \boldsymbol{\theta}_c^i$, can be formulated as,

$$\boldsymbol{\theta} = \boldsymbol{\theta}_a + \delta \cdot (\boldsymbol{\theta}_c^i - \boldsymbol{\theta}_a) \tag{32}$$

Through Equation (22), the intersection point on the KS boundary can be solved. All such intersection points can form set $\mathbf{D} := \{(\boldsymbol{\theta}_d, y_d), \forall d \in S_D\}$.

To illustrate the above method, Figure 9 shows geometric interpretation of a bivariate case. In Figure 9(a), $\boldsymbol{\theta}_c = (\theta_{c_1},\theta_{c_2})$ is the current adaptive sampling point, and the blue point $\boldsymbol{\theta}_a = (\theta_{a_1},\theta_{a_2})$ is a point in set $\mathbf{A}$ which is the farthest point from $\boldsymbol{\theta}_c$. Since the dimension P is 2 and

$\boldsymbol{\theta}_a$ is on the left of $\boldsymbol{\theta}_c$, two perturbation points around $\boldsymbol{\theta}_c$ can be expressed as follows,

$$\boldsymbol{\theta}_c^1 = (\theta_{c_1} \cdot (1+\alpha), \theta_{c_2})$$

$$\boldsymbol{\theta}_c^2 = (\theta_{c_1}, \theta_{c_2} \cdot (1+\alpha))$$

Thus, the following two direction formulations can be used to find two intersection points, $\boldsymbol{\theta}_d^1$ and $\boldsymbol{\theta}_d^2$, on the boundary,

$$\boldsymbol{\theta} = \boldsymbol{\theta}_a + \delta \cdot (\boldsymbol{\theta}_c^1 - \boldsymbol{\theta}_a)$$

$$\boldsymbol{\theta} = \boldsymbol{\theta}_a + \delta \cdot (\boldsymbol{\theta}_c^2 - \boldsymbol{\theta}_a)$$

Similarly, as shown in Figure 9(b), $\boldsymbol{\theta}_a$ is on the right of $\boldsymbol{\theta}_c$, thus,

$$\boldsymbol{\theta}_c^1 = (\theta_{c_1} \cdot (1-\alpha), \theta_{c_2})$$

$$\boldsymbol{\theta}_c^2 = (\theta_{c_1}, \theta_{c_2} \cdot (1-\alpha))$$

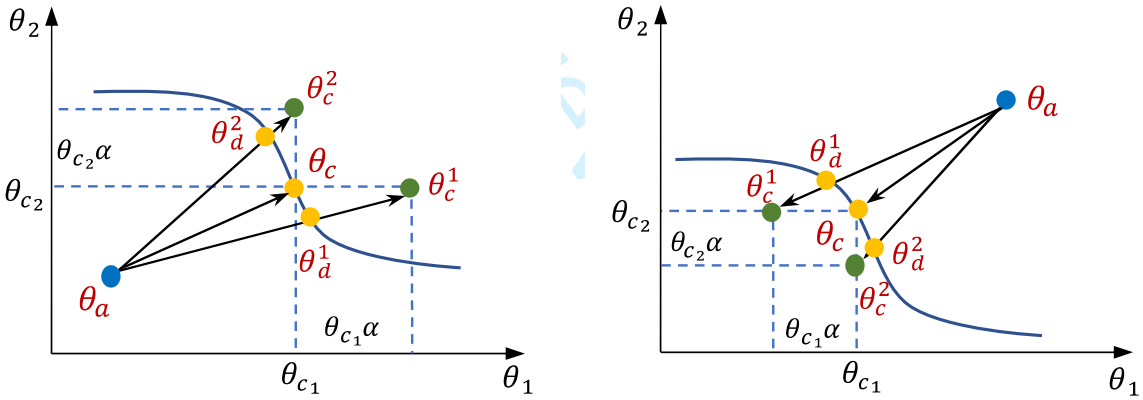
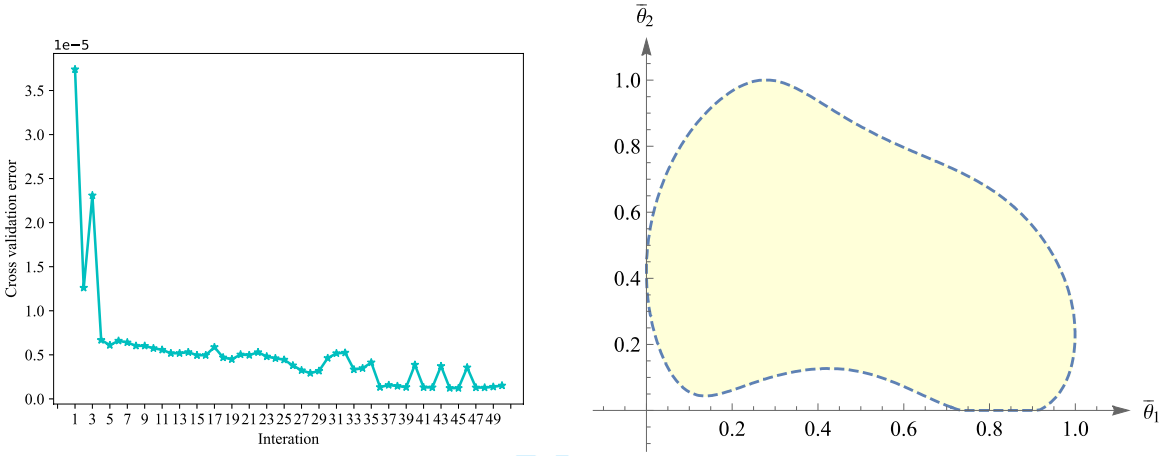


Figure 9. Perturbation points for an adaptive sampling point.

For the example, the maximum number of iterations for adaptive sampling is set to 50. Figure 8 shows the results of four different sets of points. All the points are used for polynomial fitting. Figure 10(a) shows the MSE values of 4-fold CV. The final surrogate model with 4 degrees is

$$\begin{aligned}\hat{y}_0(\bar{\theta}_1,\bar{\theta}_2) = {} & 0.332558 - 4.924684\bar{\theta}_1 + 27.677621\bar{\theta}_1^{\,2} - 48.130965\bar{\theta}_1^{\,3} + 25.497110\bar{\theta}_1^{\,4} \\ & - 1.355176\bar{\theta}_2 + 5.423242\bar{\theta}_1\bar{\theta}_2 - 54.471859\bar{\theta}_1^{\,2}\bar{\theta}_2 + 89.127179\bar{\theta}_1^{\,3}\bar{\theta}_2 \\ & - 43.856384\bar{\theta}_1^{\,4}\bar{\theta}_2 + 1.129803\bar{\theta}_2^{\,2} - 18.827193\bar{\theta}_1\bar{\theta}_2^{\,2} + 60.655279\bar{\theta}_1^{\,2}\bar{\theta}_2^{\,2} \\ & + 21.796085\bar{\theta}_1^{\,3}\bar{\theta}_2^{\,2} - 44.339153\bar{\theta}_1^{\,4}\bar{\theta}_2^{\,2} - 0.2916\bar{\theta}_2^{\,3} + 14.54047\bar{\theta}_1\bar{\theta}_2^{\,3} \\ & + 54.884484\bar{\theta}_1^{\,2}\bar{\theta}_2^{\,3} - 326.654775\bar{\theta}_1^{\,3}\bar{\theta}_2^{\,3} + 223.200627\bar{\theta}_1^{\,4}\bar{\theta}_2^{\,3} + 1.887766\bar{\theta}_2^{\,4} \\ & - 11.284009\bar{\theta}_1\bar{\theta}_2^{\,4} - 44.465319\bar{\theta}_2^{\,4} + 212.964235\bar{\theta}_1^{\,3}\bar{\theta}_2^{\,4} - 134.362005\bar{\theta}_1^{\,4}\bar{\theta}_2^{\,4}\end{aligned}$$



(a) (b)

Figure 10. Surrogate model of the example ($\rho = 2, num = 40$).

5.4. Explicit description

Since the sampling data has been normalized before performing polynomial fitting, the profile of the KS space, $\hat{y}(\bar{\boldsymbol{\theta}}) \le 0$, should be limited within a square with [0,1] sides. Considering the polynomial fitting error, the profile may be slightly outside of the square. Hence, the KS space based on the surrogate model is the intersection of $\hat{y}(\bar{\boldsymbol{\theta}}) \le 0$ and $0 \le \bar{\boldsymbol{\theta}} \le 1$, i.e.,

$$\{\bar{\boldsymbol{\theta}} \in \mathbb{R}^P | \hat{y}(\bar{\boldsymbol{\theta}}) \le 0 \wedge 0 \le \bar{\boldsymbol{\theta}} \le 1\} \tag{33}$$

The CAD method can transform Equation (33) into a triangular system, which consists of a series of subspaces, and each subspace is formulated as,

$$\begin{aligned} k_1' \le \bar{\theta}_1 \le q_1' \\ k_2'(\bar{\theta}_1) \le \bar{\theta}_2 \le q_2'(\bar{\theta}_1) \\ \vdots \\ k_P'(\bar{\theta}_1,\ldots,\bar{\theta}_{P-1}) \le \bar{\theta}_P \le q_P'(\bar{\theta}_1,\ldots,\bar{\theta}_{P-1}) \end{aligned} \tag{34}$$

For the example, the corresponding triangular system with $\overline{\theta}_1 \prec \overline{\theta}_2$ is as follows:

$$1:\begin{cases}0 \le \overline{\theta}_1 < 0.0964731 \\ Root(\Gamma_0(\overline{\theta}_1, \#1)\&,1) \le \overline{\theta}_2 \le Root(\Gamma_0(\overline{\theta}_1, \#1)\&, 2)\end{cases}$$

$$2:\begin{cases}0.0964731 \le \overline{\theta}_1 < 0.137589 \\ Root(\Gamma_0(\overline{\theta}_1, \#1)\&,3) \le \overline{\theta}_2 \le Root(\Gamma_0(\overline{\theta}_1, \#1)\&,4)\end{cases}$$

$$3:\begin{cases}0.137589 \le \overline{\theta}_1 \le 0.266024 \\ Root(\Gamma_0(\overline{\theta}_1, \#1)\&,2) \le \overline{\theta}_2 \le Root(\Gamma_0(\overline{\theta}_1, \#1)\&, 3)\end{cases}$$

$$4:\begin{cases}0.266024 < \overline{\theta}_1 \le 0.290682 \\ Root(\Gamma_0(\overline{\theta}_1, \#1)\&,2) \le \overline{\theta}_2 \le 1\end{cases}$$

$$5:\begin{cases}0.290682 < \overline{\theta}_1 \le 0.376366 \\ Root(\Gamma_0(\overline{\theta}_1, \#1)\&,2) \le \overline{\theta}_2 \le Root(\Gamma_0(\overline{\theta}_1, \#1)\&, 3)\end{cases}$$

$$6:\begin{cases}0.376366 < \overline{\theta}_1 \le 0.732135 \\ Root(\Gamma_0(\overline{\theta}_1, \#1)\&,1) \le \overline{\theta}_2 \le Root(\Gamma_0(\overline{\theta}_1, \#1)\&,2)\end{cases}$$

$$7:\begin{cases}0.732135 < \overline{\theta}_1 \le 0.91514 \\ 0 \le \overline{\theta}_2 \le Root(\Gamma_0(\overline{\theta}_1, \#1)\&, 2)\end{cases}$$

$$8:\begin{cases}0.91514 < \overline{\theta}_1 \le 1 \\ Root(\Gamma_0(\overline{\theta}_1, \#1)\&,1) \le \overline{\theta}_2 \le Root(\Gamma_0(\overline{\theta}_1, \#1)\&,2)\end{cases}$$

where $\Gamma_0(\overline{\theta}_1, \#1)$ is a pure function with the parameter $\overline{\theta}_1$, which is provided in Supporting Information. $Root(\Gamma_0(\overline{\theta}_1, \#1)\&,n)$ means nth root of $\Gamma_0(\overline{\theta}_1, \#1) = 0$ for a given $\overline{\theta}_1$. The result indicates that the bounds of $\overline{\theta}_2$ are explicit expressions of $\overline{\theta}_1$. To further test the feasibility of the subspaces, a random point $\overline{\boldsymbol{\theta}}_s$ can be converted into the KS space by Equation (35), denoted as $\boldsymbol{\theta}_s$. $KS(\boldsymbol{\theta}_s) \le 0$ means that the current subspace is feasible,

$$\boldsymbol{\theta}_s = \overline{\boldsymbol{\theta}}_s \cdot (\boldsymbol{\theta}^{max} - \boldsymbol{\theta}^{min}) + \boldsymbol{\theta}^{min} \tag{35}$$

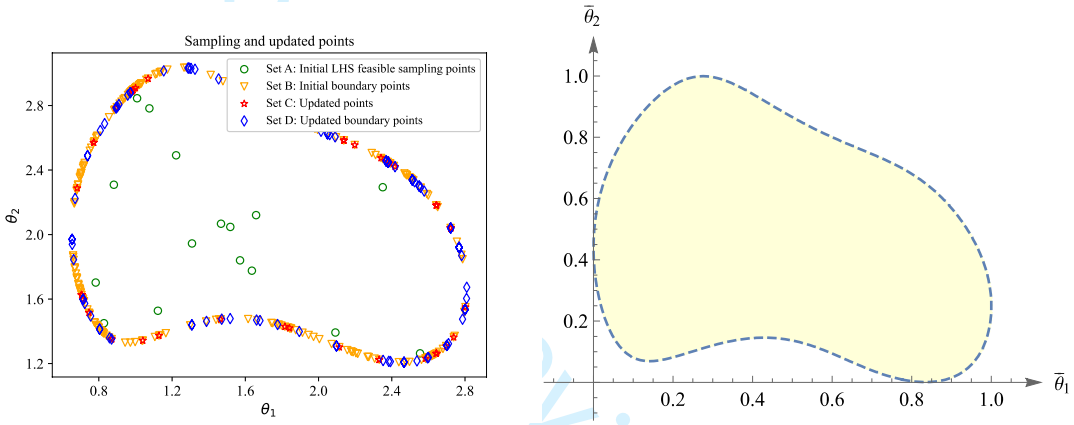
In the original data set, the maximum values of θ_1 and θ_2 are 2.809134, 3.03503, and the minimum values are 0.653351, 1.251803, respectively. Taking the first subspace as an example, a chosen point is $\overline{\boldsymbol{\theta}}_s$ = (0.05, 0.4) and the corresponding $\boldsymbol{\theta}_s$ is (0.761141, 1.965094). $KS(\boldsymbol{\theta}_s) = -0.207169 < 0$ means that the first subspace is feasible. Similarly, we can find that the other subspaces are also feasible. The final KS space is shown in Figure 10(b).

Note that through Equations (26), (34) and (35), the CAD result can be used to evaluate the KS space. Each point satisfying $\hat{y}(\overline{\boldsymbol{\theta}}) \le 0$ can be transformed into the KS space. The triangular

formulation of KS space is formulated as follows,

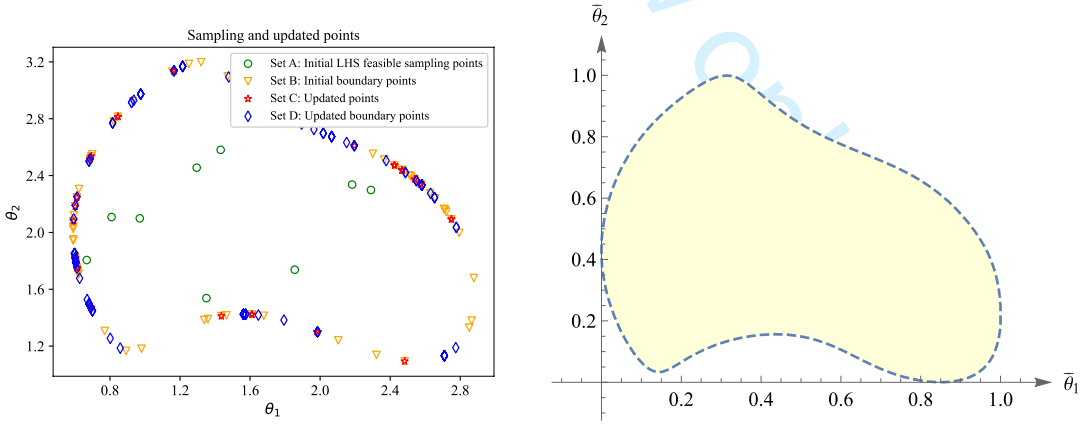
$$\begin{aligned}
&\boldsymbol{\theta}_1^{min} + \left(\boldsymbol{\theta}_1^{max} - \boldsymbol{\theta}_1^{min}\right) \cdot k_1' \le \theta_1 \le q_1' \cdot \left(\boldsymbol{\theta}_1^{max} - \boldsymbol{\theta}_1^{min}\right) + \boldsymbol{\theta}_1^{min} \\
&\boldsymbol{\theta}_2^{min} + \left(\boldsymbol{\theta}_2^{max} - \boldsymbol{\theta}_2^{min}\right) \cdot k_2'\left(\frac{\theta_1 - \boldsymbol{\theta}_1^{min}}{\boldsymbol{\theta}_1^{max} - \boldsymbol{\theta}_1^{min}}\right) \le \theta_2 \le q_2'\left(\frac{\theta_1 - \boldsymbol{\theta}_1^{min}}{\boldsymbol{\theta}_1^{max} - \boldsymbol{\theta}_1^{min}}\right) \cdot \left(\boldsymbol{\theta}_2^{max} - \boldsymbol{\theta}_2^{min}\right) + \boldsymbol{\theta}_2^{min} \\
&\vdots
\end{aligned} \tag{36}$$

Moreover, another critical parameter, the number of initial LHS points, num, is also discussed. First, compared with the result shown in Figure 8, we apply the algorithm by setting $num = 80$. In this case, 16 feasible LHS points can be found, and more expanded boundary points can be located. In Figure 11, the denser boundary points can generate a more precise surrogate model.



(a) (b)

Figure 11. Sampling points and surrogate model of the example. ($\rho = 2, num = 80$).



(a) (b)

Figure 12. Sampling points and surrogate model of the example. ($\rho = 5, num = 40$).

Then, another key parameter is ρ. Compared with $\rho = 2$, we test the KS function with $\rho = 5$. All sampling points are shown in Figure 12(a). The KS space is illustrated in Figure 12(b). We

can see that the shape of the KS space with $\rho = 5$ more completely covers the feasible region than Figure 11(b). Therefore, we can conclude that more LHS points are more conducive to the polynomial fitting, and larger ρ can make the profile closer to the original feasible region.

6. Case studies

Two case studies are presented to illustrate the proposed design space description method. Pyomo[36] is applied to define the models. The GAMS global solver, BARON, is used to solve the NLP models through the interface of Pyomo and GAMS, and Py-BOBYQA is applied to solve the DFO models. For both cases, the parameters in Table 1 are set to the same values. The difference is the parameter ρ, which will be specified at different values.

Table 1. The specifications of the parameters.

Initial LHS points, *num*	Degree of polynomial, *deg*	Perturbation, α	Maximum Iteration	MSE of CV, ε_{CV}	K-fold, k_f
100	4	0.2	50	10^{-10}	4

6.1. CSTR reaction

This case study deals with a 2-step reaction with the following mechanism[10,37],

$$A + B \xrightarrow{k_1} C,\ r_1 = k_1 \cdot c_A \cdot c_b$$
$$C \xrightarrow{k_2} D + E,\ r_2 = k_2 \cdot c_c$$

where r_j are the reaction rates. Two process parameters correspond to the residence time, θ_1, and the ratio of the concentration of B to A, θ_2. k_j correspond to the model parameters, which are fixed at their mean values $\{0.31051, 0.026650\}$. The feasible ranges of θ_1 and θ_2 are given as follows.

$$0 \le \theta_1 \le 550$$
$$0 \le \theta_2 \le 6$$

The mass balance of the CSTR reaction is given by the following set of equations,

$$c_A^0 - c_A + \theta_1 \cdot (-r_1) = 0$$
$$c_B^0 - c_B + \theta_1 \cdot (-r_1) = 0$$
$$c_C^0 - c_C + \theta_1 \cdot (r_1 - r_2) = 0$$
$$c_D^0 - c_D + \theta_1 \cdot r_2 = 0$$
$$c_E^0 - c_E + \theta_1 \cdot r_2 = 0$$

where c_i^0 are initial concentrations $\{c_A^0 = 0.53, c_B^0 = 0.53 \cdot \theta_2, c_C^0 = 0, c_D^0 = 0, c_E^0 = 0\}$. The quality constraints are the minimum yield of *D* and the minimum ratio of *D* to unreacted species.

$$\frac{c_D}{c_A^0 - c_A} \geq 0.9$$
$$\frac{c_D}{c_A + c_B + c_C} \geq 0.2$$

In this case, ρ is fixed to 20. Figure 13(a) shows the feasible LHS points and updated points. All the updated points are located at the boundary of the KS space, and we can create the following 4-degree polynomial surrogate model with this small amount of sampling points.

$$\begin{aligned}\hat{y}_1(\bar{\theta}_1,\bar{\theta}_2) = {} & 0.00003937 + 0.001384\bar{\theta}_1 - 0.000118\bar{\theta}_1^2 - 0.003772\bar{\theta}_1^3 + 0.002389\bar{\theta}_1^4 \\ & - 0.000678\bar{\theta}_2 - 0.165238\bar{\theta}_1\bar{\theta}_2 - 0.563174\bar{\theta}_1^2\bar{\theta}_2 + 0.265011\bar{\theta}_1^3\bar{\theta}_2 \\ & + 0.464374\bar{\theta}_1^4\bar{\theta}_2 + 0.003796\bar{\theta}_2^2 - 0.790496\bar{\theta}_1\bar{\theta}_2^2 + 10.757368\bar{\theta}_1^2\bar{\theta}_2^2 \\ & - 15.685365\bar{\theta}_1^3\bar{\theta}_2^2 + 5.714148\bar{\theta}_1^4\bar{\theta}_2^2 - 0.007253\bar{\theta}_2^3 + 2.002811\bar{\theta}_1\bar{\theta}_2^3 \\ & - 21.109944\bar{\theta}_1^2\bar{\theta}_2^3 + 31.940187\bar{\theta}_1^3\bar{\theta}_2^3 - 12.825599\bar{\theta}_1^4\bar{\theta}_2^3 + 0.004422\bar{\theta}_2^4 \\ & - 1.035763\bar{\theta}_1\bar{\theta}_2^4 + 10.935095\bar{\theta}_1^2\bar{\theta}_2^4 - 16.595443\bar{\theta}_1^3\bar{\theta}_2^4 + 6.691871\bar{\theta}_1^4\bar{\theta}_2^4\end{aligned}$$

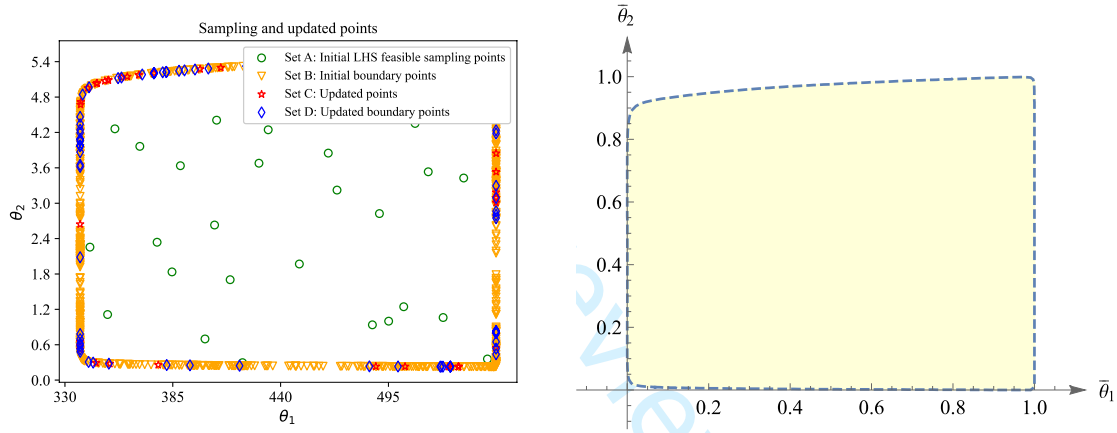
The profile of $\hat{y}_1(\bar{\theta}_1,\bar{\theta}_2) = 0$ is shown in Figure 13(b). Because of normalization, the profile is limited within a square with [0,1] sides. $\hat{y}_1(\bar{\theta}_1,\bar{\theta}_2) \leq 0$ can be used to describe the KS space. Through the CAD method, the equivalent triangular system can be obtained, which consists of 3 explicit 2D subspaces.

$$1:\begin{cases}0 \leq \bar{\theta}_1 \leq 0.107855 \\ Root(\Gamma_1(\bar{\theta}_1, \#1)\&, 2) \leq \bar{\theta}_2 \leq Root(\Gamma_1(\bar{\theta}_1, \#1)\&, 3)\end{cases}$$

$$2:\begin{cases}0.107855 < \bar{\theta}_1 < 0.90871 \\ Root(\Gamma_1(\bar{\theta}_1, \#1)\&, 1) \leq \bar{\theta}_2 \leq Root(\Gamma_1(\bar{\theta}_1, \#1)\&, 2)\end{cases}$$

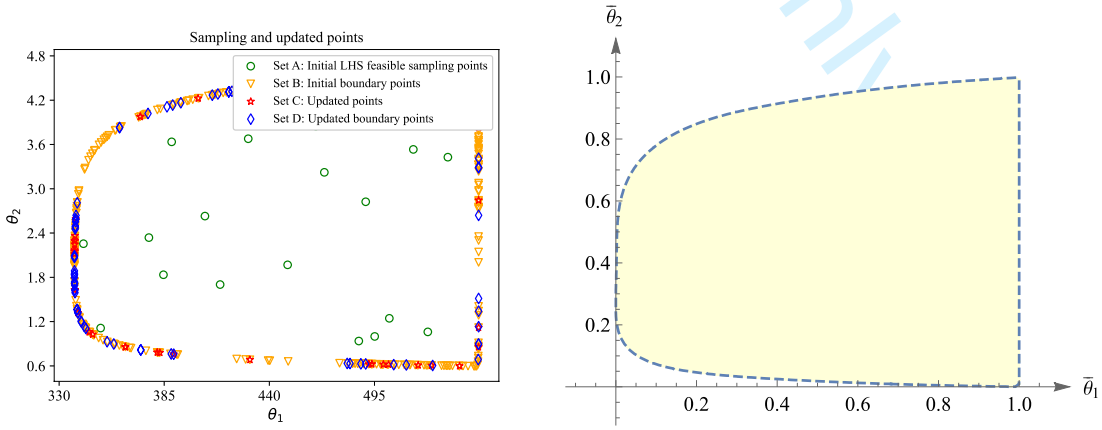
$$3:\begin{cases}0.90871 \leq \bar{\theta}_1 \leq 1 \\ 0 \leq \bar{\theta}_2 \leq Root(\Gamma_1(\bar{\theta}_1, \#1)\&, 2)\end{cases}$$

where $\Gamma_1(\overline{\theta}_1, \#1)$ is provided in Supporting Information. Due to the error of polynomial fitting, the feasibility of each subspace must be tested. In the stored data set, the maximum values of θ_1 and θ_2 are 549.94261, 5.491875; the minimum values are 337.704, 0.226248. Taking the second subspace as an example, a sampling point is chosen as $\overline{\boldsymbol{\theta}}_s$ = (0.5, 0.8). $\boldsymbol{\theta}_s$ = (443.823, 4.43875) is calculated by Equation (35). $KS(\boldsymbol{\theta}_s) = -0.011638 < 0$ indicates that the second subspace is feasible. Similarly, the other subspaces can be tested for feasibility Moreover, to show the performance of ρ, Figure 14 illustrates the results with $\rho = 5$. Since ρ is set smaller, the KS space is smaller than $\rho = 20$. The required CPU/Wall times are reported in Table 2.



(a) (b)

Figure 13. Results of design space description for Case 1. ($\rho = 20$, BARON)



(a) (b)

Figure 14. Results of design space description for Case 1. ($\rho = 5$, BARON)

6.2. Michael Addition Reaction

This case study deals with the Michael Addition Reaction with kinetics[10] described by the following equations,

$$AH + B \overset{k_1}{\rightarrow} A^- + BH^+,\ r_1 = k_1 \cdot c_{AH} \cdot c_B$$

$$A^- + C \overset{k_2}{\rightarrow} AC^-,\ r_2 = k_2 \cdot c_{A^-} \cdot c_C$$

$$AC^- \overset{k_3}{\rightarrow} A^- + C,\ r_3 = k_3 \cdot c_{AC^-}$$

$$AC^- + AH \overset{k_4}{\rightarrow} A^- + P,\ r_4 = k_4 \cdot c_{AC^-} \cdot c_{AH}$$

$$AC^- + BH^+ \overset{k_5}{\rightarrow} P + B,\ r_5 = k_5 \cdot c_{AC^-} \cdot c_{BH^+}$$

where r_i are reaction rates. The rate constants k_i correspond to the model parameters, fixed at their mean values: {49.7796, 8.9316, 1.3177, 0.3109, 3.8781}. The mass balance is as follows,

$$c_{AH}^0 - c_{AH} + \theta_1 \cdot (-r_1 - r_4) = 0$$

$$c_B^0 - c_B + \theta_1 \cdot (-r_1 + r_5) = 0$$

$$c_C^0 - c_C + \theta_1 \cdot (-r_2 + r_3) = 0$$

$$c_{A^-}^0 - c_{A^-} + \theta_1 \cdot (r_1 - r_2 + r_3 + r_4) = 0$$

$$c_{AC^-}^0 - c_{AC^-} + \theta_1 \cdot (r_2 - r_3 - r_4 - r_5) = 0$$

$$c_{BH^+}^0 - c_{BH^+} + \theta_1 \cdot (r_1 - r_5) = 0$$

$$c_P^0 - c_P + \theta_1 \cdot (r_4 + r_5) = 0$$

Two quality constraints are specifying that the conversion of C must be greater than 90%, and that the concentration of AC^- in the outlet must be less than 0.002,

$$\frac{c_C^0 - c_C - c_{AC^-}}{c_C^0} \geq 0.9$$

$$c_{AC^-} \leq 0.002$$

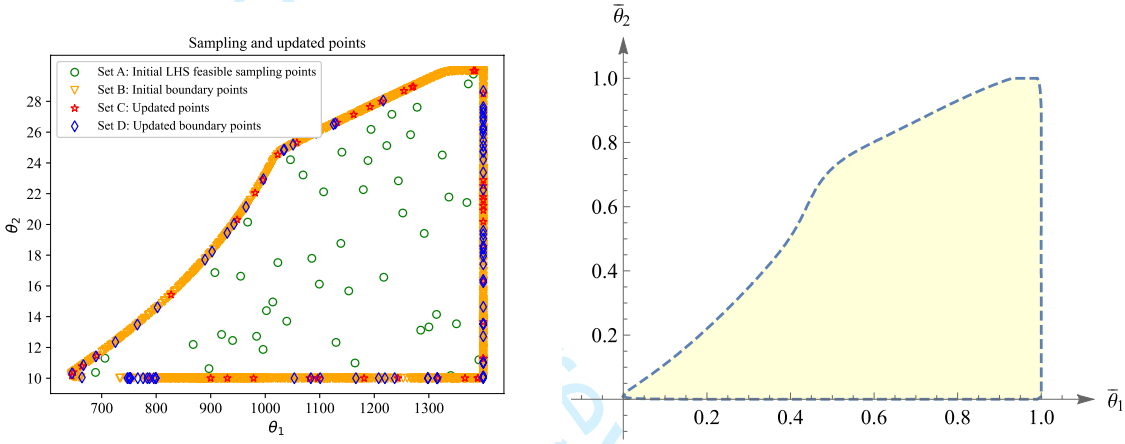
The initial concentrations $\{c_{AH}^0, c_B^0, c_C^0, c_{A^-}^0, c_{AC^-}^0, c_{BH^+}^0, c_P^0\}$ are set to $\{0.3955, 0.3955/\theta_2, 0.25, 0, 0, 0, 0\}$. The process parameters are the residence time θ_1 and the molar ratio θ_2.

$$400 \leq \theta_1 \leq 1400$$
$$10 \leq \theta_2 \leq 30$$

The KS parameter ρ is fixed to 10. Figure 15(a) shows all the feasible LHS points and updated boundary points, which can demonstrate the boundary of the KS space. Based on all the points,

the polynomial surrogate model with four degrees can be created.

$$\hat{y}_2(\bar{\theta}_1,\bar{\theta}_2) = 0.008668 - 0.070106\bar{\theta}_1 + 0.180528\bar{\theta}_1^2 - 0.193306\bar{\theta}_1^3 + 0.071808\bar{\theta}_1^4 - 1.627739\bar{\theta}_2 - 49.395899\bar{\theta}_1\bar{\theta}_2 + 212.180206\bar{\theta}_1^2\bar{\theta}_2 - 391.591254\bar{\theta}_1^3\bar{\theta}_2 + 230.359832\bar{\theta}_1^4\bar{\theta}_2 + 75.211453\bar{\theta}_2^2 - 364.868356\bar{\theta}_1\bar{\theta}_2^2 + 858.685883\bar{\theta}_1^2\bar{\theta}_2^2 - 532.802744\bar{\theta}_1^3\bar{\theta}_2^2 - 35.709792\bar{\theta}_1^4\bar{\theta}_2^2 - 38.784014\bar{\theta}_2^3 + 191.959125\bar{\theta}_1\bar{\theta}_2^3 - 902.219004\bar{\theta}_1^2\bar{\theta}_2^3 + 903.265210\bar{\theta}_1^3\bar{\theta}_2^3 - 155.198075\bar{\theta}_1^4\bar{\theta}_2^3 - 17.521365\bar{\theta}_2^4 + 153.291539\bar{\theta}_1\bar{\theta}_2^4 - 38.510764\bar{\theta}_1^2\bar{\theta}_2^4 - 103.476741\bar{\theta}_1^3\bar{\theta}_2^4 + 6.773545\bar{\theta}_1^4\bar{\theta}_2^4$$



(a) (b)

Figure 15. Results of design space description for Case 2. ($\rho = 10$, BARON)

The profile of the surrogate model $\hat{y}_2(\bar{\theta}_1,\bar{\theta}_2) = 0$ is shown in Figure 15(b). Because of the polynomial fitted error, the profile is slightly outside of the square. According to Equation (33), the CAD method can generate a triangular and explicit system as follows, where $\Gamma_2(\bar{\theta}_1, \#1)$ is provided in Supporting Information.

$$1:\begin{cases} 0 \le \bar{\theta}_1 \le 0.118986 \\ Root(\Gamma_2(\bar{\theta}_1, \#1)\&, 2) \le \bar{\theta}_2 \le Root(\Gamma_2(\bar{\theta}_1, \#1)\&, 3) \end{cases}$$

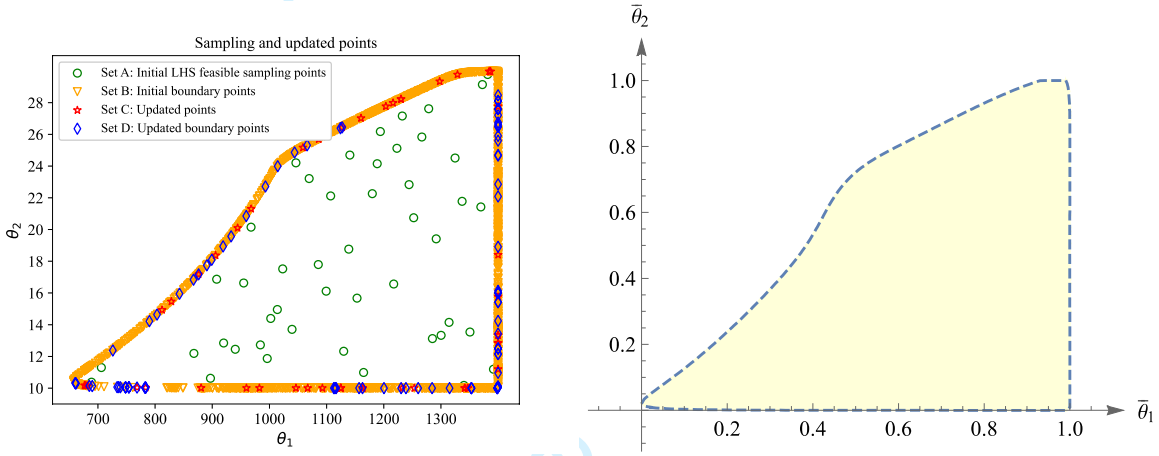
$$2:\begin{cases} 0.118986 < \bar{\theta}_1 < 0.228252 \\ Root(\Gamma_2(\bar{\theta}_1, \#1)\&,1) \le \bar{\theta}_2 \le Root(\Gamma_2(\bar{\theta}_1, \#1)\&,2) \end{cases}$$

$$3:\begin{cases} 0.228252 \le \bar{\theta}_1 \le 0.930122 \\ 0 \le \bar{\theta}_2 \le Root(\Gamma_2(\bar{\theta}_1, \#1)\&, 2) \end{cases}$$

$$4:\begin{cases} 0.930122 < \bar{\theta}_1 \le 0.990867 \\ 0 \le \bar{\theta}_2 \le 1 \end{cases}$$

$$5:\begin{cases} 0.990867 < \bar{\theta}_1 \le 1 \\ 0 \le \bar{\theta}_2 \le Root(\Gamma_2(\bar{\theta}_1, \#1)\&, 2) \end{cases}$$

In the data set of polynomial fitting, the maximum values of θ_1 and θ_2 are 1400 and 29.9977, and the minimum values are 644.084 and 10, respectively. Taking the third subspace as an example, a sampling point is chosen as $\bar{\boldsymbol{\theta}}_s$ = (0.6, 0.5) and $\boldsymbol{\theta}_s$ is (1097.63, 19.9989). $KS(\boldsymbol{\theta}_s) = -0.250481 < 0$ means that the subspace is feasible. Moreover, Figure 16 illustrates the results with $\rho = 5$. For this case, Figure 15(b) and Figure 16(b) are almost the same, which indicates that $\rho = 10$ can provide a good approximation of the feasible region.



(a) (b)

Figure 16. Results of design space description for Case 2. ($\rho = 5$, BARON)

Table 2. Summary of model information and computational expense.

	Model information		Proposed method			Use CAD method only
	Number of eqs / ineqs	Number of variables	ρ	Part 1: Adaptive sampling and surrogate modeling	Part 2: Explicit description	
Case 1	8 / 6	10	20	154.29 / 687.98 (BARON)	0.23 / 0.24	0.35 / 0.36
			5	89.81 / 578.12 (BARON)	0.26 / 0.29	
Case 2	13 / 6	15	10	352.68 / 2279.72 (BARON)	0.25 / 0.27	>7200 / >7200*
				251.68 / 476.16 (CONOPT4)	0.22 / 0.22	
			5	380.53/ 2701.63 (BARON)	0.25 / 0.25	
				265.71 / 498.40 (CONOPT4)	0.20 / 0.21	

*Cannot solve within 7200 seconds.

The computational time includes CPU time (s) / Wall time (s)

In Table 2, the model information of both cases and the computational times are summarized. Case 1 includes 8 equalities and 10 variables; Case 2 has a larger scale, involving 13 equalities

and 15 variables. The adaptive sampling and surrogate modeling are executed in Python, and the explicit description based on the CAD method is implemented in Mathematica. The CAD method transforms the surrogate model of KS function very fast. In addition, we also compare the computational time with using the CAD method only. The CAD method can directly solve Case 1 in 0.36s; however, Case 2 cannot be handled within 7200s, because of the larger scale. The proposed method can solve Case 2 in around 2500s when using the BARON solver.

7. Discussion

The proposed method consists of adaptive sampling and symbolic computation. There are some key points that can affect the results. To further clarify the performance of the proposed method, some discussion is provided in this section.

7.1. Values of critical parameters

The critical parameters mainly contain the initial LHS points, *num* and the parameter of the KS function, ρ. As shown in Figure 8 and Figure 11a, it is obvious that more initial LHS points are specified, the denser the boundary points can be found, and the more accurate the surrogate model will be. In addition, note that there may be empty spaces on the boundary, e.g., Figure 8. The main reasons are lack of feasible LHS points to generate the line projections on the boundary and/or lack of exploration during DFO search. The empty spaces can affect the polynomial fitting accuracy. In particular, if the empty area contains critical nonlinear features, the fitted polynomial will have a serious distortion. Thus, to reduce the empty areas, we can set a larger initial number of LHS points and a more stringent stopping criteria of adaptive sampling, including a larger iteration limit and a smaller termination error of K-fold cross validation.

Moreover, there are two main limitations for the KS function, i.e, the exponential functions can make the KS function strongly nonlinear; it is difficult to determine an appropriate value of ρ

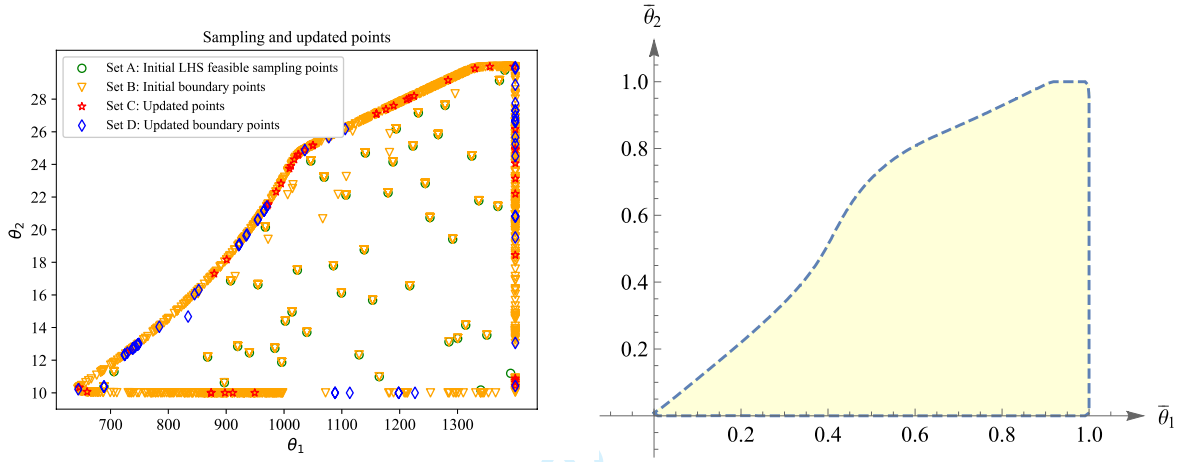
for a specific problem. The different values of ρ have been compared in the example and cases. A small value of ρ may lead to an overly conservative KS function, while too large a value of ρ may make the KS function ill-conditioned and cause unstable convergence and making it difficult to solve. Therefore, there is a tradeoff between accuracy of the KS function and ease of finding its surrogate model. A larger value of ρ can make the envelope closer to the real constrains, but possibly capture the undesired complex nonlinearity as well. It is not easy to create an accurate surrogate model for the KS function with complex or nonlinear structures. For simplicity, in this work, the value of ρ needs to be specified in advance.

7.2. Selection of NLP solvers

In the procedure of adaptive sampling, we propose two methods to expand the boundary points, as shown in Figure 7 and Figure 9. Because there must be two intersection points for a line and the boundary. The target is to locate both points; thus, the formula of the line should be defined by a parametric form, as shown in Equation (21). Both methods of expanding boundary points involve solving NLP models.

The real intersection points refer to the global optimal solutions, and common NLP solvers cannot guarantee finding the global optima. In the above cases, the global NLP solver, BARON, is adopted, and the results show that we can locate the boundary points effectively. Moreover, it is worth noting that, as the value of ρ and the number of inequalities increases, the KS function will be very complex and highly nonlinear, which will cause unstable convergence. Therefore, BARON is often time consuming. To ease the computational burden, it is necessary to compare the performance with a local solver. The local NLP solver in GAMS, CONOPT4, is selected to deal with Case 1. The results shown in Figure 17 indicate that this local NLP solver is also acceptable, and the design space can be described accurately. However, compared with Figure 15(a), Figure 17(a) shows that some points in set B overlap on the points in set A and some are located inside. The main reason is that the optimization model converges to the

local solutions. Nonetheless, we also can use these points to build the surrogate model, because these points are located within the design space. The CPU/Wall time for using CONOPT4 is 251.68s/476.16s, which is significantly less than using BARON. In addition, for an extremely complex and nonlinear model, BARON may require long computational times, and may not guarantee finding the global optimum for a fixed time limit. Therefore, there is also a tradeoff between the selection of solvers and the computational efficiency.



(a) (b)

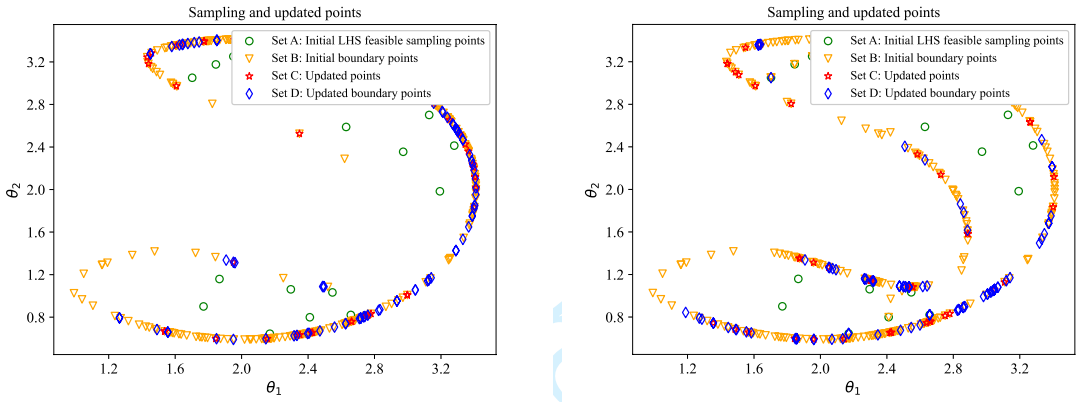
Figure 17. Results of design space description for Case 2. ($\rho = 10$, CONOPT4)

7.3. Effect of nonconvexity

The above numerical example and the Michael addition reaction case involve non-convex feasible spaces. Figure 2 indicates that the non-convex boundary can be described more accurately as the value of ρ increases. To further show the characteristics and performances of the proposed method, the example shown in Equation (6) is modified as Equation (37), which has a more nonconvex feasible region. The parameters are set to $\rho = 5, num = 80$.

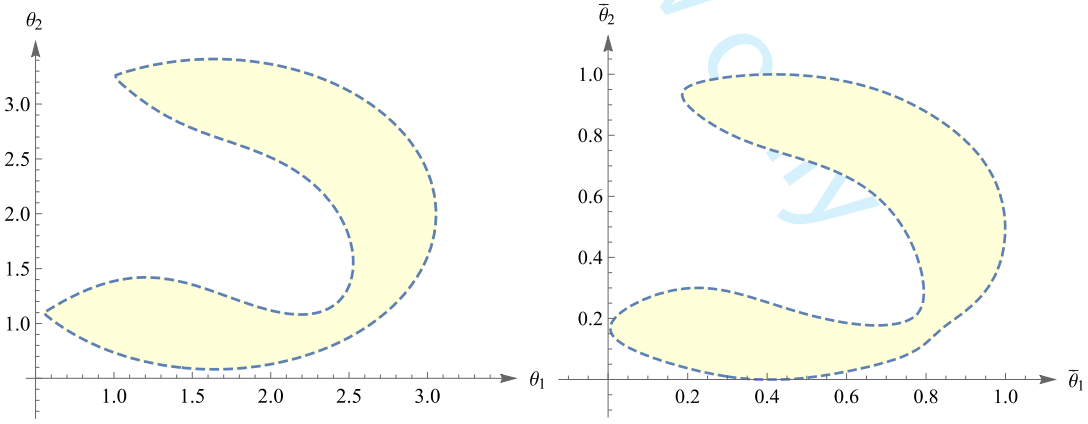
$$\begin{aligned} &g_1: -\left((\theta_2-2)^2+(\theta_1-2)^3+(\theta_2-2)(\theta_1-2)-0.5\right)\le 0\\ &g_2:(\theta_2-2)^2+(\theta_1-2)^2-2\le 0\\ &g_3:\theta_1-4\le 0\\ &g_4:-\theta_1\le 0\\ &g_5:\theta_2-4\le 0\\ &g_6:-\theta_2\le 0 \end{aligned} \tag{37}$$

In this case, BARON and CONOPT4 are compared. Figure 18(a) indicates that BARON cannot capture the inner nonconvex areas, because the solutions are always converged to the global maxima, i.e., the outermost boundary points. Thus, a large empty area on the boundary can be generated. By comparison, the local solver CONOPT4 can converge to local solutions, and the inner nonconvex areas can be described more completely, as shown in Figure 18(b). Based on these points, Figure 19 shows that the surrogate model can be created effectively. Therefore, the local solvers are more suitable for the problems with strongly nonconvex feasible regions.



(a) (b)

Figure 18. Sampling points of the modified example by using (a) BARON; (b) CONOPT4.



(a) (b)

Figure 19. Surrogate model of the modified example. (a) Real space; (b) KS space obtained by CONOPT4

8. Conclusions

In this paper, we propose a novel design space description method based on adaptive sampling and symbolic computation. The KS function is applied to aggregate all the inequality constraints, and the KS space can approximate the design space with a single constraint. Thus, we only need to focus on finding the sampling points on the boundary of the KS space. Based on the feasible LHS points and adaptive sampling points, two methods have been presented to effectively expand the set of boundary points. i.e., line projection to the boundary through any two feasible LHS points, and perturbation around the adaptive sampling points. The obtained polynomial surrogate model can be transformed into an equivalent triangular model through the CAD method, which can be further used to describe the KS space explicitly.

The case studies show that the proposed method is applicable to both convex and nonconvex feasible regions. Moreover, it is worth noting that, the CAD method is originally limited to polynomial functions, but the proposed method can address the inequality constraints with transcendental terms. This is because before performing the CAD method, the single KS function should be fitted as a multivariate polynomial by using the proposed adaptive sampling method. In addition, theoretically, the proposed method is more applicable to the cases with a small number of process parameters, because it is not easy to generate a single surrogate polynomial model in a high-dimensional space.

Acknowledgements

The authors gratefully acknowledge the financial support from Eli Lilly and Company, and the Center for Advanced Process Decision-making (CAPD) from Carnegie Mellon University.

Literature Cited

1 Hakemeyer C, McKnight N, John RS, Meier S, Trexler-Schmidt M, Kelley B, Zettl F, Puskeiler R, Kleinjans A, Lim F, Wurth, C. Process characterization and design space definition. *Biologicals*. 2016;44(5):306-318.

2 Kusumo KP, Gomoescu L, Paulen R, García-Muñoz S, Pantelides CC, Shah N, Chachuat B. Bayesian approach to probabilistic design space characterization: a nested sampling strategy. *Ind Eng Chem Res*. 2019;59(6):2396-2408.

3 Huang J, Kaul G, Cai C, Chatlapalli R, Hernandez-Abad P, Ghosh K, Nagi A. Quality by design case study: an integrated multivariate approach to drug product and process development. *Int J Pharm*. 2009;382(1-2):23-32.

4 Garcia-Munoz S, Luciani CV, Vaidyaraman S, Seibert KD. Definition of design spaces using mechanistic models and geometric projections of probability maps. *Org Process Res Dev*. 2015;19(8):1012-1023.

5 Goyal V, Ierapetritou MG. Determination of operability limits using simplicial approximation. *AIChE J*. 2002;48(12):2902-2909.

6 Rogers A, Ierapetritou MG. Feasibility and flexibility analysis of blackbox processes part 1: surrogate-based feasibility analysis. *Chem Eng Sci*. 2015;137:986-1004.

7 Rogers A, Ierapetritou MG. Feasibility and flexibility analysis of blackbox processes part 2: surrogate-based flexibility analysis. *Chem Eng Sci*. 2015;137:1005-1013.

8 Wang Z, Ierapetritou M. A novel feasibility analysis method for blackbox processes using a radial basis function adaptive sampling approach. *AIChE J*. 2017;63(2):532-550.

9 Metta N, Ramachandran R, Ierapetritou M. A novel adaptive sampling based methodology for feasible region identification of compute intensive models using artificial neural network. *AIChE J*. 2021;67(2):e17095.

10 Laky D, Xu S, Rodriguez JS, Vaidyaraman S, García-Muñoz S, Laird C. An optimization-based framework to define the probabilistic design space of pharmaceutical processes with model uncertainty. *Processes*, 2019;7(2):96.

11 Zhang Q, Grossmann IE, Lima RM. On the relation between flexibility analysis and robust optimization for linear systems. *AIChE J*. 2016;62(9):3109-3123.

12 Halemane KP, Grossmann IE. Optimal process design under uncertainty. *AIChE J*. 1983;29(3):425-433.

13 Swaney RE, Grossmann IE. An index for operational flexibility in chemical process design. Part I: Formulation and theory. *AIChE J*. 1985;31(4):621-630.

14 Swaney RE, Grossmann IE. An index for operational flexibility in chemical process design. Part II: Computational algorithms. *AIChE J*. 1985;31(4):631-641

15 Grossmann IE, Floudas CA. Active constraint strategy for flexibility analysis in chemical processes. *Comput Chem Eng*. 1987;11(6): 675-693.

16 Pulsipher JL, Zavala VM. A mixed-integer conic programming formulation for computing the flexibility index under multivariate gaussian uncertainty. *Comput Chem Eng*. 2018;119: 302-308.

17 Grossmann IE, Calfa BA, Garcia-Herreros P. Evolution of concepts and models for quantifying resiliency and flexibility of chemical processes. *Comput Chem Eng*. 2014;70:22-34.

18 Harwood SM, Barton PI. How to solve a design centering problem. *Math Methods Oper Res*. 2017;86(1):215-254.

19 Still G. Generalized semi-infinite programming: Theory and methods. *Eur J Oper Res*. 1999;119(2):301-313.

20 Vázquez FG, Rückmann JJ, Stein O, Still G. Generalized semi-infinite programming: a tutorial. *J Comput Appl Math*. 2008;217(2):394-419.

21 Zhao F, Chen X. Analytical and triangular solutions to operational flexibility analysis using quantifier elimination. *AIChE J*. 2018;64(11):3894-3911.

22 Collins GE. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. *Lect Notes Comput Sci*. 1975;33:134-183.

23 Zhao F, Zheng C, Zhang S, Zhu L, Chen X. Quantification of process flexibility via space projection. *AIChE J.* 2019;65(10):e16706.

24 Zheng C, Zhao F, Zhu L, Chen X. Operational flexibility analysis of high-dimensional systems via cylindrical algebraic decomposition. *Ind Eng Chem Res*. 2020;59(10):4670-4687.

25 Collins GE. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. *Lec Notes Comp Sci*. 1975;33:134-183.

26 Kreisselmeier G, Steinhauser R. Systematic control design by optimizing a vector performance index. Proc. *IFAC Symp. On Computer-Aided Design of Control Systems*. Zurich, Switzerland, 1979.

27 Rooney WC, Biegler LT. Optimal process design with model parameter uncertainty and process variability. *AIChE J.* 2003;49(2):438-449.

28 Martins JRRA, Alonso JJ, Reuther JJ. High-fidelity aerostructural design optimization of a supersonic business jet. *J Aircr*. 2004;41(3):523-530.

29 Brooks TR, Martins JRRA, Kennedy GJ. High-fidelity aerostructural optimization of tow-steered composite wings. *J Fluids Struct*. 2019;88:122-147.

30 Gelfand IM, Kapranov MM, Zelevinsky AV. Discriminants, resultants, and multidimensional determinants. Birkhäuser: Theory & Applications, 1994.

31 Sagraloof M, Mehlhorn K. Computing real roots of real polynomials. *J Symb Comput*. 2016; 73:46-86.

32 Wolfram Research Inc. Mathematica Version 12.2. Champaign, IL. 2020.

33 Deutsch JL, Deutsch CV. Latin hypercube sampling with multidimensional uniformity. *J Stat Plan Inference*. 2012;142(3):763-772.

34 Zhang P. Model selection via multifold cross validation. *Ann stat*. 1993;21(1):299-313.

35 Powell MJD. The BOBYQA Algorithm for Bound Constrained Optimization Without Derivatives. Technical report, Department of Applied Mathematics and Theoretical Physics,

University of Cambridge. 2009.

36 Hart WE, Watson JP, Woodruff DL. Pyomo: modeling and solving mathematical programs in Python. *Math Program Comput.* 2011;3(3):219-260.

37 Chen W, Biegler LT, García-Muñoz S. An approach for simultaneous estimation of reaction kinetics and curve resolution from process and spectral data. *J Chemom*. 2016;30(9):506-522.