Semi-Infinite Optimization with Hybrid Models[†]

Chenyu Wang, Matthew E. Wilhelm, and Matthew D. Stuber*

Process Systems and Operations Research Laboratory, Department of Chemical & Biomolecular Engineering, University of Connecticut, 191 Auditorium Road, Unit 3222, Storrs, CT 06269, USA.

E-mail: stuber@alum.mit.edu

Abstract

The robust design of performance/safety-critical process systems, from a model-based perspective, remains an existing challenge. Hybrid first-principles data-driven models offer the potential to dramatically improve model prediction accuracy, stepping closer to the digital twin concept. Within this context, worst-case engineering design feasibility and reliability problems give rise to a class of semi-infinite program (SIP) formulations with hybrid models as coupling equality constraints. Reduced-space deterministic global optimization methods are exploited to solve this class of SIPs to ϵ-global optimality in finitely many iterations. This approach is demonstrated on two challenging case studies: a nitrification reactor for a wastewater treatment system to address worst-case feasibility verification of dynamical systems; and a three-phase separation system plagued by numerical domain violations to demonstrate how they can be overcome using a nonsmooth SIP formulation with hybrid models and a validity constraint incorporated.

[†]Author's final accepted version. Published version: Wang, C., Wilhelm, M.E., and M.D. Stuber. Semi-Infinite Optimization with Hybrid Models. *Industrial & Engineering Chemistry Research.* 61, 5239–5254 (2022). DOI: doi:10.1021/acs.iecr.2c00113

Introduction

Many engineering systems are deemed safety-critical and, as such, require strict guarantees of performance and safety. Uncertainties, such as those introduced by inaccurate data, should be accounted for at the design stage of such systems. Therefore, it is necessary to identify the worst-case performance of these systems to mitigate the impacts of uncertainty on the final design. For example, in many energy-related applications, the costs associated with operational failures are extremely high; often including loss of life, substantial environmental damage, severe economic damage, and major sociopolitical fallout. From a model-based perspective, approaching design problems of this nature amounts to identifying realizations of uncertainty that result in a simulated worst-case violation of performance/safety constraints as governed by a system model. As such, deterministic global optimization methods are required to guarantee worst-case realizations of uncertainty may be identified in the general case.

Worst-case design problems have historically been treated as bilevel or more general multilevel programs. These programs have feasible sets that are characterized by other optimization problems. As such, these programs are extremely challenging or even impossible to solve directly using existing methods. Thus, early studies focused on the simplest cases of worst-case design problems with linearity and convexity conditions.[1,2] Over the years, relevant studies were extended to more complicated worst-case design problems with nonlinearity.[3–5]

Gümüş and Floudas[6] developed a global optimization algorithm based on relaxations of the feasible region for solving worst-case design problems whose bilevel formulations involve twice-differentiable nonlinear functions. A transformation was proposed to replace the inner problem with its KKT optimality conditions, transforming the inner program into nonlinear algebraic constraints under the linear independence constraint qualification. This approach requires convexity for the KKT conditions to be necessary and sufficient, however general nonconvex functions were considered by exploiting αBB relaxations within a branch-and-bound framework for the solution of the KKT-reformulated NLP. Feasibility and flexibility index prob-

lems were considered within this context in a follow-up work.[7] However, this approach cannot provide valid convergent upper bounds for bilevel programs with nonconvex inner programs, in general.[8] Mitsos et al.[8] proposed a bounding algorithm to resolve this problem that can solve nonlinear bilevel programs to global optimality without any convexity assumptions. However, the approach is limited to only considering inequality constraints (see Mitsos et al.[8], Assumption 3).

As an alternative strategy to solving bilevel programs, multiparametric programming was developed by recasting them into single-level deterministic optimization problems.[9,10] This strategy is unique in that the parametric solution of the inner program is characterized explicitly and therefore can be utilized in real-time optimization applications. However, the developed methods require the inner programs to be linear or quadratic programs.[10] For general nonconvex objectives and general nonlinear and nonconvex inner programs, multiparametric programming is not applicable. In this work, we investigate the most general worst-case design problems that may involve nonlinear coupling equality constraints, and in doing so we consider the methods that reformulate bilevel programs as equivalent semi-infinite programs (SIPs).

The solution of general nonconvex SIPs has been an active area of research in recent years, yielding approaches that perform well for solving classes of worst-case design problem formulations. Many of the recent advancements have been based on the discretization-based cutting-plane algorithm developed by Blankenship and Falk[11]. Mitsos[12] developed an algorithm with a new procedure for feasible point generation by setting a restriction condition of right-hand side of the semi-infinite constraints. Stuber and Barton[13] developed a modified version of the SIP algorithm proposed by Mitsos[12] and finally extended the method to the most general nonconvex case accounting for semi-infinite equality constraints without assuming that they admit closed-form parametric solutions.

Djelassi and Mitsos[14] developed a hybrid discretization-based algorithm for the global solution of SIPs without semi-infinite equality constraints. The algorithm proposed by Mitsos[12] is employed for upper-bounding and lower-bounding problems, and an oracle problem adapted

from the algorithm proposed by Tsoukalas and Rustem [15] is employed to generate cheap lower bounds and adaptive updates to the restriction of this algorithm. The hybrid algorithm can avoid a dense population of the discretization, and has superior computational performance as a result.

Solution methods for higher-complexity SIP formulations have also been studied. The algorithm of Mitsos [12] was extended to generalized semi-infinite programs (GSIPs) by Mitsos and Tsoukalas [16]. Djelassi et al. [17] then extended this GSIP algorithm [16] and considered the mixed-integer bilevel program to allow the presence of coupling equality constraints. In their method, a subset of the lower-level variables are treated as dependent variables to cope with convergence issues introduced by coupling equality constraints. This algorithm requires an increase in the dimensionality of continuous variables for some subproblems, but the performance penalty was not observed in their numerical experiments. Djelassi and Mitsos [18] most recently proposed an algorithm for the global solution of existence-constrained SIPs (ESIPs) that are a generalization of standard SIPs with three levels. This is the first algorithm that can solve ESIPs to global optimality without any convexity assumptions. Some other recent developments in nonconvex SIP applications and algorithms have been reviewed by Djelassi et al. [19].

A key concern pertinent to many applications of SIPs within engineering design, is the need for high-accuracy and low computational complexity models of safety-critical systems (whose performance must satisfy strict requirements).[20] In many cases, strict performance/safety requirements must be satisfied over a range of potential input disturbances or process noise. While it is possible that such disturbances and noise may be well characterized for some cases, this is not often typical for nascent designs. In addition, many process systems models involve implicit functions as their nonlinearity prohibits explicit closed-form solutions. Even though a method for solving SIPs with implicit functions has been developed by Stuber and Barton [13], the algorithm is computationally expensive and high-complexity models compound the computational cost. Hybrid modeling approaches are attractive here because they can accurately represent complicated process systems that are not fully understood, and may also reduce the

mathematical complexities caused by implicit functions, nonlinearity, complicated dynamics, and multivariate uncertainty. Thus, the central goal of our work is to utilize hybrid models within SIP formulations for the optimal design, simulation, and robustness verification of process systems (i.e., process systems that satisfy all predetermined performance/safety requirements[21]) in the face of worst-case uncertainty.

Hybrid models consist of structured combinations of rigorous first-principles models (FPMs) that account for necessary/known system mechanisms and empirical or data-driven models (DDMs) that describe phenomena that cannot be readily described using FPMs due to a lack of adequate knowledge.[22] Over the past few decades, the use of hybrid modeling approaches, particularly those that exploit machine learning approaches, have found a wide variety of applications in the process systems engineering community. These methods have enhanced process output,[20] improved controller performance,[23,24] and enabled integrated system-level designs of highly complex processes.[25,26] In this paper, we explore applications of hybrid models to worst-case design problems to investigate and verify their applicability in SIPs.

SIPs governed by hybrid models are of particular importance in process systems engineering,[26–28] yet their usage within general (nonconvex) SIP contexts remains absent. This is likely a consequence of the coupling equality constraints introduced by the FPMs that significantly complicate the problem. In this paper, we propose addressing these gaps with the following main novel contributions:

1. We formalize the approach to use hybrid models with SIPs. One application of this approach is to resolve complications due to coupling equality constraints via a *reduced-space* formulation.

2. We illustrate how this SIP formulation that incorporates hybrid models is sufficiently general such that it may be readily applied to exemplary robust design problems incorporating process dynamics.

3. We present a hybrid modeling approach that resolves numerical issues relating to domain

violations, ubiquitous in process systems engineering modeling and simulation, through use of a novel nonsmooth SIP formulation that incorporates validity constraints.

In this paper, we present new developments on the formulation and solution of SIPs with embedded hybrid models. Particular attention is paid to the models that use artificial neural networks (ANNs) with activation functions that are of interest for deep learning as the DDM. In the following sections, we detail: the mathematical conventions used in the paper (Mathematical Background), formalize optimization problems with hybrid models (Optimization of Hybrid Models); formalize SIPs with hybrid models embedded and present a solution algorithm (Semi-Infinite Optimization with Hybrid Models); present case studies that demonstrate a variety of optimization under uncertainty problems formulated as SIPs with hybrid models embedded (Case Studies); and extend the proposed approach to SIPs with implicit functions embedded (Extension to Implicit Forms). Finally, we suggest future directions for subsequent research.

Mathematical Background

In this section, the necessary mathematical preliminaries for the framework of SIPs with hybrid models are introduced.

Multilayer Perceptrons

Several DDM methods have been developed and applied to a broad range of process systems, such as support vector machines,[29] random forests,[30] and ANNs.[31,32] In this work, ANNs are utilized as a representative DDM approach to demonstrate the formulation of SIPs with hybrid models, and the corresponding notation is formalized in this section accordingly. The multilayer perceptron (MLP) is one of the most common classes of ANN structures. As illustrated in Figure 1, the MLP is composed of a directed acyclic graph (DAG) containing n layers enumerated $k = 1, \ldots, n$. The first layer with $k = 1$ represents the inputs of the MLP, whereas the

last layer with $k = n$ corresponds to the *output layer*. The $k = 2, \ldots, n-1$ layers are the *hidden layers*. Let $m^{(k)}$ be the number of neurons in layer k, $\mathbf{a}^{(k)} \in \mathbb{R}^{m^{(k)}}$ be the outputs of layer k. As defined, $\mathbf{a}^{(1)}$ is the input vector and $\mathbf{a}^{(n)}$ is the output vector of the MLP. The vector $\mathbf{a}^{(k)}$ for hidden layers $k \in \{2, \ldots, n\}$ is defined as

$$\mathbf{a}^{(k)} = f^{(k)}\left(\mathbf{W}^{(k-1)}\mathbf{a}^{(k-1)} + \mathbf{b}^{(k-1)}\right), \tag{1}$$

where $f^{(k)} : \mathbb{R} \to \mathbb{R}$ is an activation function, $\mathbf{W}^{(k-1)} \in \mathbb{R}^{m^{(k)} \times m^{(k-1)}}$ is a *weight matrix*, and $\mathbf{b}^{(k-1)} \in \mathbb{R}^{m^{(k)}}$ is a *bias vector*. For ease of introduction, we define $\mathbf{o} : \mathbb{R}^{m^{(1)}} \to \mathbb{R}^{m^{(n)}}$ as the representative input-output function for a generic DDM. Thus, as for a MLP, $\mathbf{a}^{(n)} = \mathbf{o}(\mathbf{a}^{(1)})$.

When training MLPs, the weight matrices and bias vectors are regarded as optimization variables whereas the input values of $\mathbf{a}^{(1)}$ are taken as parameters. When using fully-trained MLPs in a hybrid model for simulation or optimization, the weight matrices and bias vectors are fixed to the trained constant parameters. The notation of MLPs in this section is used for a typical class of DDMs that will be used for hybrid model formulations.

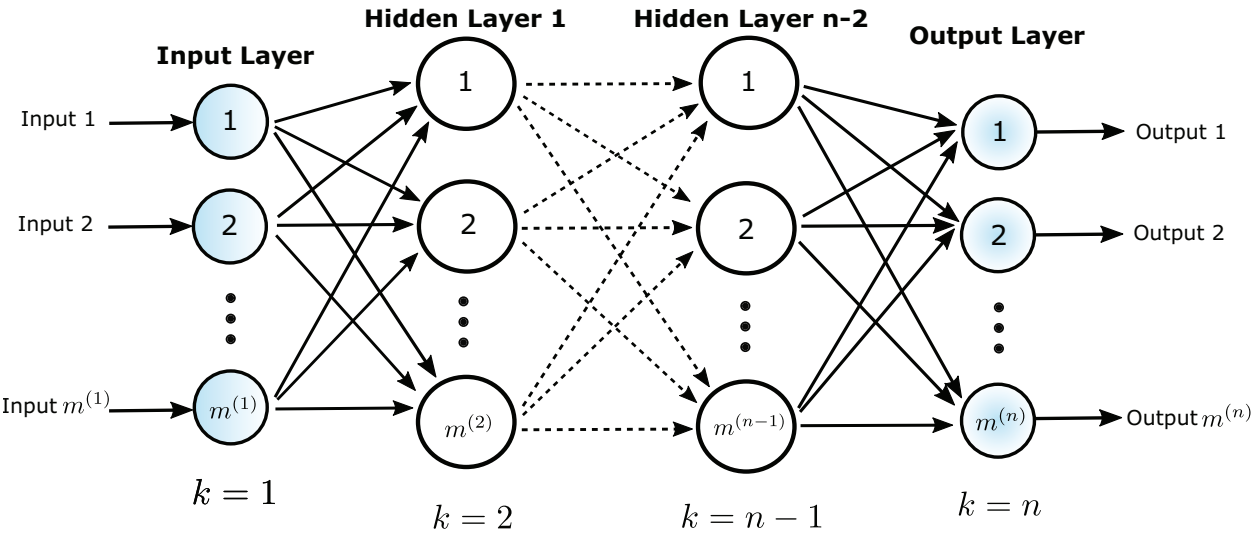


Figure 1: A multilayer perceptron with n layers is illustrated as a directed acyclic graph. The input layer corresponds to $k = 1$, the hidden layers correspond to $k = 2, \ldots, n-1$, and the output layer corresponds to $k = n$. The multilayer perceptron has a fully-connected feed-forward network where all neurons in last layer, $k-1$, are related to all neurons in the subsequent layer, k.

Hybrid Model Architecture

In this section, we formalize the notation for first-principles and data-driven sub-models as illustrated in Figure 2. A vector of independent input variables is defined as $\mathbf{y} \in Y \subset \mathbb{R}^{n_y}$, a vector $\hat{\mathbf{z}}^{FPM} \in Z^{FPM} \subset \mathbb{R}^{n_{zf}}$ represents state variables governed by the FPMs, $\mathbf{h}^{FPM} : Z^{FPM} \times Z^{DDM} \times Y \rightarrow \mathbb{R}^{n_{zf}}$ represent FPM equations, $\hat{\mathbf{z}}^{DDM} \in Z^{DDM} \subset \mathbb{R}^{n_{zd}}$ is a vector of output variables of the DDMs. Note that in the scope of this work, we consider MLPs as explicit input-output DDMs that can be represented by $\mathbf{a}^{(n)} = \mathbf{o}(\mathbf{a}^{(1)})$ as previously defined.

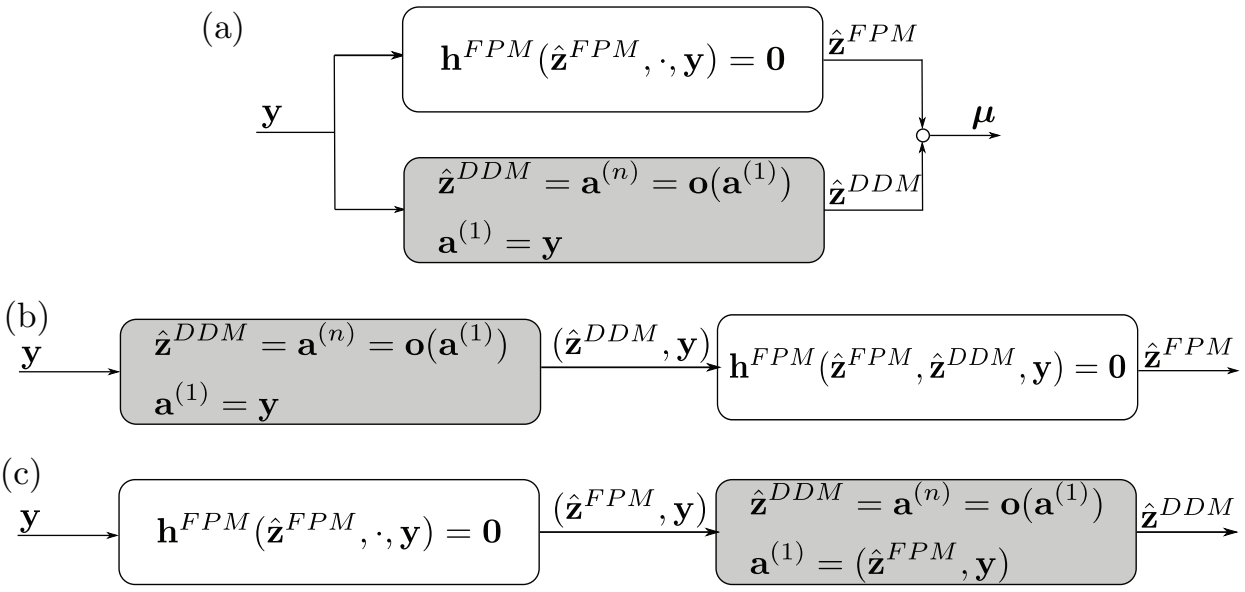


Figure 2: Flow diagrams of typical hybrid model architectures are presented in this figure (white blocks represent first-principles sub-models and shaded blocks represent data-driven sub-models). (a) A parallel hybrid model architecture maps inputs to outputs of each model type in parallel. (b) A DDM/FPM serial hybrid model architecture maps inputs of the DDM to outputs that are subsequently used as inputs in the FPM. (c) A FPM/DDM serial model architecture maps inputs to outputs of the FPM that are subsequently used as inputs to the DDM.

In general, the architecture of hybrid models is classified according to the parallel and/or serial arrangement of sub-models (see von Stosch et al.[33] for a thorough review and discussion of hybrid model architectures). The mathematical structure of a parallel hybrid model is illustrated in Figure 2(a). In this formulation, the FPM $\mathbf{h}^{FPM}(\hat{\mathbf{z}}^{FPM}, \cdot, \mathbf{y}) = \mathbf{0}$ does not have explicit dependence on $\hat{\mathbf{z}}^{DDM}$ and $\boldsymbol{\mu}$ represents the final outputs of the parallel hybrid model that

can be expressed as $\boldsymbol{\mu} = \psi(\hat{\mathbf{z}}^{FPM}, \hat{\mathbf{z}}^{DDM})$, where $\psi : \mathbb{R}^{n_{zf}} \times \mathbb{R}^{n_{zd}} \rightarrow \mathbb{R}^{n_\mu}$ represents some functional relationship involving the states of the FPMs and outputs of the DDMs. In cases employing a parallel model architecture, the FPMs may not be able to accurately capture some of the phenomena observed in real systems, resulting in a discrepancy. In this situation, a DDM can be utilized to rectify the mismatch between the prediction of FPM and the observed process data.[27]

In the serial architecture, the output of the first sub-model is taken as an intermediate variable that is input to the second sub-model. The DDM/FPM serial architecture is the most common hybrid model architecture,[33] illustrated in Figure 2(b). In chemical engineering systems, FPMs typically involve conservation laws that may have extremely complicated mathematical expressions and/or source terms that may fail to accurately capture some observed system behavior due to an incomplete understanding of underlying mechanisms. In these situations, DDMs can be used as a surrogate model to represent intractable parameters and/or subexpressions. Alternatively, the FPM/DDM serial architecture is shown in Figure 2(c). This architecture can be used to model a system whose intermediate variables are governed by the first-principles model $\mathbf{h}^{FPM}(\hat{\mathbf{z}}^{FPM}, \cdot, \mathbf{y}) = \mathbf{0}$ (with no explicit dependence on $\hat{\mathbf{z}}^{DDM}$) and the final outputs are some process parameters that are related to the intermediate state variables.[34] In the next section, the notation pertaining to hybrid model architectures in embedded optimization formulations is established.

Optimization of Hybrid Models

In this section, optimization problems with hybrid ANN models embedded are formalized. We use a general formulation to maintain applicability to a wide variety of surrogate modeling approaches being actively explored by machine learning researchers (e.g., Ghaoui et al.[35], Bai et al.[36], Winston and Kolter[37], Gu et al.[38]) including neural ordinary differential equations[39] inspired by the success of ResNet.[40] In general, a conventional formulation of an optimiza-

tion problem involving hybrid models accounts for the modeling equations as explicit equality constraints. Thus, let the optimization formulation with hybrid models be represented by the following nonlinear program (NLP):

$$
\begin{aligned}
&\min_{\mathbf{y},\hat{\mathbf{z}}^{FPM},\hat{\mathbf{z}}^{DDM}} \phi(\hat{\mathbf{z}}^{FPM},\hat{\mathbf{z}}^{DDM},\mathbf{y}) \\
&\text{s.t. } \mathbf{h}^{FPM}(\hat{\mathbf{z}}^{FPM},\hat{\mathbf{z}}^{DDM},\mathbf{y}) = \mathbf{0} \\
&\quad \mathbf{h}^{DDM}(\hat{\mathbf{z}}^{FPM},\hat{\mathbf{z}}^{DDM},\mathbf{y}) = \mathbf{0} \\
&\quad \mathbf{g}(\hat{\mathbf{z}}^{FPM},\hat{\mathbf{z}}^{DDM},\mathbf{y}) \leq \mathbf{0} \\
&\quad \mathbf{y} \in Y \in \mathbb{IR}^{n_y} \\
&\quad \hat{\mathbf{z}}^{FPM} \in Z^{FPM} \subset \mathbb{R}^{n_{zf}} \\
&\quad \hat{\mathbf{z}}^{DDM} \in Z^{DDM} \subset \mathbb{R}^{n_{zd}},
\end{aligned}
\tag{2}
$$

where $\mathbb{IR}^n$ is the set of all n-dimensional real intervals, the decision variables consist of the independent input variables $\mathbf{y}$ (e.g., design variables), the output variables of an ANN $\hat{\mathbf{z}}^{DDM}$ as previously defined, and the system state variables $\hat{\mathbf{z}}^{FPM}$ that are determined by the FPMs. It is assumed that the objective function $\phi : Z^{FPM} \times Z^{DDM} \times Y \to \mathbb{R}$ and the inequality constraints $\mathbf{g}$: $Z^{FPM} \times Z^{DDM} \times Y \to \mathbb{R}^{n_g}$ are continuous. The equality constraints $\mathbf{h}^{DDM} : Z^{FPM} \times Z^{DDM} \times Y \to \mathbb{R}^{n_{zd}}$ are expressed in standard form and represent the DDM equations (i.e., the ANN equations $\mathbf{h}^{DDM}(\hat{\mathbf{z}}^{FPM},\hat{\mathbf{z}}^{DDM},\mathbf{y}) = \hat{\mathbf{z}}^{DDM} - \mathbf{a}^{(n)} = \mathbf{0}$). The equality constraints $\mathbf{h}^{FPM} : Z^{FPM} \times Z^{DDM} \times Y \to \mathbb{R}^{n_{zf}}$ and $\mathbf{h}^{DDM} : Z^{FPM} \times Z^{DDM} \times Y \to \mathbb{R}^{n_{zd}}$ are also assumed to be continuous. In general, bounds on $\mathbf{y}$, $\hat{\mathbf{z}}^{DDM}$, and $\hat{\mathbf{z}}^{FPM}$ must be supplied to ensure that the problem is well-posed, although some variables may not require bounds known *a priori*.

The general optimization formulation (2) can be reformulated compactly as follows. Define $\hat{\mathbf{z}} = (\hat{\mathbf{z}}^{FPM},\hat{\mathbf{z}}^{DDM})$, $Z = Z^{FPM} \times Z^{DDM}$, and let $\mathbf{h} : Z \times Y \to \mathbb{R}^{n_z}$ be the concatenation of $\mathbf{h}^{FPM}$ and $\mathbf{h}^{DDM}$ such that $\mathbf{h}(\hat{\mathbf{z}},\mathbf{y}) = \left(\mathbf{h}^{FPM}(\hat{\mathbf{z}}^{FPM},\hat{\mathbf{z}}^{DDM},\mathbf{y}),\mathbf{h}^{DDM}(\hat{\mathbf{z}}^{FPM},\hat{\mathbf{z}}^{DDM},\mathbf{y})\right)$. Then, (2) can be

reformulated as the following NLP:

$$\begin{aligned} \min_{\mathbf{y}\in Y,\hat{\mathbf{z}}\in Z} \ & \phi(\hat{\mathbf{z}},\mathbf{y}) \\ \text{s.t. } & \mathbf{h}(\hat{\mathbf{z}},\mathbf{y}) = \mathbf{0} \\ & \mathbf{g}(\hat{\mathbf{z}},\mathbf{y}) \leq \mathbf{0}. \end{aligned} \tag{3}$$

In many cases that arise naturally from process flowsheet simulation, model inputs and parameters (e.g., process design specifications, controllable inputs) define unique state conditions by continuity equations (i.e., conservation laws) as equality constraints that can be solved explicitly. Thus, in this paper, we assume that there exists a unique explicit closed-form function $\mathbf{z} : Y \rightarrow Z$ such that $\mathbf{h}(\mathbf{z}(\mathbf{y}),\mathbf{y}) = \mathbf{0}$ for every $\mathbf{y} \in Y$. Under this assumption, the equality constraints can be eliminated and (3) can be simplified as:

$$\begin{aligned} \min_{\mathbf{y}\in Y} \ & \phi(\mathbf{z}(\mathbf{y}),\mathbf{y}) \\ \text{s.t. } & \mathbf{g}(\mathbf{z}(\mathbf{y}),\mathbf{y}) \leq \mathbf{0}. \end{aligned} \tag{4}$$

Since the equality constraints of (3) are entirely eliminated in this formulation, there is a (significant) reduction in problem dimensionality. Although we assumed the existence of explicit functions for the proposed approach, this does not restrict the method. The section **Extension to Implicit Forms** discusses how this assumption may be relaxed allowing for the proposed approach to be applied to more general hybrid models with implicit forms.

Remark. Note that the uniqueness assumption is required for the elimination of the coupling equality constraints from the original problem formulation (2). This assumption and approach have been commonly made in practice for addressing design under uncertainty problems (e.g., Stuber and Barton [13], Kwak and Haug [41], Halemane and Grossmann [42], Ostrovsky et al. [43], Dimitriadis and Pistikopoulos [44], Hale et al. [45], among others), and is not presented as a new approach here. However, in case of nonunique parametric solutions (e.g., multiple steady states),

without special consideration this approach would effectively restrict the feasible set. For example, consider the model $h(\hat{z}, y) = \hat{z}^2 - y = 0$. This problem has explicit closed-form solutions $z_1(y) = \sqrt{y}$ and $z_2(y) = -\sqrt{y}$ with $y = 0$ a bifurcation point. To ensure that a global solution of the original problem is obtained, both parametric solution branches z_1 and z_2 must be considered or else feasible solutions may be ignored. Within the context of hybrid models, nonuniqueness may be less of a concern since DDMs are trained as explicit input-output mappings representing a system or phenomena of interest and, when coupled to FPMs, are likely to force adherence to a single solution branch. For problems where this is not the case, each solution branch of the FPM would need to be considered separately, as in the simple example above. For parametric dynamical systems, relatively mild assumptions ensure the existence and uniqueness of parametric solution trajectories.[46]

Reduced-space approaches to deterministic global optimization originated from Epperly and Pistikopoulos[47], who detailed a convergent branch-and-bound (B&B) algorithm that branched only on a subset of the decision variables. This *reduced-space* formulation approach was subsequently generalized for many different problem and model types (e.g., by Stuber[48], Sec. 4.1, Mitsos et al.[49], Wechsung[50], Bongartz and Mitsos[51]). This approach avoids the introduction and explicit handling of auxiliary variables and equality constraints through intermediate calculations by treating the independent input variables $\mathbf{y}$ as the only decision variables of the optimization problem. Since $n_y << n_z$ in most process systems engineering problems, (4) represents a significantly lower-dimensionality problem than (3). Due to the curse of dimensionality in deterministic global optimization, this reduction in dimensionality often translates to a significant reduction in the solution time.

In general, formulations (2)-(4) are nonconvex optimization problems that are solved to guaranteed global optimality via a variation of the spatial B&B algorithm.[52–54] This consists of a presolve step, followed by the successive solution of lower- and upper-bounding problems with intermediate domain reduction. An upper bound is typically determined by solving the original nonconvex problem to either feasibility or local optimality. This is distinct from the

subproblems encountered in domain reduction and the lower bounding routines that construct and solve relaxations of the nonconvex problem through the computation of convex relaxations of the nonconvex objective and constraint functions.[55]

Within this reduced-space context, researchers have addressed the construction of convex and concave relaxations of factorable functions[49,56–58] (i.e., a function defined by a finite recursive composition of sums, products, and univariate transcendental functions) as well as specific classes of functions that break the factorability assumption. Methods for computing relaxations of parametric solutions of differential equations[46,59,60] as well as implicit functions evaluated by fixed-point methods, have both been detailed.[61] Provided that relaxations of intermediate terms may be computed, these relaxations may be readily composed in a generalized framework.[58] For instance, relaxations of the solutions of parametric differential equations may be computed provided that convex/concave relaxations of the right-hand side function are known, and then composed with an algebraic objective or constraint term. As such, this modeling framework is generally applicable to the preponderance of hybrid model architectures.

There are several existing deterministic global optimization solvers capable of addressing general problems formulated as (3). These include commercially licensed offerings such as BARON[53] and ANTIGONE,[62] as well as open-source offerings such as EAGO[63] and MAiNGO.[64] Due to limitations in how problems are represented and how relaxations of nonconvex functions are constructed, BARON[53] and ANTIGONE[62] cannot address formulation (4). Alternatively, EAGO[63] and MAiNGO[64] were developed with this class of problems in mind with more flexible modeling requirements and advanced methods for constructing relaxations of nonconvex functions. Due to the high dimensionality of formulation (3) and the curse of dimensionality in deterministic global optimization, excessively long run times are expected for solving (3). It has been demonstrated through several examples[32,47,51,56,57,61,65] that an equivalent *reduced-space* problem (4) can dramatically reduce the run time of a compatible algorithm by dramatically reducing the number of variables branched on. Moreover, the elimination of equality constraints from (3) plays a particularly important role in ensuring that SIPs of interest in the

subsequent section, are formulated in a readily solvable manner.

Semi-Infinite Optimization with Hybrid Models

In this section, the foundations for incorporating hybrid models into SIP formulations are formalized. First, consider the input variables of a hybrid model partitioned as $\mathbf{y} = (\mathbf{x}, \mathbf{p})$ with the corresponding domain $Y = X \times P$ with $\mathbf{x} \in X \in \mathbb{IR}^{n_x}$ and $\mathbf{p} \in P \in \mathbb{IR}^{n_p}$. Then, the general form of an SIP governed by a hybrid model can be expressed as:

$$\begin{aligned} \phi^* = \quad & \min_{\mathbf{x} \in X} \phi(\mathbf{x}) \\ \text{s.t. } & g(\hat{\mathbf{z}}, \mathbf{x}, \mathbf{p}) \leq 0, \ \forall \mathbf{p} \in P \\ & \mathbf{h}(\hat{\mathbf{z}}, \mathbf{x}, \mathbf{p}) = \mathbf{0}, \ \forall \mathbf{p} \in P. \end{aligned} \tag{5}$$

In this formulation, $\mathbf{x}$ represents a vector of decision variables, $\hat{\mathbf{z}}$ represents a vector of internal state variables governed by the hybrid model equations (as introduced previously), and $\mathbf{p}$ represents a vector of parameters. The objective function $\phi : X \to \mathbb{R}$ depends solely on the variables $\mathbf{x} \in X$ and the constraints $g : Z \times X \times P \to \mathbb{R}$ and $\mathbf{h} : Z \times X \times P \to \mathbb{R}^{n_z}$ are parameterized by $\mathbf{p} \in P$. Note that the hybrid model $\mathbf{h}$ is defined as in the previous section. It is assumed that the objective function ϕ, semi-infinite inequality constraint function g, and equality constraint function $\mathbf{h}$ are factorable and continuous on their domains. Note that we have made no assumptions about the smoothness of g. Therefore, multiple performance constraints $g_1, \ldots, g_n$, may be handled trivially by reformulation into a single constraint $g = \max_i g_i$.

As in formulation (3), the state variables $\hat{\mathbf{z}}$ in (5) are governed by continuous hybrid model functions $\mathbf{h}(\hat{\mathbf{z}}, \mathbf{x}, \mathbf{p}) = \mathbf{0}$ for each $(\mathbf{x}, \mathbf{p}) \in X \times P$. Similar to formulation (4), we assert that the hybrid model functions $\mathbf{h}(\hat{\mathbf{z}}, \mathbf{x}, \mathbf{p}) = \mathbf{0}$ can be solved explicitly. Thus, the state variables can be expressed as an explicit input-output mapping $\mathbf{z} : X \times P \to \mathbb{R}^{n_z}$ such that $\mathbf{h}(\mathbf{z}(\mathbf{x}, \mathbf{p}), \mathbf{x}, \mathbf{p}) = \mathbf{0}$ for

every $(\mathbf{x}, \mathbf{p}) \in X \times P$. Under these assumptions, (5) can be reformulated as:

$$\begin{aligned} \phi^* = \quad & \min_{\mathbf{x} \in X} \phi(\mathbf{x}) \\ & \text{s.t. } g(\mathbf{z}(\mathbf{x}, \mathbf{p}), \mathbf{x}, \mathbf{p}) \leq 0, \ \forall \mathbf{p} \in P. \end{aligned} \tag{6}$$

The SIP formulation covers classes of robust design and optimization under uncertainty problems of specific interest in this work.

Remark. Note that the assumption of uniqueness of $\mathbf{z}(\cdot) \in Z$ for every $\mathbf{y} \in Y$ (i.e., $\forall (\mathbf{x}, \mathbf{p}) \in X \times P$) was discussed in the previous section as a requirement for the elimination of the coupling equality constraints. Within the SIP context for robustness verification, caution must be exercised to ensure that uniqueness can be verified. For systems with multiple solution branches present, an SIP must be solved with respect to each physically meaningful solution branch. However, as remarked in the previous section, nonuniqueness is expected to be rare for systems of interest with hybrid modeling approaches and not an issue for dynamical systems under relatively mild assumptions.

Three problem types that fall under the general formulation (6) will be considered in this work for their relevance in the design of safety-critical systems. The first problem is the design under uncertainty feasibility problem. The goal with this problem is to confirm whether there exists a design that is robust to a worst-case realization of parametric uncertainty:

$$\begin{aligned} \eta^* = \quad & \min_{\mathbf{d} \in D, \eta \in H} \eta \\ & \text{s.t. } \eta \geq g(\mathbf{z}(\mathbf{d}, \boldsymbol{\pi}), \mathbf{d}, \boldsymbol{\pi}), \ \forall \boldsymbol{\pi} \in \Pi. \end{aligned} \tag{7}$$

Here, $\mathbf{d} \in D \in \mathbb{R}^{n_d}$ represents a vector of design variables, $\boldsymbol{\pi} \in \Pi$ represents uncertain parameters in the hybrid model, and $\eta \in H \in \mathbb{R}$ represents a measure of robust feasibility. With respect to the SIP formulation (6), we have $\mathbf{x} = (\mathbf{d}, \eta)$, $X = D \times H$, $\mathbf{p} = \boldsymbol{\pi}$, and $P = \Pi$. If the optimal solution value of the feasibility problem satisfies $\eta^* \leq 0$, then a design exists that is robust to worst-case

realizations of uncertainty.

The second problem of consideration is the robust optimal design problem. The objective function is directly defined as a cost function based on a technical or economic objective (e.g., total capital cost or process efficiency):

$$\begin{aligned} \phi^* = \quad & \min_{\mathbf{d}\in D} \phi(\mathbf{d}) \\ & \text{s.t. } g(\mathbf{z}(\mathbf{d},\boldsymbol{\pi}),\mathbf{d},\boldsymbol{\pi}) \leq 0,\ \forall \boldsymbol{\pi} \in \Pi. \end{aligned} \tag{8}$$

Here, with respect to formulation (6), we have $\mathbf{x} = \mathbf{d}, \mathbf{p} = \boldsymbol{\pi}, X = D$, and $P = \Pi$. A global optimal solution of this problem will be an optimal system design that is robust to worst-case realizations of uncertainty (if such a design exists).

The last problem of consideration is the operation under uncertainty feasibility problem. This formulation is used to determine whether there exist control settings or recourse such that the system of interest will always satisfy the performance and/or safety specifications. The problem is formulated as:

$$\begin{aligned} \eta^* = \quad & \max_{\boldsymbol{\pi}\in\Pi, \eta\in\mathbb{R}} \eta \\ & \text{s.t. } \eta \leq g(\mathbf{z}(\boldsymbol{\pi},\mathbf{u}),\boldsymbol{\pi},\mathbf{u}),\ \forall \mathbf{u} \in U. \end{aligned} \tag{9}$$

Here, we introduce control variables $\mathbf{u}$ that can be manipulated in response to uncertainty realizations $\boldsymbol{\pi}$. With respect to the general SIP formulation (6), we have $\mathbf{x} = \boldsymbol{\pi}, \mathbf{p} = \mathbf{u}, X = \Pi$, and $P = U$. This formulation addresses the question of operational feasibility and verifies the (non)existence of feasible control actions to mitigate the effects of worst-case uncertainty. If $\eta^* \leq 0$, then a feasible recourse control action exists that mitigates the worst-case impacts of uncertainty on the process with respect to the performance/safety specifications.

A state-of-the-art method for solving SIPs to global optimality is discussed in the following section.

Global Solution of SIPs

In this section, the SIPres algorithm introduced by Mitsos[12] is presented with respect to the formulation (6) for hybrid model systems for completeness. The algorithm flowchart is illustrated in Figure 3 and relies on solving three nonconvex subproblems (formulated below) to global optimality at each iteration. The algorithm is guaranteed to converge to ϵ-optimality in finitely many iterations under the assumptions of continuity of ϕ and g and the existence of an SIP Slater point arbitrarily close to a minimizer.

Definition 1 (Lower-Bounding Problem[13])**.** Given a finite number of constraints with respect to $\mathbf{p} \in P^{LBD}$ with $P^{LBD} \subset P$ a finite set, the lower-bounding problem is formulated as:

$$\begin{aligned} \phi^{LBD} = \quad & \min_{\mathbf{x}\in X} \phi(\mathbf{x}) \\ & \text{s.t. } g(\mathbf{z}(\mathbf{x},\mathbf{p}),\mathbf{x},\mathbf{p}) \leq 0,\ \forall \mathbf{p} \in P^{LBD}. \end{aligned}$$

Definition 2 (Inner Program[13])**.** Given a point $\bar{\mathbf{x}} \in X$, the inner program is formulated as:

$$\bar{g}(\bar{\mathbf{x}}) = \max_{\mathbf{p}\in P} g(\mathbf{z}(\bar{\mathbf{x}},\mathbf{p}),\bar{\mathbf{x}},\mathbf{p}).$$

The inner program verifies feasibility of the point $\bar{\mathbf{x}}$ with respect to the original SIP. If $\bar{g}(\bar{\mathbf{x}}) \leq 0$, $\bar{\mathbf{x}}$ is feasible in (6).

Definition 3 (Upper-Bounding Problem[13])**.** Given a finite number of constraints with respect to $\mathbf{p} \in P^{UBD}$ with $P^{UBD} \subset P$ a finite set, the upper-bounding problem is formulated as:

$$\begin{aligned} \phi^{UBD} = \quad & \min_{\mathbf{x}\in X} \phi(\mathbf{x}) \\ & \text{s.t. } g(\mathbf{z}(\mathbf{x},\mathbf{p}),\mathbf{x},\mathbf{p}) \leq -\epsilon^g,\ \forall \mathbf{p} \in P^{UBD}, \end{aligned}$$

where $\epsilon^g > 0$ is the *restriction parameter*,[12] representing a parameter for perturbing the right-hand side of the semi-infinite constraint away from zero, thereby restricting the feasible set of

the upper-bounding problem. Note that according to Mitsos [12], the upper-bounding problem should be solved to global optimality to obtain a global solution of the original SIP (6), but a valid upper bound $\phi^{UBD} \geq \phi^*$ can be obtained by a local solution $\bar{\mathbf{x}}$ of the upper-bounding problem if its feasibility in the original SIP (6) is verified. That is, any SIP feasible point is a valid upper bound.

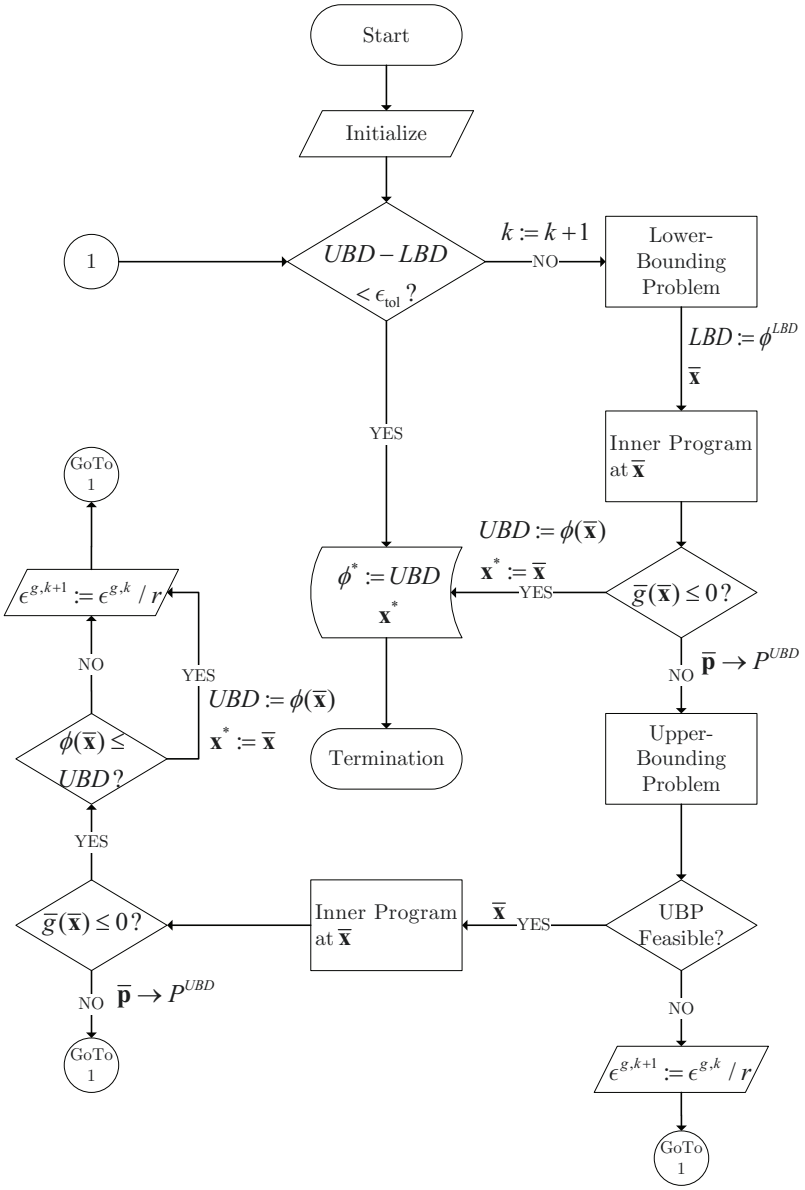


Figure 3: The SIPres algorithm is illustrated as a flowchart as adapted from that presented by Stuber and Barton [13].

Case Studies

All numerical experiments in this work were run on a single thread of an Intel Xeon E3-1270 v5 3.60/4.00GHz (base/turbo) processor with 16GB ECC RAM allocated to a virtual machine running the Ubuntu 18.04LTS operating system and Julia v1.6.1.[66] Absolute and relative convergence tolerances for the B&B algorithm of 10^{-4} were specified for all example problems, unless otherwise noted and a maximum CPU time limit of the SIPres algorithm was set to 3600 seconds. The EAGO.jl package (v0.6.1)[63] was used to solve each optimization problem. Validated interval arithmetic was computed using the package IntervalArithmetic.jl.[67] The Intel MKL package (2019 Update 2)[68] was used to perform all LAPACK[69,70] and BLAS[71] routines. The data used with and generated from the following numerical examples are openly available in the Git repository established at `https://github.com/PSORLab/RobustHybridModels` along with the corresponding problem formulations.

Case Study 1: Robust Feasibility of a Nitrification CSTR

In this case study, we consider the rigorous verification of robust feasibility of a continuous stirred-tank reactor (CSTR) undergoing nitrification reactions for wastewater treatment. The aim here is to verify the existence of a simple robust control policy that maintains the desired water quality specifications. The system involves a single continuously-flowing feed stream and a single continuously-flowing outlet stream, as shown in Figure 4. An air diffuser exists at the bottom of the tank to provide oxygen for oxidizing ammonium. The controller receives feedback signals from the conductivity sensor in the reactor and sends a control signal to the valve on the air stream to increase or decrease the flow of air (i.e., aeration) into the CSTR that, in turn, controls the nitrification reactions. In practice, this aerobic nitrification step often precedes an anaerobic nitrification step.

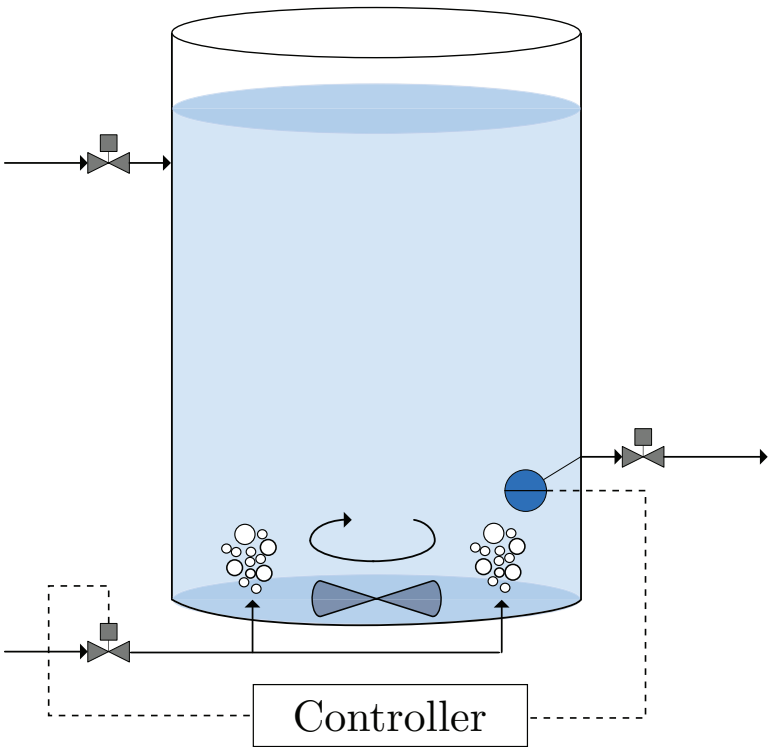


Figure 4: The dynamic nitrification CSTR system considered in the robust feasibility case study (Case Study 1), is shown. Under normal operation, oxygen is injected to control the nitrification reactions using a feedback controller utilizing measurements of the ammonium and dissolved oxygen concentrations in the outlet stream.

Hybrid Model Formulation

The reaction mechanism for this nitrification process has two steps:

$$2NH_4^+ + 3O_2 \rightarrow 2NO_2^- + 4H^+ + 2H_2O$$

$$2NO_2^- + O_2 \rightarrow 2NO_3^-$$

In the first step, the ammonium ions are oxidized to nitrite ions. In the second step, the nitrite ions are further oxidized to nitrate ions. Based on the molecular biological study, the ammonia oxidizing bacteria (AOB) predominates the first step and the nitrite oxidizing bacteria (NOB) carries out the second step.[72,73] The dynamic species mass balances in the CSTR are given by

the ODEs:[74]

$$\begin{aligned}
\frac{dC_{NH}}{dt} &= \frac{1}{V}(\dot{m}_{in}C_{in} - \dot{m}_{out}C_{NH}) - r_{AO}\cdot X_{AO} \\
\frac{dC_{NI}}{dt} &= r_{AO}\cdot X_{AO} - r_{NO}\cdot X_{NO} \\
\frac{dC_{NA}}{dt} &= r_{NO}\cdot X_{NO} \\
\frac{dC_O}{dt} &= -r_{AO}\cdot \Psi_{AO}\cdot X_{AO} - r_{NO}\cdot \Psi_{NO}\cdot X_{NO} + k_{la}\cdot (C_O^* - C_O),
\end{aligned} \tag{10}$$

where C_{NH}, C_{NI}, and C_{NA} are the concentrations (mg N/(L$\cdot$s)) for NH_4^+, NO_2^-, and NO_3^-, respectively, C_O is the oxygen concentration (mg O_2/(L$\cdot$s)), $\dot{m}_{in}$ and $\dot{m}_{out}$ are continuous inlet and outlet flow rates (L/s), C_{in} is the NH_4^+ concentration in the inlet stream, and V is the reactor volume (1000 L). The ammonium oxidation rate (mg N-NH_4^+/ (g $VSS_{AO}\cdot$min)) is given by r_{AO}, X_{AO} is the concentration of AOB (mg VSS/L), r_{NO} is the nitrite oxidation rate (mg N-NO_2^-/ (g $VSS_{NO}\cdot$min)), X_{NO} is the concentration of NOB (mg VSS/L), Ψ_{AO} is the stoichiometric ratio between oxygen and ammonia (mg O_2/ mg N-NH_4^+), Ψ_{NO} is the stoichiometric ratio between oxygen and nitrite (mg O_2/ mg N-NO_2^-), k_{la} is the volumetric mass transfer coefficient (s^{-1}), and C_O^* is the dissolved oxygen saturation concentration (9.1 mg/L at 20 °C,[75]). The rate equation for nitrite oxidation r_{NO} can be expressed further as:

$$r_{NO} = r_{NO,\max}\frac{C_{NI}}{K_{SNO} + C_{NI} + \frac{C_{NI}^2}{K_{INO}}}\cdot\frac{C_O}{K_{ONO} + C_O},$$

where $r_{NO,\max}$ is the maximum nitrite consumption rate (mg N-NO_2^-/ (g $VSS_{NO}\cdot$min)), K_{SNO} is the Monod constant of nitrite for NOB (mg N-NO_2^-/L), K_{INO} is the inhibition constant of nitrite for NOB (mg N-NO_2^-/L), and K_{ONO} is the Monod constant of oxygen for NOB (mg/L).

The aeration process is governed by the mass transfer of oxygen into the solution as the term $k_{la}(C_O^* - C_O)$ in (10), that is derived from the standard oxygen transfer rate (SOTR, mg/s) defined as: SOTR = $k_{la}C_O^*V$.[76] Assuming that the air flow rate from the air diffuser is represented by Q (mg/s), the mass flow rate of oxygen W_O in the air stream can be computed from an

empirical formula: $W_O = 0.2967Q$.[77] Then, the standard oxygen transfer rate can be calculated as SOTR = SOTE·W_O, where SOTE is the standard oxygen transfer efficiency (%). Therefore, the aeration mass transfer coefficient can be rewritten as:

$$k_{la} = \frac{0.2967Q \cdot \text{SOTE}}{C_O^* \cdot V}.$$

The parameter values used in the model are listed in Table 1.

Table 1: The model parameters used for the nitrification CSTR case study are listed in this table.

Symbol	Definition	Value	Reference
V	Liquid volume (L)	1000	This study
$\dot{m}_{in}$	Inlet volumetric flow rate (L/s)	4.167	This study
$\dot{m}_{out}$	Outlet volumetric flow rate (L/s)	4.167	This study
C_O^*	Saturated oxygen concentration (mg O_2/L	9.1	[75]
X_{AO}	Concentration of AOB (mg VSS/L)	505	[74]
X_{NO}	Concentration of NOB (mg VSS/L)	151	[74]
$r_{NO,\max}$	Max. nitrite consumption rate (mg N-NO_2^-/ (g VSS_{NO} · min))	1.07	[74]
Ψ_{AO}	Stoich. ratio of oxygen to ammonia (mg O_2/ mg N-NH_4^+)	2.5	[78]
Ψ_{NO}	Stoich. ratio of oxygen to nitrite (mg O_2/ mg N-NO_2^-)	0.32	[78]
K_{SNO}	Monod constant of nitrite for NOB (mg N-NO_2^-/L)	1.6	[78]
K_{INO}	Inhibition constant of nitrite for NOB (mg N-NO_2^-/L)	13000	[78]
K_{ONO}	Monod constant of oxygen for NOB (mg/L)	1.5	[78]
SOTE	Standard oxygen transfer efficiency (%)	10	[76]

Since this is a complicated biological reaction system in a physicochemical environment, it is very hard to obtain accurate kinetic parameters under constantly varying conditions for FPMs. There are situations such that the biological parameters cannot be easily obtained and verified by experiments. Thus, we propose to use an ANN model to estimate the rate constant r_{AO} and account for the hybrid modeling approach in this study. The rate constant r_{AO} is re-

lated to both ammonium concentration c_{NH} and oxygen concentration c_{O}. Consequently, r_{AO} is calculated as an intermediate variable from the ANN and substituted into the dynamic hybrid model, as illustrated in Figure 5 to form a dynamic serial hybrid model.

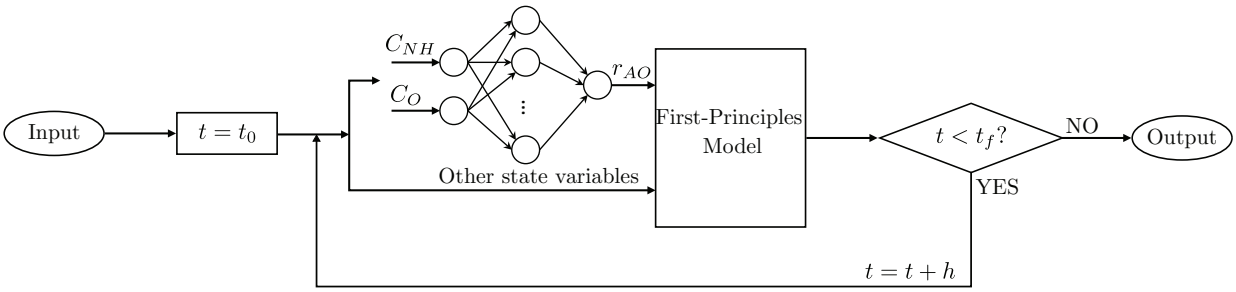


Figure 5: The hybrid model architecture used for the nitrification CSTR case study (Case Study 1), is illustrated. This model represents a DDM/FPM serial architecture with a reaction rate term modeled by an ANN.

Data-Driven Model Construction

A training data set was generated by evaluating a proposed empirical model for r_{AO} provided by Sánchez et al.[74], that relates r_{A0} to C_{NH} and C_O. The ANN model is developed to demonstrate a hybrid modeling approach in more complicated reacting systems. A preliminary investigation established physically plausible ranges for C_{NH} and C_O of $[0,40]$ and $[0,9.1]$, respectively, while values within $[0,4]$ and $[0,1]$, respectively, were typically observed from exploratory simulations of the system. For $(C_{NH}, C_O) \in [0,4] \times [0,1]$, r_{AO} varies significantly with respect to C_{NH} and C_O, while it is relatively flat outside this region. The empirical model was evaluated over two distinct Latin hypercube sampling (LHC) designs with 10^5 points each: one design on the domain $[0,40] \times [0,9.1]$, and the other on the domain $[0,4] \times [0,1]$, to ensure adequate sampling of the sensitive region. As with the previous examples, the data set was scaled using a min-max normalization and divided randomly into training (70%), validation (15%), and test (15%) sets. Training was performed using the Keras[79] module in the nightly version of Tensorflow[80] with the Adam optimizer. An early stopping protocol was performed using default parameters to prevent overfitting.[81] An ANN consisting of two hidden layers, each containing

eight neurons, with the hyperbolic tangent activation function and a sigmoid output layer was used. This network was trained using a learning rate schedule that began with a value of 0.1 and was decreased by a factor of 0.5 every 100 epochs. This training protocol lead to loss values of 1.837×10^{-6} and 1.9708×10^{-6} for the training and validation sets, respectively.

SIP Formulation and Results

According to the standard of moderate municipal wastewater,[82] the concentration of ammonium ions in the effluent is required to be below 30 mg N-NH_4^+/L. Moreover, the dissolved oxygen concentration must be lower than 2 mg O_2/L to ensure that a viable operating window exists for a secondary anaerobic denitrification step.[83] Thus, the upper specification limits for ammonium ions (USL_{NH}) and dissolved oxygen (USL_O) are set to 30 mg N-NH_4^+/L and 2 mg O_2/L, respectively. The CSTR is initially operating at steady state with a constant concentration (31 mg N-NH_4^+/L) in the inlet stream. At some moment ($t = t_0$), a concentration shock is observed in the inlet stream within a short operating window (20 s) of the treatment process. It is our desire to operate the process in an open-loop manner, and so the objective here is to determine whether there exists a feasible design for the valve setting for air flow rate that is robust to worst-case realizations of uncertainty at the end of the simulation horizon ($t = t_f = 100$ s). The design variable is taken to be the air flow rate $d = Q \in D = [440, 2000]$, which can be interpreted as the valve setting. The uncertainty comes from the disturbance in the inlet stream $\pi = C_{in} \in \Pi = [31.0, 40.0]$. Thus, the design under uncertainty feasibility problem accounts for the following semi-infinite constraints:

$$\begin{aligned} C_{NH}(t_f, d, \pi) - \mathrm{USL_{NH}} \leq 0, \ \ \forall \pi \in \Pi, \\ C_O(t_f, d, \pi) - \mathrm{USL_O} \leq 0, \ \ \forall \pi \in \Pi. \end{aligned} \tag{11}$$

The two semi-infinite constraints present in (11) are then reformulated as a single nonsmooth semi-infinite constraint, and an epigraph rearrangement of the problem is made to yield the

following SIP:

$$\eta^* = \min_{d\in D,\eta\in H} \eta \tag{12}$$
$$\text{s.t. } \max\{C_{NH}(t_f,d,\pi) - \text{USL}_{\text{NH}}, C_O(t_f,d,\pi) - \text{USL}_{\text{O}}\} \le \eta, \ \forall \pi \in \Pi.$$

This formulation corresponds with the design under uncertainty feasibility problem (7). Again, the term η represents a measure of robust feasibility. If the optimal solution value of the feasibility problem (12) satisfies $\eta^* \le 0$, then a design exists that is robust to worst-case realization of uncertainty. For this problem, a relative convergence tolerance of 10^{-3} for the SIP-feasibility problem (12) was used. An explicit Euler method was used to integrate (10) with a stepsize of $h = 10$ s. To avoid domain violations and associated difficulties that arise from overestimation of C_{O_2}, a positive value of C_{O_2} was enforced by setting $C_{O_2} = \max(C_{O_2}, \epsilon)$ with $\epsilon = 10^{-10}$ at each time step. The SIPres algorithm[12] (see Figure 3) was used to solve the SIP given in (12). The SIPres algorithm solves (12) after a single iteration in 21.86 CPU seconds with an optimal solution $\eta^* = 0.288$, illustrating that a robust design does not exist for this system with respect to the given performance/safety specifications.

This motivates a search for alternative approaches to verify robustness. Namely, we seek to determine if a robust operation is feasible. We consider the same uncertainty source $\pi \in \Pi$ and the control variable is taken as $u = Q \in U = [440, 2000]$. We aim to establish a robust operation problem to verify if a control recourse exists that mitigates the impacts of uncertainty. The semi-infinite constraints in this problem are:

$$C_{NH}(t_f,\pi,u) - \text{USL}_{\text{NH}} \le 0, \ \forall u \in U, \tag{13}$$
$$C_O(t_f,\pi,u) - \text{USL}_{\text{O}} \le 0, \ \forall u \in U.$$

Accordingly, the operation under uncertainty feasibility problem can be expressed as the fol-

lowing SIP:

$$\eta^* = \max_{\pi \in \Pi} \eta \tag{14}$$
$$\text{s.t. } \eta \leq \max\{C_{NH}(t_f, \pi, u) - \text{USL}_{\text{NH}}, C_O(t_f, \pi, u) - \text{USL}_{\text{O}}\}, \ \forall u \in U.$$

Again, the SIPres algorithm[12] with convex/concave envelopes of activation functions[84] was used to solve the SIP (14). The SIPres algorithm terminates with $\eta^* = 0.288$ after a single iteration in 21.14 CPU seconds. As a consequence, we see that a control setting recourse is not feasible given the provided specifications. Moreover, the presented formulations with dynamic hybrid models demonstrate the applicability of robustness verification approaches to relatively complicated processes with dynamic governing equations.

Case Study 2: Worst-Case Design of Subsea Production Facilities - Mitigation of Domain Violations

In Stuber et al.[85], the worst-case design of a subsea oil production facility (illustrated in Figure 6) was formulated as an operation under uncertainty feasibility problem and solved using several novel methodologies. Namely, the problem was reformulated as an SIP with implicit functions embedded. The subsea separator model uses transcendental functions with definitions on narrow domains that result in numerical difficulties when simulating and optimizing the system. For the purposes of this paper, the interest is not in the application itself, but in the model as representative of a broader class of industrially-relevant examples plagued by numerical simulation and convergence issues caused by domain violations. Within this context, it is of interest to explore how hybrid modeling approaches might be used to improve the robustness of an FPM and solvers (i.e., improve the reliable convergence to accurate solutions).

Domain violations are ubiquitous across process systems engineering applications and pose major challenges to researchers and practitioners of simulation and optimization.[86–88]

Within the broader context of numerical simulation, domain violations are encountered when a solver attempts to evaluate an expression at a point outside of its defined domain (e.g., divide by zero or square-root a negative number). Hybrid models may pose additional challenges as they may also suffer from violations of their domains of validity. That is, a solver may attempt to evaluate a DDM at a point outside of the domain of inputs for which the DDM is considered to be “valid" (i.e., accurately represents reality). When considering the optimization of hybrid models, domain violations may be frequently encountered when such domains may not be explicitly known and accounted for with appropriate constraints, without prior analysis.

In Stuber et al.[85], a method of forward-backward interval constraint propagation on the DAG,[48,89] interval contractor methods,[90] a novel convex/concave relaxation algorithm,[61] and a novel algorithm for solving SIPs[13] were all necessary to solve this problem. While these methods adequately address the problem in question, the broad and robust applicability of this approach to more general SIPs is wanting. We should note, however,that this approach reduces the problem in question from a GSIP to that of a standard SIP. This, combined with a desire to generalize the prior results to allow for the incorporation of more complex physical phenomena, further motivates our interest in this example.

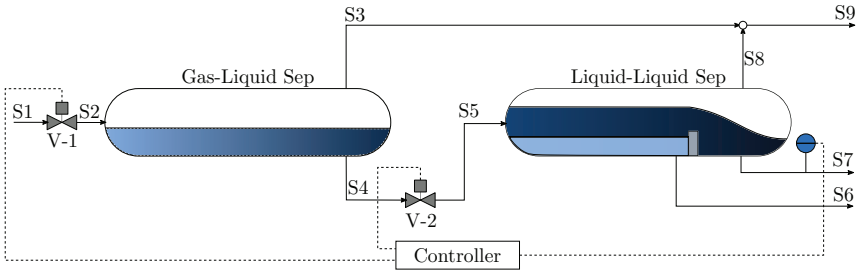


Figure 6: The process flow diagram for the subsea separator (adapted from Stuber et al.[85]), is presented in this figure. This system is considered in the subsea separator case study for the use of hybrid models to overcome numerical domain violation issues. A mixture of gas, oil, and water is fed to the system in S1. Gas is separated from the oil-water mixture in the gas-liquid separator and oil is separated from water in the liquid-liquid separator.

Hybrid Model Formulation

In this study, the focus is on a modification of the gas-liquid/liquid-liquid separation train problem presented in Case 3 of Stuber et al.[85]. To model the performance of gas separation in each separator, simple exponential decay models based on mean gas bubble sizes were assumed.[85] The relationship between inlet and outlet gas quantities may be expected to change in meaningful ways when a population-based model of bubble sizes is incorporated along with information about the equipment's geometry. Moreover, for bubbly mixtures, overflow can occur in volumes less than those considered by solely taking into account liquid levels, provided a large gas concentration is present in the inlet. In practice, this type of problem is typically characterized using a mixture of computational fluid dynamics software and empirical investigation.

We propose simplifying the published model by using an ANN surrogate model to relate the input variables to the gas-liquid separator and the control variable for the second valve (V-2) to the system outputs. This serves to eliminate the domain violation issue inherent in the model, as the activation functions considered lack domain restrictions, and allow the system-level model to be readily generalized to incorporate information from computational experiments generated by CFD models, or elsewhere. The inputs, outputs, and expected ranges of each variable in each ANN are summarized in Table 2. As the development of CFD models is often time consuming, equipment specific, and not the central focus of this work, we will forgo this and instead illustrate how this approach works at the system level. We use the prior mass balances and process specifications for the gas-liquid separator (GLS) and the liquid-liquid separator (LLS). The governing equations for the first valve (V-1), and the gas mixer will be left

unaltered. The equations governing V-1 lead to the following simplified relationships:

$$\begin{aligned} \xi_{W,1} &= 1 - \xi_{G,1} - \xi_{O,1}, \\ SG_{mix}^{-1} &= \frac{\xi_{G,1}}{SG_G} + \frac{\xi_{W,1}}{SG_W} + \frac{\xi_{O,1}}{SG_O}, \\ \dot{m}_2 &= u_1 C_{v1} \sqrt{SG_{mix}^{-1}(P_{well} - P_{GLS}) + \epsilon_d} \\ \boldsymbol{\xi}_2 &= \boldsymbol{\xi}_1. \end{aligned} \tag{15}$$

These equations specify that the mass fractions in the input stream ($\xi_{W,1}$, $\xi_{G,1}$, $\xi_{O,1}$) sum to one, provide a formula relating specific gravity of the mixture SG_{mix} to the specific gravity of individual components (SG_G, SG_W, SG_O), and relate the mass flow rate through the valve $\dot{m}_2$ to valve position (u_1), valve coefficient (C_{v1}) and a specified pressure difference ($P_{well} - P_{GLS}$) between the GLS and the wellhead. A small number $\epsilon_d = 10^{-6}$ is added to the argument of the $\sqrt{\cdot}$ function to avoid the introduction of numerically ill-posed gradients that present computational issues for local NLP subproblems encountered during global optimization.

Simple algebraic substitutions of the equations governing V-2 and the LLS behavior lead to the following algebraic expression:

$$\xi_{G,7} = \xi_{G,4} \exp\left(-\dot{m}_4 k_{LLS} \frac{V_{LLS}}{\rho_4 + \epsilon_d}\right). \tag{16}$$

While additional expressions are required to fully determine all stream characteristics in the flowsheet, the LLS performance specification (16) is sufficient to construct the SIP constraint. This specification relates the inlet gas mass fraction $\xi_{G,4}$, density ρ_4, and mass flow rate $\dot{m}_4$ to the oil product stream gas mass fraction $\xi_{G,7}$ by means of a performance constant k_{LLS}. Due to downstream equipment specifications, the oil product stream gas mass fraction may not exceed the value $G_{max} = 0.05$. The full model can be found in Stuber et al.[85] with the analysis of the DAG in Stuber[48], Sec. 8.1.

Table 2: The state variables for the subsea separator case study are listed in this table along with their corresponding bounds, units, and identification of whether they are classified as inputs or outputs for the hybrid model. Bounds directly specified by Stuber et al. [85] were used if available. Otherwise, natural interval extensions of known quantities were used to compute necessary values. The parameters C_{v1}, SG_G, SG_W, SG_O, g_a, P_{well} P_{LLS}, P_{GLS}, k_{GLS}, L_{GLS}, and R_{GLS} take the values previously specified in Stuber et al. [85].

Variable	Lower	Upper	Unit	Layer
$\dot{m}_2$	8.228	19.517	kg/s	Input
u_2	0.35	0.8	-	Input
$\xi_{G,2}$	0.35	0.5	-	Input
$\xi_{W,2}$	0.1	0.25	-	Input
$\dot{m}_4$	541.364	845.881	kg/s	Output
H_{GLS}	0.462165	0.7992	m	Output
$\xi_{G,4}$	9.463053×10^{-3}	0.36	-	Output
P_4	4.00264×10^6	4.01079×10^6	Pa	Output
ρ_4	584.6	1376.6	kg/m^3	Output

Data-Driven Model Construction

Training data was generated by repeatedly solving a feasibility problem equivalent to the nonlinear system:

$$
\begin{aligned}
(\xi_{G,2}-1)\dot{m}_2-(\xi_{G,4}-1)\dot{m}_4 &= 0 \qquad (17)\\
u_2^2 C_{v2}^2 \rho_W^o (P_4-P_{LLS})-\rho_4 \dot{m}_4^2 &= 0\\
(P_4-P_{GLS})-\rho_4 g_a H_{GLS} &= 0\\
\xi_{G,2}\exp\left(-k_{GLS}\rho_4\frac{(\xi_{G,4}-1)V_{GLS}(H_{GLS})}{(\xi_{G,2}-1)\dot{m}_2}\right)-\xi_{G,4} &= 0\\
\rho_4\frac{\xi_{G,4}}{SG_G}+\rho_4\frac{\xi_{G,2}(\xi_{G,4}-1)}{SG_W(\xi_{G,2}-1)}+\rho_4\frac{\xi_{G,2}(\xi_{G,4}-1)(1+\xi_{W,2}-\xi_{G,2})}{SG_O(\xi_{G,2}-1)}-\rho_W^o &= 0\\
V_{GLS}-L_{GLS}\left((H_{GLS}-R_{GLS})\sqrt{(2R_{GLS}H_{GLS}-H_{GLS}^2)}+R_{GLS}^2\cos^{-1}\left[1-\frac{H_{GLS}}{R_{GLS}}\right]\right) &= 0
\end{aligned}
$$

that is parameterized by $\mathbf{w}=(\dot{m}_2,u_2,\xi_{G,2},\xi_{W,2})\in W$. Ipopt[91] was used to solve (17) with a multistart approach using 16 initial guesses chosen via an LHC sampling procedure for each set of parameters considered. An LHC sampling procedure was then performed over a range of valid values given in Table 2 to generate 10^5 data points used to train the DDM. As noted in Stuber et al.[85], the implicit function characterized by (17) may not exist for some realization of uncertainty and control variables. Values that yielded a locally-infeasible result were labelled accordingly, while the solutions of the feasible problems were saved. Of the 10^5 points generated, 6,742 infeasible points were evaluated.

The approach to training the ANNs for this problem, parallel the previous examples. The data set was scaled using a min-max normalization and divided randomly into training (70%), validation (15%), and test (15%) sets. Training was performed using the Keras[79] module in the nightly version of Tensorflow[80] with the Adam optimizer. The surrogate ANN consisted of four inputs, two dense layers, twelve neurons per layer, and utilized the SiLU activation function. A sigmoid output layer was used to ensure that the output results remained within the range of the training data. The surrogate model had min-max-scaled mean-squared-error (MSE) values

of 7.74×10^{-5} and 2.2506×10^{-4} on the training and test sets, respectively. The validity constraint consists of an ANN with four inputs, two hidden layers, two neurons per layer, and utilizes the SiLU activation functions with a single hyperbolic tangent output layer that is trained using a binary cross-entropy loss function. This achieved a binary accuracy greater than 99.0% on both the test and training sets. Weights and offsets for both the surrogate model and the validity constraint can be found in the Git repository. Both the surrogate and classifier ANNs used a learning rate schedule that began with a value of 0.1 and was decreased by a factor of 0.5 every 100 epochs. We note here that, due to the nature of the application, no classifier can be expected to be exactly accurate as the valid and invalid regions adjoin one another.

SIP Formulation and Results

Any ANN can only be expected to provide valid results when interpolating and special consideration must be given to exclude invalid operating parameters. In general, two distinct outcomes must be considered: either a domain violation arises from a purely numerical consideration (e.g., instability) or one that corresponds to a nonphysical operating condition (e.g., negative density). In the former case, the accuracy of the hybrid model should be verified to guarantee the results for the corresponding robust operation problem. In the latter case, restricting the model to a domain of validity is sufficient to ensure a guarantee of robustness.

Ensuring validity regions for surrogate models remains an active area of research within the optimization community. Some approaches include restricting the function evaluations to be within the convex hull of a finite number of sampled points[22,92] or categorizing the data using a support vector machine.[29,93] In either case, this restriction can be framed as a potentially nonconvex constraint $g_c : Z \times \Pi \times U \to \{-1, 1\}$ where $g_c(\hat{\mathbf{z}}, \boldsymbol{\pi}, \mathbf{u}) = -1$ indicates a valid model for $(\hat{\mathbf{z}}, \boldsymbol{\pi}, \mathbf{u}) \in Z \times \Pi \times U$. We note that the forms addressed pertain to standard optimization formulations and the extension of these approaches to multilevel programs has yet to be developed. In keeping with surrogate modeling frameworks adopted in this paper, we choose to make use of a second ANN, $f_c^{ANN} : Z \times \Pi \times U \to \mathbb{R}$, in addition to the surrogate model, to perform a binary

classification task via logistic regression.

The binary classification task is performed as follows. Provided that $f_c^{ANN}(\hat{\mathbf{z}}, \boldsymbol{\pi}, \mathbf{u}) \leq 0$, the input features is classified as $g_c(\hat{\mathbf{z}}, \boldsymbol{\pi}, \mathbf{u}) = -1$ (valid classification). In a corresponding manner, the classification ANN predicts that the the input features will be classified as $g_c(\hat{\mathbf{z}}, \boldsymbol{\pi}, \mathbf{u}) = +1$ (invalid classification) due to a domain violation $f_c^{ANN}(\hat{\mathbf{z}}, \boldsymbol{\pi}, \mathbf{u}) > 0$. With this validity constraint, the robust feasibility constraint takes the logical form:

$$\forall \boldsymbol{\pi} \in \Pi, \exists \mathbf{u} \in U : g(\hat{\mathbf{z}}, \boldsymbol{\pi}, \mathbf{u}) \leq 0 \wedge g_c(\hat{\mathbf{z}}, \boldsymbol{\pi}, \mathbf{u}) \leq 0 \wedge \mathbf{h}(\hat{\mathbf{z}}, \boldsymbol{\pi}, \mathbf{u}) = \mathbf{0}. \tag{18}$$

For this problem, the state variables $\hat{\mathbf{z}}$ can be calculated as an explicit function $\mathbf{z} : \Pi \times U \rightarrow Z$ such that $\mathbf{h}(\mathbf{z}(\boldsymbol{\pi}, \mathbf{u}), \boldsymbol{\pi}, \mathbf{u}) = \mathbf{0}$ for every $(\boldsymbol{\pi}, \mathbf{u}) \in \Pi \times U$. The robust operation problem can then be formulated as an SIP with a nonsmooth semi-infinite constraint:

$$\begin{aligned} \eta^* = & \max_{\boldsymbol{\pi} \in \Pi, \eta \in H} \eta \\ & \text{s.t.} \quad \eta \leq \max\left\{g(\mathbf{z}(\boldsymbol{\pi}, \mathbf{u}), \boldsymbol{\pi}, \mathbf{u}), g_c(\mathbf{z}(\boldsymbol{\pi}, \mathbf{u}), \boldsymbol{\pi}, \mathbf{u})\right\}, \forall \mathbf{u} \in U. \end{aligned} \tag{19}$$

Alternatively, (19) may be reformulated as an SIP with a disjunctive constraint or as a mixed-integer SIP. Note that this form is identical to the structure encountered when relaxing a GSIP and the reader is directed to Mitsos and Tsoukalas [16] for a discussion of the numerical eccentricities associated with solving that problem class. The robust design problem for the subsea separator may then be formally stated as:

$$\begin{aligned} \eta^* = & \max_{\boldsymbol{\pi} \in \Pi, \eta \in H} \eta \\ & \text{s.t.} \ \eta \leq \max\left\{\xi_{G,7}(\boldsymbol{\pi}, \mathbf{u}) - G^{max}, g_c(\mathbf{z}(\boldsymbol{\pi}, \mathbf{u}), \boldsymbol{\pi}, \mathbf{u})\right\}, \ \forall \mathbf{u} \in U \\ & \quad U = [0.35, 0.8]^2 \\ & \quad \Pi = [0.35, 0.5]. \end{aligned} \tag{20}$$

We note that the valid region of the developed binary classifier is bounded by a 0-sublevel set, which is potentially a disconnected and nonconvex set, and therefore the following equivalence can be established:

$$\{(\boldsymbol{\pi},\mathbf{u}) \in \Pi \times U \,:\, g_c(\mathbf{z}(\boldsymbol{\pi},\mathbf{u}),\boldsymbol{\pi},\mathbf{u}) = -1\} \Leftrightarrow \{(\boldsymbol{\pi},\mathbf{u}) \in \Pi \times U \,:\, g_t(\mathbf{z}(\boldsymbol{\pi},\mathbf{u}),\boldsymbol{\pi},\mathbf{u}) \leq 0\},$$

with $g_t(\cdot,\cdot,\cdot) \equiv f_c^{ANN}(\cdot,\cdot,\cdot)$. By construction, g_t is continuous on its domain, and so this reformulation ensures that the semi-infinite constraint is continuous, and in turn, ensures that the convex/concave relaxations used in the subproblem of the SIPres algorithm[12] exhibit desirable convergence properties.[94] Under this equivalence, the SIP (20) is reformulated as:

$$\begin{aligned} \eta^* = \max_{\boldsymbol{\pi}\in\Pi,\eta\in H}\ & \eta \\ \text{s.t. } & \eta - \max\left\{\xi_{G,7}(\boldsymbol{\pi},\mathbf{u}) - G^{max}, g_t(\mathbf{z}(\boldsymbol{\pi},\mathbf{u}),\boldsymbol{\pi},\mathbf{u})\right\} \leq 0, \ \ \forall \mathbf{u} \in U \\ & U = [0.35, 0.8]^2 \\ & \Pi = [0.35, 0.5]. \end{aligned} \tag{21}$$

We first solved this hybrid model using the SIPres[12] routine provided in EAGO v0.6.1[63,95] and using the convex/concave envelope of SiLU described in a forthcoming work.[84] The SIP was solved to an absolute tolerance of 10^{-3}. The algorithm terminated in 3 iterations, taking 2.9 CPU seconds when using the envelope of SiLU when computing relaxations. The SIPres algorithm terminated after an optimal value was found in the lower-bounding problem and the maximal value of the corresponding lower-level problem was found to be nonpositive with a value of $\eta^* = -6.6 \times 10^{-4}$. In contrast, the original method in Stuber et al.[85] provided a solution value of -5.77×10^{-3} for this case study. However, it is worth noting that the method proposed by Stuber et al.[85] has an early-termination criterion whereby the algorithm terminates with a feasible suboptimal solution as soon as robustness is verified. Thus, the solution value obtained by Stuber et al.[85] is an upper bound on the global solution. Despite this, we notice that

$\eta^* > -5.77 \times 10^{-3}$, seemingly in violation of the upper bound for the full mechanistic model.[85] Since the hybrid model utilizes an ANN to approximate the original equations exhibiting numerical issues (i.e., domain violations), such discrepancies are anticipated. The level of confidence in the solution lies in the accuracy of the trained model versus the constraint satisfaction and algorithm convergence tolerances. In practice, it may be possible verify SIP feasibility of an optimal solution with respect to the full mechanistic model. However, this depends entirely on the existence and complexity of such a model. For this case, the results verify that both models ensure the robust feasibility of this operation. A performance normalization was used based on CPU single-core IPC using the Cinebench R15 (Maxon, Newbury Park, CA) single-core benchmark to enable a fair comparison of the performance of the approach in this work versus Stuber et al.[85]. The normalized results indicate a 70-fold performance improvement over the original solution time of 549.3 CPU seconds reported by Stuber et al.[85]. In this particular case, we expect this improvement to be genuine as prior comparisons of Julia/EAGO to C++/MC++ implementations differed only by at most a factor of three.[96] However, the degree of computational performance improvement for the surrogate modeling approach relative to the original work of Stuber et al.[85] will undoubtedly be model-specific. As such, we make no broad claim of superior performance for this method. However, this example does illustrate that the use of surrogate modeling represents a viable approach to eliminate the need to apply specialized parametric interval analysis,[48,90] constraint propagation techniques,[85] and implicit relaxation[61] methods when addressing bilevel optimization problems with coupling equality constraints, by replacing these models with a formulation that can be readily addressed with standard global optimization solvers.

Extension to Implicit Forms

In the **Optimization of Hybrid Models** section, the assumption was made that a unique explicit closed-form function $\mathbf{z}: Y \to Z$ exists such that $\mathbf{h}(\mathbf{z}(\mathbf{y}), \mathbf{y}) = \mathbf{0}$ for every $\mathbf{y} \in Y$. This assump-

tion was also made within the context of SIPs in the section **Semi-Infinite Optimization with Hybrid Models**. As mentioned in those sections, the explicit closed-form solution assumptions do not necessarily restrict the applicability of the approach. Stuber et al.[61] originally developed a theory for considering implicit functions within deterministic global optimization formulations. This was explored further within the context of SIPs by Stuber and Barton[13]. Summarily, Stuber and Barton[13] extended the SIP approach for solving (5) to the more general case that the equality constraints do not admit an explicit closed-form solution. In this section, we discuss the conditions under which these assumptions may be relaxed and extend the applicability to a broader class of hybrid models that may involve implicit functions, including implicit ANNs and general nonlinear mechanistic models.

The conditions for considering implicit functions are established as follows. In the previous sections, the only requirements of the equality constraints $\mathbf{h}(\hat{\mathbf{z}},\mathbf{x},\mathbf{p}) = \mathbf{0}$ of (5) representing a hybrid model, were that they are factorable and continuous. Here, we have the additional requirement that $\mathbf{h} : Z \times X \times P \to \mathbb{R}^{n_z}$ is continuously differentiable on its domain. Then, it must be assumed that there exists an implicit function $\mathbf{z} : X \times P \to Z$ such that $\mathbf{h}(\mathbf{z}(\mathbf{x},\mathbf{p}),\mathbf{x},\mathbf{p}) = \mathbf{0}$ for every $(\mathbf{x},\mathbf{p}) \in X \times P$. For the appropriate theories and methods[13,61] to hold, and therefore to be applicable to hybrid models, it must again be assumed that such a function $\mathbf{z}$ is unique in the set Z. In other words, such a Z must exist within which $\mathbf{z}$ is unique on $X \times P$.

Conditions for guaranteeing uniqueness of $\mathbf{z}$ in Z on $X \times P$ may be inferred from the structure of the feed-forward ANN (as an explicit input-output mapping) and under the conditions stipulated by the *semilocal implicit function theorem* (Neumaier[90], Thm. 5.1.3). Furthermore, existence and uniqueness tests associated with parametric interval methods (e.g., interval Newton,[90] Krawcyzk,[97] Hansen-Sengupta[90]) may be used to verify this condition. Note that this does not require that $\mathbf{h}$ has a unique solution, and in the event that multiple solution branches of $\mathbf{h}$ exist in $Z \times X \times P$, bisection-based methods may be sufficient to identify a partition such that the existence and uniqueness of an implicit function can be guaranteed for each element of the partition (see Stuber et al.[61], App. 1 and Stuber[48], Sec. 3.5 for discussions on this). For-

mal treatment of cases in which $Z \times X \times P$ encloses bifurcation points and/or multiple solution branches of $\mathbf{h}$ remains an active area of research.

Conclusion

In this work, we formalized the foundations for SIPs with hybrid first-principles and data-driven models. Particular attention was paid to surrogate modeling via ANNs as the data-driven sub-models. Reduced-space SIP formulations with hybrid models were proposed and demonstrated through three common types of robust design and optimization under uncertainty problems. The SIPres algorithm[12] was used for solving two case studies to demonstrate practicability and superiority of our approaches.

In our first case study, we illustrated how an SIP containing a hybrid model may be used to solve robust feasibility problems pertinent to a continuous nitrification CSTR for wastewater treatment. The use of hybrid models in this application allowed a data-driven approach to describe kinetic rate parameters in biological systems. A reformulation to combine two semi-infinite constraints on ammonium and dissolved oxygen concentrations in the effluent was implemented. The SIP framework presented herein was shown to be sufficiently general such that it may readily address dynamic robust feasibility problems within the context of hybrid models.

The robust simulation of a horizontal gas-liquid and liquid-liquid separator train was revisited in the second case study. This problem is especially challenging as the modeling equations are plagued by numerical issues caused by domain violations. The domain violation problem was addressed with a novel approach that incorporates validity constraints and replaces the problematic models encountering domain violations with an ANN. This problem demonstrates how the application of hybrid models may overcome numerical difficulties often encountered when simulating complicated process systems models. Moreover, the incorporation of validity constraints naturally leads to a nonsmooth SIP formulation that may readily be reformulated as a mixed-integer problem, a disjunctive formulation, or a GSIP.

One interesting application of this work is the solution of problems that require representation by multiple distinct models. These may arise when modeling dynamical systems stemming from transport phenomena whose underlying physics change markedly for different realizations of decision and uncertainty values. In this case, we can generalize the approach detailed here to associate each model with a region of validity and a nonsmooth SIP formulation of the optimization problem. The use of specialized forms of validity constraints should be considered, such as mixed-integer linear formulations, as an alternative to the general nonlinear formulation used herein. This may allow for the use of specialized algorithms that address GSIP formulations for larger and more complex applications.

Acknowledgements

Funding: This material is based upon work supported by the National Science Foundation under Grant No. 1932723. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

(1) Wen, U.-P.; Hsu, S.-T. Linear Bi-level Programming Problems—A Review. *Journal of the Operational Research Society* **1991**, *42*, 125–133, DOI: doi:10.1038/sj/jors/0420204.

(2) Ben-Ayed, O. Bilevel linear programming. *Computers & Operations Research* **1993**, *20*, 485–501, DOI: doi:10.1016/0305-0548(93)90013-9.

(3) Anandalingam, G.; Friesz, T. L. Hierarchical optimization: An introduction. *Annals of Operations Research* **1992**, *34*, 1–11, DOI: doi:10.1007/bf02098169.

(4) Vicente, L. N.; Calamai, P. H. Bilevel and multilevel programming: A bibliography review. *Journal of Global optimization* **1994**, *5*, 291–306, DOI: doi:10.1007/bf01096458.

(5) Dempe, S. Annotated Bibliography on Bilevel Programming and Mathematical Programs with Equilibrium Constraints. *Optimization* **2003**, *52*, 333–359, DOI: doi:10.1080/0233193031000149894.

(6) Gümüş, Z. H.; Floudas, C. A. Global optimization of nonlinear bilevel programming problems. *Journal of Global Optimization* **2001**, *20*, 1–31.

(7) Floudas, C. A.; Gümüş, Z. H. Global Optimization in Design under Uncertainty: Feasibility Test and Flexibility Index Problems. *Industrial & Engineering Chemistry Research* **2001**, *40*, 4267–4282, DOI: doi:10.1021/ie001014g.

(8) Mitsos, A.; Lemonidis, P.; Barton, P. I. Global solution of bilevel programs with a nonconvex inner program. *Journal of Global Optimization* **2008**, *42*, 475–513, DOI: doi:10.1007/s10898-007-9260-z.

(9) Pistikopoulos, E. N. Perspectives in multiparametric programming and explicit model predictive control. *AIChE journal* **2009**, *55*, 1918–1925, DOI: doi:10.1002/aic.11965.

(10) Oberdieck, R.; Diangelakis, N. A.; Nascu, I.; Papathanasiou, M. M.; Sun, M.; Avraamidou, S.; Pistikopoulos, E. N. On multi-parametric programming and its applications in process systems engineering. *Chemical Engineering Research and Design* **2016**, *116*, 61–82, DOI: doi:10.1016/j.cherd.2016.09.034.

(11) Blankenship, J. W.; Falk, J. E. Infinitely constrained optimization problems. *Journal of Optimization Theory and Applications* **1976**, *19*, 261–281, DOI: doi:10.1007/bf00934096.

(12) Mitsos, A. Global optimization of semi-infinite programs via restriction of the right-hand side. *Optimization* **2011**, *60*, 1291–1308, DOI: doi:10.1080/02331934.2010.527970.

(13) Stuber, M. D.; Barton, P. I. Semi-Infinite Optimization with Implicit Functions. *Industrial & Engineering Chemistry Research* **2014**, *54*, 307–317, DOI: doi:10.1021/ie5029123.

(14) Djelassi, H.; Mitsos, A. A hybrid discretization algorithm with guaranteed feasibility for the global solution of semi-infinite programs. *Journal of Global Optimization* **2017**, *68*, 227–253, DOI: doi:10.1007/s10898-016-0476-7.

(15) Tsoukalas, A.; Rustem, B. A feasible point adaptation of the Blankenship and Falk algorithm for semi-infinite programming. *Optimization Letters* **2010**, *5*, 705–716, DOI: doi:10.1007/s11590-010-0236-4.

(16) Mitsos, A.; Tsoukalas, A. Global optimization of generalized semi-infinite programs via restriction of the right hand side. *Journal of Global Optimization* **2014**, *61*, 1–17, DOI: doi:10.1007/s10898-014-0146-6.

(17) Djelassi, H.; Glass, M.; Mitsos, A. Discretization-based algorithms for generalized semi-infinite and bilevel programs with coupling equality constraints. *Journal of Global Optimization* **2019**, *75*, 341–392, DOI: doi:10.1007/s10898-019-00764-3.

(18) Djelassi, H.; Mitsos, A. Global Solution of Semi-infinite Programs with Existence Con-

straints. *Journal of Optimization Theory and Applications* **2021**, *188*, 863–881, DOI: doi:10.1007/s10957-021-01813-2.

(19) Djelassi, H.; Mitsos, A.; Stein, O. Recent advances in nonconvex semi-infinite programming: Applications and algorithms. *EURO Journal on Computational Optimization* **2021**, *9*, 100006, DOI: doi:10.1016/j.ejco.2021.100006.

(20) Zendehboudi, S.; Rezaei, N.; Lohi, A. Applications of hybrid models in chemical, petroleum, and energy systems: A systematic review. *Applied Energy* **2018**, *228*, 2539–2566, DOI: doi:10.1016/j.apenergy.2018.06.051.

(21) Stuber, M. D.; Barton, P. I. Robust simulation and design using semi-infinite programs with implicit functions. *International Journal of Reliability and Safety* **2011**, *5*, 378, DOI: doi:10.1504/ijrs.2011.041186.

(22) Kahrs, O.; Marquardt, W. The validity domain of hybrid models and its application in process optimization. *Chemical Engineering and Processing: Process Intensification* **2007**, *46*, 1054–1066, DOI: doi:10.1016/j.cep.2007.02.031.

(23) Lennartson, B.; Tittus, M.; Egardt, B.; Pettersson, S. Hybrid systems in process control. *IEEE Control Systems* **1996**, *16*, 45–56, DOI: doi:10.1109/37.537208.

(24) Doyle, F. J.; Harrison, C. A.; Crowley, T. J. Hybrid model-based approach to batch-to-batch control of particle size distribution in emulsion polymerization. *Computers & Chemical Engineering* **2003**, *27*, 1153–1163, DOI: doi:10.1016/s0098-1354(03)00043-7.

(25) Psichogios, D. C.; Ungar, L. H. A hybrid neural network-first principles approach to process modeling. *AIChE Journal* **1992**, *38*, 1499–1511, DOI: doi:10.1002/aic.690381003.

(26) Lee, D. S.; Jeon, C. O.; Park, J. M.; Chang, K. S. Hybrid neural network modeling of a full-scale industrial wastewater treatment process. *Biotechnology and Bioengineering* **2002**, *78*, 670–682, DOI: doi:10.1002/bit.10247.

(27) Thompson, M. L.; Kramer, M. A. Modeling chemical processes using prior knowledge and neural networks. *AIChE Journal* **1994**, *40*, 1328–1340, DOI: doi:10.1002/aic.690400806.

(28) Bollas, G.; Papadokonstadakis, S.; Michalopoulos, J.; Arampatzis, G.; Lappas, A.; Vasalos, I.; Lygeros, A. Using hybrid neural networks in scaling up an FCC model from a pilot plant to an industrial unit. *Chemical Engineering and Processing: Process Intensification* **2003**, *42*, 697–713, DOI: doi:10.1016/S0255-2701(02)00206-4.

(29) Schweidtmann, A. M.; Weber, J. M.; Wende, C.; Netze, L.; Mitsos, A. Obey validity limits of data-driven models through topological data analysis and one-class classification. *Optimization and Engineering* **2021**, 1–22, DOI: doi:https://doi.org/10.1007/s11081-021-09608-0.

(30) Belgiu, M.; Drăguţ, L. Random forest in remote sensing: A review of applications and future directions. *ISPRS Journal of Photogrammetry and Remote Sensing* **2016**, *114*, 24–31, DOI: doi:10.1016/j.isprsjprs.2016.01.011.

(31) Henao, C. A.; Maravelias, C. T. Surrogate-based superstructure optimization framework. *AIChE Journal* **2010**, *57*, 1216–1232, DOI: doi:10.1002/aic.12341.

(32) Schweidtmann, A. M.; Mitsos, A. Deterministic Global Optimization with Artificial Neural Networks Embedded. *Journal of Optimization Theory and Applications* **2018**, *180*, 925–948, DOI: doi:10.1007/s10957-018-1396-0.

(33) von Stosch, M.; Oliveira, R.; Peres, J.; de Azevedo, S. F. Hybrid semi-parametric modeling in process systems engineering: Past, present and future. *Computers & Chemical Engineering* **2014**, *60*, 86–101, DOI: doi:10.1016/j.compchemeng.2013.08.008.

(34) Aguiar, H. C.; Filho, R. M. Neural network and hybrid model: a discussion about different modeling techniques to predict pulping degree with industrial data. *Chemical Engineering Science* **2001**, *56*, 565–570, DOI: doi:10.1016/s0009-2509(00)00261-x.

(35) Ghaoui, L. E.; Gu, F.; Travacca, B.; Askari, A.; Tsai, A. Implicit Deep Learning. *SIAM Journal on Mathematics of Data Science* **2021**, *3*, 930–958, DOI: doi:10.1137/20m1358517.

(36) Bai, S.; Kolter, J. Z.; Koltun, V. Deep Equilibrium Models. Advances in Neural Information Processing Systems. 2019; https://proceedings.neurips.cc/paper/2019/file/01386bd6d8e091c2ab4c7c7de644d37b-Paper.pdf, accessed 2021-06-16.

(37) Winston, E.; Kolter, J. Z. Monotone operator equilibrium networks. Advances in Neural Information Processing Systems. 2020; pp 10718–10728, https://proceedings.neurips.cc/paper/2020/file/798d1c2813cbdf8bcdb388db0e32d496-Paper.pdf, accessed 2021-05-18.

(38) Gu, F.; Chang, H.; Zhu, W.; Sojoudi, S.; El Ghaoui, L. Implicit Graph Neural Networks. Advances in Neural Information Processing Systems. 2020; pp 11984–11995, https://proceedings.neurips.cc/paper/2020/file/8b5c8441a8ff8e151b191c53c1842a38-Paper.pdf, accessed 2021-06-16.

(39) Chen, R. T. Q.; Rubanova, Y.; Bettencourt, J.; Duvenaud, D. Neural Ordinary Differential Equations. 2019.

(40) He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016; pp 770–778, DOI: doi:10.1109/CVPR.2016.90.

(41) Kwak, B. M.; Haug, E. J. Optimum design in the presence of parametric uncertainty. *Journal of Optimization Theory and Applications* **1976**, *19*, 527–546, DOI: doi:10.1007/bf00934653.

(42) Halemane, K. P.; Grossmann, I. E. Optimal process design under uncertainty. *AIChE Journal* **1983**, *29*, 425–433, DOI: doi:10.1002/aic.690290312.

(43) Ostrovsky, G. M.; Achenie, L. E. K.; Wang, Y.; Volin, Y. M. A New Algorithm for Computing

Process Flexibility. *Industrial & Engineering Chemistry Research* **2000**, *39*, 2368–2377, DOI: doi:10.1021/ie9905207.

(44) Dimitriadis, V. D.; Pistikopoulos, E. N. Flexibility Analysis of Dynamic Systems. *Industrial & Engineering Chemistry Research* **1995**, *34*, 4451–4462, DOI: doi:10.1021/ie00039a036.

(45) Hale, W. T.; Wilhelm, M. E.; Palmer, K. A.; Stuber, M. D.; Bollas, G. M. Semi-infinite programming for global guarantees of robust fault detection and isolation in safety-critical systems. *Computers & Chemical Engineering* **2019**, *126*, 218–230, DOI: doi:10.1016/j.compchemeng.2019.04.007.

(46) Wilhelm, M. E.; Le, A. V.; Stuber, M. D. Global optimization of stiff dynamical systems. *AIChE Journal* **2019**, *65*, DOI: doi:10.1002/aic.16836.

(47) Epperly, T. G. W.; Pistikopoulos, E. N. A reduced space branch and bound algorithm for global optimization. *Journal of Global Optimization* **1997**, *11*, 287–311, DOI: doi:10.1023/A:1008212418949.

(48) Stuber, M. D. Evaluation of Process Systems Operating Envelopes. Ph.D. thesis, Massachusetts Institute of Technology, 2012.

(49) Mitsos, A.; Chachuat, B.; Barton, P. I. McCormick-Based Relaxations of Algorithms. *SIAM Journal on Optimization* **2009**, *20*, 573–601, DOI: doi:10.1137/080717341.

(50) Wechsung, A. Global optimization in reduced space. Ph.D. thesis, Massachusetts Institute of Technology, 2014.

(51) Bongartz, D.; Mitsos, A. Deterministic global optimization of process flowsheets in a reduced space using McCormick relaxations. *Journal of Global Optimization* **2017**, *69*, 761–796, DOI: doi:10.1007/s10898-017-0547-4.

(52) Ryoo, H. S.; Sahinidis, N. V. A branch-and-reduce approach to global optimization. *Journal of Global Optimization* **1996**, *8*, 107–138, DOI: doi:10.1007/bf00138689.

(53) Tawarmalani, M.; Sahinidis, N. V. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming* **2005**, *103*, 225–249, DOI: doi:10.1007/s10107-005-0581-8.

(54) Horst, R.; Tuy, H. *Global Optimization: Deterministic Approaches*; Springer Science & Business Media: Berlin, Germany, 2013; pp 111–169.

(55) Puranik, Y.; Sahinidis, N. V. Domain reduction techniques for global NLP and MINLP optimization. *Constraints* **2017**, *22*, 338–376, DOI: doi:10.1007/s10601-016-9267-5.

(56) Khan, K. A.; Watson, H. A. J.; Barton, P. I. Differentiable McCormick relaxations. *Journal of Global Optimization* **2016**, *67*, 687–729, DOI: doi:10.1007/s10898-016-0440-6.

(57) Khan, K. A.; Wilhelm, M.; Stuber, M. D.; Cao, H.; Watson, H. A. J.; Barton, P. I. Corrections to: Differentiable McCormick relaxations. *Journal of Global Optimization* **2018**, *70*, 705–706, DOI: doi:10.1007/s10898-017-0601-2.

(58) Scott, J. K.; Stuber, M. D.; Barton, P. I. Generalized McCormick relaxations. *Journal of Global Optimization* **2011**, *51*, 569–606, DOI: doi:10.1007/s10898-011-9664-7.

(59) Scott, J. K.; Barton, P. I. Improved relaxations for the parametric solutions of ODEs using differential inequalities. *Journal of Global Optimization* **2013**, *57*, 143–176, DOI: doi:10.1007/s10898-012-9909-0.

(60) Song, Y.; Khan, K. A. Optimization-based convex relaxations for nonconvex parametric systems of ordinary differential equations. *Mathematical Programming* **2021**, 1–45, DOI: doi:10.1007/s10107-021-01654-x.

(61) Stuber, M. D.; Scott, J. K.; Barton, P. I. Convex and concave relaxations of implicit functions. *Optimization Methods and Software* **2015**, *30*, 424–460, DOI: doi:10.1080/10556788.2014.924514.

(62) Misener, R.; Floudas, C. A. ANTIGONE: Algorithms for continuous/integer global optimization of nonlinear equations. *Journal of Global Optimization* **2014**, *59*, 503–526, DOI: doi:10.1007/s10898-014-0166-2.

(63) Wilhelm, M. E.; Stuber, M. D. EAGO.jl: easy advanced global optimization in Julia. *Optimization Methods and Software* **2020**, 1–26, DOI: doi:10.1080/10556788.2020.1786566.

(64) Bongartz, D.; Najman, J.; Sass, S.; Mitsos, A. MAiNGO: McCormick based algorithm for mixed integer nonlinear global optimization. 2018; https://git.rwth-aachen.de/avt-svt/public/maingo, accessed 2021-05-05.

(65) Kearfott, R. B.; Castille, J.; Tyagi, G. A general framework for convexity analysis in deterministic global optimization. *Journal of Global Optimization* **2013**, *56*, 765–785, DOI: doi:https://doi.org/10.1007/s10898-012-9905-4.

(66) Bezanson, J.; Edelman, A.; Karpinski, S.; Shah, V. B. Julia: A Fresh Approach to Numerical Computing. *SIAM Review* **2017**, *59*, 65–98, DOI: doi:10.1137/141000671.

(67) Sanders, D. P. et al. JuliaIntervals/IntervalArithmetic.jl: v0.18.2. 2021; https://doi.org/10.5281/zenodo.4739394, accessed 2021-05-05.

(68) Fedorov, G.; Nguyen, K. T.; Harrison, P.; Singh, A. Intel Math Kernel Library 2019 Update 2 Release Notes. 2019.

(69) Anderson, E.; Bai, Z.; Bischof, C.; Blackford, L. S.; Demmel, J.; Dongarra, J.; Croz, J. D.; Greenbaum, A.; Hammarling, S.; McKenney, A.; Sorensen, D. *LAPACK Users' Guide*, 3rd ed.; Society for Industrial and Applied Mathematics: Philadelphia, PA, 1999; DOI: doi:10.1137/1.9780898719604.

(70) Wang, E.; Zhang, Q.; Shen, B.; Zhang, G.; Lu, X.; Wu, Q.; Wang, Y. *High-Performance Computing on the Intel® Xeon Phi™*; Springer International Publishing: New York, NY, 2014; pp 167–188, DOI: doi:10.1007/978-3-319-06486-4_7.

(71) Blackford, L. S.; Petitet, A.; Pozo, R.; Remington, K.; Whaley, R. C.; Demmel, J.; Dongarra, J.; Duff, I.; Hammarling, S.; Henry, G.; Heroux, M. An updated set of basic linear algebra subprograms (BLAS). *ACM Transactions on Mathematical Software* **2002**, *28*, 135–151, DOI: doi:10.1145/567806.567807.

(72) Juretschko, S.; Timmermann, G.; Schmid, M.; Schleifer, K.-H.; Pommerening-Röser, A.; Koops, H.-P.; Wagner, M. Combined Molecular and Conventional Analyses of Nitrifying Bacterium Diversity in Activated Sludge: Nitrosococcus mobilis and Nitrospira-Like Bacteria as Dominant Populations. *Applied and Environmental Microbiology* **1998**, *64*, 3042–3051, DOI: doi:10.1128/aem.64.8.3042-3051.1998.

(73) Siripong, S.; Rittmann, B. E. Diversity study of nitrifying bacteria in full-scale municipal wastewater treatment plants. *Water Research* **2007**, *41*, 1110–1120, DOI: doi:10.1016/j.watres.2006.11.050.

(74) Sánchez, O.; Aspé, E.; Martí, M. C.; Roeckel, M. The Effect of Sodium Chloride on the Two-Step Kinetics of the Nitrifying Process. *Water Environment Research* **2004**, *76*, 73–80, DOI: doi:10.2175/106143004x141609.

(75) Baird, R. B., Eaton, A. D., Rice, E. W., Eds. *Standard Methods for the Examination of Water and Wastewater*; American Public Health Association, American Water Works Association, and Water Environment Foundation: Washington, D.C., 2017.

(76) Uby, L. Next steps in clean water oxygen transfer testing – A critical review of current standards. *Water Research* **2019**, *157*, 415–434, DOI: doi:10.1016/j.watres.2019.03.063.

(77) He, Z.; Petiraksakul, A.; Meesapya, W. Oxygen-transfer measurement in clean water. *The Journal of KMITNB* **2003**, *13*, 14–19.

(78) Wiesmann, U. *Biotechnics/Wastewater*; Springer Berlin Heidelberg: Heidelberg, Germany, 1994; pp 113–154, DOI: doi:10.1007/bfb0008736.

(79) Chollet, F., et al. Keras. 2015; https://keras.io, Software available from keras.io, accessed on 2021-06-28.

(80) Abadi, M. et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015; https://www.tensorflow.org/, Software available from tensorflow.org, accessed 2021-06-28.

(81) Bishop, C. M. *Pattern Recognition and Machine Learning*; Springer-Verlag: New York, NY, 2006; Vol. 4; pp 232–240.

(82) Henze, M.; van Loosdrecht, M. C.; Ekama, G. A.; Brdjanovic, D. *Biological Wastewater Treatment: Principles, Modelling, and Design*; IWA publishing: London, UK, 2008; pp 35–36.

(83) Higgins, W.; Jim Kern, P. Too Much Air? Understanding the Critical Role of Aeration Systems. 2016; https://www.tpomag.com/editorial/2016/03/too_much_air_understanding_the_critical_role_of_aeration_systems, accessed 2021-06-16.

(84) Wilhelm, M. E.; Wang, C.; Stuber, M. D. Convex and concave envelopes of artificial neural network activation functions for deterministic global optimization. under review.

(85) Stuber, M. D.; Wechsung, A.; Sundaramoorthy, A.; Barton, P. I. Worst-case design of subsea production facilities using semi-infinite programming. *AIChE Journal* **2014**, *60*, 2513–2524, DOI: doi:10.1002/aic.14447.

(86) Avriel, M.; Golany, B. *Mathematical programming for industrial engineers*; CRC Press: New York, NY, 1996; Vol. 20; p 446.

(87) Wang, X.; Du, G.; Wu, M.; Song, Q.; Chen, Y. Two-step Robust Design of LLC Converter with Corner Judgement and Greedy Algorithm. *IEEE Transactions on Power Electronics* **2021**, 1–1, DOI: doi:10.1109/tpel.2021.3139049.

(88) Fernández-Alcázar, L. J.; Kononchuk, R.; Kottos, T. Enhanced energy harvesting near exceptional points in systems with (pseudo-)PT-symmetry. *Communications Physics* **2021**, *4*, DOI: doi:10.1038/s42005-021-00577-5.

(89) Schichl, H.; Neumaier, A. Interval Analysis on Directed Acyclic Graphs for Global Optimization. *Journal of Global Optimization* **2005**, *33*, 541–562, DOI: doi:10.1007/s10898-005-0937-x.

(90) Neumaier, A. *Interval Methods for Systems of Equations*; Cambridge University Press: Cambridge, UK, 1991; Vol. 37; pp 171–173, DOI: doi:10.1017/cbo9780511526473.

(91) Wächter, A.; Biegler, L. T. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming* **2005**, *106*, 25–57, DOI: doi:10.1007/s10107-004-0559-y.

(92) Bae, J.; ji Lee, H.; Jeong, D. H.; Lee, J. M. Construction of a Valid Domain for a Hybrid Model and Its Application to Dynamic Optimization with Controlled Exploration. *Industrial & Engineering Chemistry Research* **2020**, *59*, 16380–16395, DOI: doi:10.1021/acs.iecr.0c02720.

(93) Quaglio, M.; Fraga, E. S.; Cao, E.; Gavriilidis, A.; Galvanin, F. A model-based data mining approach for determining the domain of validity of approximated models. *Chemometrics and Intelligent Laboratory Systems* **2018**, *172*, 58–67, DOI: doi:10.1016/j.chemolab.2017.11.010.

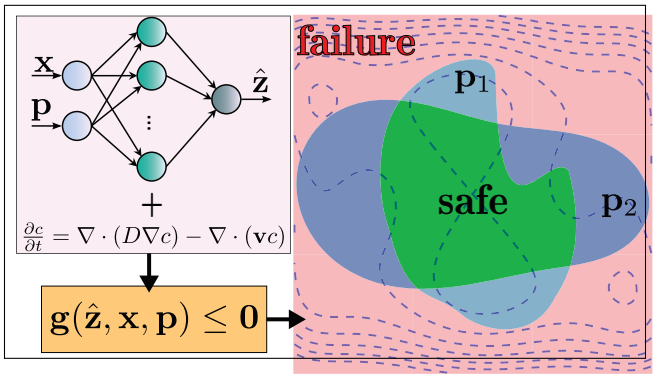
(94) Najman, J.; Mitsos, A. Convergence analysis of multivariate McCormick relaxations. *Journal of Global Optimization* **2016**, *66*, 597–628, DOI: doi:10.1007/s10898-016-0408-6.

(95) Wilhelm, M.; Stuber, M. D. Easy Advanced Global Optimization (EAGO): An open-source platform for robust and global optimization in Julia. 2017; AIChE Annual Meeting.

(96) Wilhelm, M.; Stuber, M. D. Recent Advances in the EaGO Platform: Global and Robust Optimization in Julia. 2018; AIChE Annual Meeting.

(97) Krawczyk, R. Interval iterations for including a set of solutions. *Computing* **1984**, *32*, 13–31, DOI: doi:10.1007/BF02243016.

TOC Graphic


x
p
ẑ
+
$\frac{\partial c}{\partial t} = \nabla \cdot (D\nabla c) - \nabla \cdot (\mathbf{v}c)$
$\mathbf{g}(\hat{\mathbf{z}}, \mathbf{x}, \mathbf{p}) \leq \mathbf{0}$
failure
$\mathbf{p}_1$
safe
$\mathbf{p}_2$