Available online at www.sciencedirect.com
ScienceDirect
Computers and Chemical Engineering 31 (2007) 361–373
Computers & Chemical Engineering
www.elsevier.com/locate/compchemeng

A semi-mechanistic model building framework based on selective and localized model extensions

Pedro V. Lima, Pedro M. Saraiva*
GEPSI-PSE Group
Department of Chemical Engineering, University of Coimbra, Pinhal de Marrocos, 3030-290 Coimbra, Portugal

Received 5 March 2006; received in revised form 22 July 2006; accepted 24 July 2006
Available online 11 September 2006

Abstract

In the core of many process systems engineering tasks, like design, control, optimization and fault diagnosis, a mathematical model of the underlying plant plays a key role. Such models are so important that extensive studies are available, recommending different modeling techniques to be adopted for specific processes or goals. It is usual and practical to split modeling techniques under two main groups: mechanistic methods and empirical or statistical methods. Both paradigms have been adopted, but very few frameworks were developed to combine and integrate features from both of them. In this article we describe a framework for data-driven evolution of static mechanistic models with a selective inclusion of simple empirical terms. To illustrate its practical potential, our framework is applied to the identification of a non-ideal reactor and to the optimization of the Otto–Williams benchmark reactor.

Keywords: Semi-mechanistic model; Hybrid model; Modelling; Parameter identification; Optimization; Simulation

1. Introduction

The construction of mathematical models to forecast and understand the behavior of chemical processes forms the basis for a countless number of process systems engineering tasks (planning, optimization, improvement, fault diagnosis, control, etc.). Depending on the nature of the specific process and its desired goals, several kinds of models have been developed, differing, namely in scope, level of detail, and underlying structures.

However, for many practical situations two separate schools of thought have emerged. On one hand, we have fully mechanistic approaches, where models are built based upon first-principles phenomena. On the other hand, and specially for operation improvement at existing plants with complex transformations, fully empirical techniques have also been suggested and applied, relying in operators knowledge extraction, data analysis based upon machine learning (Saraiva, 1996) or statistical tools. But only a few efforts have been done in the past to combine and integrate both of the above paradigms, although they are conceptually believed to be complementary to each other: empirical components will in general lead to better local prediction capabilities through the full exploration of all information that is available, while mechanistic elements make it possible for one to get a better understanding of the underlying physico-chemical phenomena, predict the values for unmeasured state variables, provide additional trust, reliability and extrapolation characteristics.

One of the most widespread techniques employed for plant optimization is response surface analysis (Box & Draper, 1987) based on experimentation and empirical modeling. But experiments are often expensive, its number tends to grow exponentially with the number of relevant process variables, and empirical models are known to fail on wide operation domains. The use of mechanistic model-based optimization (Garcia & Morari, 1981) has been advocated as an alternative solution. A priori process knowledge reduces the need for experimental data and optimization can be carried out over a larger space of

* Corresponding author. Tel.: +351 239798700, fax: +351 239798703.
E-mail addresses: eq3pvl@eq.uc.pt (P.V. Lima), eq1pas@eq.uc.pt (P.M. Saraiva).

0098-1354/$ – see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compchemeng.2006.07.006

Nomenclature

a, b	CSTR kinetic parameters for the Arrhenius expression
A, B	CSTR output concentration of A and B
A_0	CSTR inlet concentration of A
F_i	Otto reactor mass flow for species A and B
F_R	Otto reactor total mass flow
IMMS	initial mechanistic model structure
k_e	CSTR reaction heat constant
k_i, a_i, b_i	Otto reactor Arrhenius kinetic parameters
k_p	CSTR heat transfer constant
M	Otto reactor mass content
PMM	perfect mechanistic model
$\mathcal{P}_2$	second degree polynomial
SMMS	semi-mechanistic model structure
T	CSTR output temperature
T_p	CSTR heat transfer temperature
T_r	CSTR reference temperature
T_0	CSTR inlet temperature
v_n	neutral value reformulation variable
V	multi-response data covariance matrix
w_i	new internal variable
$\mathbf{x}$	model variables
X_i	Otto reactor mass fraction for component i
$\mathbf{y}$	response variables
$\mathbf{y_m}$	response measurements

Greek symbols

Φ	extension set
α_i	extension slot in the model structure
δ_i	CSTR stream fractions
ε_{ki}	CSTR prediction error for variable k and measurement i
τ	CSTR residence time
θ	model parameters

process variables. However, the mismatch between mechanistic model predictions and plant data leads to results that can be quite suboptimal when applied to the real plant. For plant optimization, there is thus a clear need for strategies that effectively incorporate both mechanistic and experimental knowledge.

When doing statistical based plant optimization, one often notices that the amount of information needed to build adequate empirical models requires a lot of time and resources. Furthermore, because of safety considerations, the search space for experimentation in real plants is somewhat limited around regular operating conditions, and in doing so limits some improvement potential that may be associated with other areas of the decision space. This situation makes the usage of mechanistic model-based techniques very important for some plants.

The main problem quite often consists of being able to derive a fully mechanistic model good enough to support decisionmaking. In real plant optimization, there is a clear need for combining known mechanistic models together with available operating data, in a way that goes further than just doing parameter adjustment in the mechanistic model. But when using a parallel or serial combination of a mechanistic model and an artificial neural network (or other empirical based modules) the resulting model behavior in extrapolation is known to be unstable and non-reliable (Braake, van Can, & Verbruggen, 1998). With the internal simple empirical extensions that we will propose in our framework, to be described in this paper, it is possible for one to keep the main relations between variables, provided by an initial mechanistic model of the plant, but obtain from it a derived model with a better data fitting and prediction capabilities, and therefore able to provide a closer representation of a real industrial plant. The extrapolation stability of the hybrid structures thus obtained is an important characteristic for model-based plant optimization.

Model-based plant optimization can also performed online, using a real-time optimization loop, based upon this kind of semi-mechanistic models. Forbes, Marlin, and MacGregor (1994)show that the choice of an appropriate mechanistic model structure is very important and certain criteria must be met in order to obtain convergence to the plant optimal conditions. A fully mechanistic model built with reasonable considerations may however fail completely in real-time optimization of industrial units, due to approximations and lack of data fit that originate an error surface that does not enable convergence to the real world plant optimal conditions. The hybrid mechanistic–empirical models that we propose have both parameter and structure adjustment capabilities, and therefore are able to result in improved optimization solutions, as we will see in one of the case studies to be presented.

In the limited previous work that has been done in order to build hybrid mechanistic–empirical models (Cameron & Hangos, 2001; Psichogios & Ungar, 1992; Schubert, Simutis, Dors, Havlík, & Lubbert, 1994; Su, Bhat, & McAvoy, 1992; Thompson & Kramer, 1994; Van Can, Hellinga, Luyben, & Heijnen, 1996;), one may find different perspectives, including the choice among alternative first-principles models followed by some parameter fitting to existing data or combination of the available mechanistic model with an empirical model adjusted to the residuals. Research in the area of system identification also introduced the concept of grey-box models (Lindskog & Ljung, 1995; Ljung, 1999) for representing models with physical components (white-box models) and empirical components (black-box).

However, none of them take into account the detailed fine structure of the available underlying mechanistic model, neither do they allow for partial localized and selective introduction of empirical elements. In this article we present a framework for building semi-mechanistic models with such capabilities.

In order to state, in a simple language, the goals and practical value of our work, one may say that given an initial mechanistic model structure (IMMS), and sets of collected industrial data from the plant that such a model is trying to represent (D), we try to build alternative semi-mechanistic model structures (SMMS) that make use of available phenomenological knowledge but build on top of it in order to come up with model structures that lead to a closer approximation to the plant reality, as expressed

by the available data. Our framework provides a decision support system to guide the user in the construction and test of such alternative model structures. One main difference between our framework and genetic programming is that our search space is constructed in a way that reduces the number of possible combinations to the ones that introduce simple and local changes to the original IMMS model structure. While genetic programming is able generate new complex structures, our framework is focused in finding simple extensions that build on top of the knowledge and vocabulary of terms connected with the initially available mechanistic features, expressed by IMMS. We will only deviate from it as much as is needed to come up with acceptable data fitting performances.

2. Semi-mechanistic modeling framework

As stated above, given a set D of nd industrial or experimental data records (x, y), where x are system inputs and y stand for the variables that we want a model to be able to predict, our goal is to come up with a final model structure and set of associated parameters (SMMS), able to fit the available data, and having as starting point an initial mechanistic model structure, which is however to provide an acceptable forecasting performance within the scope of its underlying structure. The search space that we will consider therefore covers both model structures and the sets of corresponding adjustable np parameters, θ. The search space of model structures that we will consider is highly constrained by the mechanistic knowledge that is available and expressed under the form of our IMMS. Since we will be using real data for parameter estimation, one must also take into account that the number of available data records must be an order of magnitude larger than the number of adjustable parameters.

Therefore, on a more formal notation, one can say that the generic representation used in our semi-mechanistic model identification problem corresponds to a non-linear programming formulation (NLP). It allows for the explicit consideration of model restrictions and differential-algebraic systems solved through discretization techniques. Our NLP representation for a multi-response least squares parameter optimization is as follows:

$$\begin{aligned} &\min_\theta \textstyle\sum (\mathbf{y_m} - \mathbf{y})^{\mathrm{T}} V^{-1} (\mathbf{y_m} - \mathbf{y}) \\ &\text{s.t.} \quad f(\mathbf{y}, \mathbf{x}, \theta) = 0 \\ &\qquad g(\mathbf{y}, \mathbf{x}, \theta) \leq 0 \end{aligned} \tag{1}$$

where f are the linear and non-linear model equations, either from the initial mechanistic model structure or other alternatives derived from it, g the model inequalities, $\mathbf{y}$ the response variables, $\mathbf{y_m}$ the response measurements, $\mathbf{x}$ the remaining model variables, θ the model parameters and V is the multi-response data covariance matrix (Bates & Watts, 1988).

We now focus on the process model equations. A standard approach is to use mechanistic model equations directly in f. But work done on constraint reformulation for convex relaxation of NLP models has shown that a generic model Eq. (2) can be transformed to an equivalent model structure that has better characteristics for the development of optimization algorithms (Quesada & Grossmann, 1995). For semi-mechanistic model identification we also propose such a transformation, in order to expose the model linear and non-linear basic building blocks. As an example of our reformulation strategy, let us take, for instance, the non-linear kinetic Eq. (2), where a, b and c are constants and r, x, y and z are the variables, as our initial mechanistic model structure. Through the introduction of new variables w_i, Eq. (2) can be represented as a set of equations as shown next:

$$r = \frac{a\, \mathrm{e}^{b/x}\, y\, z^2}{c + y + z} \equiv \begin{bmatrix} w_1 = \dfrac{b}{x} \\ w_2 = \mathrm{e}^{w_1} \\ w_3 = z^2 \\ w_4 = a w_2 y w_3 \\ w_5 = c + y + z \\ r = \dfrac{w_4}{w_5} \end{bmatrix} \tag{2}$$

The new equations in (2) are either linear equations, product terms, fractions, powers or unary functions (like exp, log, etc.). We thereby propose a representation similar to the one used for global optimization (Smith & Pantelides, 1999), but with reformulated product equations having any number of terms. This proposed representation lends to less sparse reformulated model structures, better suited for semi-mechanistic model identification and construction.

2.1. Symbolic reformulation algorithm

In this section we present the algorithm that we use for reformulation of an initial set of equations. This symbolic algorithm is based on the binary tree representation (Knuth, 1973) of the equation expression. The binary tree is an acyclic graph data structure with three types of nodes (one root node, internal nodes and leaf nodes) and arcs connecting the nodes. The binary tree representation of the kinetic expression (2) is shown in Fig. 1.

The algorithm goes through the tree nodes by recursive calling of the function presented in Algorithm 1.

Algorithm 1. Algorithm for symbolic reformation.

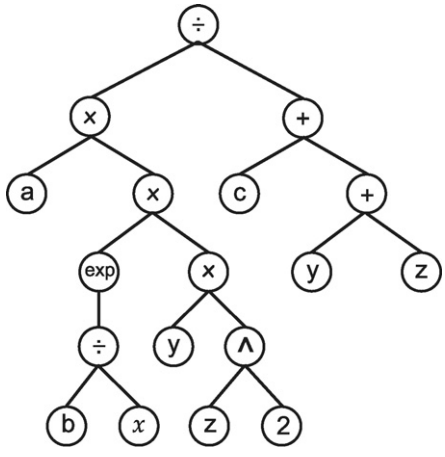

Fig. 1. Binary tree representation.

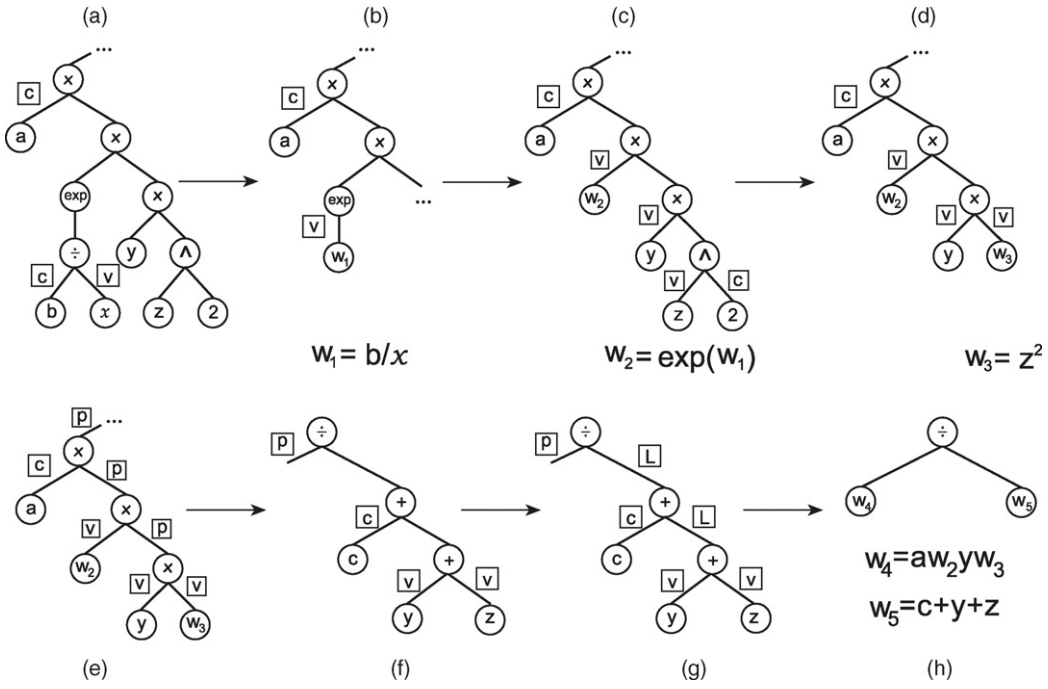

Fig. 2. Reformulation algorithm.

As an example, we will apply the algorithm to the binary tree of expression (2), under the steps illustrated in Fig. 2. The algorithm starts from the root node and goes recursively, top–down and left–right until it gets to a leaf node. The first leaf found is constant a and the recursive function returns the type C (constant) for the branch (Fig. 2a). Recursion continues with the other branch until it stops on b/x. The left branch is type C and the right branch is type V (variable). From Table 1, for the division operator, we obtain the action to create a new fraction definition, $w_1 = b/x$. The return type is V since a new variable is placed on the tree (Fig. 2b). Then, from the main function code, we have a rule to create function definitions for functions applied to V types ($w_2 = e^{w_4}$). The recursive process continues to z^2 with the result of $w_3 = z^2$ obtained from the corresponding table rule (Fig. 2c). Then, the table rule for product of two variables is to identify the subtree with type P (product). The same happens with the two upper nodes (Fig. 2d). Processing in the root right branch goes down to the leaf nodes $y + z$ and from table rule one identifies the type L (linear). Next, the constant addition is still a subtree of type L, and we get to the root node, with a left subtree of type P and a right subtree of type L. From the corresponding table rule, a new product definition is created for the left subtree $w_4 = aw_2yw_3$, a new linear definition is created for the right subtree $w_5 = c + y + z$ and then a fraction w_5/w_4 definition is created for the root node. This concludes the reformulation with a resulting expression equal to the one presented originally in (2).

2.2. Semi-mechanistic model hyperstructure

Given an initial mechanistic model structure, its reformulation performed according to the steps mentioned in the previous section sets the stage for defining the search space for finding out alternative semi-mechanistic model structures that are better adjusted to the available set of data records (D). Such a search space corresponds to the IMMS hyperstructure, i.e., the one that includes the initial model together with all possible extension slots over our initial set of equations, after IMMS has been reformulated and represented according to the sets of equations reported in Section 2.1. This kind of structure is comparable to the ones used in MINLP flowsheet optimization (Grossmann & Sargent, 1978).

Such an hyperstructure is created by adding to our initial mechanistic model all the places where extensions will be allowed (extension slots). With the model in the reformulated form, the insertion is done using a set of rules applied to the basic expression types (linear, product, fraction, etc.). For example, linear expressions like $c + y + z$ can be transformed to $c + y + z + \alpha$ with α representing an extension slot in the model.

There are many possible strategies for determining the extension slots under consideration. One simple approach is to only add slots to linear expressions like in the previous example, making the identification problem similar to linear stepwise regression. The strategy that we propose introduces extension slots in the model as addition terms, according to the simple rules:

(1) For linear definitions, add a single extension slot to the expression. For example, $x + y$ becomes $x + y + \alpha$.
(2) For non-linear definitions, add an extension slot next to each variable. For example, $e^x \rightarrow e^{x+\alpha}$, $xy \rightarrow (x + \alpha_1)(y + \alpha_2)$.

As an illustration, by applying these rules to the reformulated model (2) one originates hyperstructure (3):

Table 1
Rules for binary operators reformulation (plus, times, divide and power) under the following nomenclature: operator (Op), left child type (LT), right child type (RT), left child action (LA), right child action (RA), node action (NA), node result type (N), constant type (C), variable type (V), linear type (L), product (P), new linear definition (a), new product definition (b), new fraction definition (c) and new power definition (d)

Op	LT	RT	LA	RA	NA	N	Op	L	R	LA	RA	NA	N
±	C	C				C	×	C	C				C
	V	C				L		V	C				L
	L	C				L		L	C				L
	P	C	b			L		C	V				L
	C	V				L		C	L				L
	V	V				L		P	C				P
	L	V				L		V	V				P
	P	V	b			L		L	V	a			P
	C	L				L		P	V				P
	V	L				L		V	L		a		P
	L	L				L		L	L	a	a		P
	P	L	b			L		P	L		a		P
	C	P		b		L		C	P				P
	V	P		b		L		V	P				P
	L	P		b		L		L	P	a			P
	P	P	b	b		L		P	P				P
÷	C	C				C	∧	C	C				C
	V	C				L		V	C			d	V
	L	C				L		L	C	a		d	V
	P	C				P		P	C	b		d	V
	C	V			c	V		C	V			d	V
	V	V			c	V		V	V			d	V
	L	V	a		c	V		L	V	a		d	V
	P	V	b		c	V		P	V	b		d	V
	C	L		a	c	V		C	L		a	d	V
	V	L		a	c	V		V	L		a	d	V
	L	L	a	a	c	V		L	L	a	a	d	V
	P	L	b	a	c	V		P	L	b	a	d	V
	C	P		b	c	V		C	P		b	d	V
	V	P		b	c	V		V	P		b	d	V
	L	P	a	b	c	V		L	P	a	b	d	V
	P	P	b	b	c	V		P	P	b	b	d	V

$$
\begin{aligned}
w_1 &= \frac{b}{x} & & w_1 = \frac{b}{x+\alpha_1} \\
w_2 &= \mathrm{e}^{w_1} & & w_2 = \mathrm{e}^{w_1+\alpha_2} \\
w_3 &= z^2 & & w_3 = (z+\alpha_3)^2 \\
w_4 &= a w_2 y w_3 & \rightarrow \quad & w_4 = a\,(w_2+\alpha_4)\,(y+\alpha_5)\,(w_3+\alpha_6) \\
w_5 &= c + y + z & & w_5 = c + y + z + \alpha_7 \\
r &= \frac{w_4}{w_5} & & r = \frac{w_4+\alpha_8}{w_5+\alpha_9}
\end{aligned}
\tag{3}
$$

In some situations it may be interesting to adopt a model hyperstructure with some particular characteristics. One of such modification consists of disabling extensions in some equations, in order to maintain some known system properties. As an example of this scenario, we may want to preserve mass conservation laws from the original mechanistic model. Another modification we might want to consider is an increase in the number of slots in some parts of the model. This can be achieved through a variable transformation before the reformulation, $\mathcal{T}(x, v_n)$, where v_n is a dummy variable, replaced with a neutral value after the reformulation. For example, by changing x to x/v_n before reformulation, we obtain the expression $(x+\alpha_1)/(v_n+\alpha_2)$ in the hyperstructure. Then, by changing v_n to the neutral value we obtain the expression $(x+\alpha_1)/(1+\alpha_2)$, equivalent to the original model, but with two additional slots. In a CSTR example that we present in Section 3 we will apply this type of modification to increase the number of slots considered in the kinetic law.

2.3. Extension set

Each extension slot can accommodate one or more terms from the extension set. In the simplest case we may consider only the addition of new free parameters. We represent the extension set for free parameters as $\Phi = \{1\}$, meaning that each α slot can have a new parameter p multiplied by 1. This simple set can be useful for first exploratory runs in the search for model modifications.

The extension set we propose for developing semi-mechanistic models includes free parameters, the mechanistic model variables and the reformulated non-linear definitions having only variables from the mechanistic model. When the reformulation exposes terms like $1/x$ or xy we consider these terms as part of our basic variable set. When working with non-linear systems, it is important for the extension set to be able to model non-linear behavior. With that goal in mind, we also include in the extension set the quadratic forms of the previous terms, in

```
Function: reformulate
Input: tree
Output: type
if tree.nodeType ∈ {Constant, Number} then
 | return C;
else if tree.nodeType is Variable then
 | return V;
else if tree.nodeType is Operator then
 | if tree.operatorType is Unary then
 | | type ← reformulate(tree.child);
 | | if type is C then
 | | | return C;
 | | else if type is V then
 | | | newvar ← createDefinition(tree);
 | | | replace(tree, newvar);
 | | | return V;
 | | else if type ∈ {L, P} then
 | | | newvar ← createDefinition(tree.child);
 | | | replace(tree.child, newvar);
 | | | newvar ← createDefinition(tree);
 | | | replace(tree, newvar);
 | | | return V;
 | | end
 | else
 | | leftType ← reformulate(tree.leftChild);
 | | rightType ← reformulate(tree.rightChild);
 | | application of table 1 rules;
 | | if tree.operatorType ∈ {+, −} then
 | | | if leftType is C and rightType is C then
 | | | | return C;
 | | | else if leftType is V and rightType is C then
 | | | | return L;
 | | | else if ... then
 | | | | ...
 | | | end
 | | else if tree.operatorType is * then
 | | | if leftType is C and rightType is C then
 | | | | return C;
 | | | else if ... then
 | | | | ...
 | | | end
 | | else if ... then
 | | | ...
 | | end
 | end
end
```

Algorithm 1: Algorithm for symbolic reformulation.

order to introduce some curvature in the internal extensions.

As an example, the extension set for expression (2) results as shown:

$$\Phi = \left\{1, x, y, z, \frac{1}{x}, x^2, y^2, z^2, \left(\frac{1}{x}\right)^2\right\} \tag{4}$$

The hyperstructure, together with the extension set, make up the search space for finding semi-mechanistic model structures, i.e., for identification of possible SMMS solutions. The combinatorial problem size for adding one new parameter to the mechanistic model is $n \times m$ with n being the number of extension slots and m being the number of elements in the extension set.

The terms in the extension set are the elements needed to introduce quadratic polynomials in the mechanistic model structure. These elements may be introduced individually in the model or we can choose to add quadratic polynomials directly in the extensions. Both strategies will be used in the case studies: the first approach has the advantage of originating models with a smaller number of new parameters, while the semi-mechanistic models based on the polynomial approach use a reduced number of extension slots and are easier to compute, since their search space is smaller.

2.4. Optimization algorithm

The mechanistic model hyperstructure and its extension set define a disjunctive search space (Turkey & Grossmann, 1996). Disjunctive optimization problems are often associated with models having logical conditions or decision variables (Bjrkqvist & Westerlund, 2001). In our optimization problem, the disjunctive spaces correspond to the multiple possible extensions.

This kind of optimization problems can be transformed into mixed integer problems formulations (MINLP) and solved using relaxation and branch and bound techniques. Nevertheless, when solving relaxed non-linear models using local optimization, the branch and bound strategy may cut important parts of the search tree due to non-global convergence.

The search for semi-mechanistic models is an optimization problem that involves non-convexities from the non-linear model and from the considered non-linear extensions. In practice, as expected, we found it to be very hard to find the global optimum for the hyperstructure identification search space, but we also noticed that very good results could be obtained effectively using an approximated search strategy. On one hand, by using sequential introduction of new extensions in the model, the combinatorial space is considerably reduced. Also, the sequential solution of several non-linear optimizations can be very effective, with variables starting values for the next step inherited from the previous one. This was the case for the semi-mechanistic structures, that are in a close range of the base model structure, and thus the optimization solution for each sequential model extension can be obtained by exhaustive search in the disjunctive space. This sequential/exhaustive solution of MINLP problems was the strategy used to solve the case studies presented in the next sections. The new parameters are added sequentially to the model until one of the conditions set for the stop criterion is met: the error improvement is lower than a minimum value, the relative error goal is achieved or the maximum number of new extensions is reached.

As our hybrid model is built around a mechanistic model, enhanced with simple empirical structures that have a low number of new adjustable parameters,our SMMS possible solutions are compact in the number of parameters. Therefore, the sizes of the datasets (D) that we need to have in order to support model structure building and refinement do not have to be very large. In fact, just like for the determination of adjustable parameters in mechanistic models, it may be better to use a smaller data set by doing some data preprocessing in order to improve performance and stability of the optimization search.

In the next sections we will describe the application of our semi-mechanistic model building framework and evaluate its performance when applied to a pair of simulated case studies. So that we can benchmark performances with the best possible answer, corresponding to a perfect mechanistic model, our comparisons are carried this way: at first, a perfect mechanistic model (PMM) is used just for generating simulated operating

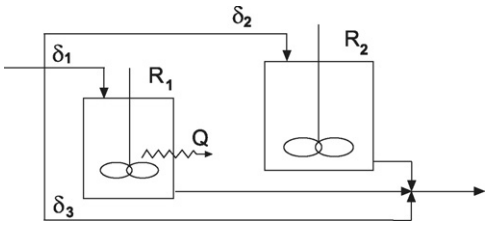

Fig. 3. Non-ideal CSTR assumed real plant behavior.

data (D), replacing the chemical plant as a data creation source; then, to emulate real world situations, we forget the existence of PMM (since no perfect models are usually available), adopted just to build D; given the nature of the plant and available mechanistic knowledge about it, an initial mechanistic model structure is adopted, corresponding to the efforts that one would carry out to represent such a plant through a mathematical model; finally, starting with IMMS and building from it several alternative SMMS solutions are created and tested, leading to a final model whose underlying structure is different but derived from IMMS.

When presenting the results, the mechanistic model should have the main focus as it is the main source of interpretability and quality of prediction. To achieve that goal we present the SMMS results as the mechanistic model plus the extension set, or, when presenting all equations together, we underline the extension terms.

3. Non-ideal CSTR application

We assume here that the real plant corresponds to a situation where we do have two reaction zones with different kinetics, heat transfer and a bypass stream (Fig. 3), corresponding to a CSTR under non-ideal conditions. Therefore, to generate simulated data collected from such a plant, we will consider as our perfect mechanistic model the set of Eq. (5). This set of equations is the one that we will be using to generate plant simulated data (D). As the departures from ideal behavior which do occur are not considered to be known, model (5) is only used to generate simulated data measurements and identify benchmark performances.

In this section we consider a continuous stirred tank reactor with first order reaction kinetics, $A \rightarrow B$. The case study scenario uses a model with two reaction zones of different kinetics, heat transfer and a bypass stream (Fig. 3), in order to simulate a CSTR with non-ideal conditions. As the departure from the ideal behavior is considered to be unknown, model (5) is only used to generate system simulated data measurements.

The system independent variables are the inlet concentration A_0 and the inlet temperature T_0, and its dependent variables are the output concentrations of A and B, and temperature T.

$$
\begin{aligned}
&\frac{1}{\tau(A_0 - A_1) - r_1} = 0\\
&\frac{1}{\tau B_1 + r_1} = 0\\
&T_0 - T_1 + k_e r_1 - k_p(T_p - T_1)^2 = 0\\
&r_1 = k_1 A_1^{1/2}\\
&k_1 = a \exp\left(-b\left(\frac{T_r}{T_1 - 1}\right)\right)
\end{aligned}
\tag{5}
$$

$$
\begin{aligned}
&\frac{1}{\tau(A_0 - A_2) - r_2} = 0\\
&\frac{1}{\tau B_2 + r_2} = 0\\
&T_0 - T_2 + k_e r_2 = 0\\
&r_2 = k_2 A_2\\
&k_2 = a \exp\left(-b\left(\frac{T_r}{T_2 - 1}\right)\right)\\
&A = \delta_1 A_1 + \delta_2 A_2 + \delta_3 A_0\\
&B = \delta_1 B_1 + \delta_2 B_2\\
&T = \delta_1 T_1 + \delta_2 T_2 + \delta_3 T_0
\end{aligned}
$$

The assumed PMM for this plant corresponds to considering the first reaction zone to behave according to the kinetic expression $r \propto A^{1/2}$and heat removal to be proportional to T^2, while for the other zone we considered $r \propto A$. The plant equations given by (5) represent our PMM, for a non-ideal reactor, where the heat transfer parameters are $k_p = 0.01\,\mathrm{K}^{-1}$ and $T_p = 300\,\mathrm{K}$, the stream fractions $\delta_1 = 0.35$, $\delta_2 = 0.60$ and $\delta_3 = 0.05$, τ the residence time (100 s), k_e the reaction heat ($k_e = (-\Delta H_r)/(\rho C_p) = 10^5\,\mathrm{K\,l/mol}$) and T_r is the reference temperature (800 K). The kinetic parameters for the Arrhenius expression are $a = 0.01717\,\mathrm{s}^{-1}$ and $b = 12.58$.

The mathematical model (5) was used to build a training dataset from simulations for different values of T_0 and A_0. The dataset comprises 30 measures of output A, B and T from values of $T_0 \in [425, 475]\,\mathrm{K}$ and $A_0 \in [0.925, 0.975]\,\mathrm{mol/l}$. They correspond to the collection of such data from operating records, with 30 sets of (x, y) measurements being adequate for parameter fitting (since the number of parameters, as we will see, ranges from 2 to 8).

Since in practice we will have operating data (D), but no perfect model structures available or known, we now will forget about PMM and consider how one usually tends to address mechanistic modelling of a CSTR, by assuming as our initial mechanistic model structure the one that corresponds to a perfect CSTR (6), where the residence time (τ) the reaction heat (k_e) and the reference temperature (T_r) have the same values as above. We start by evaluating this mechanistic model for representing the information contained in the simulated dataset.

$$
\begin{aligned}
&A_0 - A - \tau r = 0\\
&B + \tau r = 0\\
&T_0 - T + k_e \tau r = 0\\
&r = kA\\
&k = a \exp\left(-b\left(\frac{T_r}{T - 1}\right)\right)
\end{aligned}
\tag{6}
$$

The training total square error (TSE) (7) for the estimation problem is the sum of errors for concentration of A (ε_{Ai}), concentration of B (ε_{Bi}) and temperature (ε_{Ti}), being the error defined by the difference between dataset measurement and model predicted value. The values 0.02, 0.02 and 5.5 are the dataset

Table 2
Initial mechanistic model performance result

	TSE	a	b
Mechanistic model	14.597	0.01717	12.58
Mechanistic model with adjusted parameters	7.9082	0.0161	12.331

variances for concentrations A, B and temperature.

$$\text{TSE} = \sum_i \left(\frac{\varepsilon_{Ai}}{0.02}\right)^2 + \left(\frac{\varepsilon_{Bi}}{0.02}\right)^2 + \left(\frac{\varepsilon_{Ti}}{5.5}\right)^2 \quad (7)$$

The TSE result for the mechanistic model, and for the mechanistic model having kinetic parameters a and b determined by NLP minimization of the multi-response least squares (7), is given in Table 2.

The prediction quality obtained with the mechanistic model is weak (large mean square error), even with adjusted parameters, and therefore we applied to it our strategy to search for semi-mechanistic model structures. Thus, IMMS was reformulated and its hyperstructure created according to our framework (8). This reformulation used two modification rules: the first was the exclusion of extension slots in the mass balances for A and B in order to maintain the mass conservation law; the second was the inclusion of a neutral variable in the kinetic expression $r = kA/v_n$ in order to increase the number of extension slots in that expression.

$$\begin{array}{ll}
A_0 - A - \tau r = 0 & A_0 - A - \tau r = 0 \\
B + \tau r = 0 & B + \tau r = 0 \\
T_0 - T + k_e \tau r = 0 & T_0 - T + k_e \tau r + \alpha_1 = 0 \\
r - w_1 = 0 & r - w_1 + \alpha_5 = 0 \\
w_1 = kA & w_1 = (k + \alpha_2)\left(\dfrac{A + \alpha_3}{1 + \alpha_4}\right) \\
k = aw_2 & k = (a + \alpha_5)(w_2 + \alpha_6) \\
w_2 = \exp(w_3) & w_2 = \exp(w_3 + \alpha_7) \\
w_3 = bw_4 & w_3 = (b + \alpha_8)(w_4 + \alpha_9) \\
w_4 = 1 - w_5 & w_4 = 1 - w_5 + \alpha_{10} \\
w_5 = \dfrac{T_r}{T} & w_5 = \dfrac{T_r + \alpha_{10}}{T + \alpha_{12}}
\end{array} \quad (8)$$

The first extension set we will use corresponds to the set used for extension polynomials (9), where $\mathcal{P}_2(A)$ represents the second degree polynomial $\mathcal{P}_2(A) = p_1 + p_2 A + p_3 A^2$. The search results are presented in Table 3 for an increasing number of extensions over:

$$\Phi_1 = \left\{\mathcal{P}_2(A), \mathcal{P}_2(B), \mathcal{P}_2(T), \mathcal{P}_2(r), \mathcal{P}_2(k), \mathcal{P}_2\left(\frac{T_r}{T}\right), \mathcal{P}_2(kA)\right\} \quad (9)$$

The hyperstructure identification problem was also solved with a set of elementary extensions (10):

Table 3
Semi-mechanistic models with extension polynomials

Model	TSE	Extensions
SM-P1	1.0676	$\alpha_1 = 5.83 - 72.0A + 80.9A^2$
SM-P2	0.4494	$\alpha_1 = -0.02 - 0.13A + 0.27A^2$ $\alpha_3 = -12.4 + 2.95A + 6.50A^2$
SM-P3	0.2744	$\alpha_1 = -17.1 - 1.09A + 6.37A^2 + 30.2B + 3.28B^2$ $\alpha_3 = -0.24 + 0.36A + 0.01A^2$

$$\Phi_2 = \left\{1, A, B, T, r, k, \frac{T_r}{T}, kA, A^2, B^2, T^2, r^2, k^2, \left(\frac{T_r}{T}\right)^2, (kA)^2\right\} \quad (10)$$

The search results for determination of the best semi-mechanistic models with elementary extensions are presented in Table 4.

Both the polynomial and elementary strategies were able to produce semi-mechanistic models with clearly better prediction errors than the original mechanistic model, even when considering the adjustment of its kinetic parameters. The performance criteria shown above correspond to the evaluation obtained over the same training dataset. To better access the value of the semi-mechanistic models, performance was evaluated over two dataset grids, one corresponding to interpolation in the range $T_0 \in [425, 475]$ K and $A_0 \in [0.925, 0.975]$ mol/l, and the other to extrapolation in the range $T_0 \in [400, 425[\cup]475, 500]$ K and $A_0 \in [0.9, 0.925[\cup]0.975, 1.0]$ mol/l (once again, we use here PMM just for the purpose of obtaining simulated operating data, as a replacement for real plant data). The mean square errors (MSE) for both interpolation and extrapolation are shown in Table 5.

In both interpolation and extrapolation the semi-mechanistic models outperform the original mechanistic model with adjusted parameters. It is worth noting that the performance of the mechanistic model with adjusted parameters is an order of magnitude worse than the performance of a semi-mechanistic model with the same number of adjusted parameters. In this case, by adjusting the mechanistic model kinetic parameters in the IMMS structure we ended up getting biased values, in order to try to minimize errors which are actually due to the lack of a right

Table 4
Semi-mechanistic models with elementary extensions

Model	TSE	Extensions
SM-E1	1.403	$\alpha_1 = -1.11 \times 10^{-5} T^2$
SM-E2	0.745	$\alpha_1 = -1.46 \times 10^{-5} T^2$ $\alpha_2 = 1.60 \times 10^{-4} A^2$
SM-E3	0.502	$\alpha_1 = -1.40 \times 10^{-5} T^2$ $\alpha_2 = 0.001A^2 - 3.63 \times 10^{-4}\left(\frac{T_r}{T}\right)^2$
SM-E4	0.023	$\alpha_1 = -1.43 \times 10^{-5} T^2$ $\alpha_2 = 0.0019A^2 - 0.0014\left(\frac{T_r}{T}\right)^2$ $\alpha_6 = 23.653A^2$

Table 5
Testing datasets model results

Model	Interpolation MSE ($\times 10^3$)	Extrapolation MSE ($\times 10^3$)
Mechanistic	642.01	782.34
Mechanistic adjusted parameters	269.92	439.21
SM-P2	14.32	84.34
SM-P3	8.76	45.02
SM-E2	31.77	182.43
SM-E3	38.76	98.78
SM-E4	1.90	36.14

underlying model structure in IMMS. This is a situation very likely to happen when we try to adapt a mechanistic model to experimental data by adjusting some of the model parameters. When the model parameters are not known, one may be trying to overcome lack of model structure through wrong parameter estimations. This example shows that our framework can provide better alternatives and lead to the development of SMMS models with much better prediction performance than IMMS, both in interpolation as well as in extrapolation.

4. Otto–Williams reactor optimization

The Otto–Williams reactor, proposed by Williams & Otto (1960), was widely used in process optimization case studies. The model that we will use in this section is the one operating under the conditions proposed by Roberts (1979).

This plant corresponds to a continuous, stirred, isothermic reactor with the following reactions occurring in the vessel:

$$A + B \rightarrow C$$
$$B + C \rightarrow P + E$$
$$C + P \rightarrow G$$

The corresponding kinetic laws are presented in (11), with X_i being the mass fraction of component i. The kinetic rates are determined by Arrhenius equations with the parameters given in Table 6.

$$\begin{aligned} r_1 &= k_1 X_A X_B \ (\mathrm{s}^{-1}), \quad k_1 = a_1 \mathrm{e}^{-b_1/T} \\ r_2 &= k_2 X_B X_C \ (\mathrm{s}^{-1}), \quad k_2 = a_2 \mathrm{e}^{-b_2/T} \\ r_3 &= k_3 X_C X_P \ (\mathrm{s}^{-1}), \quad k_3 = a_3 \mathrm{e}^{-b_3/T} \end{aligned} \tag{11}$$

The reactor model comprises rate laws (11) and mass balances (12), where F_A and F_B are the mass flow (kg/s) of A and B, F_R the total mass flow and M is the mass content, considered constant and equal to 2104.7 kg. Similarly to what happened in the previous case study, this set of equations is what we will consider as being a perfect mechanistic model, employed to generate simulated operating data but assumed not to be known for any other purposes.

Table 6
Arrhenius law kinetic parameters

	k_1	k_2	k_3
a (s^{-1})	1.6599×10^6	7.2117×10^8	2.6745×10^{12}
b (K)	6666.7	8333.3	11111.0

$$\begin{aligned} &F_A - F_R X_A - r_1 M = 0 \\ &F_B - F_R X_B - (r_1 + r_2) M = 0 \\ &-F_R X_C + (2r_1 - 2r_2 - r_3) M = 0 \\ &-F_R X_E + 2r_2 M = 0 \\ &-F_R X_G + 1.5 r_3 M = 0 \\ &-F_R X_P + (r_2 - 0.5 r_3) M = 0 \end{aligned} \tag{12}$$

The reactor model was used to determine optimal operations values for F_A, F_B and T. The optimization profit function (13) is based on the value of final products (E and P) and on the cost of raw materials:

$$f = 5554.1 F_R X_P + 125.91 F_R X_E - 370.3 F_A - 555.42 F_B \tag{13}$$

Table 7
Optimal reactor operating conditions

	$F_A = 1.8$	$F_A = 1.9$	$F_A = 2.0$	$F_A = 2.1$	$F_A = 2.2$
F_B	4.72	4.95	5.19	5.42	5.64
T	362.7	363.3	363.9	364.4	364.9
f	921.5	943.2	963.6	982.6	1000.5

Based on the optimization function (13), the model has an optimal operation value (F_B, T) for a given value of F_A. Considering some reference values for F_A, optimal operation conditions are given in Table 7 and Fig. 4 displays some cuts of the profit surface.

In this study we will perform the reactor optimization based on approximate and empirical models. This PMM model was used to generate a dataset with 100 simulated records based on random values for $F_A \in [1.8, 1.9]$ kg/s, $F_B \in [4.5, 5.0]$ kg/s and $T \in [355, 373]$ K. This dataset will be used to identify approximate empirical and semi-empirical models (with a number of adjustable parameters varying between 4 and 14), so that a comparison can be obtained for fully mechanistic, fully empirical as well as semi-mechanistic alternatives generated through our model-building framework.

4.1. Empirical model

An empirical polynomial model (14) was developed using second order polynomials. Its parameters were estimated by standard multi-response non-linear regression, with a resulting total sum of squared errors of 8.997.

$$\begin{aligned} X_E &= a_1 + a_2 F_A + a_3 F_A^2 + a_4 F_B + a_5 F_B^2 + a_6 T_n + a_7 T_n^2 \\ X_P &= b_1 + b_2 F_A + b_3 F_A^2 + b_4 F_B + b_5 F_B^2 + b_6 T_n + b_7 T_n^2 \\ T_n &= \frac{T}{350} \\ a &= [\,-3.826 \ 0.044 \ -0.013 \ 0.022 \ -0.002 \ 7.438 \ -3.597\,]^{\mathrm{T}} \\ b &= [\,-8.715 \ 0.330 \ -0.080 \ -0.010 \ -0.002 \ 15.927 \ -7.216\,]^{\mathrm{T}} \end{aligned} \tag{14}$$

4.2. Approximate mechanistic model

An approximate mechanistic model, proposed by Roberts (1979) based on the fact that under normal operation conditions

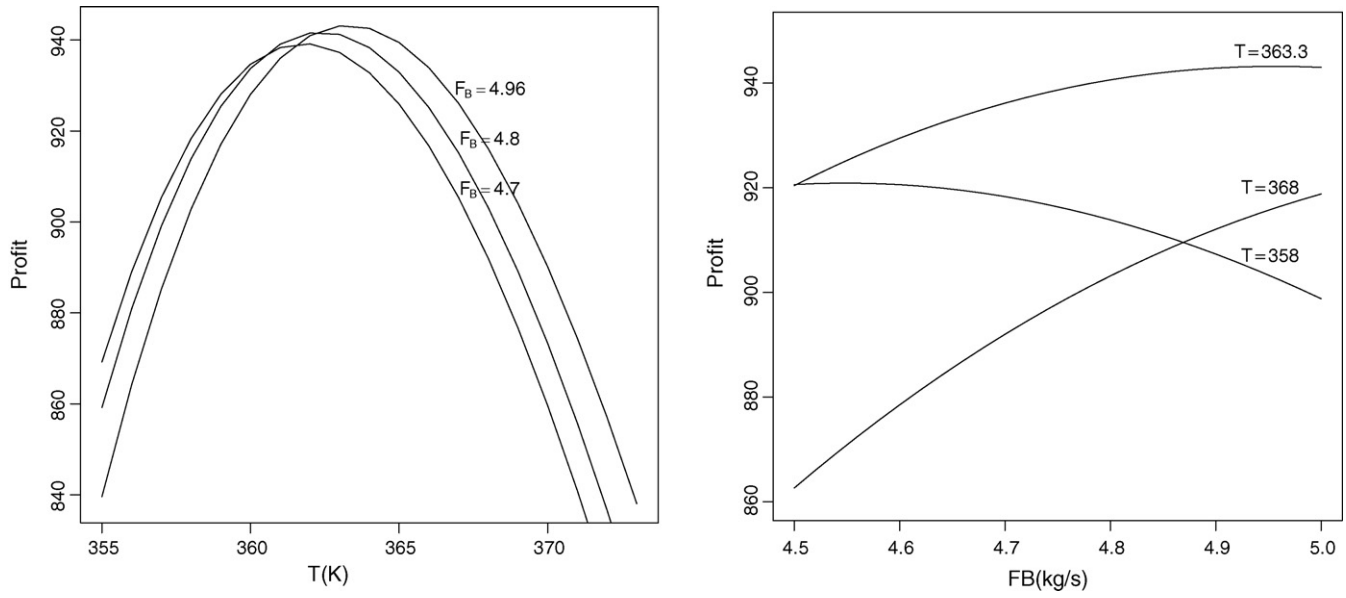

Fig. 4. Profit function versus temperature (left) and Profit function versus B mass flow (right).

very little is produced of C and G, is considered to be our IMMS, i.e., as a first mechanistic attempt to model this kind of plant. This IMMS model (15) has four species (A, B, E and P) and one reaction $A + B \rightarrow E + P$, thus leading to the following set of IMMS equations:

$$\begin{aligned}
&F_R = F_A + F_B \\
&F_A - F_R X_A - rM = 0 \\
&F_B - F_R X_B - 2rM = 0 \\
&-F_R X_E + 2rM = 0 \\
&-F_R X_P + rM = 0 \\
&r = kX_A X_B^2 \\
&k = a\mathrm{e}^{-b/T}
\end{aligned} \tag{15}$$

The kinetic parameters a and b of (15) were estimated from the available set of simulated operating data. The optimal values found for the multi-response (X_E and X_P) regression problem were $a = 0.081\ \mathrm{s}^{-1}$ and $b = 0.3726\ \mathrm{K}$, with a total square error of 3461.6. Thus, one can see that this IMMS model structure is not able to represent the data with good accuracy, as was already reported by Roberts (1979) and Forbes et al. (1994). In fact, this last author shows that this mechanistic model has a structure that is not able to find the optimal reactor conditions and thus is not suitable for usage in model-based plant optimization.

The reactor optimization performed with this model originates optimal values of $F_A = 1.8587\ \mathrm{kg/s}$, $F_B = 3.5\ \mathrm{kg/s}$ and $T = 380\ \mathrm{K}$ (were both F_B and T are in bounds of the variable ranges considered for the problem). These conditions correspond to a very weak value of 171.6 for the merit function.

Given the structural weakness of our IMMS (15), we followed our framework to derive from it a number of more interesting SMMS alternatives, according to searches conducted over the IMMS hyperstructure (16).

$$\begin{aligned}
&F_R = F_A + F_B && F_R - F_A - F_B + \alpha_1 = 0 \\
&F_A - w_1 - rM = 0 && F_A - w_1 - rM + \alpha_2 = 0 \\
&F_B - w_2 - 2rM = 0 && F_B - w_2 - 2rM + \alpha_3 = 0 \\
&-w_3 + 2rM = 0 && -w_3 + 2rM + \alpha_4 = 0 \\
&-w_4 + rM = 0 && -w_4 + rM + \alpha_5 = 0 \\
&r - w_7 = 0 && r - w_7 + \alpha_6 = 0 \\
&k - aw_8 = 0 && k - aw_8 + \alpha_7 = 0 \\
&w_1 = F_R X_A && w_1 = (F_R + \alpha_8)(X_A + \alpha_9) \\
&w_2 = F_R X_B && w_2 = (F_R + \alpha_{10})(X_B + \alpha_{11}) \\
&w_3 = F_R X_E && w_3 = (F_R + \alpha_{12})(X_E + \alpha_{13}) \\
&w_4 = F_R X_P && w_4 = (F_R + \alpha_{14})(X_P + \alpha_{15}) \\
&w_5 = X_B^2 && w_5 = (X_B + \alpha_{16})^2 \\
&w_6 = kX_A && w_6 = (k + \alpha_{17})(X_A + \alpha_{18}) \\
&w_7 = w_6 w_5 && w_7 = (w_6 + \alpha_{19})(w_5 + \alpha_{20}) \\
&w_8 = \exp(w_9) && w_8 = \exp(w_9 + \alpha_{21}) \\
&w_9 = \frac{-b}{T} && w_9 = \frac{-b}{T + \alpha_{22}}
\end{aligned} \tag{16}$$

The extension set Φ was defined to have second order polynomials of the models term set and basic units (17):

$$\begin{aligned}
\Phi = \{&\mathcal{P}_2(F_R), \mathcal{P}_2(F_A), \mathcal{P}_2(F_B), \mathcal{P}_2(X_A), \mathcal{P}_2(X_B), \mathcal{P}_2(X_E), \\
&\mathcal{P}_2(X_P), \mathcal{P}_2(k), \mathcal{P}_2(r), \mathcal{P}_2(F_R X_A), \mathcal{P}_2(F_R X_B), \\
&\mathcal{P}_2(F_R X_E), \mathcal{P}_2(F_R X_P), \mathcal{P}_2(kX_A)\}
\end{aligned} \tag{17}$$

The results obtained for the identification of SMMS structures with two and three extensions are presented in Table 8.

Both semi-mechanistic models identified have TSE values that clearly outperform IMMS scores, as well as those associated with the fully empirical model, showing that indeed they

Table 8
Semi-mechanistic models identified

Model	TSE	Extensions
SM-2	7.705	$\alpha_1 = 0.2470 - 1.9808X_E + 4.3180X_E^2$
		$\alpha_2 = 0.2718 - \frac{0.6965}{T_n} + \frac{0.4556}{T_n^2}, T_n = \frac{T}{350}$
SM-3	0.775	$\alpha_1 = 0.2881 - 2.2173X_E + 4.8305X_E^2$
		$\alpha_2 = 0.0304 - \frac{0.0695}{T_n} + \frac{0.0396}{T_n^2}, T_n = \frac{T}{350}$
		$\alpha_3 = -0.1268 - 8.0263X_A + 30.8223X_A^2$

do have better data explanation and prediction quality. In order to evaluate the usefulness of the previous models for conducting reactor optimization tasks, each one of them was used to search for optimal operating conditions in the variable ranges defined by $F_B \in [3.5, 6.0]$ kg/s and $T \in [320, 380]$ K with F_A set at different fixed values {1.8, 1.85, 1.9, 2.0, 2.1, 2.2} kg/s. These results are presented in Fig. 5, where one can notice that the IMMS model fails quite strongly in the proper identification of optimal conditions. Regarding SMMS alternatives, the SM-2 model presents a big improvement over IMMS, while both SM-3 and the empirical models find solutions close to the true optimal conditions.

Fig. 6 shows in detail the results obtained by the two best models (empirical and model SM-3). There is a significant performance decrease in the empirical model for optimization on F_A values larger than 2.0. As the dataset used in model identification comprises values of F_A in the range [1.8, 1.9] kg/s, the quality of the empirical model can be considered good in interpolation, but bad in extrapolation. On the other hand, model SM-3 maintains good performance characteristics both in the interpolation and extrapolation zone.

These results show once again a large improvement of SMMS solutions over the IMMS mechanistic model, achieved using

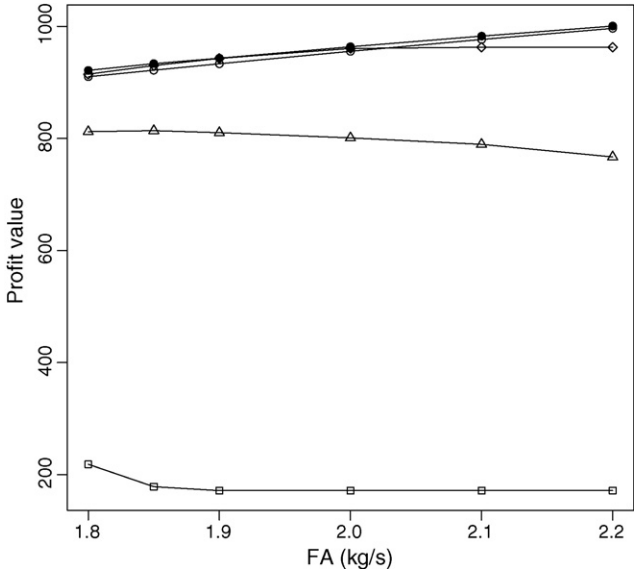

Fig. 5. Optimized operation profits ((•) true, (□) mechanistic, (△) SM-2, (○) SM-3, (◇) empirical).

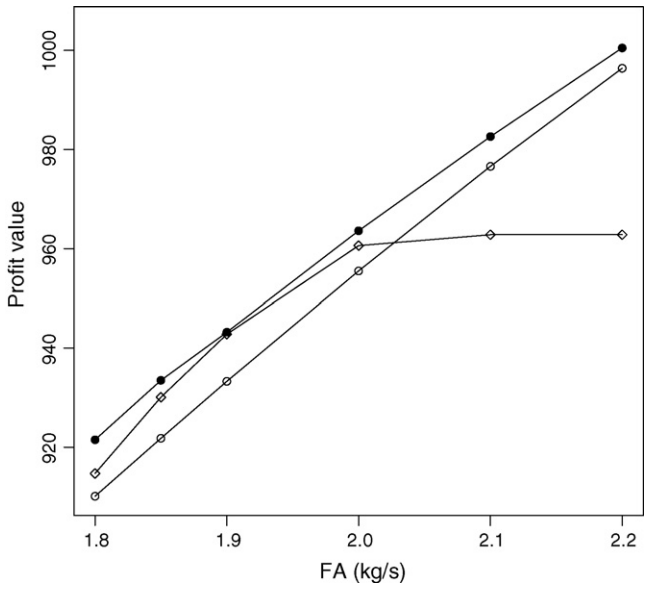

Fig. 6. Optimized profit ((•) true, (◇) empirical, (○) SM-3).

a few localized extensions introduced by our framework. The semi-mechanistic model with three polynomial extensions is able to perform model-based optimization quite well. This semi-mechanistic model was developed based on a mechanistic model that was considered in previous studies as being inadequate for model-based optimization (Forbes et al., 1994).

In fact, previous studies introduced a more elaborate mechanistic model, which may be assumed as an alternative IMMS starting point, that considers two reactions and five species (A, B, E, G and P), in order to overcome some of the data fitting limitations found in (15):

$$A + B \rightarrow E + P$$
$$A + B + P \rightarrow G$$

The new mechanistic model structure is obtained with the following mass balances and kinetic rate laws (18):

$$\begin{aligned} &F_R = F_A + F_B \\ &F_A - F_R X_A - r_1 - r_2 = 0 \\ &F_B - F_R X_B - 2r_1 - r_2 = 0 \\ &-F_R X_E + 2r_1 = 0 \\ &-F_R X_G + 3r_2 = 0 \\ &-F_R X_P + r_1 - r_2 = 0 \\ &r_1 = k_1 X_A X_B^2 M \\ &r_2 = k_2 X_A X_B X_P M \\ &k_1 = a_1 \mathrm{e}^{-b_1/T} \\ &k_2 = a_2 \mathrm{e}^{-b_2/T} \end{aligned} \quad (18)$$

The kinetic parameters for this mechanistic model (ME), determined by regression using our dataset, are presented in Table 9.

This mechanistic model is superior to mechanistic model (15), as can be seen from the large TSE differences between both of these possible IMMS starting points. Using ME as our IMMS, we applied our model building framework in order to

Table 9
Kinetic parameters for mechanistic model ME

a_1	7.887×10^7
a_2	1.390×10^{11}
b_1	7868.25
b_2	10621.28
TSE	3.908

come up with semi-mechanistic alternatives, leading to a final SMMS model:

$$\begin{aligned}
&F_R = F_A + F_B \\
&F_A - F_R(X_A + \underline{p_1 X_B^2}) - r_1 - r_2 = 0 \\
&F_B - F_R X_B - 2r_1 - r_2 = 0 \\
&-F_R X_E + 2r_1 = 0 \\
&-F_R X_G + 3r_2 = 0 \\
&-F_R X_P + r_1 - r_2 = 0 \\
&r_1 = k_1 X_A X_B^2 M \\
&r_2 = k_2 X_A X_B X_P M \\
&k_1 = a_1 \mathrm{e}^{-b_1/T} \\
&k_2 = a_2 \mathrm{e}^{-b_2/T}
\end{aligned} \tag{19}$$

$$a_1 = 19.0416, \quad b_1 = 8177.206, \quad a_2 = 2.9698,$$
$$b_2 = 10824.995, \quad p_1 = 0.1036$$

The mechanistic model ME is able to provide good estimates of the optimal conditions and in extrapolation outperforms the empirical model. The semi-mechanistic model that we determined is able to find the best optimal conditions for all the cases studied. Fig. 7 presents the results obtained for the different models in units of deviation from the true optimal values.

The Otto–Williams reactor is a model used in several optimization case studies. It was used here to show the benefits of model-based optimization, but also to point out the need to use adequate mechanistic or semi-mechanistic models to support optimization efforts. With the semi-mechanistic models obtained in this study we were able to obtain better results than with the original mechanistic models or with empirical strategies This was particularly visible with the simpler mechanistic model, a model that was considered inadequate for model optimization and was extended by our framework into a model with good optimization performance and stability in extrapolation.

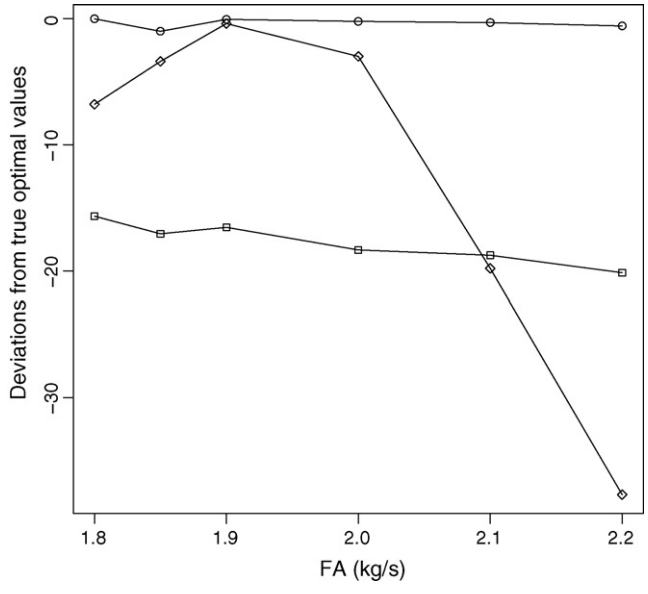

Fig. 7. Deviations from true optimal values ((◇) empirical, (○) SME-1, (□) ME).

This case study also highlights the benefits of our framework symbolic search space. We limit the search space to simple empirical extensions and a set of mechanistic model extension locations. Although the Otto–Williams is a small model, if we were to use symbolic regression based on genetic programming (Koza, 1992), the search space for the structural combinations using the equations parsed trees would require powerful computing resources. On the other hand, the search for the best hybrid Otto–Williams reactor is accomplished with few minutes of a standard desktop computer. And a case study with full Otto–Williams process (Williams & Otto, 1960), having 93 variables and 83 equations, was solved with a couple of hours of desktop computer time.

5. Conclusions

We have introduced and developed a framework that is supported by the symbolic reformulation of a set of first-principles equations, in order to derive hybrid mechanistic–empirical models. The use of this model reformulation strategy results in a set of atomic equations that allow for empirical elements to be added selectively and locally. This set is used to create a symbolic search space on neighborhood of the initial mechanistic model and determine, through combinatorial optimization, the best hybrid mechanistic–empirical that represents the information in the available data.

The development of semi-mechanistic models using this new framework has shown to be a simple and effective path for static mechanistic model evolution. In the case studies presented we were able to obtain models with an order of magnitude improvement in their performance criteria over the initial mechanistic model performance. For the Williams–Otto benchmark problem, we were also able to start with an approximate mechanistic model known to be inappropriate, and obtain a semi-mechanistic model enabling a very good model-based optimization of the unit. In both cases, the model predictions in extrapolation indicate very good performance and stability.

This framework opens new possibilities regarding the integration of empirical and mechanistic modeling. The resulting models have very good characteristics of stability and understandability and can replace mechanistic models in many process engineering model-based techniques.

Acknowledgments

The authors acknowledge financial support provided by research grants FCT POCTI/ 1999/EQU/32647 and PRAXISXXI/BD/15518/97.

References

Bates, D., & Watts, D. (1988). *Nonlinear regression analysis and its applications*. New York: John Wiley & Sons.

Bjrkqvist, J., & Westerlund, T. (2001). Parallel solution of disjunctive MINLP problems. *Chemical Engineering Communications*, *185*, 115–124.

Box, G., & Draper, N. (1987). *Empirical model-building and response surfaces*. New York: Wiley.

Braake, H., van Can, H., & Verbruggen, H. (1998). Semi-mechanistic modeling of chemical processes with neural networks. *Engineering Applications of Artifical Intelligence*, *11*, 507–515.

Cameron, I., & Hangos, K. (2001). *Process modelling and model analysis*. London: Academic Press.

Forbes, J., Marlin, T., & MacGregor, J. (1994). Model adequacy requirements for optimizing plant operations. *Computers and Chemical Engineering*, *18*, 497–518.

Garcia, C., & Morari, M. (1981). Optimal operation of integrated processing systems. *AIChE Journal* , *27*(6), 960–968.

Grossmann, I., & Sargent, R. (1978). Optimum design of heat exchange networks. *Computers and Chemical Engineering*, *2*, 1–7.

Lindskog, P., & Ljung, L. (1995). Tools for semiphysical modelling. *International Journal of Adaptive Control Signal Processing*, *6*(9), 509–523.

Ljung, L. (1999). *System identification-theory for the user* (2nd ed.). Englewood Cliffs, NJ: Prentice-Hall.

Knuth, D. (1973). *The art of computer programming-1. Fundamental Algorithms*. Addison-Wesley.

Koza, J. (1992). *Genetic programming*. MIT Press.

Psichogios, D., & Ungar, L. (1992). A hybrid neural network-first principles approach to process modeling. *AIChE Journal*, *38*(10), 1499.

Quesada, I., & Grossmann, I. (1995). A global optimization algorithm for linear fractional and bilinear programs. *Journal of Global Optimization*, *6*, 39–76.

Roberts, P. D. (1979). An algorithm for steady-state optimization and parameter estimation. *International Journal of Systems Science*, *10*(7).

Saraiva, P. (1996). Inductive and analogical learning: Data-driven improvement of process operations. In G. Stephanopoulos, & C. Han (Eds.), *Intelligent systems in process engineering* (p. 377). Academic Press.

Schubert, J., Simutis, R., Dors, M., Havlík, I., & Lubbert, A. (1994). Hybrid modeling of yeast production processes—A combination of a priori knowledge on different levels of sophisticaticon. *Chemical Engineering and Technology*, *17*, 10.

Smith, E., & Pantelides, C. (1999). A symbolic reformulation/spatial branch and bound algorithm for the global optimization of nonconvex MINLPs. *Computers and Chemical Engineering*, *23*, 457–478.

Su, H. -T., Bhat, N., & McAvoy, T. J. (1992). Integrating neural networks with first principles models for dynamic modeling. *In Preprints IFAC DYCORD+'92*, College Park, Maryland.

Thompson, M., & Kramer, M. (1994). Modeling chemical processes using prior knowledge and neural networks. *AIChE Journal*, *40*, 1328.

Turkay, M., & Grossmann, I. E. (1996). Logic-based MINLP algorithms for the optimal synthesis of process networks. *Computers and Chemical Engineering*, *20*, 959–978.

Van Can, H., Hellinga, C., Luyben, K., & Heijnen, J. (1996). Strategy for dynamic process modeling based on neural networks and macroscopic balances. *AIChE Journal*, *42*(12), 3403.

Williams, T. J., & Otto, R. E. (1960). A generalized chemical processing model for the investigation of computer control. *AIEE Transactions*, *79*, 458.