5
Model Tuning, Discrimination, and Verification

Katalin M. Hangos and Rozália Lakner

5.1
Introduction

Process models are increasing in size and complexity in current computer-aided process engineering. Therefore the methods and tools for their tuning, discrimination and verification are of great importance. The widespread use of process models for design, simulation and optimization requires the proper documentation, reuse, and retrofit of already existing models that need the above techniques. This chapter deals with computer-aided approaches and methods of model tuning, discrimination and verification that are based on a formal structured description of process models.

Basic assumptions. For the majority of process control and diagnostic applications, lumped dynamic process models are used. This model class, which is considered throughout this chapter, is obtained under the following *basic modeling assumptions*:

- Only lumped models are considered (ordinary differential and algebraic equation models).
- Only initial value problems are considered.
- All physical properties in each phase are assumed to be functions of the thermodynamic state variables (temperature T, pressure P, compositions C_k) only.

5.2
The Components and Structure of Process Models

The formal description of process models and their structure is the basis of any methods in computer-aided process systems engineering. If one considers process models as structured knowledge collection with underlying syntax and semantics then the formal methods of computer science can be applied for model discrimination and verification. The fundamentals of such an approach are briefly described in this section.

Computer Aided Process and Product Engineering. Edited by Luis Puigjaner and Georges Heyen

ISBN: 3-527-30804-0

5.2.1 The Modeling Problem and the Modeling Goal

A process model is jointly determined by the process system it describes and by its modeling goal (Hangos and Cameron 2001). The specification of a process system includes the definition of the system boundaries and the way of interactions between the system and its environment together with the description of the internal structure (subsystems, mechanisms, etc.) of the system itself. The effect of the modeling goal is much less investigated despite its importance for constructing a process model.

The modeling goal. Any process model is developed for a specific use or possibly multiple uses. These uses influence the goals that the model must fulfill. For example, the application areas of process design, control, optimization or diagnosis usually lead to different model representations for the same physical system. Meeting the stated modeling goal provides a means of determining when the modeling cycle (see below) should terminate.

A set of process models is *functionally equivalent* with respect to a modeling goal if every model of the set fulfils the inequalities in the modeling goal.

The seven-step modeling procedure. Good modeling practice requires a *systematic way of developing the model equations of a process system for a given purpose.* Although this procedure is usually cyclic, in which one often returns back to an already completed step, the systematic procedure can be regarded as a sequence of modeling steps (Hangos and Cameron 2001) that include:

1. problem setup for process modeling;
2. selection of important mechanisms;
3. analysis of data;
4. construction of model equations;
5. model verification;
6. model solution;
7. model calibration and validation.

Model tuning, discrimination and verification techniques are applied in the last four steps of this procedure.

5.2.2 The Model Equation Constructing Subprocedure and its Steps

The construction of model equations is the fourth step in the above procedure, which is a cyclic procedure in itself with the following steps:

1. system and subsystem boundary and balance volume definitions;
2. establish the balance equations;

3. transfer and reaction rate specifications;
4. property relation specifications;
5. balance volume relation specifications;
6. equipment and control constraint specifications;
7. selection of design variables.

Incremental building of balance equations. The steps of the model building procedure should be carried out in a sequential-iterative manner. This means that the model equations are built up incrementally, repeating steps of the model equation constructing subprocedure in the following order of conserved extensive quantities:

- *Overall mass submodel.* The terms and variables in the conservation balances for the overall mass in each balance volume appear in all other conservation balances. Therefore this subset of model equations is built up first.
- *Component mass submodel.* With the given conservation balances for the overall mass in each balance volume, it is easy to set up the conservation balances for component masses. This subset of model equations is added to the equations originated from the overall mass balances.
- *Energy submodel.* Finally the subset of model equations induced by the energy balances is added to the equations.

This way the kernel of the model equation constructing the subprocedure is repeated several times for every balance volume.

5.2.3
Model Equations, Initial and Boundary Conditions, and Model Parameters

The conservation balances of mass, component masses and energy are described by ordinary differential equations in a lumped process system model (Hangos and Cameron 2001). These are called conservation *balance equations*, and they are accompanied by suitable algebraic *constitutive equations*. Constitutive equations describe the underlying static relationships between model variables dictated by physics and chemistry. The process model is then a set of ordinary differential and algebraic equations (DAEs) where there are underlying semantic relationships between various variables, equations, and equation terms.

In addition to the equations themselves, it is required to specify the *initial conditions* of the ordinary DAE system in order to solve the problem. Initial conditions set the values of the differential variables at the initial time (t=0).

Note that in the case of distributed parameter systems, partial differential equations (PDEs) are used for describing the conservation balance equations of the model. In these models *boundary conditions* specifying the values of the differential variables for all time on each of the system boundaries and the specification of *initial conditions* for the whole spatial region of interest are also part of the process model.

5.2.4
Hidden Components

Besides the model elements above, a systematically constructed process model contains elements that are usually not stated in an explicit way, but are important for model discrimination and verification. These are as follows:

- *Application domain* determines the validity region of the model.
- *Inequality constrains* constrain the value of a parameter or variable often dictated by the underlying physics and chemistry (e.g., temperature should be positive).
- *Modeling assumptions* describe the decisions of the modeler in an explicit formal way.

The importance of the modeling assumptions is explained by the fact, that model building itself can be seen as a sequence of specifying, simplifying, or enlarging assumptions on the process system to be modeled (Hangos and Cameron 2001a). This way, a uniform assumption-driven approach can be developed where modeling assumptions are regarded as artifacts of the modeling steps and allow the rigorous formal description of the modeling process and its result.

5.3
Model Discrimination: Model Comparison and Model Transformations

Model discrimination is based upon systematically comparing different process models to find relationships between them. For this purpose we briefly review various model description forms and their transformations that form the basis of model discrimination.

5.3.1
Formal Representation of Process Models and their Transformations

Model elements. The differential-algebraic equation set that forms a lumped process model can be seen as a hierarchically structured knowledge collection constructed from the following *main model elements*:

- The *balance volumes* are the basic elements in process modeling as they determine the regions in which the conserved quantities are contained.
- The conserved *extensive quantities* (differential variables) are the additive properties of a system and they are used for describing the conservation principles (such as mass, component masses and energy conservation) in the balance volumes.
- The *balance equations* reflect the conservation principles for each extensive conserved quantity.

- The *transport mechanisms* such as convection, transfer, reaction, etc., correspond to an effect on the conserved extensive quantities so they appear as additive terms in the balance equations.
- The *constitutive equations* are algebraic relations that complete the model equations. They describe property relations, extensive-intensive relationships, transfer and reaction rate relations, equipment and control relations, and balance volume relations.
- The *algebraic variables* are the nondifferential variables appearing in balance equations and constitutive equations in the form of thermodynamic state variables, transfer and reaction rate variables, equipment and control variables, constants, specification variables, etc.

Any process model can also be seen as a collection of *mathematical elements*, like variables and equations of the following type:

- *differential equations* that originate from the conservation balance equations;
- *algebraic equations* describing the constitutive equations, the transport mechanisms, etc., that are evoked by the conservation balance equations;
- *differential variables* with their first time derivative present in the differential equations;
- *algebraic variables* including constants and specified (design) variables.

Finally, there can be other *auxiliary elements*, such as surfaces, for constructing a process model.

Hierarchy of model elements. Driven by the role in the process model these model elements can be organized into the following natural *hierarchy levels*:

- L1: balance volume level
- L2: balance equation level
- L3: transport mechanism level
- L4: constitutive level

A simple process model of a jacketed tank reactor with all of the above-mentioned model elements is shown in Fig. 5.1.

Modeling assumptions. A *modeling assumption* can be expressed in a natural language sentence and formally described by a triplet (Hangos and Cameron 2001a) given by:

variable-name **relation** *keyword*,

where 'variable-name' refers to a process model element described previously in this subsection, **'relation'** is an "=" ("equals") or "is" symbol in most cases, and *'keyword'* is a symbolical or numerical constant or another 'variable-name'. Thus, a modeling assumption is understood as an *assignment* to the 'variable-name' and usually translated into either additional mathematical relationships between model variables and parameters, or into constraints between already introduced variables and parame-

CONSERVATION BALANCES

Balance volume: tank

- **mass balance:** $M = const$
- **energy balance:** $\frac{dU}{dt} = v\rho c_p T_0 - v\rho c_p T + Vr(-\Delta H) - Q$
- **component mass balances:** $\frac{dm_A}{dt} = vc_{A0} - vc_A - Vr$

$$\frac{dm_B}{dt} = -vc_B + Vr$$

Balance volume: cooler

- **mass balance:** $M_c = const$
- **energy balance:** $\frac{dU_c}{dt} = v_c\rho_c c_{pc} T_{c0} - v_c\rho_c c_{pc} T_c + Q$

CONSTITUTIVE EQUATIONS

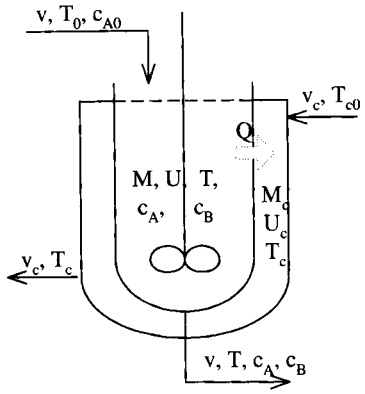
$Q = KA(T - T_c)$ $\quad U = Mc_pT$

$r = kc_A$ $\quad U_c = M_c c_{cp} T_c$

$k = k_0 e^{-\frac{E}{RT}}$ $\quad M = V\rho$

$m_A = Vc_A$ $\quad M_c = V_c\rho_c$

$m_B = Vc_B$



ASSUMPTIONS

- 2 lumped balance volumes (tank, cooler)
- 3 components in tank (A, B, solver)
- 1 component in cooler
- constant mass holdups
- constant physico-chemical properties
- *A -> B* first order exothermic reaction in tank

Figure 5.1 A simple process model example.

ters. The modeling assumptions can either be *elementary assumptions* consisting of just a single triplet or *composite assumptions* being the conjunction (logical and) of elementary assumptions.

The model equations can be formally seen as a structured string obeying syntactical and semantic rules and the modeling assumptions can then be regarded as *formal modeling transformations* on these equations resulting in another set of model equations. The effect of an assumption on a given set of equations is computed following all of the *implications of the assumption* through the syntactical and semantic rules. Formally this is performed by substituting the assignment equations describing the assumption into all of the original model equations and then performing rearrangements using algebraic transformations.

5.3.2
Algebraic Transformations, Algebraically Equivalent Models

A set of functionally equivalent process models can be *algebraically equivalent*, when one can transform any member of the set to any other one using algebraic transformations. Algebraic transformations can be applied to model equations and to model variables (including both differential and algebraic ones).

Examples of algebraic transformations on a set of process model equations are multiplying an equation by a constant number, adding two equations together, substituting one equation for another by expressing it as a variable and substituting that variable in every other equation. It is important to note that the variables do not change when applying algebraic transformations to the equations of a model, but the

CONSERVATION BALANCES

Balance volume: tank

- **mass balance:** $M = const$
- **energy balance:**

$$\frac{dT}{dt} = \frac{v}{V}(T_0 - T) + \frac{k_0 e^{-\frac{E}{RT}} c_A(-\Delta H)}{\rho c_p} - \frac{KA(T - T_c)}{V\rho c_p}$$

- **component mass balances:**

$$\frac{dc_A}{dt} = \frac{v}{V}(c_{A0} - c_A) - k_0 e^{-\frac{E}{RT}} c_A$$

$$\frac{dc_B}{dt} = -\frac{v}{V} c_B + k_0 e^{-\frac{E}{RT}} c_A$$

Balance volume: cooler

- **mass balance:** $M_c = const$
- **energy balance:**

$$\frac{dT_c}{dt} = \frac{v_c}{V_c}(T_{c0} - T_c) + \frac{KA(T - T_c)}{V_c \rho_c c_{pc}}$$

Figure 5.2 The substituted process model.

model equations do. The formal description of algebraic transformations to a set of algebraic equations, together with a canonical set of primitive algebraic transformations and their effect on computational properties, can be found in Leitold and Hangos (1998). There it is shown that certain computational properties of DAE models, such as the differential index, does not change with algebraic transformations, but others, like the decomposition of the model may change quite drastically.

Another type of transformation applicable to a model is when one applies a linear or nonlinear algebraic transformation to some of its variables and writes the model using these new transformed variables. This is analogous to coordinate transformation in geometry and is therefore called a *coordinate transformation.* It is important to note, however, that the general locally invertible nonlinear transformations, which are useful and widely used in nonlinear system theory, are not well accepted in process systems engineering because they change the engineering meaning of the variables. The extensive-intensive constitutive algebraic equations, however, are widely used to transform a process model into its intensive variable form suitable for process control and diagnostic applications. Here we transform the set of differential variables in a balance volume originally equal to the set of conserved extensive quantities to be the set of overall masses (left unchanged) and the measurable intensive quantities (such as temperatures, compositions and pressures).

Model classes. Algebraic transformations may change the mathematical form of a model, but an algebraically transformed model is the same from a process engineering point of view. Such algebraically equivalent models form a *model class.* Figure 5.2 shows an algebraically equivalent form of the simple process model of the jacketed tank reactor depicted in Fig. 5.1, where the constitutive equations have all been substituted into the differential ones.

5.3.3
Model Simplification and Model Building Transformations

Modeling assumptions can be regarded as representations of the engineering activity and decisions during the whole modeling process in constructing, simplifying and

analyzing process models and they act as modeling transformations on the process models. Assumption-driven modeling works directly with modeling assumptions, thus enabling the definition and handling of process models as structured knowledge with defined syntax and semantics.

Model building assumptions. The modeling assumptions applied in the *model building* phase determine the structure of the process model and the assumptions applied to an existing model modify the equations and may even change the structure of the model. The model building procedure is seen as a sequence of model building, specification assumptions, and their associated transformations, as well as algebraic transformations applied to a process model. This way of assumption-driven model building offers a systematic incremental way of constructing a process model in its canonical form.

Model simplification assumptions. The model simplification assumptions, which can either be elementary (atomic) or composite, and are composed of a conjunction of elementary assumptions, can be formally described as model transformations. These transformations are projections in a mathematical sense and are often performed in two substeps:

1. adding the equality describing the assumption to the already existing set of model equations and performing algebraic transformations (for example substitutions) to get a more simple form;
2. adjusting the set of differential, algebraic and design variables to satisfy the degree of freedom requirement.

The effect of a simplification assumption on a given set of equations is computed following all of the implications of the assumption through syntactical and semantic rules. Formally this is performed by substituting the assignment equations describing the assumptions into all of the original model equations and then performing rearrangements using algebraic transformations.

It is important to note that not every simplification transformation is applicable to a particular process model. Moreover, a transformation may influence only part of a process model and then this effect propagates through the relationships between the model elements. Forward reasoning can be applied to find all of the implications of a simplification transformation, and the effect of a composite transformation is computed by generating a sequence of simplified process models. It is important to note, however, that the resultant model may be different if the order of the assumptions is changed, because model simplification transformations may be related and noncommutative (Hangos and Cameron 2001a).

In conclusion we can say that algebraic manipulations can be regarded as equivalence transformations, model simplification, and enrichment assumptions as general nonequivalence modeling transformations acting on process models that bring the process model out from its original model class.

5.3.4
Model Discrimination and Model Comparison

Model discrimination aims to find an exact relationship between two given process models of the same process system. First, we have to determine if these models are developed for the same modeling goal, and if so, if they are algebraically equivalent.

Unfortunately, there are no standard formal ways of performing the above two basic tasks, mainly because the lack of our knowledge in formal description and utilization of the modeling goal itself (see Lakner, Cameron and Hangos 2003).

In the case of functionally equivalent models one should only perform model comparison to investigate if they are algebraically equivalent, and if not, give their relationship in terms of model simplification transformations that lead the more detailed model to a more simple one (see later Section 5.4.3).

Canonical form. We have already seen in Section 5.3.1 that both functionally and algebraically equivalent process models form a model class. It is then useful to define and use a *canonical form* of a process model class, which is a member of the class having each model element in the form that has a clear engineering meaning (Hangos and Cameron 2001). The differential equations of a process model in canonical form are the conservation balance equations for the overall mass, total energy (or enthalpy) and all but one component masses in their extensive form containing terms for the convective, transfer and source transport. These are supplemented by constitutive algebraic equations of standard categories such as intensive-extensive relationships, reaction rate equations, thermodynamic state equations (such as the ideal gas law), etc.

The simple process model shown in the left-hand side of Fig. 5.1 is in its canonical form.

Comparison of process models. In the case of algebraically different but functionally equivalent process models we aim at finding out if these models are from the same model class, or not. The general procedure of comparing these models is to first bring them into their canonical form and then compare them element-wise following the hierarchy of the model elements from top (balance volumes) to down (model variables and parameters). This approach requires that the models being compared are given with their entire model elements hierarchically arranged.

5.4
Model Tuning

Process models generated for a given modeling goal may often be over-simplified or over-complicated for another use, which is why there is usually a need to extend, to simplify, or to generate a new model in the worst case. In addition, process models usually contain unknown parameters to be estimated using measured data, when we need to calibrate the model to tune it to meet the modeling goal.

5.4.1 Model Simplification

In the model constructing step of the seven-step modeling procedure we may need to perform a model simplification phase for refinement of an already defined process model by additional simplifying modeling assumptions. These simplifying assumptions can be described by triplets and are usually translated into additional mathematical relationships. The simplifying procedure itself consists of two main steps. These steps are the implication of the modeling assumptions on the model equations with the aid of syntactical and semantic rules, and the rearrangement of the resulting equations make use of formal algebraic transformations. Because the process model elements are related to each other by a well-defined syntax and semantic, a model element cannot be simplified independently of the others.

In addition, the implications of a simplification assumption depend both on the assumption and the structure of the model. For example, when a modeling assumption is related to a balance volume, its implications can refer not only to the equations of the balance volume, but could modify other related balance equations in other balance volumes. The same way, a modeling assumption related to a term in a balance equation can imply modifications of other terms in other balance equations. Figure 5.3 shows how a modeling assumption on the mass convective term of a balance volume changes the energy and component mass convective terms when a simplification assumption is applied to the model of the jacketed tank reactor depicted in Fig. 5.1.

The implications of an assumption on the model equations can be determined by forward reasoning where all of the implications of the assumption are computed by respecting syntactical and semantic rules. The set of resulting modeling equations at the end of the implication stage is rearranged to an easily-solvable form by using algebraic transformations (Lakner et al. 1999).

CONSERVATION BALANCES

Balance volume: tank

- **mass balance:** $M = v\rho$
- **energy balance:** $\frac{dU}{dt} = v\rho c_p T_0 + Vr(-\Delta H) - Q$
- **component mass balances:** $\frac{dm_A}{dt} = vc_{A0} - Vr$

$$\frac{dm_B}{dt} = Vr$$

Balance volume: cooler

- **mass balance:** $M_c = const$
- **energy balance:** $\frac{dU_c}{dt} = v_c\rho_c c_{pc} T_{c0} - v_c\rho_c c_{pc} T_c + Q$

SIMPLIFICATION ASSUMPTION
- mass convective outlet in tank is negligible

CONSTITUTIVE EQUATIONS

$Q = KA(T - T_c)$ $\quad U = Mc_pT$

$r = kc_A$ $\quad U_c = M_c c_{cp} T_c$

$k = k_0 e^{-\frac{E}{RT}}$ $\quad M = V\rho$

$m_A = Vc_A$ $\quad M_c = V_c\rho_c$

$m_B = Vc_B$

Figure 5.3 A simplified process model.

5.4.2 Model Extension

Model extension procedures are widely applied in process modeling in quite different contexts:

- At the end of a modeling cycle, in the model validation step, it may turn out that the developed model fails to fulfill the modeling goal. Then one has to extend the model by including additional model elements that were originally neglected, such as balance volumes, balances, mechanisms, etc.
- The incremental assumption-driven model building (Williams et al. 2002) uses model extension procedures, too.
- The incremental building of balance equations in the model equation constructing procedure can also be seen as model extension (see Section 5.2.2).

There are two questions of critical importance in model extension procedures: the selection of default values and the methods of ensuring incremental consistence. *Default values* set the value of model elements belonging to a model element (the "children elements" in the model element hierarchy) that is just being created. *Incremental consistency* is ensured by allowing one to add only such new model elements to an already existing consistent model that are not in conflict or contradiction to any already existing model element. Details about these questions can be found in the literature on computer-aided process modeling (CAPM) tools (see, e.g., Jensen-Krogh 1998; Modkit 2000).

Empirical model building. This is a special method of model extension using empirical data and grey box models (Hangos and Cameron 2001). It is a top-down approach of model extension where the model element(s) to be changed or extended is (are) determined in a heuristic black box way by using sensitivity analysis. The submodel of the new element is also constructed in a heuristic black box way from some general approximating model class with its parameters estimated using measured data (see also model calibration in Section 5.4.4).

5.4.3 Model Comparison by Assumption Retrieval

In order to avoid any inconsistency during model simplification and extension, it is extremely useful to register explicitly all modeling assumptions applied in the construction and modification of the process model. The documentation of the model contains these modeling assumptions (Hangos and Cameron 2001a) in the ideal case, but this documentation can often be incomplete or even missing. In order to complete the model documentation with all modeling assumptions, an assumption retrieval procedure could be used.

The retrieval of modeling assumptions from a pair of process models for model analysis and comparison is an important but unusual problem where not only effi-

cient algorithms but the engineering understanding is lacking. The reason for this is that assumption retrieval can be regarded as the inverse task of model transformations where modeling assumptions are determined from two related (one detailed and one simplified) process models of the same process system. As model transformations are projections in mathematical sense, it is not possible in general to retrieve fully the original model from the transformed one and from the transformations. Because of this, the result of assumption retrieval from the original and the transformed models may not be, and in general will not be, unique. Because of the nonuniqueness of the assumption retrieval task, an intelligent exhaustive search algorithm (Lakner et al. 2002) is needed for its solution.

5.4.4 Model Calibration (Model Parameter Estimation)

Process models developed from first engineering principles almost always contain model elements, model parameters and/or other elements like reaction rate expressions, the value of which is unknown. While the modeling approach fixes the structure of the model, these unknown elements make the model "grey," that is, partially unknown. Measured data from the real process system to be modeled is used along with model parameter and/or structure estimation methods to fine-tune the model for meeting the modeling goal. This fine-tuning of process models is called *model calibration* and is a standard step in the seven-step modeling procedure (Hangos and Cameron 2001).

There are several key points to take special care of when performing model calibration:

- *Selection of model parameters to be estimated.* Besides the real unknown model parameters, one often has parameters or model elements with large uncertainty associated to their values. If the model is sensitive with respect to the value of these uncertain parameters, then it is advisable to consider them as unknown and estimate their values using measured data (Németh et al. 2003).
- *Nonlinear parameter estimation.* In most of the cases the parameters to be estimated enter the model in a nonlinear way and the model itself is dynamic. This makes the parameter estimation problem especially difficult and can only be solved by numerical optimization techniques (Hangos and Cameron 2001; Ailer et. al 2002).
- *Quality of the data and the estimated parameters.* The statistical nature of model parameter estimation requires one to check carefully the following key ingredients and properties of the parameter estimation method:
  - quality of the measured data (steady-state, no outliers and gross errors, etc.) and the presence of sufficient excitation;
  - quality of the prediction error sequence (if this is a realization of a white noise stochastic process);
  - quality of the estimated parameters considering their nonbiasness, variance, and covariance matrix.

5.5
Model Verification

Having completed the model equation constructing step of the seven-step modeling procedure (see Section 5.2.2), one needs to perform model verification, that is, to check the model against engineering insight and expectations, before attempting its solution. Model verification includes checks of syntax and semantics, as well as the well-posedness of the model from mathematical sense, and analysis of computational and dynamic properties.

5.5.1
Formal Methods for Checking Syntax and Semantics

Before a mathematical model is used for solvability analysis or is solved, it is useful to check and ensure its consistency. There are several methods for consistency checking that are applicable both in computer-aided modeling tools and in process systems engineering practices:

- *Dimension analysis* is a useful simple check for consistency of the model equations in terms of units of measure. This very useful but not widespread method is used in ASCEND (Evans et. al 1979) and VeDa (Bogusch and Marquardt 1997) modeling languages, for example.
- *Syntactical verification methods* (checking bracketing, vector operations, etc.) are especially important for computer-aided modeling tools in which the model equations can be directly defined by the users. An example for this is the ICAS/MoDev modeling system (Jensen-Krogh 1998).
- *Logical checking* (hierarchical consistency, material characterization consistency, chemical reaction rate equation derivation, etc.) is used for examining the modeling assumptions' consistency. This very important verification method, accomplished before generating the model equations, is used in the majority of computer-aided modeling tools.

It is important to note that the above-introduced model verification methods are applicable only for *partial consistency* checking and they cannot insure full model consistency.

5.5.2
Structural Analysis of Computational Properties of Process Models

The structural analysis of dynamic lumped process models form an important step in the seven-step modeling procedure and is used for the determination of the solvability and computational properties of the model. This analysis includes the determination of the degrees of freedom (DOF), the differential index and the structural components of the model.

Analysis of DOF and differential index. In order to solve a mathematical model, a sufficient number of variables have to be specified so that the number of unknown variables exactly equals the number of equations. The DOF, i.e., the difference between the number of unknown variables and the number of equations in the mathematical representation, is equal to the number of variables that must be specified for obtaining a solvable equation system. There are three possible values for DOF to obtain (Hangos and Cameron 2001):

- DOF=0. This implies that the number of independent unknowns and independent equations is the same and a unique solution may exist.
- DOF>0. This implies that the number of independent variables is greater than the number of independent equations and the problem is underspecified. In this case some of the independent variables have to be specified by some external considerations in order for the DOF to be reduced to zero.
- DOF<0. This implies that the number of variables is less than the number of equations and the problem is over-specified. If this occurs it is necessary to check and make sure that you have included all relevant equations.

It is important that the DOF analysis can be applied both on the entire equation system and on the subsets (mass-related equations, energy-related equations, etc.) of model equations separately.

The *differential index* of a DAE is defined as the minimum number of differentiations with respect to time that the algebraic system of equations has to undergo to convert the system into a set of ordinary differential equations (ODE) (Hangos and Cameron 2001). The index of a pure ODE system is zero by definition. When the Jacobian of the algebraic equation set of DAE is of full rank, then the index of DAE is one. In this case the initial values of the differential variables can be selected arbitrarily, and the DAE can easily be solved by conventional methods such as Runge-Kutta or Backward Differentiation methods. If, however, the index is higher than 1, special care should be taken in assigning the initial values of the variables, since some "hidden" constraints lie behind the problem specifications.

Structural decompositions. Effective graph-theoretical methods have been proposed in the literature based on the analysis tools developed by Murota et al. (1987) for the determination of the most important solvability property of lumped dynamic process models (Leitold and Hangos 2001): the differential index and the structural components. The analysis is based on constructing the structural representation graph of the DAE model equations where the variables are represented as vertices and the equations as edges (dependencies) between vertices. Labels are associated with the vertices of the graph indicating the computational property of the associated variable. The reduced representation graph, together with the L- and M-components and their hierarchy, are determined by the analysis, which can effectively be used to select a suitable numerical solution method and to determine the computational path. In addition, one can artificially structure a DAE model by using algebraic transformations to be able to solve it more efficiently (Robertson and Cameron 1997).

5.5.3
Analysis of Structural Dynamical Properties (Controllability, Observability, Stability)

One simple yet powerful method of model verification is to analyze the structural dynamical properties of the developed model and compare the result with engineering expectations. The first step of the analysis is to transform the lumped process model in its DAE form into a nonlinear state-space model form, which is only possible for index 1 models.

Structure graph. Then the so-called *structure graph*, a weighted (signed) directed graph (SDG) of the model is constructed, which contains the state, input and output variables as vertices, and the model equations determine its directed edges in such a way that a directed edge points towards variable v_i from v_j when v_j appears on the right-hand side of the equation that determines v_i. The weight of the edges contains the sign of the effect (the *sign* of the partial derivative) the edge is associated with.

Note that an SDG model corresponds to a *class* of process model with the same structure. We say that a structural dynamical property, such as structural controllability, observability, or stability, holds for a class of process models if almost every member of the class (with the exception of null-measure sets) possesses the property.

Check of structural properties. Given an SDG model, there are simple-to-check combinatorial conditions for the underlying process model class to be structurally controllable or observable (Hangos and Cameron 2001; Hangos et al. 2001). For example, a process model is structurally controllable (observable) if its state structure matrix is of full structural rank and its SDG graph is input (output) connectable, that

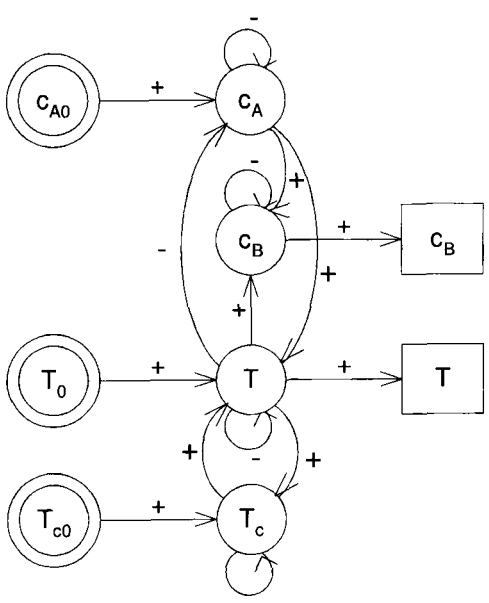


Figure 5.4 The SDG of the simple process example in Fig. 5.2 (input: c_{A0}, T_0, T_{c0}; output: c_B, T).

is, there exists at least one directed path to every state (output) variable vertex from an input (state) variable vertex.

The check of structural stability is more computationally demanding and requires finer qualitative information on the value of the model parameters. A simple, but not thorough enough, general method of checking structural stability of a process model class is the method of *conservation matrices* (Hangos and Cameron 2001). Here we use the state matrix of a locally linearized process model and check if it is a conservation matrix or not. A real square matrix is a conservation matrix if its diagonal elements are negative, all other elements are nonnegative and the diagonal elements are row (or column) dominants, i.e., the absolute value of the diagonal element is greater that the sum off the off-diagonal elements in every row (or column).

Acknowledgments

This research was partially supported by the Hungarian National Research Fund through contract numbers T042710 and T047198 and is gratefully acknowledged.

References

1 *Ailer, P., Szederkényi, G., Hangos, K. M.* (2002) Parameter-estimation and model validation of a low-power gas turbine. *Proceedings of the IASTED International Conference, Modelling, identification and control*, Innsbruck, Austria, pp. 604–609.

2 *Bogusch, R., Marquardt, W.* (1997) A formal representation of process model equations. *Computers and Chemical Engineering* 21, 1105–1115.

3 *Evans, L. B., Boston, J. F., Britt, H. I., Gailler, P. W., Gupta, P. K., Joseph, B., Mahalec, V., Seider, W. D., Yagi, H.* (1979) ASPEN: An advanced system for process engineering. *Computers and Chemical Engineering* 3, 319–327.

4 *Hangos, K. M., Cameron, I. T.* (2001) *Process Modelling and Model Analysis.* Academic Press, London, pp. 1–543.

5 *Hangos, K. M., Cameron, I. T.* (2001a) A formal representation of assumptions in process modelling. *Computers and Chemical Engineering* 25, 237–255.

6 *Hangos, K. M., Lakner, R., Gerzson, M.* (2001) *Intelligent Control Systems: An Introduction with Examples.* Kluwer Academic Publishers, pp. 1–301.

7 *Jensen-Krogh, J. A.* (1998) *Generation of Problem Specific Simulation Models Within an Integrated Computer-Aided System.* PhD thesis, Danish Technical University.

8 *Lakner, R., Cameron, I., Hangos, K. M.* (1999) An assumption-driven case-specific model editor. *Computers and Chemical Engineering* 23, 695–698.

9 *Lakner, R., Hangos, K. M., Cameron, I. T.* (2002) Assumption retrieval from process models. *Computer-Aided Chemical Engineering* 9, 195–200.

10 *Lakner R., Hangos, K. M., Cameron, I.* (2003) Construction of minimal models for control purposes. *Computer-Aided Chemical Engineering* 14, 755–760.

11 *Leitold, A., Hangos, K. M.* (1998) On algebraic model transformations. *In: CPM'98. 3rd IEEE European workshop on computer-intensive methods in control and data processing.* Prague, Czech Republic, pp. 133–138.

12 *Leitold, A., Hangos, K. M.* (2001) Structural solvability analysis of dynamic process models. *Computers and Chemical Engineering* 25, 1633–1646.

13 *ModKit* (2000) Computer-Aided Process Modeling (ModKit), http: www.lfpt.rwth-aachen.de/Research/Modeling/modkit.html.

14 *Murota, K.* (1987) *Systems Analysis by Graphs and Matroids.* Springer, Berlin.

15 *Németh, H., Palkovics, L., Hangos, K. M.* (2003) System identification of an electro-pneumatic protection valve. *Research Report* SCL-001/2003. Systems and Control Labora-

tory. Computer and Automation Research Institute, Budapest, Hungary, http: daedalus. scl. sztaki. hu/PCRG/.

16 *Robertson, G. A., Cameron, I. T.* (1997) Analysis of dynamic process models for structural insight and model reduction I: structural identification measures. *Computers and Chemical Engineering* 21, 455–473.

17 *Williams, R. P. B., Keays, R., McGahey, S., Cameron, I. T., Hangos, K. M.* (2002) SCHEMA: Describing models using a modeling object language. CHEMECA–2002, pp. 922.