An improved method for nonlinear model reduction using balancing of empirical gramians

Juergen Hahn, Thomas F. Edgar *

Department of Chemical Engineering, The University of Texas at Austin, Austin, TX 78712-1062, USA

Received 17 December 1999; received in revised form 17 April 2002; accepted 17 April 2002

Abstract

Nonlinear model predictive control has become increasingly popular in the chemical process industry. Highly accurate models can now be simulated with modern dynamic simulators combined with powerful optimization algorithms. However, computational requirements grow with the complexity of the models. Many rigorous dynamic models require too much computation time to be useful for real-time model based controllers. One possible solution to this is the application of model reduction techniques. The method introduced here reduces nonlinear systems while retaining most of the input–output properties of the original system. The technique is based on empirical gramians that capture the nonlinear behavior of the system near an operating point. The gramians are then balanced and the less important states reduced via a Galerkin projection which is performed onto the remaining states. This method has the advantage that it only requires linear matrix computations while being applicable to nonlinear systems. © 2002 Elsevier Science Ltd. All rights reserved.

Keywords: Nonlinear system; Real-time model; Gramians

1. Introduction

Model-based control methods such as linear-quadratic regulator, $H_\infty$, or model predictive control utilize models that can be very complex, consisting of ordinary differential equations (ODEs), differential algebraic equations (DAEs) or even partial differential equations (PDEs). One normally discretizes models containing PDEs in their spatial dimensions to reduce them to a set of ODEs. The result is a high order model that is adequate for off-line simulations, but is usually too complex to be solved in real-time. Therefore it is desirable to reduce a high order model into one with fewer state variables, while retaining most of the input–output behavior. It must be stressed that achieving faster simulation and optimization times is not the only goal for applying model reduction. It is just as important to gain insight into the true cause of the observable dynamics of a system.

There exist many different model reduction techniques, but only a few are optimal in some sense. One example of those is the balancing method for model reduction of controlled systems which consists of two steps. The first step is to find a transformation that balances the observability and controllability gramians in order to determine which states have the greatest contribution to the input–output behavior. The second step is to perform a Galerkin projection onto the states corresponding to the largest singular values of the balanced gramians for the region of interest in state-space (Lall, Marsden, & Glavaski, 1999).

Methods and supporting theories for the reduction of linear models are well established as surveyed by Van Woerkom (1990). For nonlinear systems no complete theory for model reduction currently exists (Marquardt, 2001). The approach presented here combines the ease of use of the linear system approach, with the flexibility required for nonlinear systems and leads to an efficient algorithm that can be used for the reduction of nonlinear models.

* Corresponding author. Tel.: + 1-512-471-3080; fax: + 1-512-471-7060
E-mail address: edgar@mail.utexas.edu (T.F. Edgar).
2. Previous work—nonlinear model reduction

Hendriks and van Bergen (1992) developed a reduction method for phase equilibria calculations. Their method reduces the number of equations that is required to describe the thermodynamic properties and is found to be potentially useful for process and reservoir simulation. Perregaard (1993) proposes a more general model reduction and simplification procedure. One part of this procedure is the substitution of rigorous thermodynamic models with simpler ones. For this calculation parameters have to be fitted for each specific application. The model reduction procedure includes partitioning of the model equations, where different integration techniques can be used on the partitions in order to achieve shorter simulation times. Biegler, Grossmann, and Westerberg (1985) comment on the use of simple thermodynamic models for simulation and optimization purposes. These simple physical properties models perform well in simulations, but can lead to problems for optimization calculations. This is due to the fact that the reduced model has different gradients than the rigorous one and this can lead to false optima. Ganesh and Biegler (1987) extended this idea and developed a framework, that makes use of simplified models, but guarantees convergence to the rigorous model optimum.

Van Breusegem and Bastin (1991) used a singular perturbation approach to reduce dynamic reaction models. The assumption was made that some of the reactions are much faster than others and that these fast reactions can be assumed to be in steady state. Harmon, Duboc, and Bonvin (1995) used target factor analysis (TFA) to model a batch reactor. TFA generates an observed stoichiometric space from batch reactor data using the singular value decomposition. The main idea behind TFA is to test target stoichiometries to see if they lie within the observed stoichiometric space. TFA will estimate parameters and therefore introduces system identification elements into the reduction process. Lumping is another method that combines reduction with parameter estimation. This method has been used for over 30 years in the petroleum refining industry for reactions where the detailed kinetics for individual species are unknown. Some references for lumping are given by Wei and Kuo (1969a,b) and Jacob, Gross, Voltz, and Weekman (1976). Edwards (1997) investigated kinetic model reduction using mixed integer nonlinear programming (MINLP) and genetic algorithms (GA). The MINLP approach worked well for smaller reaction systems, whereas the GA method was applicable to large reaction networks as well.

While all of the thermodynamic and kinetic model reduction methods work well for their specific tasks, very few can be extended to other nonlinear systems. One method for which this is possible is singular perturbations. Robertson and Cameron (1997a,b) used singular and regular perturbations to identify system dynamics and reduce the order of the model of a multi-stage compressor. Their approach splits the system up into slow, medium and fast modes. The modes are identified by examining the eigenvalues of the linearized system. Therefore, the system must be linearized at points spanning the whole operating range. This method is a hybrid approach that uses techniques from model reduction for linear systems.

Scherpen (1993) described how to obtain grammars for certain classes of stable nonlinear systems and under what conditions a balanced realization exists. Furthermore, some of the ideas from Anderson and Liu (1989), Liu and Anderson (1986) about balancing of systems with a LQG controller using coprime factorization for linear systems were extended to nonlinear systems by Scherpen (1994). However, the given procedures present computational difficulties and in general a closed form solution is not guaranteed.

Newman and Krishnaprasad (1998b) compared chemical vapor deposition models reduced by principal component analysis (PCA) and balancing. The conclusion is that both approaches work equally well for this specific application, but no predictions can be made on how this translates to other applications. Furthermore, they linearize the system in order to apply balancing to it. In another paper by the same authors (Newman & Krishnaprasad, 1998a) a Monte-Carlo approach is used to provide an estimate for the controllability and observability function of a mildly nonlinear system. The resulting singular value functions show that one of the states has more impact on the system behavior than the other. However, this approach was only used for a system with two states and the projection step that actually reduces the system still needs to be performed. This work has been extended (Newman & Krishnaprasad, 2000) in order to include all the steps that are required for reducing nonlinear models with the proposed method. However, after the coordinate transformation is applied, and even without reducing the model, the transformed system does not exhibit the same input–output behavior as the original system due to the approximations that were applied during the computational procedure.

Due to the problems encountered with general nonlinear balancing procedures, several methods that perform a Galerkin projection, based upon a linear coordinate transformation, have been developed. Pallaske (1987) investigated a procedure where the linear transformation is found from a covariance matrix that is computed from data collected along system trajectories. These trajectories represent the system behavior under a constant input, but starting from different initial conditions. Löfler and Marquardt (1991) extended this model reduction approach to models de-
scribed by DAE systems. They investigated the case of trajectories generated by different initial conditions under constant inputs as well as the case where the trajectories start at the steady state operating point and are generated by step changes in the inputs to the system. Similarities between these methods (Pallaske, 1987; Löfler & Marquardt, 1991) and balanced model reduction have been pointed out in the work by Hahn, Edgar, and Marquardt (in press). Lee, Eom, Chung, Choi, and Yang (2000) computed the linear coordinate transformation by balancing a system that was generated using subspace identification. This identified system was generated from data collected along system trajectories.

3. Review of model reduction for linear systems

Linear model reduction techniques are based on first principles models, empirical models, or a combination of both. The most common basic approaches are system identification (Van Woerkom, 1990), lumping (Jacobs et al., 1976; Wei & Kuo, 1969a; Wei & Kuo, 1969b), singular perturbation (Robertson & Cameron, 1997a,b), and balancing (Samar, Postlethwaite, & Gu, 1995; Skogestad & Postlethwaite, 1997).

Model reduction via balancing, which is an approach that does not require any parameter fitting, seems to be the most promising of these techniques for extension to nonlinear systems.

3.1. Balancing for linear systems

A linear time invariant system
\[ \dot{x}(t) = Ax(t) + Bu(t) \]
\[ y(t) = Cx(t) \]
(1)

that is open loop stable has controllability and observability gramians, given by Definition 1 and Definition 2 below.

Definition 1 (Linear controllability gramian).
\[ W_C = \int_0^\infty e^{At}BB^Te^{A^T}dt \]

If the system is stable and controllable then the controllability gramian will have full rank.

Definition 2 (Linear observability gramian).
\[ W_O = \int_0^\infty e^{A^T}C^Te^{At}dt \]

For stable and observable systems the observability gramian will have full rank. If the open loop system is unstable or marginally stable then the gramians can not be computed because \( W_C \to \infty \) and \( W_O \to \infty \) (Skogestad & Postlethwaite, 1997). Furthermore the linear gramians \( W_C \) and \( W_O \) are the unique positive definite solutions of the Lyapunov equations (Eq. (2) and Eq. (3)).

\[ AW_C + W_C A^T = -BB^T \]
(2)
\[ A^TW_O + W_O A = -C^TC \]
(3)

Definition 3 (Balanced form). A system whose gramians are equal and have the following form
\[ W_C = W_O = \Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma_n \end{bmatrix} \]

\( \sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \ldots \sigma_n \geq 0 \)
is called balanced. The transformed gramians are given by,
\[ \bar{W}_C = TW_C T^T \]
\[ \bar{W}_O = (T^{-1})^TW_O T^{-1} \]

where \( T \) is a transformation matrix and the \( \sigma_i \)'s are the Hankel singular values.

It can be shown that there exists a state space transformation
\[ \bar{x} = Tx \]
(4)
such that the transformed system given by
\[ \bar{x} = TAT^{-1}\bar{x} + TBu = \bar{A}\bar{x} + \bar{B}u \]
\[ y = CT^{-1}\bar{x} + \bar{C}\bar{x} \]
(5)
is in balanced form. The new system given by Eq. (5) is then called a balanced realization.

Moore (1981) introduced balancing with the aim of using it as a tool for model reduction. The main idea is that the singular values of the controllability gramian correspond to the amount of energy that has to be put into the system in order to move the corresponding states. For the observability gramian, its singular values refer to the energy that is generated by the corresponding states.

If a linear system is in balanced form, the Hankel singular values provide a measure for the importance of the states, because the state with the largest singular value is the one which is affected the most by control moves and the output is most affected by a change of this state. Therefore the states corresponding to the largest singular values influence the input–output behavior the most. For model reduction, the states that contribute very little to the input–output behavior can
be eliminated, and the reduced system retains the best possible approximation to the full-order system (Moore, 1981).

3.2. Model reduction by truncation for linear systems

Once the system is in balanced form the state vector can be partitioned into more important states \((\bar{x}_1)\) and less important components \((\bar{x}_2)\) as shown in Eq. (6).

\[
\begin{pmatrix}
\dot{\bar{x}}_1 \\
\dot{\bar{x}}_2
\end{pmatrix} =
\begin{pmatrix}
\bar{A}_{11} & \bar{A}_{12} \\
\bar{A}_{21} & \bar{A}_{22}
\end{pmatrix}
\begin{pmatrix}
\bar{x}_1 \\
\bar{x}_2
\end{pmatrix} +
\begin{pmatrix}
\bar{B}_1 \\
\bar{B}_2
\end{pmatrix} u
\]

\[ y = (\bar{C}_1 \bar{C}_2)\begin{pmatrix}
\bar{x}_1 \\
\bar{x}_2
\end{pmatrix} \tag{6} \]

One approach to model reduction is balanced truncation of the states that correspond to small singular values. This reduction method results in a good approximation to the original system over the whole frequency range. The disadvantage of this approach is that it does not preserve the steady state behavior of the original system and therefore will result in offset (Skogestad & Postlethwaite, 1997).

Reduction by truncation leads to the balanced reduced system given by Eq. (7).

\[
\begin{pmatrix}
\dot{\bar{x}}_1 \\
\dot{\bar{x}}_2
\end{pmatrix} = \bar{A}_{11}\bar{x}_1 + \bar{B}_1 u
\]

\[ y = \bar{C}_1\bar{x}_1 \tag{7} \]

The result is a system of ODEs which contains fewer states than the original system. The number of states that can be truncated depends on the system itself and on the accuracy that is required for the system behavior. Skogestad and Postlethwaite (1997) provide a bound for the error for balanced truncation of linear systems. Furthermore, a comprehensive review of balanced model reduction for linear systems can be found in the literature (Skogestad & Postlethwaite, 1997; Zhou, Doyle, & Glover, 1996).

3.3. Model reduction by residualization for linear systems

If it is important to maintain the same steady state behavior for the reduced system as for the full-order system, residualization can provide better results than truncation. Residualization is based on the idea that the derivatives of the states corresponding to small Hankel singular values can be approximated to zero while the rest of the system is retained.

\[
\begin{pmatrix}
\dot{\bar{x}}_1 \\
\dot{\bar{x}}_2 \\
\dot{\bar{x}}_3
\end{pmatrix} =
\begin{pmatrix}
\bar{A}_{11} & \bar{A}_{12} & \bar{A}_{13} \\
\bar{A}_{21} & \bar{A}_{22} & \bar{A}_{23} \\
\bar{A}_{31} & \bar{A}_{32} & \bar{A}_{33}
\end{pmatrix}
\begin{pmatrix}
\bar{x}_1 \\
\bar{x}_2 \\
\bar{x}_3
\end{pmatrix} +
\begin{pmatrix}
\bar{B}_1 \\
\bar{B}_2 \\
\bar{B}_3
\end{pmatrix} u
\]

\[ y = (\bar{C}_1 \bar{C}_2)\begin{pmatrix}
\bar{x}_1 \\
\bar{x}_2
\end{pmatrix} \tag{8} \]

The resulting set of DAEs can be further simplified by eliminating the states \(\bar{x}_3\) from the set of DAEs, transforming it into a system of ODEs in Eqs. (9a) and (9b).

\[
\dot{\bar{x}}_1 = \bar{A}\bar{x}_1 + \bar{B}_1 u
\]

\[ y = \bar{C}\bar{x}_1 + \bar{D} u \tag{9a} \]

where

\[
\bar{A} = \bar{A}_{11} - \bar{A}_{12}\bar{A}_{22}^{-1}\bar{A}_{21}
\]

\[
\bar{B} = \bar{B}_1 - \bar{A}_{12}\bar{A}_{22}^{-1}\bar{B}_2
\]

\[
\bar{C} = \bar{C}_1 - \bar{C}_2\bar{A}_{22}^{-1}\bar{A}_{21}
\]

\[
\bar{D} = -\bar{C}_2\bar{A}_{22}^{-1}\bar{B}_2 \tag{9b} \]

Both balanced residualization and truncation methods are related to modal reduction. Whereas modal reduction keeps most of the system behavior intact, i.e. modal truncation preserves the poles and zeros of a system, the reduction methods based upon balancing retain most of the input–output behavior of the system. Due to this, modal reduction is optimal for controlled systems because it is based upon the dynamic behavior that the states of the system exhibit. On the other hand modal reduction does not consider the influence of the input-to-state or the state-to-output behavior. Therefore, balanced reduction methods are more suitable than methods based upon modal reduction for systems that will be used for controller design because they preserve the input–output behavior. This can also be concluded from the error bounds given in Skogestad and Postlethwaite (1997) for modal reduction as well as balanced reduction of linear systems.

The following example illustrates the balancing procedure and the different results that balanced truncation and residualization produce.

Example 1 (Model reduction of a reaction network). Consider an isothermal CSTR with the following reactions

\[
A \rightarrow B \rightarrow C
\]

where \(k_1 = 1.0, k_2 = 0.1\) and both reactions are first order. The inlet concentration of A can be manipulated \((u)\) and the concentration of C at the outlet can be measured. The state vector is defined as the concentrations of species A, B, and C.

This leads to the linear time invariant system (Eq. (10)), which is stable, observable and controllable.

\[
\begin{pmatrix}
\dot{x}_1 \\
\dot{x}_2 \\
\dot{x}_3
\end{pmatrix} =
\begin{pmatrix}
-2 & 0 & 0 \\
1.0 & -1.1 & 0 \\
0 & 0.1 & -1
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2 \\
x_3
\end{pmatrix} +
\begin{pmatrix}
2 \\
0 \\
0
\end{pmatrix} u
\]
\[ y = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \]  

Equation (10)

The gramians for this system are given by

\[ W_C = \begin{pmatrix} 1.0 & 0.3226 & 0.01075 \\ 0.3226 & 0.2933 & 0.01908 \\ 0.01075 & 0.01908 & 0.001908 \end{pmatrix} \]

\[ W_O = \begin{pmatrix} 0.0004771 & 0.0009542 & 0.007937 \\ 0.0009542 & 0.002165 & 0.02380 \\ 0.007937 & 0.02380 & 0.5 \end{pmatrix} \]  

Equation (11)

This system is not in balanced form, because the gramians are not equal. However, the transformation matrix that balances this linear system is

\[ T = \begin{pmatrix} -0.07729 & -0.1845 & -2.530 \\ 0.08866 & 0.08595 & -2.776 \\ 0.04344 & -0.1528 & 1.343 \end{pmatrix} \]  

Equation (12)

The resulting balanced gramian matrices are identical, i.e.

\[ \Sigma = \begin{pmatrix} 0.05939 & 0 & 0 \\ 0 & 0.01525 & 0 \\ 0 & 0 & 0.001316 \end{pmatrix} \]  

Equation (13)

From this balanced realization it can be concluded that the third state of the balanced system contributes much less to the input–output behavior than the other two, because its singular value is at least an order of magnitude smaller than the other singular values.

Using \( T \), the balanced realization of this system is given by

\[ \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = \begin{pmatrix} -0.2012 & -0.621 & 0.2212 \\ 0.621 & -1.031 & -1.106 \\ -0.2212 & -1.106 & -2.868 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} -0.1546 \\ 0.1773 \\ 0.08688 \end{pmatrix} u \]

\[ y = (-0.1546 & -0.1773 & 0.08688) \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \]  

Equation (14)

This system has the same input–output properties as the original system. However, its gramians are balanced and the states are ordered from top to bottom so that the each state contributes more to the input–output behavior than any subsequent (higher-numbered) state. Hence, model reduction should affect the states at the bottom the most, because this will preserve most of the properties of the original system. For comparison the full-order system with three states was reduced to two systems consisting of two states, one generated by truncation and the other by residualization. Fig. 1 compares the three systems using Bode plots. The solid line is the full-order system, the dashed line the system that was reduced via truncated model and the dotted line corresponds to the residualized model. As can be
seen in Fig. 1 all three systems behave very similar at low frequencies. At high frequencies the response of the truncated system provides a better approximation to the real system than the residualized model. However, the models are usually chosen to represent a system mostly at low frequencies and the high frequency response is therefore less important for controller operation (Skogestad & Postlethwaite, 1997).

Another important measure for model comparison is the steady state behavior. As can be seen in Table 1, the output of the residualized system has the same steady state behavior. As can be seen in Table 1, the truncated system includes an offset of 2.6%.

### 4. Model reduction for nonlinear systems

Because most engineering processes are nonlinear, it is sometimes not satisfactory to use linear model-based controllers. In these cases nonlinear MPC will usually lead to better controller performance, but requires the solution of a nonlinear programming problem at each time step, which can be computationally intensive. Since the more accurate models tend to be complex, it is desirable to reduce these models while retaining most of the input–output behavior.

To extend the model reduction approaches discussed previously to nonlinear systems several new challenges arise, such as numerical problems and a lack of suitable theory.

#### 4.1. Balancing for nonlinear systems

A general balancing scheme for nonlinear systems is not available. One possibility is to linearize the system and apply the methods described in Section 3. However, the nonlinear behavior will be lost and some results for a specific class of nonlinear system as shown below in Eq. (15) do exist. In order to go into more detail a few terms need to be defined.

A nonlinear control-affine system has a representation of the following form

\[
\begin{align*}
\dot{x}(t) &= f(x(t)) + g(x(t))u(t) \\
y(t) &= h(x(t))
\end{align*}
\]

where \( f, g, h \) are functions of class \( C^\infty \), \( f(0) = 0 \), and \( h(0) = 0 \). Since no theory for general nonlinear balancing exists, control-affine systems are the only type of systems that can be balanced with the methods presented in this section. A system that is not control-affine has to be linearized with regard to the control input, in order to balance it.

#### Definition 4 (Controllability energy function).

\[
L_C = \min_{u \in L_2((\infty,0),\mathbb{R})} \frac{1}{2} \int_0^\infty \|u(t)\|^2 dt
\]

#### Definition 5 (Observability energy function).

\[
L_O = \frac{1}{2} \int_0^\infty \|y(t)\|^2 dt
\]

where \( x(0) = x_0 \), \( u(t) \equiv 0 \), \( 0 \leq t < \infty \).

These observability and controllability energy functions are not necessarily finite. In particular, \( L_O \) will not be finite if the system is unstable. If \( x_0 \) can not be reached from the origin, then by convention \( L_C(x_0) \) will be infinite.

These energy functions are related to the gramians and their relation for the linear case is given by Eq. (16) and Eq. (17).

\[
\begin{align*}
L_C(x_0) &= \frac{1}{2} x_0^T W_C^{-1} x_0 \quad (16) \\
L_O(x_0) &= \frac{1}{2} x_0^T W_O x_0 \quad (17)
\end{align*}
\]

Unfortunately it is not easy to calculate the energy functions in the nonlinear case. In the linear case the calculation is performed by solving the given Lyapunov equations from Section 3. For nonlinear systems either the optimal control problem given by Definition 4 and Definition 5 has to be solved for each grid point or the Hamilton–Jacobi PDEs (Eq. (18) and Eq. (19)) need to be solved, assuming that the origin is an asymptotically stable equilibrium point of \( f(x) \) (Scherpen, 1993).

\[
\begin{align*}
\frac{\partial L_O(x)}{\partial x} f(x) + \frac{1}{2} h^T(x) h(x) &= 0, L_O(0) = 0 \quad (18) \\
\frac{\partial L_C(x)}{\partial x} f(x) + \frac{1}{2} g(x) g^T(x) \frac{\partial^T L_C(x)}{\partial x} &= 0, L_C(0) = 0 \quad (19)
\end{align*}
\]

For linear systems Eq. (18) and Eq. (19) reduce to the already stated Lyapunov equations (Eq. (2) and Eq. (3)).

Under certain conditions Scherpen (1993, 1994) has shown that a balanced realization of the energy functions exists in a neighborhood of the origin. The balanced energy functions are given below by Eq. (21) and Eq. (22).

\[
x = \psi(\eta^{-1}(\hat{x}))
\]
The balanced realization contains a diagonal matrix with ordered singular value functions \( \sigma_i \). These singular value functions have to be evaluated over the state space region of interest. As in the linear case the magnitude of the singular value functions represents the contribution of a state to the input–output behavior. The transformed system is of the form given by Eq. (23).

\[
\begin{align*}
\dot{x}(t) &= f(x(t), u(t)) + g(x(t))u(t) \\
y(t) &= h(x(t))
\end{align*}
\]

(23)

The approach presented here calculates empirical gramians from process data. These gramians are then balanced by the same procedure as is used for linear systems. The balancing transformation is used within a Galerkin projection in order to transform the nonlinear system into balanced form. The resulting nonlinear equations can be reduced using different truncation or residualization methods.

5.1. Empirical gramians

Since it is not satisfactory to reduce nonlinear systems based on linear gramians and nonlinear energy functions are too computationally intensive to compute, a hybrid method is necessary. This can be done using empirical gramians defined by Lall et al. (1999). For the nonlinear system

\[
\begin{align*}
\dot{x}(t) &= f(x(t), u(t)) \\
y(t) &= h(x(t))
\end{align*}
\]

the following sets need to be defined for empirical gramians:

\[
T^n = \{T_1, \ldots, T_r \} \quad T_i \in \mathbb{R}^{n \times n}, \quad T_i^T T_i = I, \quad i = 1, \ldots, r
\]

\[
M = \{c_1, \ldots, c_s \} \quad c_i \in \mathbb{R}, \quad c_i > 0, \quad i = 1, \ldots, s
\]

\[
E^n = \{e_1, \ldots, e_m \} \quad \text{standard unit vectors in } \mathbb{R}^n
\]

\( r, \) number of matrices for excitation/impulse directions; \( s, \) number of different excitation/perturbation sizes for each direction; \( n, \) number of inputs to the system for Definition 6 and number of states of the full-order system for Definition 7.

Definition 6 (Empirical controllability gramian). Let \( T^n, \ E^n, \) and \( M \) be given sets as described above, where \( p \) is the number of inputs. The empirical controllability gramian is defined by

\[
W_C = \sum_{i=1}^r \sum_{m=1}^s \sum_{l \in M} \frac{1}{T_i \Psi_{2lm}(t) T_i^T} \int_0^{\infty} \Phi_{1lm}(t) dt
\]

where \( \Phi_{1lm}(t) \in \mathbb{R}^{n \times n} \) is given by \( \Phi_{1lm}(t) = (x_{1lm}(t) - x_{0,1lm})T_i^T (x_{1lm}(t) - x_{0,1lm}) \), and \( x_{1lm}(t) \) is the state of the nonlinear system corresponding to the impulse input \( u(t) = c_i T_i e \), \( e \) is the steady state of the system.

Definition 7 (Empirical observability gramian). Let \( T^n, \ E^n, \) and \( M \) be given sets as described above, where \( n \) is the number of states. The empirical observability gramian is defined by

\[
W_O = \sum_{i=1}^r \sum_{m=1}^s \sum_{l \in M} \frac{1}{T_i \Psi_{2lm}(t) T_i^T} \int_0^{\infty} \Psi_{1lm}(t) T_i^T dt
\]

where \( \Psi_{1lm}(t) \in \mathbb{R}^{n \times n} \) is given by \( \Psi_{1lm}(t) = (y_{1lm}(t) - y_{0,1lm})^T (y_{1lm}(t) - y_{0,1lm}) \), and \( y_{1lm}(t) \) is the output of the
nonlinear system corresponding to the initial condition 
\( x(0) = c_m T \epsilon_i + x_0 \). The \( y_u^{(in)} \) refers to the output measurement corresponding to the steady state of the system.

Lall et al. (1999) have shown that both of these empirical gramians reduce to linear gramians for linear models. Furthermore, the empirical gramians will reduce to gramians of the linearized system for small perturbations around the operating point. It was originally assumed that the empirical gramians can be calculated for any nonlinear system of the above form. However, we believe that the use of empirical gramians is limited to control-affine systems, because these are the only systems where impulse response behavior can be calculated.

These empirical gramians have to be determined from experimental or simulation data, collected within a region where the process is to be controlled. The empirical gramian matrices capture part of the nonlinear behavior within the region of operation.

5.2. Galerkin projection

The Galerkin projection is based upon the idea that the dynamics of a system can be replaced with the dynamics based upon a subspace of the original system. This technique has been used extensively to find sets of ODEs to replace a PDE as well as to construct mathematical models of lower order.

Given an autonomous system it is possible to approximate the system with a lower order model of the form given in Eq. (25).

\[
\dot{x}(t) = \bar{f}(\bar{x}(t))
\]

(25)

To apply the Galerkin projection, \( x(t) \) has to be rewritten as

\[
x(t) = \bar{P}^T \bar{x}(t) + r(t)
\]

(26)

where \( r(t) \) is the residual. If \( \bar{P} \) is a square matrix of full rank and its dimension is equal to the number of states of the system, then the residual is zero. However, in order to obtain a model of reduced dimension, \( \bar{P} \) needs to be chosen in such a way that the residual is orthogonal to the reduced subspace. The projection results in the following system of equations that exactly represents the original system:

\[
\dot{\bar{x}}(t) = \bar{P} \bar{f}(\bar{P}^T \bar{x}(t) + r(t))
\]

(27)

When the residual \( r(t) \) is deleted, some of the system behavior is lost and the reduced model can only approximate the behavior of the original model. In order to achieve a good approximation \( \bar{P} \) has to be chosen in such a way that \( r(t) \) is minimized. The resulting reduced-order system is then given by

\[
\dot{\bar{x}}(t) = \bar{P} \bar{f}(\bar{P}^T \bar{x}(t))
\]

(28)

Special attention has to be given to the choice of \( \bar{P} \). For uncontrolled systems it is sufficient to use PCA to find the projection, but for controlled systems balancing has to be used in order to account for the input–output behavior of the system as well (Lall et al., 1999). The reason PCA can be applied to uncontrolled systems is that it describes how well a set of data can be approximated by projecting it into a subspace of the original state-space. The measure for performance of this method is how much of the variance of a correlated data set can be captured in the reduced subspace. However, this method does not take into account the input–output behavior, which is important for controlled systems.

The method that is introduced here implements a slightly modified version of the Galerkin projection. This method chooses \( \bar{P} \) to be the product of two matrices: the transformation matrix \( T \) that balances the empirical gramians and a reduction matrix \( P \) that eliminates the part of the subspace that contributes little or nothing to the input–output behavior of the system. \( P \) is a matrix of appropriate dimension that includes an identity matrix with rank equal to the number of states in the reduced-order model. The remaining rows and columns of \( P \) are filled with zero entries.

This method has the advantage that it only requires simple matrix computations for the reduction of nonlinear models. Compared to the methods in Section 4 this is a substantial improvement. A drawback is that a linear projection is used and the global behavior of the system can not be predicted using the reduced model. However, every process has a certain operating region and it is within this region that the reduced system should approximate the behavior of the full-order model. The region in which this reduced model provides a good approximation to the full-order model is certainly greater than for a simple linearized model.

5.3. Computational procedure

All the tools that are required for nonlinear model reduction have been presented. One has to choose the region of operation for a process. Then data is collected within this region either by experiments or simulations and empirical gramians are calculated. It should be noted that the collected data should reflect the physical behavior of the system within the whole operating region. For the examples in this paper the impulse inputs and initial states were uniformly distributed on the unit sphere. This seems like an arbitrary choice, but no generally applicable optimal experiment design exists for nonlinear systems. Next, the gramians are balanced by the methods for linear model reduction as described in Section 3. From the balanced gramians it
can be concluded which states contribute the most to the input–output behavior of the system. States that are either unobservable or uncontrollable do not influence the input–output behavior and can be eliminated. This can be done simultaneously with the reduction of the less important states, since these states will have singular values of zero. The general balancing algorithm as described by Hahn and Edgar (2002) has been implemented and applied to positive semi-definite gramian matrices for this case.

Changes of an order of magnitude or more in the singular values of the balanced gramians indicate which states should be eliminated. Once the reduced number of states has been decided a Galerkin projection is performed to produce the reduced-order nonlinear system. The number of remaining states is adjusted by a trial and error procedure to achieve optimum performance. If the system behavior of the reduced model is very different from the original system then the number of states for the reduced model should be increased. If the match of the input–output behavior is close to perfect, a smaller order model might still perform satisfactorily. If the system response to changes in the input is correct, but the offset at steady state degrades controller performance, model residualization instead of truncation should be performed.

5.4. Nonlinear model reduction by balanced truncation

As in the linear case balanced truncation can be performed on the system. This results in the following system given by Eq. (29).

\[
\dot{x}(t) = P T f(T^{-1} P^T \bar{x}(t)) + P T g(T^{-1} P^T \bar{x}(t)) u(t)
\]

\[
y(t) = h(T^{-1} P^T \bar{x}(t))
\]  

(29)

In this reduced system \(T\) represents the transformation matrix that balances the gramians and \(P = [I \ 0]\) is the projection matrix which has the rank of the reduced system.

This approach leads to a system of ODEs of reduced-order, which, unlike regular PCA, takes into account the input–output behavior (Lall et al., 1999).

Example 2 (Model reduction of two nonisothermal CSTRs in series). Consider a system of two CSTRs in series with an irreversible reaction \(A \rightarrow B\). Each reactor has a mass, component, and energy balance, resulting in six nonlinear differential equations (Appendix A). The volume and temperature of the second reactor can be measured. The manipulated variables are the valve position at the outlet of the second reactor, as well as the heat transfer rate from the first reactor. This results in a system with six states, two inputs and two outputs, which makes this system controllable and observable. Based upon a model by Henson and Seborg (1997), this example also includes volume balances for the two reactors, a controller for the flowrate leaving the second reactor, and the reactor temperature is controlled via the heat transfer rate in order to make the system control-affine. The operating region for the computation of the empirical gramians was chosen to be \(\pm 8\%\) around the steady state values. The empirical gramians are given by Eq. (30) and Eq. (31):

\[
W_C = \begin{bmatrix}
0.3289 \times 10^4 & -0.3711 \times 10^4 & 0.1833 \times 10^4 & 0.3290 \times 10^4 & -0.93 & 0.7811 \times 10^4 \\
-0.3711 \times 10^4 & 0.7513 \times 10^{-1} & -0.4473 \times 10^3 & -0.7418 \times 10^4 & 0.68 \times 10^{-2} & -0.3286 \times 10^2 \\
0.1833 \times 10^4 & -0.4473 \times 10^2 & 0.2684 \times 10^5 & 0.4387 \times 10^4 & -0.41 \times 10^4 & 0.2028 \times 10^5 \\
0.3290 \times 10^4 & -0.7418 \times 10^4 & 0.4387 \times 10^4 & 0.6638 \times 10^4 & -0.235 \times 10^4 & 0.2348 \times 10^4 \\
-0.93 & 0.68 \times 10^{-2} & -0.41 \times 10^1 & -0.235 \times 10^1 & 0.1169 \times 10^{-2} & -0.9275 \times 10^1 \\
0.7811 \times 10^4 & -0.3286 \times 10^2 & 0.2028 \times 10^5 & 0.2348 \times 10^2 & -0.9275 \times 10^1 & \quad 0.9261 \times 10^5
\end{bmatrix}
\]  

(30)

\[
W_O = \begin{bmatrix}
0.2682 \times 10^5 & 0.3478 \times 10^2 & 0.1218 \times 10^4 & 0.4501 \times 10^4 & 0.1525 \times 10^2 & 0.1942 \times 10^5 \\
0.3478 \times 10^2 & 0.6429 \times 10^1 & 0.3985 \times 10^3 & -0.3153 \times 10^2 & 0.3625 & 0.4602 \times 10^3 \\
0.1218 \times 10^4 & 0.3985 \times 10^3 & 0.2483 \times 10^6 & -0.1647 \times 10^4 & 0.2047 \times 10^2 & 0.2601 \times 10^4 \\
0.4501 \times 10^4 & -0.3153 \times 10^2 & -0.1647 \times 10^4 & 0.8041 \times 10^4 & -0.6255 \times 10^1 & -0.7983 \times 10^4 \\
0.1525 \times 10^2 & 0.3625 & 0.2047 \times 10^2 & -0.6255 \times 10^2 & 0.5745 \times 10^{-1} & 0.7559 \times 10^2 \\
0.1942 \times 10^5 & 0.4602 \times 10^3 & 0.2601 \times 10^5 & -0.7983 \times 10^4 & 0.7559 \times 10^2 & \quad 0.1113 \times 10^6
\end{bmatrix}
\]  

(31)
and the balanced gramians are shown in Eq. (32):

\[
\Sigma = \begin{bmatrix}
0.108 \times 10^6 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.1947 \times 10^5 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.8625 \times 10^4 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.1523 \times 10^3 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.1841 \times 10^{-2} & 0 \\
0 & 0 & 0 & 0 & 0 & 0.3312 \times 10^{-3}
\end{bmatrix}
\]

(32)

It can be concluded that a reduced system with four states will describe the model appropriately, since there is a gap of several orders of magnitude between the magnitude of the fourth and fifth Hankel singular value. This is illustrated in Figs. 2 and 3, where the trajectories of the outputs are shown for an 8% step change in the valve position at the outlet of the second reactor.

The response for the reduced-order nonlinear model obtained by balanced truncation approximates the full-order system fairly well. This can be compared to the full-order linearized system, which behaves very differently than the nonlinear one for the second output. This is due to the fact that the differential equation describing the volume is only slightly nonlinear, whereas the energy balance contains reaction terms and is highly nonlinear.

However, this truncation method might not yield the desired results; the trajectory of the reduced-order model in Fig. 3 shows inverse response behavior as well as offset. Further simulations have shown that the system behavior at the start of the simulation is mainly influenced by the change of the system due to the truncation method and not by a change in the inputs. The reason that this problem does not appear for reduced-order linear systems is that for linear systems the variables are always given in deviation form and their steady states values are equal to zero. For nonlinear systems, however, the steady state values are usually different from zero and neglecting the steady state values of these terms results in the behavior seen in the

Fig. 2. Volume of reactor 2 vs. time.
above example. Therefore we recommend reducing the number of states during truncation, but keeping the reduced states at their steady state value. This new procedure will result in a reduced system given by Eq. (33).

\[
\begin{align*}
\dot{x}_1(t) &= PT_f(T^{-1} \dot{x}(t)) + PT_g(T^{-1} \dot{x}(t))u(t) \\
\dot{x}_2(t) &= \dot{x}_{2ss}(0) \\
y(t) &= h(T^{-1} \dot{x}(t))
\end{align*}
\] (33)

Table 2 summarizes the required CPU times for integrating the models from time zero to the final time. It is interesting to note that the truncation method proposed in this paper results not only in more accurate models than the truncation method given by Eq. (29), but it also decreases computation time for this case. The reason for the shorter CPU time is that the model based on the new truncation method stays closer to its steady state value than the reduced model, given by Eq. (29), which exhibits inverse response behavior, before it reaches its steady state. Therefore the new system is in general numerically easier to integrate over the time horizon.

From the results shown in Table 2 and Table 3 it can be concluded that the CPU time for the reduced nonlinear system will not always be less than for the original full-order one. While the reduced balanced system has fewer equations, these tend to be more complex since each balanced state consists of a combination of the previous states. However, by eliminating a sufficient number of states the reduced model will be faster to solve than the full-order one. This is illustrated in Table 3, where the truncated model requires less computation time than the full-order model when at least one state is eliminated.

While Table 2 and Table 3 show a 35% reduction in computation time for the reduced-order model with four states compared to the full-order model for integrating the system from time zero to the final time, it has to be considered that a larger reduction in computation time can be achieved when the reduced-order model is used within a model predictive control algorithm that takes advantage of the number of equations in the model. For example, for linear model

<table>
<thead>
<tr>
<th>Full-order system</th>
<th>Truncated model with 4 states given by system (Eq. (29))</th>
<th>Truncated model with 4 states given by system (Eq. (33))</th>
</tr>
</thead>
<tbody>
<tr>
<td>13.30</td>
<td>11.99</td>
<td>9.283</td>
</tr>
</tbody>
</table>

Fig. 3. Temperature of reactor 2 vs. time.
predictive control the computational efforts scales with the cubic of the number of equations in the model. Therefore it is possible to achieve a significant reduction in the computation time for closed-loop applications, even if only a more modest decrease in the reduction of the computation time has been achieved for the open-loop case.

5.5. Nonlinear model reduction by balanced residualization

If the steady state behavior of the nonlinear system that is reduced by balanced truncation is not satisfactory, residualization will usually provide better results. Residualization is based upon the idea that the derivatives of the less important states are approximated by zero while the rest of the system is unchanged. This results in the reduced system given by Eq. (34).

\[ \dot{x}(t) = P_1 T f(T^{-1} x(t)) + P_1 T g(T^{-1} x(t))u(t) \]

\[ y(t) = h(T^{-1} x(t)) \]

(34)

Again, \( T \) is the transformation matrix that balances the gramians. \( P_1 = [I \ 0] \) is the projection matrix with rank of the reduced-order system, and the rank of \( P_2 = [0 \ I] \) is equal to the number of states that were reduced. The original state vector needs to be split up into \( \bar{x}_1 \) which contains the states of the reduced system and \( \bar{x}_2 \) which are the states that were reduced in the process. One of the main differences between residualization for linear and nonlinear systems is that in the nonlinear case we can not in general solve explicitly for the states \( \bar{x}_2 \). Instead the original system of ODEs reduces to a DAE system that contains fewer differential equations than the original system.

Example 2 revisited (Comparison of different methods).

Three different reduced-order models with four states each are derived from the full-order one and compared. The solid line represents the full-order model with six states. The dashed line corresponds to a model, which was reduced by Lall’s procedure. The dotted line is produced by a model that was reduced using the truncation method proposed in this paper and the dashed dotted line refers to a model which has been residualized. It can be concluded from Fig. 4 and Fig. 5 that the modified truncation method provides better results than the original truncation method, because it does not produce any inverse response and has a smaller offset, while the computational effort is comparable. For this example the residualized model is closest to the full-order one. However, this comes at the cost of having to solve a system of DAEs instead of ODEs as for the other two reduction methods.

Example 3 (Model reduction of a distillation column model).

Consider a distillation column with 30 trays for the separation of a binary mixture as it was used by Benallou, Seborg, and Mellichamp (1986) and Horton, Bequette, and Edgar (1991). The column has 32 states and is assumed to have a constant relative volatility of 1.6, and symmetric product compositions. The feed stream is introduced at the middle of the column on stage 17 and has a composition of \( x_F = 0.5 \). Distillate and bottoms purities are \( x_D = 0.935 \) and \( x_B = 0.065 \), respectively. The reflux ratio is set to 3.0 and can be controlled and the purity of the distillate is measured. The equations for this example are provided in Appendix B.

The operating region for this problem was chosen to be \( \pm 10\% \) around the steady state values. The empirical gramians for this example were computed by solving the system from time 0 to 125 min for the given excitation or state perturbation. The computation results in two matrices with 32 rows and columns each, which are not shown due to their size. The Hankel singular values in decreasing order are

\[ \begin{align*}
0.303 \times 10^{-2} & , 0.111 \times 10^{-4} & , 0.206 \times 10^{-5} & , 0.281 \times 10^{-6} & , 0.131 \times 10^{-6} & , 0.105 \times 10^{-7} , \end{align*} \]

and all remaining singular values are smaller than \( 10^{-8} \). It can be concluded from the Hankel singular values that the first state of the system captures the vast majority of the input–output behavior of the system, because there is a gap of more than two orders of magnitude between the magnitude of the first and second Hankel singular value. Based upon the
analysis of the magnitude of the Hankel singular values, it would also be useful to investigate systems that contain 2, 3, and 5 states (and possibly some systems that contain more states if the reduced-order system still does not reflect the input–output behavior to a desired degree). After reducing the model and investigating the reduced-order models it was determined that for this example that a system with five states results in

Fig. 4. Volume of reactor 2 vs. time.

Fig. 5. Temperature of reactor 2 vs. time.
an excellent approximation of the dynamics of the full-order system. A system with three states still results in a very close approximation and even a system with only a single state qualitatively describes the input–output behavior of the original system.

Fig. 6 contains four graphs for a decrease in the reflux ratio by 10%. The full-order system is represented by the solid line, the dashed line corresponds to a system that has been reduced by the new truncation method to three states and the dash-dotted line belongs to a truncated system with two states. Finally, the dotted graph represents the system response of a truncated system with only one state. It can be concluded by visual inspection of the graphs that the system behavior degenerates as the number of states is reduced. This can be concluded by inspecting when the first visible deviation between the full-order and the reduced-order model occurs as well as by the amount of offset that the system exhibits for this step-input change. The model with three states results in a very close approximation to the behavior of the full-order system, but even the model with just a single state describes the behavior of the original system qualitatively. The computation times for the full-order and the reduced systems are given in Table 4.

The main reason that only a few states are required to represent the system response is that many states are very close to being unobservable or uncontrollable. These states will be eliminated by the model reduction algorithm, since they give only a minor contribution to the input–output behavior. This phenomenon is generally observed for models describing distillation columns, since a distillation column model has a certain structure that makes it very suitable for model reduction.

It has been shown in Example 2 that for some models a reduced-order nonlinear model can result in a better approximation to the input–output behavior of the system than a full-order linear model. It has also been shown that reduced-order nonlinear models can result in a very good approximation to the behavior of the original model. However, often model reduction is performed in order to result in a smaller model that is easier to compute such that a controller can be designed that is based upon the reduced-order model. Investigating the behavior that a process exhibits, if it is controlled by a controller based upon a reduced order nonlinear model is investigated in the following example.

Example 4 (Comparison of closed-loop behavior). One of the main reasons for model reduction is to develop smaller models for on-line control applications. There-

<table>
<thead>
<tr>
<th>Number of States</th>
<th>Full-order</th>
<th>Truncated system as in Eq. (33) with</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 states</td>
<td>3.406</td>
<td>3.375</td>
</tr>
<tr>
<td>2 states</td>
<td>4.969</td>
<td>4.969</td>
</tr>
<tr>
<td>1 state</td>
<td>3.172</td>
<td>3.172</td>
</tr>
</tbody>
</table>
Therefore, the reduced models described in the previous examples have been used within a model predictive control scheme for set point changes and disturbance rejection. Additionally, in the case of the reduced nonlinear as well as for the linear model, model mismatch also exists between the model used to derive the controller and the real process. No computation times are given for this example, because the model predictive control algorithm that was implemented does not take special advantage of the reduced number of equations and, therefore, achieves reduction in computation times comparable to what was achieved for the open-loop case.

Different controllers based upon the three models are used to control the process described in Example 2 for a set point change as well as disturbance rejection and the results compared. The models the controllers are based upon are the original nonlinear model, a model that was reduced to four states via the new truncation method, and a linear model. The set point change occurs immediately, where the process is at its steady state and the set point is changed to 110 l for the volume and 445 K for the temperature of the second reactor. After reaching steady state 10 min into the simulation, the system is subjected to an output disturbance of 5 K in the temperature measurement of the second reactor. The dynamic responses to these conditions are shown in Figs. 7 and 8.

It can be concluded from the results that all of the models result in good performance for the set point change in the volume of the reactor. This is not surprising, since all three open-loop responses describe the dynamics of the volume fairly well. However, major differences can be seen for the temperature. The reduced model stays very close to the behavior of the full-order system that does not exhibit model mismatch between the controller and the plant model. This contrasts with the behavior of the linear model that takes much longer to reach its new set point and exhibits overshoot and oscillations.

The achievable closed-loop response of the distillation column model has also been investigated, in a similar fashion as was done for the reactor example. Two different controllers are compared in the following. The first one is a controller based upon the nonlinear model of the process itself, which will be the reference trajectory, since it contains no model mismatch and no other model can result in better controller performance. The other controller is based upon a nonlinear truncated model with one state. Fig. 9 represents the trajectories generated by a set point change to 0.95 for the distillate concentration. In addition to this, after 100 min of operation, the distillate concentration is subjected to an output disturbance of 0.01. This example tests both set point tracking and disturbance rejection. It can be seen that the reduced model results in a controller that closely matches the performance achieved by a controller based upon the full-order nonlinear model.
Fig. 8. Closed-loop responses of the temperature for set point change and disturbance rejection.

Fig. 9. Closed-loop responses of the concentration for set point change and disturbance rejection.
6. Conclusions

This paper presented a new approach for the reduction of nonlinear models. The proposed method combines elements from PCA and Galerkin projections with the balancing technique used in control engineering. The empirical gramians that were introduced in the process capture some of the nonlinear behavior of the system while being simple to compute. This is in stark contrast to full nonlinear balancing, which is computationally intensive for small systems and virtually impossible for industrial size problems.

The empirical gramians are balanced by simple matrix computations and a projection is used to map the nonlinear system to the reduced states. This paper introduced model residualization as well as a modification of balanced truncation, both of which provide better results than existing techniques for the reduction step. Balanced truncation will lead to a reduced system of ODEs, whereas nonlinear balanced residualization will result in a DAE system, which has the same steady state behavior as the original system.

Results were presented for the achievable performance for a reduced-order controller that is used to control the plant described by the full-order model. It was shown that the reduced-order controllers can result in excellent performance and can achieve better results than a controller that is based upon a full-order linear model.

Appendix A. Equations for Example 2

Volume balance for the first reactor:

\[
\frac{dV_1}{dt} = q_F - q_1
\]

Component balance for the first reactor:

\[
\frac{d(V_1C_{A,1})}{dt} = q_FC_{A,F} - k_0C_{A,1}V_1 \exp\left(-\frac{E}{RT_1}\right) - q_1C_{A,1}
\]

Energy balance for the first reactor:

\[
\frac{d(V_1T_1)}{dt} = q_FT_F + \frac{\Delta Hk_0C_{A,1}V_1}{\rho c_p} \exp\left(-\frac{E}{RT_1}\right) - q_1T_1
\]

\[-V_1Q_c
\]

Volume balance for the second reactor:

\[
\frac{dV_2}{dt} = q_1 - q_2
\]

Component balance for the second reactor:

\[
\frac{d(V_2C_{A,2})}{dt} = q_1C_{A,1} - k_0C_{A,2}V_2 \exp\left(-\frac{E}{RT_2}\right) - q_2C_{A,2}
\]

Energy balance for the second reactor:

\[
\frac{d(V_2T_2)}{dt} = q_1T + \frac{\Delta Hk_0C_{A,2}V_2}{\rho c_p} \exp\left(-\frac{E}{RT_2}\right) - q_2T_2
\]

Further equations:

\[
q_1 = c_1\sqrt{V_1 - V_2}
\]

\[
q_2 = c_1\sqrt{V_2u_1}
\]

\[
Q_c = C_2u_2
\]

Variable description:

\[
V_1 \quad \text{volume of the liquid in the first reactor}
\]

\[
V_2 \quad \text{volume of the liquid in the second reactor}
\]

\[
C_{A,F} \quad \text{concentration of species A in the feed}
\]

\[
C_{A,1} \quad \text{concentration of species A in the first reactor}
\]

\[
C_{A,2} \quad \text{concentration of species A in the second reactor}
\]

\[
T_F \quad \text{feed temperature}
\]

\[
T_1 \quad \text{temperature of the first reactor}
\]

\[
T_2 \quad \text{temperature of the second reactor}
\]

\[
q_F \quad \text{flow rate}
\]

\[
q_1 \quad \text{flow rate of the first reactor}
\]

\[
q_2 \quad \text{flow rate of the second reactor}
\]

\[
k_0 \quad \text{pre-exponential factor}
\]

\[
E \quad \text{activation energy}
\]

\[
R \quad \text{gas constant}
\]

\[
\rho \quad \text{density of the fluid}
\]

\[
c_p \quad \text{heat capacity of the fluid}
\]

\[
Q_c \quad \text{cooling heat flow}
\]

\[
\Delta H \quad \text{energy of reaction}
\]

\[
u_1, u_2 \quad \text{control inputs}
\]

\[
c_1, c_2 \quad \text{constants}
\]

Appendix B. Equations for Example 3

This example is a binary distillation column with 30 trays. Constant molar overflow is assumed as well as constant relative volatility. The feed is introduced in the middle of the column as a saturated liquid.

Condenser:

\[
\frac{dx_{A,1}}{dt} = \frac{1}{A_{\text{Cond}}} V(y_{A,2} - x_{A,1})
\]

Trays in the rectification section (i = 2, ..., 16):

\[
\frac{dx_{A,i}}{dt} = \frac{1}{A_{\text{Tray}}} [L_i(x_{A,i-1} - x_{A,i}) - V(y_{A,i} - y_{A,i+1})]
\]

Feed tray:
\[
\frac{dx_{A,17}}{dt} = \frac{1}{A_{\text{Tray}}} \left[ Fx_{A,\text{Feed}} + L_1x_{A,16} - L_2x_{A,17} - V(y_{A,17} - y_{A,18}) \right]
\]

Trays in the stripping section \((i = 18, \ldots, 31):\)

\[
\frac{dx_{A,i}}{dt} = \frac{1}{A_{\text{Tray}}} \left[ L_2(x_{A,i-1} - x_{A,i}) - V(y_{A,i} - y_{A,i+1}) \right]
\]

Reboiler:

\[
\frac{dx_{A,32}}{dt} = \frac{1}{A_{\text{Reboiler}}} \left[ L_2x_{A,31} - (F - D)x_{A,32} - Vy_{A,32} \right]
\]

Further equations:

\[
\begin{align*}
V &= L_1 + D \\
L_2 &= F + L_1 \\
RR &= \frac{L_1}{D} \\
z_{A,B} &= \frac{y_A(1 - x_A)}{(1 - y_A)x_A}
\end{align*}
\]

Variable description:

- \(A_{\text{Cond}}\) total molar holdup in the condenser
- \(A_{\text{Tray}}\) total molar holdup on each tray
- \(A_{\text{Reboiler}}\) total molar holdup in the reboiler
- \(F\) feed flowrate
- \(D\) distillate flowrate
- \(L_1\) flowrate of the liquid in the rectification section
- \(L_2\) flowrate of the liquid in the stripping section
- \(V\) vapor flowrate in the column
- \(RR\) reflux ratio
- \(x_{A,i}\) liquid composition of component A on the \(i\)th stage
- \(x_{A,\text{Feed}}\) feed composition of component A
- \(y_{A,i}\) vapor composition of component A on the \(i\)th stage
- \(z_{A,B}\) relative volatility

References


