

Application of Balanced Truncation to Nonlinear Systems

Ivan Dones *, *Sigurd Skogestad* ⁺¹, *Heinz A. Preisig* ⁺

* Hamworthy Combustion Engineering

Via Gramsci 11, I-20851 Lissone (MB), Italy

⁺ Chemical Engineering Department, NTNU

Sem Sælands vei 4, N-7491 Trondheim, Norway

May, 2011

¹ Corresponding author:

Ph.: +47-735-94154

Fax: +47-735-94080

Email: skoge@chemeng.ntnu.no

ABSTRACT

The balanced truncation method for reducing the size of a model was originally developed for linear systems. When extended to nonlinear systems, some considerations need to be faced. First of all, the calculation of the balancing transformation matrix is not unique. This may result in non-physical values for the reconstructed states, which may lead to failure, for example, in thermodynamic routines. To reduce this problem, it is recommended to include all the states in the balancing outputs. To further reduce the effect of nonlinearities in the original model, it is recommended to use a linearising static transformation of the states, if available. In this paper, distillation column models are used as a case study, and in this case a logarithmic transformation of the compositions is beneficial.

KEYWORDS

Nonlinear models, balanced truncation, model reduction, variable transformation

1. INTRODUCTION

As discussed in the literature, model simplification and reduction are important areas in process system engineering, and also in control theory¹⁻³. Simplified and reduced models are used both to gain insight into the process behaviour and to ease the computational efforts of simulation and analysis⁴. For distillation columns, several model reduction and simplification methods have been developed^{3,5-11}. In this paper, the balanced truncation method is considered.

Balanced truncation is a popular model reduction technique that was introduced in the early 1980's¹², and has also been applied to nonlinear distillation column models¹³. The method consists of two steps:

1. Application of a coordinate change (a variable transformation) so that each new state is equally controllable and observable (balanced system);
2. Reduction of the model by truncating the new states that show the weakest input-output dependency.

Alternatively, balanced residualisation can be used in step 2, but in this case the dynamic equations corresponding to the new states with relatively weak input-output behaviour are transformed into algebraic equations. This reduces the number of dynamic states, but it does not reduce the sum of the number of the dynamic and algebraic states. This may not give any computational simplification for nonlinear systems, so balanced residualisation is not considered in this paper.

1.1. Balanced truncation of linear models

Let us first analyse a linear model (for sake of simplicity, a stable linear system with constant coefficients) given in terms of deviation variables from a nominal steady-state:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \\ \mathbf{x}(t) &\in \mathbf{R}^n, \quad \mathbf{u}(t) \in \mathbf{R}^l, \quad \mathbf{y}(t) \in \mathbf{R}^m\end{aligned}\tag{1}$$

The associated controllability and observability Gramians \mathbf{W}_c and \mathbf{W}_o are found by solving the Lyapunov's equations :

$$\begin{aligned}\mathbf{A}\mathbf{W}_c + \mathbf{W}_c\mathbf{A}^T + \mathbf{B}\mathbf{B}^T &= \mathbf{0} \\ \mathbf{A}^T\mathbf{W}_o + \mathbf{W}_o\mathbf{A} + \mathbf{C}^T\mathbf{C} &= \mathbf{0}\end{aligned}\tag{2}$$

A balanced form of the system in (1) is obtained through a transformation matrix \mathbf{T} , see equation (3), such that the resulting Gramians of the transformed system $\bar{\mathbf{W}}_c$ and $\bar{\mathbf{W}}_o$ are equal, and on the simple form given in equation (4),

$$\begin{aligned}
\dot{\mathbf{z}}(t) &= \mathbf{T}\mathbf{A}\mathbf{T}^{-1}\mathbf{z}(t) + \mathbf{T}\mathbf{B}\mathbf{u}(t) \\
\mathbf{y}(t) &= \mathbf{C}\mathbf{T}^{-1}\mathbf{z}(t) + \mathbf{D}\mathbf{u}(t) \\
\mathbf{z}(t) &= \mathbf{T}\mathbf{x}(t), \quad \mathbf{z}(t) \in \mathbf{R}^n, \quad \mathbf{T} \in \mathbf{R}^{n \times n}
\end{aligned} \tag{3}$$

$$\begin{aligned}
\bar{\mathbf{W}}_c = \bar{\mathbf{W}}_o = \boldsymbol{\Sigma} &= \begin{bmatrix} \sigma_1 & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2 & 0 & \cdots & 0 \\ 0 & 0 & \sigma_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma_n \end{bmatrix} \\
\bar{\mathbf{W}}_c &= \mathbf{T}\mathbf{W}_c\mathbf{T}^T \\
\bar{\mathbf{W}}_o &= (\mathbf{T}^{-1})^T \mathbf{W}_o \mathbf{T}^{-1}
\end{aligned} \tag{4}$$

Here σ_i 's are the Hankel singular values, ordered according to:

$$\sigma_1 > \sigma_2 > \sigma_3 > \dots > \sigma_n \geq 0 \tag{5}$$

The input-output behaviour of (3) is identical to the input-output behaviour of (1).

Let

$$\mathbf{z}(t) = \begin{bmatrix} \mathbf{z}_I(t) \\ \mathbf{z}_{II}(t) \end{bmatrix} \tag{6}$$

Then, one approach to model reduction is balanced truncation, where the states $\mathbf{z}_H(t)$ that correspond to small Hankel singular values are deleted.

Considering a case with n states and with $\sigma_1 > \dots > \sigma_{\hat{n}} \gg \sigma_{\hat{n}+1} > \dots > \sigma_n$, one can write the matrices \mathbf{T}_l and \mathbf{T}_r as nonsquare submatrices of \mathbf{T} and \mathbf{T}^{-1} respectively, as:

$$\mathbf{T} = \begin{bmatrix} \overbrace{\begin{matrix} T_{1,1} & \cdots & T_{1,\hat{n}} & \cdots & T_{1,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ T_{\hat{n},1} & \cdots & T_{\hat{n},\hat{n}} & \cdots & T_{\hat{n},n} \end{matrix}}^{\mathbf{T}_l} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ T_{n,1} & \cdots & T_{n,\hat{n}} & \cdots & T_{n,n} \end{bmatrix} \quad (7)$$

$$\mathbf{T}^{-1} = \begin{bmatrix} \overbrace{\begin{matrix} T_{1,1}^i & \cdots & T_{1,\hat{n}}^i \\ \vdots & \ddots & \vdots \\ T_{\hat{n},1}^i & \cdots & T_{\hat{n},\hat{n}}^i \\ \vdots & \ddots & \vdots \\ T_{n,1}^i & \cdots & T_{n,\hat{n}}^i \end{matrix}}^{\mathbf{T}_r} & \cdots & \begin{matrix} T_{1,n}^i \\ \vdots \\ T_{\hat{n},n}^i \\ \vdots \\ T_{n,n}^i \end{matrix} \end{bmatrix} \quad (8)$$

Truncation of the system in (3) gives:

$$\begin{aligned}
\dot{\mathbf{z}}_r(t) &= \mathbf{T}_r \mathbf{A} \mathbf{T}_r \mathbf{z}_r(t) + \mathbf{T}_r \mathbf{B} \mathbf{u}(t) \\
\tilde{\mathbf{y}}(t) &= \mathbf{C} \mathbf{T}_r \mathbf{z}_r(t) + \mathbf{D} \mathbf{u}(t) \\
\mathbf{z}_r(t) &\in \mathbf{R}^{\hat{n}}, \quad \tilde{\mathbf{x}}(t) = \mathbf{T}_r \mathbf{z}_r(t) \in \mathbf{R}^n, \quad \hat{n} < n
\end{aligned} \tag{9}$$

where $\tilde{\mathbf{y}}(t)$ is the vector of the outputs of the reduced-order model and $\tilde{\mathbf{x}}(t)$ are the reconstructed states.

This reduction method results in a good approximation of the original system over the whole frequency range¹³. One disadvantage of this approach is that it does not preserve the steady-state behaviour of the original system and therefore will result in offset¹⁴. The method guarantees preserved stability and comes with an *a priori* error bound¹³⁻²¹.

1.2. Balanced truncation of nonlinear models

When extending the balanced truncation method to nonlinear systems, some considerations must be made^{13, 18, 22-30}.

Similar to equation (1), a nonlinear system can be written in the form (10):

$$\begin{aligned}
\dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \\
\mathbf{y}(t) &= \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)) \\
\mathbf{x}(t) &\in \mathbf{R}^n, \quad \mathbf{u}(t) \in \mathbf{R}^l, \quad \mathbf{y}(t) \in \mathbf{R}^m
\end{aligned} \tag{10}$$

Using the linear approximation to obtain the transformation matrices \mathbf{T}_l and \mathbf{T}_r , see (3)-(8), the balanced truncated form of (10) becomes:

$$\begin{aligned}
 \dot{\mathbf{z}}_l(t) &= \mathbf{T}_l \mathbf{f}(\tilde{\mathbf{x}}(t), \mathbf{u}(t)) \\
 \tilde{\mathbf{y}}(t) &= \mathbf{g}(\tilde{\mathbf{x}}(t), \mathbf{u}(t)) \\
 \tilde{\mathbf{x}}(t) &= \mathbf{T}_r \mathbf{z}_l(t), \quad \mathbf{z}_l(t) \in \mathbf{R}^{\hat{n}}, \quad \tilde{\mathbf{x}}(t) \in \mathbf{R}^n, \quad \hat{n} < n
 \end{aligned} \tag{11}$$

where $\tilde{\mathbf{x}}(t) = \mathbf{T}_r \mathbf{z}_l(t)$ are the reconstructed states used when evaluating the functions \mathbf{f} and \mathbf{g} .

If the nonlinear model in (10) is written in deviation variables, the elimination of the states $\mathbf{z}_r(t)$ of the balanced system will still preserve the matching of the full and the reduced model at the starting steady-state point, as in the linear case.

On the other hand, one may sometimes choose to not write the nonlinear model in deviation variables. This case is studied by Hahn and Edgar¹³ and to match the initial steady-state, the balanced truncated approximation can be written (even without deviation variables) as:

$$\begin{aligned}
\dot{\mathbf{z}}_I(t) &= \mathbf{T}_I \mathbf{f}(\tilde{\mathbf{x}}(t), \mathbf{u}(t)) \\
\mathbf{z}_{II}(t) &= \mathbf{z}_{II}(t_0) \\
\tilde{\mathbf{y}}(t) &= \mathbf{g}(\tilde{\mathbf{x}}(t), \mathbf{u}(t)) \\
\mathbf{z}(t) &= \begin{bmatrix} \mathbf{z}_I(t) \\ \mathbf{z}_{II}(t) \end{bmatrix}, \quad \mathbf{z}(t) \in \mathbf{R}^n, \quad \mathbf{z}_I(t) \in \mathbf{R}^{\hat{n}}, \quad \mathbf{z}_{II}(t) \in \mathbf{R}^{n-\hat{n}}, \quad \hat{n} < n \\
\tilde{\mathbf{x}}(t) &= \mathbf{T}^{-1} \mathbf{z}(t)
\end{aligned} \tag{12}$$

For more details, see the Appendix.

1.3. The issue of the choice of outputs for balancing (reconstruction of the states)

A feature of the balanced truncation method, both for linear and nonlinear models, is that the method focuses on the behaviour between the inputs \mathbf{u} and the outputs \mathbf{y} . The dynamic behaviour of the states $\mathbf{x}(t)$ may be not accurately described and can in some cases assume values completely different from the full model's behaviour. This happens mainly because of the non-unique (and therefore non-exact) reconstruction $\tilde{\mathbf{x}}(t)$ of the original states.

For a numerical example, let use the linear model quoted by Hahn and Edgar¹³, where the reactions $A \rightarrow B \rightarrow C$ takes place in a continuously stirred-tank reactor. The system has one input, three states and one output:

$$\begin{aligned} \dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{bmatrix} &= \begin{bmatrix} -2.0 & 0.0 & 0.0 \\ 1.0 & -1.1 & 0.0 \\ 0.0 & 0.1 & -1.0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} + \begin{bmatrix} 2.0 \\ 0.0 \\ 0.0 \end{bmatrix} u(t) \\ y(t) &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} \end{aligned} \quad (13)$$

The original states $\mathbf{x}(t)$ represent the concentration of the three species, in deviation variables. The inlet concentration of A (x_1) can be manipulated ($u = x_1$) and the concentration of C (x_3) at the outlet can be measured ($y = x_3$).

This simple system is stable, observable and controllable. If just the state x_3 is considered as the output for balancing (which is the optimal in terms of input-output behaviour), the transformation balancing the linear system (13) is as follows:

$$\mathbf{T} = \begin{bmatrix} -0.07729 & -0.1845 & -2.530 \\ 0.08866 & 0.08595 & -2.776 \\ 0.04344 & -0.1528 & 1.343 \end{bmatrix} \quad (14)$$

and the resulting balanced Gramians are

$$\bar{\mathbf{W}}_c = \bar{\mathbf{W}}_o = \boldsymbol{\Sigma} = \begin{bmatrix} 0.05939 & 0.0 & 0.0 \\ 0.0 & 0.01525 & 0.0 \\ 0.0 & 0.0 & 0.001316 \end{bmatrix} \quad (15)$$

It can be seen that the third balanced state of the balanced system contributes much less to the input-output behaviour than the other two, because its Hankel singular value is more than one order of magnitude smaller than the other ones. Eliminating this state, the truncated transformation matrices become:

$$\mathbf{T}_l = \begin{bmatrix} -0.07729 & -0.1845 & -2.530 \\ 0.08866 & 0.08595 & -2.776 \end{bmatrix}; \quad \mathbf{T}_r = \begin{bmatrix} -2.7615 & 5.6741 \\ -2.1436 & 0.0546 \\ -0.1546 & -0.1773 \end{bmatrix} \quad (16)$$

\mathbf{T}_l and \mathbf{T}_r are the topmost rows and leftmost columns of \mathbf{T} and \mathbf{T}^{-1} , respectively, as in equations (7) and (8).

The truncated balanced system is therefore:

$$\begin{aligned} \dot{\mathbf{z}}_l(t) &= \begin{bmatrix} \dot{z}_1(t) \\ \dot{z}_2(t) \end{bmatrix} = \begin{bmatrix} -0.2012 & -0.621 \\ 0.621 & -1.031 \end{bmatrix} \begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix} + \begin{bmatrix} -0.1546 \\ 0.1773 \end{bmatrix} u(t) \\ \tilde{\mathbf{y}}(t) &= \begin{bmatrix} -0.1546 & 0.1773 \end{bmatrix} \begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix} \end{aligned} \quad (17)$$

As predicted by theory, the truncated system preserves well the input-output behaviour of the full model. However, the reconstruction of the states \mathbf{x} can be poor for $t > t_0$. Just as an example, assume that at a certain time t^* the actual concentrations (in deviation variables) are:

$$\mathbf{x}(t^*) = \begin{bmatrix} x_1(t^*) \\ x_2(t^*) \\ x_3(t^*) \end{bmatrix} = \begin{bmatrix} 0.05 \\ -0.25 \\ 0.2 \end{bmatrix} \quad (18)$$

Transforming the vector in equation (18) into the corresponding reduced-order z coordinates gives:

$$\mathbf{z}_l(t^*) = \begin{bmatrix} z_1(t^*) \\ z_2(t^*) \end{bmatrix} = \mathbf{T}_l \begin{bmatrix} x_1(t^*) \\ x_2(t^*) \\ x_3(t^*) \end{bmatrix} = \begin{bmatrix} 0.4637 \\ -0.5722 \end{bmatrix} \quad (19)$$

One loses some information about the states, and, reconstructing the state vector from the truncated model, gives:

$$\tilde{\mathbf{x}}(t^*) = \begin{bmatrix} \tilde{x}_1(t^*) \\ \tilde{x}_2(t^*) \\ \tilde{x}_3(t^*) \end{bmatrix} = \mathbf{T}_r \begin{bmatrix} z_1(t^*) \\ z_2(t^*) \end{bmatrix} = \begin{bmatrix} -1.9666 \\ 0.9631 \\ 0.1732 \end{bmatrix} \quad (20)$$

The vector of the relative errors is:

$$\boldsymbol{\varepsilon}_{rel} = \begin{bmatrix} \left| \frac{x_i(t^*) - \tilde{x}_i(t^*)}{x_i(t^*)} \right| \\ \left| \frac{x_i(t^*) - \tilde{x}_i(t^*)}{x_i(t^*)} \right| \\ \left| \frac{x_i(t^*) - \tilde{x}_i(t^*)}{x_i(t^*)} \right| \end{bmatrix} = \begin{bmatrix} 4033\% \\ 485\% \\ 13\% \end{bmatrix} \quad (21)$$

As expected, the third state is the closest to the original value (13% error), because it was chosen as the output y . But the reconstruction of the two other states is completely wrong.

This problem does not affect the success of the balanced truncated method in linear systems, because the reconstructed states are not needed for the computations. On the other hand in nonlinear systems, the error, for example resulting in a negative concentration, may have catastrophic effects, because the reconstructed states $\tilde{\mathbf{x}}(t)$ are used when evaluating the functions \mathbf{f} and \mathbf{g} .

1.4. Choice of the balancing transformation matrix for nonlinear systems

For a nonlinear system it is not unique how to obtain the transformation matrix \mathbf{T} , its truncation and its inverse.

The simplest approach is to linearise the model at a single nominal steady-state (in this paper called the “simple method”) and to calculate the matrix \mathbf{T} from the linearised model. The advantages of using a single point are the simplicity of the procedure and the fact that the linearised model is guaranteed to be consistent in the linearisation point. As an alternative, Hahn, Edgar and co-workers^{13, 22-24} suggest to calculate the balancing transformation matrix \mathbf{T} through empirical Gramians. Here state data are collected while impulse input signals of relatively large magnitude are injected in different directions. Matlab routines to calculate the empirical Gramians are available on the Internet³¹.

2. IMPROVING THE BALANCED TRUNCATION OF NONLINEAR SYSTEMS

In this section we attempt to provide solutions to some of the problems in extending the balanced truncation method to nonlinear systems.

2.1. Choice of outputs for balancing (reconstruction of the states)

For nonlinear systems, large errors in the state reconstruction may be fatal for the simulation of the reduced-order models, because they may result, for example, in negative compositions, which result in failure in some routines (such as thermodynamic packages).

To reduce this problem, and thus to give more robustness to the truncation, we propose to augment the vector of outputs \mathbf{y} with all (or, generally, a weighted set of all) the integration states \mathbf{x} . Since the balanced truncation method is developed to maintain the input-output behaviour, this will guarantee that the reconstructed states $\tilde{\mathbf{x}}$ are not too different from \mathbf{x} .

For example, consider again the linear continuously stirred-tank reactor model in (13). We use all the states as outputs ($\mathbf{y} = [x_1 \ x_2 \ x_3]^T$), and not just x_3 as in Section 1.3. The transformation balancing the linear system is

$$\mathbf{T} = \begin{bmatrix} 0.6227 & 0.5675 & 0.0473 \\ -0.4663 & 1.0114 & 0.2046 \\ 0.0897 & -0.4635 & 5.5819 \end{bmatrix} \quad (22)$$

and the resulting balanced Gramians are

$$\bar{\mathbf{W}}_c = \bar{\mathbf{W}}_o = \Sigma = \begin{bmatrix} 0.7119 & 0.0 & 0.0 \\ 0.0 & 0.2191 & 0.0 \\ 0.0 & 0.0 & 0.0157 \end{bmatrix} \quad (23)$$

Again, the truncation of the last balanced state is justifiable from (23), and, from (7) and

(8), the matrices \mathbf{T}_l and \mathbf{T}_r are:

$$\mathbf{T}_l = \begin{bmatrix} 0.6227 & 0.5675 & 0.0473 \\ -0.4663 & 1.0114 & 0.2046 \end{bmatrix}; \quad \mathbf{T}_r = \begin{bmatrix} 1.1326 & -0.6294 \\ 0.5172 & 0.6850 \\ 0.0247 & 0.0670 \end{bmatrix} \quad (24)$$

Once again, assume that at a certain time instant t^* the actual concentrations (in deviation variables) are like in (18). From (24), the corresponding reduced states are:

$$\mathbf{z}_l(t^*) = \begin{bmatrix} z_1(t^*) \\ z_2(t^*) \end{bmatrix} = \mathbf{T}_l \begin{bmatrix} x_1(t^*) \\ x_2(t^*) \\ x_3(t^*) \end{bmatrix} = \mathbf{T}_l \begin{bmatrix} 0.05 \\ -0.25 \\ 0.2 \end{bmatrix} = \begin{bmatrix} -0.1013 \\ -0.2352 \end{bmatrix} \quad (25)$$

Reconstructing from (26) the original states gives:

$$\tilde{\mathbf{x}}(t^*) = \begin{bmatrix} \tilde{x}_1(t^*) \\ \tilde{x}_2(t^*) \\ \tilde{x}_3(t^*) \end{bmatrix} = \mathbf{T}_r \begin{bmatrix} z_1(t^*) \\ z_2(t^*) \end{bmatrix} = \begin{bmatrix} 0.0333 \\ -0.2135 \\ -0.0183 \end{bmatrix} \quad (26)$$

The relative error between the original and reconstructed states is:

$$\boldsymbol{\varepsilon}_{rel} = \left| \frac{x_i(t^*) - \tilde{x}_i(t^*)}{x_i(t^*)} \right| = \begin{bmatrix} 33\% \\ 15\% \\ 109\% \end{bmatrix} \quad (27)$$

The largest errors are much smaller than in (21). However, the third reconstructed state \tilde{x}_3 now has a relative larger deviation. To improve on this, one may introduce a weighting matrix, $\mathbf{y} = \mathbf{C}\mathbf{x}$, where

$$\mathbf{C} = \begin{bmatrix} \omega_1 & 0 & \cdots & 0 \\ 0 & \omega_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \omega_n \end{bmatrix} \quad (28)$$

The ω 's are tuning factors, which give more or less weight to a state.

2.2. Reducing nonlinearities

To improve the robustness of the balanced truncation of nonlinear systems, one should reduce the nonlinearities of the model by using static variable transformations in order to change the state space before applying the model reduction. This is often feasible because the dominant nonlinearities are often at steady-state.

Below, this is illustrated on the distillation case study, where the logarithmic transformation has been used to linearise the behaviour.

3. THE CASE STUDY

To study the extension of balanced truncation to nonlinear systems, we analyse a simple nonlinear distillation model, namely Skogestad's "Column A" ³². A simple Matlab code of Column A is available on the Internet ³³. The main assumptions for the model are:

- binary mixture;
- constant pressure;
- constant relative volatility;
- equilibrium on all stages;
- total condenser;
- constant molar flows;
- no vapour holdup;
- linearised liquid dynamics.

These assumptions and simplifications may seem restrictive, but they capture the main effects important for dynamics and control.

The column has 40 theoretical stages plus a total condenser and separates a binary mixture with relative volatility of $\alpha = 1.5$ into products of 99% purity.

The dynamic total and light component material balances are, for a generic stage i (counting from the bottom and up) ²:

$$\frac{dM_i}{dt} = \underbrace{L_{i+1} - L_i + V_{i-1} - V_i}_{f_{M_i}} \quad (29)$$

$$\frac{d(M_i x_i)}{dt} = \underbrace{L_{i+1} x_{i+1} - L_i x_i + V_{i-1} y_{i-1} - V_i y_i}_{f_{x_i}} \quad (30)$$

where from the vapour-liquid equilibrium

$$y_i = \frac{\alpha x_i}{1 + (\alpha - 1)x_i} \quad (31)$$

and with the assumption of linearised liquid flow dynamics

$$L_i(t) = L_i^0 + k_i (M_i(t) - M_i^0) \quad (32)$$

² Here x_i is the mole fraction of light component in the liquid phase, and y_i in the gas phase

The vapour flows are assumed constant up the column ($V_i = V_{i+1}$).

For the reboiler, condenser and feed tray the balances are slightly different, but still show the same characteristics as (29) and (30). The total number of dynamic states of this ODE (Ordinary Differential Equations) model is 82.

The column levels are stabilised using the LV control structure³⁴⁻³⁶ with proportional controllers for the reboiler and condenser levels.

Writing the model as in equation (10) and in terms of input-output representation, the states \mathbf{x} are the mass holdups and the liquid compositions of each distillation stage. The two process control outputs of the systems are the top vapour and bottom liquid compositions, $y_D = x_{41}$ and $x_B = x_1$. The vector of independent variables u is composed of reflux stream L , boilup V , feed rate F and feed composition z_F (the last two are actually disturbances).

4. SIMULATION RESULTS

In this section we report the results obtained on “Column A” simulated in open-loop mode with level control included and with boilup V and reflux L as inputs.

The model is not written in deviation variables; so to avoid the offset at time t_0 between the full model and the balanced truncated models, Hahn’s and Edgar’s suggestions reported in equation (12) are used.

In the simulations, we are mainly interested in the dynamic response of the top and bottom compositions (y_D and x_B). The low-order balanced truncated model is in the form (12).

The 82 reconstructed states $\tilde{\mathbf{x}} = [\tilde{x}_1, \dots, \tilde{x}_{41}, \tilde{M}_1, \dots, \tilde{M}_{41}]^T$ are used to evaluate the functions in (29)-(30). The main nonlinearities come from the vapour-liquid equilibrium (31).

At time $t_0 = 100$ minutes, a step input change (10% disturbance respect the nominal value) is applied in the feed rate, with all the other input variables kept constant (open-loop operation). The CPU times to simulate the reduced models between 100 and 1300 minutes are recorded. The computer used is a laptop Intel[®] Core[™] 2 Duo CPU, 2.00 GHz, 2.00 GB of RAM.

4.1. Cases studied

The distillation case study is used to evaluate the following options that can be chosen when applying balanced truncation to nonlinear systems.

1. Choice of method for the calculation of the transformation matrix \mathbf{T} :
 - a. “Simple method” (linearisation in the nominal point);
 - b. Empirical Gramians³¹.
2. Choice of outputs for balancing:
 - a. Only the control outputs (y_D and x_B);
 - b. All the states \mathbf{x} , equally weighted;

- c. All the states \mathbf{x} , with y_D and x_B weighted 100 times more than the other states, see (28).
3. Use of static state transformations:
- a. No transformations;
 - b. Logarithmic transformation for all liquid compositions (half of the state vector), to make the response of distillation models more linear^{32, 37-38}. On stage i

$$X_i = \ln \frac{x_{L,i}}{x_{H,i}} \quad (33)$$

In (33), the subscripts L and H denote the light and heavy components, respectively. This transformation linearises the steady-state and dynamic responses of the column³⁹. Since we are dealing with a binary mixture, (33) simplifies to:

$$X_i = \ln \frac{x_i}{1-x_i} \quad (34)$$

where x_i is the liquid concentration (mole fraction) of the light component.

The backwards transformation is

$$x_i = \frac{e^{x_i}}{1 + e^{x_i}} \quad (35)$$

The logarithmic compositions guarantee that the actual compositions (mole fractions) are never negative. The transformation back from and to logarithmic compositions must be made each time we evaluate the functions f_{M_i} and f_{x_i} , and this will increase the computation time.

4. Choice of number of states in the truncated system.

4.2. Results

The main results are summarised in Table 1. The methods of “simple” transformation based on linearising in a single point is compared with the “empirical Gramians” based on the average Gramians found over several operating points¹³. For each of these we compare transformations using only the outputs, all the states (unweighted) and all the states with the outputs weighted extra. In addition we consider with and without using a nonlinear output transformation (“log compositions”) given a total of 12 cases. In order to ease the discussion, the table is limited to one set of truncated (reduced-order) models, namely those with 9 states out of 82.

Table 1. Summary of results for reduced distillation models with 9 states.

CASE	Method for obtaining T	Outputs for balancing	Output transformation	Results for 9 states			
				Relative error in responses [%]			CPU time [ms]
				y_D	x_B	sum	
full model							183 (82 states)
case 1	simple	y_D and x_B	none	<i>fails with less than 72 states</i>			
case 2	simple	all unweighted states,	none	0.71	19.85	20.56	46
case 3	simple	all states, weighted y_D and x_B	none	18.15	18.98	37.12	46
case 4	simple	y_D and x_B	log compositions	12.38	19.37	31.75	76
case 5	simple	all unweighted states,	log compositions	9.61	12.86	22.47	94
case 6	simple	all states, weighted y_D and x_B	log compositions	12.49	19.46	31.95	77
case 7	empirical Gramians	y_D and x_B	none	<i>fails with less than 15 states</i>			
case 8	empirical Gramians	all unweighted states,	none	8.70	1.34	10.04	28
case 9	empirical Gramians	all states, weighted y_D and x_B	none	0.29	2.20	2.49	24
case 10	empirical Gramians	y_D and x_B	log compositions	2.62	1.05	3.67	37
case 11	empirical Gramians	all unweighted states,	log compositions	7.02	8.10	15.11	37
case 12	empirical Gramians	all states, weighted y_D and x_B	log compositions	0.69	0.40	1.09	33

The different cases in Table 1 are here compared and evaluated in terms of CPU time, robustness and precision to reproduce the full model. The parameter to evaluate the precision is \mathcal{E}_{rel}^{sum} . The relative errors are defined as:

$$\mathcal{E}_{rel}^{y_D} = \left| \frac{y_D^{red}(t_\infty) - y_D^{full}(t_\infty)}{y_D^{full}(t_\infty) - y_D^{full}(t_0)} \right| \quad (36)$$

$$\mathcal{E}_{rel}^{x_B} = \left| \frac{x_B^{red}(t_\infty) - x_B^{full}(t_\infty)}{x_B^{full}(t_\infty) - x_B^{full}(t_0)} \right| \quad (37)$$

$$\mathcal{E}_{rel}^{sum} = \mathcal{E}_{rel}^{y_D} + \mathcal{E}_{rel}^{x_B} \quad (38)$$

Before discussing the results, let us briefly consider the distribution of the Hankel singular values.

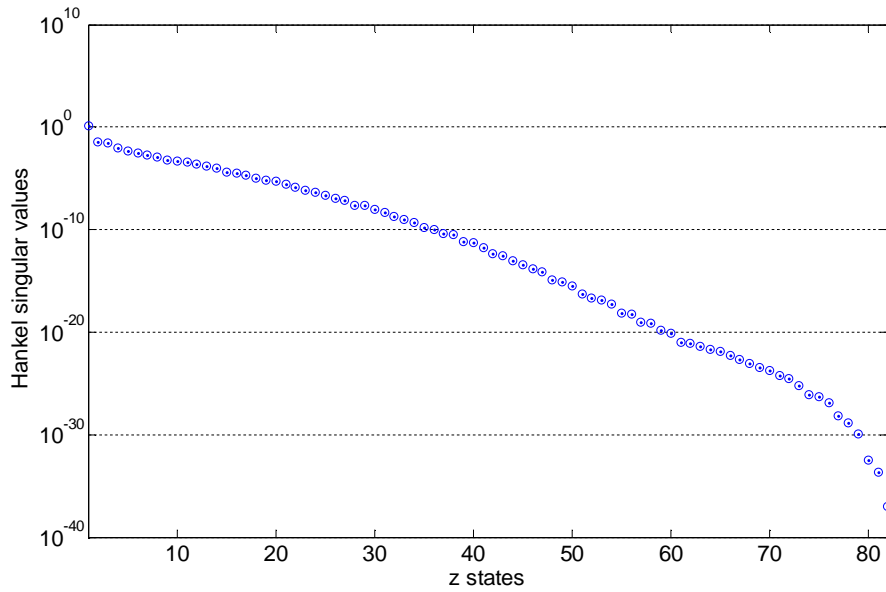


Figure 1. Hankel singular values with only y_D and x_B as balancing outputs (case 1).

In Figure 1 are shown the Hankel singular values of the system balanced using the nominal linear model (“simple method”), and considering only y_D and x_B as balancing outputs (case 1). The Hankel singular values are very different, varying from about 10^{-38} to 1, which offers a large potential for truncating the system. However, as mentioned above, using only y_D and x_B as balancing outputs may lead to non-physical reconstructed states, and to avoid this problem one may use all the states as balancing outputs. In Figure 2 are shown the Hankel singular values for this case (case 2).

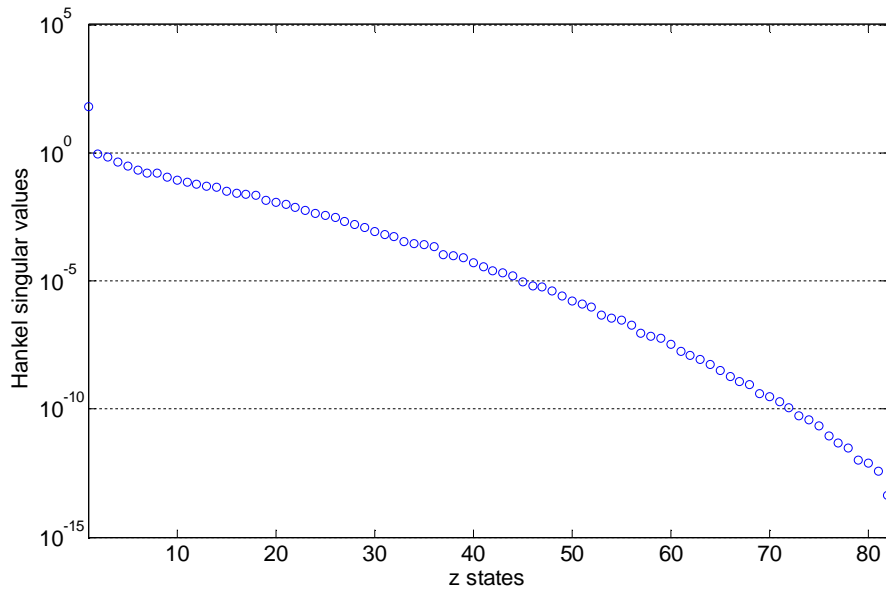


Figure 2. Hankel singular values with all the states as balancing outputs (case 2).

The spread of the Hankel singular values is much less in this case, which is expected. However, the spread is still very large (10^{-13} to 100), and thus a large potential for model reduction is offered.

5. DISCUSSION OF THE RESULTS

5.1. Choice of method for the calculation of the transformation matrix

Compared to the empirical Gramians, the linearisation in the nominal point (“simple method”) has the advantage of simpler computations of the transformation matrix \mathbf{T} . However, the use of the empirical Gramians was found to be better in most cases. First

consider the case with only the control outputs (y_D and x_B) as balancing outputs. The “simple method” fails when reducing to 71 states or less (case 1), whereas the use of the empirical Gramians is more robust and fails with 14 states or less (case 7).

Next, consider the results with all states as balancing outputs and/or logarithmic compositions. When reduced to 9 states we see from Table 1 that the empirical Gramians give a smaller error and a shorter CPU time for the simulation (compare cases 2-6 with cases 8-12). The recommendation is therefore to use the empirical Gramians for the calculation of the transformation matrix \mathbf{T} (*recommendation 1*).

5.2. Choice of outputs for balancing

The data in Table 1 show that including all the states as balancing outputs makes the model reduction more robust. Truncations that are not possible with only y_D and x_B as outputs (case 1, for example), are feasible with all the states included in the output vector (case 2). On the other hand, the critical states (the control outputs y_D and x_B) may lose precision (compare cases 10 and 11). The weighting factors for y_D and x_B can be tuned to give smaller relative errors in y_D and x_B ; for example, a factor 100 is helpful when the empirical Gramians are used (cases 9 and 12). If the “simple method” is adopted, then a more careful tuning of the weighting factors must be made to improve the responses of the reduced models. For example, if we change in case 3 the weighting factor for y_D from 100 to 10, then the sum of the relative errors is reduced from 37.1% to 17.5%. The tuning of

the weighting factors can be made more systematic by an optimisation procedure (not considered in this paper).

The computation time appears relatively independent on the choice of the balancing outputs.

In conclusion, the recommendation is to include all the states in the balancing outputs (*recommendation 2*). This may sacrifice the precision in some cases, but will give more robustness to the model reduction, which is a key requirement.

5.3. Use of static state transformation

Reducing the nonlinearities of the model with a static state transformation has generally a beneficial effect on the balanced truncation method, since truncations that are not feasible (cases 1 and 7) become feasible with logarithmic compositions (cases 4 and 10). Even though there is not necessarily a gain in precision, the procedure is more robust with the logarithmic compositions.

Since robustness is very important, our recommendation (*recommendation 3*) is therefore to minimise the nonlinearities of the model by use of a static state transformation (when it is possible).

5.4. Truncation limits

In Table 1, all results are for truncated models with 9 states out of 82. In Table 2 it is shown the truncation limit for some cases which performed well with 9 states. The truncation limit is the fewest number of states that gives a solution that converges, and seems to vary from 5 (case 9) to 3 (cases 10 and 12).

Table 2. Truncation limits.

CASE	Method for obtaining T	Outputs for balancing	Output transfor.	number of states	Relative error [%]			CPU time [ms]
					y_D	x_B	sum	
case 9	empirical Gramians	all states, weighted y_D and x_B	none	5	23.75	8.91	32.66	32
case 10	empirical Gramians	y_D and x_B	log comp.	4	0.53	12.39	12.92	31
				3	11.69	41.82	53.51	24
case 12	empirical Gramians	all states, weighted y_D and x_B	log comp.	4	1.64	13.02	14.66	36
				3	11.67	41.81	53.48	28

When logarithmic transformation is applied to the compositions (cases 10 and 12), it is found that distillation model with only 4 states still give good accuracy. For case 9, the integration with only 4 states fails; therefore for this case the truncation limit is 5 states. Dynamic simulations are shown in Figure 3.

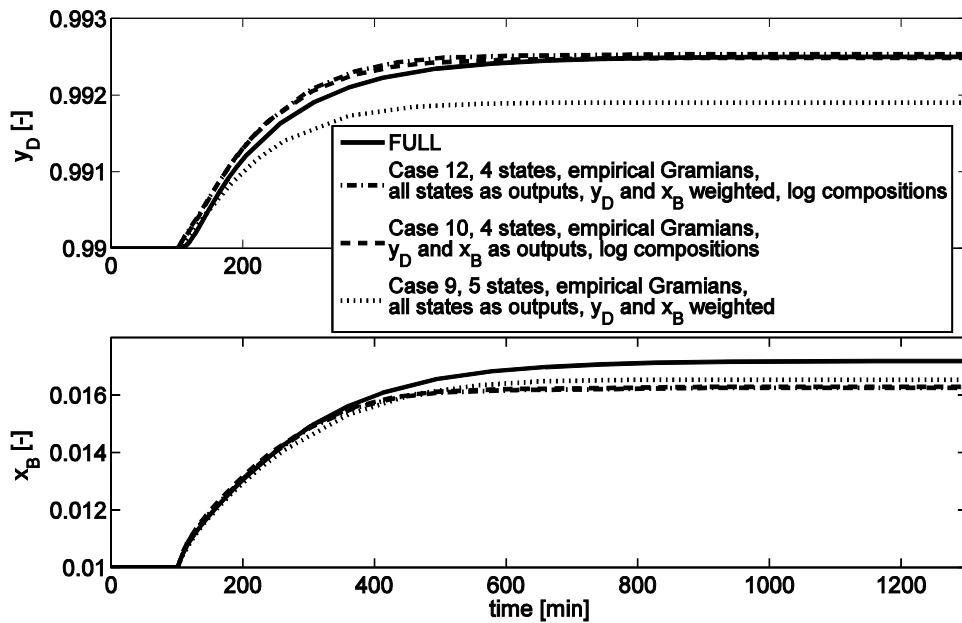


Figure 3. Simulation of case 9 with 5 states and case 10 and 12 with 4 states (the latter two have very similar responses).

Actually, for cases 10 and 12, models with only 3 states can be simulated (Figure 4), but the precision is questionable, and in particular the initial part of the response is poor. Thus, we recommend to use 4 states for cases 10 and 12, in order to keep the dynamics closer to those of the full model.

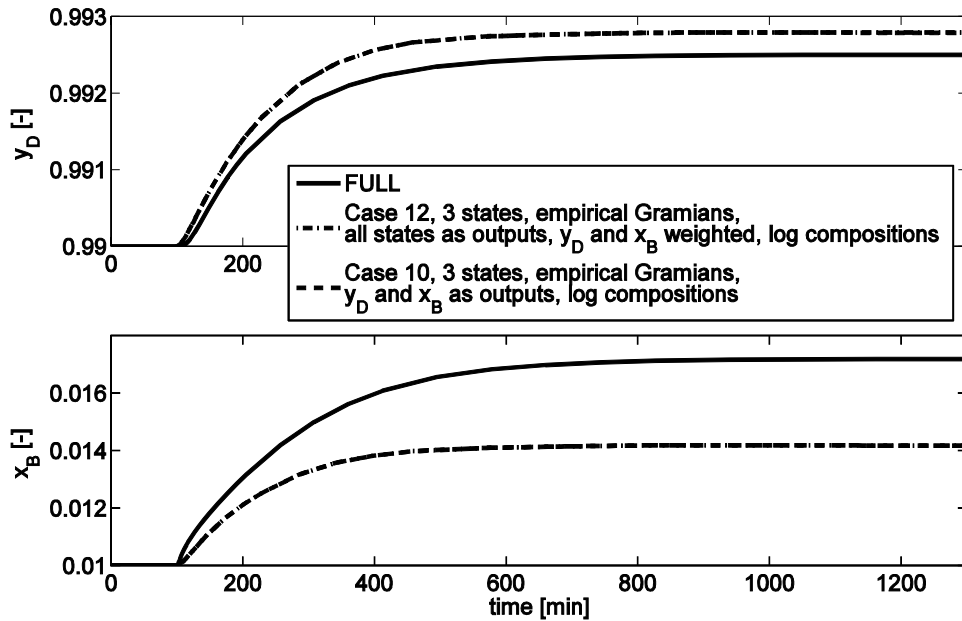


Figure 4. Cases 10 and 12 with only 3 states (very similar responses).

5.5. Remarks about the computation time

Comparing the reduced models with the full model, the full model has a higher computation time than the truncated (reduced) models, and the difference is typically a factor 5. About the computation times, some additional words can be said. A full distillation model has a well-defined block-diagonal structure⁴⁰⁻⁴⁴. On the other hand, whereas balancing and truncating reduces the number of dynamic states, it spoils some of the block-diagonal structure. As consequence, the computation time may actually increase in some cases⁴⁵⁻⁴⁶, and the model size is so not necessary related to the CPU time. For example, case 9 with 9 states (24 ms) is faster than with 5 states (32 ms).

6. CONCLUSIONS

The balanced truncation method was originally developed for linear systems. When applied to nonlinear systems, based on the distillation case study, we have the following recommendations:

- *Recommendation 1:* When calculating the balancing transformation matrix \mathbf{T} , use empirical Gramians and not the “simple method” with linearisation in the nominal points.
- *Recommendation 2:* Include all the states in the output vector to avoid non-physical values for reconstructed states used, for example, in thermodynamic calculations.. Weighting factors may be used to give more emphasis to particular states.
- *Recommendation 3:* To minimise nonlinearities a static state transformation should be used when appropriate. For the distillation model a logarithmic transformation on the compositions is recommended, in particular, to improve the robustness.

7. APPENDIX

The methodology suggested by Hahn and reported in equation (12) guarantees that $\tilde{\mathbf{x}}(t_0) = \mathbf{x}(t_0)$ for a generic nonlinear model, even without deviation variables.

Hahn’s and Edgar’s contribution¹³ was originally developed to avoid the inverse response in the truncated models. The inverse response occurs because the system’s behaviour at the start of the simulation is mainly influenced by the change of the system due to truncation method itself, and not by a change in the inputs. For nonlinear systems the steady-state

values are usually different from zero and neglecting the steady-state values of these terms may result in inverse response.

Alternatively to equation (12), one can adopt a vector of shifting addend $\Delta \mathbf{x}_{rec}$ able to nullify the mismatch at the starting point of the reduced model. What one can do is to calculate $\Delta \mathbf{x}_{rec}$ at the starting point of the implementation of the truncation method, in order to correct the reconstructed states, and use it all along the simulation with the reduced-order model.

Equation (39) summarises the approach.

$$\begin{aligned}
 \dot{\mathbf{z}}_I(t) &= \mathbf{T}_I \mathbf{f}(\tilde{\mathbf{x}}(t), \mathbf{u}(t)) \\
 \tilde{\mathbf{y}}(t) &= \mathbf{g}(\tilde{\mathbf{x}}(t), \mathbf{u}(t)) \\
 \Delta \mathbf{x}_{rec} &= \mathbf{x}(t_0) - \mathbf{T}_r \mathbf{z}_I(t_0) \\
 \tilde{\mathbf{x}}(t) &= \mathbf{T}_r \mathbf{z}_I(t) + \Delta \mathbf{x}_{rec} \\
 \mathbf{z}_I(t) &\in \mathbf{R}^{\hat{n}}, \quad \tilde{\mathbf{x}}(t) \in \mathbf{R}^n, \quad \hat{n} < n
 \end{aligned} \tag{39}$$

This alternative approach is equivalent to Hahn's suggestions reported in equation (12), since reconstructing the states as

$$\tilde{\mathbf{x}}(t) = \mathbf{T}^{-1} \begin{bmatrix} \mathbf{z}_I(t) \\ \mathbf{z}_{II}(t_0) \end{bmatrix} \tag{40}$$

is analytically the same as

$$\tilde{\mathbf{x}}(t) = \mathbf{T}_r \mathbf{z}_I(t) + \Delta \mathbf{x}_{rec} \quad (41)$$

being \mathbf{z}_I the balanced states important for the input-output behaviour, while \mathbf{z}_{II} the truncated ones.

Using Hahn's nomenclature, $\mathbf{P} = [\mathbf{I} \quad \mathbf{0}]$ is the projection matrix which has the rank of the reduced system, and the matrices \mathbf{T}_I and \mathbf{T}_r can be defined as follow:

$$\mathbf{T}_I = \mathbf{P}\mathbf{T} \quad (42)$$

$$\mathbf{T}_r = \mathbf{T}^{-1}\mathbf{P}^T \quad (43)$$

The proof of the equivalence between equations (40) and (41) is here described:

$$\mathbf{T}_r \mathbf{z}_I(t) + \Delta \mathbf{x}_{rec} = \mathbf{T}^{-1} \begin{bmatrix} \mathbf{z}_I(t) \\ \mathbf{z}_{II}(t_0) \end{bmatrix} \quad (44)$$

Substituting equation (43) we obtain:

$$\mathbf{T}^{-1}\mathbf{P}^T\mathbf{z}_I(t) + \Delta\mathbf{x}_{rec} = \mathbf{T}^{-1} \begin{bmatrix} \mathbf{z}_I(t) \\ \mathbf{z}_{II}(t_0) \end{bmatrix} \quad (45)$$

From equation (39), the definition of $\Delta\mathbf{x}_{rec}$ is substituted into equation (45):

$$\mathbf{T}^{-1}\mathbf{P}^T\mathbf{z}_I(t) + \mathbf{x}(t_0) - \mathbf{T}^{-1}\mathbf{P}^T\mathbf{z}_I(t_0) = \mathbf{T}^{-1} \begin{bmatrix} \mathbf{z}_I(t) \\ \mathbf{z}_{II}(t_0) \end{bmatrix} \quad (46)$$

With

$$\mathbf{x}(t_0) = \mathbf{T}^{-1} \begin{bmatrix} \mathbf{z}_I(t_0) \\ \mathbf{z}_{II}(t_0) \end{bmatrix} \quad (47)$$

equation (46) becomes:

$$\mathbf{T}^{-1}\mathbf{P}^T\mathbf{z}_I(t) + \mathbf{T}^{-1} \begin{bmatrix} \mathbf{z}_I(t_0) \\ \mathbf{z}_{II}(t_0) \end{bmatrix} - \mathbf{T}^{-1}\mathbf{P}^T\mathbf{z}_I(t_0) = \mathbf{T}^{-1} \begin{bmatrix} \mathbf{z}_I(t) \\ \mathbf{z}_{II}(t_0) \end{bmatrix} \quad (48)$$

Equation (48) finally reduces into an identity, since simplifications lead to:

$$\begin{aligned}\mathbf{P}^T \mathbf{z}_I(t) - \mathbf{P}^T \mathbf{z}_I(t_0) &= \begin{bmatrix} \mathbf{z}_I(t) \\ \mathbf{z}_{II}(t_0) \end{bmatrix} - \begin{bmatrix} \mathbf{z}_I(t_0) \\ \mathbf{z}_{II}(t_0) \end{bmatrix} \\ \mathbf{P}^T \mathbf{z}_I(t) - \mathbf{P}^T \mathbf{z}_I(t_0) &= \begin{bmatrix} \mathbf{z}_I(t) - \mathbf{z}_I(t_0) \\ 0 \end{bmatrix}\end{aligned}\tag{49}$$

Which is what was wanted to be demonstrated.

8. ACKNOWLEDGMENTS

StatoilHydro is acknowledged for funding. We also thank Andreas Linhart for helpful discussions.

9. NOMENCLATURE AND LIST OF SYMBOLS

x integration states, or concentrations

t time

y outputs

\mathbf{u}	inputs
\mathbf{W}_c	controllability Gramian
\mathbf{W}_o	observability Gramian
\mathbf{T}	transformation matrix
\mathbf{P}	projection matrix
\mathbf{z}	balanced integration states
\mathbf{z}_I	balanced integration states important for the input-output relation
\mathbf{z}_{II}	balanced integration states not important for the input-output relation
$\tilde{\mathbf{x}}$	reconstructed states
$\tilde{\mathbf{y}}$	outputs of the truncated model
X	logarithmic compositions
L	down-going liquid stream
V	up-going vapour stream
M	mass

10. REFERENCES

1. Allgöwer, F. & Zheng, A. (2000). Nonlinear model predictive control. Progress in system theory, Basel, Birkhäuser
2. Dones, I. & Preisig, H.A. (2009). Model simplification and time-scale assumptions applied to distillation modelling. Computers & Chemical Engineering to appear, DOI: 10.1016/j.compchemeng.2009.11.002
3. Marquardt, W. (2001). Nonlinear model reduction for optimization based control of transient chemical processes. Chemical Process Control VI, Tucson, Arizona (USA)
4. Taylor, S.R., Doyle 3rd, F.J. & Petzold, L.R. (2008). Oscillator Model Reduction Preserving the Phase Response: Application to the Circadian Clock. Biophysical Journal 95, 1658-1673
5. Benallou, A., Seborg, D.E. & Mellichamp, D.A. (1986). Dynamic Compartmental-Models for Separation Processes. Aiche Journal 32, 1067-1078
6. Cho, Y.S. & Joseph, B. (1983). Reduced-Order Steady-State and Dynamic-Models for Separation Processes .1. Development of the Model-Reduction Procedure. Aiche Journal 29, 261-269
7. Khowinij, S., Henson, M.A., Belanger, P. & Megan, L. (2005). Dynamic compartmental modeling of nitrogen purification columns. Separation and Purification Technology 46, 95-109

8. Kienle, A. (2000). Low-order dynamic models for ideal multicomponent distillation processes using nonlinear wave propagation theory. *Chemical Engineering Science* 55, 1817-1828
9. Kumar, A. & Daoutidis, P. (2003). Nonlinear model reduction and control for high-purity distillation columns. *Industrial & Engineering Chemistry Research* 42, 4495-4505
10. Linhart, A. & Skogestad, S. (2009). Computational performance of aggregated distillation models. *Computers & Chemical Engineering* 33, 296-308
11. Marquardt, W. (1990). Traveling waves in chemical processes. *International Chemical Engineering* 30, 585
12. Moore, B. (1981). Principal component analysis in linear systems: Controllability, observability, and model reduction. *Automatic Control, IEEE Transactions on* 26, 17-32
13. Hahn, J. & Edgar, T.F. (2002). An improved method for nonlinear model reduction using balancing of empirical Gramians. *Computers & Chemical Engineering* 26, 1379-1397
14. Skogestad, S. & Postlethwaite, I. (2001). *Multivariable feedback control*, Wiley.
15. Enns, D. (1984). *Model Reduction for Control System Design*. PhD thesis, Stanford University, Stanford, California, USA.

16. Enns, D. (1984). Model reduction with balanced realizations: an error bound and a frequency weighted generalization, Proc. Conf. Dec. Contr., Las Vegas, Nevada, USA.
17. Glover, K. (1984). All optimal Hankel-norm approximations for linear multivariable systems and their L-infinity-error bounds. *International Journal of Control* 39, 1115-1193.
18. Nilsson, O. (2009). On modeling and nonlinear model reduction in automotive systems. PhD thesis, Lund University, Lund, Sweden.
19. Pernebo, L. & Silverman, L. (1982). Model reduction via balanced state space representations. *Automatic Control, IEEE Transactions on* 27, 382-387.
20. Zhou, K. (1995). Frequency-weighted L-infinity norm and optimal Hankel norm model reduction. *Automatic Control, IEEE Transactions on* 40, 1687-1699.
21. Zhou, K. & Doyle, J.C. (1998). *Essentials of robust control*, Prentice Hall.
22. Hahn, J. & Edgar, T.F. (2001). A Gramian Based Approach to Nonlinearity Quantification and Model Classification. *Industrial & Engineering Chemistry Research* 40, 5724-5731.
23. Hahn, J. & Edgar, T.F. (2002). Balancing Approach to Minimal Realization and Model Reduction of Stable Nonlinear Systems. *Industrial & Engineering Chemistry Research* 41, 2204-2212.

24. Hahn, J., Edgar, T.F. & Marquardt, W. (2003). Controllability and observability covariance matrices for the analysis and order reduction of stable nonlinear systems. *Journal of Process Control* 13, 115-127.
25. Lall, S. (2002). A subspace approach to balanced truncation for model reduction of nonlinear control systems. *International journal of robust and nonlinear control* 12, 519.
26. Rewienski, M.J. (2003). A trajectory piecewise-linear approach to model order reduction of nonlinear dynamical systems. PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA.
27. Scherpen, J.M.A. (1993). Balancing for nonlinear systems. *Systems & Control Letters* 21, 143-153.
28. Scherpen, J.M.A. & Fujimoto, K. (2003). Nonlinear balanced realization based on singular value analysis of Hankel operators, *Proceedings of the 42nd IEEE Conference on Decision & Control*, Maui, USA.
29. Scherpen, J.M.A. & Fujimoto, K. (2004). Balancing and model reduction for discrete-time nonlinear systems based on Hankel singular value analysis, *Proceedings of the 16th International Symposium on Mathematical Theory of Networks and Systems*. Leuven, Belgium.
30. Vasilyev, D., Rewienski, M.J. & White, J. (2006). Macromodel generation for biomems components using a stabilized balanced truncation plus trajectory

- piecewise-linear approach. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 25.
31. Hahn, J. (2009). Nonlinear model reduction routines for MATLAB, Texas A&M University,
http://cheweb.tamu.edu/orgs/groups/hahn/Model_Reduction/index.html.
 32. Skogestad, S. & Morari, M. (1988). Understanding the Dynamic Behavior of Distillation-Columns. Industrial & Engineering Chemistry Research 27, 1848-1862.
 33. Skogestad, S. (2009). MATLAB Distillation column model, Norwegian University of Science and Technology, <http://www.nt.ntnu.no/users/skoge/distillation>.
 34. Skogestad, S. (1992). Dynamics and control of distillation columns - A critical survey, IFAC Symposium, Maryland.
 35. Skogestad, S. (1997). Dynamics and control of distillation columns: a tutorial introduction. Trans IChemE 75, 539-562.
 36. Skogestad, S. (2007). The dos and don'ts of distillation column control. Chemical Engineering Research & Design 85, 13-23.
 37. Joseph, B. & Brosilow, C.B. (1976). Multi-temps give better control, Hydrocarbon Processing.
 38. Shinskey, F.G. (1977). Distillation control, McGraw-Hill.

39. Mejdell, T. & Skogestad, S. (1991). Estimation of distillation composition from multiple temperature measurements using PLS regression. *Industrial & Engineering Chemistry Research* 30, 2543-2555.
40. Gani, R., Ruiz, C.A. & Cameron, I.T. (1986). A generalized model for distillation columns--I : Model description and applications. *Computers & Chemical Engineering* 10, 181-198.
41. Krishnamurthy, R. & Taylor, R. (1985). Nonequilibrium stage model of multicomponent separation processes. *AIChE Journal* 31, 1973-1985.
42. Luyben, W.L. (1996). *Process Modeling, simulation and control for chemical engineers*. 2nd edition, McGraw-Hill, Chemical Engineering Series.
43. Taylor, R., Kooijman, H.A. & Hung, J.S. (1994). A second generation nonequilibrium model for computer simulation of multicomponent separation processes. *Computers & Chemical Engineering* 18, 205-217.
44. Venkataraman, S. & Lucia, A. (1988). Solving distillation problems by Newton-like methods. *Computers & Chemical Engineering* 12, 55-69.
45. Buzzi-Ferraris, G. (1998). *Metodi numerici e software in C++*, Addison Wesley Longman Italia Editoriale s.r.l.
46. Manenti, F., Dones, I., Buzzi Ferraris, G. & Preisig, H.A. (2009). Efficient Numerical Solver for Partially Structured DAE Systems. *Industrial & Engineering Chemistry Research* 48, 9979-9984.