Multiple nonlinear parameter estimation using PI feedback control

P.F. van Lith*, H. Witteveen, B.H.L. Betlem, B. Roffel

Department of Chemical Technology, University of Twente, P.O. Box 217, 7500 AE Enschede, Netherlands

Received 13 December 1999; accepted 8 September 2000

Abstract

Nonlinear parameters often need to be estimated during the building of chemical process models. To accomplish this, many techniques are available. This paper discusses an alternative view to parameter estimation, where the concept of PI feedback control is used to estimate model parameters. The approach can be used for multiple parameter estimation problems if the parameters can be determined independently. In addition, similarities with Kalman filtering are illustrated. Both approaches are applied and compared in multiple parameter estimation problems. Furthermore, the PI controller approach is applied in parameter estimation for an experimental batch distillation setup. It is shown that the PI controller approach performs comparably to Kalman filtering, but is much easier to set up.

Keywords: Feedback control; Parameter estimation; Kalman filter

1. Introduction

The estimation of nonlinear parameters is often an important step in building models for chemical processes. Over the years, much research has been done and numerous papers and text books have appeared on the subject (see for example well known text books such as Eykhoff (1974); Luyben (1990); Ramirez (1994); Seinfeld & Lapidus (1974)). This paper will discuss an alternative view to parameter estimation by using the concept of a PI feedback controller. This view provides an elegant and simple way to estimate parameters. It is well suited to multiple parameter estimation problems, provided that these parameters can be determined independently, i.e. when there is little interaction between these parameters in the process model. Furthermore, the approach can also be used under dynamical circumstances. The approach will be compared to conventional Kalman filtering, with which it has some similarities.

To illustrate the technique, two test cases are selected that represent a large class of systems. In these systems, the state equations are coupled (directly or indirectly) through time varying model parameters. For each of the two cases, the controller approach will be compared to Kalman filtering. Furthermore, the approach will be illustrated using a pilot-plant batch distillation-column setup.

The paper will start with a discussion of the theoretical background on the feedback controller approach and will compare it with Kalman filtering. Subsequently, the two test cases and the practical application will be presented.

2. Nonlinear parameter estimation

In chemical engineering, mathematical models of the systems under study often have the form of nonlinear ordinary differential equations (ODEs), supplemented with nonlinear algebraic equations describing system parameters. Thus they are of the following form:

\[
\frac{dx}{dt} = f(x, u, \theta),
\]

(1)

\[
\theta = f_\theta(x, u),
\]

(2)

\[
y = Hx
\]

(3)

with \(x\) the system state vector, \(u\) the system input vector, \(\theta\) a nonlinear parameter vector, \(H\) the measurement matrix and \(y\) the system output vector. This paper deals with the estimation of nonmeasurable parameters of the form of Eq. (2).
Eykho (1974) makes a distinction between two classes of parameter estimation: explicit methods (where a set of explicit mathematical relations is solved to obtain the desired parameters) and implicit methods (where estimates of model parameters are manipulated in such a way that the model characteristics approach the characteristics of the system under study in some predefined sense). Explicit methods (such as least squares or Markov estimators) mainly deal with constant or slowly varying parameters. However, the parameter in the system under study is nonlinear and time varying. Implicit methods are more useful here. They work iteratively and converge to a solution by using a self correcting procedure which helps them to find a local optimum solution.

The Kalman filter (Kalman, 1960) provides one of the most important solutions to the parameter estimation problem. It can obtain the optimal estimate of the state of a system and can be extended to nonlinear systems and to parameter estimation. Basically, a Kalman filter uses a priori knowledge in the form of a process model to make initial state estimates and then optimizes these estimates using principles from optimal control theory (Ramirez, 1994). It is an on-line method. By reformulating a process parametervector as a process statevector, the approach can, in addition, be used to estimate model parameters or can be used for combined parameter and state estimation (see for example Eykho, 1974). The Kalman filter is originally an approach for linear models, but can be used for nonlinear models in its extended form, where local linearization is applied.

3. An alternate view

Consider the nonlinear parameter of the system under study as a system input. The effect on the behavior of the model of this new “input” is dictated by the model equations (as with normal system inputs), while the nonlinear time-varying behavior of the parameter itself needs to be determined by some other means.

Since the parameter is viewed as an input, this input can be used to control the behavior of the model. This behavior needs to be the same as the actual physical behavior of the system described by measurements. The actual behavior can be seen as a trajectory that the model needs to follow. The correct nonlinear estimation of this new “input” will result in correct model behavior.

This task can be accomplished by designing a simple feedback controller which controls the behavior of the model by manipulating the nonlinear parameter. The controller output serves as an estimate for the behavior of the nonlinear parameter. This is only true if the complete initial state is known. The controller takes the difference between the controlled model output and the desired trajectory (i.e. the innovation) as input. The controller is chosen to be a familiar PI-controller for fast response and elimination of offset:

$$\theta = L_1 i + L_2 \int i \, dt,$$

(4)

$$i = y - \hat{y},$$

(5)

where $L_1$ and $L_2$ are diagonal matrices of appropriate dimensions (with respect to the parameter vector $\theta$).

The “control scheme” is shown in Fig. 1. In this figure, the “control action” is concatenated to the input vector of the system to obtain the model input vector $u^*$:

$$u^* = \begin{bmatrix} u \\
\theta \end{bmatrix}.$$  

(6)

Following more conventional PI-controller notation, the tuning parameters, the elements on the diagonals of $L_1$ and $L_2$, can be written in terms of the controller gain $K$ and integral time constant $\tau_i$ as follows:

$$L_{1,i} = K,$$

(7)

$$L_{2,i} = \frac{K}{\tau_i},$$

(8)

4. Comparison with Kalman filtering

The controller interpretation of parameter estimation shows some analogies with conventional parameter estimation using state estimators. If a state estimator, such as a Kalman filter (or Luenberger observer) is used to make parameter estimations, the parameter is introduced as an additional state variable using the following state equation:

$$\frac{d\theta}{dt} = 0,$$

(9)

in which $\theta$ denotes the parameter. The filter configuration is shown in Fig. 2. In Fig. 2, Eq. (9) is a part of the
function block \( f \). Rearranging the equations yields the configuration shown in Fig. 3. The extra integral block is the state equation for the parameter (Eq. (9)), which is taken outside the function block \( f \). This results in \( L \) being replaced with a different gain matrix \( L_1 \) and the introduction of an additional gain matrix \( L_2 \). Thus Figs. 2 and 3 are different representations of the same Kalman filter.

The configurations in Figs. 1 and 3 are quite similar with respect to integral action. But there are also some differences. First of all, in the controller configuration there is an additional proportional term that adjusts the estimates for \( \theta \). In the Kalman filter, this term is not present. Secondly, the matrix \( L_1 \) of the Kalman filter in Fig. 3 adjusts the estimates of the complete state \( \hat{x} \). These adjustments are made in addition to the adjustments that are only made to parameter \( \theta \). So the estimates of \( \theta \), propagated through the model \( f \), result in estimates of \( \hat{x} \), which are subsequently adjusted by the innovation \( i \) and \( L_1 \). Thus this mechanism can correct estimation errors in \( \hat{x} \) caused by the estimates of \( \theta \). This mechanism is not present in the controller configuration. However, this can be seen as an advantage, because in the controller configuration, \( \hat{x} \) is only calculated accurately if the estimates of \( \theta \) are accurate.

Obviously, the calculation of the gain matrices \( L \), \( L_1 \) and \( L_2 \) differs. The different gains \( L_1 \) and \( L_2 \) are fixed and are set using PI tuning procedures. In practice, tuning of the Kalman filter is not done by adjusting \( L \), which is calculated by the Kalman equations, but by adjusting the process noise covariance matrix \( Q \), from which \( L \) is calculated.

5. Controller configuration

When the controller configuration is used in multiple parameter estimation problems, appropriate “control loops” need to be chosen. In other words, for each parameter which state is used to obtain the estimates has to be decided. Analysis has to be done to determine the interactions between the various parameters and states, so that sensible control loops can be chosen. Using the relative gain array (see Bristol, 1966), pairs of inputs \( \theta_j \) (the parameters that have to be estimated) and outputs \( x_j \) (the states that are “controlled” by the parameter estimates) can be selected in order to minimize the amount of interaction among the resulting loops.

If the static gain matrix \( G_{\text{static}} \) of the transfer functions of the system is available, the relative gain array can also be calculated as follows (Roffel & Chin, 1987):

\[
\lambda_{ij} = g_{ij} (G_{\text{static}})^{-1}_{ij},
\]

in which \( g_{ij} \) denotes the \( ij \)th element of the static gain matrix \( G_{\text{static}} \).

The relative gain array provides a measure of the interaction based on steady-state considerations. Therefore, the rule given above for the selection of loops does not guarantee that the dynamic interaction between the loops will also be minimal. The relative gain array can be replaced by its dynamic counter-part to account for this (Roffel & Chin, 1987). The interaction can then be calculated for different frequencies.

6. Estimating parameters in interacting systems

The proposed controller configuration and the standard Kalman filter approach are compared using two (simulated) test cases. In each of these cases, two parameters of the model will be estimated simultaneously. The state equations of the models are coupled through these parameters, although there is no (strong) cross-coupling, that is, each parameter that is estimated does not influence all states. The two cases represent a large class of systems that possess this property or that can be formulated so that cross-coupling is limited. Examples of estimation problems for systems with this property are the estimation of reaction rates, heat transfer coefficients, mass transfer coefficients, time varying separation coefficients, etc.
Since there is no (complete) cross-coupling, tuning the multiple controllers in the controller configuration can be done separately. This will be illustrated in both test cases.

6.1. Fed-batch bioreactor

The first case concerns a (simulated) fed-batch penicillin bioreactor (Thompson & Kramer, 1994). The model for this reactor consists of four mass balances and several additional algebraic equations, as shown in Appendix A. It is assumed that no relationship for the net biomass growth rate $\dot{x}$ is available. In addition, it assumed that the product formation rate $q_p$ also is not measurable. These parameters thus need to be estimated simultaneously. Measurements of the process were generated using a reference model of the bioreactor as given in the appendix, which was implemented in discrete form. The initial conditions can also be found there. Process noise was added to the system, and it can be anticipated that the dynamic form of the array $H$ is not observable. In this equation, $\Phi_k$ denotes the cumulative state transition matrix of the linearized system at time step $k$.

$$
\Phi_k = \phi_{k-1} \phi_{k-2} \phi_{k-3} \cdots \phi_{k_0},
$$

The tuning of the filter was done by setting the process noise covariance matrix $Q$ using trial and error. The major indication for good tuning was the innovation $i$ of the filter: the filter is tuned very well if the innovation $i$ is a white noise sequence. A more detailed discussion on the calculation of $L$ can be found in Brown and Hwang (1992).

6.3. Controller configuration

The model $f$ (see Fig. 1) used in the controller configuration consists of the state equations for the biomass concentration (Eq. (11)) and the product concentration (Eq. (12)). The two control loops consist of the net biomass growth rate $\dot{x}$ controlling the biomass concentration $X$ and the product formation rate $q_p$ controlling the product concentration $P$. This follows from the relative gain array, which was calculated using Eq. (10).

$$
O_{k_0, k_f} = \sum_{k=k_0}^{k_f} \Phi_k H^T H \Phi_k
$$

The innovation is defined as

$$
i = \begin{bmatrix} X \\ P \end{bmatrix} - \begin{bmatrix} \hat{X} \\ \hat{P} \end{bmatrix}.$$

where $X$ and $P$ are the measurements of the biomass concentration and the product concentration, while $\hat{X}$ and $\hat{P}$ are their estimates. This system is observable, as follows from the observability criterion given in Ramirez (1994). The criterion states that the system is observable from a series of measurements over the discrete time interval $[k_0, \ldots, k_f]$ if the observability Gramian $O_{k_0, k_f}$ is nonsingular. In this equation, $\Phi_k$ denotes the cumulative state transition matrix of the linearized system at time step $k$.

$$
X
$$

The model $f$ (see Fig. 1) used in the controller configuration consists of the state equations for the biomass concentration (Eq. (11)) and the product concentration (Eq. (12)). The two control loops consist of the net biomass growth rate $\dot{x}$ controlling the biomass concentration $X$ and the product formation rate $q_p$ controlling the product concentration $P$. This follows from the relative gain array, which was calculated using Eq. (10).

Although this is the static form of the relative gain array, it can be anticipated that the dynamic form of the array will not give different results. By analyzing the model equations for the penicillin fermentation process, it can be seen that there only exists a coupling from $x$ to $P$ through $X$.

Define the input vector as

$$
\begin{bmatrix} x \\ q_p \end{bmatrix}
$$

and the output vector as

$$
\begin{bmatrix} X \\ P \end{bmatrix}
$$

To determine the static loop gain matrix $G_{\text{static}}$, the nonlinear dynamic system was linearized using a first-order Taylor expansion and transformed into deviation variables, assuming stationary operation. Although the system is non-stationary, the only purpose of the relative gain array and thus the openloop static gain matrix in this application is to determine which input should control which output, so this assumption can be
Table 1
Kalman filter and PI-controller settings bioreactor

<table>
<thead>
<tr>
<th>Estimator type</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kalman filter</td>
<td>diag(Q) = [1e−4, 1e−4, 5e−6, 7e−5]</td>
</tr>
<tr>
<td>PI-Controller x</td>
<td>K = 0.06, τ_x = 2</td>
</tr>
<tr>
<td>PI-Controller q_p</td>
<td>K = 0.15, τ_q = 10</td>
</tr>
</tbody>
</table>

made. The linearization resulted in the following static gain matrix

\[
G_{\text{static}} = \begin{bmatrix}
-\frac{X_0}{\bar{z}_0 - F_0/V_0} & 0 \\
q_p \frac{-X_0}{F_0/V_0 + K\bar{z}_0 - F_0/V_0} & \frac{X_0}{F_0/V_0 + K}
\end{bmatrix},
\tag{21}
\]

in which the subscript “0” denotes the working point where the linearization was made. Element \(ij\) of \(G_{\text{static}}\) is the open loop static gain of output \(i\) with respect to input \(j\). Using Eq. (10), the relative gain array becomes

\[
A = \begin{bmatrix}
1 & 0 \\
0 & 1
\end{bmatrix}.
\tag{22}
\]

The relative gain array is independent of the point of linearization. The obvious choice is to control \(X\) with \(x\) and \(P\) with \(q_p\). This also can readily be seen from the state equations: \(q_p\) has no impact on \(X\) so it should not be used to control it.

Two controller equations are required

\[
x = L_1^x i_1 + L_2^x \int i_1 \, dt,
\tag{23}
\]

\[
q_p = L_1^q i_2 + L_2^q \int i_2 \, dt,
\tag{24}
\]

in which \(i_k\) denotes the \(k\)th element of the innovation vector \(i\). The innovation is the same as for the Kalman filter (Eq. (16)). The control parameters were set manually.

The model structure allows the controllers to be tuned separately. First, the controller for \(X\) is tuned. Since, \(q_p\) has no influence on \(X\), the controller for \(P\) has no influence on \(X\), which means that this can be done. If this controller is tuned well, the error in the estimates of \(P\) is only caused by the error in \(q_p\). So the controller for \(P\) can be tuned subsequently.

6.4. Simulation results

Measurements of the biomass concentration \(X\), the input conditions (input flow rate \(F\) and volume \(V\)) were available, as were measurements of the product concentration \(P\). Measurement sample time during the batch run was 0.2 h. Tuning of both the Kalman filter and the controllers is given in Table 1. The controllers and the Kalman filter were tuned as well as possible. The results of the estimation of \(x\) using a Kalman filter as well as the controller interpretation are given in Fig. 4. In this figure, the estimations of \(x\) for one batch run of the bioprocess are given (left figure), based on measurements of \(X\) (right figure). Fig. 5 shows the results of the estimation of \(q_p\) for one batch run.

The estimates of \(x\) of both techniques are very good: both techniques follow the measured trajectory of the biomass \(X\) closely. The estimates of the biomass \(X\) are virtually identical for both techniques: the curves in Fig. 4 overlap. Both estimation approaches generate reasonable amounts of noise in the estimates for \(x\). The Kalman filter produces less noisy estimates in the first part of the batch run, but it is somewhat slower than the PI-controller, clearly visible in the first few hours.

The Kalman filter adjusts the estimates of \(x\) as well as the estimates of \(X\) to minimize the innovation \(i\), whereas

Fig. 4. Estimation results for one batch run.
the PI-controller only uses the estimates of $x$. Thus the biomass $X$ is only estimated well if $x$ is estimated well, making the estimates of $x$ plausible. A similar statement can only be made for the Kalman filter if the tuning parameters (the elements of the process noise covariance matrix $Q$) corresponding to the measured states are set at low values, although in this case, some direct correction on the measured states remains.

The estimates of the product concentration $P$ are very good, as can be seen in Fig. 5. Both estimation approaches produce estimates of $q_p$ that result in excellent estimates of $P$, once again overlapping for both techniques. In this case, the Kalman filter is somewhat faster than the controller configuration. The estimates of $q_p$ contain more noise than the estimates of $P$. This is the result of the noise present in the estimate of the biomass concentration $X$.

As can be seen during the latter hours, the estimates of $x$ and $q_p$ generated by both the Kalman filter and the PI-controller are similar. The sensitivity to noise in this region is the same for this example and is a result of the tuning of the filter and the controller. This illustrates that for noisy signals the controller and the filter can be tuned so that they perform comparably.

6.5. Continuous reactor with exothermal reaction

The second case represents a simulated continuous ideally stirred tank reactor (CISTR) in which a simple exothermal reaction takes place, as shown in Eq. (25):

$$A \rightarrow P.$$ (25)

The reactor contains a cooling coil which runs all the way through the reactor. Its capacity to cool the reactor contents thus depends on the level in the reactor. The complete model as well as the operating assumptions are given in Appendix B. Two parameters will be estimated for this model. First of all, it is assumed that the reaction kinetics and thus the reaction rate $R$ are not known. Secondly, it is assumed that the heat transfer coefficient $U_s$ is temperature dependent and that this dependency is not known. These two parameters will be estimated using the measurements of the concentration of component $A$, the reactor temperature $T$, the level in the reactor $h$ and the temperature of the cooling liquid $T_k$.

6.6. Kalman filter

For this system, a Kalman filter was designed using the following state equations:

$$\frac{dc_A}{dt} = \frac{F_{in}}{A_s h} (c_{A,in} - c_A) - \frac{R}{\rho},$$ (26)

$$\frac{dT}{dt} = \frac{F_{in}}{A_s h} (T_{in} - T) + \frac{R \Delta H}{\rho c_p} - \frac{U_s A_s}{\rho c_p A_t h_{max}} (T - T_k),$$ (27)

$$\frac{dR}{dt} = 0,$$ (28)

$$\frac{dU_s}{dt} = 0,$$ (29)

while the cooling liquid temperature $T_k$ and the liquid level $h$ are viewed as system inputs. The coupling of the parameters in this case is somewhat different than with the bioreactor. In the bioreactor case, the coupling of $x$ to $P$ is through $X$, while with the CISTR the coupling of $R$ to $T$ is more direct.

With the measurements for $c_A$ and $T$ available, the measurement matrix $H$ for this system is given by

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$ (30)
The innovation is defined as

\[
i = \begin{bmatrix} c_A \\ T \end{bmatrix} - \begin{bmatrix} \hat{c}_A \\ \hat{T} \end{bmatrix}
\]

(31)
in which \(c_A\) and \(T\) are the measurements of the concentration and reactor temperature, while \(\hat{c}_A\) and \(\hat{T}\) are their estimates. This system is observable. As with the bioreactor case, the tuning of the filter was done by setting the process noise covariance matrix \(Q\) manually.

6.7. Controller configuration

As in the bioreactor case, the model \(f\) (see Fig. 1) used in the controller configuration consists of the state equations for the concentration \(c_A\) in the reactor (Eq. (26)) and reactor temperature (Eq. (27)). The two control loops consist of the reaction rate \(R\) controlling the concentration \(c_A\) and the heat transfer coefficient \(U_s\) controlling the reactor temperature \(T\). This configuration can easily be derived from the model equations and is confirmed by the relative gain array, which was calculated using Eq. (10).

If the input vector is defined as

\[
\begin{bmatrix} R \\ U_s \end{bmatrix}
\]

(32)
and the output vector as

\[
\begin{bmatrix} c_A \\ T \end{bmatrix}
\]

(33)
The static open loop gain matrix \(G_{static}\) can be determined by linearization.

This results in the following matrix:

\[
G_{static} = \begin{bmatrix} -\frac{c_{A0}}{\rho} & 0 \\ \frac{F_{i0}/A_{i0}h_0 + R_0/\rho}{\Delta H/\rho} & A_s(T_0 - T_{i0})/\rho c_p A_s h_{max} \\ \frac{F_{i0}/A_{i0}h_0 + U_{s0}A_s/\rho c_p A_s h_{max}}{F_{i0}/A_{i0}h_0 + U_{s0}A_s/\rho c_p A_s h_{max}} \\ \end{bmatrix}
\]

(34)
in which the subscript “0” denotes the working point where the linearization was made. Element \(ij\) of \(KG_{static}\) is the open loop static gain of output \(i\) with respect to input \(j\). Using Eq. (10), the relative gain array becomes

\[
A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}
\]

(35)
Based on the relative gain array, the concentration \(c_A\) was controlled with the reaction rate \(R\) and the temperature \(T\) with the heat transfer coefficient \(U_s\). This leads to the following controller equations:

\[
R = L^R_i i_1 + L^R_i \int i_1 \, dt,
\]

(36)
\[
U_s = L^U_i i_2 + L^U_i \int i_2 \, dt,
\]

(37)
in which \(i_k\) denotes the \(k\)th element of the innovation vector \(i\). The innovation is the same as for the Kalman filter (Eq. (31)). The controller parameters were set manually.

As with the bioreactor case, the controllers can be tuned separately. This can easily be derived from the model structure. Firstly, the controller for \(c_A\) is tuned, resulting in correct estimates for \(R\). After this, the controller for \(T\) is tuned.

6.8. Simulation results

To test both estimation approaches under non-steady state conditions, the reactor was simulated during a start-up phase. White noise was added to the various input flows of the reactor. Initial conditions can be found in appendix B. Measurements of the states and inputs were available with a sample time of 1 s. Both the Kalman filter and the controllers were tuned manually and the corresponding settings can be found in Table 2.

For this case, it can be seen in Figs. 6 and 7 that the controllers are somewhat faster than the Kalman filter. Although both techniques produce comparable estimates, the estimates of the Kalman filter are noisier than the estimates of the controller. This could not be improved by adjusting the tuning. The noisy estimates are the result of the limited sensitivity of the states with respect to the parameters: if noisy estimates are produced due to noise on some of the measurements, then this will not be present in the estimated states. This complicates the tuning.

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Kalman filter and PI-controller settings CISTR case</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimator type</td>
<td>Settings</td>
</tr>
<tr>
<td>Kalman filter</td>
<td>diag(Q) = [1e−7, 1e−7, 2e−6, 5e−5]</td>
</tr>
<tr>
<td>PI-Controller (R)</td>
<td>(K = -10, t_i = 500)</td>
</tr>
<tr>
<td>PI-Controller (U_s)</td>
<td>(K = -0.3, t_i = 200)</td>
</tr>
</tbody>
</table>
for the measured states (the curves overlap). This was also found for the bioreactor.

If the estimates are compared to “real values” taken from the reference model (not shown), it was found that the estimates of the controller configuration are better than those of the Kalman filter. This is caused by the fact that the Kalman filter needs more time to zoom in than the controllers, resulting in larger errors in the first part of the simulation.

7. Practical application

In addition to the two simulated test cases, the approach is applied to a practical case. For the experiments, a batch distillation column with 21 bubble cap trays has been used with an internal diameter of 76 mm and a Murphree vapor tray efficiency of about 53%. It concerns a binary separation of ethanol and 1-propanol. The vapor rate is kept constant just below the flooding constraint, which results in the maximum production rate. The only manipulated variable is the reflux ratio $R^*$.

The tray mixing time $\tau_x$ is 9–11 s, whereas the hydraulic time constant $\tau_L$ is 2.5–3 s. The maximum exhaustion time is about 4 h.

To describe the medium term to exhaustion dynamics of this column, a simplified model comprised of the static overall separation–approximation combined with first-order dynamics for the exhaustion has been derived (Betlem, 1997). The model is given below.

\[
d\frac{M_{\text{col}}}{dt} = -L_D, \tag{38}
\]

\[
M_{\text{col}} \frac{dx_{\text{col}}}{dt} = -L_D(x_{n+1}^0 - x_{\text{col}}), \tag{39}
\]

\[
x_{n+1}^0 = f(R^*, x_{\text{col}}), \tag{40}
\]

in which $M_{\text{col}}$(mol) is the mass of the column, $L_D$(mol/h) is the distillate flow rate, $x_{\text{col}}$ is the average molar ethanol fraction in the column, $x_{n+1}^0$ is the molar ethanol fraction of the product and $R^*$ is the reflux ratio. $M_{\text{col}}$ and $x_{\text{col}}$ are introduced to avoid the need for tray-to-tray equations to describe the column (Betlem, 1997).
Table 3
Initial conditions for batch distillation column

<table>
<thead>
<tr>
<th>Run No.</th>
<th>( M_{\text{col,0}} ) (mol)</th>
<th>( x_{\text{col,0}} ) (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>194.3</td>
<td>0.5</td>
</tr>
<tr>
<td>2</td>
<td>201.6</td>
<td>0.4</td>
</tr>
<tr>
<td>3</td>
<td>200.2</td>
<td>0.6</td>
</tr>
</tbody>
</table>

Table 4
PI-controller settings batch distillation column

<table>
<thead>
<tr>
<th>Estimator type</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>PI-Controller</td>
<td>( K = 1.2, \tau_i = 0.003 )</td>
</tr>
</tbody>
</table>

The function \( f \) in Eq. (40) is given by two additional equations as shown below.

\[
S = \frac{x_{n+1}^0(1 - x_{\text{col}})}{x_{\text{col}}(1 - x_{n+1}^0)}
\]

(41)

\[
S = \left( \frac{x^*}{1 + x^*} \right)^{NE, +1}.
\]

(42)

In these equations, \( S \) is the separation factor, \( N \) is the number of trays and \( E_v \) is the tray efficiency, which is 53\%, \( x^* \) is a measure for the relative volatility. Since the separation factor \( S \) is not based on the bottom composition but on the overall column composition, \( x^* \) changes significantly during a batch run of the column. This is caused by the simplifications that are used in this model; effects that are neglected manifest themselves in this parameter. This parameter will be estimated using a PI controller.

7.1. Controller configuration

Measurements of the reflux ratio \( R^* \), the flow rate \( L_0 \), and the top quality (estimated using a PLS estimator) \( x_{n+1}^0 \) are available, as are measurements of the initial conditions \( M_{\text{col,0}} \) and \( x_{\text{col,0}} \). Since \( x_{n+1}^0 \) is measurable, \( x^* \) will be estimated using the model error in \( x_{n+1}^0 \).

The configuration is quite different than with the two simulated cases. With these cases, the model parameters were estimated by “controlling” the states of the process. Here, an internal function parameter is estimated by the “controller”. This requires the function describing the model parameter to be available (which is not the case with the bioreactor or the continuous reactor, for example, no functional relation for \( q_p \) is required). The coupling of the estimated parameter with the states is thus through the model parameter \( x_{n+1}^0 \). In addition, the model states are needed to evaluate the function, which also provides coupling.

It is interesting to note that if this estimation problem is solved using a Kalman filter, the model needs to be reformulated in order to be used in the filter structure, since the Kalman filter requires the estimated parameter to be available as a process state. Furthermore, the measurements also need to represent these states (one way or another through the measurement matrix \( H \)).
Using a PI-controller, this problem is solved much more easily.

7.2. Experimental results

$z^*$ was estimated for three different batch runs. Initial conditions are provided in Table 3. The sample rate for the measurements was 1 min. The measurements represent the controlled process; the top quality $x_{\text{top}}$ was kept constant by adjusting the reflux ratio $R^*$. The PI controller for the estimates was tuned manually and the settings are shown in Table 4. All three runs used the same controller settings.

Figs. 8–15 show the estimation results. In Figs. 10 and 11 the states of the process for the three runs are shown. It can be seen that production starts after about half an
hour. Before that, the column is in startup phase with infinite reflux (no production). During the first few minutes of startup, the measurement system may give anomalous readings, as can be seen in Figs. 12 and 13. When production starts, the quality drops slightly to its setpoint value (see Figs. 12-14), which results in a drop in $a^*$ at about the same time. After that, $a^*$ increases significantly in order to obtain a model match with the measurements of the top quality $x_{col}$. The PI-controller performs excellently for each of the batch runs. Fig. 16 shows the model performance for constant $a^*$ for run 2, clearly illustrating the need for a varying $a^*$. 

Fig. 11. Estimation results for $x_{col}$.

Fig. 12. Estimation results for $x_{a+1}^0$, run 1.
8. Conclusions

With regard to the computational structure, the feedback controller approach to nonlinear parameter-estimation shows many resemblances with Kalman filtering. It can be a good alternative to Kalman filtering. The approach has been illustrated on two simulated multiple parameter-estimation problems and an experimental setup. In each of the cases, time-varying parameters were successfully estimated by the PI-controller.
The results were comparable with Kalman filter estimates, while with the experimental case, the controller approach is much simpler to implement that a Kalman filter, because no model reformulation is necessary. Although the Kalman filter has versatile application possibilities in combined state and parameter estimation, the PI-controller has the advantage that it is simple, easy to use and easy to tune for simple single or multiple parameter-estimation problems and therefore may be preferred to Kalman filtering.
Appendix A. Penicillin fermentation model

The model describes four states of the process, namely the cell biomass concentration $X$ (gDCW/l, where DCW means dry cell weight), the substrate concentration $S$ (g/l), the product concentration $P$ (g/l) and the reactor volume $V$ (l) (Tables 5 and 6). The state equations are given by

$$\frac{dX}{dt} = X\left(\alpha - \frac{F}{V}\right),$$  \hspace{1cm} (A.1)

$$\frac{dS}{dt} = -\sigma X + (S_f - S)\frac{F}{V},$$  \hspace{1cm} (A.2)

$$\frac{dP}{dt} = q_p X - P\left(\frac{F}{V} + K\right),$$  \hspace{1cm} (A.3)

$$\frac{dV}{dt} = F,$$  \hspace{1cm} (A.4)

where $F$ is the input flow (l/h), $S_f$ the substrate concentration in feed (525 g/l), and $K$ the product decay constant (0.01 h$^{-1}$).

All the other symbols govern specific rates which play a role in the process. They are defined below.

Net growth rate:

$$\alpha = \frac{\mu_m S}{K_x X + 10} - \frac{c_{Lm} X}{K_L + X + 1}e^{-s/100},$$  \hspace{1cm} (A.5)

where $\mu_m$ (0.11 h$^{-1}$), $K_x$ (0.3), $c_{Lm}$ (0.0084 h$^{-1}$), and $K_L$ (0.05) are constants.

Product formation rate:

$$q_p = \frac{1.5q_{pm} SX}{4K_p + SX(1 + s/3K_f)},$$  \hspace{1cm} (A.6)

where $K_p$ (0.001 g/l), $q_{pm}$ (0.004 h$^{-1}$), and $K_f$ (1.0 g/l) are constants.

Table 5

<table>
<thead>
<tr>
<th>$X$ (gDCW/l)</th>
<th>$S$ (g/l)</th>
<th>$P$ (g/l)</th>
<th>$V$ (l)</th>
<th>$F$ (l/h)</th>
<th>$S_f$ (g/l)</th>
</tr>
</thead>
<tbody>
<tr>
<td>30.0</td>
<td>0.5</td>
<td>0.0</td>
<td>20.0</td>
<td>0.110</td>
<td>525</td>
</tr>
</tbody>
</table>

Table 6

<table>
<thead>
<tr>
<th>Variable</th>
<th>Noise amplitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X$</td>
<td>$[-0.2, +0.2]$</td>
</tr>
<tr>
<td>$S$</td>
<td>$[-0.1, +0.1]$</td>
</tr>
<tr>
<td>$F$</td>
<td>$[-0.005, +0.005]$</td>
</tr>
</tbody>
</table>

Substrate consumption rate:

$$\sigma = \frac{\mu}{Y_{x/s}} + \frac{q_p}{Y_{p/s}} + m_x, \hspace{1cm} (A.7)$$

where $Y_{x/s}$ is a yield factor (0.47), $Y_{p/s}$ a yield factor (1.2), and $m_x$ the maintenance energy factor.

Maintenance energy:

$$m_x = \frac{m_{sm} X}{X + 10^1}, \hspace{1cm} (A.8)$$

where $m_{sm}$ is constant (0.029 h$^{-1}$).

Appendix B. Continuous reactor model

Four state equations are needed to describe the behavior of the CISTR (Tables 7 and 8). They comprised the component balance describing $c_A$ (kg/kg), the energy balance for the reactor giving the reactor temperature $T(K)$, the mass balance describing $h$ (m) and the energy balance of the cooling coil giving the temperature of the cooling liquid $T_k$ (K).

$$\frac{dc_A}{dt} = \frac{F_{in}}{A_t h} (c_{A,in} - c_A) - \frac{R}{\rho}, \hspace{1cm} (B.1)$$

$$\frac{dT}{dt} = \frac{F_{in}}{A_t h} (T_{in} - T) + \frac{R\Delta H}{\rho c_p} - \frac{U_A A_t h_{max}}{\rho c_p A_t h_{max}} (T - T_k), \hspace{1cm} (B.2)$$

$$A_t \frac{dh}{dt} = F_{in} - F_{out}, \hspace{1cm} (B.3)$$

$$\frac{dT_k}{dt} = \frac{F_k}{V_k} (T_{k,in} - T_k) + \frac{U_A A_t h_{max} V_k}{\rho c_p A_t h_{max}} (T - T_k), \hspace{1cm} (B.4)$$

where $F_{in}$ is the input flow (0.005 m$^3$/s), $A_t$ the reactor cross sectional area (2 m$^2$), $c_{A,in}$ the input flow concentration (1 kg/kg), $\rho$ the reactor contents density (800 kg/m$^3$), $T_{in}$ the temperature input flow (353 K), $\Delta H$ the reaction enthalpy (209 kJ/kg), $c_p$ the heat capacity reactor
contents (1 kJ/kgK), $A_z$ the maximum heat exchange area (20 m$^2$), $h_{\text{max}}$ the maximum reactor contents level (5 m), $F_{\text{out}}$ the output flow (0.005 m$^3$/s), $F_k$ the cooling liquid flow (0.005 m$^3$/s), $V_k$ the cooling coil volume (1 m$^3$), and $T_{k,\text{in}}$ the cooling liquid input flow temperature (288 K), $c_{p,k}$ the heat capacity cooling liquid (4.2 kJ/kgK).

During operation, it is assumed that the level in the reactor is controlled ideally (so $F_{\text{in}} = F_{\text{out}}$). The other symbols are governed by algebraic equations, given below.

Reaction rate:
$$R = k_0 e^{-\frac{E}{RT_c} A} \quad (B.5)$$
where $k_0$ is the pre-exponential constant (15000 kg/m$^3$/s), $E$ the activation energy (30 kJ/mol), and $R$ the gas constant ($8.31 \, e - 3 \, \text{kJ/mol/K}$).

Heat transfer coefficient:
$$U_s = a + b e^{\alpha T} \quad (B.6)$$
where $a$ (0.5 kW/m$^2$K), $b$ (4.5 $\alpha - 5 \, \text{kW/m}^2\,\text{K}$) and $c$ (0.03) are constants.

References