Using Modelica/Matlab for parameter estimation in a bioethanol fermentation model

Juan I. Videla Bernt Lie
Telemark University College
Department of Electrical Engineering, Information Technology, and Cybernetics
Porsgrunn, 3901 Norway

Abstract

Bioethanol production from fermentation of a substrate using biomass as catalyst is considered. Four alternative reaction rate models with different levels of details are derived and implemented in Modelica. The problem of parameter estimation of models using state/parameter estimation techniques in a Modelica-Dymola/Matlab setup is discussed. Practical aspects concerning the different implementations of nonlinear estimators are analyzed (EKF, UKF, and EnKF). The use of Modelica-Dymola for "on-line" applications such as state estimation poses the additional problem of the efficiency of the code; this will also be discussed. The four reaction rate models are fitted using fictitious experimental data generated from one of the models to illustrate the parameter estimation procedure.

Keywords: bioethanol fermentation, parameter estimation, nonlinear estimators

1 Introduction

Alcoholic fermentation is an important biochemical process which has been known for some 5000 years. Ethyl alcohol, or more commonly ethanol, has chemical formulae C_2H_5OH, and finds uses as (i) alcoholic beverage (beer, wine, spirits), (ii) solvent, (iii) raw material in chemical synthesis, and (iv) fuel.

With the current focus on CO_2 release and global warming, there is a considerable interest in producing fuel from biomass. Production of ethanol from fermentation typically involves a two step process: (a) the main process where substrate (glucose) is converted to ethanol and non-fossil CO_2 in an enzymatic process, and (b) the aerobic yeast growth through the consumption of substrate and oxygen.

In continuous reactors, yeast is contiuously washed out, leading to a less efficient use of the yeast. The use of immobilized yeast increases the efficiency of the process, as less substrate is "wasted" for yeast production. In fermentation, salts are involved as co-enzymes. The resulting ions affect the oxygen uptake in the reaction mixture.

The produced (bio-) ethanol can be used as fuel after some additional processing –filter yeast, remove water by distillation, etc. Alternatively, the ethanol can be converted to methane by microorganism.

The efficient production of ethanol in a fermentation reactor requires quantitative analysis of how raw materials are converted to products. Static models are often used for design purposes for continuous reactors, while dynamic models are required for batch reactors (e.g. beer production) and for control analysis and design in continuous reactors. A simple numeric dynamic model for the continuous fermentation of glucose using the yeast *saccharomyces cerevisae* is given in [1]. The model is somewhat simplified in that the dynamics of the overall reactor volume is neglected, the role of the salts as co-enzymes is neglected, and somewhat simple kinetic reaction rates are used. A more systematic development of reaction rates for the continuous ethanol fermentation process is presented in [2]. In [1], the effect of ions on the oxygen uptake in the reactor mixture is included, but the effect of glucose is neglected; expressions for the effect of ions and sugars are given in [3]. Most of the parameters of the model of [1] are given in their publication; however there is one or two typos, and the effect of salt ions on the oxygen uptake is as if the salinity of the reaction mixture was similar to that of sea water (due to some mole-to-gram conversion problem).

It is of interest to study the parameter estimation problem of the fermentation model for the different rates of reaction models with the purposes of control and identification. Online parameter identification can be achieved using recursive state/parameter estimators. For linear systems with normally distributed process and measurement noise, the optimal recursive estimator is the Kalman filter. Estimation for nonlinear systems is considerably more difficult and admits a wider variety of suboptimal solutions. The extended Kalman filter (EKF), unscented Kalman filter (UKF), and the ensemble Kalman filter (EnKF) are implemented using Modelica-Dymosim and Matlab. The fermentation with the different reaction rates is implemented in Modelica and compiled into Dymosim. The parameters are directly estimated using the parameter state-augmented approach and the discrete version of the estimators are implemented in Matlab.

The paper is organized as follows. In the next section, an overview of the fermentation process and its implementation is given. Different kinetic reaction rates for the fermentation process are presented in accordance with biochemical engineering principles. We give a brief introduction of the implementation of the proposed models in Modelica. In section 3, we discuss the problem of parameter estimation of models using recursive nonlinear state/parameter estimation techniques in a Modelica/Matlab setup. The traditional use of the Extended Kalman Filter poses some questions regarding the computation of the Jacobians of the system. In more modern techniques such as the Unscented Kalman Filter, and Monte Carlo techniques such as the Ensemble Kalman Filter, the computation of Jacobians is avoided. Also, these more modern techniques handle nonlinearities in a better way than the Extended Kalman Filter. In particular when these estimation techniques are used for parameter estimation, some of the filter constants need to be carefully tuned, and we discuss this problem. Also, the use of Modelica for "on-line" application such as state estimation poses some particular problems with regards to the efficiency of Modelica implementations; this will be discussed. Finally, we assume that the model of [1] has been fitted well to experimental data. We then generate fictitious experimental data from the model of Agachi et al., and we illustrate the parameter estimation procedure by fitting the new models to the generated experimental data.

2 Fermentation model

2.1 Description

The nutrients in biochemical reactions are known as substrates. The substrate for the ethanol production process is thus glucose. For the yeast growth process, the substrates are glucose and oxygen. In the sequel we will use symbol S do denote glucose. Since oxygen has a relatively simple chemical formulae, we will not introduce a particular notation for oxygen. Furthermore, we will use symbol P for the main product, which is ethanol, and symbol X for the yeast.
The *original* reaction kinetics given by [1] can be seen in Table 3 with the superscript o for every specie r_j^o.
The fermentation reactor for the production of ethanol is sketched in Fig. 1. Glucose (*substrate* S, sugar) in a water solution is continuously fed to the well stirred reactor; the volumetric feed flow is $\dot{V}_i$ [volume/time]. The reactor contains yeast (microorganisms X), which reacts with substrate to produce ethanol (*product* P). We consider this reaction 1, with kinetic reaction rate r_1 [mass/(volume time)] [1].

$$\mathrm{S} \xrightarrow{\mathrm{E_1}} \mathrm{P} \tag{1}$$

Simultaneously, in a second reaction (2), the microorganism breed under the consumption of oxygen to produce more yeast; the kinetic reaction rate is r_2 [mass/(volume time)].

$$\mathrm{S} + \mathrm{O_2} \xrightarrow{\mathrm{E_2}} \mathrm{X} \tag{2}$$

The relationships between the rates of generation r_j [mass/(volume time)] with $j \in \{\mathrm{P}, \mathrm{X}, \mathrm{S}, \mathrm{O_2}\}$ can be seen in Table 3. All reaction rates have dimension mass/(volume time). It follows that $r_1 = r_\mathrm{P}$ is the mass of ethanol produced per volume and time, etc. Factor Y_SP has the meaning of *mass of ethanol (product) produced per mass of glucose (substrate) consumed*. Similar interpretations are valid for Y_SX and Y_OX.

[1] The $\mathrm{CO_2}$ specie is not considered in the expression.

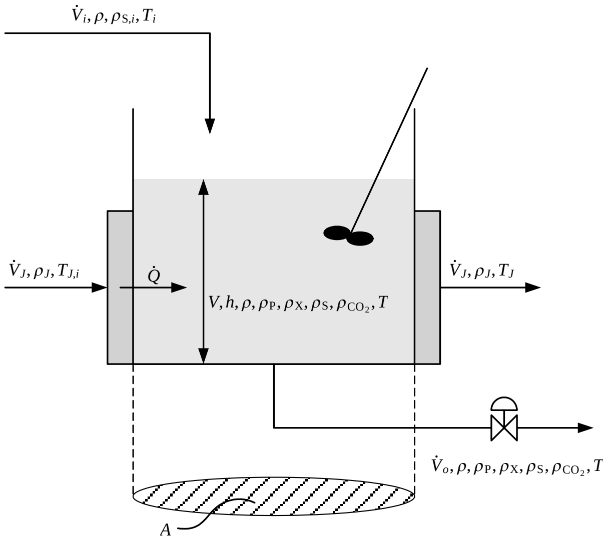
$\dot{V}_i, \rho, \rho_{S,i}, T_i$
$\dot{V}_J, \rho_J, T_{J,i}$
$\dot{Q}$
$V, h, \rho, \rho_P, \rho_X, \rho_S, \rho_{CO_2}, T$
$\dot{V}_J, \rho_J, T_J$
$\dot{V}_o, \rho, \rho_P, \rho_X, \rho_S, \rho_{CO_2}, T$
A

Fig. 1: Sketch of fermentation reactor.

Tab. 1: Fermentation model.

Reactor total mass and species balances:
$\frac{d}{dt}m = \dot{m}_i - \dot{m}_o$
$\frac{d}{dt}m_j = \dot{m}_{j,i} - \dot{m}_{j,o} + \dot{m}_{j,g}^k$ with $j \in \{\mathrm{P}, \mathrm{X}, \mathrm{S}, \mathrm{O}_2\}$
Reactor rates of generation:
$\dot{m}_{j,g} = r_j^k V$ with $j \in \{\mathrm{P}, \mathrm{X}, \mathrm{S}, \mathrm{O}_2\}$
Reactor outputs:
$\dot{m}_o = k\sqrt{V}$
$\dot{m}_{j,o} = \dot{V}_o \rho_j$ with $j \in \{\mathrm{P}, \mathrm{X}, \mathrm{S}, \mathrm{O}_2\}$
Reactor inputs:
$\dot{m}_i = \rho \dot{V}_i$
$\dot{m}_{\mathrm{P},i} \equiv 0$
$\dot{m}_{\mathrm{X},i} \equiv 0$
$\dot{m}_{\mathrm{S},i} = \rho_{\mathrm{S},i} \dot{V}_i$
$\dot{m}_{\mathrm{O}_2} = \dot{m}_{\mathrm{O}_2,a}$
Oxygen interface transport:
$\dot{m}_{\mathrm{O}_2,a} = k_{\ell a} \left(\rho_{\mathrm{O}_2}^* - \rho_{\mathrm{O}_2}\right) V$
$\rho_{\mathrm{O}_2}^* = \rho_{\mathrm{O}_2,0}^* (T) \exp\left(-\sum_n I_n - \sum_m S_m\right)$
with $n \in \{\mathrm{Na}^+, \mathrm{Cl}^-, \mathrm{Ca}^{+2}, \mathrm{CO}_3^{-2}, \mathrm{Mg}^{+2}, \mathrm{H}^+, \mathrm{OH}^-\}$
$I_n = \frac{1}{2} H_j z_j^2 c_j$
$\sum_m S_m = S_\mathrm{S} = K_\mathrm{S} c_\mathrm{S}$
$\rho_{\mathrm{O}_2,0}^* (T) = \alpha_0 + \alpha_1 T + \alpha_2 T^2 + \alpha_3 T^3$
Reactor energy balance:
$\rho \hat{c}_p V \frac{dT}{d} = \rho \hat{c}_p \dot{V}_i (T_i - T) + \Delta H_{r,2} V r_{\mathrm{O}_2}^j - \dot{Q}_\mathrm{heatex}$
Water jacket mass balance:
$\dot{m}_{J,i} + \dot{m}_{J,o} = 0$
Water jacket energy balance:
$\rho_J \hat{c}_{p,J} V_J \frac{dT_J}{dt} = \rho_J \hat{c}_{p,J} \dot{V}_J (T_{J,i} - T_J) + \dot{Q}_\mathrm{heatex}$
Water jacket-reactor heat transfer:
$Q_\mathrm{heatex} = U_x A_x (T - T_J)$

Both in the inlet stream and in the reaction medium, water is dominant such that the density ρ of the mixture can be assumed to be constant. For oxygen, there is an input flow $\dot{m}_{\mathrm{O}_2,a}$ in that oxygen is transported from air to dissolved oxygen in the reaction medium,

$$\dot{m}_{\mathrm{O}_2,a} = k_{\ell a} \left(\rho_{\mathrm{O}_2}^* - \rho_{\mathrm{O}_2}\right) V, \tag{3}$$

where $k_{\ell a}$ [1/time] depends on the temperature, V is the volume of the reaction medium, ρ_{O_2} is the mass based concentration [mass/volume], and $\rho_{\mathrm{O}_2}^*$ is the equilibrium concentration of oxygen in water. $\rho_{\mathrm{O}_2}^*$ depends on salts in the mixture. It is assumed that there is no O_2 in the feed water stream. In both reactions 1 and 2, CO_2 is released as a byproduct; here we do not model the carbon dioxide production.

The total mass, species balances, and energy balance for the reactor and the water jacket models are presented in Table 1. The fermentation boundary conditions defined as inputs and outputs are defined in this table. The fermentation model parameters of the *original* model developed in [1] are shown in Table 2.

2.1.1 Fermentation reaction and rates

The *elementary* reaction rate r_1^e for the ethanol production is developed considering the substrate-enzyme interactions, the resulting rate is given by the *Michaelis-Menten kinetics*. Additionally, the presence of ethanol inhibits the ethanol production rate (inactive enzymes), this effect is also included in this reaction rate. The combined effect is shown in Table 3.

A common simplified model for the effect of competition for active sites yields the *simplified* rate r_1^s, where a specie that competes for an active site and participate in the reaction has the form $\rho_\mathrm{S}/(K_{\mathrm{S},1}+\rho_\mathrm{S})$, while a specie that competes for an active site and does not participate in the reaction has the form $1/(1 + k_{\mathrm{P},1}\rho_\mathrm{P})$.

Another possible model for ethanol production with ethanol inhibition is to notice that $\exp\left(-k_{\mathrm{P},1}\rho_\mathrm{P}\right) \approx 1/\left(1 + k_{\mathrm{P},1}\rho_\mathrm{P}\right)$. This exponential term can be explained by assuming inhibition by ethanol may be caused by *intracellular* mechanisms.

A similar analysis can be done for the reaction rate for the yeast production for the different ap-

Tab. 2: Parameters for the fermentation reactor with original reaction rates.

Reactor/Water jacket parameters:	
$\rho = 1080\,\mathrm{g/l}$	$\Delta\tilde{H}_{r,\mathrm{O_2}} = -518\,\mathrm{kJ/mol_{O_2}}$
$\rho_J = 1000\,\mathrm{g/l}$	$V_J = 50\,\mathrm{l}$
$\hat{c}_p = 4.18\,\mathrm{J/(g\,°C)}$	$U_x A_x = 3.6\,\mathrm{E5\ J/(h\,°C)}$
$\hat{c}_{p,J} = 4.18\,\mathrm{J/(g\,°C)}$	$(k_{la})_0 = 38\,\mathrm{h^{-1}}$
Rate of generation parameters:	
$\mu_1 = 1.79\,\mathrm{h^{-1}}$	$K_{\mathrm{S},2} = 1.03\,\mathrm{g/l}$
$\mu_{\mathrm{O_2}} = 0.5\,\mathrm{h^{-1}}$	$K_{\mathrm{O_2}} = 8.86\,\mathrm{mg/l}$
$A_1 = 9.5\,\mathrm{E8\ h^{-1}}$	$k_{\mathrm{P},1} = 0.070\,\mathrm{l/g}$
$A_2 = 2.55\,\mathrm{E33\ h^{-1}}$	$k_{\mathrm{P},2} = 0.139\,\mathrm{l/g}$
$E_{a1}/R = 6.6185\,\mathrm{E3\ K}$	$Y_{\mathrm{SX}} = 0.607\,\mathrm{g_X/g_S}$
$E_{a2}/R = 26.474\,\mathrm{E3\ K}$	$Y_{\mathrm{SP}} = 0.435\,\mathrm{g_P/g_S}$
$K_{\mathrm{S},1} = 1.68\,\mathrm{g/l}$	$Y_{\mathrm{OX}} = 0.970\,\mathrm{g_X/g_{O_2}}$
Oxygen interface transport parameters:	
$z_{\mathrm{Na^+}} = +1$	$H_{\mathrm{Mg^{+2}}} = -0.314\,\mathrm{l/mol}$
$z_{\mathrm{Cl^-}} = -1$	$H_{\mathrm{Ca^{+2}}} = -0.303\,\mathrm{l/mol}$
$z_{\mathrm{Ca^{+2}}} = +2$	$K_{\mathrm{S}} = 0.119\,\mathrm{l/mol}$
$z_{\mathrm{CO_3^{-2}}} = -2$	$M_{\mathrm{NaCl}} = 58.44\,\mathrm{g}$
$z_{\mathrm{Mg^{+2}}} = +2$	$M_{\mathrm{MgCl_2}} = 95.21\,\mathrm{g/mol}$
$z_{\mathrm{H^+}} = +1$	$M_{\mathrm{CaCO_3}} = 100.09\,\mathrm{g/mol}$
$z_{\mathrm{OH^-}} = -1$	$M_{\mathrm{O_2}} = 32\,\mathrm{g/mol}$
$H_{\mathrm{Na^+}} = -0.55\,\mathrm{l/mol}$	$M_{\mathrm{S}} = 180.15\,\mathrm{g/mol}$
$H_{\mathrm{Cl^-}} = 0.84\,\mathrm{l/mol}$	$\alpha_0 = 14.16\,\mathrm{mg/l}$
$H_{\mathrm{OH^-}} = 0.94\,\mathrm{l/mol}$	$\alpha_1 = -0.394\,\mathrm{mg/(l\,°C)}$
$H_{\mathrm{CO_3^{-2}}} = 0.48\,\mathrm{l/mol}$	$\alpha_2 = 7.71\,\mathrm{E{-3}\,mg/(l\,°C^2)}$
$H_{\mathrm{H^+}} = -0.77\,\mathrm{l/mol}$	$\alpha_3 = -6.4\,\mathrm{E{-5}\,mg/(l\,°C^3)}$

Tab. 3: Parameters for the fermentation reactor with original reaction rates.

Reaction rates 1:

$$r_1^o = \mu_1 \rho_\mathrm{X} \frac{\rho_\mathrm{S}}{K_{\mathrm{S},1}+\rho_\mathrm{S}} \exp\left(-k_{\mathrm{P},1}\rho_\mathrm{P}\right)$$

$$r_1^e = \mu_1 \rho_\mathrm{X} \frac{\rho_\mathrm{S}}{K_{\mathrm{S},1}+\left(1+k_{\mathrm{P},1}\rho_\mathrm{P}\right)+\rho_\mathrm{S}}$$

$$r_1^s = \mu_1 \rho_\mathrm{X} \frac{\rho_\mathrm{S}}{K_{\mathrm{S},1}+\rho_\mathrm{S}} \frac{1}{1+k_{\mathrm{P},1}\rho_\mathrm{P}}$$

$$r_1^i = \mu_1 \rho_\mathrm{X} \frac{\rho_\mathrm{S}}{K_{\mathrm{S},1}+\rho_\mathrm{S}} \exp\left(-k_{\mathrm{P},1}\rho_\mathrm{P}\right)$$

Reaction rates 2:

$$r_2^o = \mu_2 \rho_\mathrm{X} \frac{\rho_\mathrm{S}}{K_{\mathrm{S},2}+\rho_\mathrm{S}} \exp\left(-k_{\mathrm{P},2}\rho_\mathrm{P}\right)$$

$$r_2^e = \mu_2 \rho_\mathrm{X} \frac{\rho_\mathrm{S}\rho_{\mathrm{O_2}}}{K_{\mathrm{S},2}K_{\mathrm{O_2}}+\left(1+k_{\mathrm{P},2}\rho_\mathrm{P}\right)+K_{\mathrm{O_2}}\rho_\mathrm{S}+\rho_\mathrm{S}\rho_{\mathrm{O_2}}}$$

$$r_2^s = \mu_2 \rho_\mathrm{X} \frac{\rho_\mathrm{S}}{K_{\mathrm{S},2}+\rho_\mathrm{S}} \frac{\rho_{\mathrm{O_2}}}{K_{\mathrm{O_2}}+\rho_{\mathrm{O_2}}} \frac{1}{1+k_{\mathrm{P},2}\rho_\mathrm{P}}$$

$$r_2^i = \mu_2 \rho_\mathrm{X} \frac{\rho_\mathrm{S}}{K_{\mathrm{S},2}+\rho_\mathrm{S}} \frac{\rho_{\mathrm{O_2}}}{K_{\mathrm{O_2}}+\rho_{\mathrm{O_2}}} \exp\left(-k_{\mathrm{P},2}\rho_\mathrm{P}\right)$$

Rates of reactions for $\mathrm{P}, \mathrm{X}, \mathrm{S}, \mathrm{O_2}$

$$\begin{array}{lcl}
r_\mathrm{P}^k = r_1^k & , & k = \{o, e, s, i\} \\
r_\mathrm{X}^k = r_2^k & , & k = \{o, e, s, i\} \\
r_\mathrm{S}^k = -\frac{1}{Y_\mathrm{SP}} r_1^k - \frac{1}{Y_\mathrm{SX}} r_2^k & , & k = \{o, e, s, i\} \\
r_{\mathrm{O_2}}^{k^*} = -\frac{1}{Y_\mathrm{OX}} r_2^{k^*} & , & k^* = \{e, s, i\} \\
r_{\mathrm{O_2}}^o = -\frac{1}{Y_\mathrm{OX}} \mu_{\mathrm{O_2}} \rho_\mathrm{X} \frac{\rho_{\mathrm{O_2}}}{K_{\mathrm{O_2}}+\rho_{\mathrm{O_2}}} & &
\end{array}$$

proximations.

The *original* rates are closely related to the developed rates where product inhibition is explained via *intracellular* transport. The *original* model neglects the oxygen dependence of the *intracellular* model and neglects the substrate dependence and the product inhibition. Clearly, when the kinetic rates change their functional form, the parameter/temperature functions change. The different rate reaction rates are shown in Table 3.

2.2 Implementation

In Modelica it is important to implement a good structure to enable easy modification of the models. The core model of the fermentation reactor is the basic volume model, there is where the total mass, species mass balances, and energy balance are defined. This model exchanges heat with the water jacket model through an MSL heat port. It also has a chemical port (i.e. *intensive variables*: temperature and mass concentration vector; and *extensive variables*: mass flow rates vector and heat flow rate) that connects with the rate of generation replaceable model and the oxygen transport model, and two thermofluid ports (i.e. *intensive variables*: pressure, specific enthalpy, and mass fraction vector; and *extensive variables*: enthalpy flow rate vector, mass flow rate vector, and total mass flow rate) to connect the basic volume with the incoming mass flow rate in and the outcoming mass flow rate out of the model. The basic volume is then connected to the water jacket model, to the oxygen transport model, to the rate of generation replaceable model as shown in Fig. 2.

The four different reaction kinetic rates (i.e. *original*, *elementary*, *simplified*, and *intracellular*) are implemented using a replaceable component. A common set of parameters and equations are defined in a partial model called rate of generation. Specifics of every reaction rate model are defined separately in each model that inherits the rate of generation partial model. The heat of reaction is also defined in these models. The water jacket model uses two MSL flow ports.

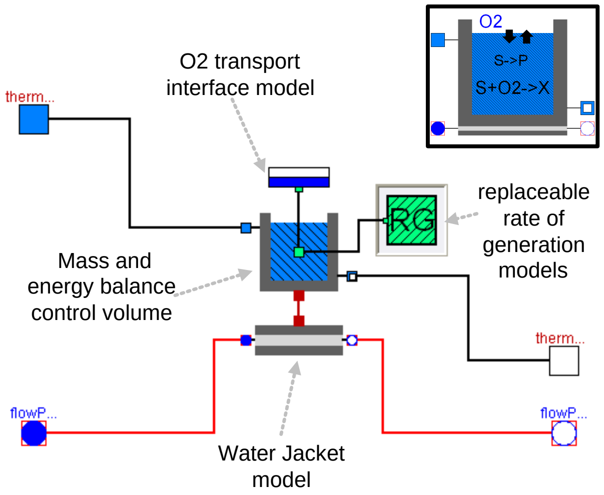
O2
S->P
S+O2->X
O2 transport interface model
therm...
replaceable rate of generation models
RG
Mass and energy balance control volume
therm...
flowP...
flowP...
Water Jacket model

Fig. 2: Dymola diagram layout of the fermentation reactor component.

Tab. 4: EKF algorithm.

```
Initialization:
$\hat{x}_{0|0} \sim \mathcal{N}(\bar{x}_0, P_0)$
$P_{0|0} = P_0$
for $k = 1, 2, \ldots$
Propagation step:
    (a priori covariance estimate)
    $F_{k-1} = \frac{\partial f_{k-1}}{\partial x_{k-1}}\Big|_{\hat{x}_{k-1|k-1}} \qquad L_{k-1} = \frac{\partial f_{k-1}}{\partial w_{k-1}}\Big|_{\hat{x}_{k-1|k-1}}$
    $P_{k|k-1} = F_{k-1} P_{k-1|k-1} F_{k-1}^T + L_{k-1} Q_{k-1} L_{k-1}^T$
    (a priori state-output estimate)
    $\hat{x}_{k|k-1} = f_{k-1}(\hat{x}_{k-1|k-1}, u_{k-1}, 0)$
    $\hat{y}_{k|k-1} = h_k(\hat{x}_{k|k-1}, 0)$
Measurement update:
    (Kalman gain calculation)
    $H_k = \frac{\partial h_k}{\partial u_k}\Big|_{\hat{x}_{k|k-1}} \qquad M_k = \frac{\partial h_k}{\partial v_k}\Big|_{\hat{x}_{k|k-1}}$
    $K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + M_k R_k M_k^T)^{-1}$
    (a posteriori state-covariance estimate)
    $\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (y_k - \hat{y}_{k|k-1})$
    $P_{k|k} = (I - K_k H_k) P_{k|k-1}$
```

3 Nonlinear estimators

3.1 Description

The ethanol fermentation reactor model can be written in the general discrete nonlinear state space form:

$$\begin{aligned} x_k &= f_{k-1}\left(x_{k-1}, u_{k-1}, w_{k-1}\right) \\ y_k &= h_k\left(x_k, v_k\right) \end{aligned} \tag{4}$$

where $f_{k-1} : \mathbb{R}^{n_x+n_u+n_w} \rightarrow \mathbb{R}^{n_x}$ is the discrete state function, $x_k \in \mathbb{R}^{n_x}$ is the discrete state vector, $u_{k-1} \in \mathbb{R}^{n_u}$ is the discrete input, $w_{k-1} \in \mathbb{R}^{n_w}$ is the discrete process noise vector, $h_k : \mathbb{R}^{n_x+n_v} \rightarrow \mathbb{R}^{n_x}$ is the discrete output function, $v_k \in \mathbb{R}^{n_v}$ is the discrete measurement noise vector, $y_k \in \mathbb{R}^{n_y}$ is the output vector, and k is the time index. The noise vector sequences $\{w_{k-1}\}$ and $\{v_k\}$ are assumed Gaussian, white, zero-mean, uncorrelated, and have the known covariance matrices $Q_k \in \mathbb{R}^{n_x \times n_w}$ and $R_k \in \mathbb{R}^{n_y \times n_v}$.

3.2 Augmented states

The augmented state space approach can be directly used to simultaneously solve the state and the parameter estimation problem (e.g. see [4]). An augmented state space representation is formulated by adding the vector of parameters to be estimated $\theta_k \in \mathbb{R}^{n_\theta \times 1}$ as new states:

$$\begin{bmatrix} x_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} f_{k-1}\left(x_{k-1}, u_{k-1}, w_{k-1}^{(x)}\right) \\ \theta_{k-1} + T_s w_{k-1}^{(\theta)} \end{bmatrix} \tag{5}$$

$$y_k = h_k\left(x_k, v_k\right) \tag{6}$$

where T_s is the sampling time step, $w_{k-1}^{(x)} \in \mathbb{R}^{n_w^{(x)}}$ is the process noise vector that affects the original states, and $w_{k-1}^{(\theta)} \in \mathbb{R}^{n_w^{(\theta)}}$ is the process noise vector that affects the added parameter states. The noise vector sequences $\{w_{k-1}\}$ and $\{v_k\}$ are assumed Gaussian, white, zero-mean, uncorrelated, and have the known covariance matrices $Q_k \in \mathbb{R}^{(n_x+n_\theta)\times(n_w^{(x)}+n_w^{(\theta)})}$ and $R_k \in \mathbb{R}^{n_y \times n_v}$

$$\begin{aligned} w_k &\sim \mathcal{N}(0, \mathrm{blkdg}(Q_k^{(x)}, Q_k^{(\theta)})) \\ v_k &\sim \mathcal{N}\left(0, R_k\right) \end{aligned}$$

During the propagation step, the augmented states corresponding to parameters θ_k are considered equal to the previous time step θ_{k-1} with some additive process noise $w_{k-1}^{(\theta)}$. If it is assumed that the parameters do not change at all, then there is no process noise vector $w_{k-1}^{(\theta)}$, but for the more general case of time-varying parameters (e.g. fouling, etc.), the value of $Q_k^{(\theta)}$ will be given by the admissible range of variation of θ_k. During the measurement update step the parameter values are corrected.

For notational simplicity in the estimators algorithms that follow, the augmented state vector is referred to as x_k, the state augmented function (5) is referred to as f_{k-1}, and the augmented process noise vector is referred to as w_{k-1}.

Tab. 5: UKF algorithm.

```
Initialization:
$L = n_x + n_w + n_v$, $\lambda = \alpha^2(L+\kappa) - L$
$\gamma = \sqrt[2]{L+\lambda}$, $\Lambda_m^0 = \lambda/(\lambda+L)$
$\Lambda_c^0 = \lambda/(\lambda+L) + (1-\alpha^2+\beta)$
   for $i = 1, 2, \ldots, 2L$
   $\Lambda_m^i = (2(\lambda+L))^{-1}$, $\Lambda_c^i = \Lambda_m^i$
$\hat{x}_{0|0} \sim \mathcal{N}(\bar{x}_0, P_0)$
$P_{0|0} = P_0$
for $k = 1, 2, \ldots$
Propagation step:
(sigma points propagation)
$\tilde{P}_{k-1|k-1} = \mathrm{blkdiag}(P_{k-1|k-1}, Q_k, R_k)$
$\tilde{x}^0_{k-1|k-1} = [(\hat{x}_{k-1|k-1})^T, 0_{1\times n_W}, 0_{1\times n_v}]^T$
   for $i = 1, 2, \ldots, L$
   $\tilde{x}^i_{k-1|k-1} = \tilde{x}^0_{k-1|k-1} + \gamma\,\mathrm{chol}(\tilde{P}_{k-1|k-1}, i)$
   $\tilde{x}^{i+L}_{k-1|k-1} = \tilde{x}^0_{k-1|k-1} - \gamma\,\mathrm{chol}(\tilde{P}_{k-1|k-1}, i+L)$
   $\tilde{x}^{(x)i}_{k|k-1} = f_{k-1}(\tilde{x}^{(x)i}_{k-1|k-1}, u_{k-1}, \tilde{x}^{(w)i}_{k-1|k-1})$
   $\tilde{y}^i_{k|k-1} = h_k(\tilde{x}^{(x)i}_{k|k-1}, \tilde{x}^{(v)i}_{k-1|k-1})$
( a priori state-output estimate)
$\hat{x}_{k|k-1} = \sum_{i=0}^{2L} \Lambda_m^i \tilde{x}^{(x)i}_{k|k-1}$
$\hat{y}_{k|k-1} = \sum_{i=0}^{2L} \Lambda_m^i \tilde{y}^i_{k|k-1}$
( a priori state covariance estimate)
$\tilde{e}^i_{x,k|k-1} = (\tilde{x}^{(x)i}_{k|k-1} - \hat{x}_{k|k-1})$
$P_{k|k-1} = \sum_{i=0}^{2L} \Lambda_c^i (\tilde{e}^i_{x,k|k-1})(\tilde{e}^i_{x,k|k-1})^T$
Measurement update:
(Kalman gain calculation)
$\tilde{e}^i_{y,k|k-1} = (\tilde{y}^i_{k|k-1} - \hat{y}_{k|k-1})$
$P_y = \sum_{i=0}^{2L} \Lambda_c^i (\tilde{e}^i_{y,k|k-1})(\tilde{e}^i_{y,k|k-1})^T$
$P_{xy} = \sum_{i=0}^{2L} \Lambda_c^i (\tilde{e}^i_{x,k|k-1})(\tilde{e}^i_{y,k|k-1})^T$
$K_k = P_{xy} P_y^{-1}$
( a posteriori state-covariance estimate)
$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - \hat{y}_{k|k-1})$
$P_{k|k} = P_{k|k-1} - K_k P_y K_k^T$
```

3.3 Nonlinear Recursive Estimators

The nonlinear estimation problem can be formulated as a recursive Bayesian estimation problem with a propagation and a measurement update step. This is the optimal way of predicting a state probability density function (pdf) $p\,(x_k)$ for any system in state space representation with process and measurement noise[2].

Tab. 6: EnKF algorithm.

```
Initialization:
(initial ensemble)
for $i = 1, 2, \ldots, N$
   $x^i_{0|0} \sim \mathcal{N}(\bar{x}_0, P_0)$
for $k = 1, 2, \ldots$
Propagation step:
(ensemble propagation)
for $i = 1, 2, \ldots, N$
   $x^i_{k|k-1} = f_{k-1}(x^i_{k-1|k-1}, u_{k-1}, w^i_{k-1})$
   $y^i_{k|k-1} = h_k(x^i_{k|k-1}, v^i_{k-1})$
(estimated state-output propagation)
$\hat{x}_{k|k-1} = (N)^{-1} \sum_{i=1}^N x^i_{k|k-1}$
$\hat{y}_{k|k-1} = (N)^{-1} \sum_{i=1}^N y^i_{k|k-1}$
(covariance calculation)
$e^i_{x,k|k-1} = (x^i_{k|k-1} - \hat{x}_{k|k-1})$
$P_{k|k-1} = (N-1)^{-1} \sum_{i=1}^N (e^i_{x,k|k-1})(e^i_{x,k|k-1})^T$
Measurement update:
(Kalman gain calculation)
$e^i_{y,k|k-1} = (y^i_{k|k-1} - \hat{y}_{k|k-1})$
$P_y = (N-1)^{-1} \sum_{i=0}^N (e^i_{y,k|k-1})(e^i_{y,k|k-1})^T$
$P_{xy} = (N-1)^{-1} \sum_{i=0}^N (e^i_{x,k|k-1})(e^i_{y,k|k-1})^T$
$K_k = P_{xy} {P_y}^{-1}$
(state-out-covariance update)
$x^i_{k|k} = x^i_{k|k-1} + K_k((y_k + v^i_k) - y^i_{k|k-1})$
$\hat{x}_{k|k} = (N)^{-1} \sum_{i=1}^N x^i_{k|k}$
$P_{k|k} = P_{k|k-1} - K_k P_y K_k^T$
```

Assuming that the initial state pdf $p\,(x_0)$, the process noise pdf $p\,(w_{k-1})$, and the measurement noise pdf $p\,(v_k)$ are known, a recursive solution of the estimation problem can be found using first the Chapman-Kolmogorov equation to calculate the *a priori* pdf for the state x_k based on the previous measurement y_{k-1} (propagation step)

$$p(x_k|y_{k-1}){=}\int p(x_k|x_{k-1})p(x_{k-1}|y_{k-1})dx_{k-1} \quad (7)$$

where $p\,(x_k|x_{k-1})$ can be calculated from the state function f_{k-1} and the pdf of the process noise w_k.

Secondly, the Bayes rule to update the pdf of the state x_k with the new measurement y_k (measurement update) is

[2] Markov process of order one.

$$p\left(x_k|y_k\right) = \frac{p\left(y_k|x_k\right) p\left(x_k|y_{k-1}\right)}{\int p\left(y_k|x_k\right) p\left(x_k|y_{k-1}\right) dx_k} \qquad (8)$$

where $p\left(y_k|x_k\right)$ is available from our knowledge of the output function h_k and the pdf of v_k, and $p\left(x_k|y_{k-1}\right)$ is known from (7). Although the initial state pdf $p\left(x_0\right)$, the process noise pdf $p\left(w_{k-1}\right)$, and the measurement noise pdf $p\left(v_k\right)$ are needed to solve the recursive Bayesian estimation, no specific statistical distribution is required.

The recursive relations (7) and (8) used to calculate the *a posteriori* pdf $p\left(x_k|y_k\right)$ are a conceptual solution and only for very specific cases can these be solved analytically. In general, approximations are required for practical problems. Three main groups of suboptimal techniques with significant performance and computational cost differences are used to approximate the recursive Bayesian estimation problem: the classical nonlinear extension of the Kalman filter (EKF), the Unscented Kalman filter (UKF), and the Ensemble Kalman filter (EnKF) approaches.

3.4 Extended Kalman Filter (EKF)

The discrete EKF is probably the most used sequential nonlinear estimator nowadays. It was originally developed as a nonlinear extension by Schmidt [5] of the seminal work of Kalman [6]. Based on the Kalman filter, it assumes that the statistical distribution of the state vector remains Gaussian after every time step[3] so it is only necessary to propagate and update the mean and covariance of the state random variable x_k. The main concept is that the estimated state (i.e. estimated mean of x_k) is sufficiently close to the true state (i.e. true mean of x_k) so the nonlinear state/output model equations can be linearized by a truncated first-order Taylor series expansion around the previously estimated state.

The discrete algorithm is given in table 4. In general, this algorithm works for many practical problems, but no general convergence or stability conditions can be established[4] and its final performance will depend on the specific case study. For highly nonlinear models with unknown initial conditions, the EKF assumptions may prove to be poor and the filter may fail or have a poor performance. The main tuning parameters are the estimator covariance matrices Q_k and R_k.

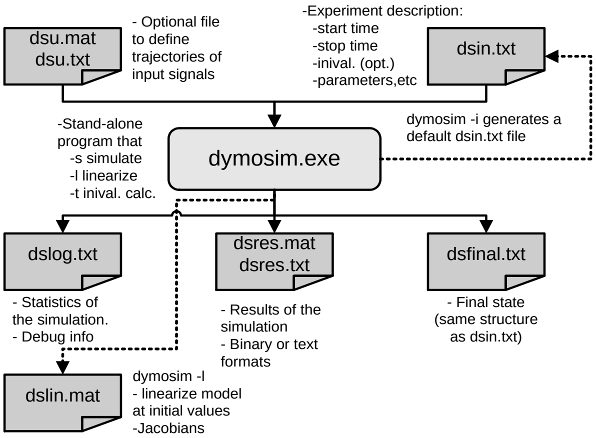
dsu.mat
dsu.txt
- Optional file to define trajectories of input signals
-Experiment description:
-start time
-stop time
-inival. (opt.)
-parameters,etc
dsin.txt
-Stand-alone program that
-s simulate
-l linearize
-t inival. calc.
dymosim.exe
dymosim -i generates a default dsin.txt file
dslog.txt
- Statistics of the simulation.
- Debug info
dsres.mat
dsres.txt
- Results of the simulation
- Binary or text formats
dsfinal.txt
- Final state (same structure as dsin.txt)
dslin.mat
dymosim -l
- linearize model at initial values
-Jacobians

Fig. 3: Dymosim and related input and output files.

3.5 Unscented Kalman Filter (UKF)

The unscented Kalman filter was originally developed by Julier and Uhlman [8, 9, 10, 11].

In the unscented Kalman filters, instead of approximating the nonlinear state/output functions, it is the probability distribution that is approximated. Basically, a set of points, called sigma points, are generated to match the state mean and state covariance of the probability distribution of the previously estimated state, then they are propagated through the nonlinear function. The projected points are used to approximate the first two moments (i.e. the *a priori* estimated state and state covariance) that are necessary during the measurement update step. This filter normally outperforms the previously presented EKF. Its more general form has a higher computational cost but it does not require the calculation of any Jacobian matrices (i.e. derivatives). The algorithm is given in table 5.

The tuning parameters of the UKF are also the estimator process and measurement noise covariance matrices, and the scalar parameters $\{\alpha, \kappa, \beta\}$: α determines the spread of the sigma points around the previous estimate, and the β value depends on the type of distribution assumed (for more details about their values see [11]).

3.6 Ensemble Kalman Filter (EnKF)

The EnKF uses an ensemble (i.e. particle set) during the propagation step, but the classical Kalman

[3] this assumption is in general not true for nonlinear systems.

[4] except for some special cases [7].

measurement update equations (instead of using the resampling with replacement approach of the particle filters) during the measurement update step. The covariances matrices P_{xy} and P_y obtained from the propagation of the ensemble elements through the nonlinear state-space are used to calculate the Kalman gain K_k. The *a posteriori* ensemble is calculated from the Kalman gain matrix and an artificially generated measurement particle set that is normally distributed with mean equal to the current measurement y_k and covariance equal to R_k. The *a posteriori* ensemble is used to calculate the *a posteriori* state and covariance estimate, and it is used for the next filter iteration of the algorithm. For details about the algorithm, see table 6. The EnKF was originally developed in [12] to overcome the curse of dimensionality in large scale problems (i.e. weather data assimilation). It is suggested in the literature [13] that ensembles (i.e. particle sets) of 50 to 100 are often adequate for systems with thousands of states, but no conclusive work has been done on this.

Besides the estimator process and measurement noise covariance matrices, the other tuning parameter for this filter is the number of ensemble elements.

3.7 Implementation

The fermentation model is written in Modelica and compiled in Dymola into a stand-alone executable file called Dymosim. The different estimators are implemented in Matlab from where Dymosim is sequentially called during the propagation step to project the state vector (i.e. integrate over the sampling time) in the estimator algorithms. The parameter state vector θ_k is directly propagated within the Matlab code so the original model does not need to be modified to include the parameter dynamic equations.

Within the Modelica model the input vector u_k, the process noise input vector w_k, and the parameter input vector θ_k must be defined. This can be done in the following way at the top level of the model:

```
model fermentation
...
input Real u_u1;  // define model inputs
input Real u_w1;  // define noise inputs
input Real u_p1;  // define param. inputs
...
parameter Real p_u1;
parameter Real p_w1;
parameter Real p_p1;
parameter Real p_i1;
equation
fluidBCv.u[1]=u_u1+p_u1;
reactor.basicVol.w[1]=u_w1+p_w1;
reactor.RG.p_mu1=u_p1+p_p1;
reactor.i_rho[1]=p_i1;

end fermentation;
```

Additionaly, the discrete EKF estimator requires the calculation of the discrete Jacobians $F_{k-1}, L_{k-1}, H_k, M_k$. This can be done calculating a linearized model around the previous state estimate defined by the operating point $op = [x_{k-1}^T, u_{k-1}^T, 0, \theta_{k-1}^T]^T$ with the following Matlab code:

```
eval( ['! dymosim ', '-l ','dsin.txt'] );
```

In the file “dsin.txt” (see Fig. 3) the operating point is defined using parameters and the initial state for every iteration. The calculated linearized model is written in the file “dslin.mat” and then it can be loaded into Matlab using the Dymola add-on function `tloadlin` which loads the matrices `A,B,C,D` and the string vectors `uname`, `yname`, and `xname`. These matrices correspond to

$$\begin{aligned}
A &= \left.\frac{\partial f(x,u,w)}{\partial x}\right|_{op} \\
B &= \left[\left.\frac{\partial f(x,u,w)}{\partial u}\right|_{op}, \frac{\partial f(x,u,w)}{\partial w}, \Big|_{op}, \left.\frac{\partial f(x,u,w)}{\partial \theta}\right|_{op}\right] \\
C &= \left.\frac{\partial h(x,u)}{\partial x}\right|_{op} \\
D &= \left.\frac{\partial h(x,u)}{\partial u}\right|_{op}
\end{aligned}$$

The parameter augmented state space discrete jacobians are approximated from the A, B, C, D matrices

$$\begin{aligned}
A^e &= \begin{bmatrix} A & B_{(:,n_u+n_w+1:\text{end})} \\ 0_{n_p \times n_x} & 0_{n_p \times n_p} \end{bmatrix} \\
F_{k-1} &= \left.\frac{\partial f_{k-1}}{\partial x_{k-1}}\right|_{op} \approx \exp\left(A^e \Delta t\right) \\
B^e &= \begin{bmatrix} B_{(:,n_u+1:\text{end})} & 0_{n_x \times n_p} \\ 0_{n_p \times n_x} & 1_{n_p \times n_p} \end{bmatrix} \\
L_{k-1} &= \left.\frac{\partial f_{k-1}}{\partial w_{k-1}}\right|_{op} \\
&\approx [I\Delta t + \tfrac{1}{2!}A^e \Delta t^2 + \tfrac{1}{3!}A^{e2}\Delta t^3 + \ldots]B^e \\
H_k &= \begin{bmatrix} C \; 0_{n_y \times n_x} \end{bmatrix} \\
M_k &= D
\end{aligned}$$

where n_u is the input vector dimension, n_w is the process noise vector dimension, and so on. For notation simplicity, the matrices in the previous equations use Matlab notation.

4 Results

Due to the lack of experimental measurements, simulated data sets from the model with the original kinetic reaction rates are generated. The system model is simulated for 1000 h and data samples are collected every 1 h. Because the transient response is relevant to parameter identification, step-like input sequences with high frequency content are used (see Fig.4). The initial state vector for the fermentation model is

$$\begin{aligned} x_0 &= \left[\dot{V}, \rho_{\mathrm{P}}, \rho_{\mathrm{X}}, \rho_{\mathrm{S}}, \rho_{\mathrm{O}_2}, T, T_J\right]^{T\prime} \\ &= [1000, 12.9, 0.9, 28.6, 3.9, 30.4, 26.9]^T \end{aligned}$$

The system model process and measurement noise vector sequences $\{w_{k-1}\}$ and $\{v_k\}$ are Gaussian, white, zero-mean, uncorrelated, and have constant covariance matrices

$$\begin{aligned} Q_k &= \mathrm{blkdiag}\left(Q_k^{(x)}, Q_k^{(\theta)}\right) \\ Q_k^{(x)} &= \mathrm{diag}\left([1000, 15, 2, 100, 5, 35, 30]\right) * 1\,\mathrm{E}-7 \\ Q_k^{(\theta)} &= \mathrm{diag}\left([1, 1, 1, 1, 1, 1, 1, 1]\right) * 1\,\mathrm{E}-7 \\ R_k &= \mathrm{diag}\left([15, 2, 100, 5, 35, 30]\right) * 2\,\mathrm{E}-3 \end{aligned}$$

A subset of 8 parameters $\theta = [\mu_1, K_{\mathrm{S},1}, K_{\mathrm{S},2}, k_{\mathrm{P},1}, k_{\mathrm{P},2}, Y_{\mathrm{SP}}, Y_{\mathrm{SX}}, Y_{\mathrm{OX}}]^T$ is estimated for every estimator (i.e., the EKF, the UKF, and the EnKF) using every reaction rate model (i.e., the *original*, the *elementary*, the simple, and the *intracellular* reaction kinetic models). The initial parameter values for every reaction rate model are adjusted to ensure that all simulation results give the same steady state values at initial time $t = 0$. The estimators inputs are equal to the system model inputs $u = [\dot{V}_{J,i}, \rho_{\mathrm{S},i}]^T$, and the measured outputs are $y = [\rho_{\mathrm{P}}, \rho_{\mathrm{X}}, \rho_{\mathrm{S}}, \rho_{\mathrm{O}_2}, T, T_J]^T$ (see Fig.4). The estimators are simulated for 1000 h with a sampling time of 1 h.

The estimators' initial state vectors are drawn from a normal distribution with mean and covariance equal to

$$\begin{aligned} \hat{x}_{0|0} &\sim \mathcal{N}([\bar{x}_0^T, \bar{\theta}_0^T]^T, \mathrm{blkdiag}\left(P_0^{(x)}, P_0^{(\theta)}\right)) \\ \bar{x}_0 &= [990, 13.9, 0.8, 27.6, 4.9, 27.4, 24.9]^T \\ \bar{\theta}_0 &= [1.49, 1.48, 1.23, 7.3, 1.19, 5.07, 4.55, 9.3]^T \\ P_0^{(x)} &= \mathrm{diag}(\left(0.125 * \bar{x}_{0|0}\right).\hat{\ }2) \\ P_0^{(\theta)} &= \mathrm{diag}(\left(0.125 * \bar{\theta}_{0|0}\right).\hat{\ }2) \end{aligned}$$

The UKF parameters are $\{\alpha, \kappa, \beta\} = \{1\,\mathrm{E}-3, 0, 0\}$, and the EnKF is evaluated for an ensemble of $N = 100$ elements. The estimators process and measurement noise sequences are Gaussian, white, zero-mean, uncorrelated, and have constant covariance matrices equal to the system model.

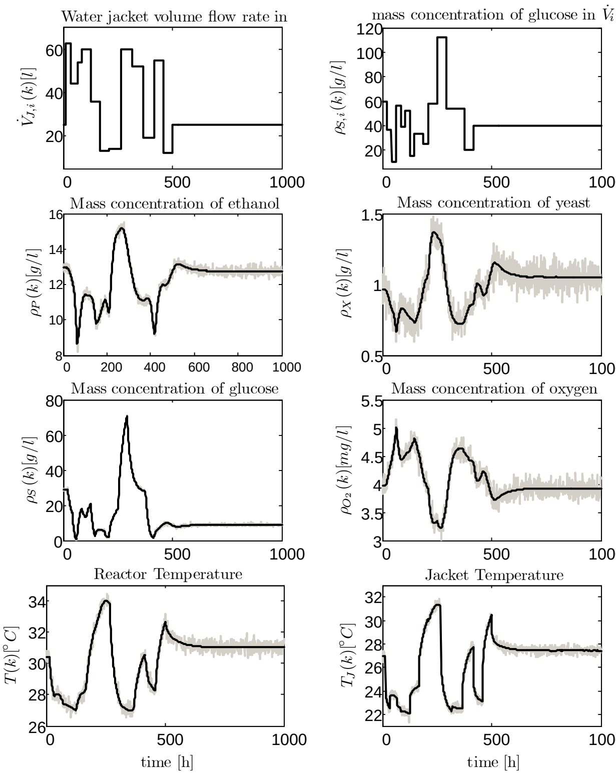

Fig. 4: Process inputs ($\dot{V}_{J,i}, \rho_{\mathrm{S},i}$), and measured outputs ($\rho_{\mathrm{P}}, \rho_{\mathrm{X}}, \rho_{\mathrm{S}}, \rho_{\mathrm{O}_2}, T, T_J$) with measurement noise (grey line) and without it (black line).

Every different reaction rate is evaluated for every estimator using 50 Monte Carlo simulations. As a general notation, consider an ensemble $\{x_j^i\,(k)\}$ where i indicates the realization, j the state/parameter, and k the time index. The ensemble average (over the realizations) is denoted $\langle x_j^i\,(k)\rangle$:

$$\langle x_j^i\,(k)\rangle \triangleq \frac{\sum_i x_j^i(k)}{n_{\mathrm{simul}}}$$

where n_{simul} is the number of realizations.

For every estimator with the different reaction rates two performance values (averaged over the number of Monte Carlo simulations) are calculated for each estimated parameter j: the *averaged estimated parameter* for every time index k that is used to evaluate the parameter estimation bias wrt. the true parameter value $\langle\hat{\theta}_j^i\,(k)\rangle$, and the *averaged absolute estimated parameter error* defined as $\langle |e^i_{\theta_j}(k)|\rangle \triangleq \langle|\hat{\theta}^i_j\text{-}\langle\hat{\theta}^i_j(k)\rangle|\rangle$ for every time index k. This second performance value is used to evaluate the convergence and consistency of every estimator.

The Monte Carlo averaged performance of the estimators using the *original* reaction rate model is shown in Fig. 5. The averaged estimated parameters $\langle\hat{\theta}_j(k)\rangle$ converge to the true parameters for all the parameters except for the slightly biased $\hat{Y}_{\mathrm{SP}}$ estimate and the more biased $\hat{Y}_{\mathrm{SX}}$ estimate. In Fig. 5 column (b), the averaged estimated parameter errors for the parameters $\{\hat{k}_{\mathrm{P},2}, \hat{Y}_{\mathrm{SP}}, \hat{Y}_{\mathrm{O}_2}\}$ converge at a faster rate than for the other estimated parameters $\{\hat{\mu}_1, \hat{K}_{\mathrm{S},1}, \hat{K}_{\mathrm{S},2}, \hat{k}_{\mathrm{P},1}, \hat{Y}_{\mathrm{SX}}\}$. The EKF and the UKF have comparable averaged estimated parameters, while the EnKF has slightly biased averaged estimated parameters. The best performance wrt. the averaged absolute estimated parameter error $\langle |e^i_{\theta_j}(k)|\rangle$ is achieved for the EKF followed by the UKF and the EnKF.

The Monte Carlo averaged performance of the estimators using the *elementary* reaction rate model is shown in Fig. 6. It can be seen that the averaged estimated parameters no longer converge to the true parameters of the *original* rate model used in the system model simulations. It is to be expected that some of the parameters will be time-varying to compensate for the different kinetic rates (between the system and the estimator kinetic rate models) and, in this way, keep a good state estimation performance besides their differences. For this case, the averaged estimated parameters $\langle\hat{\theta}_j(k)\rangle$ are considered as an unbiased estimate of the true (possibly time-varying) parameters. The averaged estimated parameters $\langle\hat{\theta}_j(k)\rangle$ take different shapes over time depending on the specific estimator evaluated. In Fig. 6 column (b), the averaged absolute estimated parameter errors $\langle |e^i_{\theta_j}(k)|\rangle$ for the parameters $\{\hat{k}_{\mathrm{P},1}, \hat{k}_{\mathrm{P},2}, \hat{Y}_{\mathrm{SX}}, \hat{Y}_{\mathrm{SP}}\}$ and the EKF diverge while the UKF achieves the best performance followed by the EnKF. It is then reasonable to consider that the averaged estimated parameters $\langle\hat{\theta}_j(k)\rangle$ that correspond to the UKF are the best estimate of the true parameters $\theta_j(k)$ for this estimator reaction rate model.

The Monte Carlo averaged performance of the estimators using the *simplified* reaction rate model is shown in Fig. 7. As for the *elementary* case, the averaged estimated parameters $\langle\hat{\theta}_j(k)\rangle$ are considered as an unbiased estimate of the true (possibly time-varying) parameters. The averaged estimated parameters $\langle\hat{\theta}_j(k)\rangle$ have similar values for the EKF and the UKF and slightly different for the EnKF. In Fig. 7 column (b), the lowest averaged absolute estimated parameter errors $\langle |e^i_{\theta_j}(k)|\rangle$ are achieved for the EKF, closely followed by the UKF performance. For all the estimators the averaged absolute estimated parameter errors decrease over time.

The Monte Carlo averaged performance of the estimators using the *intracellular* reaction rate model is shown in Fig. 8. As for the *elementary* and *simplified* cases, the averaged estimated parameters $\langle\hat{\theta}_j(k)\rangle$ are considered as an unbiased estimate of the true (possibly time-varying) parameters. The averaged estimated parameters $\langle\hat{\theta}_j(k)\rangle$ have similar values for the EKF and the UKF and slightly different for the EnKF. In Fig. 8 column (b), the averaged absolute estimated parameter errors $\langle |e^i_{\theta_j}(k)|\rangle$ decrease over time for all the parameters and estimators, except for the estimated parameter $\hat{Y}_{\mathrm{SX}}$ with the EnKF.

In Table 7 the different reaction rate models are evaluated for each filter using the normalized mean RMSE defined as

$$\overline{\mathrm{RMSE}\,(x)} = \sum_j^{n_x} \frac{\sum_i^{n_{\mathrm{simul}}} \sqrt[2]{\frac{\sum_k^{n_t}\left(\hat{x}_j(k) - x_j^{\mathrm{true}}(k)\right)^2}{n_t}}}{\max\left(x_j^{\mathrm{true}}\right) - \min\left(x_j^{\mathrm{true}}\right)}$$

Tab. 7: Normalized mean RMSE for the estimated state x and parameter θ vectors .The best results for every case is indicated by parentheses.

RMSE (.)		EKF	UKF	EnKF
Original	x	9.11E-2	9.13E-2	(8.44E-2)
	θ	(1.22)	1.74	1.95
Elementary	x	2.09E-1	1.00E-1	(9.35E-2)
	θ	5.57E-1	(2.73E-1)	7.54E-1
Simplified	x	8.36E-2	(8.05E-2)	8.77E-2
	θ	(3.21E-1)	3.56E-1	3.28E-1
Intracellular	x	8.94E-2	1.05E-1	(8.60E-2)
	θ	(3.98E-1)	5.69E-1	4.71E-1

5 Conclusions

The recursive parameter estimation problem is analyzed for an ethanol fermentation process with different reaction rate models. The model is implemented in Modelica and three nonlinear estimators are evaluated using the compiled Modelica model (Dymosim) with Matlab. Implementation details (e.g. how to calculate Jacobians, defined noise inputs, etc.) are presented.

Some relevant model parameters are estimated using the EKF, the UKF, and the EnKF from sim-

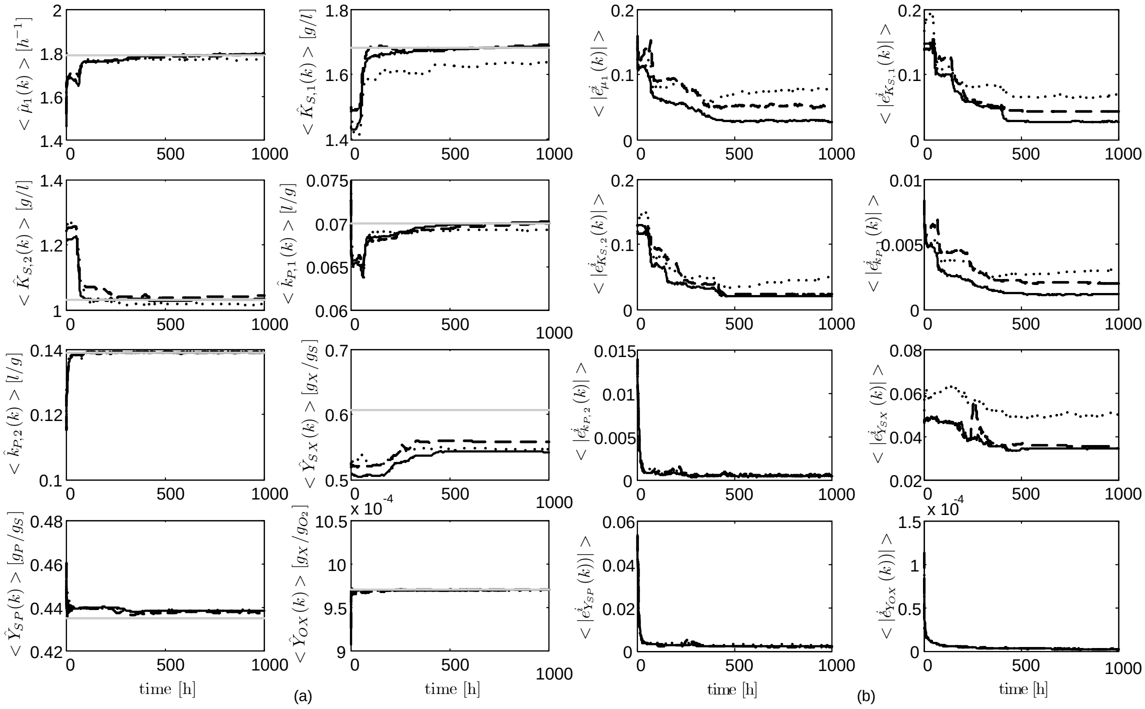

Fig. 5: *Original* kinetic rate model parameter estimation results, averaged over 50 Monte Carlo simulations for the EKF (black solid line), the UKF (black dash line), and the EnKF (black dotted line): (a) mean parameter estimates $\langle\hat{\theta}_j(k)\rangle$ for every time index k and true parameters θ_j (grey solid line); (b) mean absolute estimated parameter error, $\langle|e^i_{\theta_j}|\rangle = \langle|\hat{\theta}^i_j - \langle\hat{\theta}_j(k)\rangle|\rangle$ for every time index k.

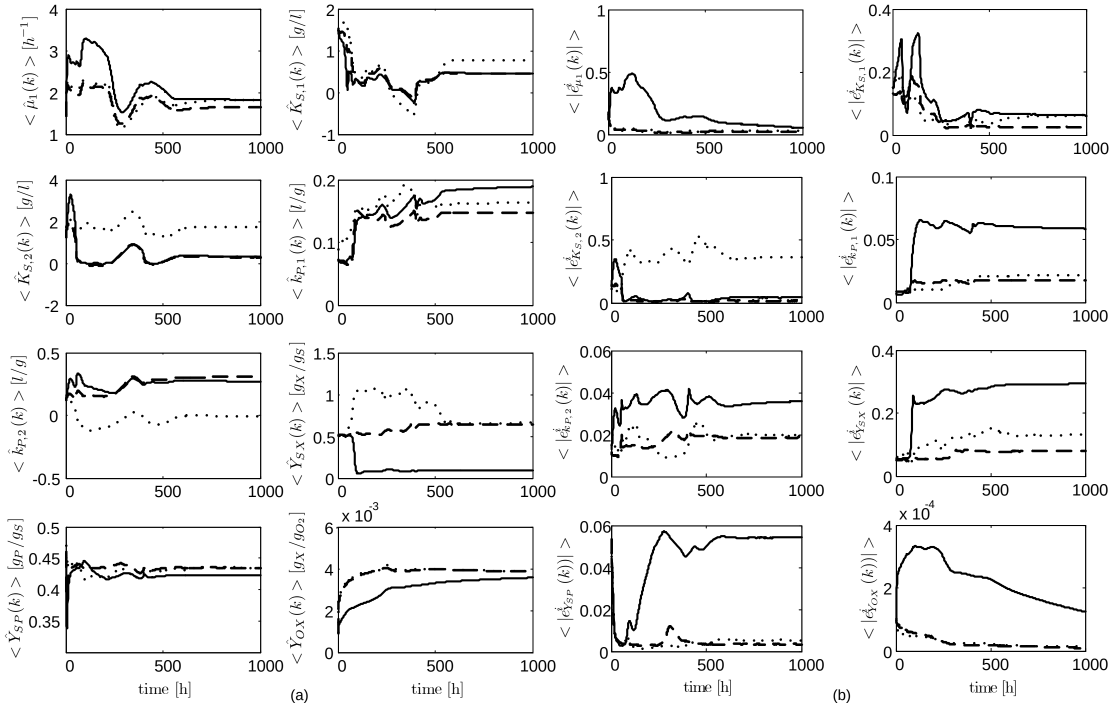

Fig. 6: *Elementary* kinetic rate model parameter estimation results, averaged over 50 Monte Carlo simulations for the EKF (black solid line), the UKF (black dash line), and the EnKF (black dotted line): (a) mean parameter estimates $\langle\hat{\theta}_j(k)\rangle$ for every time index k; (b) mean absolute estimated parameter error, $\langle|e^i_{\theta_j}|\rangle = \langle|\hat{\theta}^i_j - \langle\hat{\theta}_j(k)\rangle|\rangle$ for every time index k.

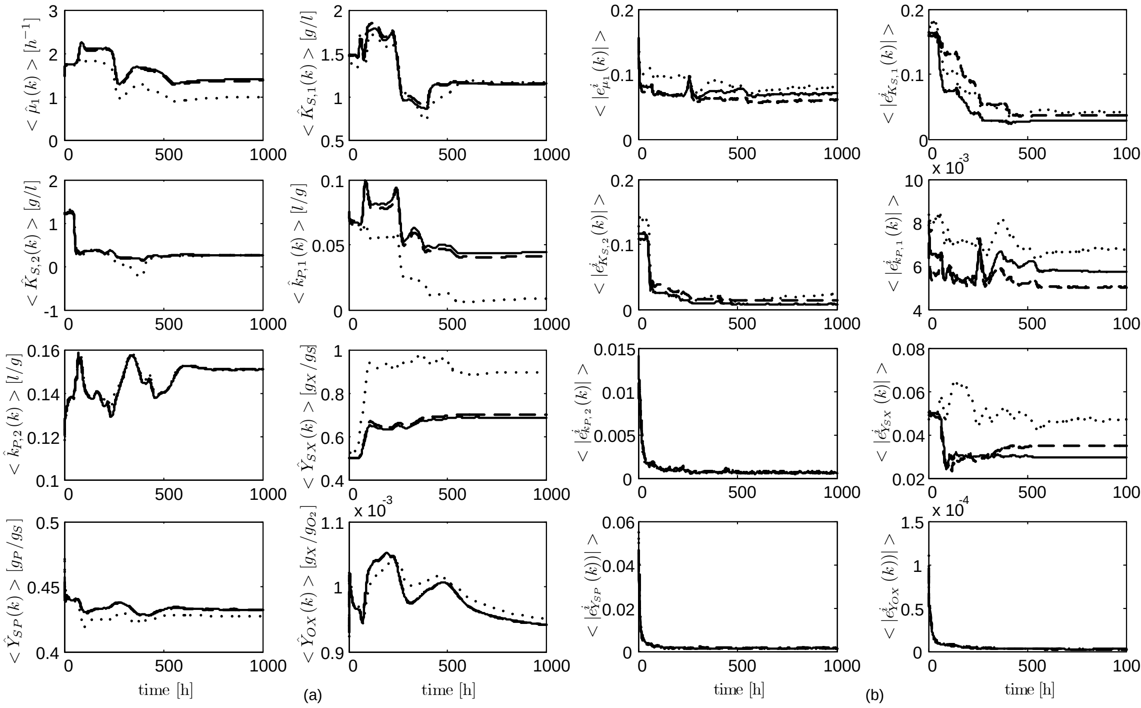

Fig. 7: *Simplified* kinetic rate model parameter estimation results, averaged over 50 Monte Carlo simulations for the EKF (black solid line), the UKF (black dash line), and the EnKF (black dotted line): (a) mean parameter estimates $\langle\hat{\theta}_j(k)\rangle$ for every time index k; (b) mean absolute estimated parameter error, $\langle|e^i_{\theta_j}|\rangle = \langle|\hat{\theta}^i_j - \langle\hat{\theta}_j(k)\rangle|\rangle$ for every time index k.

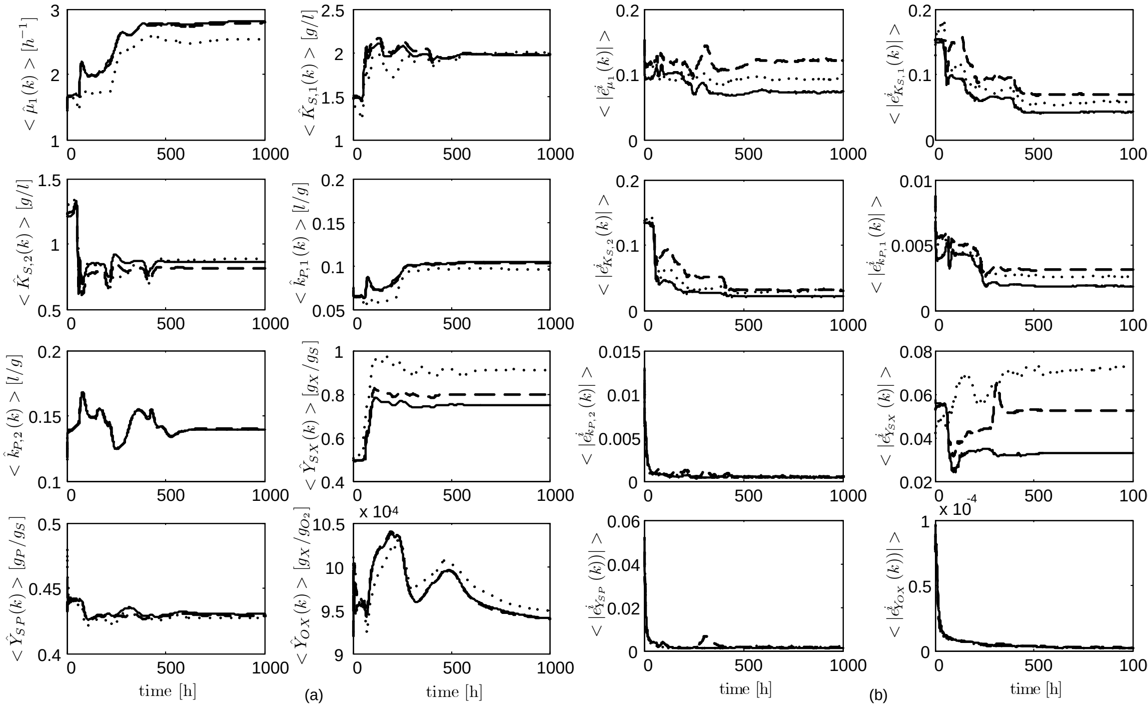

Fig. 8: *Intracellular* kinetic rate model parameter estimation results, averaged over 50 Monte Carlo simulations for the EKF (black solid line), the UKF (black dash line), and the EnKF (black dotted line): (a) mean parameter estimates $\langle\hat{\theta}_j(k)\rangle$ for every time index k; (b) mean absolute estimated parameter error, $\langle|e^i_{\theta_j}|\rangle = \langle|\hat{\theta}^i_j - \langle\hat{\theta}_j(k)\rangle|\rangle$ for every time index k.

ulated data sets over 50 Monte Carlo simulations. Four different reaction rate models are used by the estimators while the simulated data sets are generated assuming that the *original* reaction rate parameters have been estimated experimentally. When using the *original* reaction rate model in the estimator, the best parameter estimation is achieved by the EKF with slightly poorer performances for the UKF and the EnKF. The lower performance of the UKF can be explained by the lack of tunig of its parameters. For the estimator using the *elementary* reaction rate model, the best parameter estimation corresponds to the UKF, while the EnKF has a poorer performance and the EKF diverges for some of the parameters. For the estimator with the *simplified* reaction rate model similar performances are achieved for the 3 estimators; the UKF slightly outperforms the other two. For the estimator with the *intracellular* reaction rate model, the best parameter estimation performance corresponds to the EKF.

The EnKF has a poor parameter estimation performance for most of the cases but when considering the mean RMSE of the estimated states it outperforms the other estimators for three of the four cases (see Table 7).

The computational cost of the estimators increases considerably from the EKF to the EnKF because of the number of projections required for every estimator iteration. The fermentation model is run from a Dymosim executable file and this slows down the computational performance of the estimators (i.e. the computational time required for every estimator interation) mainly because Dymosim uses a slow file input/output interface. Despite this practical disadvantage, nonlinear estimators can be evaluated with complex Modelica models in a simple way. Our future work will focus on the parameter identifiability of the complete model.

References

[1] P. S. Agachi, Z. K. Cristea, and A. Imre-Lucaci, *Model Based Control. Case Studies in Process Engineering.* Weinheim: Wiley-VCH Verlag GmbH&Co., 2006.

[2] B. Lie and J. I. Videla, "Continuous bioethanol production by fermentation," in *Green Energy with energy management and IT*, Stockholm, 2008.

[3] P. M. Doran, *Bioprocess Engineering Principles.* San Diego: Academic Press, 1995.

[4] J. L. Crassidis and J. L. Junkins, *Optimal estimation of dynamic systems*, ser. CRC applied mathematics and nonlinear science series. Chapman & Hall, 2000.

[5] S. F. Schmidt, *Application of State-Space Methods to Navigation Problems*, c.t. leondes ed. Academic Press, New York, San Francisco, London, 1966, vol. 3, pp. 293–340.

[6] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[7] D. Simon, *Optimal State Estimation – Kalman, H_∞, and Nonlinear Approaches.* Hoboken, New Jersey: John Wiley & Sons, Inc., 2006.

[8] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *Proceedings of the 1995 American Control Conference*, Seattle, WA, 1995, pp. 1628–1632.

[9] S. Julier and J. Uhlmann, "A general method for approximating nonlinear transformations of probability distributions," tech. rep., RRG, Dept. of Engineering Science, University of Oxford, Nov 1996, Tech. Rep., 1996.

[10] ——, "A new extension of the Kalman filter to nonlinear systems," in *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls, Orlando, FL*, 1997.

[11] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation (invited paper)," in *Proceedings of the IEEE*, vol. 92(3). IEEE Institute of Electrical and Electronics, 2004, pp. 401–422.

[12] G. Evensen, "The ensemble kalman filter: Theoretical formulation and practical implementation," *Ocean Dynamics*, vol. 53, pp. 343–367, 2003.

[13] S. Gillijns, O. B. Mendoza, J. Chandrasekar, B. L. R. D. Moor, D. S. Bernstein, and A. Ridley, "What is the ensemble kalman filter and how well does it work?" in *Proceedings of the 2006 American Control Conference*, Minneapolis, Minnesota, USA, June 2006.