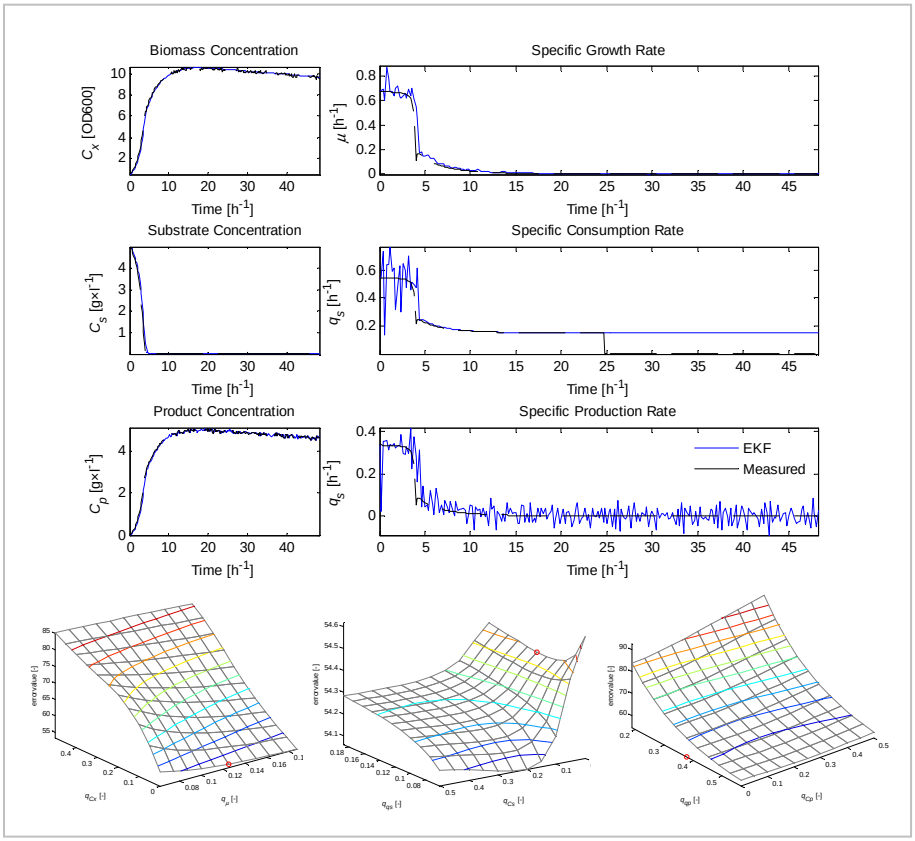
Systems and Control Group

BSc Thesis Systems and Control

Design of a cell culture monitoring system

J.E. Vuist

July 2010



WAGENINGEN UNIVERSITY
AGROTECHNOLOGY AND
FOOD SCIENCES

Design of a cell culture monitoring system

Name course	:	BSc Thesis System and Control
Number	:	SCO-80424
Study load	:	24 ECTS
Date	:	July 2010

Student	:	J.E. Vuist
Registration number	:	880227-921-040
Study programme	:	BBT (Biotechnology)

Supervisor(s)	:	dr.ir. A.J.B. van Boxtel
Examiners	:	Prof.dr.ir. G. van Straten,
Group	:	Systems and Control Group
Address	:	Bornse Weilanden 9 6708 WG Wageningen Tel: (0317) 48 21 24 Fax: (0317) 48 49 57

WAGENINGEN UNIVERSITY
AGROTECHNOLOGY AND
FOOD SCIENCES

Summary

In this work, the online monitoring system for cell activity in a novel disposable wave reactor is discussed. This reactor is equipped with sensors for biomass, glucose, lactate concentration, pH and dissolved oxygen. To monitor the cell activity in this reactor an online method is preferred because this leads to new opportunities to control the reactor.

The cell activity to be monitored concerns the specific biomass growth, the nutrient consumption and product formation rates. A traditional way to calculate the specific rates can be used, but these methods suffer from measurement noise that leads to an unreliable estimation. Two estimation methods look promising to estimate the specific rates.

The Extended Kalman Filter (EKF) is a method, which has settings to account for the measurement noise and system noise. The measurement noise can be found in the sensors specifications; the system noise can be used as a tool to tune the EKF's performance.

The Recursive Least Squares (RLS) is a method that is based on the Least Squares method, but is adapted to allow for time varying rates. It has only one tuning variable, which accounts for the 'memory' or the weight of the measurement versus the estimated values.

In this work the EKF proved to have superior performance over RLS. The EKF has shown that it can estimate the specific rate accurately with relative low amounts of disturbance by the measurement noise, despite the non-linear equations. The RLS has shown that it is incapable in the form used to handle the linear form of the non-linear equations and did not produced usable results on parameters beside the specific growth rate.

For the EKF a graphical user interface is constructed to make the usage more user-friendly and for demonstrations purposes.

Contents

1. Introduction

Cell culture is widely used in biotechnological and pharmaceutical production. The cultivation of cells is not limited to bacteria and yeast; it extends to all kinds of cells including animal and plant cells. These cells produce a variety of products. To maintain a constant quality the cultivation in a bioreactor is executed exactly the same for every batch. This means that the initial conditions are checked and the feed strategy is every time the same. Unfortunately biomass doesn't perform the same every time(Gnoth *et al.*, 2007); between the batches differences occur. These differences can lead to an inconsistent product quality.

The bioreactor considered in this work is a new disposable fed batch reactor that will be equipped with sensors for biomass, glucose, lactate, dissolved oxygen and pH. These sensors can give a good overview of what is happening in the bioreactor, but they do not give a good overview in what is happening with the organism. Quantities like growth rate, substrate consumption rate and product production rate give more information about the state of the organism. Unfortunately, there are no sensors available for these rates, so other techniques, like estimations, have to be used instead.

Traditional techniques to calculate the rates can be adapted to be used online. One of the drawbacks is that with decreasing time step the noise plays a major role in the measurement and the outcome is not a reliable rate. Another drawback is that if the time step chosen is too large, a sudden change in the parameter is not noticed. An example of such a equation(Neeleman, 2002) is given in equation (1.1). In this example, the growth rate (μ) is calculated for every measurement at time step t_k from the biomass concentration (C_x). In Figure 1 the result contains a lot of noise due to the small time step. In Figure 2 the result is smoother but if a sudden change in μ should occur, it can be undetected or delayed because the time step is large.

$$\mu = \frac{\ln\left(\frac{C_{x,k+1}}{C_{x,k}}\right)}{t_{k+1} - t_k} \qquad (1.1)$$

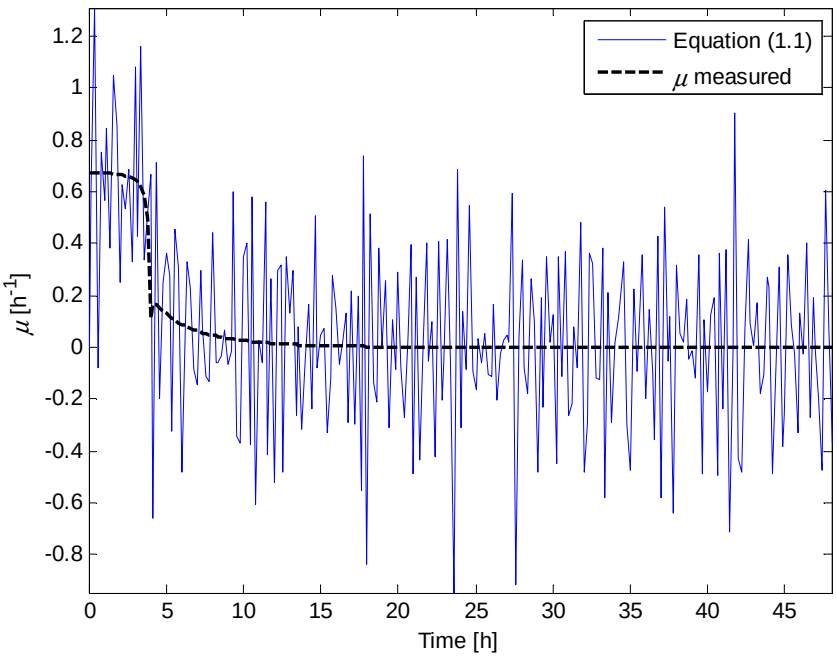


Figure 1: Result for equation (1.1) for a fed batch cultivation with a time step of 0.2 hours.

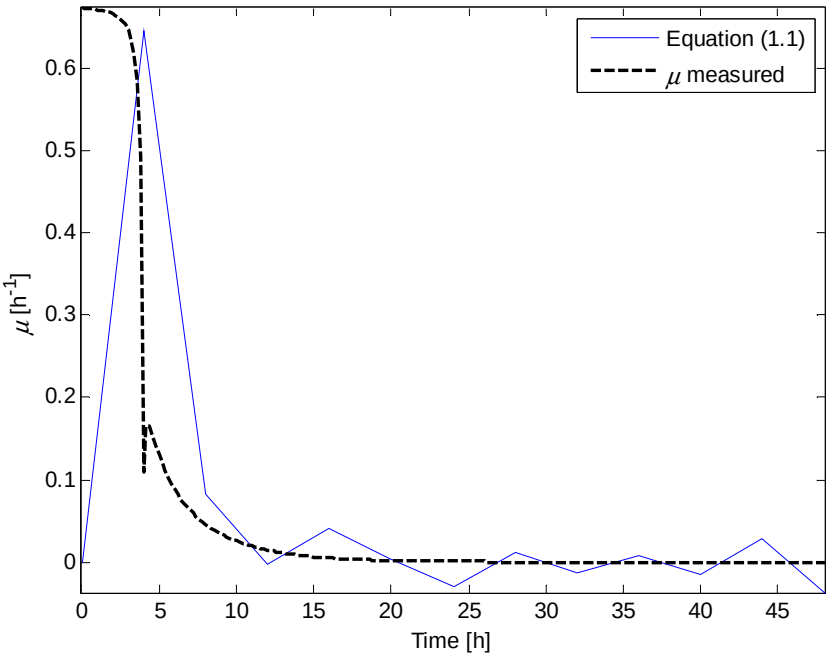


Figure 2: Result for equation (1.1) for a fed batch cultivation with a time step of 4 hours

To overcome the problem of noise alternative methods are developed. These methods use an observer to estimate the state of the system, in this work the Extended Kalman Filter (EKF) and the Recursive Least Squares (RLS) method will be considered. The EKF can be used to estimate growth, consumption and production rate from biomass, glucose and product measurements and has the advantage that the measurement noise and system noise can be used to tune the performance of the EKF. The EKF is used before by Rienksma(Rienksma, 2008) to model the growth, substrate consumption, oxygen consumption and penicillin production rate of *Penicillium chrysogenum.* There were problems with the estimation of the parameters, this was probably because many parameters are estimated at once which leads to a poor estimation.

The RLS method is an adaptation of the least squares method to be used recursively. This leads to an approach in which the least squares method can cope with variable parameters. The performance of this method can be tuned by a forgetting factor, which influences the importance of the measurements in the estimation. An advantage of the RLS method is that there is no need to know the measurement noise.

The focus of this thesis will lie on finding estimation methods that gives an accurate estimation of the desired parameters. To be able to check this experimental data will be generated with various levels of noise in the measurements and different patterns in the parameters. The EKF and the RLS method then will be tuned to reduce the error in comparison to the measurement data. Then the 'best' method will be selected to be used in an application description.

A graphical user interface (GUI) will be constructed to make the use of the selected method more user-friendly.

2. Cell growth

2.1. Mass balances

In general, mass balances will describe cell growth; this is done in the form:

Accumulation = In - Out + Production

Mass balances will be set up for every element present in the fermentation. Elements can be defined in two ways as true elements like carbon and oxygen or they can be defined as products like biomass, substrate and product. In this thesis, the experimental data will be modelled in the latter case.

There are basically three different ways to run a bioreactor, batch wise, fed batch wise and continuously(Bastin and Dochain, 1990). These methods differ in their in and out flow of fluids in the bioreactor, which leads to different opportunities to control the organism inside bioreactor.

Table 1 shows, for the different types of bioreactors, the mass balances. The dilution rate (D) can be chosen and gives an opportunity to control the bioreactor even further in compare to a batch reactor in which the only controls are the initial conditions.

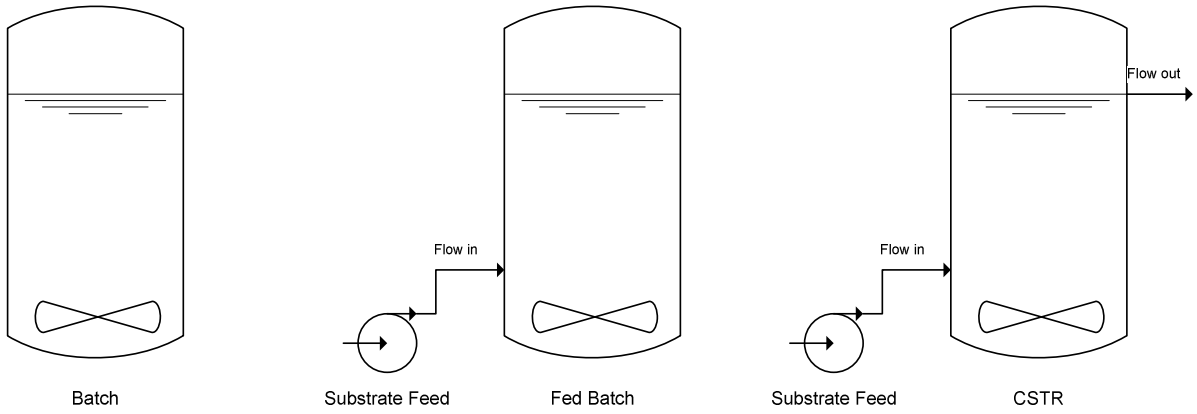


Figure 3: Overview of types of reactors

Table 1: Overview of mass balance equations for different types of bioreactors

Accumulation	In	-	Out	+	Production
Batch reactor					
$\frac{dV}{dt}$					
$\frac{dC_x}{dt}$					μC_x
$\frac{dC_s}{dt}$					$-q_s C_x$
$\frac{dC_p}{dt}$					$q_p C_x$
Fed batch reactor					
$\frac{dV}{dt}$	F_{in} or DV where $D = \frac{F_{in}}{V}$				
$\frac{dC_x}{dt}$					$\mu C_x - DC_x$
$\frac{dC_s}{dt}$	$DC_{s,in}$			+	$-q_s C_x - DC_s$
$\frac{dC_p}{dt}$					$q_p C_x - DC_p$
Continuous stirred tank reactor					
$\frac{dV}{dt} = 0$	F_{in} or DV where $D = \frac{F_{in}}{V}$	-	F_{out}		
$\frac{dC_x}{dt} = 0$		-	DC_x	+	μC_x
$\frac{dC_s}{dt} = 0$	$DC_{s,in}$	-	DC_s	+	$-q_s C_x$
$\frac{dC_p}{dt} = 0$		-	DC_p	+	$q_p C_x$

2.2. Specific growth rate (μ)

The specific growth rate (μ) is a property of the organism, is influenced by many chemical and biological factors, and is depended on factors like substrate, biomass, product, pH, temperature. The specific growth rate is modelled in many relations depending on the importance of the factors. Some of them will be presented in the following paragraph; the equations are summarized in Table 2.

The simplest is zero order kinetics, which means the substrate grows with its maximum constant specific growth rate (μ_{max}) until the substrate is depleted. The most wide spread model is the Monod law, which is derived from Michaelis-Menten kinetics and incorporates substrate concentration term in the model. This is sufficient if the temperature during the experiment is constant and there is not too much substrate present. To overcome this, the Monod law is extended to the Haldane law, which incorporates terms for substrate inhibition. If the inhibition term is neglected, the Haldane law reduces to the Monod law. Sometimes at high biomass concentrations, the biomass inhibits itself from growing faster. A simple model to explain this is the "logistic" model from Verhulst, where only the biomass concentration has an influence on the specific growth rate. Contois proposed a model that extended the Monod law in which substrate and biomass concentration is included. Product inhibition can also play a role in the specific growth rate, which is apparent in the production of ethanol by yeast, because most products are in higher concentration poisonous for the organism. An example from Aiba et al. is given in the table.

Table 2: Analytical models for the specific growth rate(Bastin and Dochain, 1990)

Model	Equation
Zero-order	$if\ (C_s > 0)$ $\lvert \mu = \mu_{\max}$ $if\ (C_s = 0)$ $\lvert \mu = 0$
Monod law	$\mu = \frac{\mu_{\max} C_s}{K_M + C_s}$
Haldane law	$\mu = \frac{\mu_0 C_s}{K_M + C_s + \frac{C_s^2}{K_I}}$ *where* $\mu_0 = \mu_{\max}\left(1 + \sqrt{\frac{K_M}{K_I}}\right)$
Verhulst	$\mu = \mu_{\max}(1 - aC_x)$ where a is an inhibition constant
Contois	$\mu = \frac{\mu_{\max} C_s}{K_C C_x + C_s}$
Aiba et al.	$\mu = \mu_{\max} e^{-K_P C_P}$

2.3. Specific substrate consumption rate (q_s)

The rate of substrate consumption (q_s, carbon source) is generally derived from Pirt's Law(Rinzema, 2009).

$$q_s = \frac{\mu}{Y_{xs}} + \frac{q_p}{Y_{ps}} + m_s \qquad (2.1)$$

In Pirt's Law, the substrate consumption rate is related to the growth and production rate, where it should be noted that the product is something not originating from the respiratory chain or growth related, and the maintenance of the cells (m_s). The true yield coefficients (Y_{xs}, Y_{ps}) do not depend on the concentration of the substrate or the products. Pirt's Law can also be adapted for oxygen consumption rate but taking measurements for estimating the parameters is a lot of work and measurement error tend to be high.

2.4. Specific product production rate (q_p)

The rate of product production (q_p) can be growth-associated or can be independent on the growth or a mixture of these two(Bastin and Dochain, 1990).

$$q_p = \frac{\mu}{Y_{px}} + c_p \qquad (2.2)$$

The first term is the growth-associated production rate and the second term is the independent production rate.

2.5. Data model

To have a basis to evaluate and test the estimation methods, data with known parameters has to be generated. This model is based on a 100L fed batch operated disposable wave bioreactor that for heating and cooling depends on the incubator, it is assumed that the incubator has the capacity to maintain a constant temperature. The volume of the bioreactor should not have an effect on the outcome of the tuning. It is also assumed that the pH is controlled in such a way that it remains more or less constant during the fermentation and has no influence on the growth rate. The oxygen transfer is assumed to be non-limiting because the disposable wave reactor is known for its high oxygen transfer(CELLutionBiotech, 2010). The organism is a virtual aerobic organism, which grows on glucose and produces one product. A scheme of the bioreactor is presented in Figure 4.

In order to test the performance of the observers a Monod, a block and a sine pattern with different maximum growth rates will be used for the growth rate, the consumption and production rate will be coupled to the growth rate a described in equations 2.1 and 2.2. The product produced is a result from the respiratory chain. The model is simulated in continuous time and the output of 48 h is sampled afterward with a time interval of 0.2 h. Different levels of noise (1% and 5%) will be applied to the biomass, glucose and product measurements, the noise will be applied afterwards with help of a Gaussian random number generator. The Monod pattern will be used for tuning and the sine and block pattern are for evaluating purpose only because they have illogical values for all variables.

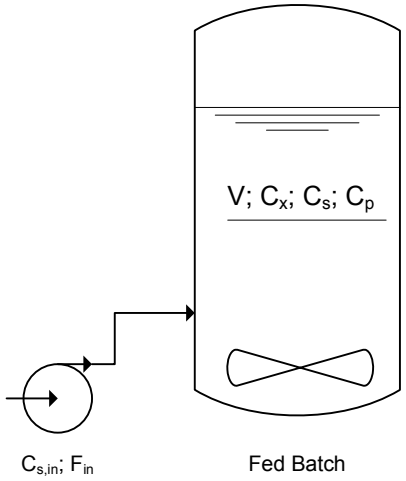


Figure 4: Schematic for the modelled bioreactor.

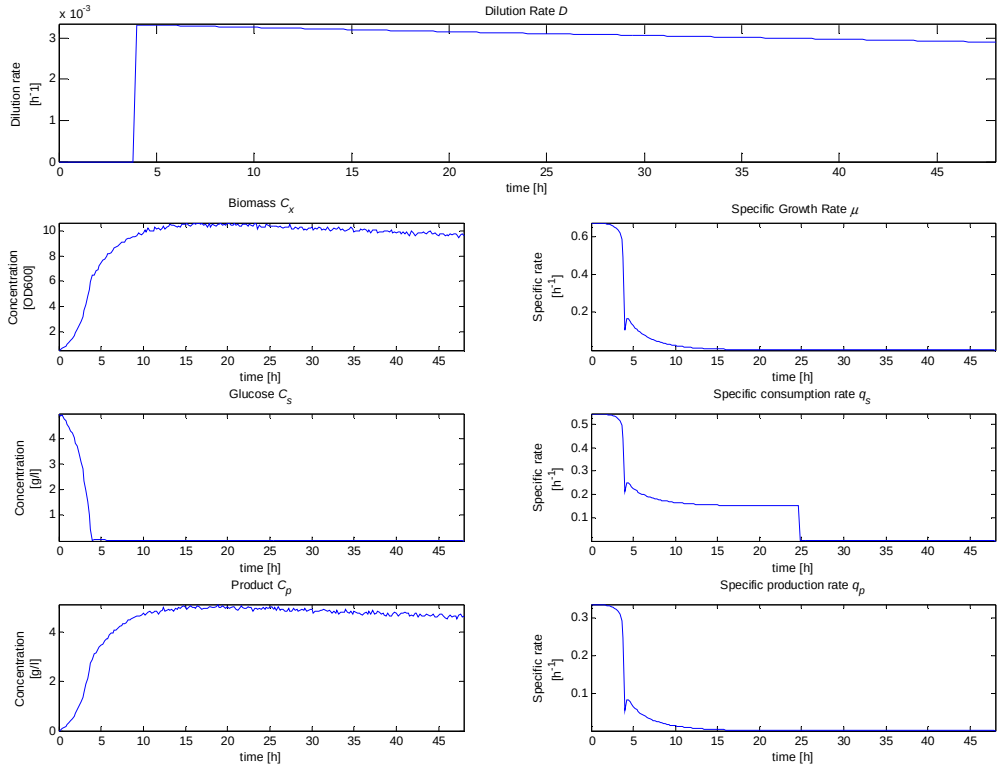


Figure 5: Monod growth pattern.

Table 3: Parameters for Monod simulation

Parameter	Value
μ_{max}	0.7 h^{-1}, 0.35 h^{-1}, 1.4 h^{-1}
K_s	0.2 g×L^{-1}
Y_{xs}	1.7 OD600×g^{-1} glucose
m_s	0.15 g glucose×$OD600^{-1}$×h
Y_{px}	2 g glucose×$OD600^{-1}$
$C_{s,in}$	500 g glucose×L^{-1}
F_{in}	0.02 L×h^{-1}

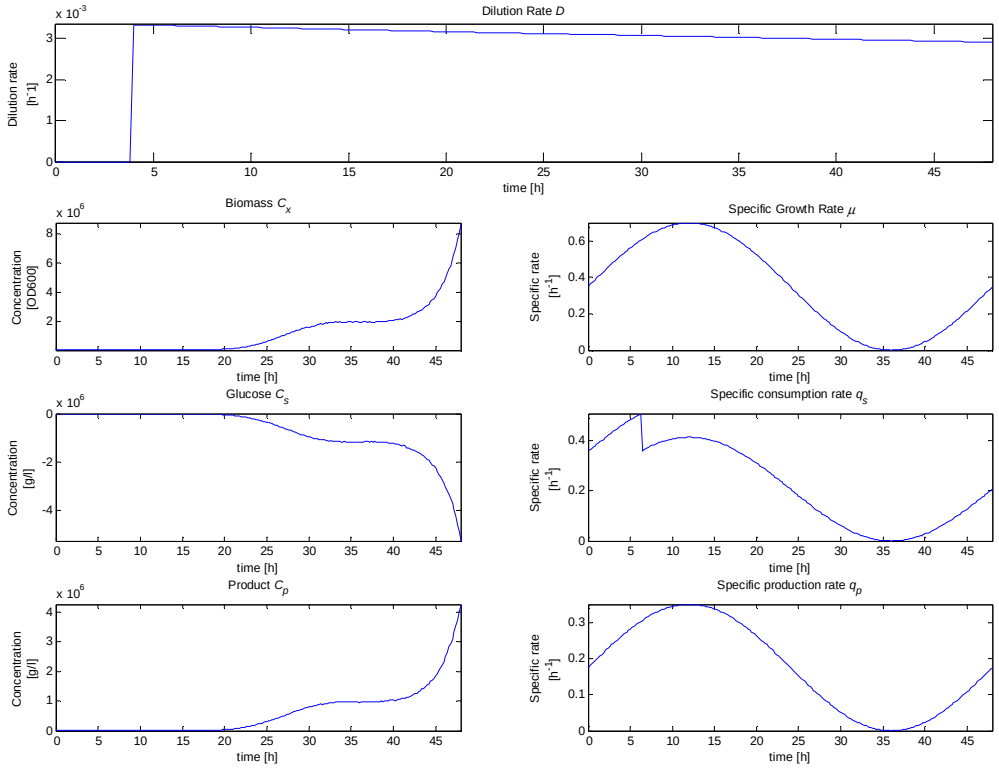


Figure 6: Sine growth pattern

Table 4: Parameters for sine simulation

Parameter	Value
μ_{max}	0.7 h^{-1}
μ *function*	$\mu_{\max} \times \sin(\frac{2\pi \times t}{48}) + 0.5 \times \mu_{\max}$
Y_{xs}	1.7 OD600×g^{-1} glucose
m_s	0.15 g glucose×$OD600^{-1}$×h
Y_{px}	2 g glucose×$OD600^{-1}$
$C_{s,in}$	500 g glucose×L^{-1}
F_{in}	0.02 L×h^{-1}

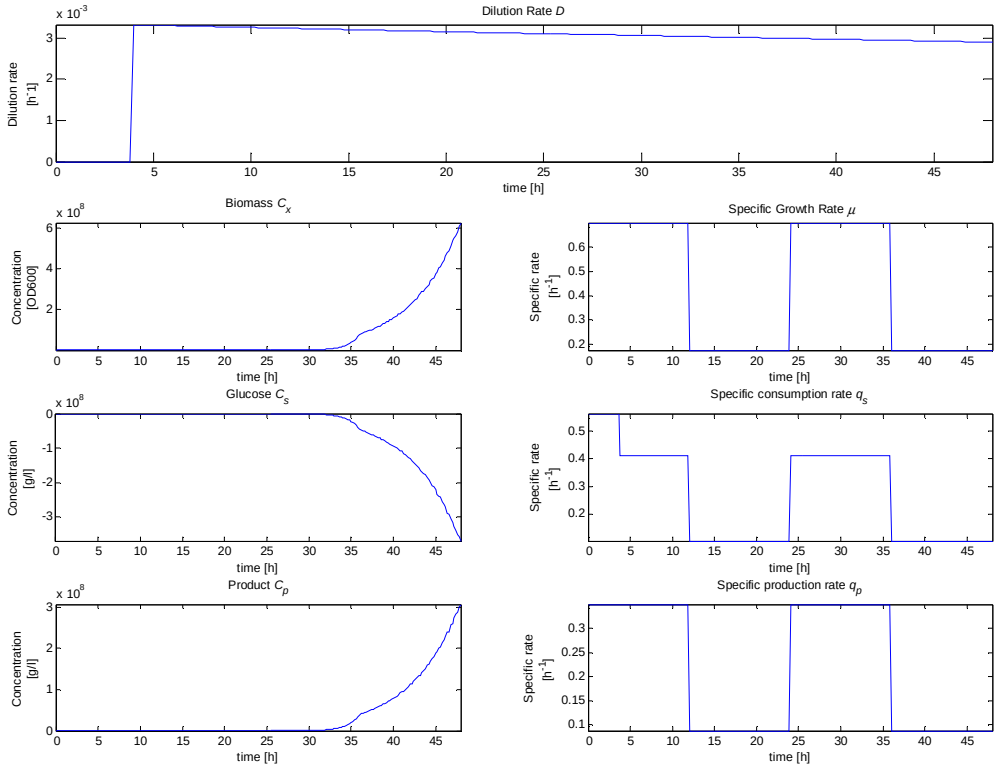


Figure 7: Block growth pattern

Table 5: Parameters for block simulation

Parameter	Value
μ_{max}	0.7 h^{-1}
μ *function*	$if\ 0<t<12 \vee 24<t<36$ $\lvert \mu_{\max}$ $else$ $\lvert 0.25\times\mu_{\max}$
Y_{xs}	1.7 OD600×g^{-1} glucose
m_s	0.15 g glucose×$OD600^{-1}$×h
Y_{px}	2 g glucose×$OD600^{-1}$
$C_{s,in}$	500 g glucose×L^{-1}
F_{in}	0.02 L×h^{-1}

2.6. External metabolic map

If all parameters are known these can be used to create an external metabolic map of the organism, in which its external properties, like growth and production rates, are known. This can be used to design a controller in such a way that the rates can be set to certain values. This can be an advantage in production because after a certain time with a constant production rate a known amount of product is produced.

3. Extended Kalman Filter

To reconstruct the production rates an observer, in the form of an EKF, is used. An explanation is given how this observer can reconstruct the parameters; furthermore, the observability conditions will be discussed.

3.1. State space representation

A system model can be represented by set of differential equations, which can be converted in a state space representation of the system. In a state space representation the model consist out of state variables (x), input variables (u), parameters (p) and output variables (y). A general state space representation is given below:

$$\begin{aligned} \dot{x}(t) &= f(x,u) \\ y(t) &= g(x) \end{aligned} \tag{3.1}$$

The first equation is the state equation and the second is the output equation. The function f can consist out of multiple equations. The system is called time varying if the functions f and g involve time (t). The state space representation of the modelled bioreactor is as follows. The parameters in μ, q_s and q_p are estimated as a whole and not split up in different parameters as in equations 2.1 and 2.2 or an equation from Table 2.

$$x(t) = \begin{pmatrix} C_x \\ C_s \\ C_p \end{pmatrix} \tag{3.2}$$

$$u(t) = \begin{pmatrix} D \end{pmatrix} \tag{3.3}$$

$$p = \begin{pmatrix} \mu \\ q_s \\ q_p \\ C_{s,in} \end{pmatrix} \tag{3.4}$$

$$\dot{x}(t) = \begin{pmatrix} p_1 \times x_1(t) - u_1(t) \times x_1(t) \\ p_4 \times u_1(t) - p_2 \times x_1(t) - u_1(t) \times x_2(t) \\ p_3 \times x_1(t) - u_1(t) \times x_3(t) \end{pmatrix} = \begin{pmatrix} \mu \times C_x - D \times C_x \\ D \times C_{s,in} - q_s C_x - D \times C_s \\ q_p \times C_x - D \times C_p \end{pmatrix} \tag{3.5}$$

Although V is measured it is not used in the state space representation, the volume is included in the input D. This is done to eliminate the need for estimating the volume, which leads to a better estimation because the estimator has fewer equations to handle.

To be of use in the EKF observer later the state space representation should be altered because of the interest in the parameters μ, q_s and q_p as a function of time. The parameters should be added to the state variables so they appear in the outcome of the observer.

$$x(t) = \begin{pmatrix} C_x \\ \mu \\ C_s \\ q_s \\ C_p \\ q_p \end{pmatrix} \tag{3.6}$$

$$u(t) = (D) \tag{3.7}$$

$$p = (C_{s,in}) \tag{3.8}$$

$$\dot{x}(t) = \begin{pmatrix} \mu \times C_x - D \times C_x \\ 0 \\ D \times C_{s,in} - q_s C_x - D \times C_s \\ 0 \\ q_p \times C_x - D \times C_p \\ 0 \end{pmatrix} \tag{3.9}$$

The reason that for the parameters of interest to have a changing value of zero is that there is no predicted change in them known in advance and so they are predicted with a general model.

3.2. Extended Kalman Filter

The EKF uses a linear and discrete time model for the estimation. However, the system mentioned in the previous paragraph is continuous and non-linear; therefore, these models should be put in linear and discrete form. The linearization is shown in Table 6. In equation 3.10 the linear form is shown, in equation 3.11 the discrete form of equation 3.10 is shown, w_k is the system noise and v_k is the measurement noise.

$$\begin{aligned} \dot{x}(t) &= A \times x(t) + B \times u(t) \\ y(t) &= C \times x(t) \end{aligned} \tag{3.10}$$

Becomes

$$\begin{aligned} x_{k+1} &= A_k x_k + B_k u_k + w_k \\ y_k &= C_k x_k + v_k \end{aligned} \tag{3.11}$$

With:

$$A_k = e^{AT} \tag{3.12}$$

$$B_k = \int_0^T e^{A\tau} \times B d\tau \tag{3.13}$$

$$C_k = C \tag{3.14}$$

$$w_k = (0, Q_k) \tag{3.15}$$

$$v_k = (0, R_k) \tag{3.16}$$

T is equal to the sampling interval. The matrix C_k is equal to the matrix C. The Q matrix defines the Gaussian system noise presented as a covariance matrix; the R matrix defines the Gaussian measurement noise presented as a covariance matrix. The system noise represents the uncertainty on the state variables; the measurement noise represents the uncertainty on the measurements that is due to stirring and measurement error on the sensors.

Table 6: General form matrices EKF

Matrix	General form	Size
A	$\begin{bmatrix} \frac{\partial f_1(x,u)}{\partial x_1} & \frac{\partial f_1(x,u)}{\partial x_2} & \cdots & \frac{\partial f_1(x,u)}{\partial x_n} \\ \frac{\partial f_2(x,u)}{\partial x_1} & \frac{\partial f_2(x,u)}{\partial x_2} & \cdots & \frac{\partial f_2(x,u)}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m(x,u)}{\partial x_1} & \frac{\partial f_m(x,u)}{\partial x_2} & \cdots & \frac{\partial f_m(x,u)}{\partial x_n} \end{bmatrix}$	m = number of equations n = number of state variables $m = n$
B	$\begin{bmatrix} \frac{\partial f_1(x,u)}{\partial u_1} & \frac{\partial f_1(x,u)}{\partial u_2} & \cdots & \frac{\partial f_1(x,u)}{\partial u_p} \\ \frac{\partial f_2(x,u)}{\partial u_1} & \frac{\partial f_2(x,u)}{\partial u_2} & \cdots & \frac{\partial f_2(x,u)}{\partial u_p} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m(x,u)}{\partial u_1} & \frac{\partial f_m(x,u)}{\partial u_2} & \cdots & \frac{\partial f_m(x,u)}{\partial u_p} \end{bmatrix}$	m = number of equations p = number of input variables
C	$\begin{bmatrix} \frac{\partial g_1(x,u)}{\partial x_1} & \frac{\partial g_1(x,u)}{\partial x_2} & \cdots & \frac{\partial g_1(x,u)}{\partial x_n} \\ \frac{\partial g_2(x,u)}{\partial x_1} & \frac{\partial g_2(x,u)}{\partial x_2} & \cdots & \frac{\partial g_2(x,u)}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_q(x,u)}{\partial x_1} & \frac{\partial g_q(x,u)}{\partial x_2} & \cdots & \frac{\partial g_q(x,u)}{\partial x_n} \end{bmatrix}$	q = number of measured state variables n = number of state variables
Q	$\begin{bmatrix} Q_{x_1} & 0 & \cdots & 0 \\ 0 & Q_{x_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Q_{x_n} \end{bmatrix}$	n = number of state variables
R	$\begin{bmatrix} R_{y_1} & 0 & \cdots & 0 \\ 0 & R_{y_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_{y_q} \end{bmatrix}$	q = number of measured state variables

The used EKF algorithm (Neeleman, 2002; Lewis, 1986) is given in equations 3.17, 3.18, 3.19, 3.20 and 3.21. The algorithm has two steps. In the first step, a time update is executed where a one sample ahead prediction of the states and the covariance matrix P is calculated. The prediction is calculated with a non-linear numerical integrator function from MATLAB. Then a predicted covariance matrix P is calculated with the previous updated covariance, system matrix A_k and the Gaussian system noise, which is defined in the Q matrix. The second step consists out of a measurement update in which the difference between the measurements and the estimated value that is weighed by the Kalman gain matrix and this is added to the estimate of the Time Update.
Time Update or Prediction

$$\bar{x}_{k+1} = \hat{x}_k + \int_{t_k}^{t_{k+1}} f(\hat{x}_k, u_k)dt \qquad (3.17)$$

$$\bar{P}_{k+1} = A_k \hat{P}_k A_k{}^T + Q_k \qquad (3.18)$$

Measurement Update or Correction

$$K_{k+1} = \bar{P}_{k+1} C^T (C\bar{P}_{k+1} C^T + R_{k+1})^{-1} \qquad (3.19)$$

$$\hat{P}_{k+1} = (I - K_{k+1} C)\bar{P}_{k+1} \qquad (3.20)$$

$$\hat{x}_{k+1} = \bar{x}_{k+1} + K_{k+1}(y_{k+1} - C\bar{x}_{k+1}) \qquad (3.21)$$

3.3. Observability

A system is observable if every state can be determined from the available outputs over a set time interval. The observability(Ogata, 2002) depends on the amount of measured outputs and on the amount of reconstructed states and parameters. The system is observable if the matrix O has full rank, if not the amount of measured states should be increased or the amount of parameters should be lowered.

$$O = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \qquad (3.22)$$

If all states are measured the system is always observable because the matrix C has already full rank, if not all states are measured it depends on the outcome of matrix A if the system is observable.

3.4. Tuning

Tuning of the EKF is done to have an optimal tracking of the state variables. The tuning involves choosing the matrices Q and R, which are considered a degree freedom for the designer. Choosing R is relatively simple because it is measurement noise, the measurement noise can be found by calculating the deviation of sample measurements or in the specification of the measurement apparatus. Finding Q is more difficult because it is often not known in advance. One technique in finding Q is to run an MATLAB function like fmincon to find an optimal value for Q, the function should find a minimal deviation between the true values and the modelled values. To calculate this deviation a squared error is calculated. The model data from the simulation is subtracted from the reconstructed values. This is done for every variable. To account for the differences in magnitude between the variables, every error is divided by its mean.

$$E = \sum_{n=1}^{N} \frac{\sqrt{\sum_{k=1}^{K} [\hat{x}_n(k) - x_{n,\mathrm{mod}}(k)]^2}}{\frac{\sum_{k=1}^{K} x_{n,\mathrm{mod}}(k)}{K}} \qquad (3.23)$$

4. Recursive Least Squares Parameter Estimation

Instead of an EKF, a recursive estimation algorithm can be used to determine the desired production rates. Recursive estimators use part of the past data to reconstruct the next data point. The behaviour is controlled by a forgetting factor, which determines how many data from the past is used to determine the next estimate. For the recursive estimation, different methods exist. Two methods are discussed. The RLS(Ljung and Söderström, 1987) and the forgetting algorithm described by Dochain et al.(Dochain and Bastin, 1984). The advantage of these types of algorithm is that the tuning is done in a single parameter instead of whole covariance matrices; another advantage is that the measurement noise does not have to be known in advance. The disadvantage is that it only can handle linear models only.

4.1. General formulation

The RLS algorithm is based on the least squares method, the goal of this method is to minimize the equation error by adjusting the parameters accordingly. In the recursive algorithm, the parameters are estimated for every data point thus if the parameter varies in time it will show up in the algorithm. The equations(Ljung and Söderström, 1987) below describe the RLS algorithm.
Suppose:

$$\hat{y}(k) = \varphi(k)\hat{\Theta}(k) \tag{4.1}$$

Then the RLS formulation will be:

$$\hat{\Theta}(k) = \hat{\Theta}(k-1) + L(k)[y(k) - \hat{\Theta}^T(k-1)\varphi(k)] \tag{4.2}$$

$$L(k) = \frac{P(k-1)\varphi(k)}{\lambda + \varphi^T(k)P(k-1)\varphi(k)} \tag{4.3}$$

$$P(k) = P(k-1) - \frac{P(k-1)\varphi(k)\varphi^T(k)P(k-1)}{\lambda + \varphi^T(k)P(k-1)\varphi(k)} \tag{4.4}$$

The first equation calculates the new parameter estimate from the previous estimate by adding the weighed differences between the true data (*y(t)*) and the data point from the model. The second equation calculates the weighing factor in which a forgetting factor (λ) is included to tune the behaviour of the algorithm. The third equation calculates the covariance; this is the degree of freedom in which the parameter can change. Θ is a vector, which contains the estimated parameters, due to first-order Euler approximation of the system the vector divided by the result of the approximation. The first-order Euler approximation is described in the equations below.
Consider

$$\dot{x}(t) = f(x,u) \tag{4.5}$$

Linearization

$$\dot{x} = Ax + Bu \tag{4.6}$$

Approximation

$$x(k) = (1 + AT)x(k-1) + BTu(k-1) \tag{4.7}$$

φ is a matrix containing the measurements but they are shifted back in time, for every quantity that is measured, there is a column.
The initial conditions for the algorithm can be estimated as shown in Ljung(Ljung and Söderström, 1987) but the algorithm quickly converges to the true value so the initial values can be guessed instead.

4.2. Recursive Least Squares Algorithm by Dochain et al.

The algorithm by Dochain(Dochain and Bastin, 1984) is based on the RLS algorithm. Given is the state space representation based on the fed batch equations from chapter 2.

$$\begin{aligned} \dot{C}_x &= \mu C_x - DC_x && = [\mu - D]C_x \\ \dot{C}_s &= DC_{s,in} - k_1 \mu C_x - DC_s && = -k_1 \mu C_x + D(C_{s,in} - C_s) \\ r_p &= k_2 \mu C_x && = k_2 \mu C_x \end{aligned} \qquad (4.8)$$

Note r_p, which is the non-specific production rate. This system is put in discrete time by using a first-order Euler approximation for C_x, C_s and r_p, with a sampling period T. This gives the following equations:

$$\begin{aligned} C_x(k+1) &= C_x(k) + T\mu(k)C_x(k) - TD(k)C_x(k) + v_1(k) \\ C_s(k+1) &= C_s(k) + Tk_1\mu(k)C_x(k) - TD(k)[C_{s,in} - C_s(k)] + v_2(k) \\ r_p(k) &= k_2\mu(k)C_x(k) \end{aligned} \qquad (4.9)$$

By assuming that

$$r_p(k+1) - r_p(k) = k_2\mu(C_x(k+1) - C_x(k)) \qquad (4.10)$$

Then substituting C_x by r_p

These equations are the basic discrete time model that can be used in the parameter estimation algorithm. In the model $v_x(t)$ represent errors due to noise, discretization and approximation. Since this model is linear in the parameters, a recursive least squares estimates can be obtained.

$$\begin{aligned} \hat{\mu}(k+1) &= \hat{\mu}(k) + TP(k)r_p(k)[r_p(k+1) - r_p(k) + TD(k)C_{s,in} - T\hat{\mu}(k)r_p(k)] \\ \hat{k}(k+1) &= \hat{k}(k) + TP(k)r_p(k)[C_s(k+1) - C_s(k) - TD(k)[C_{s,in} - C_s(k)] - T\hat{k}(k)r_p(k)] \end{aligned} \qquad (4.11)$$

$$with\ \hat{k} = -\frac{k_1}{k_2}$$

$$P(k) = \frac{P(k-1)}{\lambda}\left(1 - \frac{T^2 r_p(k)^2 P(k-1)}{\lambda + T^2 r_p(k)^2 P(k-1)}\right) \qquad (4.12)$$

$$\text{with } P_0 >> 0 \text{ and } 0 < \lambda \leq 1$$

λ is a forgetting factor to allow the tracking of the time-varying parameters and accounts for the variations that occur in these parameters.

4.3. Tuning

Tuning of these methods can be done in the values of P_0 and λ. Tuning in P_0 is for both methods not a feasible way unless the parameters estimated are constant, because P_0 determines the range of variance for the values of the estimated parameters. If P_0 is chosen to small the values of the estimates become locked onto a single value and have a hard time tracking changes, therefore P_0 is generally chosen large, for example 10^4 to overcome this problem. λ is the forgetting factor, in the general algorithm the factor is allowed to have a value between zero and one and in the algorithm adapted by Dochain the value should be between zero and one. The factor is the balance between tracking of the estimate of the tracking of the measurements. The tuning is executed in the same way as for the EKF, by means of the equation error (Equation 3.23).

5. Estimation of the parameters

The quality of a estimation method is defined as its ability to reconstruct the metabolic parameters correctly. To evaluate the methods estimation results are compared to the known parameter used for the simulated data. To obtain the best the estimation, thus lowering the error, the matrix Q_k for the EKF is optimized by "fmincon" and for the RLS the value of λ is optimized by "fmincon".

5.1. Extended Kalman Filter

To reconstruct the metabolic parameters with the EKF an incremental approach is taken, instead of directly using the whole system from equation 3.9. This is done because initial results show a poor estimation when all states are reconstructed at once. With the incremental method, first the system is optimized for the equation for biomass and specific growth rate, second the system is optimized for biomass, specific growth rate, substrate and specific substrate consumption rate, and finally the whole system is reconstructed and optimized.

5.1.1. Choosing the measurement noise matrix

The measurement error matrix R_k is chosen to be equal to the applied noise on the measurements, when the EKF is applied to real measurements the measurement error from the measuring devices should be applied. In 5.1 and 5.2, the measurement error matrices are shown for the 1% and the 5% noise case.

$$R_k = \begin{bmatrix} (0.01 \cdot C_x)^2 & 0 & 0 \\ 0 & (0.01 \cdot C_s)^2 & 0 \\ 0 & 0 & (0.01 \cdot C_p)^2 \end{bmatrix} \quad (5.1)$$

$$R_k = \begin{bmatrix} (0.05 \cdot C_x)^2 & 0 & 0 \\ 0 & (0.05 \cdot C_s)^2 & 0 \\ 0 & 0 & (0.05 \cdot C_p)^2 \end{bmatrix} \quad (5.2)$$

5.1.2. Choosing the boundaries for the system noise matrix

The system noise will be used as a tool to tune the EKF. Because not every value for the system noise will make sense, boundaries in the range where the optimization method should search were given. Values used were percentages from the estimated value. The lower boundary was easily chosen this should be zero for the noise on all states, because the values are squared and a negative value squared is the same as its positive counterpart. For the upper boundary two cases were defined, a case where the upper values were restricted to 0.1% for the measured variables and 100% for the estimated variables and a case where the upper values are restricted 1000% for all states. In the first case, the tracking behaviour of the EKF tends to be better and to the estimation of the measured values more filtering is applied. The latter case the filtering on the measured values should be less and sudden changes in the parameters should be detected earlier. In the optimization, both cases will be compared.

5.2. Recursive Least Squares

To reconstruct the parameters with RLS, the general formulation will be used. This because the formulation by Dochain (Chapter 4.2) has proven too inadaptable to the system used (Equation 3.9). This is due to assumption made on the parameters and on the covariance in chapter 4.2. The assumption on the parameters prevents the estimation of the specific consumption and specific production rate separately, because the yield constants are estimated as a fraction of each other. The covariance is calculated for the production rate and not for the other states. In the general formulation, all the parameters will be estimated at once and λ will be optimized.

5.2.1. Setting up the Recursive Least Squares

To use the RLS the contents of φ should be known. The contents of φ can be found in the linear state space equations (3.10), because in all equations the parameters are depended on C_x. A row is present for every parameter.

$$\varphi = \begin{bmatrix} C_x(k-1) & C_x(k-1) & C_x(k-1) \end{bmatrix} \tag{5.3}$$

Θ contains the rest of the linear equations, to obtain the desired parameters Θ should be adapted afterwards to exclude the linearization and show the parameters to compare then to the simulated data.

$$\Theta = \begin{bmatrix} 1+\mu T \\ -q_s T \\ q_p T \end{bmatrix} \tag{5.4}$$

6. Results of the estimations

The results obtained from the estimations are given in this chapter. Simulated data is used simulated according to Chapter 2.5. A table with all the results for the EKF can be found in Appendix A.

6.1. Extended Kalman Filter with low boundaries

Figure 8 shows the equation error (3.23) for the different optimization and evaluation. The optimization is executed on the Monod dataset with 1 % noise on the measurement the rest of the datasets are used for validation. Figure 9 shows the summarized results for the equation error for different values of μ_{max}, the optimization concerns μ_{max} = 0.7 h^{-1}. For both optimizations, the upper limit is shown in 6.1.

$$Q_{up} = \begin{bmatrix} (1\times10^{-3}\cdot C_x)^2 \\ (1\cdot\mu)^2 \\ (1\times10^{-3}\cdot C_s)^2 \\ (1\cdot q_s)^2 \\ (1\cdot C_p)^2 \\ (1\cdot q_p)^2 \end{bmatrix} \cdot I \qquad (6.1)$$

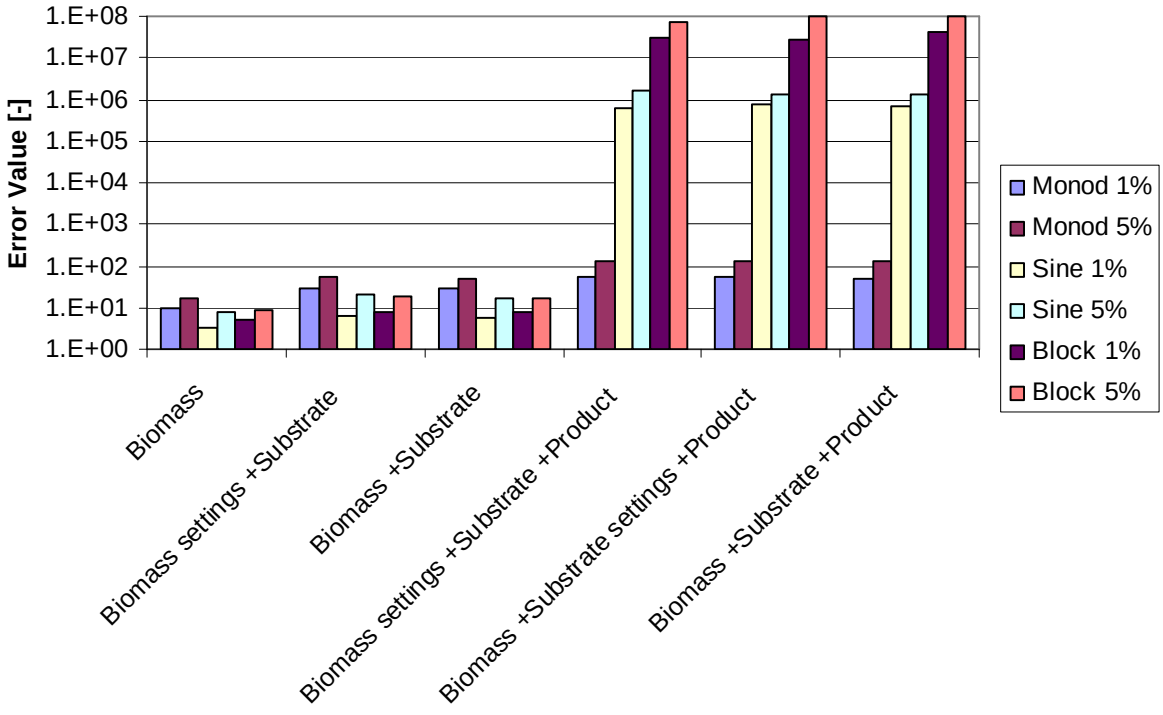


Figure 8:Results for optimization and evaluation for μ_{max}= 0.7 h^{-1}. Biomass means that there is tuned and evaluated for biomass only. Biomass setting +Substrate means that the setting from the biomass tuning are used for the system noise setting for biomass concentration and specific growth rate and the tuning algorithm only search for system noise settings for substrate concentration and specific consumption rate.

Noticeable in Figure 8 is the increase in equation error when more variables are estimated, this is due to the nature of the error calculation, it adds up the errors found on all the variables thus when

comparing for example biomass estimation and biomass + substrate estimation the error is approximately doubled. However, for the sine and block pattern the increase in the equation error is dramatic.

When comparing the performance if settings from an optimization with fewer variables are used, it shows that this does not make much of difference in the equation error, thus settings for the system noise are a property of a variable.

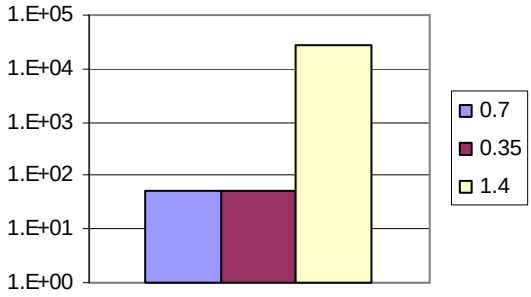


Figure 9: Results for estimation of a Monod dataset with 1 % noise and different values for μ_{max}

Figure 9 shows that if there is optimized for a certain value of μ_{max}, the settings can be applied to a lower value of μ_{max}, but not to a higher value of μ_{max}. In that case the system noise on μ is too low to accommodate the larger change necessary for a higher μ_{max}, recommended is tuning for the highest expected μ_{max}.

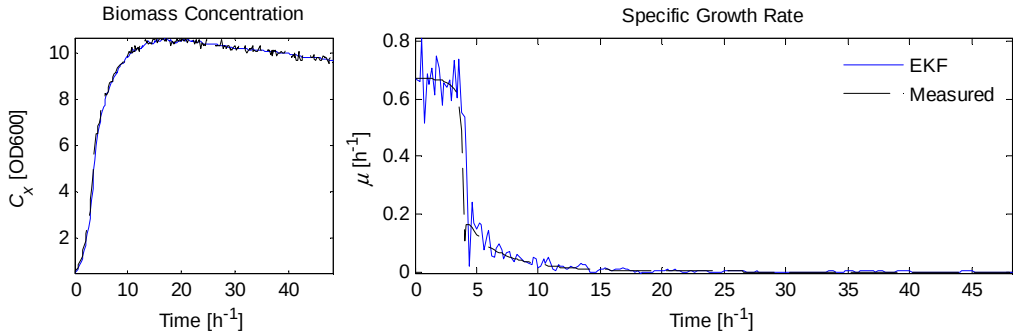


Figure 10: Results of the estimation by EKF on a Monod dataset with 1% noise

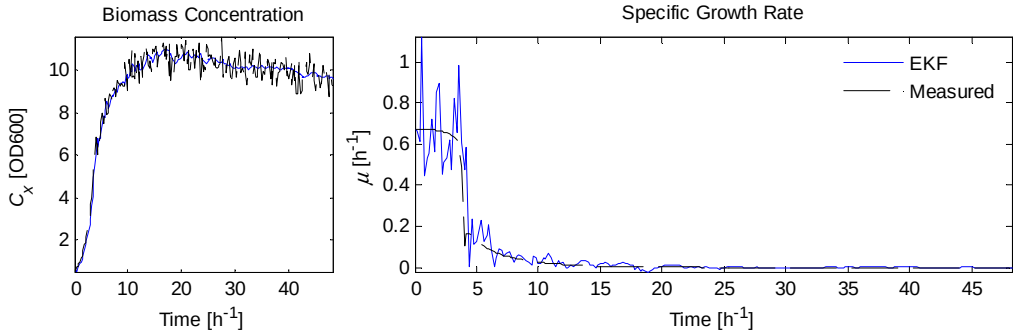


Figure 11: Results of the estimation by EKF on a Monod dataset with 5% noise

Comparison of Figure 10 and Figure 11 shows the influence of the noise on the estimation: The estimation of μ shows more noise in Figure 11. The biomass concentration plot shows the filtering

applied by the EKF clear. For the other estimations, the same differences are seen when 5% noise is applied.

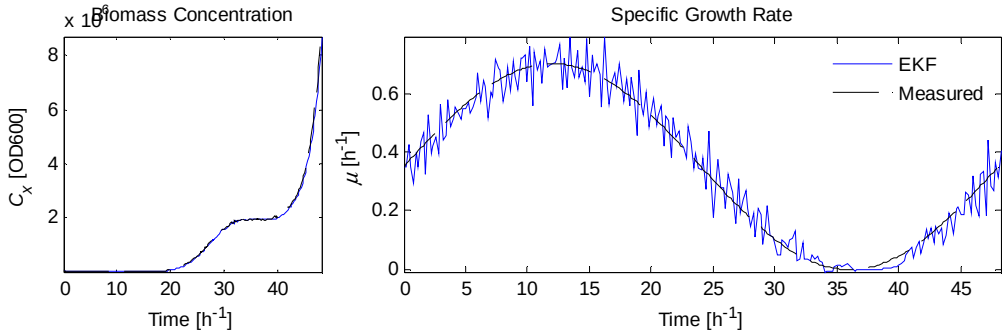


Figure 12: Results of the estimation by EKF on a sine dataset with 1% noise

In Figure 12 is shown that the EKF can follow a smooth change in time in the variables. Notice the extremely high biomass concentration, in reality this cannot occur.

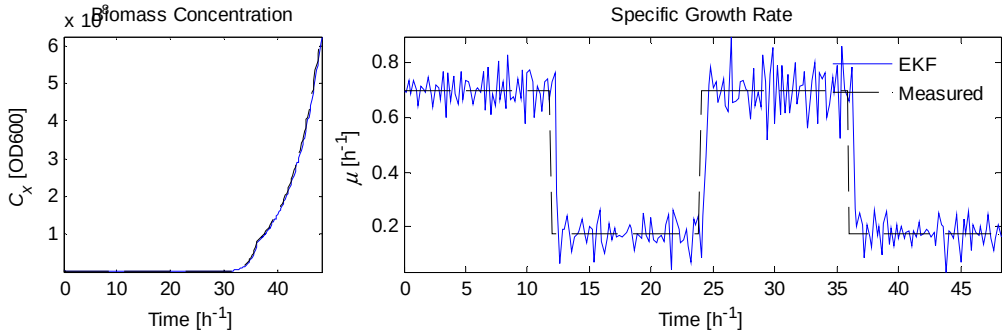


Figure 13: Results of the estimation by EKF on a block dataset with 1% noise

Figure 13 shows the EKF reconstructing abrupt changes in the variables accurately, notice the lag in the estimation at the moment of the abrupt change.

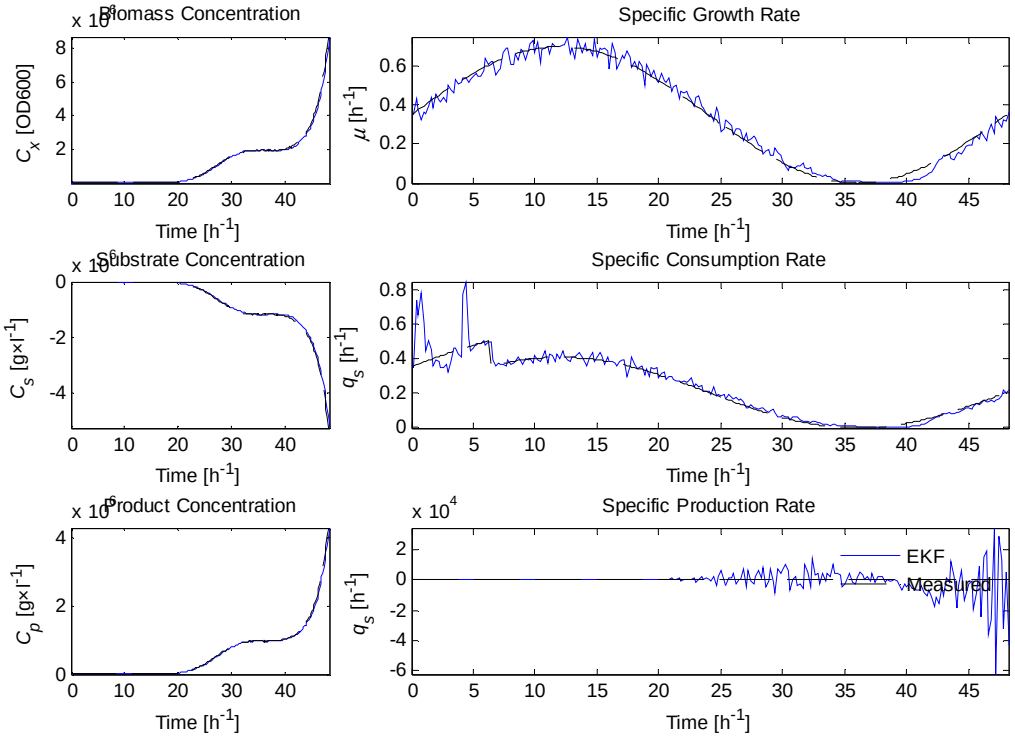


Figure 14: Results of the estimation by EKF on a sine dataset 1% noise

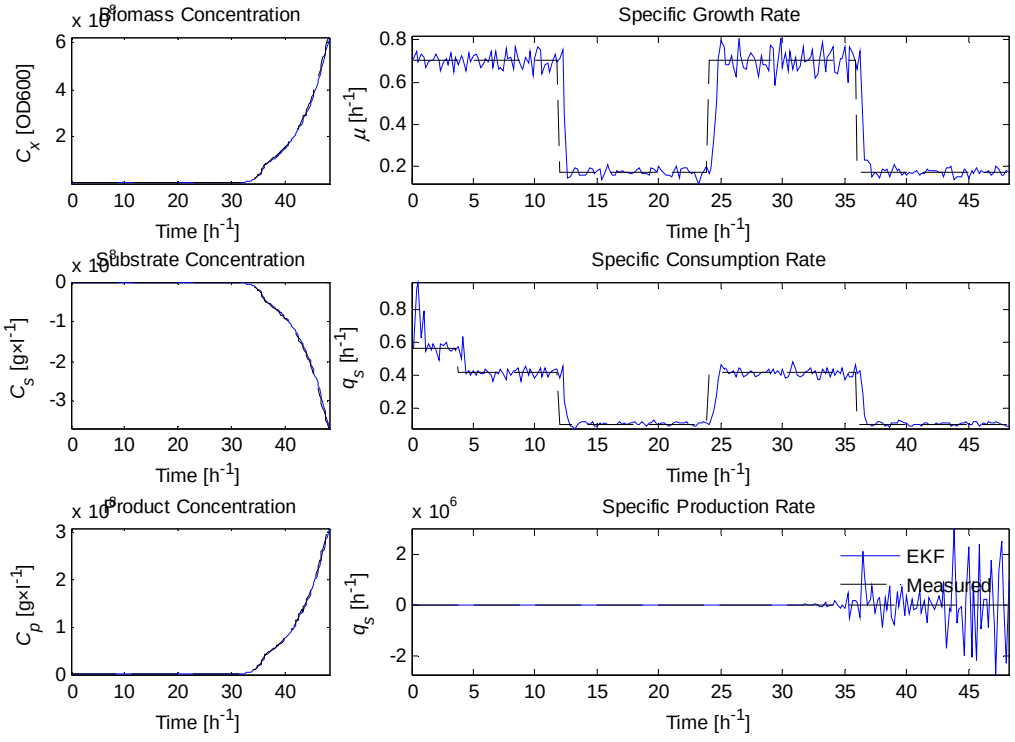


Figure 15: Results of the estimation by EKF on a block dataset 1% noise

Figure 14 and Figure 15 show the cause for the high equation error. The algorithm starts to go astray on the specific production rate, this is due to the change to a low value of the parameter and the high value for the accompanying variable that has relatively lots of noise.

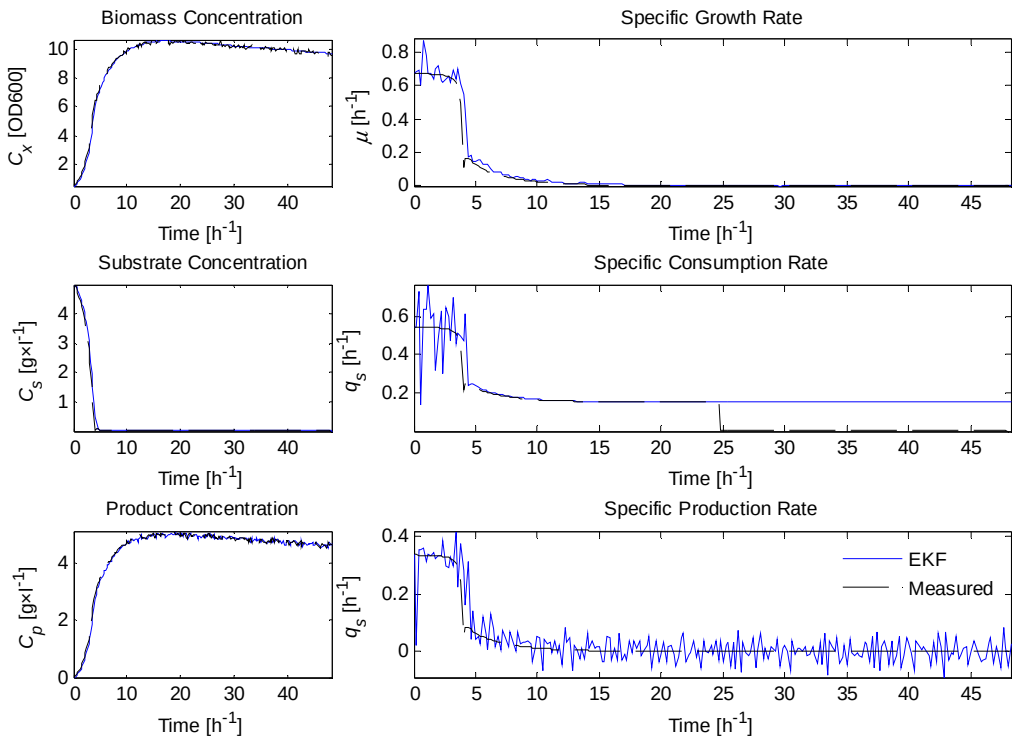


Figure 16: Results of the estimation by EKF on a Monod dataset with μ_{max}= 0.7 h^{-1} 1% noise

Figure 16 shows the EKF estimating the specific growth rate, specific consumption rate and the specific production rate. The specific production rate shows more noise than the specific growth rate and the specific consumption rate, this is because the uncertainty of the algorithm shows up in the specific production rate.

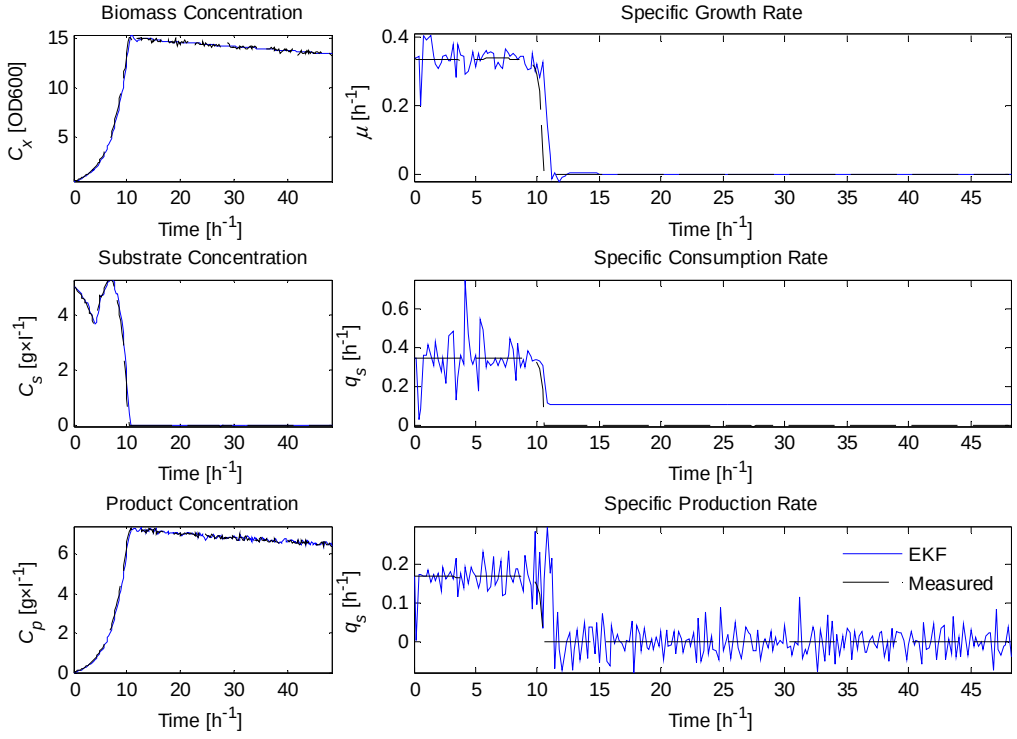


Figure 17: Results of the estimation by EKF on a Monod dataset with μ_{max}= 0.35 h^{-1} 1% noise

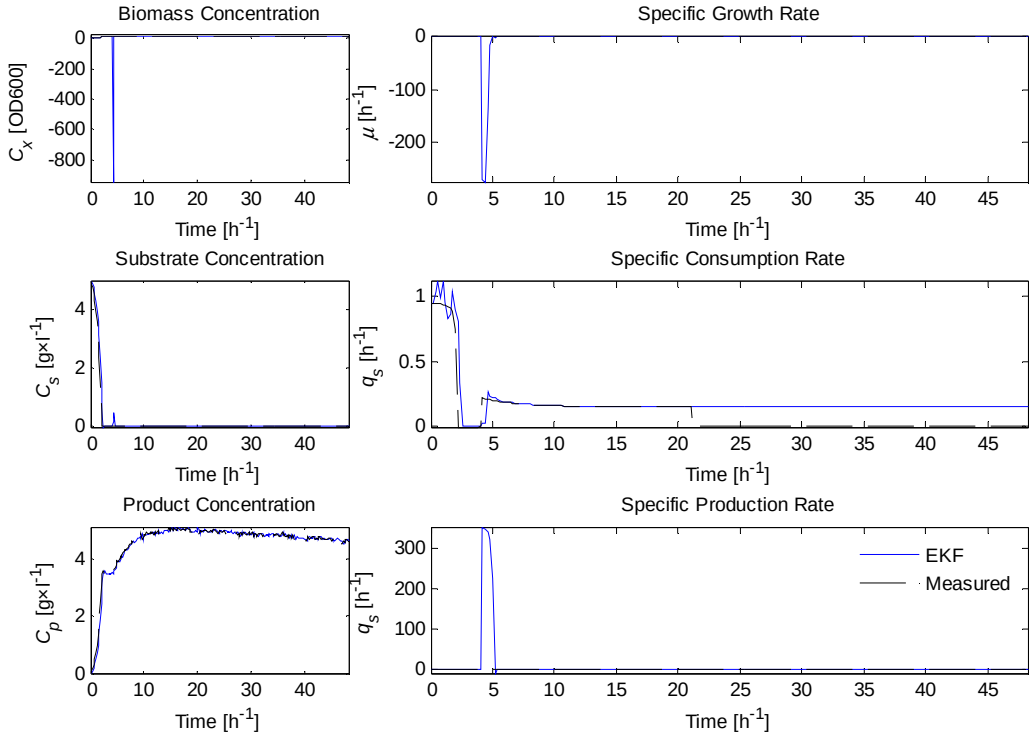


Figure 18: Results of the estimation by EKF on a Monod dataset with μ_{max}= 1.4 h^{-1} 1% noise

Figure 16, Figure 17 and Figure 18 use the same system noise settings. Figure 17 shows the EKF reconstructing the parameters almost equally well as in Figure 16, however Figure 18 shows that if a higher maximum specific growth rate is used on system noise settings for a lower value, the algorithm makes mistakes in reconstructing the biomass concentration, the specific growth rate and the specific production rate.

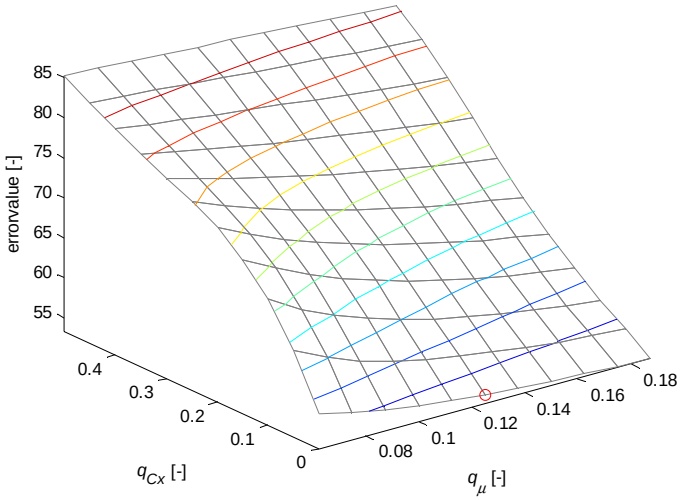


Figure 19: Contour-mesh plot of the equation error as a function of q_{Cx} and q_{μ}, the red dot indicates the value of the system noise settings.

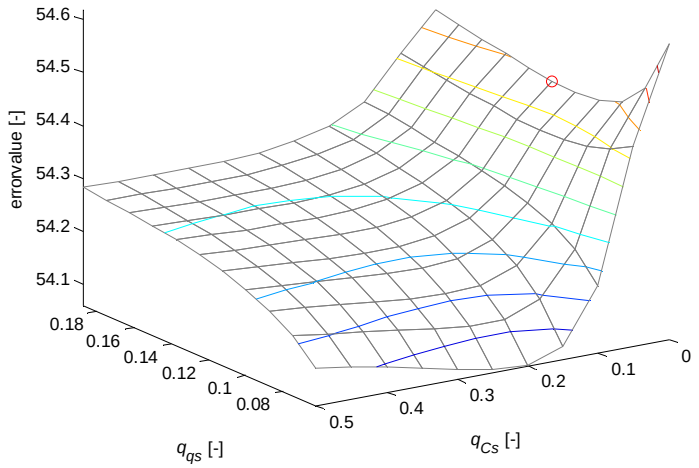


Figure 20: Contour-mesh plot of the equation error as a function of q_{Cs} and q_{qs}, the red dot indicates the value of the system noise settings.

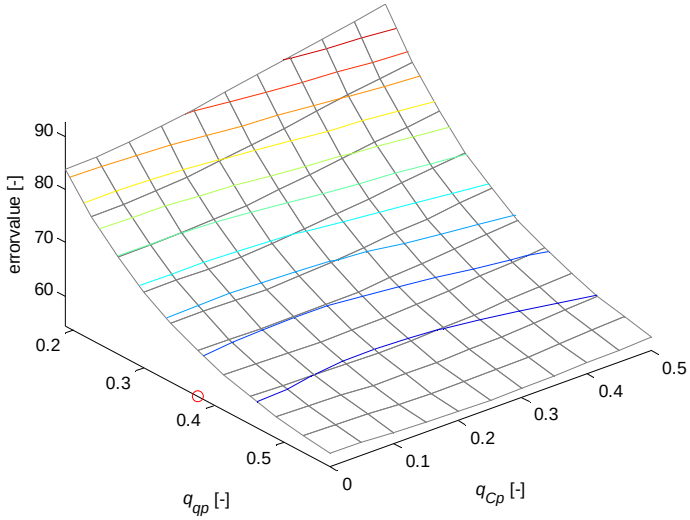


Figure 21: Contour-mesh plot of the equation error as a function of q_{Cp} and q_{qp}, the red dot indicates the value of the system noise settings.

Figures 19, 20 and 21 show the place of the optimized system noise in the error space, this indicates that indeed the optimal values are found however, Figure 20 shows that het value can have a lower equation error but the setting is also depending on the system noise setting for the other variables.

6.2. Extended Kalman Filter with high boundaries

Figure 22 shows the equation error (3.23) for the different optimization and evaluation. The optimization is executed on the Monod dataset with 1 % noise on the measurement the rest of the datasets are used for evaluation. The optimization upper limit is shown in 6.2.

$$Q_{up} = \begin{bmatrix} (10 \cdot C_x)^2 \\ (10 \cdot \mu)^2 \\ (10 \cdot C_s)^2 \\ (10 \cdot q_s)^2 \\ (10 \cdot C_p)^2 \\ (10 \cdot q_p)^2 \end{bmatrix} \cdot I \qquad (6.2)$$

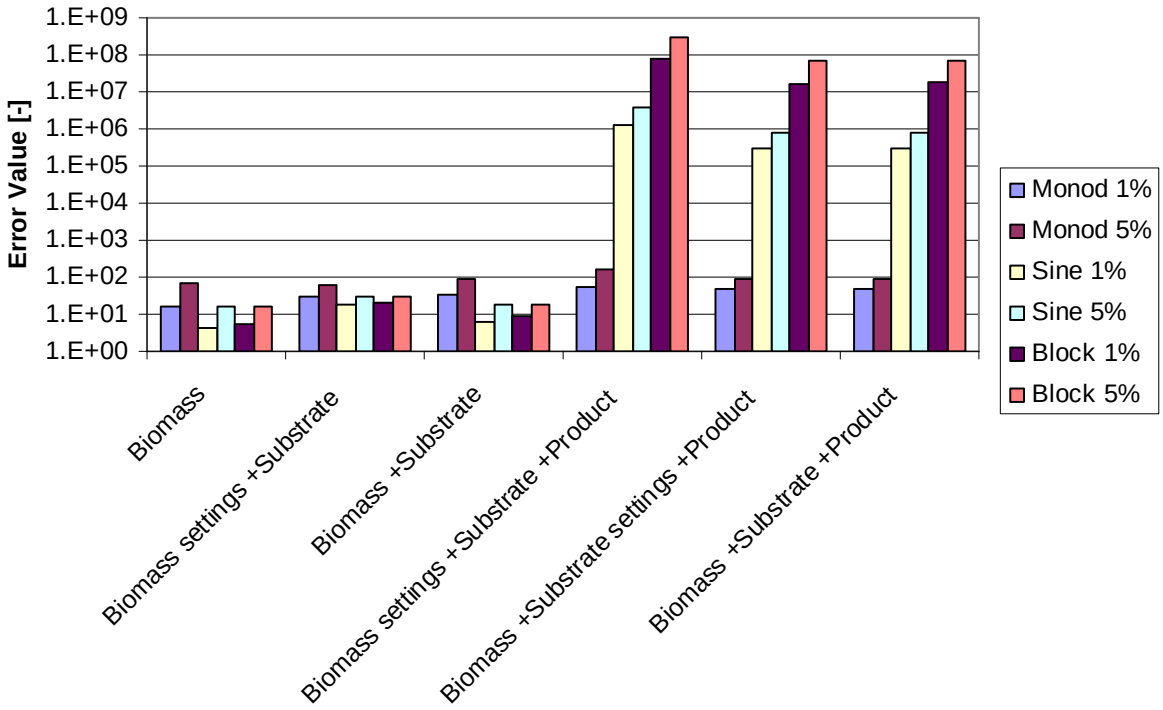


Figure 22: Results for optimization and evaluation

The EKF show for the high boundaries the same pattern in the equation error as the low boundaries. However, for the cases when biomass or biomass and substrate are reconstructed the error is higher. The estimations for all variables suffer the same fault as for the low boundary case, but for the Monod dataset, the performance is better than it is the case for the low boundaries.

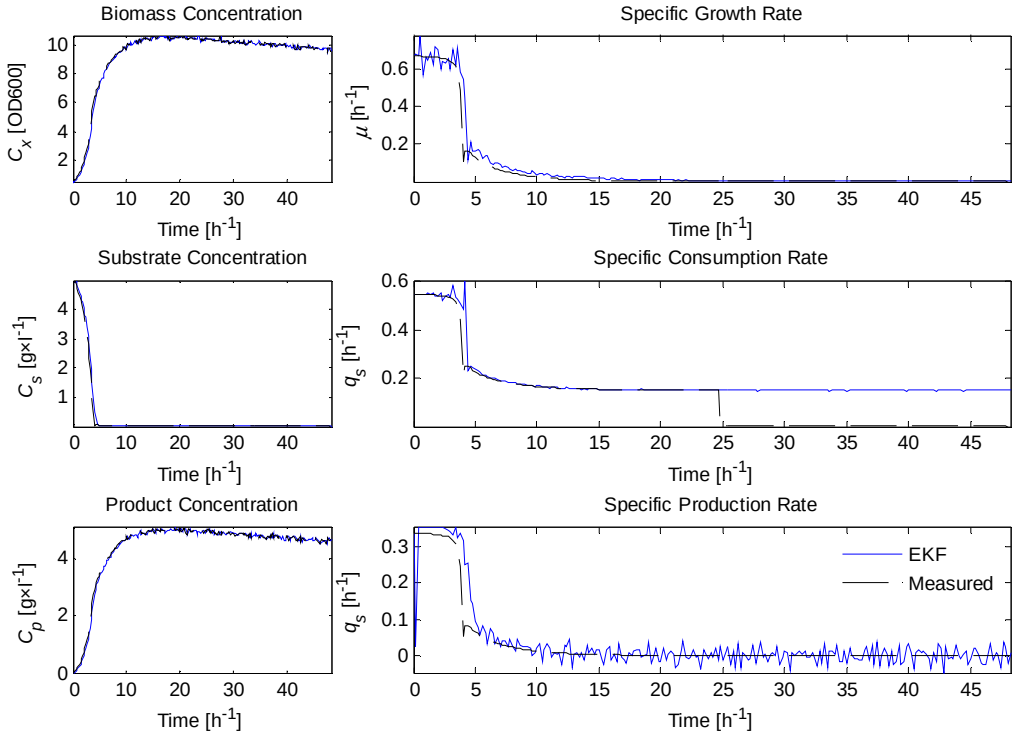


Figure 23: Results of the estimation by EKF on a Monod dataset with μ_{max}= 0.7 h^{-1} 1% noise

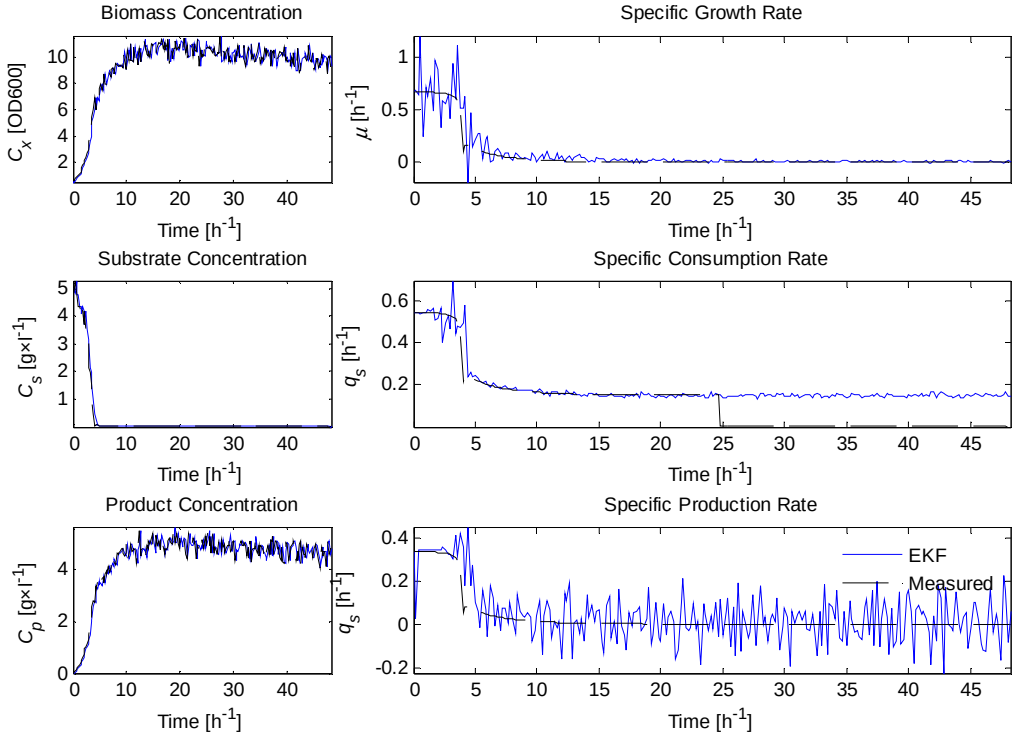


Figure 24: Results of the estimation by EKF on a Monod dataset with μ_{max}= 0.7 h^{-1} 5% noise

Note in Figure 23 and Figure 24 the filtering on the measured variables is less than the case with low boundaries but the noise on the specific production rate is lower.

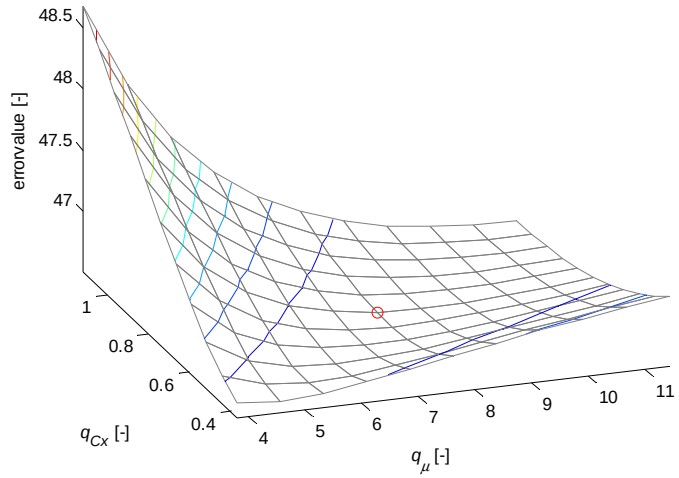


Figure 25: Contour-mesh plot of the equation error as a function of q_{Cx} and q_{μ}, the red dot indicates the value of the system noise settings.

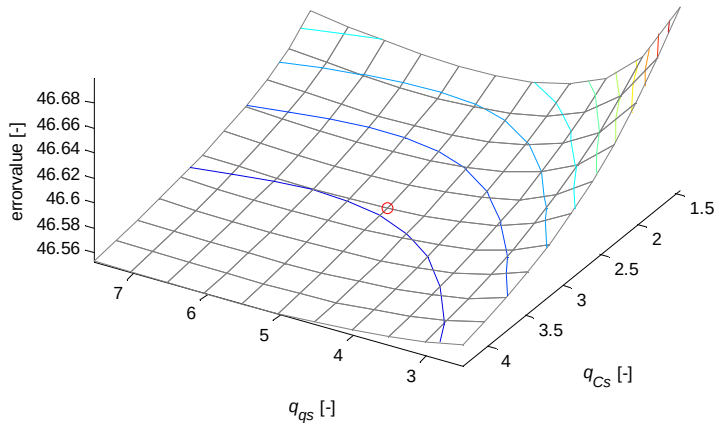


Figure 26: Contour-mesh plot of the equation error as a function of q_{Cs} and q_{qs}, the red dot indicates the value of the system noise settings.

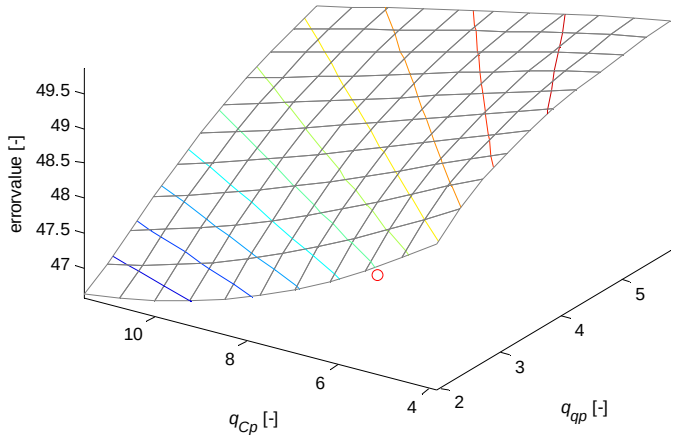


Figure 27: Contour-mesh plot of the equation error as a function of q_{Cp} and q_{qp}, the red dot indicates the value of the system noise settings.

In figures 25, 26 and 27 the place of the optimized value for the system noise in the error space is shown. Figure 25 shows that there is some correlation in the system noise for two related variables. In figures 26 and 27 it is noticeable that the value aren't optimal jet, this is because to total equation error is influenced by six settings for the system noise while in a plot only two of them can be shown while the other remain at a fixed value. The search algorithm also has problems with further optimizing because the error space is relatively flat in this region.

6.3. Recursive least squares.

The recursive least squares formulation used in this work is shown in equation 6.3. For this formulation was chosen because the original formulation did not function correctly in MATLAB. R is the equivalent of λ except is does not have the limit that λ has.

$$\begin{aligned}
&\varphi(k) = \begin{bmatrix} C_x(k-1) & C_x(k-1) & C_x(k-1) \end{bmatrix} \\
&K(k) = P(k-1)\cdot\varphi(k)\cdot(\varphi(k)\cdot P(k-1)\cdot\varphi(k)^T + R) \\
&P(k) = (I - K(k)\cdot\varphi(k))\cdot P(k-1)\cdot(I - \varphi(k)^T\cdot K(k)^T) + K(k)\cdot R\cdot K(k)^T \\
&\Theta(k) = \Theta(k-1) + K(k)\cdot(y(k) - \varphi(k)\cdot\Theta(k-1))
\end{aligned} \tag{6.3}$$

Because of the linearization, Θ contains more parameters that are shown in 6.4.

$$\Theta = \begin{bmatrix} \mu - D\cdot T & 0 & 0 \\ -q_s\cdot T & -D\cdot T & 0 \\ q_p\cdot T & 0 & -D\cdot T \end{bmatrix} \tag{6.4}$$

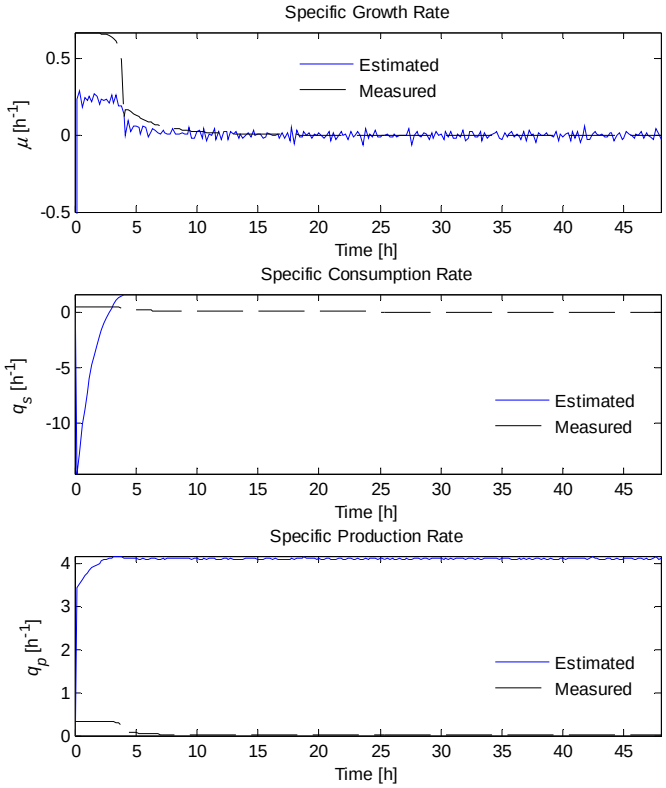


Figure 28: Results of the estimation by RLS on a Monod dataset with μ_{max}= 0.7 h^{-1} 1% noise

Figure 28 shows the estimation of all variables by the RLS is poor, this is due to the input (D) applied on the system. The RLS algorithm is not designed to handle more than one input and due to the linearization, the input can be excluded from the estimation. Both q_s and q_p are depending on two different measurements, C_x, C_s and C_x, C_p respectively. The algorithm cannot pinpoint which parameter should change to reflect the system at best. This leads to filling of the non-diagonal parts of the covariance matrix P, which indicates that the parameters are influencing each other.

To overcome it the parameters are estimated separately. Equations 6.5, 6.6 and 6.7 show the contents of φ and Θ.

$$\varphi(k) = C_x(k-1)$$
$$\Theta = 1 + \mu \cdot T \qquad (6.5)$$

$$\varphi(k) = C_x(k-1)$$
$$\Theta = -q_s \cdot T \qquad (6.6)$$

$$\varphi(k) = C_x(k-1)$$
$$\Theta = q_p \cdot T \qquad (6.7)$$

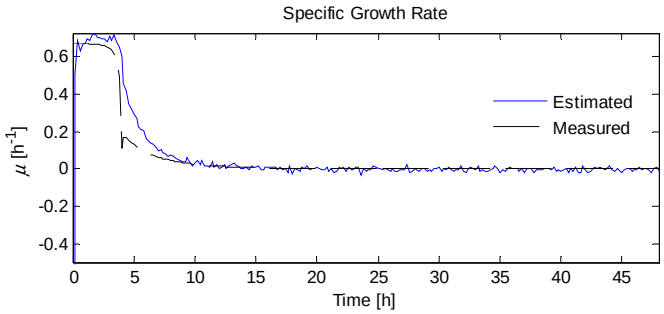


Figure 29: Results of the estimation by RLS on a Monod dataset with μ_{max}= 0.7 h^{-1} 1% noise

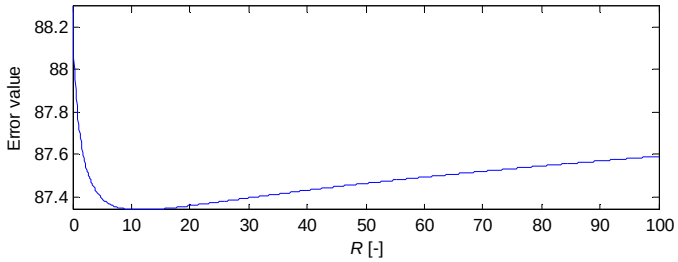


Figure 30: Error as a function of R for the estimation of μ only

If the specific growth rate is reconstructed as a single parameter (Figure 29), the covariance problem does not play a role anymore. The fact that the input does not disturb the estimation in this case because it acts on the same variable.

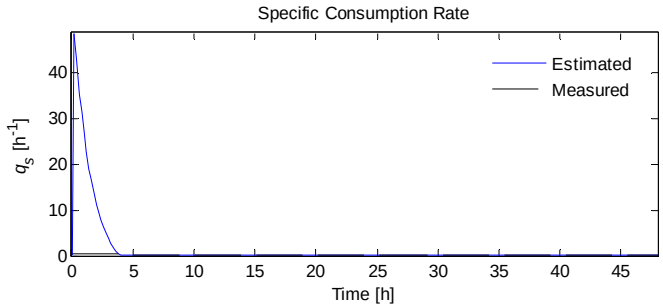


Figure 31: Results of the estimation by RLS on a Monod dataset with μ_{max}= 0.7 h^{-1} 1% noise

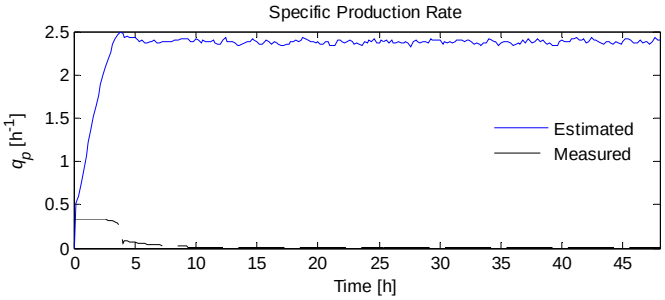


Figure 32: Results of the estimation by RLS on a Monod dataset with μ_{max}= 0.7 h^{-1} 1% noise

Figures 31 and 32 show the poor estimation of the other parameters, this is due to the property of the RLS, which accept only one variable per equation.

7. Graphical User Interface

For the EKF a GUI is build to make the tuning and usage more user-friendly. In Figure 33 the GUI is shown. In this figure, an example of an optimized estimation is shown.

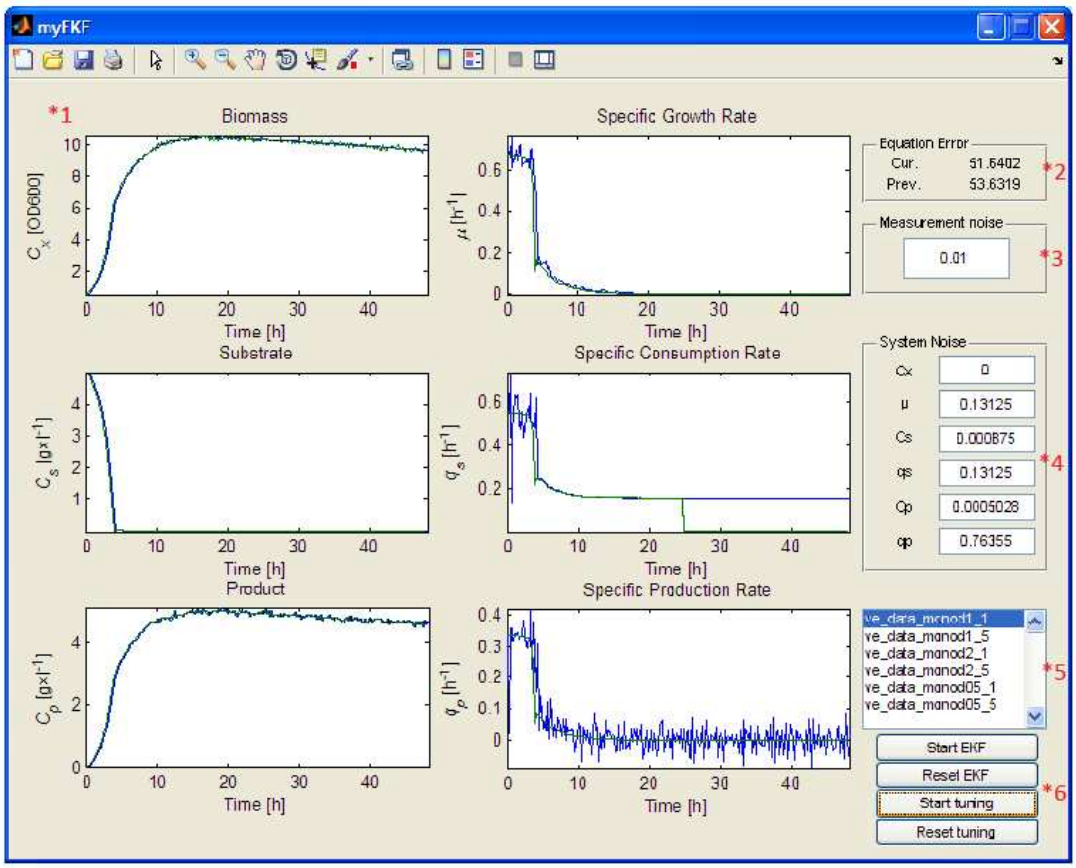


Figure 33: Screen dump of the EKF GUI

---

*1 This is the graph area, it shows all variables from the estimation. The biomass, substrate and product concentration are shown with next to them the specific rate.

*2 Shows the current and previous equation error, this shows if the tuning improved the previous settings.

*3 The measurement noise for all measurements can be set here, if the application will be used for a real estimation there should be separate settings for all measurements.

*4 The system noise can be set here, but it can also be filled in by the tuning method.

*5 Here a dataset can be selected for tuning, but it will be omitted for online use after a tuning has took place.

*6 These four buttons control the programs functions. The 'Start EKF'-button starts the estimation on the selected dataset with the settings set in *3 and *4.
The 'Reset EKF'-button clears the graphs.
The 'Start tuning'-button starts the tuning on the selected dataset.
The 'Reset tuning'-button resets the system noise to the default value.

8. Conclusion

8.1. Conclusion on the EKF algorithm

In general, the performance of EKF for the estimation of specific growth rate, substrate construction and product formation rates is good as long as there are not too many variables reconstructed at once. If a third parameter is reconstructed, it will show more disturbance than the other two variables. The other problem with three parameters is the inability to estimate a different pattern than the ideal case, which is a drawback to apply this to an unknown system. To overcome this problem, the EKF algorithm should be ran with only two parameters to ensure a good estimation of these parameters, to estimate more parameter a secondary EKF should be used in parallel with the first.

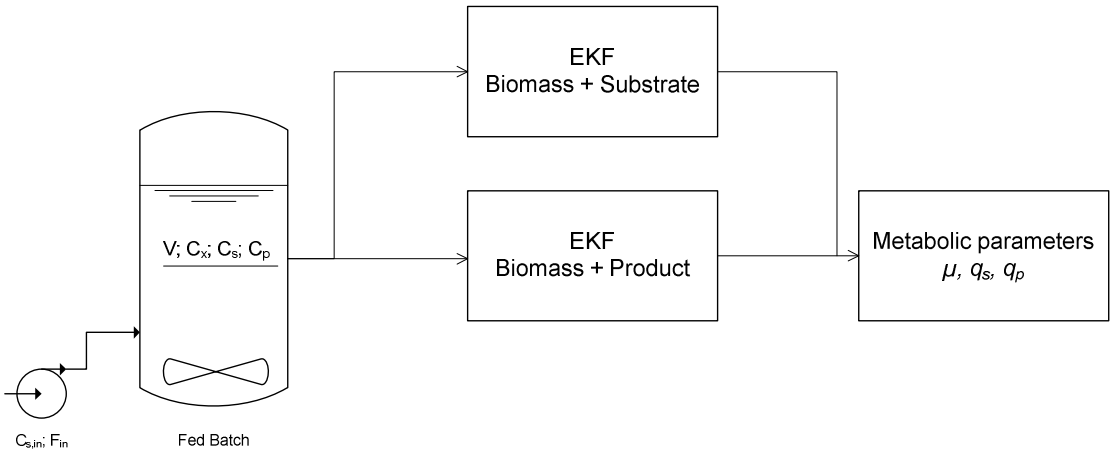


Figure 34: Schematic EKF

The biomass should be estimated in both the EKF's because both the substrate and the product equation depend on biomass and if it is included unfiltered the estimation accuracy of the parameters would be lower, because the noise is not filtered out of the data.

Comparing the two tuning strategies, it shows that the low boundaries case results in more filtering on the measurements than is the case for high boundaries. The tuning with low boundaries is preferred because it shows less equation error and with a visual comparison, it shows a better tracking of the parameters. The error plots as a function of the value of Q also support this.

8.2. Conclusion on the RLS algorithm

The RLS algorithm has shown poor performance, due to the input acting on a different variable then the parameter estimated, which leads to an inaccurate estimation.
If the parameters are estimated individually, the estimation is only possible for the specific growth rate. In the estimation of q_s and q_p, the biomass must be used as an input, which is not the only input, because here the other variable is the input, which also influences the estimated parameter.

8.3. Conclusion for the application

To construct an application for the monitoring of the external metabolic parameters, the EKF in the form of Figure 34 is recommended if the system noise is tuned with low boundaries for Q. In this work, a fixed time interval for the data availability is used. The algorithm should then be adapted to allow for a dynamic time interval, so if the measurements cannot be supplied at a set interval the application can cope with it. A tuning routine should be integrated to find the settings for the system noise if a simulated dataset is supplied.

9. List of symbols

Subscript	Description	
$_k$	Number of current recursive step	
$_K$	Total number of recursive steps	
$_{mod}$	Model outcome	
$_n$	Number of state variable	
$_N$	Total number of state variables	
Superscript	Description	
$\hat{}$	Estimated value	
$\bar{}$	Predicted value	
$\dot{}$	First derivative	
Symbol	Description	Unit
a	Verhulst inhibition constant	gram×litre^{-1}
A	Linear system matrix	-
B	Linear input matrix	-
C	Linear output matrix	-
C_p	Concentration of product	gram×litre^{-1}
c_p	Growth independent production constant	h^{-1}
C_s	Concentration of substrate	gram×litre^{-1}
$C_{s,in}$	Concentration of ingoing substrate	gram×litre^{-1}
C_x	Concentration of biomass	OD600
D	Dilution rate	h^{-1}
F_{in}	Ingoing flow rate	litre×h^{-1}
F_{out}	Outgoing flow rate	litre×h^{-1}
I	Identity matrix	-
K	Kalman gain (EKF) or total number of recursive steps (3.23)	-
k	Combined yield constants	-
K_C	Contois constant	gram×litre^{-1}
K_I	Inhibition constant	gram×litre^{-1}
K_M	Monod constant	gram×litre^{-1}
K_P	Aiba product constant	gram×litre^{-1}
L	RLS gain	-
m_s	Maintenance constant	h^{-1}
p	Parameter of parameter vector	-
P	Covariance matrix	-
Q	System noise matrix	-
q_p	Specific production rate of product	h^{-1}
q_s	Specific consumption rate of substrate	h^{-1}
R	Measurement noise matrix	-
r_p	Production rate	OD600×h^{-1}
t	Time	h
T	Time interval	h
u	Input variable or input variable vector	-
v_k	Measurement noise	-
w_k	System noise	-
x	State variable or state variable vector	-
y	Simulation results	-

Y_{ps}	Yield constant product on substrate	-
Y_{px}	Yield constant product on biomass	-
Y_{xs}	Yield constant biomass on substrate	-
Θ	Parameter vector	-
λ	Forgetting factor	-
φ	Measurement input vector	-
μ	Specific growth rate	h^{-1}
μ_{max}	Maximum specific growth rate	h^{-1}

10. References

Bastin, G. &Dochain, D. (1990). *On-line estimation and adaptive control of bioreactors.* Amsterdam [etc.]: Elsevier.

CELLutionBiotech (2010) CELL-tainer | Optimal Mass Transfer http://www.celltainer.com/highlights.html 15-06-2010

Dochain, D. &Bastin, G. (1984). Adaptive identification and control algorithms for nonlinear bacterial growth systems. *Automatica* 20(5): 621-634.

Gnoth, S., Jenzsch, M., Simutis, R. &Lübbert, A. (2007). Process Analytical Technology (PAT): Batch-to-batch reproducibility of fermentation processes by robust process operational design and control. *Journal of Biotechnology* 132(2): 180-186.

Lewis, F. L. (1986). *Optimal estimation : with an introduction to stochastic control theory.* New York [etc.]: Wiley.

Ljung, L. &Söderström, T. (1987). *Theory and practice of recursive identification.* Cambridge [etc.]: MIT Press.

Neeleman, R. (2002). *Biomass performance : monitoring and control in bio-pharmaceutical production.* [S.l.: s.n.].

Ogata, K. (2002). *Modern control engineering.* Upper Saddle River, NJ: Prentice Hall International.

Rienksma, R. A. (2008).Monitoring the metabolic activity of *Penicillium chrysogenum*. In *System and Control Group*Wageningen: Wageningen University.

Rinzema, A. (2009).Bioreactor Design: A Bird's Eye View. 111 Wageningen: Bioprocess Engineering Group, Wageningen University.

Appendices

Appendix A: Results Optimization and evaluation

Low boundaries

Growth pattern	Variables	µmax	R	Q	Error value
Monod	Biomass	0.7	$R_k = \left[(0.01 \cdot C_x)^2\right] \cdot I$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (0.5250 \cdot \mu)^2 \end{bmatrix} \cdot I$	9.9457
Monod	Biomass	0.7	$R_k = \left[(0.05 \cdot C_x)^2\right] \cdot I$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (0.5250 \cdot \mu)^2 \end{bmatrix} \cdot I$	17.003
Sine	Biomass	0.7	$R_k = \left[(0.01 \cdot C_x)^2\right] \cdot I$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (0.5250 \cdot \mu)^2 \end{bmatrix} \cdot I$	3.4499
Sine	Biomass	0.7	$R_k = \left[(0.05 \cdot C_x)^2\right] \cdot I$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (0.5250 \cdot \mu)^2 \end{bmatrix} \cdot I$	7.9525
Block	Biomass	0.7	$R_k = \left[(0.01 \cdot C_x)^2\right] \cdot I$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (0.5250 \cdot \mu)^2 \end{bmatrix} \cdot I$	5.0983
Block	Biomass	0.7	$R_k = \left[(0.05 \cdot C_x)^2\right] \cdot I$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (0.5250 \cdot \mu)^2 \end{bmatrix} \cdot I$	8.7407
Monod	Biomass Substrate	0.7	$R_k = \begin{bmatrix} (0.01 \cdot C_x)^2 \\ (0.01 \cdot C_s)^2 \end{bmatrix} \cdot I$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (0.5250 \cdot \mu)^2 \\ (0 \cdot C_s)^2 \\ (0.1313 \cdot q_s)^2 \end{bmatrix} \cdot I$	29.7992
Monod	Biomass Substrate	0.7	$R_k = \begin{bmatrix} (0.05 \cdot C_x)^2 \\ (0.05 \cdot C_s)^2 \end{bmatrix} \cdot I$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (0.5250 \cdot \mu)^2 \\ (0 \cdot C_s)^2 \\ (0.1313 \cdot q_s)^2 \end{bmatrix} \cdot I$	53.883
Sine	Biomass Substrate	0.7	$R_k = \begin{bmatrix} (0.01 \cdot C_x)^2 \\ (0.01 \cdot C_s)^2 \end{bmatrix} \cdot I$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (0.5250 \cdot \mu)^2 \\ (0 \cdot C_s)^2 \\ (0.1313 \cdot q_s)^2 \end{bmatrix} \cdot I$	6.2014
Sine	Biomass Substrate	0.7	$R_k = \begin{bmatrix} (0.05 \cdot C_x)^2 \\ (0.05 \cdot C_s)^2 \end{bmatrix} \cdot I$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (0.5250 \cdot \mu)^2 \\ (0 \cdot C_s)^2 \\ (0.1313 \cdot q_s)^2 \end{bmatrix} \cdot I$	21.4341

Block	Biomass Substrate	0.7	$R_k = \begin{bmatrix} (0.01 \cdot C_x)^2 \\ (0.01 \cdot C_s)^2 \end{bmatrix}$.	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (0.5250 \cdot \mu)^2 \\ (0 \cdot C_s)^2 \\ (0.1313 \cdot q_s)^2 \end{bmatrix} \cdot I$	7.9388
Block	Biomass Substrate	0.7	$R_k = \begin{bmatrix} (0.05 \cdot C_x)^2 \\ (0.05 \cdot C_s)^2 \end{bmatrix}$.	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (0.5250 \cdot \mu)^2 \\ (0 \cdot C_s)^2 \\ (0.1313 \cdot q_s)^2 \end{bmatrix} \cdot I$	18.6701
Monod	Biomass Substrate	0.7	$R_k = \begin{bmatrix} (0.01 \cdot C_x)^2 \\ (0.01 \cdot C_s)^2 \end{bmatrix}$.	$Q_k = \begin{bmatrix} (4\times10^{-6} \cdot C_x)^2 \\ (0.1250 \cdot \mu)^2 \\ (2.8\times10^{-5} \cdot C_s)^2 \\ (0.1250 \cdot q_s)^2 \end{bmatrix} \cdot I$	28.9379
Monod	Biomass Substrate	0.7	$R_k = \begin{bmatrix} (0.05 \cdot C_x)^2 \\ (0.05 \cdot C_s)^2 \end{bmatrix}$.	$Q_k = \begin{bmatrix} (4\times10^{-6} \cdot C_x)^2 \\ (0.1250 \cdot \mu)^2 \\ (2.8\times10^{-5} \cdot C_s)^2 \\ (0.1250 \cdot q_s)^2 \end{bmatrix} \cdot I$	48.2460
Sine	Biomass Substrate	0.7	$R_k = \begin{bmatrix} (0.01 \cdot C_x)^2 \\ (0.01 \cdot C_s)^2 \end{bmatrix}$.	$Q_k = \begin{bmatrix} (4\times10^{-6} \cdot C_x)^2 \\ (0.1250 \cdot \mu)^2 \\ (2.8\times10^{-5} \cdot C_s)^2 \\ (0.1250 \cdot q_s)^2 \end{bmatrix} \cdot I$	5.4234
Sine	Biomass Substrate	0.7	$R_k = \begin{bmatrix} (0.05 \cdot C_x)^2 \\ (0.05 \cdot C_s)^2 \end{bmatrix}$.	$Q_k = \begin{bmatrix} (4\times10^{-6} \cdot C_x)^2 \\ (0.1250 \cdot \mu)^2 \\ (2.8\times10^{-5} \cdot C_s)^2 \\ (0.1250 \cdot q_s)^2 \end{bmatrix} \cdot I$	17.3063
Block	Biomass Substrate	0.7	$R_k = \begin{bmatrix} (0.01 \cdot C_x)^2 \\ (0.01 \cdot C_s)^2 \end{bmatrix}$.	$Q_k = \begin{bmatrix} (4\times10^{-6} \cdot C_x)^2 \\ (0.1250 \cdot \mu)^2 \\ (2.8\times10^{-5} \cdot C_s)^2 \\ (0.1250 \cdot q_s)^2 \end{bmatrix} \cdot I$	7.5980
Block	Biomass Substrate	0.7	$R_k = \begin{bmatrix} (0.05 \cdot C_x)^2 \\ (0.05 \cdot C_s)^2 \end{bmatrix}$.	$Q_k = \begin{bmatrix} (4\times10^{-6} \cdot C_x)^2 \\ (0.1250 \cdot \mu)^2 \\ (2.8\times10^{-5} \cdot C_s)^2 \\ (0.1250 \cdot q_s)^2 \end{bmatrix} \cdot I$	15.9863

Monod	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.01 \cdot C_x)^2 \\ (0.01 \cdot C_s)^2 \\ (0.01 \cdot C_p)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (0.5250 \cdot \mu)^2 \\ (0.0005 \cdot C_s)^2 \\ (0.1250 \cdot q_s)^2 \\ (1.18\times10^{-6} \cdot C_p)^2 \\ (0.4390 \cdot q_p)^2 \end{bmatrix} \cdot I$	52.7272
Monod	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.05 \cdot C_x)^2 \\ (0.05 \cdot C_s)^2 \\ (0.05 \cdot C_p)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (0.5250 \cdot \mu)^2 \\ (0.0005 \cdot C_s)^2 \\ (0.1250 \cdot q_s)^2 \\ (1.18\times10^{-6} \cdot C_p)^2 \\ (0.4390 \cdot q_p)^2 \end{bmatrix} \cdot I$	135.5359
Sine	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.01 \cdot C_x)^2 \\ (0.01 \cdot C_s)^2 \\ (0.01 \cdot C_p)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (0.5250 \cdot \mu)^2 \\ (0.0005 \cdot C_s)^2 \\ (0.1250 \cdot q_s)^2 \\ (1.18\times10^{-6} \cdot C_p)^2 \\ (0.4390 \cdot q_p)^2 \end{bmatrix} \cdot I$	6.13×10^{5}
Sine	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.05 \cdot C_x)^2 \\ (0.05 \cdot C_s)^2 \\ (0.05 \cdot C_p)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (0.5250 \cdot \mu)^2 \\ (0.0005 \cdot C_s)^2 \\ (0.1250 \cdot q_s)^2 \\ (1.18\times10^{-6} \cdot C_p)^2 \\ (0.4390 \cdot q_p)^2 \end{bmatrix} \cdot I$	1.63×10^{6}
Block	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.01 \cdot C_x)^2 \\ (0.01 \cdot C_s)^2 \\ (0.01 \cdot C_p)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (0.5250 \cdot \mu)^2 \\ (0.0005 \cdot C_s)^2 \\ (0.1250 \cdot q_s)^2 \\ (1.18\times10^{-6} \cdot C_p)^2 \\ (0.4390 \cdot q_p)^2 \end{bmatrix} \cdot I$	3.07×10^{7}
Block	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.05 \cdot C_x)^2 \\ (0.05 \cdot C_s)^2 \\ (0.05 \cdot C_p)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (0.5250 \cdot \mu)^2 \\ (0.0005 \cdot C_s)^2 \\ (0.1250 \cdot q_s)^2 \\ (1.18\times10^{-6} \cdot C_p)^2 \\ (0.4390 \cdot q_p)^2 \end{bmatrix} \cdot I$	7.23×10^{7}

Monod	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.01 \cdot C_x)^2 \\ (0.01 \cdot C_s)^2 \\ (0.01 \cdot C_p)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (4\times10^{-6} \cdot C_x)^2 \\ (0.1250 \cdot \mu)^2 \\ (8\times10^{-5} \cdot C_s)^2 \\ (0.1250 \cdot q_s)^2 \\ (0.0001 \cdot C_p)^2 \\ (1 \cdot q_p)^2 \end{bmatrix} \cdot I$	55.5558
Monod	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.05 \cdot C_x)^2 \\ (0.05 \cdot C_s)^2 \\ (0.05 \cdot C_p)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (4\times10^{-6} \cdot C_x)^2 \\ (0.1250 \cdot \mu)^2 \\ (8\times10^{-5} \cdot C_s)^2 \\ (0.1250 \cdot q_s)^2 \\ (0.0001 \cdot C_p)^2 \\ (1 \cdot q_p)^2 \end{bmatrix} \cdot I$	136.0641
Sine	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.01 \cdot C_x)^2 \\ (0.01 \cdot C_s)^2 \\ (0.01 \cdot C_p)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (4\times10^{-6} \cdot C_x)^2 \\ (0.1250 \cdot \mu)^2 \\ (8\times10^{-5} \cdot C_s)^2 \\ (0.1250 \cdot q_s)^2 \\ (0.0001 \cdot C_p)^2 \\ (1 \cdot q_p)^2 \end{bmatrix} \cdot I$	7.90×10^{5}
Sine	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.05 \cdot C_x)^2 \\ (0.05 \cdot C_s)^2 \\ (0.05 \cdot C_p)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (4\times10^{-6} \cdot C_x)^2 \\ (0.1250 \cdot \mu)^2 \\ (8\times10^{-5} \cdot C_s)^2 \\ (0.1250 \cdot q_s)^2 \\ (0.0001 \cdot C_p)^2 \\ (1 \cdot q_p)^2 \end{bmatrix} \cdot I$	1.38×10^{6}
Block	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.01 \cdot C_x)^2 \\ (0.01 \cdot C_s)^2 \\ (0.01 \cdot C_p)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (4\times10^{-6} \cdot C_x)^2 \\ (0.1250 \cdot \mu)^2 \\ (8\times10^{-5} \cdot C_s)^2 \\ (0.1250 \cdot q_s)^2 \\ (0.0001 \cdot C_p)^2 \\ (1 \cdot q_p)^2 \end{bmatrix} \cdot I$	2.77×10^{7}
Block	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.05 \cdot C_x)^2 \\ (0.05 \cdot C_s)^2 \\ (0.05 \cdot C_p)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (4\times10^{-6} \cdot C_x)^2 \\ (0.1250 \cdot \mu)^2 \\ (8\times10^{-5} \cdot C_s)^2 \\ (0.1250 \cdot q_s)^2 \\ (0.0001 \cdot C_p)^2 \\ (1 \cdot q_p)^2 \end{bmatrix} \cdot I$	9.80×10^{7}

Monod	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.01 \cdot C_x)^2 \\ (0.01 \cdot C_s)^2 \\ (0.01 \cdot C_p)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (0.1250 \cdot \mu)^2 \\ (0 \cdot C_s)^2 \\ (0.1250 \cdot q_s)^2 \\ (0 \cdot C_p)^2 \\ (0.3769 \cdot q_p)^2 \end{bmatrix} \cdot I$	51.6315
Monod	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.05 \cdot C_x)^2 \\ (0.05 \cdot C_s)^2 \\ (0.05 \cdot C_p)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (0.1250 \cdot \mu)^2 \\ (0 \cdot C_s)^2 \\ (0.1250 \cdot q_s)^2 \\ (0 \cdot C_p)^2 \\ (0.3769 \cdot q_p)^2 \end{bmatrix} \cdot I$	135.7086
Sine	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.01 \cdot C_x)^2 \\ (0.01 \cdot C_s)^2 \\ (0.01 \cdot C_p)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (0.1250 \cdot \mu)^2 \\ (0 \cdot C_s)^2 \\ (0.1250 \cdot q_s)^2 \\ (0 \cdot C_p)^2 \\ (0.3769 \cdot q_p)^2 \end{bmatrix} \cdot I$	6.88×10^{5}
Sine	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.05 \cdot C_x)^2 \\ (0.05 \cdot C_s)^2 \\ (0.05 \cdot C_p)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (0.1250 \cdot \mu)^2 \\ (0 \cdot C_s)^2 \\ (0.1250 \cdot q_s)^2 \\ (0 \cdot C_p)^2 \\ (0.3769 \cdot q_p)^2 \end{bmatrix} \cdot I$	1.38×10^{6}
Block	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.01 \cdot C_x)^2 \\ (0.01 \cdot C_s)^2 \\ (0.01 \cdot C_p)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (0.1250 \cdot \mu)^2 \\ (0 \cdot C_s)^2 \\ (0.1250 \cdot q_s)^2 \\ (0 \cdot C_p)^2 \\ (0.3769 \cdot q_p)^2 \end{bmatrix} \cdot I$	4.05×10^{7}
Block	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.05 \cdot C_x)^2 \\ (0.05 \cdot C_s)^2 \\ (0.05 \cdot C_p)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (0.1250 \cdot \mu)^2 \\ (0 \cdot C_s)^2 \\ (0.1250 \cdot q_s)^2 \\ (0 \cdot C_p)^2 \\ (0.3769 \cdot q_p)^2 \end{bmatrix} \cdot I$	9.79×10^{7}

Monod	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.01 \cdot C_x)^2 \\ (0.01 \cdot C_s)^2 \\ (0.01 \cdot C_p)^2 \end{bmatrix} \cdot I$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (0.5354 \cdot \mu)^2 \\ (0 \cdot C_s)^2 \\ (0.5696 \cdot q_s)^2 \\ (0.0005 \cdot C_p)^2 \\ (0.8125 \cdot q_p)^2 \end{bmatrix} \cdot I$	52.8753
Monod	Biomass Substrate Product	1.4	$R_k = \begin{bmatrix} (0.01 \cdot C_x)^2 \\ (0.01 \cdot C_s)^2 \\ (0.01 \cdot C_p)^2 \end{bmatrix} \cdot I$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (0.5354 \cdot \mu)^2 \\ (0 \cdot C_s)^2 \\ (0.5696 \cdot q_s)^2 \\ (0.0005 \cdot C_p)^2 \\ (0.8125 \cdot q_p)^2 \end{bmatrix} \cdot I$	2.75×10^4
Monod	Biomass Substrate Product	0.35	$R_k = \begin{bmatrix} (0.01 \cdot C_x)^2 \\ (0.01 \cdot C_s)^2 \\ (0.01 \cdot C_p)^2 \end{bmatrix} \cdot I$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (0.5354 \cdot \mu)^2 \\ (0 \cdot C_s)^2 \\ (0.5696 \cdot q_s)^2 \\ (0.0005 \cdot C_p)^2 \\ (0.8125 \cdot q_p)^2 \end{bmatrix} \cdot I$	51.7667

High boundaries

Growth pattern	Variables	µmax	R	Q	Error value
Monod	Biomass	0.7	$R_k = \left[(0.01 \cdot C_x)^2\right] \cdot I$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (10 \cdot \mu)^2 \end{bmatrix} \cdot I$	16.045
Monod	Biomass	0.7	$R_k = \left[(0.05 \cdot C_x)^2\right] \cdot I$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (10 \cdot \mu)^2 \end{bmatrix} \cdot I$	68.8441
Sine	Biomass	0.7	$R_k = \left[(0.01 \cdot C_x)^2\right] \cdot I$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (10 \cdot \mu)^2 \end{bmatrix} \cdot I$	4.3332
Sine	Biomass	0.7	$R_k = \left[(0.05 \cdot C_x)^2\right] \cdot I$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (10 \cdot \mu)^2 \end{bmatrix} \cdot I$	17.1506
Block	Biomass	0.7	$R_k = \left[(0.01 \cdot C_x)^2\right] \cdot I$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (10 \cdot \mu)^2 \end{bmatrix} \cdot I$	5.4526
Block	Biomass	0.7	$R_k = \left[(0.05 \cdot C_x)^2\right] \cdot I$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (10 \cdot \mu)^2 \end{bmatrix} \cdot I$	15.8189

Monod	Biomass Substrate	0.7	$R_k = \begin{bmatrix} (0.01 \cdot C_x)^2 \\ (0.01 \cdot C_s)^2 \end{bmatrix}$.	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (10 \cdot \mu)^2 \\ (0.1763 \cdot C_s)^2 \\ (0.0027 \cdot q_s)^2 \end{bmatrix} \cdot I$	30.9774
Monod	Biomass Substrate	0.7	$R_k = \begin{bmatrix} (0.05 \cdot C_x)^2 \\ (0.05 \cdot C_s)^2 \end{bmatrix}$.	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (10 \cdot \mu)^2 \\ (0.1763 \cdot C_s)^2 \\ (0.0027 \cdot q_s)^2 \end{bmatrix} \cdot I$	58.1793
Sine	Biomass Substrate	0.7	$R_k = \begin{bmatrix} (0.01 \cdot C_x)^2 \\ (0.01 \cdot C_s)^2 \end{bmatrix}$.	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (10 \cdot \mu)^2 \\ (0.1763 \cdot C_s)^2 \\ (0.0027 \cdot q_s)^2 \end{bmatrix} \cdot I$	17.7480
Sine	Biomass Substrate	0.7	$R_k = \begin{bmatrix} (0.05 \cdot C_x)^2 \\ (0.05 \cdot C_s)^2 \end{bmatrix}$.	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (10 \cdot \mu)^2 \\ (0.1763 \cdot C_s)^2 \\ (0.0027 \cdot q_s)^2 \end{bmatrix} \cdot I$	29.698
Block	Biomass Substrate	0.7	$R_k = \begin{bmatrix} (0.01 \cdot C_x)^2 \\ (0.01 \cdot C_s)^2 \end{bmatrix}$.	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (10 \cdot \mu)^2 \\ (0.1763 \cdot C_s)^2 \\ (0.0027 \cdot q_s)^2 \end{bmatrix} \cdot I$	20.5155
Block	Biomass Substrate	0.7	$R_k = \begin{bmatrix} (0.05 \cdot C_x)^2 \\ (0.05 \cdot C_s)^2 \end{bmatrix}$.	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (10 \cdot \mu)^2 \\ (0.1763 \cdot C_s)^2 \\ (0.0027 \cdot q_s)^2 \end{bmatrix} \cdot I$	28.7030
Monod	Biomass Substrate	0.7	$R_k = \begin{bmatrix} (0.01 \cdot C_x)^2 \\ (0.01 \cdot C_s)^2 \end{bmatrix}$.	$Q_k = \begin{bmatrix} (0.6632 \cdot C_x)^2 \\ (7.6980 \cdot \mu)^2 \\ (2.6228 \cdot C_s)^2 \\ (3.7125 \cdot q_s)^2 \end{bmatrix} \cdot I$	35.1508
Monod	Biomass Substrate	0.7	$R_k = \begin{bmatrix} (0.05 \cdot C_x)^2 \\ (0.05 \cdot C_s)^2 \end{bmatrix}$.	$Q_k = \begin{bmatrix} (0.6632 \cdot C_x)^2 \\ (7.6980 \cdot \mu)^2 \\ (2.6228 \cdot C_s)^2 \\ (3.7125 \cdot q_s)^2 \end{bmatrix} \cdot I$	89.9057
Sine	Biomass Substrate	0.7	$R_k = \begin{bmatrix} (0.01 \cdot C_x)^2 \\ (0.01 \cdot C_s)^2 \end{bmatrix}$.	$Q_k = \begin{bmatrix} (0.6632 \cdot C_x)^2 \\ (7.6980 \cdot \mu)^2 \\ (2.6228 \cdot C_s)^2 \\ (3.7125 \cdot q_s)^2 \end{bmatrix} \cdot I$	6.2373

Sine	Biomass Substrate	0.7	$R_k = \begin{bmatrix} (0.05 \cdot C_x)^2 \\ (0.05 \cdot C_s)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (0.6632 \cdot C_x)^2 \\ (7.6980 \cdot \mu)^2 \\ (2.6228 \cdot C_s)^2 \\ (3.7125 \cdot q_s)^2 \end{bmatrix} \cdot I$	18.9325
Block	Biomass Substrate	0.7	$R_k = \begin{bmatrix} (0.01 \cdot C_x)^2 \\ (0.01 \cdot C_s)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (0.6632 \cdot C_x)^2 \\ (7.6980 \cdot \mu)^2 \\ (2.6228 \cdot C_s)^2 \\ (3.7125 \cdot q_s)^2 \end{bmatrix} \cdot I$	8.9893
Block	Biomass Substrate	0.7	$R_k = \begin{bmatrix} (0.05 \cdot C_x)^2 \\ (0.05 \cdot C_s)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (0.6632 \cdot C_x)^2 \\ (7.6980 \cdot \mu)^2 \\ (2.6228 \cdot C_s)^2 \\ (3.7125 \cdot q_s)^2 \end{bmatrix} \cdot I$	18.8053
Monod	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.01 \cdot C_x)^2 \\ (0.01 \cdot C_s)^2 \\ (0.01 \cdot C_p)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (10 \cdot \mu)^2 \\ (0.6270 \cdot C_s)^2 \\ (0.4317 \cdot q_s)^2 \\ (10 \cdot C_p)^2 \\ (4.0540 \cdot q_p)^2 \end{bmatrix} \cdot I$	55.6330
Monod	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.05 \cdot C_x)^2 \\ (0.05 \cdot C_s)^2 \\ (0.05 \cdot C_p)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (10 \cdot \mu)^2 \\ (0.6270 \cdot C_s)^2 \\ (0.4317 \cdot q_s)^2 \\ (10 \cdot C_p)^2 \\ (4.0540 \cdot q_p)^2 \end{bmatrix} \cdot I$	156.7604
Sine	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.01 \cdot C_x)^2 \\ (0.01 \cdot C_s)^2 \\ (0.01 \cdot C_p)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (10 \cdot \mu)^2 \\ (0.6270 \cdot C_s)^2 \\ (0.4317 \cdot q_s)^2 \\ (10 \cdot C_p)^2 \\ (4.0540 \cdot q_p)^2 \end{bmatrix} \cdot I$	1.26×10^6
Sine	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.05 \cdot C_x)^2 \\ (0.05 \cdot C_s)^2 \\ (0.05 \cdot C_p)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (10 \cdot \mu)^2 \\ (0.6270 \cdot C_s)^2 \\ (0.4317 \cdot q_s)^2 \\ (10 \cdot C_p)^2 \\ (4.0540 \cdot q_p)^2 \end{bmatrix} \cdot I$	3.92×10^6

Block	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.01 \cdot C_x)^2 \\ (0.01 \cdot C_s)^2 \\ (0.01 \cdot C_p)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (10 \cdot \mu)^2 \\ (0.6270 \cdot C_s)^2 \\ (0.4317 \cdot q_s)^2 \\ (10 \cdot C_p)^2 \\ (4.0540 \cdot q_p)^2 \end{bmatrix} \cdot I$	7.54×10^7
Block	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.05 \cdot C_x)^2 \\ (0.05 \cdot C_s)^2 \\ (0.05 \cdot C_p)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (0 \cdot C_x)^2 \\ (10 \cdot \mu)^2 \\ (0.6270 \cdot C_s)^2 \\ (0.4317 \cdot q_s)^2 \\ (10 \cdot C_p)^2 \\ (4.0540 \cdot q_p)^2 \end{bmatrix} \cdot I$	3.12×10^8
Monod	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.01 \cdot C_x)^2 \\ (0.01 \cdot C_s)^2 \\ (0.01 \cdot C_p)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (0.6632 \cdot C_x)^2 \\ (7.6980 \cdot \mu)^2 \\ (2.6228 \cdot C_s)^2 \\ (3.7125 \cdot q_s)^2 \\ (0.8934 \cdot C_p)^2 \\ (0.4375 \cdot q_p)^2 \end{bmatrix} \cdot I$	46.5560
Monod	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.05 \cdot C_x)^2 \\ (0.05 \cdot C_s)^2 \\ (0.05 \cdot C_p)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (0.6632 \cdot C_x)^2 \\ (7.6980 \cdot \mu)^2 \\ (2.6228 \cdot C_s)^2 \\ (3.7125 \cdot q_s)^2 \\ (0.8934 \cdot C_p)^2 \\ (0.4375 \cdot q_p)^2 \end{bmatrix} \cdot I$	89.3046
Sine	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.01 \cdot C_x)^2 \\ (0.01 \cdot C_s)^2 \\ (0.01 \cdot C_p)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (0.6632 \cdot C_x)^2 \\ (7.6980 \cdot \mu)^2 \\ (2.6228 \cdot C_s)^2 \\ (3.7125 \cdot q_s)^2 \\ (0.8934 \cdot C_p)^2 \\ (0.4375 \cdot q_p)^2 \end{bmatrix} \cdot I$	2.95×10^5
Sine	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.05 \cdot C_x)^2 \\ (0.05 \cdot C_s)^2 \\ (0.05 \cdot C_p)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (0.6632 \cdot C_x)^2 \\ (7.6980 \cdot \mu)^2 \\ (2.6228 \cdot C_s)^2 \\ (3.7125 \cdot q_s)^2 \\ (0.8934 \cdot C_p)^2 \\ (0.4375 \cdot q_p)^2 \end{bmatrix} \cdot I$	7.61×10^5

Block	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.01 \cdot C_x)^2 \\ (0.01 \cdot C_s)^2 \\ (0.01 \cdot C_p)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (0.6632 \cdot C_x)^2 \\ (7.6980 \cdot \mu)^2 \\ (2.6228 \cdot C_s)^2 \\ (3.7125 \cdot q_s)^2 \\ (0.8934 \cdot C_p)^2 \\ (0.4375 \cdot q_p)^2 \end{bmatrix} \cdot I$	1.70×10^{7}
Block	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.05 \cdot C_x)^2 \\ (0.05 \cdot C_s)^2 \\ (0.05 \cdot C_p)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (0.6632 \cdot C_x)^2 \\ (7.6980 \cdot \mu)^2 \\ (2.6228 \cdot C_s)^2 \\ (3.7125 \cdot q_s)^2 \\ (0.8934 \cdot C_p)^2 \\ (0.4375 \cdot q_p)^2 \end{bmatrix} \cdot I$	6.67×10^{7}
Monod	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.01 \cdot C_x)^2 \\ (0.01 \cdot C_s)^2 \\ (0.01 \cdot C_p)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (0.7426 \cdot C_x)^2 \\ (7.6319 \cdot \mu)^2 \\ (2.8836 \cdot C_s)^2 \\ (5.0351 \cdot q_s)^2 \\ (7.6942 \cdot C_p)^2 \\ (3.8633 \cdot q_p)^2 \end{bmatrix} \cdot I$	46.5634
Monod	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.05 \cdot C_x)^2 \\ (0.05 \cdot C_s)^2 \\ (0.05 \cdot C_p)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (0.7426 \cdot C_x)^2 \\ (7.6319 \cdot \mu)^2 \\ (2.8836 \cdot C_s)^2 \\ (5.0351 \cdot q_s)^2 \\ (7.6942 \cdot C_p)^2 \\ (3.8633 \cdot q_p)^2 \end{bmatrix} \cdot I$	91.6137
Sine	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.01 \cdot C_x)^2 \\ (0.01 \cdot C_s)^2 \\ (0.01 \cdot C_p)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (0.7426 \cdot C_x)^2 \\ (7.6319 \cdot \mu)^2 \\ (2.8836 \cdot C_s)^2 \\ (5.0351 \cdot q_s)^2 \\ (7.6942 \cdot C_p)^2 \\ (3.8633 \cdot q_p)^2 \end{bmatrix} \cdot I$	3.10×10^{5}
Sine	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.05 \cdot C_x)^2 \\ (0.05 \cdot C_s)^2 \\ (0.05 \cdot C_p)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (0.7426 \cdot C_x)^2 \\ (7.6319 \cdot \mu)^2 \\ (2.8836 \cdot C_s)^2 \\ (5.0351 \cdot q_s)^2 \\ (7.6942 \cdot C_p)^2 \\ (3.8633 \cdot q_p)^2 \end{bmatrix} \cdot I$	7.91×10^{5}

Block	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.01 \cdot C_x)^2 \\ (0.01 \cdot C_s)^2 \\ (0.01 \cdot C_p)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (0.7426 \cdot C_x)^2 \\ (7.6319 \cdot \mu)^2 \\ (2.8836 \cdot C_s)^2 \\ (5.0351 \cdot q_s)^2 \\ (7.6942 \cdot C_p)^2 \\ (3.8633 \cdot q_p)^2 \end{bmatrix} \cdot I$	1.74×10^7
Block	Biomass Substrate Product	0.7	$R_k = \begin{bmatrix} (0.05 \cdot C_x)^2 \\ (0.05 \cdot C_s)^2 \\ (0.05 \cdot C_p)^2 \end{bmatrix}$	$Q_k = \begin{bmatrix} (0.7426 \cdot C_x)^2 \\ (7.6319 \cdot \mu)^2 \\ (2.8836 \cdot C_s)^2 \\ (5.0351 \cdot q_s)^2 \\ (7.6942 \cdot C_p)^2 \\ (3.8633 \cdot q_p)^2 \end{bmatrix} \cdot I$	7.27×10^7

Appendix B: MATLAB scripts

Only the algorithm parts of the scripts will be shown here the rest of the files are available on CD.

Script EKF for biomass, substrate and product

```
% Error function with extendedKalmanfilter
% for use with fmincon
% Usage:
% [e] = errorkalman2(x,data,input,param,tijd)
% Where data = [Cx mu Cs qs]
% and input  = [D]
% param = [Csi]
% Programmer: Jan-Eise Vuist
% Last edit: 08 April 2010

function [e] = errorkalmanbiomassglucoseproduct(x,data,input,param,tijd,no)
qqp = x(6);
qcp = x(5);
qqs = x(4);
qcs = x(3);
qmu = x(2);
qcx = x(1);
rcx = 0.05;
Tb = tijd;
p = param;
Csi = param;
%Initial conditions of the states
Cx0 = data(1,1);   %OD600
mu0 = data(1,2);   %
Cs0 = data(1,3);   %g/L
qs0 = data(1,4);   %
Cp0 = data(1,5);   %g/L
qp0 = data(1,6);   %

x0 = [Cx0 mu0 Cs0 qs0 Cp0 qp0]';

% Creating P0
I=eye(length(x0));
P0=zeros(length(x0));
for i=1:(length(x0))
    P0(i,i)=10*x0(i,1);
end

Pupd=P0;
xupd=x0;

%Make the states equal to the initial conditions
Cx=Cx0;mu=mu0;Cs=Cs0;qs=qs0;Cp=Cp0;qp=qp0;
Xr(1,:)=[Cx,mu,Cs,qs,Cp,qp];
Tr(1,1)=[0];
dt=0.2;
time_interval=[0 dt];
[k]=length(tijd);
observ = zeros(length(Tb),1);
innov = zeros(length(Tb),3);
for i=1:k
        % Calculating A B C
        A = [mu-input(i) Cx 0 0 0 0;
```

```
     0 0 0 0 0 0;
     -qs 0 -input(i) -Cx 0 0;
     0 0 0 0 0 0;
     qp 0 0 0 -input(i) Cx;
     0 0 0 0 0 0];

B = [-Cx 0 Csi-Cs 0 -Cp 0]';

C = [1 0 0 0 0 0;
     0 0 1 0 0 0;
     0 0 0 0 1 0];

%R (measurement error matrix)
R = eye(1);
R(1,1) = (rcx*Cx)^2;
R(2,2) = (rcx*Cs)^2;
R(3,3) = (rcx*Cp)^2;

%Q matrix
Q = eye(2);
Q(1,1) = (qcx*Cx)^2;
Q(2,2) = (qmu*mu)^2;
Q(3,3) = (qcs*Cs)^2;
Q(4,4) = (qqs*qs)^2;
Q(5,5) = (qcp*Cp)^2;
Q(6,6) = (qqp*qp)^2;

%observability criterion
OB=[C;C*A];
m=rank(OB);
n=length(A);
if m==n ;  %system is observable
    v=1;
else       %system is not observable
    v=0;
end
observ(i,:)=v;

%prediction
u=input(i,1);
[T,X]=ode45(@filtermodeldbgp,time_interval,xupd,[],u,p);
xpred=X(end,:)';
Ak=expm(A*dt);
Ppred=Ak*Pupd*Ak'+Q;

%update
Kupd=Ppred*C'*(inv(C*Ppred*C'+R));
Pupd=(I-Kupd*C)*Ppred;
inno=[data(i,1) data(i,3) data(i,5)]'-C*xpred;
xupd=xpred+Kupd*inno;

Cx=xupd(1);mu=xupd(2);Cs=xupd(3);qs=xupd(4);Cp=xupd(5);qp=xupd(5);

%put the results in a vector
Xr(i+1,:)=xupd';
Ur(i+1,:)=u';
Tr(i+1,1)=i*dt;
innov(i+1,:)=inno';
end
```

```
errcx = sqrt(sum((Xr(1:no,1)-
data(:,1)).^2))/(sum(data(:,1))/numel(data(:,1)));
errmu = sqrt(sum((Xr(1:no,2)-
data(:,2)).^2))/(sum(data(:,2))/numel(data(:,2)));
errcs = sqrt(sum((Xr(1:no,3)-
data(:,3)).^2))/(sum(data(:,3))/numel(data(:,3)));
errqs = sqrt(sum((Xr(1:no,4)-
data(:,4)).^2))/(sum(data(:,4))/numel(data(:,4)));
errcp = sqrt(sum((Xr(1:no,5)-
data(:,5)).^2))/(sum(data(:,5))/numel(data(:,5)));
errqp = sqrt(sum((Xr(1:no,6)-
data(:,6)).^2))/(sum(data(:,6))/numel(data(:,6)));
e = errcx + errmu + errcs + errqs + errcp + errqp;
```

Script RLS for biomass, substrate and product

```
function [e,Pv,K,theta,Pt] = rls4(x,y,tijd,mug,qs,qp,D,Csin,Q)
% Initial values
R = x;
I= eye(3);
P = 1e3*I;
Pt=P;
theta= [1 1 1;
        0 0 0;
        0 0 0];

%RLS
for k=2:length(tijd)
    phi = [y(k-1,1) y(k-1,1 ) y(k-1,1)];
    Pt       = P+(Q*I);
    K        = Pt * phi'*inv(phi*Pt*phi'+R);
    P        = (I-K*phi)*Pt*(I-phi'*K')+K*R*K'
    Pv(:,k)  = diag(P);
    theta = theta + K * ([y(k,1) y(k,2) y(k,3)] - phi*theta);
    p1(k)  = theta(1,1);
    p2(k) = theta(2,1);

    p4(k)  = theta(1,2);
    p5(k) = theta(2,2);

    p7(k)  = theta(1,3);
    p8(k) = theta(2,3);

end

e = sqrt(sum((((p1-1)/0.2)-mug').^2))/(sum(mug)/numel(mug)) + sqrt(sum(((-
p5-0.2)-qs').^2))/(sum(qs)/numel(qs)) +sqrt(sum(((p7/0.2)-
qp').^2))/(sum(qp)/numel(qp));
```