



Nonlinear filtering for a Chemostat model

Oussama Hadj Abdelkader, Mohamed Amine Hadj Abdelkader

► **To cite this version:**

Oussama Hadj Abdelkader, Mohamed Amine Hadj Abdelkader. Nonlinear filtering for a Chemostat model. 2016. <hal-01316362>

HAL Id: hal-01316362

<https://hal.inria.fr/hal-01316362>

Submitted on 16 May 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

Nonlinear filtering for a Chemostat model

Oussama Hadj Abdelkader Mohamed Amine Hadj Abdelkader

Monday 16th May, 2016

Contents

1	A stochastic chemostat model	1
2	Nonlinear filtering	2
2.1	Extended Kalman Filter	3
2.2	Unscented Kalman Filter	5
2.3	Particle Filter “Bootstrap”	8
3	Simulation results	11
	References	14

1 A stochastic chemostat model

The stochastic model of the chemostat given by equation (1) was first introduced by ?

$$\begin{cases} dX_t^1 = (\mu(X_t^2) - D)X_t^1 dt + c_1\sqrt{X_t^1} d\omega_t^1 \\ dX_t^2 = D(s_{in} - X_t^2) dt - k_{cs}\mu(X_t^2)X_t^1 dt + c_2\sqrt{X_t^2} d\omega_t^2 \end{cases} \quad (1)$$

where $\mu(X_t^2) = \mu_{max} \frac{X_t^2}{K_s + X_t^2}$, ω_t^1 and ω_t^2 are two mutually independent Brownian motions of covariance matrix Q^w .

For the model output, we suppose that only the substrate concentration $X_{t_k}^2$ is measured at discrete time instants $t_k = k\Delta$ where Δ is the observation step. This output is subject to a measurement noise v_k of standard deviation σ which is supposed proportional to $X_{t_k}^2$?. This leads to the output equation (2).

$$y_k = X_{t_k}^2 + \sigma X_{t_k}^2 v_k \quad (2)$$

where $v_k \stackrel{iid}{\sim} N(0, Q^v)$ ¹.

¹iid : independant and identically distributed.

This system could be simulated using an explicite Euler-Maruyama scheme with the simulation step δ where $\delta < \Delta$. Algorithm (1) presents the state space model (1) simulation.

```

% initialization
 $\delta = T / (N_{syst} * N_{obs})$ 
 $v_0, \dots, v_{N_{obs}} \sim N(0, 1)$ 
 $w_0^1, \dots, w_{N_{syst} * N_{obs}}^1 \sim N(0, 1)$ 
 $w_0^2, \dots, w_{N_{syst} * N_{obs}}^2 \sim N(0, 1)$ 
 $(X_0^1, X_0^2) \sim p(x_0)$ 
% iterations
For  $k = 0, \dots, N_{obs}$  do
  For  $j = 1, \dots, N_{syst}$  do
     $\mu = \mu_{max} \frac{X_{j-1}^2}{k_s + X_{j-1}^2}$ 
     $X_j^1 = \max(0, X_{j-1}^1 + (\mu - D)X_{j-1}^1\delta + c_1\sqrt{X_{j-1}^1}\sqrt{\delta}w_j^1)$ 
     $X_j^2 = \max(0, X_{j-1}^2 - k_{cs}\mu X_{j-1}^1\delta + D(s_{in} - X_{j-1}^2)\delta + c_2\sqrt{X_{j-1}^2}\sqrt{\delta}w_j^2)$ 
  End For
   $X_k^2 = X_j^2$ 
   $y_k = X_k^2 + \sigma X_k^2 v_k$ 
% State initialization of the next  $k$  loop
   $\mu = \mu_{max} \frac{X_j^2}{k_s + X_j^2}$ 
   $X_0^1 = \max(0, X_j^1 + (\mu - D)X_j^1\delta + c_1\sqrt{X_j^1}\sqrt{\delta}w_j^1)$ 
   $X_0^2 = \max(0, X_j^2 - k_{cs}\mu X_j^1\delta + D(s_{in} - X_j^2)\delta + c_2\sqrt{X_j^2}\sqrt{\delta}w_j^2)$ 
End For

```

Algorithm 1: Simulation of system (1) using the Euler-Maruyama scheme.

2 Nonlinear filtering

Nonlinear filtering is the task of estimating recursively in time the sequence of marginal posteriors $\left\{ p_{X_{t_k} | Y_{0:k}}(x_{t_k} | y_{0:k}) \right\}_{k \geq 0}$ called the filtering densities, that is, the estimation of the current state X_{t_k} given the measurements y_0, \dots, y_k . There are many methods to explicitly perform filtering task for nonlinear systems. In this paper, we will present three of them and compare their results. In this section, we suppose that all the model parameters are known.

Previous research work was carried out on the state estimation for the deterministic state-space model of the chemostat. For example, ? used an invariant observer for this purpose. These methods, as well as the used deterministic models, do not take account of the noises the system and the observations are subject to. Hence, these results could not be considered trustworthy.

the state-space model (1) along with the output equation (2) are more descriptive of the stochastic nature of the system. The state estimation for this stochastic model could not be performed using classical observers but it requires the use of nonlinear filtering techniques. Among these methods, we could cite the Extended Kalman Filter EKF [Citation], the Unscented Kalman Filter UKF ? and the Particle Filtering PF techniques [Citation]. The EKF and the UKF are used for state estimation of nonlinear models where the state and measurement noises are Gaussians, whereas the PF techniques could be applied for all types of noises.

Many authors in the litterature used the nonlinear filtering methods for the state estimation of stochastic biological models. ? used an EKF for state and parameter estimation on a non-linear stochastic dynamic chemostat model in an aerobic yeast continuous cultivation process. The filter algorithm uses accessible inputs such as the input substrate concentration and the dilution rate to estimate the biomass concentration, the maximum specific growth rate, the saturation constant and the substrate yield coefficient. This model however only features additive Gaussian noise on the state and output equations (.../...).

In the next sections, we present the application of these methods for the state estimation of model (1) and we compare their results.

2.1 Extended Kalman Filter

The application of the EKF to the stochastic model of the chemostat is given by Algorithm (2). The EKF gives estimation of both state mean and covariance matrix in two steps, the prediction step and the correction step. In the prediction step we linearize system (1) around the state mean estimation at the previous instant, then use the Euler scheme to compute the predicted mean and variance of the current state. Next, in the correction step we use the output equation (2), its linearization and the measurements y_0, \dots, y_k to compute the Kalman gain and update the mean and variance estimation.

This algorithm could not be implemented directly because of the nonlinearity between the state variable and the noise in the observation equation (2). This problem could be resolved by replacing the observation y_k by $\tilde{y}_k = \log(y_k)$ in the EKF algorithm. Indeed, measuring y_k or $\log(y_k)$ is not of an issue on the estimation quality since the same information is acquired in both cases. The new measure \tilde{y}_k is given by :

$$\begin{aligned} y_k &= X_{t_k}^2 + \sigma X_{t_k}^2 v_k \\ y_k &= X_{t_k}^2 (1 + \sigma v_k) \\ \tilde{y}_k &= \log(y_k) \\ \tilde{y}_k &= \log(X_{t_k}^2 (1 + \sigma v_k)) \\ &= \log(X_{t_k}^2) + \log(1 + \sigma v_k) \end{aligned}$$

Using a first order Taylor series development of $\log(1 + \sigma v_k)$ at point 1, we obtain:

$$\tilde{y}_k = \log(X_{t_k}^2) + \log(1) + \sigma v_k \log'(1)$$

$$\tilde{y}_k = \log(X_{t_k}^2) + \sigma v_k \quad (3)$$

Then the function h in the EKF algorithm becomes :

$$h(X_{t_k}) = \log(X_{t_k}^2)$$

Let F_t and H_t be the derivatives of $f(X_t)$ and $h(X_t)$ with respect to X_t respectively given by :

$$F_t = \nabla f(X_t) = \begin{bmatrix} \mu(X_t^2) - D & \mu'(X_t^2) X_t^1 \\ -k_{cs}\mu(X_t^2) & -D - k_{cs}\mu'(X_t^2) X_t^1 \end{bmatrix}$$

$$H_t = \nabla h(X_t) = \begin{bmatrix} 0 & \frac{1}{X_t^2} \end{bmatrix}$$

```

% initialization
 $\delta = T / (N_{syst} * N_{obs})$ 
 $X_0 \sim N(\mu_0, Q_0)$ 
 $\hat{X}_0 \leftarrow \mu_0$ 
 $R_0 = Q_0$ 
 $Q^w = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$  % state noise variance matrix
 $Q^v = 1$  % output noise variance
% iterations
For  $k = 0, \dots, N_{obs}$  do
% prediction step
  For  $j = 1, \dots, N_{syst}$  do
     $\hat{X}_{j-}^1 \leftarrow \max(0, \hat{X}_{j-1}^1 + f_1(\hat{X}_{j-1}^1, \hat{X}_{j-1}^2) \delta)$ 
     $\hat{X}_{j-}^2 \leftarrow \max(0, \hat{X}_{j-1}^2 + f_2(\hat{X}_{j-1}^1, \hat{X}_{j-1}^2) \delta)$ 
     $R_{j-} \leftarrow R_{j-1} + \left( R_{j-1} F^* + F R_{j-1} + g(\hat{X}_{j-1}^1, \hat{X}_{j-1}^2) Q^w g(\hat{X}_{j-1}^1, \hat{X}_{j-1}^2)^* \right) \delta$ 
     $\hat{X}_j^1 \leftarrow \hat{X}_{j-}^1$ 
     $\hat{X}_j^2 \leftarrow \hat{X}_{j-}^2$ 
     $R_j \leftarrow R_{j-}$ 
  End For
% correction step
   $\hat{X}_{k-}^1 = \hat{X}_{j-}^1$ 
   $\hat{X}_{k-}^2 = \hat{X}_{j-}^2$ 
   $R_{k-} = R_{j-}$ 
   $K_k = R_{k-} H^* (H R_{k-} H^* + \sigma Q^v \sigma)^{-1}$  % Kalman gain
   $\begin{bmatrix} \hat{X}_k^1 \\ \hat{X}_k^2 \end{bmatrix} \leftarrow \begin{bmatrix} \hat{X}_{k-}^1 \\ \hat{X}_{k-}^2 \end{bmatrix} + K_k \left( \log(y_k) - h \left( \begin{bmatrix} \hat{X}_{k-}^1 \\ \hat{X}_{k-}^2 \end{bmatrix} \right) \right)$ 
   $R_k \leftarrow (I - K_k H) R_{k-}$ 
% State estimate initialization of the next  $k$  loop
   $\hat{X}_0^1 \leftarrow \max(0, \hat{X}_k^1 + f_1(\hat{X}_k^1, \hat{X}_k^2) \delta)$ 
   $\hat{X}_0^2 \leftarrow \max(0, \hat{X}_k^2 + f_2(\hat{X}_k^1, \hat{X}_k^2) \delta)$ 
   $R_0 \leftarrow R_k + \left( R_k F^* + F R_k + g(\hat{X}_k^1, \hat{X}_k^2) Q^w g(\hat{X}_k^1, \hat{X}_k^2)^* \right) \delta$ 
End For

```

Algorithm 2: State estimation for the chemostat using EKF.

2.2 Unscented Kalman Filter

The application of the UKF to the stochastic model of the chemostat is given by Algorithm (3)-(4). Unlike The EKF, the UKF does not linearize the system, but uses the unscented transform ? to directly approximate the mean and covariance of the target distribution instead of trying to approximate the non-linear function. The idea of the unscented transform is to create a fixed number of wheighted sigma points $\tilde{\mathcal{X}}_{j-1,i}$ from the previous estimation of the mean and covariance matrix. Then, these sigma points are propagated

through the non-linearity in the state equation to obtain new sigma points $\hat{\mathcal{X}}_{j-,i}$, and the predicted mean and covariance of the current state are estimated from them. The same method is used in the correction step, we propagate the sigma points $\tilde{\mathcal{X}}_{k-,i}$ created from the predicted mean and covariance matrix through the non-linearity in the output equation. The resulting sigma points $\hat{\mathcal{Y}}_{k,i}$ are used to compute the Kalman gain and the estimation of the current state's mean and covariance matrix.

Again, because of the non-linearity between the noise and the state variables in the system equations (1) and the output equation (2), we should add the noise variables

in the state vector to create an augmented variable with mean $\tilde{X}_{j-1} = \begin{bmatrix} \hat{X}_{j-1}^1 \\ \hat{X}_{j-1}^2 \\ 0 \\ 0 \end{bmatrix}$ and

variance $\tilde{R}_{j-1} = \begin{bmatrix} R_{j-1} & 0 \\ 0 & Q^w \end{bmatrix}$ in the prediction step and $\tilde{X}_{k-} = \begin{bmatrix} \hat{X}_{k-}^1 \\ \hat{X}_{k-}^2 \\ 0 \end{bmatrix}$ and variance

$\tilde{R}_{k-} = \begin{bmatrix} R_{k-} & 0 \\ 0 & Q^v \end{bmatrix}$ in the correction step then compute the sigma points from them.

$[\cdot]_i$ denotes de i^{th} column of the matrix in between brackets.

```

% initialization
delta = T/(N_syst * N_obs)
X_0 ~ N(mu_0, Q_0)
X_hat_0 <- mu_0
R_0 = Q_0
n_1 = n_x + n_w % state dimension + state noise dimension
n_2 = n_x + n_v % state dimension + output noise dimension
lambda_1 = alpha^2(n_1 + kappa) - n_1 , lambda_2 = alpha^2(n_2 + kappa) - n_2
% weights computation
W_0^{m,l} = lambda_l / (n_l + lambda_l) , l = 1, 2
W_0^{c,l} = lambda_l / (n_l + lambda_l) + (1 - alpha^2 + beta) , l = 1, 2
W_i^{m,l} = lambda_l / (2(n_l + lambda_l)) , i = 1, ..., 2n_l, l = 1, 2
W_i^{c,l} = lambda_l / (2(n_l + lambda_l)) , i = 1, ..., 2n_l, l = 1, 2
% iterations
For k = 0, ..., N_obs Do
    For j = 1, ..., N_syst Do
        % prediction step
        X_tilde_{j-1} = [ X_hat_{j-1}^1
                        X_hat_{j-1}^2
                        0
                        0 ]
        R_tilde_{j-1} = [ R_{j-1}  0
                        0      Q^w ]
        X_tilde_{j-1,0} = X_tilde_{j-1}
        X_tilde_{j-1,i} = X_tilde_{j-1} + sqrt(n_1 + lambda_1) [ sqrt(R_tilde_{j-1}) ]_i , i = 1, ..., n_1
        X_tilde_{j-1,i+n_1} = X_tilde_{j-1} - sqrt(n_1 + lambda_1) [ sqrt(R_tilde_{j-1}) ]_i , i = 1, ..., n_1
        X_hat_{j-,i} = max(0, X_tilde_{j-1,i}^x + f(X_tilde_{j-1,i}^x)delta + g(X_tilde_{j-1,i}^x)sqrt(delta)X_tilde_{j-1,i}^w) , i = 0, ..., 2n_1
        % where X_tilde_{j-1,i}^x denotes the n_x first components of X_tilde_{j-1,i} and X_tilde_{j-1,i}^w denotes its n_w last
        components
        X_hat_{j-} = sum_{i=0}^{2n_1} W_i^{m,1} X_hat_{j-,i}
        R_{j-} = sum_{i=0}^{2n_1} W_i^{c,1} (X_hat_{j-,i} - X_hat_{j-})(X_hat_{j-,i} - X_hat_{j-})^*
        X_hat_j <- X_hat_{j-}
        R_j <- R_{j-}
    End For

```

Algorithm 3: State estimation for the chemostat using UKF (1st section)

% correction step

$$\tilde{X}_{k^-} = \begin{bmatrix} \hat{X}_{k^-}^1 \\ \hat{X}_{k^-}^2 \\ 0 \end{bmatrix}$$

$$\tilde{R}_{k^-} = \begin{bmatrix} R_{k^-} & 0 \\ 0 & Q^v \end{bmatrix}$$

$$\tilde{\mathcal{X}}_{k^-,0} = \tilde{X}_{k^-}$$

$$\tilde{\mathcal{X}}_{k^-,i} = \tilde{X}_{k^-} + \sqrt{n_2 + \lambda_2} \begin{bmatrix} \sqrt{\tilde{R}_{k^-}} \\ \end{bmatrix}_i, \quad i = 1, \dots, n_2$$

$$\tilde{\mathcal{X}}_{k^-,i+n_2} = \tilde{X}_{k^-} - \sqrt{n_2 + \lambda_2} \begin{bmatrix} \sqrt{\tilde{R}_{k^-}} \\ \end{bmatrix}_i, \quad i = 1, \dots, n_2$$

$$\hat{\mathcal{Y}}_{k,i} = h(\tilde{\mathcal{X}}_{k^-,i}^x) + \sigma(\tilde{\mathcal{X}}_{k^-,i}^x) \tilde{\mathcal{X}}_{k^-,i}^v, \quad i = 0, \dots, 2n_2$$

% where $\tilde{\mathcal{X}}_{k^-,i}^x$ denotes the n_x first components of $\tilde{\mathcal{X}}_{k^-,i}$ and $\tilde{\mathcal{X}}_{k^-,i}^v$ denotes its n_v last components

$$\mu = \sum_{i=0}^{2n_2} W_i^{m,2} \hat{\mathcal{Y}}_{k,i}$$

$$S = \sum_{i=0}^{2n_2} W_i^{c,2} (\hat{\mathcal{Y}}_{k,i} - \mu)(\hat{\mathcal{Y}}_{k,i} - \mu)^*$$

$$C = \sum_{i=0}^{2n_2} W_i^{c,2} (\tilde{\mathcal{X}}_{k^-,i} - \hat{X}_{k^-})(\hat{\mathcal{Y}}_{k,i} - \mu)^*$$

$$K_k = C S^{-1} \quad \% \text{ Kalman gain}$$

$$\hat{X}_k \leftarrow \hat{X}_{k^-} + K_k (y_k - \mu)$$

$$R_k \leftarrow R_{k^-} + K_k S_k K_k^*$$

% State estimate initialization of the next k loop

$$\tilde{X}_k = \begin{bmatrix} \hat{X}_k^1 \\ \hat{X}_k^2 \\ 0 \\ 0 \end{bmatrix}$$

$$\tilde{R}_k = \begin{bmatrix} R_k & 0 \\ 0 & Q^w \end{bmatrix}$$

$$\tilde{\mathcal{X}}_{k,0} = \tilde{X}_k$$

$$\tilde{\mathcal{X}}_{k,i} = \tilde{X}_k + \sqrt{n_1 + \lambda_1} \begin{bmatrix} \sqrt{\tilde{R}_k} \\ \end{bmatrix}_i, \quad i = 1, \dots, n_1$$

$$\tilde{\mathcal{X}}_{k,i+n_1} = \tilde{X}_k - \sqrt{n_1 + \lambda_1} \begin{bmatrix} \sqrt{\tilde{R}_k} \\ \end{bmatrix}_i, \quad i = 1, \dots, n_1$$

$$\hat{\mathcal{X}}_{0^-,i} = \max(0, \tilde{\mathcal{X}}_{k,i}^x + f(\tilde{\mathcal{X}}_{k,i}^x) \delta + g(\tilde{\mathcal{X}}_{k,i}^x) \sqrt{\delta} \tilde{\mathcal{X}}_{k,i}^w), \quad i = 0, \dots, 2n_1$$

$$\hat{X}_0 = \sum_{i=0}^{2n_1} W_i^{m,1} \hat{\mathcal{X}}_{0^-,i}$$

$$R_0 = \sum_{i=0}^{2n_1} W_i^{c,1} (\hat{\mathcal{X}}_{0^-,i} - \hat{X}_0)(\hat{\mathcal{X}}_{0^-,i} - \hat{X}_0)^*$$

End For

Algorithm 4: State estimation for the chemostat using UKF (2nd section)

2.3 Particle Filter “Bootstrap”

The application of particle filter to the stochastic model of the chemostat is given by Algorithm (5). This application was first carried out in ?. The particle filter gives a

Monte Carlo approximation to the filtering density $p_{X_{t_k}|Y_{0:k}}(x_{t_k}|y_{0:k})$ of the form

$$p_{X_{t_k}|Y_{0:k}}(x_{t_k}|y_{0:k}) \approx \frac{1}{N} \sum_{i=1}^N \delta_{\xi^i}(x_{t_k}) \quad (4)$$

where ξ^i are the particles and they are distributed according to $p_{X_{t_k}|Y_{0:k}}(x_{t_k}|y_{0:k})$ and δ_{ξ^i} is the Dirac measure at point ξ^i . However, since the density $p_{X_{t_k}|Y_{0:k}}(x_{t_k}|y_{0:k})$ is unknown, the particles are drawn from a proposal density $q_{X_{t_k}|Y_{0:k}}(x_{t_k}|y_{0:k})$ and the approximation is corrected using weights ω^i . The simplest method to do this is to consider the process transition density $p_{X_t|X_{t-\delta}}(x_t|x_{t-\delta})$ as a proposal density and to compute the weights from the likelihood function $p_{Y_k|X_{t_k}}(y_k|x_{t_k})$ given by :

$$p_{Y_k|X_{t_k}}(y_k|x_{t_k}) = \frac{1}{\sigma x_{t_k}^2 \sqrt{2\pi}} \exp\left(-\frac{1}{2(\sigma x_{t_k}^2)^2} |y_k - x_{t_k}^2|^2\right) \quad (5)$$

In Algorithm (5), we used the state equation (1) to approximate the transition density $p_{X_t|X_{t-\delta}}(x_t|x_{t-\delta})$ and to draw the particles ξ^i . The number of particles used is $N = 1000$. A resampling procedure is performed within the correction step to replace the insignificant low-weight particles by the high-weight ones. There exist several resampling methods which are compared in ?. For our algorithm, we used the residual resampling method.

```

% initialization
 $\delta = T / (N_{syst} * N_{obs})$ 
N = number of particles
 $w_0^1, \dots, w_{N_{syst} * N_{obs}}^1 \sim N(0, 1)$ 
 $w_0^2, \dots, w_{N_{syst} * N_{obs}}^2 \sim N(0, 1)$ 
 $\xi^{1:N} \stackrel{iid}{\sim} \tilde{p}_{X_0}(x)$  , where  $\xi^{1:N} = \begin{bmatrix} \xi_{X^1}^{1:N} \\ \xi_{X^2}^{1:N} \end{bmatrix}$  % initialization
 $\omega^i \leftarrow \frac{1}{N}$   $i = 1, \dots, N$  % importance weights
 $\hat{X}_0^1 = \frac{1}{N} \sum_{i=1}^N \xi_{X^1}^i$ 
 $\hat{X}_0^2 = \frac{1}{N} \sum_{i=1}^N \xi_{X^2}^i$ 
% iterations
For  $k = 0, \dots, N_{obs}$  Do
  For  $j = 1, \dots, N_{syst}$  Do
    % prediction step
     $\mu(\xi_{X^2}^i) = \mu_{max} \frac{\xi_{X^2}^i}{k_s + \xi_{X^2}^i}$   $i = 1, \dots, N$ 
     $\tilde{\xi}_{X^1}^i = \max\left(0, \xi_{X^1}^i + (\mu(\xi_{X^2}^i) - D) \xi_{X^1}^i \delta + c_1 \sqrt{\xi_{X^1}^i} \sqrt{\delta} w_j^1\right)$   $i = 1, \dots, N$ 
     $\tilde{\xi}_{X^2}^i = \max\left(0, \xi_{X^2}^i - k_{cs} \mu(\xi_{X^2}^i) \xi_{X^1}^i \delta + D (s_{in} - \xi_{X^2}^i) \delta + c_2 \sqrt{\xi_{X^2}^i} \sqrt{\delta} w_j^2\right)$   $i =$ 
    1, ..., N
     $\xi^i \leftarrow \tilde{\xi}^i$ 
     $\hat{X}_j^1 = \frac{1}{N} \sum_{i=1}^N \xi_{X^1}^i$ 
     $\hat{X}_j^2 = \frac{1}{N} \sum_{i=1}^N \xi_{X^2}^i$ 
  End For
    % correction step
     $\omega^i \leftarrow p_{Y_k|X_k}(y_k | \tilde{\xi}^i)$   $i = 1, \dots, N$  % importance weights
     $\omega^i \leftarrow \omega^i / \text{sum}(\omega^{1:N})$   $i = 1, \dots, N$  % normalization
     $\xi^{1:n} \leftarrow \text{resample}(\tilde{\xi}^{1:n}, \omega^{1:n})$  % resampling step
     $\hat{X}_k^1 = \frac{1}{N} \sum_{i=1}^N \xi_{X^1}^i$ 
     $\hat{X}_k^2 = \frac{1}{N} \sum_{i=1}^N \xi_{X^2}^i$ 
    % State estimate initialization of the next k loop
     $\mu(\xi_{X^2}^i) = \mu_{max} \frac{\xi_{X^2}^i}{k_s + \xi_{X^2}^i}$   $i = 1, \dots, N$ 
     $\tilde{\xi}_{X^1}^i = \max\left(0, \xi_{X^1}^i + (\mu(\xi_{X^2}^i) - D) \xi_{X^1}^i \delta + c_1 \sqrt{\xi_{X^1}^i} \sqrt{\delta} w_0^1\right)$   $i = 1, \dots, N$ 
     $\tilde{\xi}_{X^2}^i = \max\left(0, \xi_{X^2}^i - k_{cs} \mu(\xi_{X^2}^i) \xi_{X^1}^i \delta + D (s_{in} - \xi_{X^2}^i) \delta + c_2 \sqrt{\xi_{X^2}^i} \sqrt{\delta} w_0^2\right)$   $i =$ 
    1, ..., N
     $\xi^i \leftarrow \tilde{\xi}^i$ 
  End For

```

Algorithm 5: State estimation for the chemostat using the Bootstrap particle filter.

3 Simulation results

A comparison between the three previous filtering algorithms is proposed in this section to find out which one has the best performances for the chemostat's state estimation. In perfect conditions, all these methods give good results. However, the proposed second degree model is a drastic simplification of the real system, the biomass and substrate concentrations could have very large and noisy variations and unknown initial conditions. The output of this system is measured at widely separated instants and could have a very high noise variance. Therefore, the filtering algorithm should stay able to follow the state's variations, should be robust against high variance observation noises and far initial conditions, and must remain efficient even for low-frequency observations. For these reasons, many tests are proposed to select the most suited algorithm for the chemostat model.

First, we present the estimation results of each algorithm in the normal conditions and compare their estimation errors. Fig 1 and Fig 2 represent the estimation of the biomass and substrat concentrations respectively using the EKF; Fig 3 and Fig 4 represent their estimation using the UKF and Fig 5 and Fig 6 using the particle filter. The estimation error comparison is represented in Fig 7.

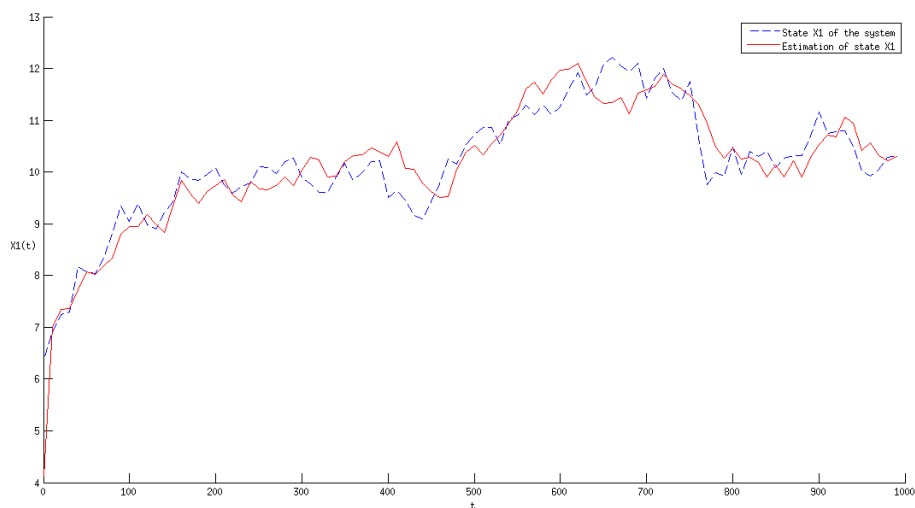


Figure 1: State X^1 estimation using EKF

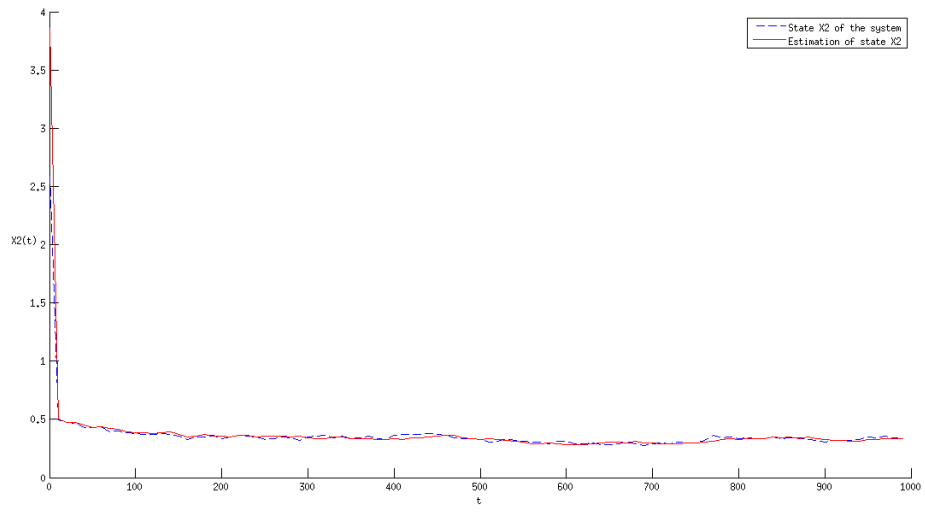


Figure 2: State X^2 estimation using EKF

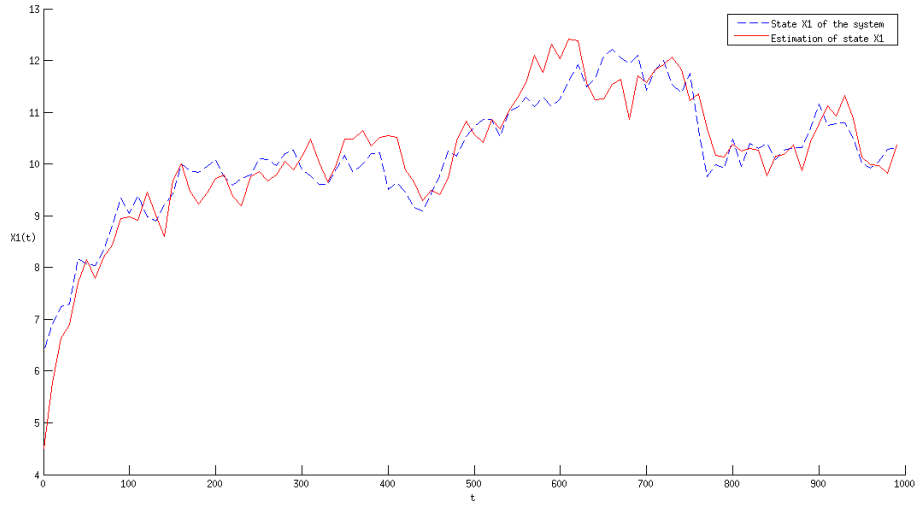


Figure 3: State X^1 estimation using UKF

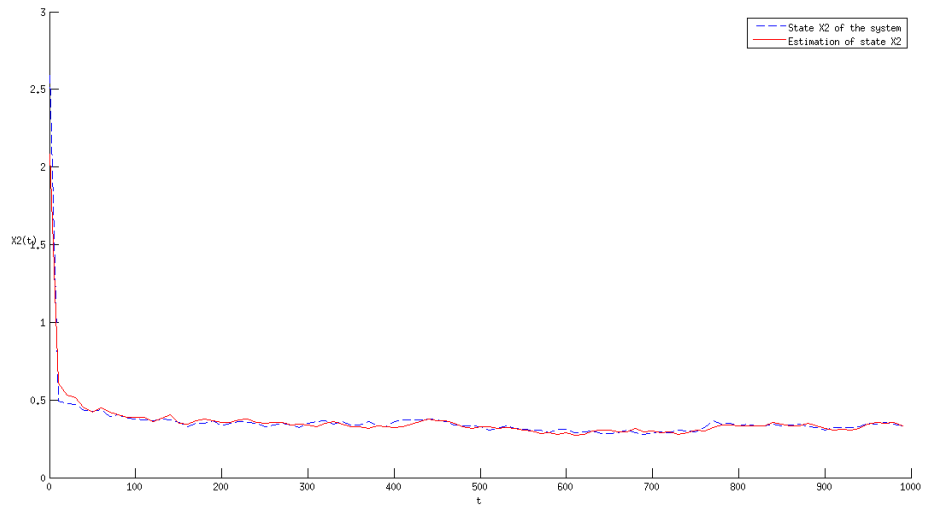


Figure 4: State X^2 estimation using UKF

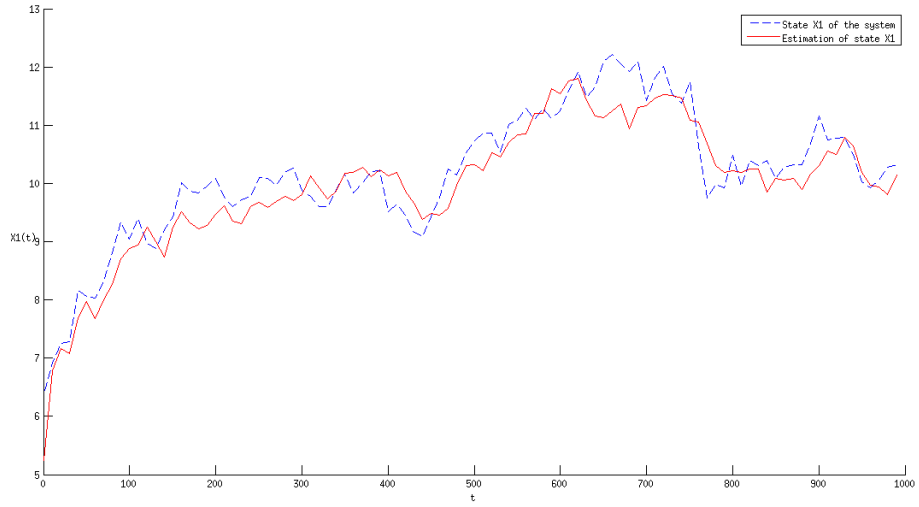


Figure 5: State X^1 estimation using PF

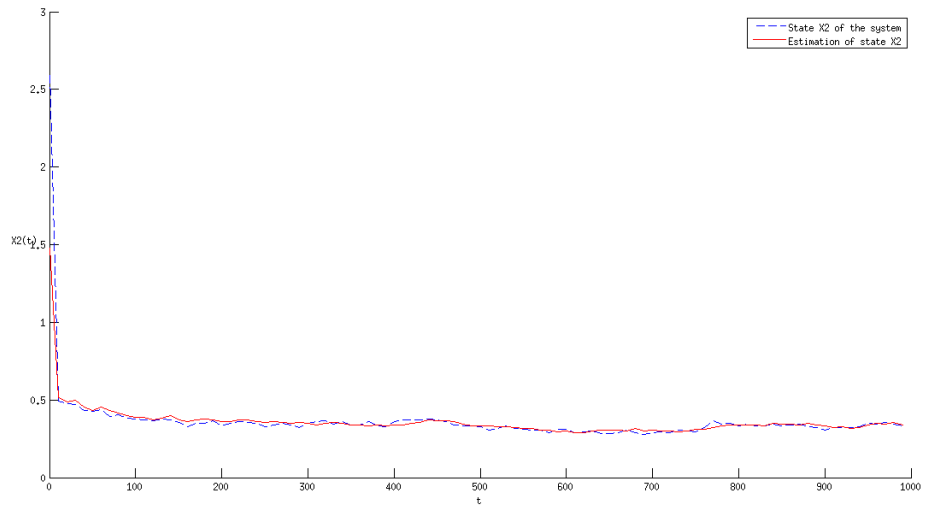


Figure 6: State X^2 estimation using PF

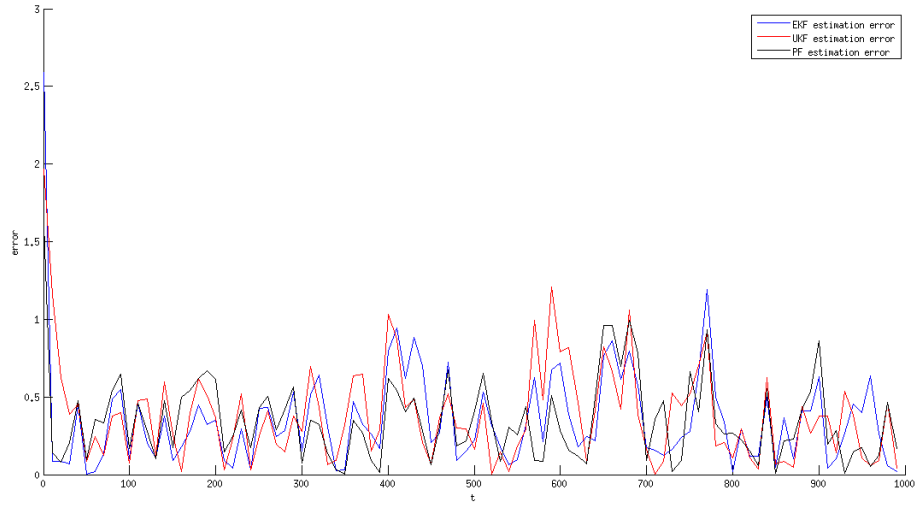


Figure 7: Estimation error comparison