Decentralized Estimation of Overflow Losses in a Hopper-Dredger

Zs. Lendek [a,*] R. Babuška [a] J. Braaksma [a] C. de Keizer [b]

[a] *Delft Center for Systems and Control, Delft University of Technology*
Mekelweg 2, 2628 CD Delft, The Netherlands

[b] *IHC Systems B.V. P.O. Box 41, 3360 AA Sliedrecht, The Netherlands*

Abstract

The Kalman filter and its nonlinear variants have been widely used for filtering and state estimation. However, models with severe nonlinearities are not handled well by Kalman filters. Such an application is presented in this paper: the estimation of the overflow losses in a hopper-dredger. The overflow mixture density and flow-rate have to be estimated based on noisy measurements of the total hopper volume, mass, incoming mixture density and flow-rate. We propose a decomposition of the nonlinear process model into two simpler subsystems. Different types of observers are considered for each sub-process. The performance is evaluated for simulated and real-world data and compared for the centralized and distributed settings and four combinations of the particle filter and the unscented Kalman filter. The results indicate that the distributed observer achieves the same performance as the centralized one, while leading to increased modularity, reduced complexity, lower computational costs and easier tuning.

Key words: State estimation, nonlinear systems, nonlinear observers, Kalman filter, Particle filter

* Corresponding author. Tel.:+31 (0)15 27 88573, Fax:+31 (0)15 27 86679
Email addresses: zs.lendek@tudelft.nl (Zs. Lendek),

1 Introduction

Many problems require the estimation of states and possibly uncertain parameters based on a dynamic system model and a sequence of noisy measurements. Dynamic systems are usually modeled in the state-space framework, using a state-transition model, which describes the evolution of states over time and a measurement model, which relates the measurement to the states. These models can be deterministic as well as stochastic.

The most well-known and widely used probabilistic estimation methods are the Kalman filter (KF) and its extension to nonlinear systems, the Extended Kalman Filter (EKF) (Kalman, 1960; Welch and Bishop, 2002). However, these methods have severe limitations and may become unstable even for linear processes. The Unscented Kalman Filter (UKF), introduced by Julier and Uhlmann (1997), overcomes some of these deficiencies. The estimates obtained by the UKF are in general more accurate, since the filter does not rely on linearization, but uses directly the nonlinear state-transition function. Its superior performance has been reported in several publications (Li *et al.*, 2004; Stenger *et al.*, 2001; van der Merwe and Wan, 2003). Though more accurate and reliable than the EKF, the UKF still assumes a unimodal distribution of the states and the handling of multimodal distributions remains problematic.

Over the last years, particle filters (PF) (Doucet *et al.*, 2000; Arulampalam *et al.*, 2002) have been extensively studied. These filters have been successfully applied to state-estimation problems, mainly in the field of tracking (Hue *et al.*, 2002; Nait-Charif and McKenna, 2004; Sullivan *et al.*, 2001). The basic idea behind this technique is to represent probability densities by a set of samples. Thus, a wide range of probability densities can be represented, allowing the handling of nonlinear, non-

`r.babuska@tudelft.nl` (R. Babuška), `j.braaksma@tudelft.nl` (J. Braaksma), `cdkeizer@ihcsystems.com` (C. de Keizer).

Gaussian dynamical systems. However, this representation comes with a higher computational cost, which may render the filter unusable for online or real-time estimation.

Since the above-mentioned methods are suboptimal, their performance varies, depending on the application considered. While for a highly nonlinear and non-Gaussian model a particle filter is likely the best option, UKF may also yield good performance with considerably lower computational costs.

Decentralized estimation has been studied in the context of large-scale processes and distributed systems. The architecture in general takes the form of a network of sensor nodes, each with its own processing facility. In case of a fully decentralized system, computation is performed locally and communication occurs between any two nodes. Each node shares information with other nodes and computes a local estimate. Computation and communication is distributed over the network so that a global estimate can be computed. Several topologies have been proposed, depending on the particular application. In case of large scale processes (Vadigepalli and Doyle, 2003a,b), the network is in general in a hierarchical form, with several intermediate and one final fusion node. For distributed systems, such as multiagent societies (López-Orozco *et al.*, 2000; Roumeliotis and Bekey, 2002; Schmitt *et al.*, 2002), several fusion nodes exist, which process the data and send the information to the rest of the nodes. Observers for distributed estimation include, but are not limited to decentralized Kalman and Extended Kalman filter (Durrant-Whyte *et al.*, 1990), information filter and particle filters (Bolic *et al.*, 2004; Coates, 2004).

In this paper, we propose the decomposition of the nonlinear system model into cascaded subsystems, with the possibility of using different estimation methods for the subsystems. The idea behind this type of estimation is that many nonlinear systems can be represented as cascaded, observable subsystems, which alone are less complex than the original system. Separate observers can be designed for the individual subsystems. Moreover, different types of observers can be combined, depending on

the complexity and nonlinearity of the subsystems. This setting can be regarded as a cooperative multiagent system. Each agent has the task of observing one of the subsystems, possibly using different methods and relying on its own measurement and the information gathered from other agents. In turn, each agent communicates its own results to other agents.

The proposed distributed observer design is applied to the estimation of the overflow losses in a hopper-dredger. The estimation of overflow losses is an essential step toward the optimization of the separation process in the hopper, which is of vital importance for future improvement in efficiency, accuracy and from the viewpoint of labor saving. In the considered process, only two of the three state variables are measured, both heavily corrupted by noise, and the remaining state need to be estimated online for monitoring and control purposes. The system is highly nonlinear, and for global state estimation a particle filter would be required. However, the model can be represented as two cascaded subsystems, which allows the use of two observers. We consider for these observers the combinations of UKF and PF and compare the performance of a centralized PF for the whole system with the four possible combinations in the distributed setting, both on simulated and experimental data.

The structure of the paper is as follows. Section 2 reviews the Unscented Kalman Filter and the Particle Filter methodology. In Section 3, the proposed cascaded observer setting is given, while Section 4 presents the dynamic sedimentation model and the models used for estimation purposes. Sections 5 and 6 give the results for the simulated and experimental data, respectively, and Section 7 discusses these results. Finally, Section 8 concludes the paper.

2 Estimation Methods

In this section, two methods for estimating the states and parameters of a nonlinear system are presented. Consider the following possibly time-varying, discrete-time nonlinear system:

$$x_k = f(x_{k-1}, v_{k-1}) \tag{1}$$

$$y_k = h(x_k, \eta_k) \tag{2}$$

where:

- k - current time step
- x - state variables
- v, η - noises of known distributions
- y - measurements
- f - state transition model
- h - measurement model

Note that the functions f and h may also depend on known inputs or other known parameters. However, for the ease of notation, these variables are omitted. It is assumed that the system (1)–(2) is observable, in order to be able to estimate the states.

The goal is to estimate the states of interest. We consider two methods: the Unscented Kalman filter and the Particle Filter. Both methods use notions from probability theory, however, the UKF is a deterministic method, while particle filters are stochastic. Both filters are recursive algorithms, that use all the provided information (model and observations) to estimate the current state of the system. The filters work in two steps: prediction and update. The *prediction* step uses the system model and the information incorporated so far in order to predict the process' states. This stage is also known as the time update step, as it projects the current state forward in time. The *update* stage uses the latest measurement to modify (cor-

rect) the projected state. This stage is also known as measurement update, since it incorporates the information brought by the new measurement.

2.1 Unscented Kalman Filter

In the case of linear systems, corrupted by white Gaussian noise, the Kalman filter is proved to be an optimal filter in the least mean square sense. For nonlinear systems, several extensions exist: the extended Kalman filter (based on linearizing the models around the current states), and the family of sigma point Kalman filters (Julier and Uhlmann, 2002a; van der Merwe, 2004).

The fundamental problem of the EKF is that the approximations will only be valid if all the higher order derivatives of the nonlinear functions are effectively zero around the current estimate. Another variation of the Kalman filter for nonlinear systems, the Unscented Kalman Filter (UKF) was developed by Julier and Uhlmann (1997). Since the original formulation of the UKF, a number of sigma-point filters were described, which are based on approximating the distribution of the states by deterministically chosen samples (sigma points) and differ mostly on how these samples are generated. These filters preserve the normal distribution and are valid up to the second order of the Taylor expansion of the nonlinear functions.

The UKF is based on the unscented transformation, which computes the statistics of a random variable undergoing a nonlinear transformation. To compute the statistics, a number of weighted samples called “sigma points” are chosen deterministically, so that they completely capture the mean and covariance of the random variable.

2.1.1 *Unscented Transformation*

Assume that an n_x-dimensional random variable x has to be propagated through the nonlinear function g in order to generate y:

$$y = g(x)$$

Assume also that x has a known mean x_0 and a known covariance P_x. In this case, $2n_x + 1$ sigma points can be generated deterministically, so that they capture the mean and variance. One of the selection procedures (Julier and Uhlmann, 2002b) computes the sigma points as follows:

$$\begin{aligned}
&\mathcal{X}_0 = x_0 && w_0 = \kappa/(n_x + \kappa) && \\
&\mathcal{X}_i = x_0 + [\sqrt{(n_x + \kappa)P_x}]_i && w_i = 1/[2(n_x + \kappa)] && i = 1, 2, ...n_x \\
&\mathcal{X}_i = x_0 - [\sqrt{(n_x + \kappa)P_x}]_i && w_i = 1/[2(n_x + \kappa)] && i = n_x + 1, ...2n_x
\end{aligned} \tag{3}$$

where κ is a scaling parameter, $[\sqrt{(n_x + \kappa)P_x}]_i$ is the i^{th} row of the matrix square root of $(n_x + \kappa)P_x$, and w_i is the weight associated with the i^{th} sample.

The sigma points are now propagated though the nonlinear function g: $\mathcal{Y}_i = g(\mathcal{X}_i)$, $i = 0, 1, \ldots, 2n_x$. The mean and covariance of y are estimated as:

$$y_0 = \sum_{i=0}^{2n_x} w_i \mathcal{Y}_i \tag{4}$$

$$P_y = \sum_{i=0}^{2n_x} w_i (\mathcal{Y}_i - y_0)(\mathcal{Y}_i - y_0)^T \tag{5}$$

These estimates are accurate to the second order of the Taylor series expansion of $g(x)$, for any nonlinear function. However, in certain cases the computed covariance matrix can be non-positive semidefinite, in which case the filter collapses.

2.1.2 *Unscented Kalman Filter Algorithm*

In order to apply the Kalman filter using the unscented transformation to the system (1-2), the state variables are augmented with the state transition and measurement

noise, and the state covariance with the state transition and measurement covariance. Assuming that the noises have means v and η, and covariances Q and R, respectively, the augmented variables can be expressed as:

$$\begin{aligned} x^a_{k-1} &= [x^T_{k-1}\ v^T_{k-1}\ \eta^T_{k-1}]^T \qquad n_a = \dim(x^a_{k-1}) \\ P^a_{k-1} &= \mathrm{diag}([P^x_{k-1}\ Q_{k-1}\ R_{k-1}]) \end{aligned} \tag{6}$$

The sigma points are computed based on the augmented state and covariance.

The prediction is extended with respect to the Kalman filter, since the sigma points have to be computed and propagated through the state transition model to predict the new states. The predicted states also have to be propagated through the measurement model in order to predict the measurement. The equations are described below.

Generate the sigma points from (6), according to (3). κ provides an extra degree of freedom to "fine tune" the higher order moments of the approximation, and can be used to reduce the overall prediction errors. When the states are assumed Gaussian, a useful heuristic is to select $n_a + \kappa = 3$ (Julier and Uhlmann, 2002b). For other distributions, a different choice of κ might be more appropriate. In this paper we assume that the states are Gaussian and follow the guideline above.

Since the augmented state from which the sigma points are generated has the form (6), the sigma points also have the form:

$$\mathcal{X}^a_{i,k-1} = [\mathcal{X}^x_{k-1}\ \mathcal{X}^v_{k-1}\ \mathcal{X}^\eta_{k-1}]_i, \qquad i = 0, \ldots, 2n_a \tag{7}$$

Propagate the sigma points through the state transition model:

$$\mathcal{X}^x_{i,k|k-1} = f(\mathcal{X}^x_{i,k-1},\ \mathcal{X}^v_{i,k-1}) \qquad i = 0, \ldots, 2n_a \tag{8}$$

Predict the next state:

$$\hat{x}_{k|k-1} = \sum_{i=0}^{2n_a} w_i \mathcal{X}^x_{i,k|k-1} \tag{9}$$

and covariance:

$$P^x_{k|k-1} = \sum_{i=0}^{2n_a} w_i(\mathcal{X}^x_{i,k|k-1} - \hat{x}_{k|k-1})(\mathcal{X}^x_{i,k|k-1} - \hat{x}_{k|k-1})^T \tag{10}$$

Propagate the transformed sigma points through the measurement model:

$$\mathcal{Y}_{i,k|k-1} = h(\mathcal{X}^x_{i,k|k-1},\ \mathcal{X}^\eta_{i,k-1}) \qquad i = 0, \ldots, 2n_a \tag{11}$$

Predict the measurement:

$$\hat{y}_{k|k-1} = \sum_{i=0}^{2n_a} w_i \mathcal{Y}_{i,k|k-1} \tag{12}$$

and its covariance:

$$P^y_{k|k-1} = \sum_{i=0}^{2n_a} w_i(\mathcal{Y}_{i,k|k-1} - \hat{y}_{k|k-1})(\mathcal{Y}_{i,k|k-1} - \hat{y}_{k|k-1})^T \tag{13}$$

Compute the cross-correlation matrix:

$$P^{xy}_{k|k-1} = \sum_{i=0}^{2n_a} w_i(\mathcal{X}^x_{i,k|k-1} - \hat{x}_{k|k-1})(\mathcal{Y}_{i,k|k-1} - \hat{y}_{k|k-1})^T \tag{14}$$

The update stage remains the same as in the Kalman filter:

Compute the Kalman gain:

$$K_k = P^{xy}_{k|k-1}(P^y_{k|k-1})^{-1} \tag{15}$$

Correct the predicted state:

$$\hat{x}_k = \hat{x}_{k|k-1} + K_k(y_k - \hat{y}_{k|k-1}) \tag{16}$$

Correct the covariance:

$$P^x_k = P^x_{k|k-1} + K_k P^y_{k|k-1} K^T_k \tag{17}$$

A generic UKF algorithm is given in Algorithm 1.

The presented procedure is a general form of the unscented Kalman filter. For special cases, such as additive state transition and/or measurement noise, the computational complexity can be reduced (Julier and Uhlmann, 1997).

Algorithm 1 Unscented Kalman filter

Input: $u,\ y,\ Q,\ R,\ f,\ h,\ P_0,\ x_0,\ v_0,\ \eta_0$

Output: $x,\ P$

for $k = 1, 2, \ldots$ **do** ▷ each sample

Prediction:

$x^a_{k-1} = [x^T_{k-1}\ v^T_{k-1}\ \eta^T_{k-1}]^T \qquad n_a = \dim(x^a_{k-1})$ ▷ augment states

$P^a_{k-1} = \mathrm{diag}[P^x_{k-1}\ Q_{k-1}\ R_{k-1}]$ ▷ augment covariance

$\mathcal{X}^a_{0,k-1} = x^a_{k-1}$

$\mathcal{X}^a_{i,k-1} = x^a_{k-1} + [\sqrt{(n_a+\kappa)P^a_{k-1}}]_i \qquad i = 1, 2, ...n_a$

$\mathcal{X}^a_{i,k-1} = x^a_{k-1} - [\sqrt{(n_a+\kappa)P^a_{k-1}}]_i \qquad i = n_a+1, ...2n_a$ ▷ compute sigma points

$\mathcal{X}^x_{i,k|k-1} = f(\mathcal{X}^x_{i,k-1},\ \mathcal{X}^v_{i,k-1}) \qquad i = 0, \ldots, 2n_a$ ▷ propagate sigma points

$\hat{x}_{k|k-1} = \sum_{i=0}^{2n_a} w_i \mathcal{X}^x_{i,k|k-1}$ ▷ predict next state

$P^x_{k|k-1} = \sum_{i=0}^{2n_a} w_i (\mathcal{X}^x_{i,k|k-1} - \hat{x}_{k|k-1})(\mathcal{X}^x_{i,k|k-1} - \hat{x}_{k|k-1})^T$ ▷ predict covariance

$\mathcal{Y}_{i,k|k-1} = h(\mathcal{X}^x_{i,k|k-1},\ \mathcal{X}^\eta_{i,k-1}) \qquad i = 0, \ldots, 2n_a$ ▷ propagate transformed sigma points

$\hat{y}_{k|k-1} = \sum_{i=0}^{2n_a} w_i \mathcal{Y}_{i,k|k-1}$ ▷ predict measurement

$P^y_{k|k-1} = \sum_{i=0}^{2n_a} w_i (\mathcal{Y}_{i,k|k-1} - \hat{y}_{k|k-1})(\mathcal{Y}_{i,k|k-1} - \hat{y}_{k|k-1})^T$ ▷ predict measurement covariance

$P^{xy}_{k|k-1} = \sum_{i=0}^{2n_a} w_i (\mathcal{X}^x_{i,k|k-1} - \hat{x}_{k|k-1})(\mathcal{Y}_{i,k|k-1} - \hat{y}_{k|k-1})^T$ ▷ cross-correlation matrix

Update:

$K_k = P^{xy}_{k|k-1} P_(k|k-1^y)^{-1}$ ▷ Kalman gain

$\hat{x}_k = \hat{x}_{k|k-1} + K_k(y_k - \hat{y}_{k|k-1})$ ▷ correct the state

$P^x_k = P^x_{k|k-1} + K_k P^y_{k|k-1} K^T_k$ ▷ correct the covariance

end for

The Unscented Kalman Filter is not restricted to Gaussian noises, though its best performance is achieved when the random variables are Gaussian.

The UKF is a rather general solution for nonlinear state estimation, but it cannot be used successfully in all situations. The filter may collapse due to the lack of robustness: the estimated posterior covariance can increase in an unbounded fashion in the case of model-plant mismatch.

2.2 Particle Filters

Most Kalman filters represent the distribution of the random variables as Gaussians. However, for arbitrary distributions, there is no general method to compute the resulting distribution analytically. Moreover, these methods may become unstable for highly nonlinear processes. This is why the particle filters represent the distributions by samples, which can be easily computed with, rather than by a compact parametric form.

The particle filter (PF) uses probabilistic models for the state transition function and the measurement function, respectively:

$$p(x_k|x_{k-1}), \qquad p(y_k|x_k)\,.$$

The objective is to recursively construct the posterior PDF $p(x_k|y_k)$ of the state, given the measured output y_k and assuming conditional independence of the measurement sequence, given the states. The PF works in two stages:

(1) The *prediction stage* uses the state-transition model to predict the state PDF one step ahead. The PDF obtained is called the *prior*.
(2) The *update stage* uses the current measurement to correct the prior via the Bayes rule. The PDF obtained after the update is called the *posterior* PDF.

Particle filters represent the PDF by N random samples (particles) x_k^i with their associated weights w_k^i, normalized so that $\sum_{i=1}^{N} w_k^i = 1$. At time instant k, the posterior obtained in the previous step, $p(x_{k-1}|y_{k-1})$, is represented by N samples x_{k-1}^i and the corresponding weights w_{k-1}^i. To approximate the posterior $p(x_k|y_k)$,

new samples x_k^i and weights w_k^i are generated. Samples x_k^i are drawn from a (chosen) *importance density function* $q(x_k^i|x_{k-1}^i, y_k)$, and the weights are updated, using the current measurement y_k

$$\tilde{w}_k^i = w_{k-1}^i \frac{p(y_k|x_k^i)\, p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{k-1}^i, y_k)} \tag{18}$$

and normalized

$$w_k^i = \frac{\tilde{w}_k^i}{\sum_{j=1}^N \tilde{w}_k^j}\,.$$

If the importance density $q(x_k|x_{k-1}, y_k)$ is chosen equal to the state-transition PDF $p(x_k|x_{k-1})$, the weight update equation (18) becomes:

$$\tilde{w}_k^i = w_{k-1}^i p(y_k|x_k^i)\,.$$

The use of the transition prior as the importance density is a common choice (Arulampalam *et al.*, 2002) and it has the advantage that it can be easily sampled and the weights are easily evaluated.

The posterior PDF is represented by the set of weighted samples, conventionally denoted by:

$$p(x_k|y_k) \approx \sum_{i=1}^{N} w_k^i \delta(x_k - x_k^i)$$

where δ is the Dirac delta measure.

The PF algorithm is summarized in Algorithm 2. A common problem of PF is the particle degeneracy: after several iterations, all but one particle will have negligible weights. Therefore, particles must be resampled. A standard measure of the degeneracy is the effective sample size:

$$N_{\text{eff}} = \frac{1}{\sum_{i=1}^{N} (w_k^i)^2}$$

If N_{eff} drops below a specified threshold $N_{\text{T}} \in [1, N]$, particles are resampled by using Algorithm 3.

Algorithm 2 Particle filter

Input: $p(x_k|x_{k-1})$, $p(y_k|x_k)$, $p(x_0)$, N, N_{T}

Initialize:

for $i = 1, 2, \ldots, N$ **do**

Draw a new particle: $x_1^i \sim p(x_0)$

Assign weight: $w_1^i = \frac{1}{N}$

end for

At every time step $k = 2, 3, \ldots$

for $i = 1, 2, \ldots, N$ **do**

Draw a particle from importance distribution: $x_k^i \sim p(x_k^i|x_{k-1}^i)$

Use the measured y_k to update the weight: $\tilde{w}_k^i = w_{k-1}^i p(y_k|x_k^i)$

end for

Normalize weights: $w_k^i = \frac{\tilde{w}_k^i}{\sum_{j=1}^N \tilde{w}_k^j}$

if $\frac{1}{\sum_{i=1}^N (w_k^i)^2} < N_{\mathrm{T}}$ **then**

Resample using Algorithm 3.

end if

Algorithm 3 Resampling

Input: $\{(x^i, w^i)\}_{i=1}^N$

Output: $\{(x_{\mathrm{new}}^i, w_{\mathrm{new}}^i)\}_{i=1}^N$

for $i = 1, 2, \ldots, N$ **do**

Compute cumulative sum of weights: $w_c^i = \sum_{j=1}^i w_k^j$

end for

Draw u_1 from $\mathcal{U}(0, \frac{1}{N})$

for $i = 1, 2, \ldots, N$ **do**

Find x^{+i}, the first sample for which $w_c^i \geq u_i$.

Replace particle i: $x_{\mathrm{new}}^i = x^{+i}$, $w_{\mathrm{new}}^i = \frac{1}{N}$

$u_{i+1} = u_i + \frac{1}{N}$

end for

The state estimate is computed as the weighted mean of the particles:

$$\hat{x}_k = \sum_{i=1}^N w_k^i x_k^i \,.$$

For more details on particle filters, refer to (Doucet *et al.*, 2000; Arulampalam *et al.*, 2002).

Particle filters have an important advantage over Kalman filters: they can handle not only highly nonlinear processes, but also arbitrary distributions. In such case, however, the mean of the posterior cannot be considered a correct estimate of the state, since no guaranties exist that the posterior is unimodal. Also, due to the approximation of the posterior with weighted samples, a large number of samples are necessary for good performance. Hence, the algorithm is computationally involved, and not suitable for fast processes.

While in theory particle filters are the best of the above estimators, for non-linear systems, they cannot be considered a universal solution. Mild nonlinearities or fast processes can be better handled by a Kalman filter, even if the estimate is not perfect.

3 Centralized and Cascaded Observers

Consider a general, observable nonlinear system, described as:

$$\begin{aligned} \dot{x}_1 &= f_1(\boldsymbol{x}, \boldsymbol{u}) & y_1 &= h_1(\boldsymbol{x}, \boldsymbol{u}) \\ \dot{x}_2 &= f_2(\boldsymbol{x}, \boldsymbol{u}) & &\vdots \\ &\vdots & y_m &= h_m(\boldsymbol{x}, \boldsymbol{u}) \\ \dot{x}_n &= f_n(\boldsymbol{x}, \boldsymbol{u}) & & \end{aligned} \tag{19}$$

for which an observer has to be designed. For a large number of states, and nonlinear equations, the design of an observer is clearly problematic. If the states and/or measurements are also corrupted by noise and one uses a particle filter, then a very large sample set is needed, in which case the computational costs may render the observer unusable for online estimation.

A solution is to decompose such a system, and design separate observers for each

subsystem, allowing that one subsystem may use the results of another one. Such a decomposition is presented in Figure 1. In this setting we denote as "centralized observer" an observer designed for the system (19) as a whole, while distributed / cascaded observers are designed for the subsystems.

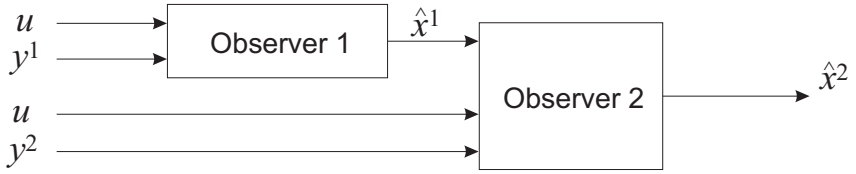


Fig. 1. Cascaded observers.

First one has to determine under which conditions such a decomposition is possible. A condition for the decomposition of the observers is that system (19) can be partitioned as:

$$\begin{aligned} \dot{\boldsymbol{x}}^1 &= \boldsymbol{f}^1(\boldsymbol{x}^1, \boldsymbol{u}) \\ \dot{\boldsymbol{x}}^2 &= \boldsymbol{f}^2(\boldsymbol{x}^1, \boldsymbol{x}^2, \boldsymbol{u}) \\ \\ \boldsymbol{y}^1 &= \boldsymbol{h}^1(\boldsymbol{x}^1, \boldsymbol{u}) \\ \boldsymbol{y}^2 &= \boldsymbol{h}^2(\boldsymbol{x}^1, \boldsymbol{x}^2, \boldsymbol{u}) \end{aligned} \tag{20}$$

so that the subsystem

$$\begin{aligned} \dot{\boldsymbol{x}}^1 &= \boldsymbol{f}^1(\boldsymbol{x}^1, \boldsymbol{u}) \\ \boldsymbol{y}^1 &= \boldsymbol{h}^1(\boldsymbol{x}^1, \boldsymbol{u}) \end{aligned} \tag{21}$$

is observable.

Since both systems (19) and (21) are observable, this also means that the subsystem

$$\begin{aligned} \dot{\boldsymbol{x}}^2 &= \boldsymbol{f}^2(\boldsymbol{x}^1, \boldsymbol{x}^2, \boldsymbol{u}) \\ \boldsymbol{y}^2 &= \boldsymbol{h}^2(\boldsymbol{x}^1, \boldsymbol{x}^2, \boldsymbol{u}) \end{aligned} \tag{22}$$

is observable. In fact, for the subsystem (22), $\boldsymbol{x}^1$ can be considered as input.

Assuming that $m \geq 2$ and such a partition exists, observers may be designed for the two parts separately. At this point, a comparison of the performance of cascaded observers and a centralized observer for the system (19) does not exist in the litera-

ture. However, several advantages of such a decomposition and the use of cascaded observers can already be recognized: the design and tuning of the subobservers is simpler, computational costs can be reduced, different observers can be used for the subsystems.

4 Estimation of Overflow Losses in a Hopper-Dredger

The Particle filter and the Unscented Kalman filter are applied in the cascaded setting (20) to the estimation of the overflow losses in a hopper dredger. Information on the amount of overflow losses is essential both for decision support and automatic control. Unfortunately, these losses cannot be reliably measured, due to the presence of air in the overflow pipe. However, as shown in this paper, they can be estimated by using mathematical models and the available on-line measurements.

Before stating the estimation problem, the principle of the dredging process is briefly explained. The dredger uses a drag head to excavate soil from the sea bottom. A mixture of soil and water is transported through a pipe to the hopper, which is a large storage tank inside the ship (see Figure 2).

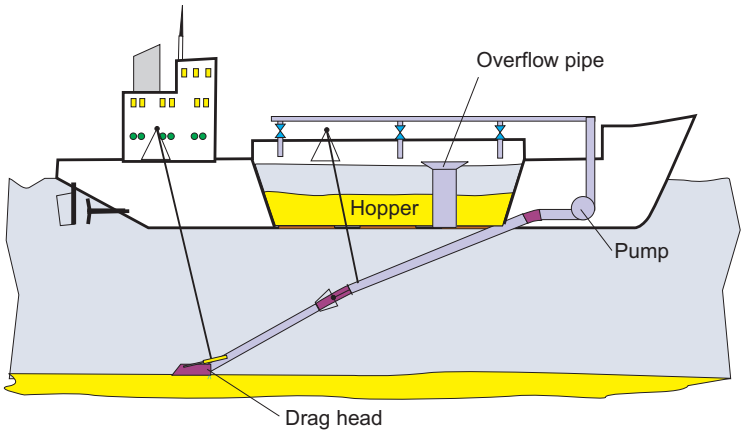


Fig. 2. Schematic drawing of a hopper dredger.

The soil gradually settles at the bottom of the hopper, while excessive water (in fact low-density mixture) is discharged through an overflow pipe whose level can be

adjusted. As the height of the settled sand layer rises, so does the concentration of the overflow mixture and eventually the overflow losses become so high that it is no longer economical to continue dredging. The ship then sails back to deliver the load. After the sand is discharged, the ship sails again to the dredging location and the whole cycle repeats.

The efficiency of the sedimentation process heavily depends on the type of soil and is influenced by the flow-rate and density of the incoming mixture and the manner the overflow pipe is controlled. An important factor in the optimization of the dredging performance is the minimization of the overflow losses. In the literature, a number of sedimentation models have been proposed (Camp, 1946; van Rhee, 2002), however, these models cannot be used as a basis for control or optimization of the dredging process. The reason is that they are based on detailed (often PDE) modeling of the physical phenomena and contain too many uncertain parameters. Therefore, we use simplified models, along with advanced signal processing and estimation techniques.

4.1 Dynamic Sedimentation Model

The sedimentation process in a hopper-dredger can be described by a model with three state variables: the total mass in the hopper m_t, the total volume V_t of the mixture in the hopper and the mass of the sand bed m_s (see Figure 3).

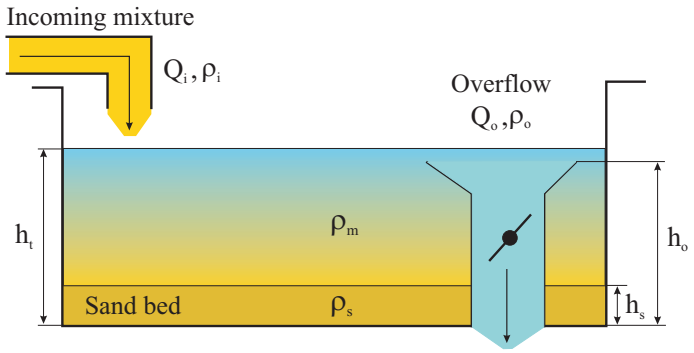


Fig. 3. The sedimentation process in the hopper.

While the first two variables can be derived from on-line measurements (the ship draught and the total level h_t, respectively), the mass of the sand bed is not measurable. The flow-rate Q_i of the incoming mixture and the overflow height h_o are the manipulated inputs and the incoming mixture density ρ_i is in this context regarded as a measured disturbance. The volume and mass balance equations are given by:

$$\dot{V}_t = Q_i - Q_o \tag{23}$$

$$\dot{m}_t = Q_i \rho_i - Q_o \rho_o \,. \tag{24}$$

The rate of material sedimentation is as a function the settling velocity (f_s) and the scouring (erosion) effects (f_e)

$$\dot{m}_s = f_s\left(\rho_m\right) f_e\left(Q_o, h_m\right) \,. \tag{25}$$

The overflow rate Q_o, the density ρ_o and the functions in (25) are modeled by using static relationships as detailed below.

If the outgoing mixture freely flows through the overflow pipe, the flow-rate Q_o is given by

$$Q_o = k_o \max(h_t - h_o, 0)^{\frac{3}{2}} \tag{26}$$

where k_o is an uncertain parameter depending on the overflow pipe shape and circumference. However, if the overflow pipe is full (e.g., because a valve inside the pipe is engaged), the following model must be used:

$$Q_o = k_o' \sqrt{2g \max(h_t - h_o, 0)} \,. \tag{27}$$

Clearly, there is some uncertainty in the modeling of the overflow rate. Moreover, due to the model's switching nature, it is not straightforward to estimate its parameters.

The density profile in the mixture above the sand bed can be approximated as a decreasing function of the height above the sand. The exact form of this function is highly uncertain and time varying. In this paper, we use the following saturated

affine linear approximation:

$$\rho_o = \max\Big(\rho_s - k_\rho(h_o - h_s), \rho_w\Big)\,. \tag{28}$$

The slope k_ρ must be determined at every time instant such that the average mixture density ρ_m, computed from the mass-balance relations, equals to the average of the density profile:

$$\rho_m = \frac{1}{h_m}\int_{h_s}^{h_t} \max\Big(\rho_s - k_\rho(h - h_s), \rho_w\Big)\mathrm{d}h$$

with $h_m = h_t - h_s$. Solving this constraint for the model (28) yields the following equation for the slope:

$$k_\rho = \begin{cases} \frac{2(\rho_s - \rho_m)}{h_m} & \text{for} \quad \rho_m > \frac{1}{2}(\rho_w + \rho_s) \\ \\ \frac{(\rho_s - \rho_w)^2}{2h_m(\rho_m - \rho_w)} & \text{otherwise} \end{cases}$$

where the average mixture density is given by:

$$\rho_m = \frac{m_t - m_s}{V_t - \frac{m_s}{\rho_s}} = \frac{\rho_s(m_t - m_s)}{V_t\rho_s - m_s}\,.$$

Validation based on measured data has shown that this model is not very accurate, but it suffices for the tuning and first evaluation of the particle filter.

The settling function f_s describes how the rate of sedimentation depends on the undisturbed settling velocity v_s and the mixture density:

$$f_s(\rho_m) = A\rho_s v_s \frac{\rho_m - \rho_w}{\rho_s - \rho_m}\left(\frac{\rho_q - \rho_m}{\rho_q - \rho_w}\right)^\beta\,. \tag{29}$$

The scouring function describes the effect of erosion on the sand bed due to the flow in the mixture (which is considered to be equal to the overflow rate in steady state):

$$f_e\left(Q_o, h_m\right) = \max\left(1 - \frac{{Q_o}^2}{k_c {h_m}^2}, 0\right)\,. \tag{30}$$

The parameters of the entire model have been determined by fitting the outputs of the simulation model to real data from a ship, by using non-linear least-squares optimization.

In order to estimate at each time step the overflow density and flow-rate, the volume and mass balance equations were discretized by using the Euler method:

$$V_{t,k} = V_{t,k-1} + T(Q_{i,k-1} - Q_{o,k-1}) \tag{31}$$

$$m_{t,k} = m_{t,k-1} + T(Q_{i,k-1}\rho_{i,k-1} - Q_{o,k-1}\rho_{o,k-1}) \tag{32}$$

where the sampling period is $T = 5\,\text{s}$, which is also the sampling period of the on-board data-acquisition system. The state equations are augmented with a random-walk model for Q_o and ρ_o:

$$Q_{o,k} = Q_{o,k-1} + \epsilon_{Q,k-1} \tag{33}$$

$$\rho_{o,k} = \rho_{o,k-1} + \epsilon_{\rho,k-1} \tag{34}$$

The motivation for this choice results from the process description in Section 4. The sedimentation models are based on empirical modeling of the physical phenomena and contain too many uncertain parameters. By using a random walk model, the use of the uncertain overflow model (26)-(27), the settling and scouring functions (29)-(30), and the uncertain parameters is circumvented. We define the augmented state, input and output vectors as:

$$x = \begin{pmatrix} V_t \\ m_t \\ Q_o \\ \rho_o \end{pmatrix}, \qquad u = \begin{pmatrix} Q_i \\ \rho_i \end{pmatrix}, \qquad y = \begin{pmatrix} V_t \\ m_t \end{pmatrix}$$

Measurements are available for the inputs Q_i and ρ_i and the outputs V_t and m_t. The objective is to estimate Q_o and ρ_o on-line. For estimation purposes, the corrupting noises are considered zero-mean Gaussians ($\epsilon_{xi,k} \sim \mathcal{N}(0, \nu_{xi})$), and their standard deviations are determined experimentally.

Note, that in this particular case, the estimation model is described as:

$$x_k = f(x_{k-1}) + v_{k-1}$$
$$y_k = h(x_{k-1)} + \eta_k$$

where v_k and η_k are zero mean Gaussian noises, with covariances Q and R, respectively. The equivalent probabilistic model is expressed as:

$$p(x_k|x_{k-1}) = \mathcal{N}(x_k; f(x_{k-1}); Q)$$
$$p(y_k|x_k) = \mathcal{N}(y_k; h(x_k); R)$$

Using the estimation model above, a centralized PF has already been successfully implemented in the data-acquisition and monitoring system of a hopper-dredger. The different observers, using the presented estimation model, are compared in terms of their performance. We consider the following cases:

(1) use of a centralized observer to simultaneously estimate the values of both Q_o and ρ_o, see Figure 4.

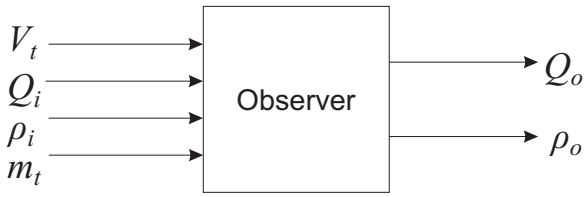


Fig. 4. Centralized observer.

(2) cascaded observers: the first observer estimates Q_o based on the volume balance (31), and the second estimates ρ_o based on the mass balance (32) and the values obtained for Q_o by the first observer, see Figure 5.

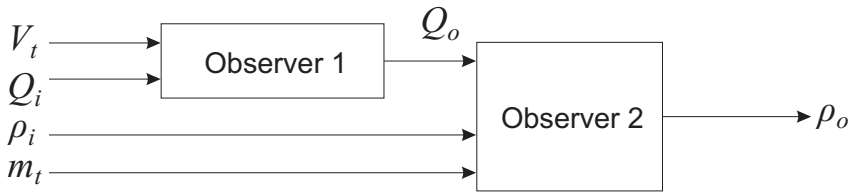


Fig. 5. Cascaded observers.

Note, that the models (31)-(34) only approximate the underlying true process. If the data were generated based on these models, a Kalman filter could be used. For the data generated from the sedimentation model, however, the results obtained by the Kalman filter are too noisy and the Kalman filter becomes unstable for the experimental data.

We found that the UKF cannot simultaneously estimate both Q_o and ρ_o. Therefore, for the centralized observer, only PF is considered, while in the cascaded setting both PF and UKF are used. The observers are first applied to the simulated data and then, with the same parameters, to real measured data.

There is one more setting of observers that was considered for this specific application: independent observers. In this setting, one uses two observers: the first estimates Q_o based on the volume balance (31). The second estimates $Q_o\rho_o$ based on the mass balance (32). The value of ρ_o can be computed afterwards by dividing the estimate of $Q_o\rho_o$ by the estimate of Q_o obtained by the first observer. However, when working with experimental data, the computation of ρ_o means dividing noisy variables and leads to huge errors and therefore the results are not presented here.

5 Results for Simulated Data

Recall that the model used for simulation is the one presented in Section 4.1, while we use for estimation:

$$\begin{aligned} V_{t,k} &= V_{t,k-1} + T(Q_{i,k-1} - Q_{o,k-1}) + \epsilon_{V,k-1} \\ m_{t,k} &= m_{t,k-1} + T(Q_{i,k-1}\rho_{i,k-1} - Q_{o,k-1}\rho_{o,k-1}) + \epsilon_{m,k-1} \\ Q_{o,k} &= Q_{o,k-1} + \epsilon_{Q,k-1} \\ \rho_{o,k} &= \rho_{o,k-1} + \epsilon_{\rho,k-1} \end{aligned} \tag{35}$$

with $\epsilon_{V,k} \sim \mathcal{N}(0, \nu_V)$, $\epsilon_{m,k} \sim \mathcal{N}(0, \nu_m)$, $\epsilon_{Q,k} \sim \mathcal{N}(0, \nu_Q)$, $\epsilon_{\rho,k} \sim \mathcal{N}(0, \nu_\rho)$ and $\mathcal{N}(0, \nu)$ being a zero-mean, ν^2 covariance Gaussian random noise.

For this simulation, only the inputs Q_i and ρ_i are fed with experimental data, corrupted by noise. The remaining variables are computed in simulation without adding noise.

The results obtained with the different configuration of observers are compared to the simulated values of ρ_o and Q_o. The standard deviations of the state transition and measurement noises are given in Table 1. The particle filter used 1000 samples,

Table 1
Standard deviations used in the estimation model (35).

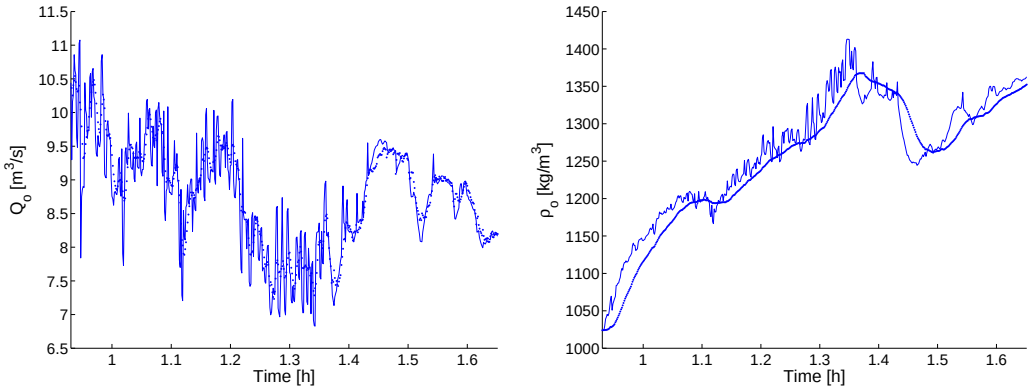
Variable	State transition	Measurement
V_t [m^3]	0	10
m_t [kg]	3000	12000
Q_o [m^3/s]	0.25	–
ρ_o [kg/m^3]	5	–

with resampling at a threshold of $N_T = 900$. The presented results are the average of 30 simulations.

5.1 Centralized Observer

The results obtained with a particle filter, based on the model (35) are presented in Figure 6. The maximum standard deviation computed point-wise for the 30 Monte Carlo simulations, is 0.1 for Q_o and 5.1 for ρ_o.

The residuals are computed as the difference between the simulated and estimated values of Q_o and ρ_o, respectively. The distribution of the residuals is presented in Figure 7, while their statistics are given in Table 2.



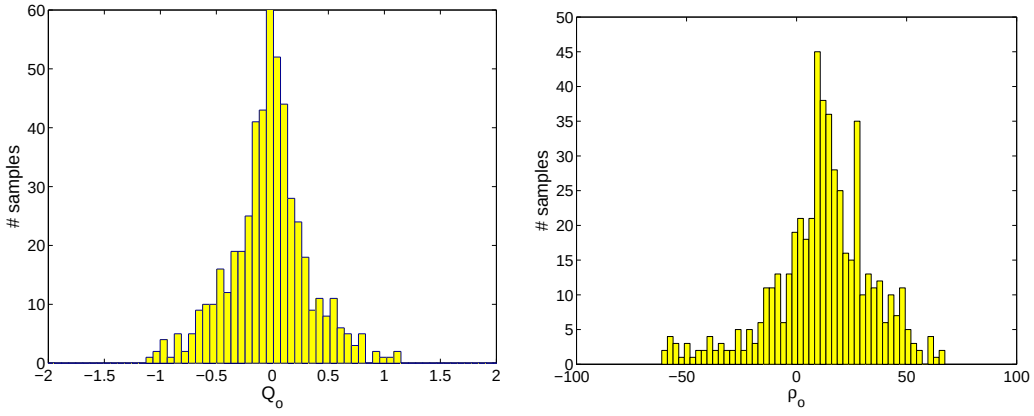
(a) (b)

Fig. 6. Centralized observer: results for Q_o (a) and ρ_o (b) using the particle filter (solid line – simulated data, dotted line – estimate).

Table 2

Statistics of residuals for the centralized observer.

	Mean	Standard deviation
Q_o [m^3/s]	-0.0135	0.5863
ρ_o [kg/m^3]	11.6858	22.6426



(a) (b)

Fig. 7. Centralized observer: residuals for Q_o (a) and ρ_o (b).

5.2 Cascaded Observers

This setting involves two observers in a cascade. The first one estimates Q_o using the volume balance (31) and a random walk model for Q_o. The second observer

estimates ρ_o based on the mass balance (32), a random walk model for ρ_o and the result obtained for Q_o by the first observer (see also Figure 5). Two types of filters are compared: the Unscented Kalman Filter and the Particle Filter.

The dynamic model is decomposed into two subsystems. The first observer uses the model

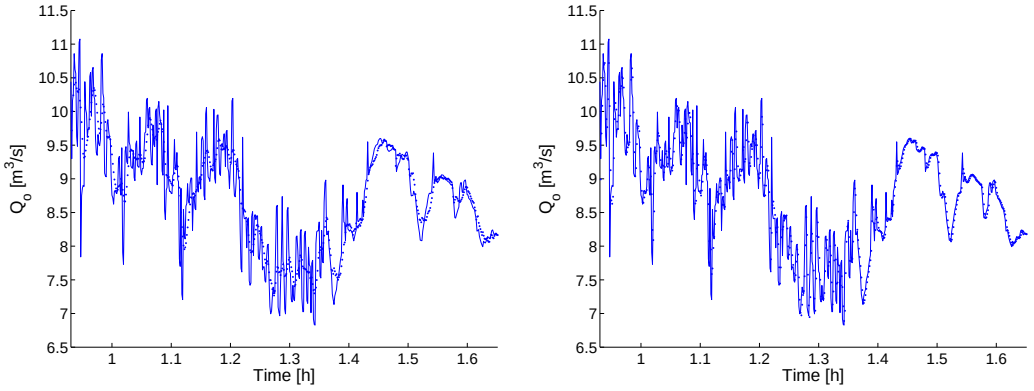
$$\begin{aligned} V_{t,k} &= V_{t,k-1} + T(Q_{i,k-1} - Q_{o,k-1}) + \epsilon_{V,k-1} \\ Q_{o,k} &= Q_{o,k-1} + \epsilon_{Q,k-1} \end{aligned} \tag{36}$$

where V_t is the measured output. The second observer uses the model

$$\begin{aligned} m_{t,k} &= m_{t,k-1} + T(Q_{i,k-1}\rho_{i,k-1} - Q_{o,k-1}\rho_{o,k-1}) + \epsilon_{m,k-1} \\ \rho_{o,k} &= \rho_{o,k-1} + \epsilon_{\rho,k-1} \end{aligned} \tag{37}$$

where m_t is the measured output.

The results obtained for Q_o are presented in Figure 8.



(a) (b)

Fig. 8. Cascaded observers: results for Q_o using PF (a) and UKF (b) (solid line – simulated data, dotted line – estimate).

The maximum standard deviation computed point-wise for the 30 Monte Carlo simulations, is 0.095. The statistics of the distributions of the residuals are given in Table 3. These statistics are comparable with that obtained with the centralized observer.

For the cascaded observer the following combinations are considered: UKF for both Q_o and ρ_o, PF for both Q_o and ρ_o, UKF for Q_o and PF for ρ_o, and PF for Q_o

Table 3

Statistics of residuals of Q_o [m^3/s].

	Mean	Standard deviation
PF	0.0094	0.5953
UKF	0.0222	0.5582

and UKF for ρ_o. In what follows, the observer setting is denoted as observer 1 - observer 2, i.e. PF-PF denotes that PF is used for both Q_o and ρ_o, UKF-PF means that UKF is used for Q_o and PF for ρ_o, etc.

The results obtained for ρ_o using Q_o estimated by PF and UKF are presented in Figure 9. The maximum point-wise standard deviation of the 30 Monte Carlo simulations for ρ_o, based on the results of Q_o given also by PF is 5.8, while for ρ_o based on based on the results of Q_o given by UKF is 2.96. The distribution of the residuals is given in Figure 10. The statistics of the residuals can be found in Table 4.

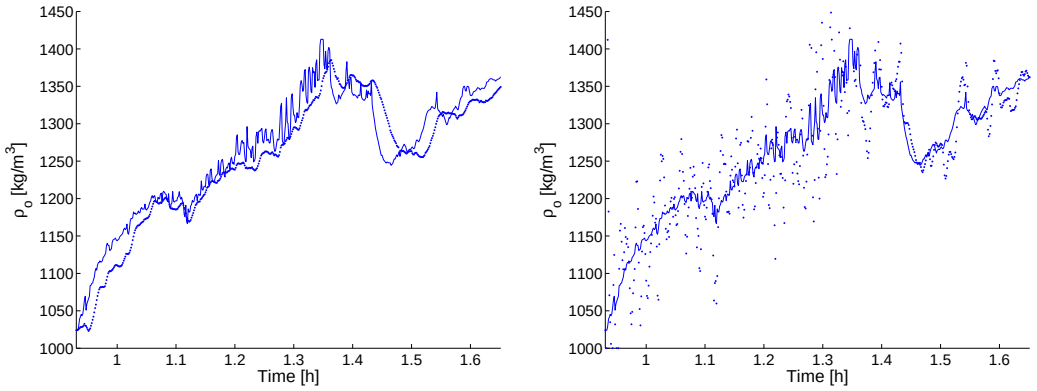
Table 4

Statistics of residuals of ρ_o [kg/m^3].

Q_o	ρ_o	Mean	Standard deviation
PF	PF	12.1986	25.1174
PF	UKF	0.3354	54.3970
UKF	PF	10.6790	21.5436
UKF	UKF	-1.2536	54.2936

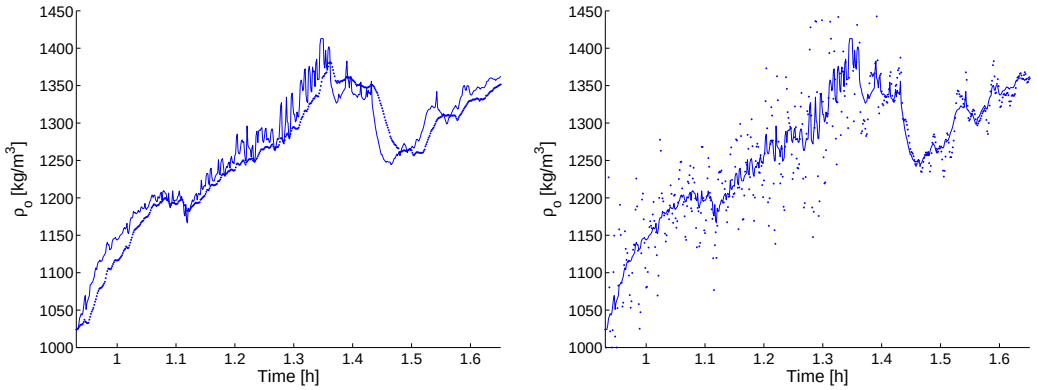
5.3 Discussion

For simulated data, using a centralized observer (particle filter), a good estimate is obtained for Q_o, but the estimate of ρ_o is delayed relative to the one simulated



(a) PF – PF.

(b) PF – UKF.



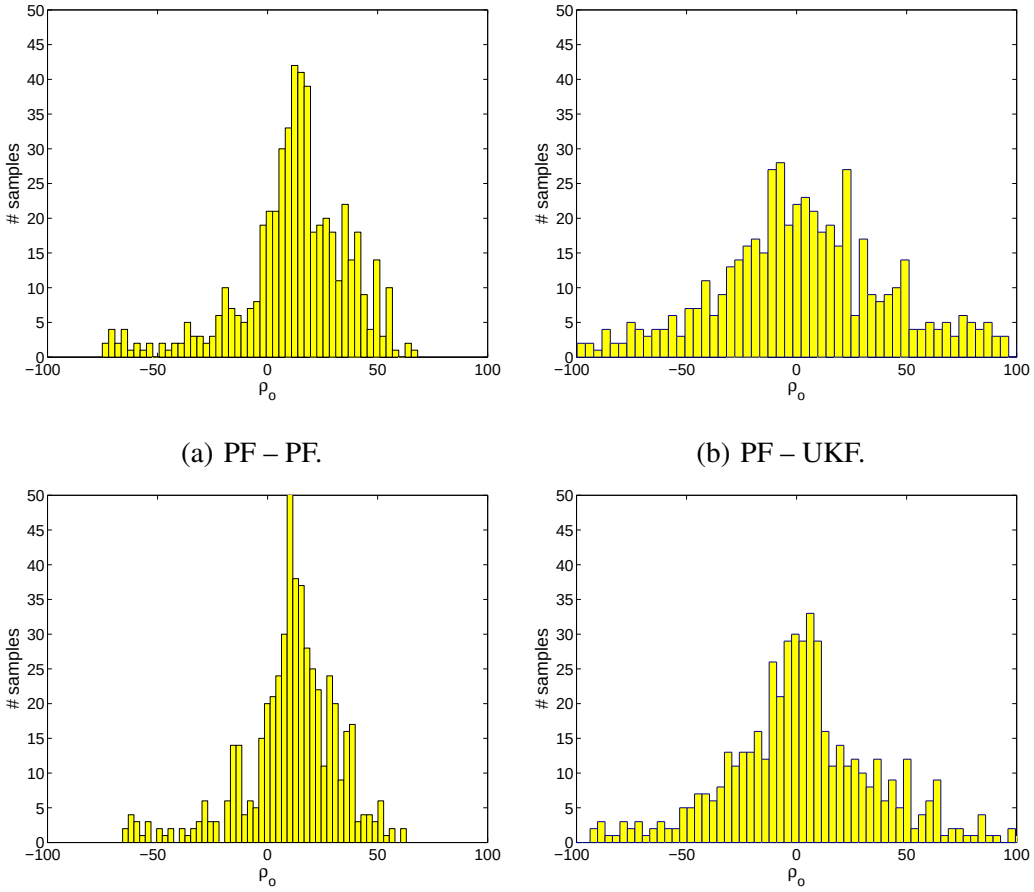
(c) UKF –PF.

(d) UKF – UKF.

Fig. 9. The four possible estimates of ρ_o (solid line – simulated data, dotted line – estimate).

(Figure 6).

Of the cascaded observers, considering only the mean of the residuals (Table 4), one can conclude that the UKF performs the best for the estimation of ρ_o, much better than the centralized observer or the PF. However, both Figure 10 and the standard deviations (Table 4) indicate differently: the estimates of ρ_o obtained by UKF are much noisier than those obtained by the PF, both centralized and cascaded. Comparing the results obtained by PF for ρ_o with the centralized observer (Table 2), it can be seen that using a combination of two particle filters leads to approximately the same results as a centralized one. However, the best result, based on the statistics presented in Tables 3 and 4 was obtained with cascaded observers, the combination UKF for Q_o and PF for ρ_o.



(a) PF – PF. (b) PF – UKF.

(c) UKF – PF. (d) UKF – UKF.

Fig. 10. Cascaded observers: residuals for the estimates of ρ_o.

6 Results for Experimental Data

For the experimental data, we consider the same model and parameter values as for the simulated data and the same combinations of observers. The presented results for the particle filter are the average of 30 simulations.

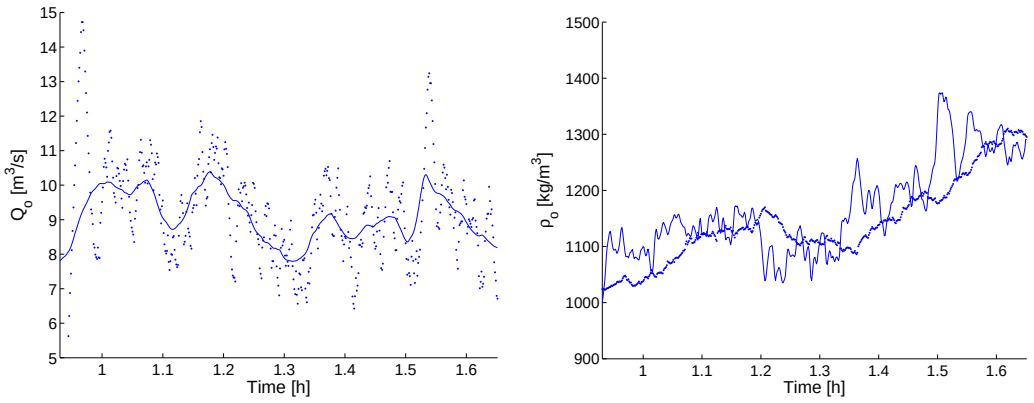
Since no measurements of Q_o and ρ_o are available, the results are compared to the values of ρ_o and Q_o directly computed from the volume and mass balance, i.e., equations (31)-(32). The overflow rate and the density are computed as:

$$\begin{aligned} Q_{o,k} &= Q_{i,k} - \frac{1}{T}(V_{t,k+1} - V_{t,k}) \\ \rho_{o,k} &= \frac{Q_{i,k}\rho_{i,k} - \frac{1}{T}(m_{t,k+1} - m_{t,k})}{Q_{o,k}}. \end{aligned} \tag{38}$$

As the result of this computation is very noisy, a first order anti-causal low-pass filter was applied to the measured data before computing Q_o and ρ_o, with an experimentally chosen cut-off frequency of 0.05 Hz.

6.1 Centralized Observer

The results obtained with the centralized observer (particle filter) are presented in Figure 11. The maximum standard deviation computed point-wise over the 30 Monte Carlo simulations is 1.76 for Q_o and 42.87 for ρ_o.



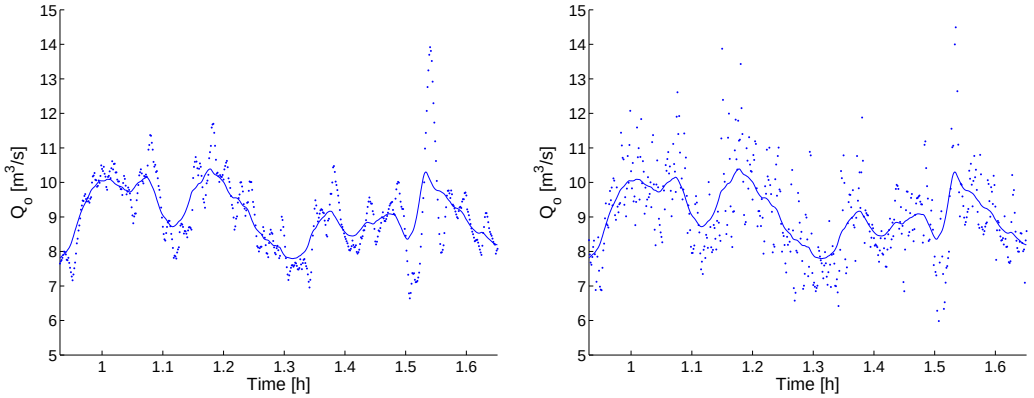
(a) (b)

Fig. 11. Centralized observer: results for Q_o (a) and ρ_o (b) using particle filter (solid line – computed by (38), dotted line – estimate).

6.2 Cascaded Observers

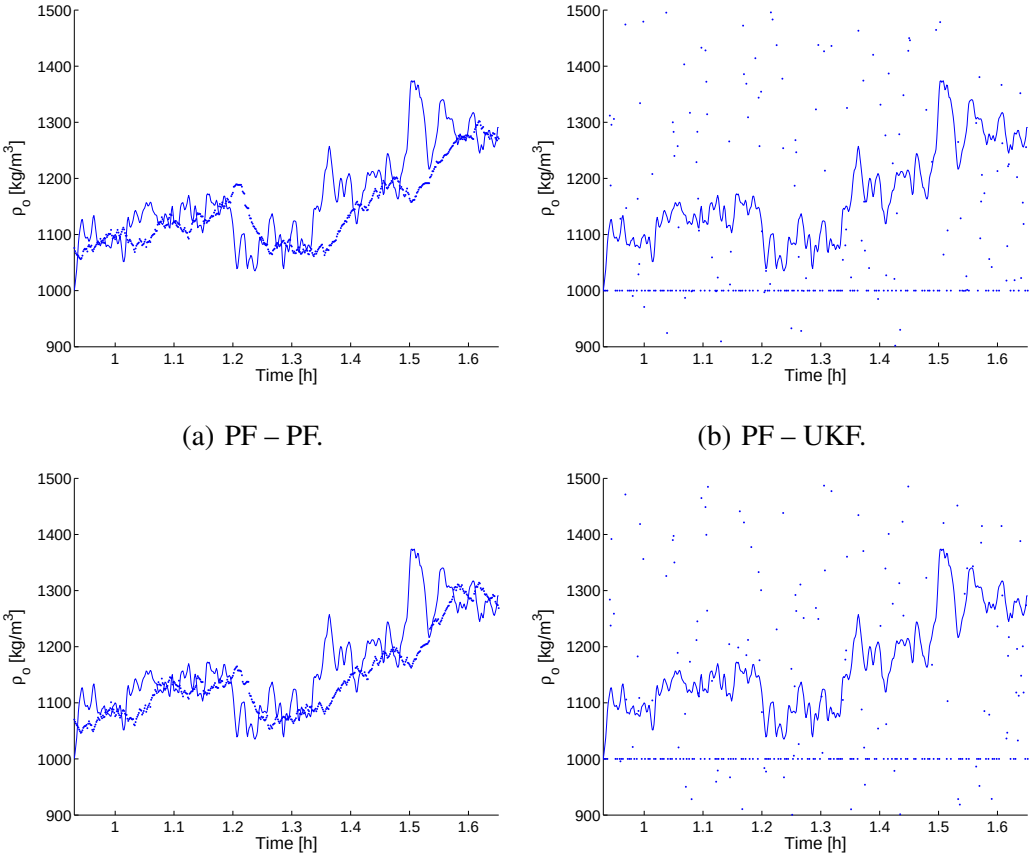
The results obtained by the UKF and PF for Q_o are presented in Figure 12. The estimate of Q_o obtained by the UKF is noisier than that obtained by the PF, but comparable to the result obtained by the centralized observer. The maximum sample-wise standard deviation of the Monte Carlo simulations for Q_o is 0.55.

For the cascaded setting, the previous four combinations are considered. The results obtained for ρ_o with the four combinations are presented in Figure 13.



(a) (b)

Fig. 12. Cascaded observers: results for Q_o using PF (a) and UKF (b) (solid line – computed by (38), dotted line – estimate).



(a) PF – PF. (b) PF – UKF.

(c) UKF – PF. (d) UKF – UKF.

Fig. 13. Cascaded observers: estimates of ρ_o using the four possible combinations (solid line – computed by (38), dotted line – estimate).

The sample-wise maximum standard deviation of the 30 Monte Carlo simulations for ρ_o, based on the results of Q_o given also by PF is 36.91, while for ρ_o based on the results of Q_o given by UKF is 36.05.

6.3 Discussion

For experimental data, a centralized observer (Figure 11) obtains a reasonably good estimate for ρ_o, but the estimate of Q_o is noisier than the one computed by (38).

From Figure 13 it can be seen that the UKF cannot handle the estimation of ρ_o, probably due to the high noise level of the measured variables. By using the combinations of PF–PF and UKF–PF, approximately the same results are obtained for ρ_o. These results are also comparable to those obtained by the centralized observer.

7 Conclusions

In this paper, we proposed a distributed setting for nonlinear state-estimation. In many real-life applications, a complex process model can be decomposed in simpler subsystems, and observers can be designed for these individual subsystems. This partitioning of a process and observer leads to increased modularity and reduced complexity of the problem, with reduced computational costs. Moreover, since the design and tuning of the observers becomes simpler, the overall performance can also be improved.

The distributed observer setting was applied to the estimation of overflow losses in a hopper dredger. The results were compared with those obtained by the centralized observer. The overflow losses, represented by the overflow mixture density are estimated on the basis of the measured total hopper volume, hopper mass, incoming mixture density and flow-rate.

The proposed approach uses straightforward nonlinear mass balance equations and does not rely on the overflow and sedimentation models which contain many uncertain parameters and empirical functional relationships.

The performance was evaluated in simulations and with real measurements. The simulation results for this application clearly indicate the best combination: cascaded observers, using UKF or PF for the simpler subsystem (flow-rate) and PF for the more complex one (density).

In our future research, we will investigate the theoretical conditions under which such a distribution of the process and the estimation is possible while maintaining the same performance as a centralized observer.

Acknowledgements: This research is in part sponsored by Senter, Ministry of Economic Affairs of the Netherlands within the projects Artificial Intelligence for the Control of a Hopper Dredger (grant no. TSMA 2017) and Interactive Collaborative Information Systems (grant no. BSIK03024). The authors also thank the dredging company DEME for supplying the data.

References

Arulampalam, S., Maskell, S., Gordon, N. J., and Clapp, T. (2002). A tutorial on particle filters for on-line nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50:174–188.

Bolic, M., Djuric, P. M., and Hong, S. (2004). Resampling algorithms and architectures for distributed particle filters. *IEEE Tranzactions on Signal Processing*, 53(7):2442– 2450.

Camp, T. (1946). Sedimentation and the design of settling tanks. *Trans. ASCE*, pages 895–936.

Coates, M. (2004). Distributed particle filters for sensor networks. In *IPSN'04*, pages 99–107.

Doucet, A., Godsill, S., and Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10:197–208.

Durrant-Whyte, H., Rao, B., and Hu, H. (1990). Toward a fully decentralized architecture for multi-sensor data fusion. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1331 – 1336.

Hue, C., Cadre, J. L., and Perez, P. (2002). Tracking multiple objects with particle filtering. *IEEE Transactions on Aerospace and Electronic Systems*, 38(3):791–812.

Julier, S. J. and Uhlmann, J. K. (1997). A new extension of the Kalman filter to nonlinear systems. In *Proceedings of 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, pages 182–193.

Julier, S. J. and Uhlmann, J. K.(2002). Reduced sigma point filters for the propagation of means and covariances through nonlinear transformations. In *Proceedings of the 2002 American Control Conference*, volume 2, pages 887– 892.

Julier, S. J. and Uhlmann, J. K. (2002). The Scaled Unscented Transformation. In *Proceedings of the IEEE American Control Conference*, pages 4555–4559, Anchorage AK, USA.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82:35–45.

Li, P., Zhang, T., and Ma, B. (2004). Unscented Kalman filter for visual curve tracking. *Image and Vision Computing*, 22:157–164.

López-Orozco, J. A., de la Cruz, J. M., Besada, E., and Ruiprez, P. (2000). An asynchronous, robust, and distributed multisensor fusion system for mobile robots. *Intl. Journal of Robotics Research*, 19(12):914–932.

Nait-Charif, H. and McKenna, S. J. (2004). Tracking poorly modelled motion using particle filters with iterated likelihood weighting. In *The Asian Conference on Computer Vision Systems (ACCV04)*, pages 156–161.

Roumeliotis, S. and Bekey, G. (2002). Distributed multirobot localization. *IEEE Transactions on Robotics and Automation*, 18:781 – 795.

Schmitt, T., Hanek, R., Beetz, M., Buck, S., and Radig, B. (2002). Cooperative

probabilistic state estimation for vision-based autonomous mobile robots. *IEEE Transactions on Robotics and Automation*, 18(5):670–684.

Stenger, B., Mendonça, P. R. S., and Cipolla, R. (2001). Model-based hand tracking using an unscented Kalman filter. In *Proc. British Machine Vision Conference*, volume I, pages 63–72, Manchester, UK.

Sullivan, J., Blake, A., Isard, M., and MacCormick, J. (2001). Bayesian object localisation in images. *International Journal of Computer Vision*, 44(2):111–135.

Vadigepalli, R. and Doyle, F. J. (2003). Structural analysis of large-scale systems for distributed state estimation and control applications. *Control Engineering Practice*, 11:895–905.

Vadigepalli, R. and Doyle, F.J., I. (2003b). A distributed state estimation and control algorithm for plantwide processes. *IEEE Transactions on Control Systems Technology*, 11:119 – 127.

van der Merwe, R. (2004). *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models*. PhD thesis, University of Washington.

van der Merwe, R. and Wan, E. (2003). Gaussian mixture sigma-point particle filters for sequential probabilistic inference in dynamic state-space models. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 701–704, Hong Kong.

van Rhee, C. (2002). *On the sedimentation process in a suction hopper dredger*. PhD thesis, TU Delft.

Welch, G. and Bishop, G. (2002). An introduction to the Kalman filter. Technical Report TR 95-041, Department of Computer Science, University of North Carolina, NC, USA.