Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Bayesian Fill Volume Estimation Based on Point Level Sensor Signals ⋆

Johannes Zumsande * **Karl-Philipp Kortmann** *
Mark Wielitzka * **Tobias Ortmaier** *

* *Institute of Mechatronic Systems, Leibniz University Hannover, Germany (e-mail: johannes.zumsande@imes.uni-hannover.de)*

Abstract: In dry bulk and fluid processing, the composites are usually stored in hoppers, tanks, or other containers. Due to the economic advantages, binary point level sensors, which detect fill level exceeding, are widely used for process monitoring and control. In this paper, we propose different filters for estimating the probability distribution of the fill volume based on a time-variant measurement distribution and a stochastic physical model with white process noise. A filter based on the model prediction with separated measurement update and two Bayesian particle filters are proposed and compared with a simulated ground truth. The performance measures are the root-mean-square error, the precision of the 95 % and 75 % credible intervals, and the average value of the estimated probability density function at the simulated fill volumes.

Keywords: Bayesian filter, data fusion, probabilistic models, stochastic approximation, estimation algorithms

1. INTRODUCTION

The ongoing digitalization in modern production systems leads to transparent and holistic information stored in data warehouses, see Qi and Tao (2018). Methods of data mining, modeling, and control use this comprehensive information pool to offer extracted more relevant information to the user, see Maurer et al. (2017) and Hameed et al. (2010), or enhance system's performance by advanced prediction models and control algorithms, see O'Donovan et al. (2015) and Tao et al. (2018). Modern storage management systems have a huge impact on this transformation providing high-valuable, precise information of location and storage time of each stored object. However, considering (semi-) continuous work pieces, like bulk materials or fluids, the state of the art is less precise. Continua are mainly stored in containers, tanks, or hoppers, which are, for economic reasons, often equipped with binary point level sensors and controlled by a hysteresis control (bang-bang control). However, a continuous fill level information leads to multi-dimensional benefits, like precise process monitoring and control, storage time determination, consideration of time dependent property changes, or preventive reordering of composites. Though continuous fill level sensors (e.g. radar or capacitive, see Chetpattananondh et al. (2014)) are available, point level sensors are widely used due to their robustness, their lower acquisition costs and the long depreciation periods of the plants.

To make the advantages of a continuous fill level information available to existing plants (brownfield plants) equipped with binary point level sensors, this paper presents Bayesian filters for measurement and model based fill volume estimation. A similar problem is the object tracking in binary sensor networks, where each sensor detects if an object is located below a certain distance, see Teng et al. (2010). For continuous sensor signals a fill level and parameter estimation approach can be found in Itävuo et al. (2017) and Babuska et al. (2006). A Bayesian filter, see Särkkä (2013), predicts the state x_k based on the probability density function of a stochastic physical model of first order $p\left(x_k|x_{k-1}\right)$ and updates the prediction considering the measurement likelihood $p\left(\boldsymbol{y}_k|x_k\right)$ of the sensor signals $\boldsymbol{y}_k$. Given the filtered probability density function $p\left(x_{k-1}|\boldsymbol{y}_{1:k-1}\right)$ at time $k-1$, with $\boldsymbol{y}_{1:k-1} = (\boldsymbol{y}_1, ..., \boldsymbol{y}_{k-1})$, the probability density function of the state x_k at time k

$$p\left(x_k|\boldsymbol{y}_{1:k-1}\right) = \int p\left(x_k|x_{k-1}\right) p\left(x_{k-1}|\boldsymbol{y}_{1:k-1}\right) \mathrm{d}x_{k-1} \quad (1)$$

can be predicted based on the physical model. Considering the measurement likelihood, the probability density function of state x_k can be determined based on Bayes' rule

$$p\left(x_k|\boldsymbol{y}_{1:k}\right) = \frac{1}{Z} p\left(\boldsymbol{y}_k|x_k\right) p\left(x_k|\boldsymbol{y}_{1:k-1}\right), \quad (2)$$

with (often called model evidence)

$$Z = \int p\left(\boldsymbol{y}_k|x_k\right) p\left(x_k|\boldsymbol{y}_{1:k-1}\right) \mathrm{d}x_k. \quad (3)$$

Our testbed container is one return sand hopper of the foundry Heinrich Meier Eisengießerei GmbH & Co. KG and the proposed methods are based on their existing sensor infrastructure. For data anonymization all plant related data is normalized by the total hopper volume, so that the fill volume is in $[0, 1]$.

In section 2, we will describe the sand hopper system at the foundry. The physical model as well as the parameter and noise identification will be proposed in section 3. In section 4, a model prediction filter and different particle filters

⋆ This work is supported by the European Regional Development Fund grant ZW 6-85018381. We would like to thank our industrial partners Heinrich Meier Eisengießerei GmbH & Co. KG, IAV GmbH and KÜNKEL WAGNER Germany GmbH for their support.

will be proposed and validated regarding an exemplary scenario and a simulated fill volume signal in section 5.

2. DESCRIPTION OF THE SYSTEM

In a green sand foundry, the main-composite of the casting molds is quartz sand. To save resources, the molding material returns after it was separated from the casting and passes through sand processing to restore its properties for building new molds again. Mixers homogenize return sand, additives, and water regarding a specific recipe. Because the mixing is done batch wise, hoppers are needed to store the return sand temporarily. In this publication, we will focus on one hopper after the return sand cooler. A schematic view of the sand hopper and the related plant components can be seen in Fig. 1. The outlet flap (B) angle α_k of the return sand cooler (A) and the sand scraper (D) position s_k control the input sand flow of the hopper. Two sensors detect if the fill level is below or above their mounting height. Before a new batch is mixed, the return sand is portioned by switching on the outlet belt (F) until the desired weight is reached in the scale (G). Following, the used symbols are listed for a time step $k \in K = \{0, ..., \hat{k}\}$:

α_k:	Outlet flap angle,
$d_k \in \{0,1\}$:	Activity of the outlet belt,
Δt :	Transportation time from (B) to (D),
ρ :	Density of return sand,
$s_k \in \{0,1\}$:	Sand scraper position,
t_s :	Sample time,
V_h :	Volume at mounting height of sensor $h \in H = \{0, ..., \hat{h}\}$,
$V_{\mathrm{L},k}$, $V_{\mathrm{U},k}$:	Lower and upper valid vol. interval,
w_k :	Scale signal,
x_k :	Fill volume,
$\boldsymbol{y}_k$:	Sensor signal tuple,
$y_{h,k} \in \{0,1\}$:	Signal of (virtual) point level sensor $h \in H$,
$p(\cdot)$:	Probability distribution or density,
$\mathrm{U}(a,b)$:	Uniform distribution with lower bound a and upper bound b
$\mathrm{N}(\mu,\sigma^2)$:	Normal distribution with mean μ and variance σ^2,
$\mathrm{T}(\mu,\sigma^2,a,b)$:	Truncated normal distribution with mean μ, variance σ^2, lower bound a and upper bound b,
$\lfloor\cdot\rceil$,$\lfloor\cdot\rfloor$,$\lceil\cdot\rceil$:	Round, floor and ceil function.
$\neg$:	Logical not.

As the hopper control prevents the volume in the return sand hopper exceeding the maximum capacity $V_{\hat{h}} = 1$ or falling below $V_0 = 0$, virtual sensors are added in the model with $y_{0,k} = 1, y_{\hat{h},k} = 0 \ \forall \ k$.

3. STOCHASTIC MODELING AND IDENTIFICATION

3.1 Physical Modeling

For modeling the following simplifications were made: the fill level is ideal flat and horizontal, the return sand is incompressible, only the outlet flap influences the sand volume on the conveyor belt, and the sand scraper takes off a constant proportion of the incoming sand (ideally 100 %). The input variables are the opening height of the

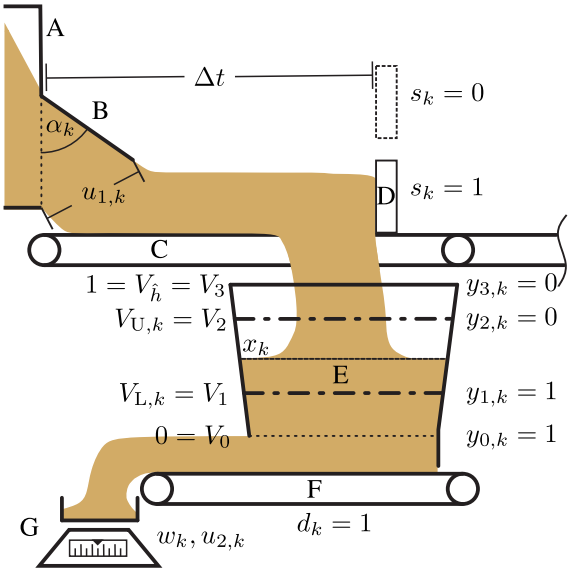

Fig. 1. The sand hopper system with return sand cooler (A), outlet flap (B), conveyor belt (C), sand scraper (D), sand hopper (E) with two point level sensors ($|H| = 3$), outlet belt (F), and weighing scale (G).

sand cooler $u_{1,k}$ (with normalized flap height of 1), which can be determined by the flap geometry, and the mass difference of the scale $u_{2,k}$ between time step $k+1$ and k

$$u_{1,k} = 2 s_k \sin\left(\frac{\alpha_{k-\lfloor\frac{\Delta t}{t_s}\rceil}}{2}\right), \tag{4}$$

$$u_{2,k} = d_{k+1}\left(w_{k+1} - w_k\right). \tag{5}$$

The physical model can be expressed as a time-discrete state space model with process noise $q_{k-1} \sim \mathrm{N}\left(0, Q_{k-1}\right)$ and the unknown parameter ϑ_μ, that represents the volume flow per normalized opening height of the flap,

$$x_k = x_{k-1} + \underbrace{\begin{pmatrix}\vartheta_\mu & -\frac{1}{\rho}\end{pmatrix}\begin{pmatrix}u_{1,k-1}\\ u_{2,k-1}\end{pmatrix}}_{\delta_{x_{k-1}}} + q_{k-1}, \tag{6}$$

$$\boldsymbol{y}_k = \left(y_{0,k}, ..., y_{\hat{h},k}\right), \text{ with } \quad y_{h,k} = \begin{cases}0 & \text{if } x_k < V_h\\ 1 & \text{else.}\end{cases} \tag{7}$$

As the weighing signal w_k is needed to determinate x_k, this representation is not real-time capable. For use in real-time scenarios, the past weighing difference between time step $k-1$ and $k-2$ has to be used. Following (6), the stochastic state space model is given by

$$\begin{aligned} x_k \sim p\left(x_k \mid x_{k-1}\right) &= \mathrm{N}\left(x_k \mid x_{k-1} + \delta_{x_{k-1}}, Q_{k-1}\right)\\ &= \mathrm{N}\left(x_k - x_{k-1} \mid \delta_{x_{k-1}}, Q_{k-1}\right). \end{aligned} \tag{8}$$

Using (8) in (1), leads to a convolution of two probability densities and we obtain the intuitive model prediction of adding the random variable $\Delta_{x_{k-1}} \sim \mathrm{N}\left(\delta_{x_{k-1}}, Q_{k-1}\right)$ to the previous fill volume estimation. Considering (7) the valid interval $[V_{\mathrm{L},k}, V_{\mathrm{U},k})$ is defined by the highest sensor, which signal is 1, and the sensor above

$$V_{\mathrm{L},k} = V_{\check{h}}, \ V_{\mathrm{U},k} = V_{\check{h}+1}, \ \check{h} = h \in H \mid y_{h,k} \neq y_{h+1,k}, \tag{9}$$

which leads to the measurement likelihood

$$p\left(\boldsymbol{y}_k \mid x_k\right) = \begin{cases}1 & x_k \in [V_{\mathrm{L},k}, V_{\mathrm{U},k})\\ 0 & \text{otherwise.}\end{cases} \tag{10}$$

3.2 Parameter and Process Noise Identification

In comparison with the model uncertainties and simplifications, a changing sensor signal is precise so that the measurement noise is approximately zero. In this paper, we will call this precise volume information a volume measurement. The bang-bang control rules for the fill volume

$$s_k = \neg y_{\hat{h}-1,k} \quad \text{and} \quad d_k = y_{1,k} \tag{11}$$

lead to often constant volume measurements. 92.1 % of the volume measurements are caused by the fill level exceeding or falling below the same sensor. Therefore, the identification problem is ill-conditioned, if both parameters ϑ_μ and ρ have to be identified. To prevent the parameters from converging towards the trivial solution with $\rho = \infty$ and $\vartheta_\mu = 0$ the return sand density ρ was measured. As shown in Schütte et al. (1997) for a five-step control, the input signal is nearly uncorrelated with the output in case of a bang-bang (two-step) control. Therefore, the parameter can be identified in the closed-loop control system. The identification samples were generated based on the volume measurements, see Fig. 2. The set of time steps k that show a volume measurement is

$$\tilde{K} = \left\{ k \in K \mid \boldsymbol{y}_k \neq \boldsymbol{y}_{k-1} \right\}, \tag{12}$$

which can be sorted in sequence (increasing k) in the tuple $\tilde{\boldsymbol{k}}$. For the corresponding volume index $h \in H$ of a volume measurement, we will define a function for later use

$$\tilde{h}\left(k\right) = h \in H \mid y_{h,k} \neq y_{h,k-1}. \tag{13}$$

To identify the unknown parameter, the modeled volume difference between two volume measurements at time steps $\tilde{k}_{m-1}$ and $\tilde{k}_m$ is compared with the difference of the volume measurements. With the proper subset

$$S_m = \left\{ \tilde{k}_{m-1}, ..., \tilde{k}_m - 1 \right\} \supset K \tag{14}$$

of time steps between the volume measurements $\tilde{k}_{m-1}$ and $\tilde{k}_m$

$$e_m = \vartheta_\mu \sum_{k \in S_m} u_{2,k} - \frac{1}{\rho} \sum_{k \in S_m} u_{1,k} - V_{\tilde{h}(\tilde{k}_m)} + V_{\tilde{h}(\tilde{k}_{m-1})} \tag{15}$$

forms the m-th identification sample with model error e_m. The variance of the white process noise of the system (6)

$$Q_k = \vartheta_\sigma \begin{pmatrix} \vartheta_\mu & \frac{1}{\rho} \end{pmatrix} \begin{pmatrix} |u_{1,k}| \\ |u_{2,k}| \end{pmatrix} \tag{16}$$

is modeled proportional to the total volume flow, the sum of input and output flow's absolute values, as it is assumed that the model uncertainty is caused by noisy excitation and is zero if $u_{1,k} = u_{2,k} = 0$. The unknown parameters $\boldsymbol{\vartheta} = (\vartheta_\mu, \vartheta_\sigma)$ can be determined by maximizing the log-likelihood $\ln\left(p\left(\boldsymbol{e} \mid \boldsymbol{\vartheta}\right)\right)$, see DeGroot and Schervish (2012). For the present assumptions, this leads to the parameter estimate

$$\boldsymbol{\vartheta} = \underset{\boldsymbol{\vartheta} \in \mathbb{R}^2_{>0}}{\arg\min} \sum_{m \in M} \ln\left(\sqrt{2\pi\sigma_m^2}\right) + \frac{e_m^2}{2\sigma_m^2} \tag{17}$$

with

$$\sigma_m^2 = \sum_{k \in S_m} Q_k. \tag{18}$$

The parameters were identified based on 30e6 samples of input and output data and $|M| = 44\text{e}3$ identification samples. The confidence intervals of ϑ_μ and ϑ_σ are $3784\text{e}{-6} \pm 7\text{e}{-6}$ and $436\text{e}{-6} \pm 12\text{e}{-6}$ respectively.

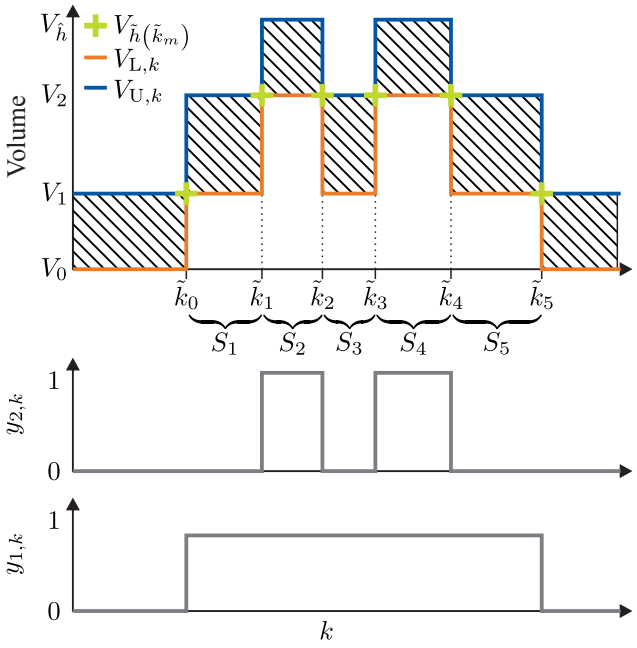

Fig. 2. The valid volume interval $[V_\mathrm{L}, V_\mathrm{U})$ with volume measurements (top) and the corresponding signals of the non-virtual sensors $y_{1,k}$ and $y_{2,k}$ (bottom). The identification samples S_m with $m \in \{1, ..., 5\}$ are generated based on the volume measurements $\tilde{\boldsymbol{k}}$.

4. FILL VOLUME ESTIMATION

In this chapter, we will introduce the investigated filters for fill volume estimation: a filter based on model prediction, with separate measurement update, and two particle filters, that consider the measurement update sequentially. All filters have in common, that, if a new volume measure occurs ($\boldsymbol{y}_k \neq \boldsymbol{y}_{k-1}$), the fill volume estimation is reset to the volume measurement $V_{\tilde{h}(k)}$ with low variance $\epsilon^2 \ll 1$.

4.1 Model Prediction Filter

First, we will introduce a filter considering the model prediction in the first instance, which leads to normal distributed fill volume estimations. The measurement update is done separately by truncating the normal distribution. Advantageously, the model prediction probability distribution can be determined analytically with low computational effort. Additionally, the past non-truncated estimations can also be updated analytically by the Rauch-Tung-Striebel smoother, see Rauch et al. (1965), when new volume measurements occur. Disadvantageously, it is to be expected that the estimation of the fill volume becomes worse with increasing time difference to the last volume measurement since impossible fill volumes are considered in every iteration. The fill volume prediction distribution based on the last occurred volume measurement

$$\hat{m}_k = \max\left(\left\{ m \in \{0, ..., k\} \mid \boldsymbol{y}_m \neq \boldsymbol{y}_{m-1} \right\}\right) \tag{19}$$

is

$$\hat{p}\left(x_k \mid \boldsymbol{y}_{\hat{m}_k}\right) = \int p\left(x_k \mid x_{k-1}\right) \hat{p}\left(x_{k-1} \mid \boldsymbol{y}_{\hat{m}_k}\right) \mathrm{d}x_{k-1} \tag{20}$$

$$= \mathrm{N}\left(x_k \mid \mu_k, P_k\right), \tag{21}$$

with, see (6), (8) and (13),

$$P_0 = \frac{1}{\epsilon^2}, \qquad P_k = \begin{cases} \epsilon^2 & \boldsymbol{y}_k \neq \boldsymbol{y}_{k-1} \\ Q_{k-1} + P_{k-1} & \text{otherwise,} \end{cases} \tag{22}$$

$$\mu_0 = \frac{V_{\mathrm{L},0} + V_{\mathrm{U},0}}{2}, \ \mu_k = \begin{cases} V_{\tilde{h}(k)} & \boldsymbol{y}_k \neq \boldsymbol{y}_{k-1} \\ \delta_{x_{k-1}} + \mu_{k-1} & \text{otherwise.} \end{cases} \tag{23}$$

The measurement update step takes place separately upon this ongoing model prediction, limiting the normal distribution to the valid interval $[V_{\mathrm{L},k}, V_{\mathrm{U},k})$

$$x_k \sim \mathrm{T}\left(x_k \mid \mu_k, P_k, V_{\mathrm{L},k}, V_{\mathrm{U},k}\right), \tag{24}$$

which leads approximately to a uniform distributed x_0.

4.2 Bayesian Particle Filters

A normal distributed fill volume will always consider impossible states x_{k-1} while model prediction. This recursive error increases continuously until a new volume measurement leads to a new initialization of the distribution. This can be prevented by updating the model prediction with the measurement likelihood function (10). In this paper, we will compare two particle filters: a sequential Monte Carlo filter, see N. J. Gordon et al. (1993) and Kitagawa (1996), with random particle initialization and a filter with deterministic particle initialization that is used to approximate the cumulative distribution function of the new state's probability distribution.

Sequential Monte Carlo Filter: In literature, see Särkkä (2013), the sequential Monte Carlo filter is also known as particle filter and consists of the following steps to approximate the posterior distribution $p\left(x_k \mid \boldsymbol{y_{1:k}}\right)$:

- Draw N particles $x_0^{(i)}$ from the initial distribution $p\left(x_0\right)$ and set their weights $w_0^{(i)}$ by
$$x_0^{(i)} \sim p\left(x_0\right), \quad w_0^{(i)} = \frac{1}{N} \forall i \in \{1, ..., N\}. \tag{25}$$
- For $k = 1, ..., \hat{k}$:
  (1) Draw N particles from the importance distributions
  $$x_k^{(i)} \sim q\left(x_k \mid x_{k-1}^{(i)}, \boldsymbol{y}_{1:k}\right), \tag{26}$$
  (2) calculate the new weights by
  $$w_k^{(i)} = w_{k-1}^{(i)} \frac{p\left(\boldsymbol{y}_k \mid x_k^{(i)}\right) p\left(x_k^{(i)} \mid x_{k-1}^{(i)}\right)}{q\left(x_k^{(i)} \mid x_{k-1}^{(i)}, \boldsymbol{y}_{1:k}\right)} \tag{27}$$
  and normalize them to sum to unity.
  (3) If the effective number of particles is to low, draw a new set of N equal weighted particles with probability $w_k^{(i)}$ for particle $x_k^{(i)}$ to be drawn (resampling).

In the presented application the initial particles are drawn from a uniform distribution

$$x_0^{(i)} \sim p\left(x_0\right) = \mathrm{U}\left(x_0 \mid V_{\mathrm{L},0}, V_{\mathrm{U},0}\right), \tag{28}$$

as only the valid interval is known. The importance distribution in Eq. 26 is either the model prediction (8) or, in case of a volume measurement, a truncated normal distribution with low variance, which leads to

$$x_k^{(i)} \sim \begin{cases} \mathrm{T}\left(x_k \mid V_{\tilde{h}(k)}, \epsilon^2, V_{\mathrm{L},k}, V_{\mathrm{U},k}\right) & \boldsymbol{y}_k \neq \boldsymbol{y}_{k-1} \\ p\left(x_k \mid x_{k-1}^{(i)}\right) & \text{otherwise.} \end{cases} \tag{29}$$

The weight update in Eq. 27 than simplifies to

$$w_k^{(i)} = \begin{cases} \frac{1}{N} & \boldsymbol{y}_k \neq \boldsymbol{y}_{k-1} \\ w_{k-1}^{(i)} p\left(\boldsymbol{y}_k \mid x_k^{(i)}\right) & \text{otherwise.} \end{cases} \tag{30}$$

The particles are yet reset infrequently if $\boldsymbol{y}_k \neq \boldsymbol{y}_{k-1}$. Additionally, resampling takes place when selected particles violate the valid interval as the measurement update (10) sets their weights to zero. For every particle z with zero weight a new particle is drawn from the model prediction (8) for a randomly selected particle R of the particle distribution at time step $k-1$ (considering $w_{k-1}^{(i)} = \frac{1}{N} \ \forall \ i \in \{1, ..., N\}$) until the sampling condition $\sum_{i=1}^{N} w_k^{(i)} = 1$ is fulfilled by

$$x_k^{(z)} \sim p\left(x_k \mid x_{k-1}^{(R)}\right) \ \forall \ z \in \{1, .., N\} \mid w_k^{(z)} = 0, \tag{31}$$

$$w_k^{(z)} = \frac{p\left(\boldsymbol{y}_k \mid x_k\right)}{N} \ \forall \ z \in \{1, .., N\} \mid w_k^{(z)} = 0 \tag{32}$$

with

$$R \sim \mathrm{U}\left(1, N\right). \tag{33}$$

Deterministic Particle Filter: For the second particle filter, the particles are not drawn from the distribution but created by a deterministic rule. In Huber and Hanebeck (2008) a particle selection capturing the mean and variance of the true density and minimizing the Cramér–von Mises distance is proposed. Tenne and Singh (2003) present an extension of the unscented Kalman Filter to capture moments of higher order of the true density. As in the presented scenario the measurement likelihood truncates the model prediction to the valid interval, it seems reasonable to distribute the particles evenly over the entire distribution. Therefore, the particles are placed in intervals, defined by the cumulative distribution function $P\left(x\right)$. Each interval is represented by one particle, which leads to particles with equal probability and therefore equal weights. In this paper the particle creation function $C\left(p(x), N\right)$ for N particles from probability distribution p are based on the interval's median

$$C_{\mathrm{M}}\left(p(x), N\right) = x^{(i)} = P^{-1}\left(\frac{i - \frac{1}{2}}{N}\right) \tag{34}$$

or mean

$$C_{\mathrm{E}}\left(p(x), N\right) = x^{(i)} = \frac{\int_{x_{\mathrm{L}}}^{x_{\mathrm{U}}} x p\left(x\right) \mathrm{d}x}{P\left(x_{\mathrm{U}}\right) - P\left(x_{\mathrm{L}}\right)}, \tag{35}$$

with

$$P\left(x_{\mathrm{L}}\right) = \frac{i-1}{N}, \quad P\left(x_{\mathrm{U}}\right) = \frac{i}{N}. \tag{36}$$

Equation (36) also defines the lower x_{L} and upper x_{U} interval borders. Placing the particles on the interval's mean value need the borders to be computed and it is not guaranteed for all probability distributions that the integral has an analytic solution or converges. The generated particles are shown in Fig. 3. Like the sequential Monte Carlo filter, the particles are initialized by a uniform distribution

$$x_0^{(i)} = C\left(\mathrm{U}\left(x_0 \mid V_{\mathrm{L},0}, V_{\mathrm{U},0}\right), N\right), \tag{37}$$

$$w_0^{(i)} = \frac{1}{N}. \tag{38}$$

The combination of all particles $x_{k-1}^{(i)}$ with all particles created from the model change distribution

$$\Delta_{x_{k-1}}^{(i)} = C\left(\mathrm{N}\left(\Delta_{x_{k-1}} \mid \delta_{x_{k-1}}, Q_{k-1}\right), N\right) \tag{39}$$

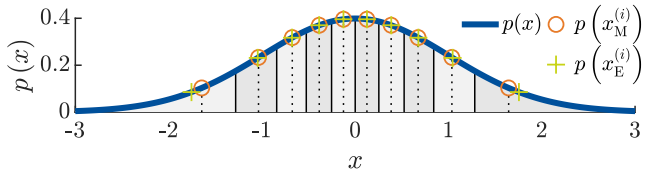

Fig. 3. Generated median $x_{\mathrm{M}}^{(i)}$ and mean $x_{\mathrm{E}}^{(i)}$ particles with corresponding intervals for the probability density function of a standard normal distribution.

lead to N^2 model prediction particles $\tilde{x}_k^{(j)}$

$$\tilde{x}_k^{(j)} = x_{k-1}^{\left(1+\lfloor \frac{j-1}{N} \rfloor\right)} + \Delta_{x_{k-1}}^{\left(j-\lfloor \frac{j-1}{N} \rfloor N\right)}, \tag{40}$$

$$\tilde{w}_k^{(j)} = \frac{p\left(\boldsymbol{y}_k \mid \tilde{x}_k^{(j)}\right)}{N^2}. \tag{41}$$

The cumulative distribution function $P\left(x_k \mid \boldsymbol{y}_{1:k}\right)$ can be determined by sorting the particles $\tilde{x}_k^{(j)}$ increasingly so that $\tilde{x}_k^{(j)} \leq \tilde{x}_k^{(j+1)}\ \forall\ j \in \{1, ..., N^2 - 1\}$. To keep the particle size with non-zero weight constant, N particles are selected by the creation functions, see (34) and (35), adapted for discrete distributions. For the median creation rule the particles are drawn by

$$x_k^{(i)} = \begin{cases} C_{\mathrm{M}}\left(\mathrm{T}\left(x_k \mid V_{\tilde{h}(k)}, \epsilon, V_{\mathrm{L},k}, V_{\mathrm{U},k}\right), N\right) & \boldsymbol{y}_k \neq \boldsymbol{y}_{k-1} \\ \tilde{x}_k^{(c_i)} & \text{otherwise,} \end{cases} \tag{42}$$

where c_i are the particles minimizing the distance from the discrete cumulative distribution function to the interval's medians at $P(x) = \frac{i-\frac{1}{2}}{N}$

$$c_i = \arg\min_{c_i} \left| \frac{\sum_{j=1}^{c_i} \tilde{w}_k^{(j)}}{\sum_{j=1}^{N^2} \tilde{w}_k^{(j)}} - \frac{i-\frac{1}{2}}{N} \right|. \tag{43}$$

In case of the mean representation the particles are drawn by

$$x_k^{(i)} = \begin{cases} C_{\mathrm{E}}\left(\mathrm{T}\left(x_k \mid V_{\tilde{h}(k)}, \epsilon, V_{\mathrm{L},k}, V_{\mathrm{U},k}\right), N\right) & \boldsymbol{y}_k \neq \boldsymbol{y}_{k-1} \\ \frac{\sum_{j=L_i}^{U_i} \tilde{x}_k^{(j)}}{|U_i - L_i|} & \text{otherwise,} \end{cases} \tag{44}$$

where L_i and U_i are the lower and upper boundary particle, that minimize the distance of the discrete cumulative distribution function to the interval's borders at $P(x) = \frac{i-1}{N}$

$$L_i = \min\left(b\right), \text{ s.t.} \frac{\sum_{j=1}^{b} \tilde{w}_k^{(j)}}{\sum_{j=1}^{N^2} \tilde{w}_k^{(j)}} \geq \frac{i-1}{N}, \tag{45}$$

$$U_i = \max\left(b\right), \text{ s.t.} \frac{\sum_{j=1}^{b} \tilde{w}_k^{(j)}}{\sum_{j=1}^{N^2} \tilde{w}_k^{(j)}} < \frac{i}{N}. \tag{46}$$

5. VALIDATION AND COMPARISON

The introduced filters will be compared in two scenarios. First, the goodness of estimating a ground truth probability distribution is investigated in an exemplary scenario, where the fill level gets closer to a sensor position without reaching it. In the second scenario, the hopper scenario, the filters will be compared to a simulated fill volume based on the real plant inputs.

5.1 Exemplary scenario

In the exemplary scenario, the fill volume is initialized at $x_0 = 0.75$ and the volume at the sensor position is $V_1 = 0.7$. In every iteration the fill volume is decreased by the probability distribution $\delta_{x_{k-1}} \sim \mathrm{N}\left(-0.001, 0.004\right)$. The relevant sensor signals are $y_{1,k} = 1$ and $y_{2,k} = 0$ so that the fill volume is always beyond V_1. The following filters will be compared:

- the model prediction only filter with subsequent measurement update (MPF),
- a sequential Monte Carlo filter with $N = 1\mathrm{e}2$ (MCF_2),
- a sequential Monte Carlo filter with $N = 1\mathrm{e}4$ (MCF_4),
- a deterministic particle filter with median creation rule and $N = 1\mathrm{e}2$ ($\mathrm{DPF}_{\mathrm{M}}$),
- a deterministic particle filter with mean creation rule and $N = 1\mathrm{e}2$ ($\mathrm{DPF}_{\mathrm{E}}$).

The number of particles for the MCF_4 was chosen to lead to an approximately equal computational effort compared to the deterministic particle filters. In 1e5 calculation the mean computing times were 0.91 ms for the MCF_4, 1.64 ms for the $\mathrm{DPF}_{\mathrm{M}}$ and 0.83 ms for the $\mathrm{DPF}_{\mathrm{E}}$. For the comparison of accuracy the ground truth is approximated by a Monte Carlo simulation with 1e7 initialized particles and in every iteration step k the invalid particles are deleted; 6.5e5 particles remain at $k = 100$. The filters are compared regarding mean, variance, and the symmetric Kullback Leibler divergence between the ground truth p and the estimated distribution $\hat{p}$. As discrete and continuous distributions with different resolutions have to be compared, the symmetric Kullback Leibler divergence

$$D_{\mathrm{SKL}}\left(p||\hat{p}\right) = D_{\mathrm{KL}}\left(p||\hat{p}\right) + D_{\mathrm{KL}}\left(\hat{p}||p\right), \tag{47}$$

with

$$D_{\mathrm{KL}}\left(p||\hat{p}\right) = \sum_{i=1}^{N+1} \Delta P_i \log \frac{\Delta P_i}{\Delta \hat{P}_i} \tag{48}$$

and

$$\Delta P_i = P\left(\lambda_i\right) - P\left(\lambda_{i-1}\right),\ P\left(\lambda_0\right) = 0,\ P\left(\lambda_{N+1}\right) = 1, \tag{49}$$

$$\lambda_i = \hat{P}^{-1}\left(\frac{i-\frac{1}{2}}{N}\right),\ i \in \{1, ..., N\} \tag{50}$$

was evaluated in the intervals defined by the particle generation (34) for $N = 100$ particles (and 101 intervals), which is the minimum number of particles of the compared filters. The results are shown in Fig. 4. The MPF shows the best performance of all filters in the first iterations as long as the fill volume distribution is not truncated significantly by the sensor measurement (also visible by the linear decreasing mean and increasing variance). As expected, with increasing k the performance deteriorates, except the variance estimation. The sequential Monte Carlo filters MCF_2 and MCF_4 show a random walk for all k as all particles are generated and updated randomly, which is less pronounced as the number of particles increases. However, a constant drift is not visible and the Kullback-Leibler divergence is approximately constant. The deterministic particle filters $\mathrm{DPF}_{\mathrm{E}}$ and $\mathrm{DPF}_{\mathrm{M}}$ create particles by identical rules in every iteration and underestimate the variance (see Fig. 3). In combination with the measurement update, deleting the particles mainly one-sided if the estimated probability

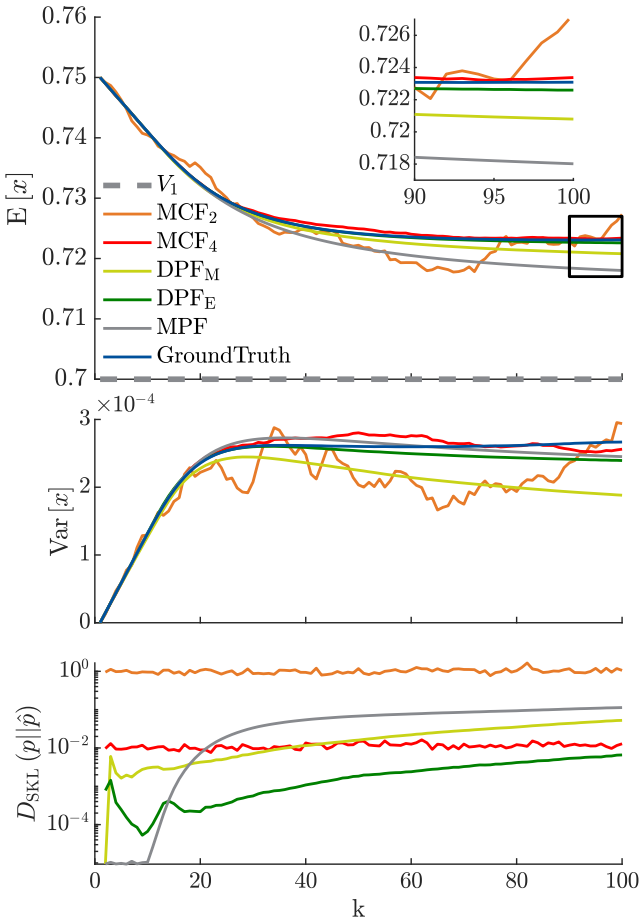

Fig. 4. Mean and variance of the filters compared to the ground truth and symmetric Kullback Leibler divergence of the filters (from top to bottom). The Monte Carlo filter results are one possible estimation and vary in every simulation.

function is not symmetric to the valid interval's center, the filters show a constant drift in mean and variance estimation. The DPF_E shows a better performance regarding the three accuracy measures as the particles are generated in the probability centers of the intervals. More complex creation functions, considering the centric moments of the distribution that is to be approximated, are imaginable for higher accuracy but out of the scope of this paper. Considering the exemplary scenario, the MPF, DPF_E, and MCF_4 will be investigated in the hopper scenario. The MPF is expected to show the best performance if the model prediction is a good approximation of the real system and the fill level is only occasionally near the sensor mounting positions. The DPF_E is expected to show the best performance if the fill level is often near the sensor position and occasionally volume measurements occur. The MCF_4 is expected to show the best performance if volume measurements occur rare.

5.2 Hopper scenario

The MPF, DPF_E, and MCF_4 are compared in the hopper scenario. A sequence of 2e6 plant's input samples α_k and w_k are used to simulate the model in closed loop, see (6), (7) and (11). As no identification errors are taken into consideration and $\epsilon^2 = 5.4\text{e}{-5}$ was determined by the simulated fill level, the results are supposed to be the most exact and probably deteriorate in the real scenario. Exemplary parts of the sequence of the estimation compared to the simulated fill volume $x_{S,k}$ can be seen in Fig. 5. The differences are relatively small and for the MCF_4 and DPF_E not visible. As in the exemplary scenario, the MPF estimated probability distribution is closer to the sensor volume. For comparison the estimation of the

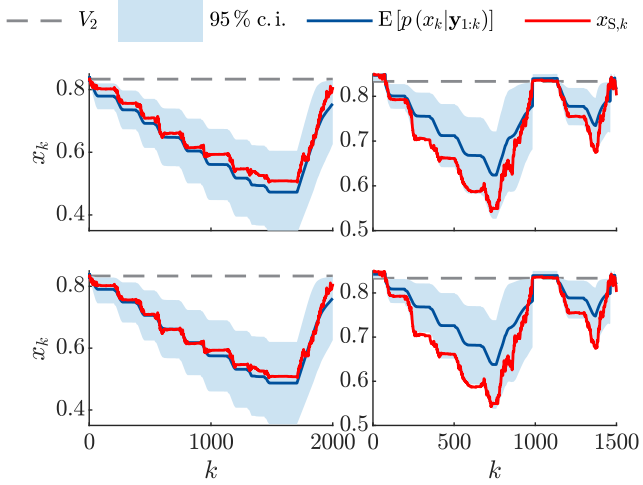

Fig. 5. The simulated fill volume, the estimated mean and 95 % credible interval (c. i.) for a randomly selected fill cycle (left) and the fill cycle with maximum RMSE (right) for the DPF_E and MCF_4 (top), which have almost identical results, and the MPF (bottom). To improve readability, time index k is set to zero at the origin of every plot. Nevertheless, all presented plots are small windows from one long sequence.

fill volume distribution only based on the measurement $p(x_k \mid \boldsymbol{y}_k) = \mathrm{U}(x_k \mid V_{L,k}, V_{U,k})$ and the model prediction $p(x_k \mid x_{k-1})$, see (8), are also evaluated. This leads to different dependencies, $\boldsymbol{y}_{1:k}$, $\boldsymbol{y}_k$, and x_{k-1}, of the considered fill volume distributions and we will generally denote D_k in the following, which is either $\boldsymbol{y}_{1:k}$ or $\boldsymbol{y}_k$ or x_{k-1} dependent on the fill volume distribution.

Four validation metrics are proposed based on the estimated fill volume distribution $p(x_k \mid D_k)$:

- the root-mean-square error (RMSE) of the mean,
- the amount of simulated fill volumes within the $p_{ci} = 95\,\%$ and $p_{ci} = 75\,\%$ credible intervals, limited by the lower $x_{L,k}$ and upper bound $x_{U,k}$

$$x_{L,k} = P^{-1}\left(\frac{1-p_{ci}}{2} \mid D_k\right), \tag{51}$$

$$x_{U,k} = P^{-1}\left(\frac{1+p_{ci}}{2} \mid D_k\right), \tag{52}$$

- the average value of the probability density function of the estimated fill volume distribution at the simulated fill volume, which is e. g. a common loss function for training and validation in Gaussian process regression (see Bousquet (2004)). Average in this context mean

$$\overline{p} = \sqrt[\hat{k}]{\prod_{k=1}^{\hat{k}} p(x_{S,k}|D_k)}. \tag{53}$$

The truncated normal fill volume distribution is identified from the particle representation by minimizing the symmetric Kullback Leibler divergence in (47). The performance measures can be seen in table 1. The results of the particle filter are the mean of three separated iterations. As the standard deviation would not be visible in the chosen resolution, it is not shown. The differences between the filters are low, with the MCF_4 showing the best results for the selected hopper. It is expected that the Bayesian filters perform better when applied to the real systems as they consider in every iteration also the measurement signal, which leads to a higher robustness against model errors.

Table 1. Results for one hopper

	RMSE	95 % c. i.	75 % c. i.	$\overline{p}$
$p(x_k \mid \boldsymbol{y}_k)$	0.302	62.08 %	7.26 %	1.50
$p(x_k \mid x_{k-1})$	2.422	99.84 %	70.09 %	0.13
MPF	0.020	95.63 %	76.95 %	18.35
MCF_4	0.018	95.69 %	76.14 %	19.57
DPF_E	0.018	94.98 %	74.78 %	19.56

6. CONCLUSION AND FUTURE WORK

In this paper, different filters for estimating the fill volume probability distribution of a sand hopper in casting processes based on a stochastic model and a measurement likelihood from binary point level sensors were proposed. It was shown how the stochastic physical model was created and identified considering the available plant data. The compared filters were the model prediction filter, where the measurement update is performed separately from the ongoing model prediction, a sequential Monte Carlo filter with random particle selection, and a deterministic particle filter, where the particles are created based on the cumulative distribution function. In an exemplary scenario, the differences of the filters in approximating a fill level distribution, were investigated, and in a simulated scenario of the closed loop sand hopper, the filters were compared with respect to the estimation of a specific fill volume, showing low differences in the performance measures. The main aspect for higher estimation accuracy is therefore expected in the model accuracy so that other process noise will be investigated in future work. Even if it is known that the hopper is only filled, white process noise will consider the possibility of a decreasing fill volume. Separated one-side limited process noise distribution (e. g. gamma distribution) for filling and emptying may perform better in the real scenario. Applying the filters to the real sand hopper system and compare different smoothers when new volume measurements occur, will be the following steps to investigate and improve the fill volume estimation.

REFERENCES

Babuska, R., Lendek, Z., Braaksma, J., and de Keizer, C. (2006). Particle filtering for on-line estimation of overflow losses in a hopper dredger. In *2006 American Control Conference*, 6–pp.

Bousquet, O. (ed.) (2004). *Advanced lectures on machine learning: ML Summer Schools 2003, Canberra, Australia, February 2 - 14, 2003, Tübingen, Germany, August 4 - 16, 2003 ; revised lectures*, volume 3176 of *Tutorial*. Springer, Berlin. doi:10.1007/b100712.

Chetpattananondh, K., Tapoanoi, T., Phukpattaranont, P., and Jindapetch, N. (2014). A self-calibration water level measurement using an interdigital capacitive sensor. *Sensors and Actuators A: Physical*, 209, 175–182.

DeGroot, M.H. and Schervish, M.J. (2012). *Probability and statistics*. Addison-Wesley, Boston, Mass., 4. ed. edition.

Hameed, B., Khan, I., Durr, F., and Rothermel, K. (2010). An rfid based consistency management framework for production monitoring in a smart real-time factory. In *Internet of things (IOT), 2010*, 1–8. IEEE, Piscataway, NJ.

Huber, M.F. and Hanebeck, U.D. (2008). Gaussian filter based on deterministic sampling for high quality nonlinear estimation. *IFAC Proceedings Volumes*, 41(2), 13527–13532. doi:10.3182/20080706-5-KR-1001.02291.

Itävuo, P., Hulthén, E., and Vilkko, M. (2017). Feedhopper level estimation and control in cone crushers. *Minerals Engineering*, 110, 82–95.

Kitagawa, G. (1996). Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(1), 1–25.

Maurer, I., Riva, M., Hansen, C., and Ortmaier, T. (2017). Cloud-based plant and process monitoring based on a modular and scalable data analytics infrastructure. In T. Schüppstuhl, J. Franke, and K. Tracht (eds.), *Tagungsband des 2. Kongresses Montage Handhabung Industrieroboter*. Springer Vieweg, Berlin.

N. J. Gordon, D. J. Salmond, and A. F. M. Smith (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings F - Radar and Signal Processing*, 140(2), 107–113.

O'Donovan, P., Leahy, K., Bruton, K., and O'Sullivan, D.T.J. (2015). An industrial big data pipeline for data-driven analytics maintenance applications in large-scale smart manufacturing facilities. *Journal of Big Data*, 2(1), 117.

Qi, Q. and Tao, F. (2018). Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison. *IEEE Access*, 6, 3585–3593.

Rauch, H.E., Striebel, C.T., and Tung, F. (1965). Maximum likelihood estimates of linear dynamic systems. *AIAA journal*, 3(8), 1445–1450.

Särkkä, S. (2013). *Bayesian filtering and smoothing*, volume 3 of *Institute of Mathematical Statistics textbooks*. Cambridge University Press, Cambridge.

Schütte, F., Beineke, S., Grotstollen, H., Fröhleke, N., Witkowski, U., Rückert, U., and Rüping, S. (1997). Structure-and parameter identification for a two-mass-system with backlash and friction using a self-organizing map. In *European Conference on Power Electronics and Applications*, volume 3, 3–358.

Tao, F., Qi, Q., Liu, A., and Kusiak, A. (2018). Data-driven smart manufacturing. *Journal of Manufacturing Systems*.

Teng, J., Snoussi, H., and Richard, C. (2010). Decentralized variational filtering for target tracking in binary sensor networks. *IEEE Transactions on Mobile Computing*, 9(10), 1465–1477.

Tenne, D. and Singh, T. (2003). The higher order unscented filter. In *Proceedings of the 2003 American Control Conference*, 2441–2446. IEEE, Piscataway, N.J. doi:10.1109/ACC.2003.1243441.