CODEN:LUTEDX/(TEIE-7291)/1-29/(2022)

Industrial Electrical Engineering and Automation

Detecting sensor faults using the CUSUM method and a soft sensor for the return activated sludge flow at Henriksdal WWTP

Hanna Molin

Division of Industrial Electrical Engineering and Automation
Faculty of Engineering, Lund University

Detecting sensor faults using the CUSUM method and a soft sensor for the return activated sludge flow at Henriksdal WWTP

January 12, 2022

Contents

Abstract

A soft sensor based on three different models for estimating the return activated sludge (RAS) flow rate at Henriksdal wastewater treatment plant, Stockholm, Sweden, was used together with the CUSUM algorithm to detect sensor faults. A fault generation model was implemented to simulate three types of faults: drift, shift, and complete failure, in five different sensors. The CUSUM algorithm was used to detect faults by comparing the three soft sensor model outputs to a decided reference flow rate. All types of faults were detected in all sensors, but with different detection delays. When using a reference flow independent of the model outputs, the work presented her could be extended to not only detect the faults, but to determine which sensor is faulty as well. Using a references flow that is dependent on the model outputs lowers the detection delay but makes it difficult to isolate the faults.

1 Introduction

The Henriksdal wastewater treatment plant (WWTP) in Stockholm, Sweden, is currently undergoing a large rebuild to face future challenges as the city grows and the requirements on the effluent becomes stricter. Once finalized, the plant will have a membrane bioreactor (MBR) process with pre and post denitrification. The number of people connected to Henriksdal WWTP is estimated to be 1 621 000 people by 2040 (Stockholm Vatten och Avfall, 2017).

The MBR process is similar to the more conventional activated sludge process (ASP) but the solids in the water exiting the post denitrification basin are removed from the effluent by filtration in aerated membranes instead of a traditional settling tank (Figure 1). The majority of the sludge is recirculated as return activated sludge (RAS) that is pumped, with a lift height of approximately 1 m, back to a deoxidation zone. The excess sludge, or waste activated sludge (WAS), is thickened and digested to produce biogas. The membranes reduce suspended solids effectively and can achieve very low levels of suspended solids in the effluent (Hammer and Hammer, 2014).

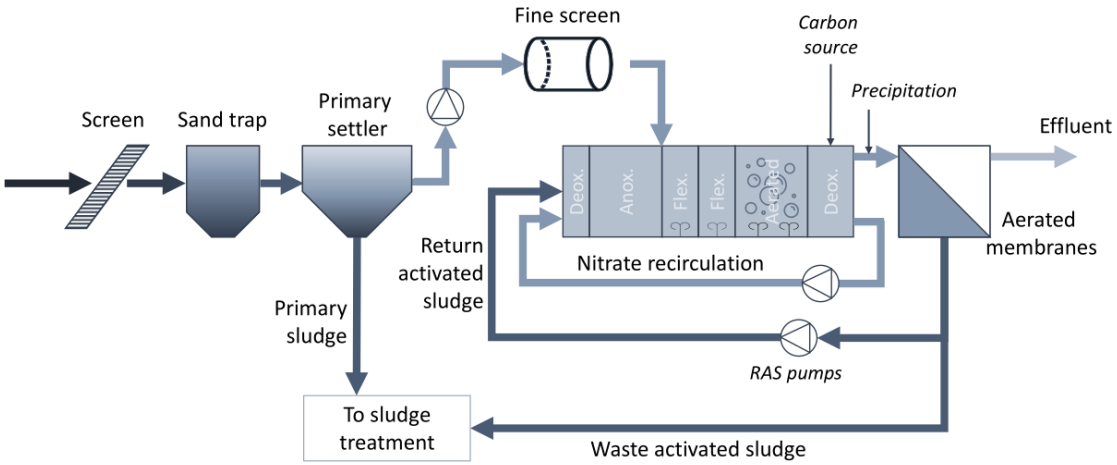

Figure 1. A schematic overview of the treatment process at Henriksdal WWTP.

The sludge concentration in the aerated zone should be kept stable and constant as an average over time, which is controlled by the excess sludge flow rate Q_{WAS}. The partitioning of sludge between the membrane tank and the bioreactor is controlled by the return flow rate, Q_{RAS}. Due to physical constraints at the treatment plant it is not possible to measure Q_{RAS} which makes it difficult to supervise the process, control the pumps and in the extension also the treatment process. The RAS flow is currently estimated based on the pump frequency – an estimate that is only valid as long as no disturbances are present in the pumps, and that does not account for the water level in the pump tank. The flow is per default controlled to be proportional to the incoming flow rate, Q_{bio} with a factor 4.

The MBR process is one of the most energy demanding parts of the treatment plant. The high demand mostly stems from the aeration of the membranes but the RAS pumps that pump the RAS flow back to the aeration also have a big contribution. The treatment process will be split in seven lines, each with eight RAS pumps, adding up to a total of

56 RAS pumps. Better knowledge of the RAS flow may help in controlling the process, which in turn can optimize the control of the pumps and thereby lower the energy demand.

1.1 Estimation of the return activated sludge flow rate

A soft sensor that make use of three different models of the RAS flow rate, derived from (1) a static mass balance over the membrane tank, (2) the empirical relation between water level and flow rate over rectangular weirs, and (3) a pump model (an extension of the estimate currently used at the real plant), was recently developed. The models are presented in more detail below.

1.1.1 Static mass balance over the membrane tank

The static mass balance model was derived from a mass balance over the membrane reactor (Figure 2). The full mass balance at steady-state is given by Equation (1)

$$Q_{tot}SS_{bio} = Q_{RAS}SS_{RAS} + Q_{out}SS_{out} + Q_{WAS}SS_{RAS} \tag{1}$$

where Q_{tot} is the total flow rate coming in to the membrane reactor, SS_{bio} is the suspended solids concentration in the bioreactor, Q_{RAS} is the RAS flow rate, SS_{RAS} is the suspended solids concentration of the RAS, Q_{out} is the effluent flow rate, SS_{out} is the suspended solids concentration in the effluent, and Q_{WAS} is the WAS flow rate. Under the assumption that $SS_{out} = 0$ mg/l and $Q_{WAS} = 0$ m^3/s, Q_{RAS}, can be estimated as:

$$Q_{RAS} = \frac{Q_{bio}SS_{bio}}{SS_{RAS} - SS_{bio}} \tag{2}$$

where Q_{bio}, SS_{RAS} and SS_{bio} are measured on-line as shown in Figure 2.

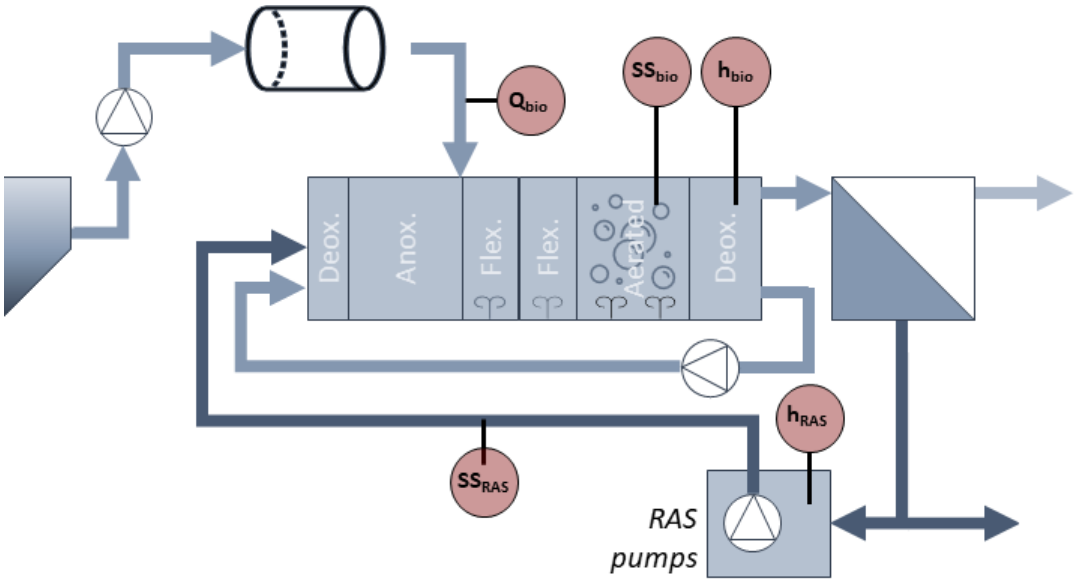

Figure 2. A schematic overview of the sensors' location in Henriksdal WWTP.

1.1.2 Flow modelling over the weir

The total flow exits the bioreactor over a rectangular weir and can be estimated from Poleni's equation (Persson et al., 2014) as

$$Q_{tot} = \frac{2\mu L\sqrt{2g}\,(h_{bio} - h_{ref})^{3/2}}{3} \tag{3}$$

where

$$Q_{tot} = Q_{RAS} + Q_{bio}, \tag{4}$$

μ is an empirical coefficient that accounts for friction losses, L is the length of the weir, g is the gravitational constant, h_{bio} is the water level in the bioreactor and h_{ref} is the height to the crest of the weir. h_{bio} is measured on-line (Figure 2), the remaining parameters are known. μ is given by

$$\mu = 0.602 + 0.075\frac{h_{bio} - h_{ref}}{h_{ref}} \tag{5}$$

or on average, μ = 0.65. Combining Equations (4) and (3) gives Q_{RAS}:

$$Q_{RAS} = \frac{2\mu L\sqrt{2g}\,(h_{bio} - h_{ref})^{3/2}}{3} - Q_{bio} \tag{6}$$

1.1.3 Pump model

Lastly, a pump model first developed by Saagi et al. (2016) and later revised to fit the RAS pumps at Henriksdal by Blomstrand and Jemander (2017) was used to estimate the RAS flow rate based on the water level in the pump tank, h_{RAS}, (Figure 2) and the frequency the pumps currently operate at. The pump model consist of a polynomial estimated from two separate operational points in the system and pump curves of the pump. The pump curve describes the flow at different pressure heads/lift heights, whereas the system curve describes the system the pump is located in. The system curve accounts for a static pressure and the friction losses at different pump flow rates. The operational point is where the two curves intersect. The pump model is based on these two curves (Figure 3). The static pressure will vary with the water level in the tank in which the RAS pumps are installed. Therefore, the model accounts for h_{RAS}. A system curve was first developed by fitting a polynomial to data from the pump supplier:

$$H = 1.08 * 10^{-9}Q^2 + r \tag{7}$$

where H is the static pressure, Q is the pump flow rate, and r is a correction factor that was added to account for the varying water level. The pump curve was assumed to be linear around the operational point and the intercept with the system curve. The affinity laws;

$$\frac{Q_1}{Q_2} = \frac{n_1}{n_2} \tag{8}$$

$$\frac{H_1}{H_2} = \left(\frac{n_1}{n_2}\right)^2 \tag{9}$$

$$\frac{P_1}{P_2} = \left(\frac{n_1}{n_2}\right)^3 \tag{10}$$

where Q is the flow, n is the number of rounds (easily translated to frequency), H is the total pressure head, and P is the power demand, was used to approximate the pump curves at different frequencies. Combining Equation (7) with the pump curves yields in a second order equation that was solved to estimate the RAS flow rate at different frequencies and static pressure (Figure 3). To change the flow rate, the operational point has to be moved which can either be done by changing the frequency or to change the static head or frictional head loss (Karassik et al., 2007).

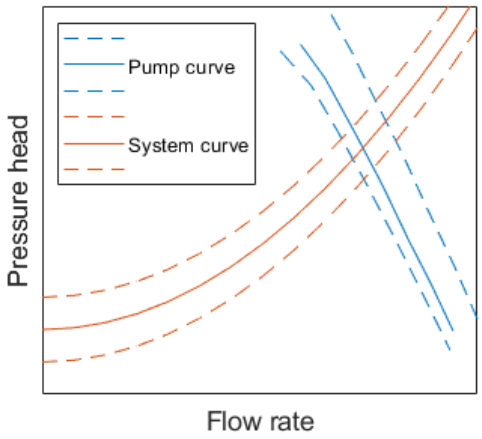

Figure 3. An illustration of system curves at different water levels (static pressure) and pump curves at different frequencies.

1.1.4 Soft sensor

The three models described above, and the median of the output from them (Q_{median}), were used as a soft sensor. There was no known true RAS flow rate to calibrate the models. Instead, they were partly evaluated by adding white noise to the input signals h_{bio}, Q_{bio}, SS_{bio}, SS_{RAS} and h_{RAS}, one at a time. The white noise was chosen with mean zero and standard deviations to match historic data from Henriksdal WWTP (Table 1). The output from the models were compared to the output when no disturbances were added. The statistics are shown in Table 2. The local sensitivity analysis showed that the weir model has the highest standard deviation compared to its mean (46%) which indicates that it is most sensitive to disturbances. The pump model has very low standard deviation which indicates that it is least sensitive. However, a global sensitivity analysis should be performed to understand the full effect of uncertainties in the models and their inputs.

Table 1. Summary of historic data from Henriksdal WWTP.

Signal	Mean	Standard deviation	Standard deviation / Mean
Q_{bio}	0.82 m^3/s	0.18 m^3/s	22%
h_{bio}	1.0 m	0.030 m	3%
SS_{bio}	5 768 g/m^3	419 g/m^3	7%
SS_{RAS}	8 013 g/m^3	604 g/m^3	8%
h_{RAS}	1.2 m	0.18 m	15%

Table 2. The standard deviation (in m^3/s and percentage of the average flow) of the difference between a noise free flow estimate and one with white noise added to all input signals according to Table 1

Flow model	Standard deviation	Std. dev. / average flow
Static mass balance (Q_{SMB})	0.88 m^3/s	23%
Weir (Q_{weir})	1.7 m^3/s	46%
Pump (Q_{pump})	0.012 m^3/s	0.33%
Median (Q_{median})	0.045 m^3/s	19%

1.2 Objective

The objective of this study was to evaluate if it is possible to use the different models of Q_{RAS} as a basis for fault detection. Two cases were considered where it was assumed that:

Case 1: the pump model gives the true RAS flow rate, and

Case 2: the median of the three model outputs gives the true flow rate.

The reference flow rate in Case 1 is independent of the other flow rate estimates, whereas in Case 2, the reference flow rate is dependent of all three model outputs. By analysing the residuals of the RAS flow rate estimates it should in theory be possible to detect when any of the estimates deviates from the assumed true flow rate, and from this detect faults in the partaking sensors. The study was done with simulated data from a process model of Henriksdal WWTP, based on the Benchmark Simulation Models (BSM).

2 Fault detection

Wastewater treatment plants are, broadly speaking, controlled to operate in a certain way to achieve a predetermined goal. The control of the treatment processes relies on on-line measurements, complimented by lab analyses. Irregularities in the WWTP and the process data can either come from the process itself due to disturbances, fluctuations or a slow change of state; the mechanical equipment like pumps or valves; or sensor faults. The sensors have a certain accuracy and uncertainty that influence the data, but can also be prone to an array of faults: drift, bias, malfunction, incorrect scaling, amongst others

(Newhart et al., 2019; Samuelsson, 2021). Fault detection may help to improve the data quality which in turn will improve the overall process control. It can also be used for planning maintenance of mechanical equipment or re-calibration of sensors.

A good fault detector should have few false alarms and a short detection delay. The fault detector should be designed so that it is insensitive to noise in the data, while at the same time it should give the alarm when the fault happens. There must therefore be a trade-off between the rapidness and the accuracy of the fault detector. A trade-off between the complexity and the efficiency of the fault detection method must be consider as well (Basseville, 1985). Advances in computer science and computational capacity have opened doors for more complex models without compromising the efficiency. In recent years, more advanced methods such as neural networks, deep learning, or reinforcement learning, have all been implemented for fault detection in various fields (see e.g. Zhang et al., 2018; Newhart et al., 2019; Mandipoor et al., 2020). However, the focus in this study is not to develop or test new techniques for fault detection but rather on determining whether it is possible or not to detect faults in the sensors by using multiple RAS rate flow estimates. Therefore, a simple method described in the following chapter was used as a start. Other methods could be tested later if the approach seems feasible.

2.1 The CUSUM algorithm

A simple yet effective method to detect changes in the mean of a variable is the CUSUM algorithm (Spindler and Vanrolleghem, 2012). It was first introduced by Page (1954) and uses a log-likelihood ratio to answer if a data point belongs to a certain distribution, or if it does not. If it does not belong to the predefined distribution, the results are interpreted as a change or fault has occurred. For the application at hand, this can be done by comparing the calculated flows to a reference flow rate (the decided 'true' flow rate). The residuals (ϵ) are assumed to be normally distributed with mean μ_0 and variance σ_0^2 when no faults occur, i.e.

$$\epsilon = Q_{ref} - Q_m \sim \mathcal{N}(\mu_0, \sigma_0^2) \tag{11}$$

$$\mathcal{N}(\mu_0, \sigma_0^2) = \frac{1}{\sigma_0\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\epsilon_i - \mu_0}{\sigma_0}\right)^2} \tag{12}$$

where Q_{ref} is Q_{pump} or Q_{median} for Case 1 and Case 2 respectively, and m = {'weir', 'SMB', 'DMB','pump'}. Ideally, the four RAS flow rate models should give the same output, and thus, μ_0 should be zero when no faults occur. However, there is a small offset between the models and therefore μ_0 was calculated.

The comparison relies on the probability, or likelihood p_{θ_0}, that observation ϵ_i belongs to the distribution $\mathcal{N}(\mu_0, \sigma_0^2)$

$$p_{\theta_0}(\epsilon_i) \approx \frac{1}{\sigma_0\sqrt{2\pi}} e^{-\frac{(\epsilon_i - \mu_0)^2}{2\sigma_0^2}} \tag{13}$$

and the likelihood p_{θ_1} that it does not (i.e. a change has occurred)

$$p_{\theta_1}(\epsilon_i) \approx \frac{1}{\sigma_1\sqrt{2\pi}} e^{-\frac{(\epsilon_i-\mu_1)^2}{2\sigma_1^2}} = 1 - p_{\theta_0} \tag{14}$$

where μ_0 and σ_0 are the mean and standard deviation of the data before the change, and μ_1 and σ_1 are the mean and standard deviation of the data after the change. The log-likelihood ratio is calculated and summed up cumulatively

$$S_k = \sum_{i=1}^{k} \ln \frac{p_{\theta_1}(\epsilon_i)}{p_{\theta_0}(\epsilon_i)}. \tag{15}$$

The detection rule is defined to compare the log-likelihood ratio and its current minimum value to a defined threshold h in the following manner:

$$g_k = S_k - \min_{1\leq j\leq k} S_j \geq h. \tag{16}$$

The stopping (or alarm) time is

$$t_{stop} = \min\{k : S_k \geq \min_{1\leq j\leq k} S_j + h\}. \tag{17}$$

The threshold is in one way adaptive, as it keeps and consider all past observations. The CUSUM algorithm as presented above is designed to detect one-sided changes (Basseville and Nikiforov, 1993). As the problem at hand is to detect both increases and decreases in the mean, a two-sided CUSUM algorithm that combines an upper and a lower threshold must be used. Like the one-sided CUSUM algorithm, this method is defined by a stopping time t_{stop}

$$t_{stop} = \min\{k : g_k^+ \geq h \cup g_k^- \leq -h\} \tag{18}$$

and a detection rule (g_k^+ and g_k^-), in this case defined as

$$\begin{aligned} g_k^+ &= g_{k-1}^+ + \epsilon_k - \mu_0 - \frac{v}{2} \\ g_k^- &= g_{k-1}^- + \epsilon_k - \mu_0 + \frac{v}{2} \end{aligned} \tag{19}$$

where ϵ_k is the residual at time step k

$$\epsilon_k = Q_{ref}(k) - Q_m(k) \tag{20}$$

μ_0 is the mean before the change, and v is a predefined parameter that decides the magnitude of the faults to be detected. Q_{ref} is Q_{pump} or Q_{median} for Case 1 and Case 2 respectively, and m = {'weir', 'SMB', 'DMB','pump'}. The detection rule is thus defined so to detect changes where the mean after a change is either $\mu_1^+ \geq \mu_0 + v/2$ or $\mu_1^- \leq \mu_0 - v/2$. If the magnitude of the change is known, v is chosen accordingly. By setting the parameter very low, changes of small magnitudes will be detected but this gives many false alarms. The threshold h should be chosen to minimize the amount of false alarms while still having a short detection time.

3 Detecting sensor faults by analysing the RAS flow rate

One problem with fault detection is that we often do not know if and when faults such as slow drifts and small shifts have occurred by only analysing the data. Abrupt changes like a complete failure are relatively easy to detect both by using fault detection methods, but also based only on experience and knowledge of the process. In this study, the CUSUM algorithm will be applied to a simulated data set from a controlled environment, where all faults either are known or can be back-traced. This was done by implementing the fault generation model presented in next chapter. Chapter 3.2 describes how the fault detection method was implemented and evaluated.

3.1 Sensor fault generation

In wastewater modelling, traditionally ideal sensors have been used. Rieger et al. (2003) characterized sensors and classified them depending on their response time and measuring interval. They suggested sensor models that account for response time, noise, drift, and calibration and cleaning intervals. Alex et al. (2003) followed up evaluating how the sensor characteristics and behaviour impact the control result and concluded that sensor models are important to achieve more realistic results. Rosén et al. (2008) suggested a statistical framework to account for seven different types of faults that occur in sensors in WWTPs. Its foundation lies in a Markov chain that describes the different possibilities of a sensor to be in a specific state.

To some extent, all sensors are subject to failure, but the type of failure, how often it happens and how long it takes to repair or replace a faulty sensor differs between WWTPs. The states of which the sensor can enter can be classified in seven categories (Rosén et al., 2008):

- Fully functional - no faults
- Drift - the sensor starts drifting with an increased offset (positive or negative) from the true value
- Shift - the sensor has a constant bias
- Fixed value - the sensor gives a constant value
- Complete failure - the sensor stops working
- Wrong gain - the sensor is badly calibrated outside a certain concentration interval
- Calibration - the faulty sensor is taken out of operation to be calibrated

A Markov chain consist of states between which the system can go based on defined probabilities. The Markov chain has no memory, i.e. the history of the system is stored in the present state. The faults are represented by a state that the sensor can be in and go between (Figure 4). By assigning different probabilities for the system to go between the states, more realistic simulated sensor data can be retrieved from the WWTP model (Rosén et al., 2008).

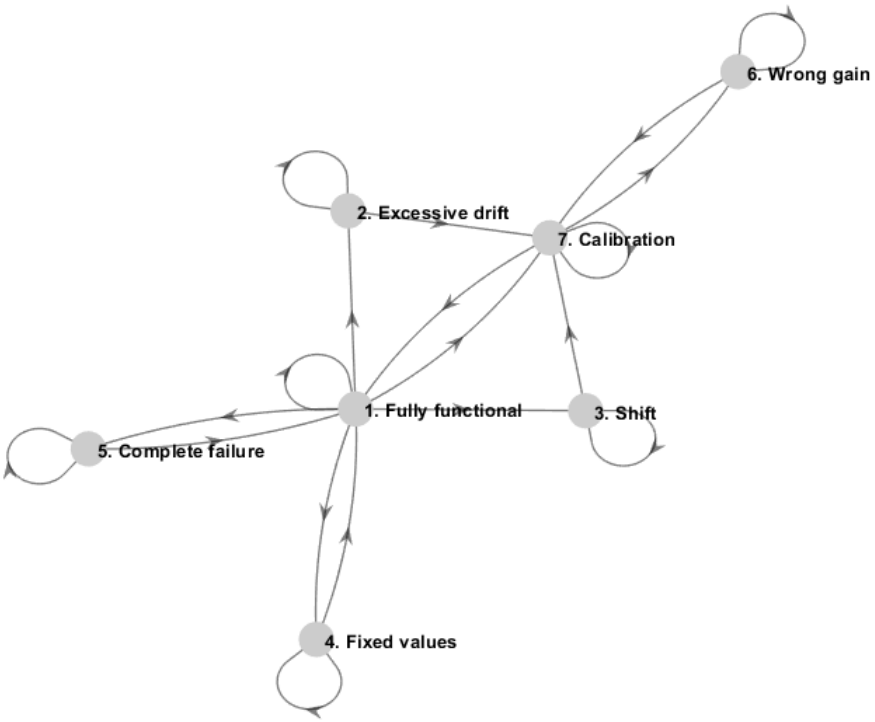

Figure 4. Markov chain for the sensor with the seven states: (1) Fully functional, (2) Drift, (3) Shift, (4) Fixed value, (5) Complete failure, (6) Wrong gain, and (7) Calibration. The figure is adapted from Rosén et al. (2008).

The sensors of interest in this application are flow meters, level indicators, and suspended solids sensors which are all type A sensors according to Rieger et al. (2003). The fault generation model was added to a MBR model of Henriksdal WWTP (based on the Benchmark Simulation Model no. 2 (BSM2) and its underlying models with minor modifications). The input to the model included a probability matrix with the probabilities to go between states in the Markov chain, and a fault vector to match the Simulink model (Table 3). In addition to the faults in Table 3 white noise was added so that the signal had realistic disturbances even when fully functional.

Out of the seven states (Table 3), five were considered in this study: *fully functional*, *drift*, *shift*, *complete failure*, and *calibration*. The parameters in Table 3, the probabilities in the probability matrix, as well as the variance of the white noise were chosen individually for each sensor type to mimic their real life behaviour (Table 4).

Table 3. States, fault vectors and parameters.

States	Fault vector	Parameters
Fully functional	$\begin{bmatrix}1 & 0 & 1 & 0\end{bmatrix}$	-
Excessive drift	$\begin{bmatrix}1 & (t-t_0)f_r & 0 & 0\end{bmatrix}$	t = current time t_0 = start of drift event f_r = drift rate
Shift	$\begin{bmatrix}1 & f_b & 1 & 0\end{bmatrix}$	f_b = magnitude of shift bias
Complete failure	$\begin{bmatrix}0 & 0 & 0 & 0\end{bmatrix}$	-
Calibration	$\begin{bmatrix}0 & 0 & 0 & 0\end{bmatrix}$	-

Table 4. Variance of white noise and parameters in the fault vectors for each of the sensors.

Sensor	Variance of white noise	Drift rate, f_r	Shift magnitude f_b
Q_{bio}	0.032 m^3/s	0.064	0.16
h_{bio}	0.00091 m	0.0018	0.0046
SS_{bio}	0.018 kg/m^3	0.036	0.09
SS_{RAS}	0.036 kg/m^3	0.072	0.18
h_{RAS}	0.032 m	0.064	0.16

The different fault types were simulated one at a time for each sensor. Figure 5 illustrates how the sensors behave at each fault failures. Note that the complete failure and the calibration will give the same effect on the output of the sensor. To examine how the different flow estimates responded to each type of fault, one sensor with one fault type was evaluated at a time. This was done prior to the implementation of the fault detection algorithm to get a first indication on whether or not it would be possible to differentiate between the fault and sensors types. A brief summary of the results and some conclusions are presented in the following sections.

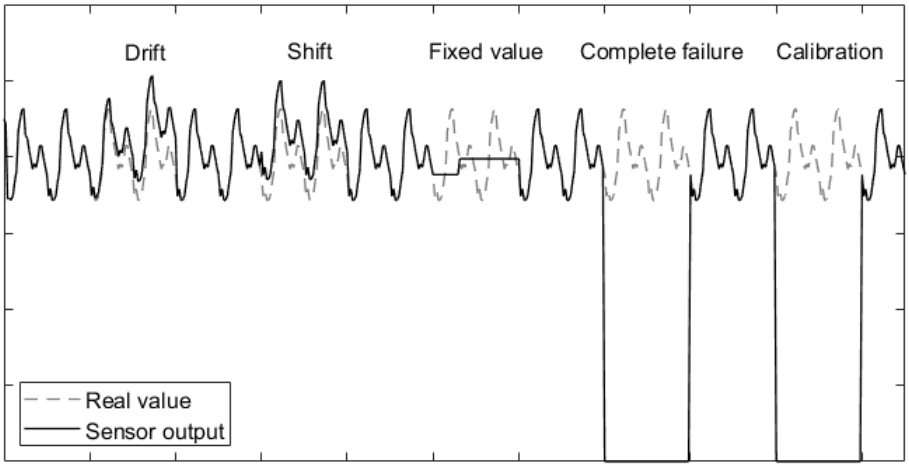

Figure 5. An illustration of different types of faults.

3.1.1 Level indicators

All sensor faults in the level indicator in the bioreactor impacts Q_{weir} which is not surprising as it is the basis of the calculation. The drift and shift events, which in both cases resulted in an overestimated water level, led to and overestimation of the flow. The complete failure event caused an underestimation of the flow, meaning that there is a direct relation between the water level and the flow rate.

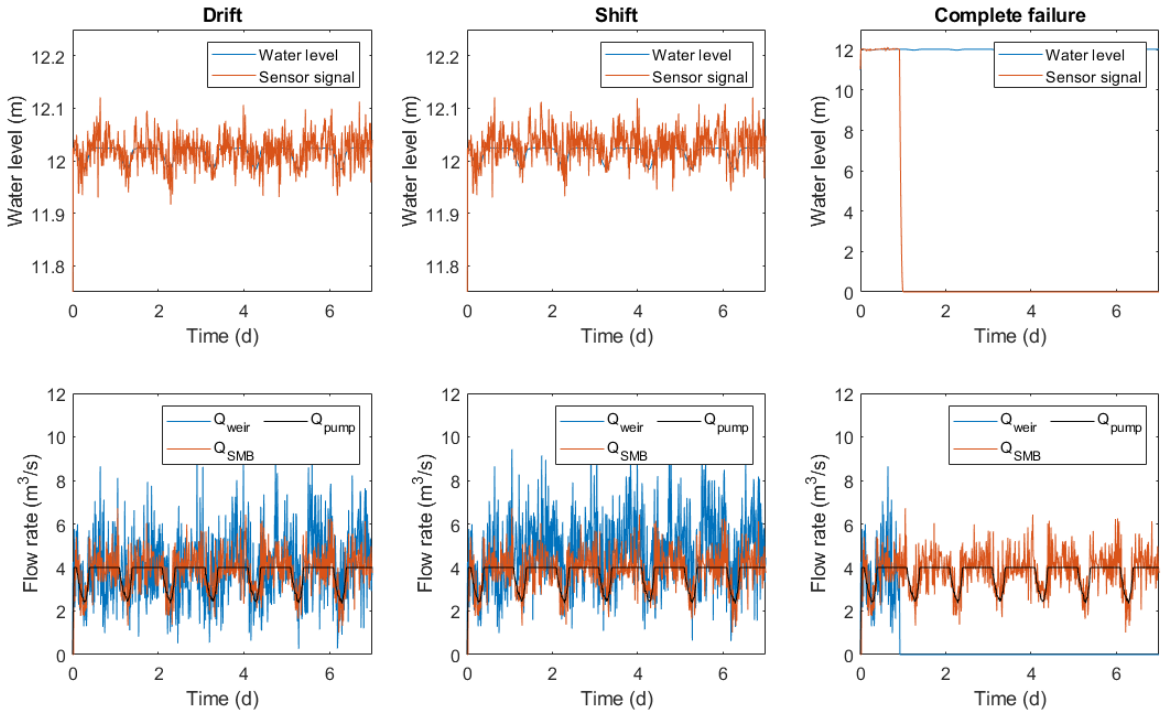

Figure 6. The water level and the sensor signal for different types of faults.

The static mass balance model is independent of the level over the weir and was therefore not affected at all. The pump model was not affected by faults in h_{bio}. It was however

affected by adding faults in the level indicator in the RAS tank (h_{RAS}). This measurement is only used in the pump model and there for it was the only estimate that was affected.

3.1.2 Flow meter

Faults in the flow meter in to the bioreactor (Q_{bio}) affect both the mass balance model and the weir model. The mass balance have a direct relation to faults in the flow meter, meaning that an increase in the flow rate also leads to an increase in the RAS flow rate estimate. It was the opposite for Q_{weir}, where an increase in the flow rate causes a decrease in the estimated flow rate, yet the effect was rather small. The pump model was unaffected as it do not account for the incoming flow rate to the bioreactor.

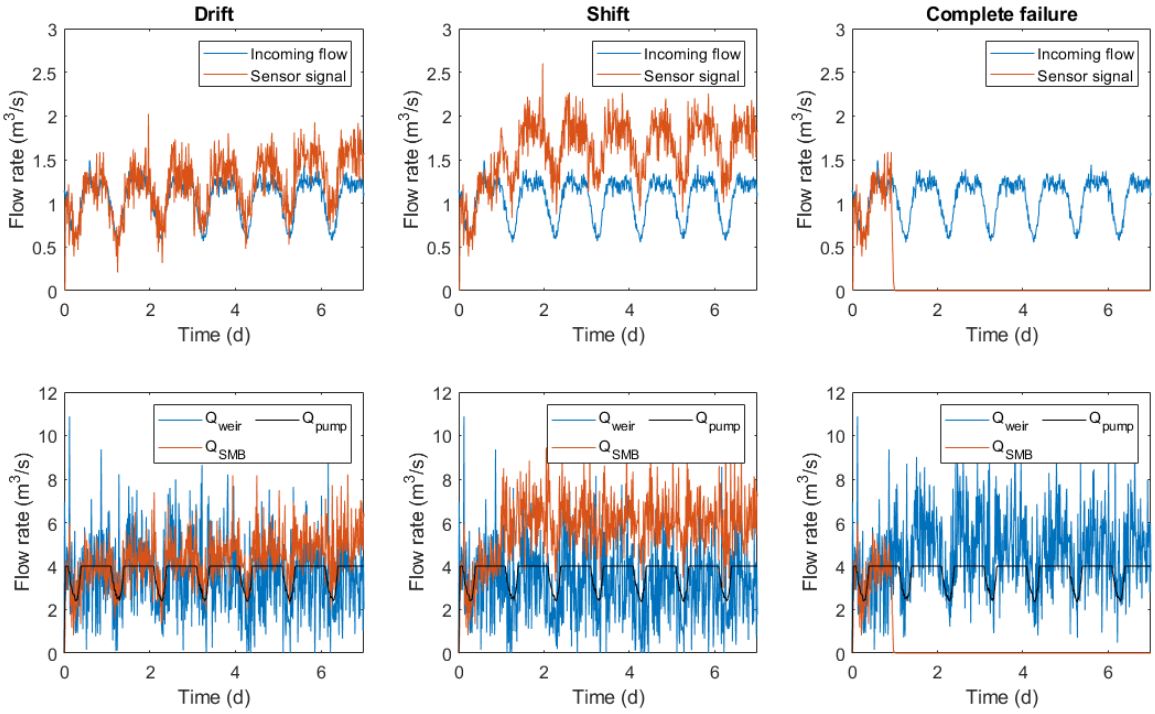

Figure 7. The incoming flow rate and the sensor signal for different types of faults.

3.1.3 Suspended solids

The faults in the suspended solids sensors (SS_{bio} and SS_{RAS}) affect only the mass balance model. The faults in the SS_{bio} sensor has a direct relation to the flow rate estimate, whereas the SS_{RAS} sensor has a reversed relation, meaning that an increase in the SS_{RAS} concentration causes a decrease in the RAS flow rate. Looking at the SS_{bio} sensor, the drift event has a big impact on the flow rate estimate. The increased SS_{bio} causes a rapid increase in the flow rate. The shift event is not as obvious but still noticeable (Figure 8). The faults in the SS_{RAS} sensor are not as prominent in the flow rate estimates as the faults in the SS_{bio} sensor, yet still visually detectable (Figure 9). Faults in the suspended solids sensors do not affect the weir or pump model output.

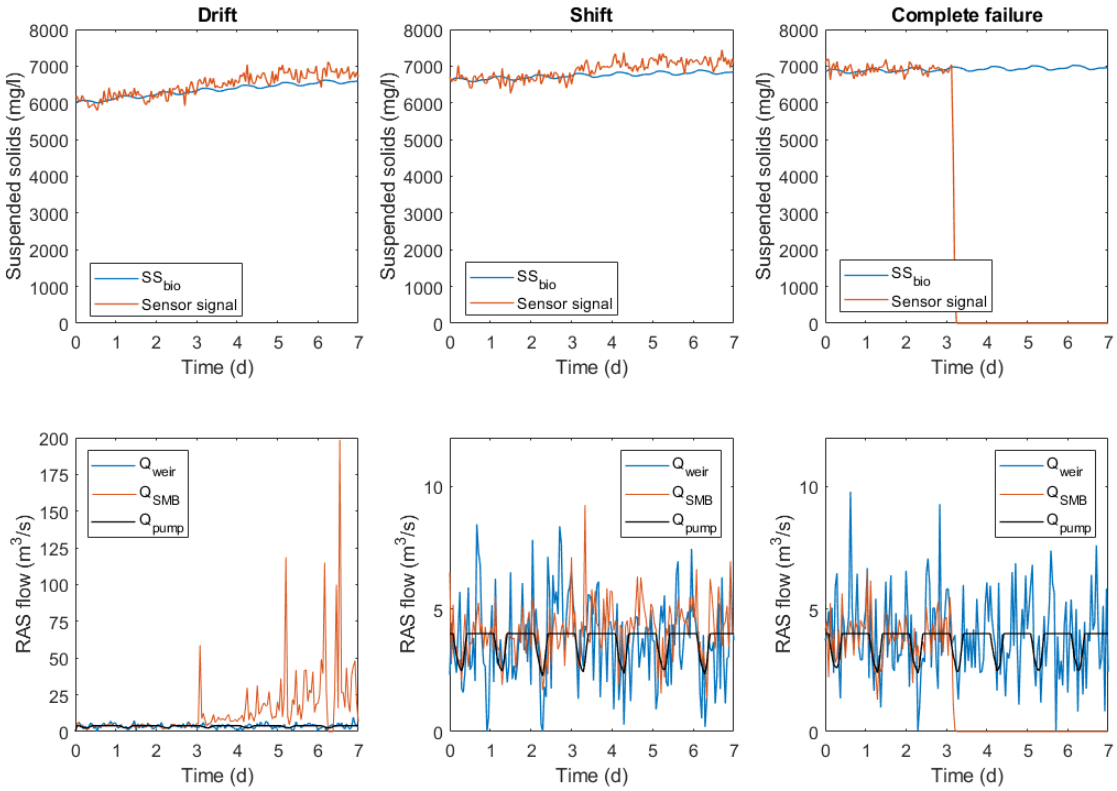

Figure 8. The SS_{bio} and the sensor signal for different types of faults.

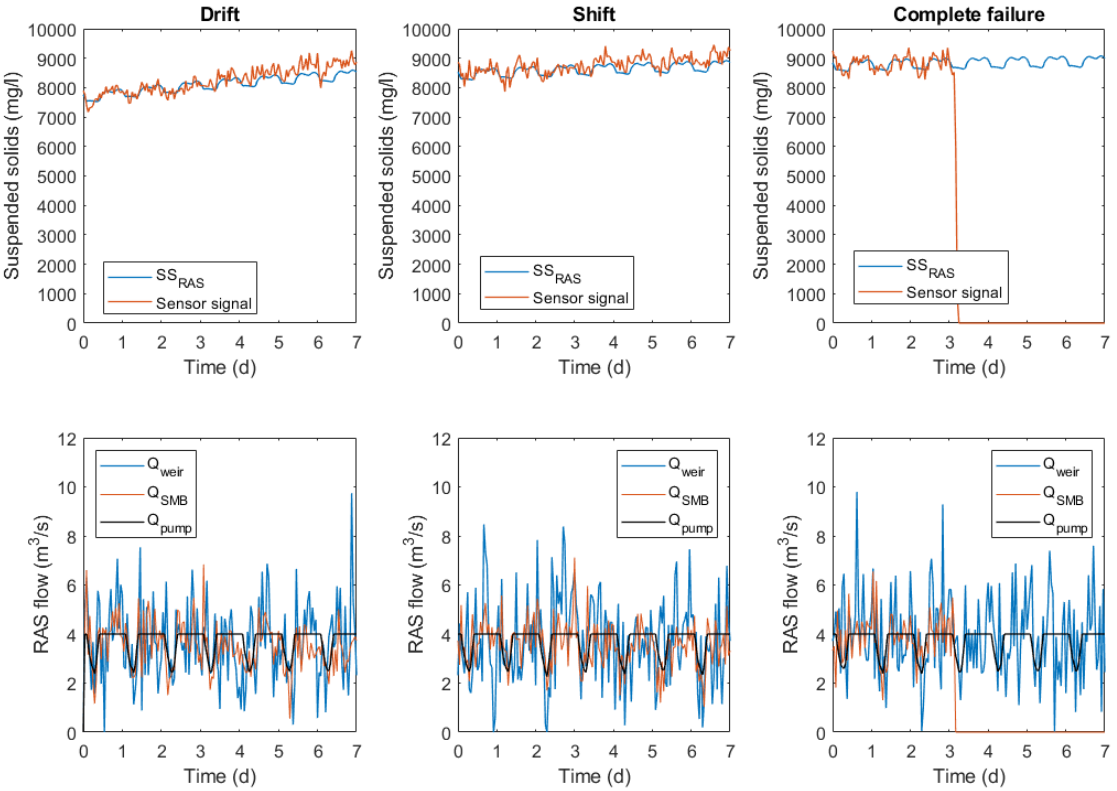

Figure 9. The SS_{RAS} and the sensor signal for different types of faults.

3.1.4 Preliminary conclusions from fault generator integration

Based on the results from the initial test it can be concluded that it, theoretically, will be possible to not only detect the faults but also isolate the faults (Table 5). A fault in Q_{bio} will affect all flow rate estimates except the pump model. Faults in the suspended solids sensors will only affect the mass balance model. Faults in the level indicators (h_{bio} or h_{RAS}) will only affect Q_{weir} or Q_{pump}.

Table 5. The different models and what sensors they are affected by. (-) indicates a reversed relation, i.e. if the input increases, the output decreases. (+) indicates a direct relation, i.e. if the input increases, so does the output.

Sensor / Model	Q_{bio}	h_{bio}	SS_{bio}	SS_{RAS}	h_{RAS}
Q_{weir}	(-)	(+)			
Q_{SMB}	(+)		(+)	(-)	
Q_{pump}					(+)

The two cases proposed assumes that either the pump model or the median of the four calculations is the true RAS flow. These cases seems like a good starting point based on the initial investigations.

3.2 Fault detection

The CUSUM control chart exists as a built-in function in Matlab (see MathWorks) but was implemented by hand to gain more insight and control of the results. Three different sets of simulated data were used:

1. the output of the four RAS flow rate models with disturbances added to all input signals,
2. the output of the four RAS flow rate models with disturbances added to all input signals and known faults (time, magnitude, type) added to one input signal/sensor at a time, and
3. the output of the four RAS flow rate models with disturbances added to all input signals and unknown faults (time, magnitude, type) added to one or several input signals.

Two cases were investigated, where it was assumed that (1) the pump model gives the true RAS flow rate, and (2) the median of the four calculations gives the true flow rate. Based on the conclusions drawn in Chapter 3.1.4 it should be possible to not only detect the faults, but by analyzing the flows and the detection functions also determine which sensor is faulty.

3.2.1 Case 1: Q_{RAS} from pump model

In the first case, the pump model output was used as the reference flow rate. The calculated flow rates from the mass balance and the weir model were compared to the reference flow rate. Initially the residuals when only white noise was added were analyzed to define thresholds h and detection limits v for each flow rate estimate. The distributions of the residuals are slightly skewed, but were assumed to be normally distributed (Figure 10).

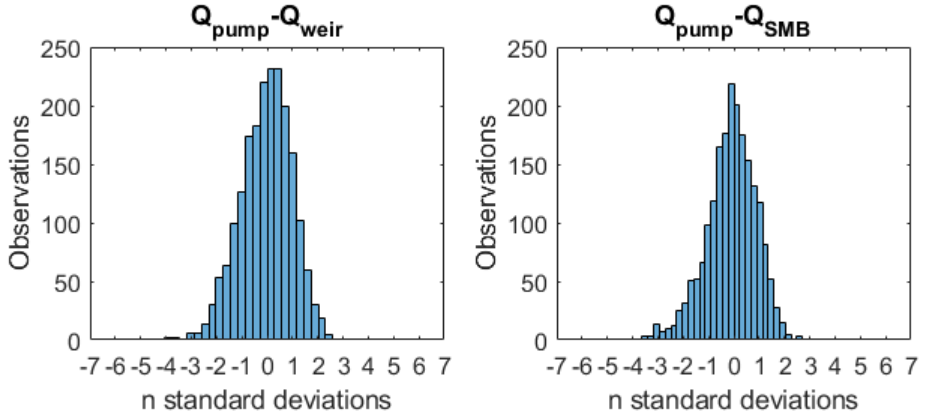

Figure 10. Histogram of the residuals of the two flow rate estimates Q_{weir} and Q_{SMB} in relation to Q_{pump}.

Thresholds were decided so that all 'normal' operating conditions (only white noise) would fall within the thresholds (Figure 10). v was chosen as 2σ so to detect changes that deviate from the mean by one standard deviation (Table 6).

Table 6. Mean and standard deviation of the residuals, with decided thresholds and magnitude of change.

Flow	Residuals		Threshold	Magnitude
	μ_0	σ_0	h	v
Q_{weir}	-0.0630	1.73	$\pm 7\sigma_{0,weir}$	$2\sigma_{0,weir}$
Q_{SMB}	-0.0981	0.884	$\pm 7\sigma_{0,SMB}$	$2\sigma_{0,SMB}$

Simulations where known faults were introduced to each sensor at a time was run. As an example, results from detecting a drift in the h_{bio} sensor (Figure 11) is shown but the method was applied to all sensors and all faults. Figure 12 shows the detection function g_k and the threshold h for each method. A fault is detected when $g_k^- < -h$ (or $g_k^+ > h$) – in this case at $t_{stop} = 12.9$ d

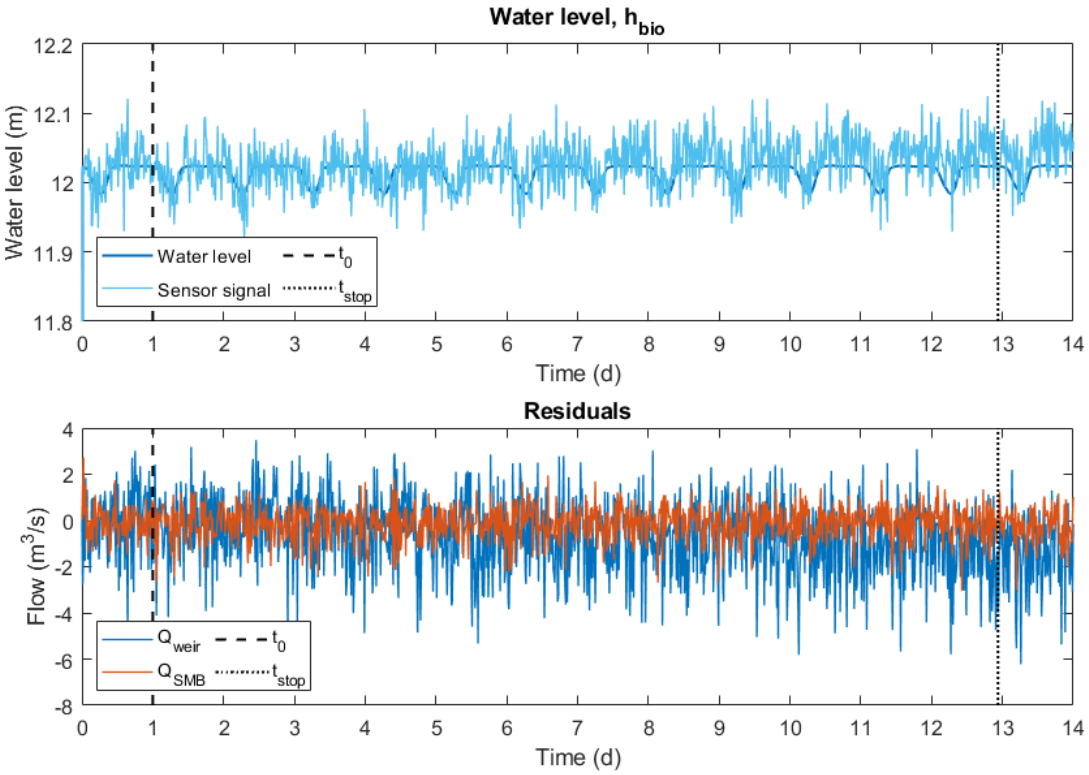

Figure 11. Water level and sensor signal, as well as the residuals, with the start (t_0) and alarm time (t_{stop}) of the drift event marked.

The start time of the fault event, t_0 = 1 day for all types of faults. The algorithm was tested to one fault and sensor at a time and t_{stop} was noted. The results are presented as the detection delay $\Delta t = t_{stop} - t_0$ (Table 7). The shortest delays were recorded for the complete failures in all sensors. Except that, the shift event in the SS_{RAS} sensor was fastest to detect. The delay time was very long for the drift events in the h_{bio} and h_{RAS} sensors. No false alarms were recorded.

Table 7. The detection delay time $\Delta t = t_{stop} - t_0$ for each fault type and sensor.

	Drift		Shift		Complete failure	
Sensor	Δt	Detected from	Δt	Detected from	Δt	Detected from
h_{bio}	11.2	Weir	2.92	Weir	0.09	Weir
Q_{bio}	1.22	Mass balance	0.11	Mass balance	0.08	Weir
SS_{bio}	1.22	Mass balance	0.13	Mass balance	0.03	Mass balance
SS_{RAS}	5.83	Mass balance	0.10	Mass balance	0.02	Mass balance
h_{RAS}	19.4	Mass balance	4.39	Mass balance	0.34	Mass balance

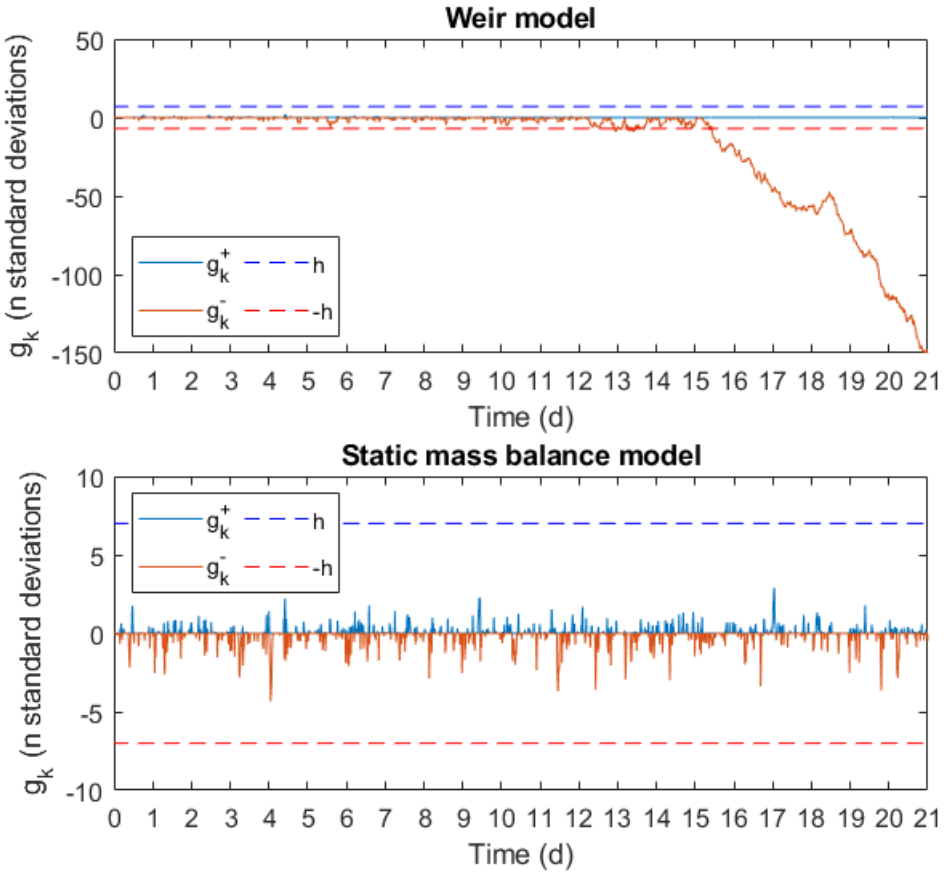

Figure 12. The decision function g_k with thresholds h shown for the three different flow estimates. The fault is detected when the decision function for the weir goes below $-h$. The y-axes are scaled based on each methods residuals standard deviation.

A simulation with unknown fault events (type and number of faults) at unknown times was run. The faults were generated randomly in line with the theory presented in Chapter 3.1. The algorithm detected a fault from the mass balances at t_{stop} = 6.15 d (Figure 13). After examining the data, it was concluded that the fault event started at t_0 = 4.40 d and that it was a drift in the SS_{bio} sensor. The peak in g_k^+ for the static and dynamic mass balance model are explained by an abrupt change in the estimates, due to a complete failure of the SS_{bio} sensor. No other faults were detected.

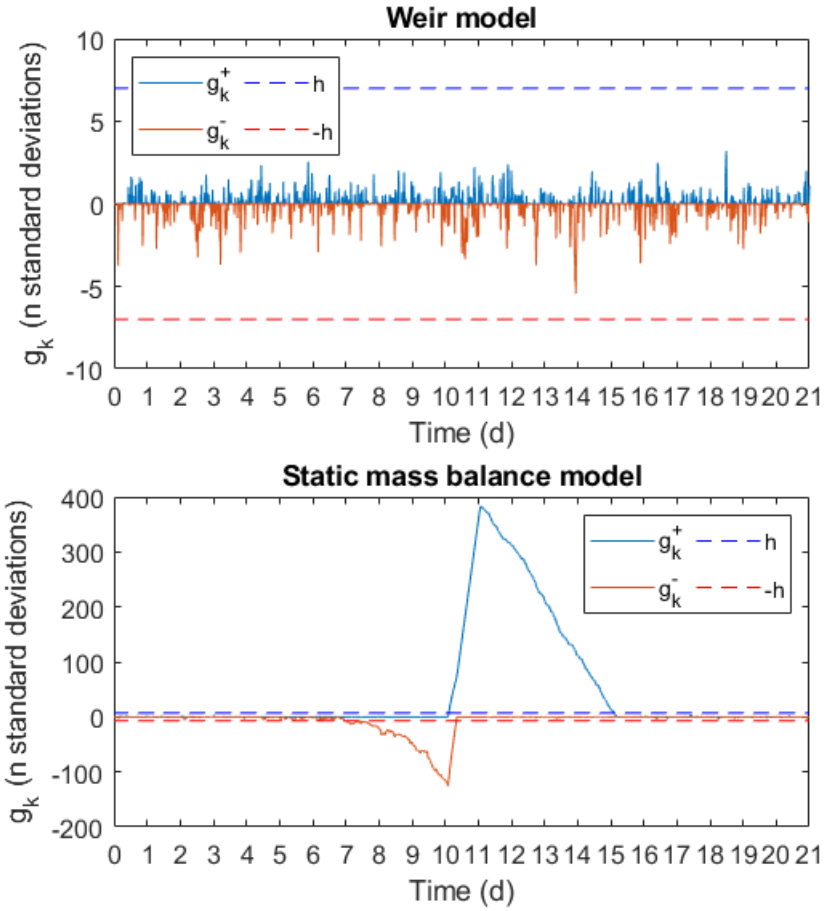

Figure 13. The detection function for the three different flow estimates. A fault in the SS_{bio} sensor can be seen at t = 6.15 d. The y-axes are scaled based on each methods residuals standard deviation.

3.2.2 Case 2: Q_{RAS} from median

In Case 2, the median of the three model outputs was used as the reference flow rate. The calculated flow rates from the mass balance model, the weir, and the pump model were compared to the median. The residuals when only white noise was added were analysed to define thresholds h and detection limits v for each flow rate estimate. The distributions of some of the residuals were slightly skewed but all were assumed to be normally distributed (Figure 14).

Thresholds were decided so that all 'normal' operating conditions (only white noise) would fall within the thresholds (Figure 14). v was chosen as 2σ so to detect changes that deviate from the mean by one standard deviation (Table 8).

Table 8. Mean and standard deviation of the residuals, with decided thresholds and magnitude of change.

Flow	Residuals		Threshold	
	μ_0	σ_0	h	v
Q_{weir}	-0.0334	1.22	$\pm 6\sigma_{0,weir}$	$2\sigma_{0,weir}$
Q_{SMB}	-0.0490	0.482	$\pm 10\sigma_{0,SMB}$	$2\sigma_{0,SMB}$
Q_{pump}	-0.0554	0.505	$\pm 7\sigma_{0,pump}$	$2\sigma_{0,pump}$

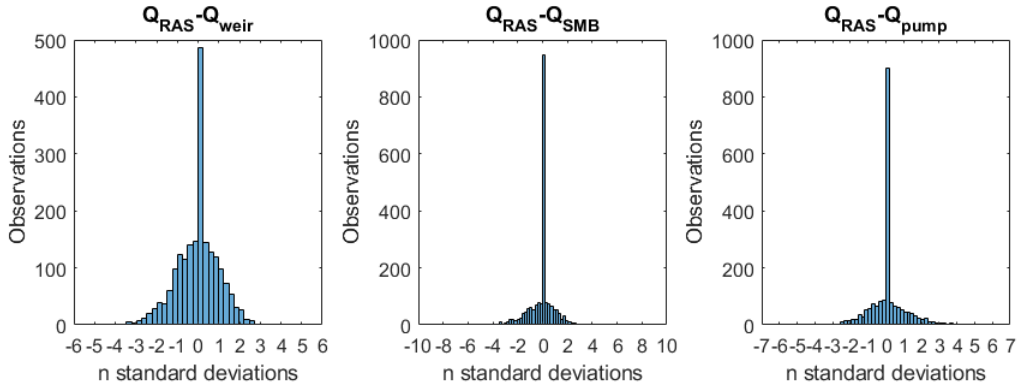

Figure 14. Histogram of the residuals of the three flow rate estimates Q_{weir}, Q_{SMB}, and Q_{pump} in relation to the median flow rate Q_{RAS}.

Simulations where known faults were added to each sensor at a time was run. As an example, results from detecting a shift in the Q_{bio} sensor is shown (Figure 15) but the method was applied to all sensors and all fault types. The value of the detection function (g_k^+ and g_k^-) is shown in Figure 16. The fault in the Q_{bio} sensor is detected at $t = 1.11$ d.

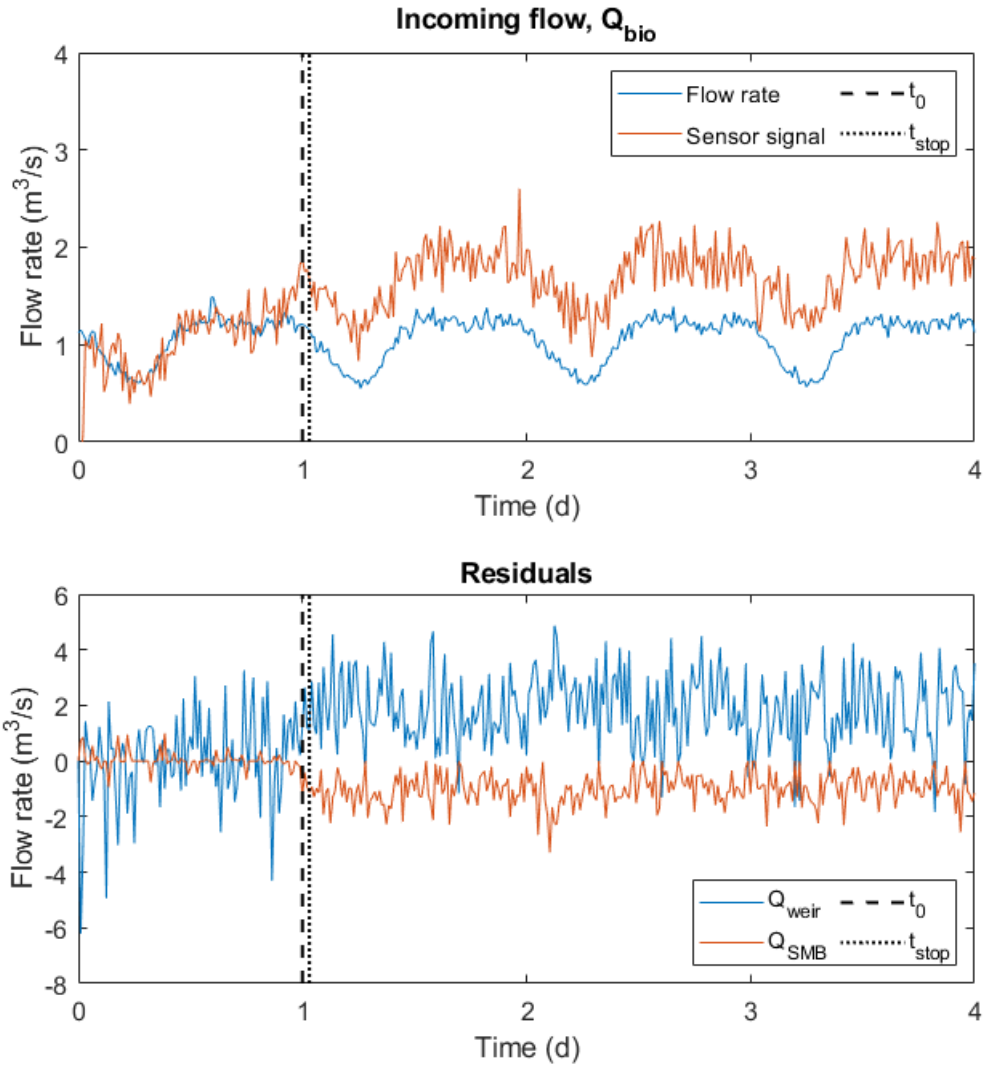

Figure 15. The incoming flow rate Q_{bio} and the sensor signal, with the start (t_0) and alarm time (t_{stop}) of the drift marked. The residuals and the start and alarm time are shown in the bottom figure.

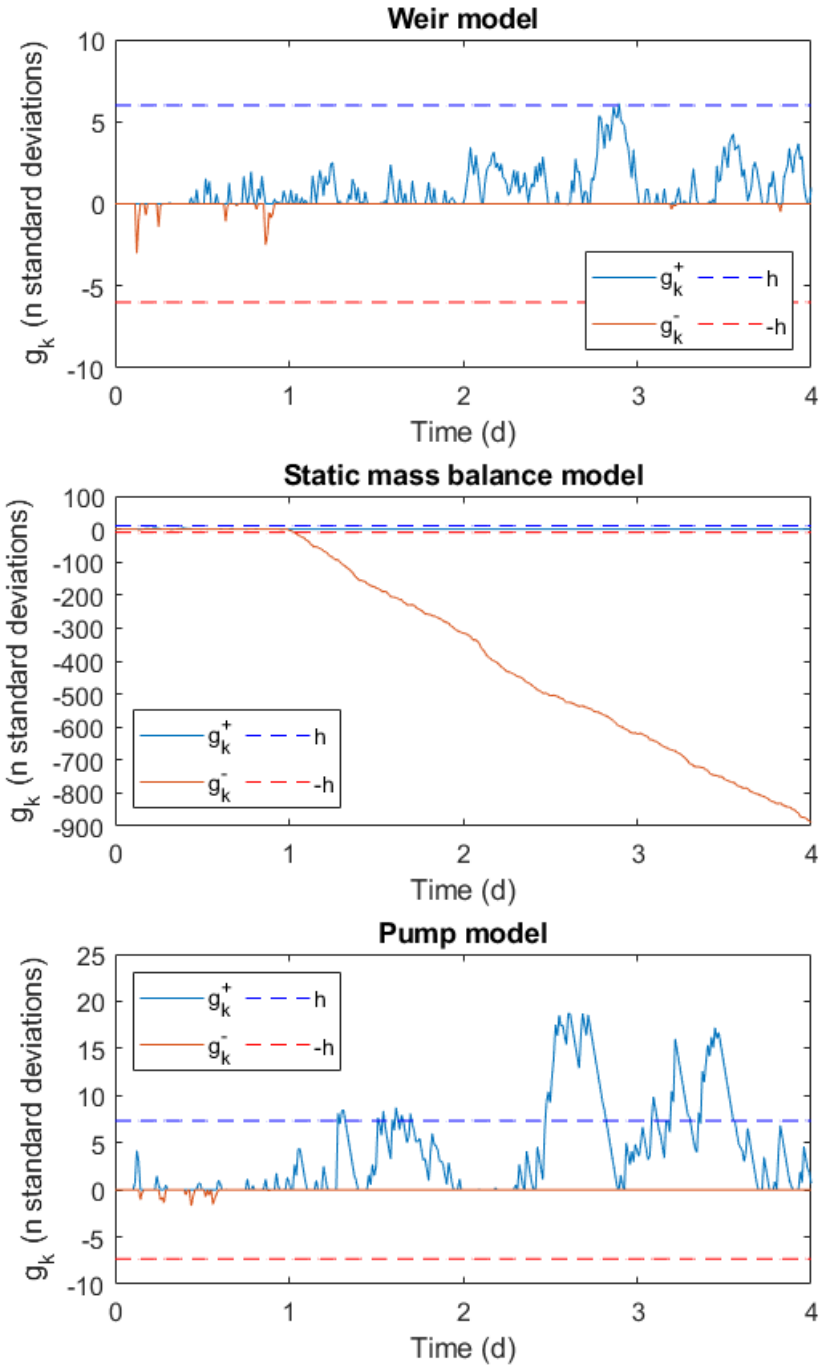

Figure 16. The detection function g_k with thresholds h shown for the three flow rate estimates. The fault is detected when the detection functions are less than $-h$, or exceeds h. The y-axes are scaled based on each methods residuals standard deviation.

The algorithm was tested to one fault and sensor at a time and the detection delay was calculated, with t_0 = 1 day (Table 7). The fastest responses was to detect complete failures in all sensors. Apart from that, the algorithm was quite fast to detect shifts in all sensors, especially Q_{bio}, SS_{bio} and SS_{RAS}. Faults in h_{RAS} were detected, but the threshold only was exceeded temporarily.

Table 9. The detection delay time $\Delta t = t_{stop} - t_0$ for each fault type and sensor. The thresholds were only temporarily exceeded when it comes to the drift and shift in the h_{RAS} sensor, which is why the numbers are in brackets.

	Drift		Shift		Complete failure	
Sensor	Δt	Detected from	Δt	Detected from	Δt	Detected from
h_{bio}	4.48	Weir	2.72	Weir	0.0279	Weir
Q_{bio}	2.13	Mass balance	0.111	Mass balance	0.0383	Weir + Mass bal.
SS_{bio}	1.22	Mass balance	0.143	Mass bal. + pump	0.0279	Mass bal. + pump
SS_{RAS}	7.75	Mass bal. + pump	0.163	Mass bal. + pump	0.0175	Mass bal. + pump
h_{RAS}	(8.65)		(6.42)		1.27	Pump

The fault generation model as described in Chapter 3.1 was used to generate a data set with unknown faults in the five sensors. A fault was detected in the Q_{weir} RAS flow estimate at $t_{stop} = 5.03$ d (Figures 17 and 18). By analysing the detection functions for each flow estimate it was concluded that it was only Q_{weir} that was affected by this first fault, leading to the conclusion that the h_{bio} sensor was faulty. Later in the time series there is a fault that was detected in all models at $t_{stop} \approx 8$ d (Figure 17). After analysing the input data it was concluded the fault was a drift in the SS_{RAS}. Since the mass balance model affect the median (reference RAS flow rate), this fault is detected from the pump model as well as the mass balance model.

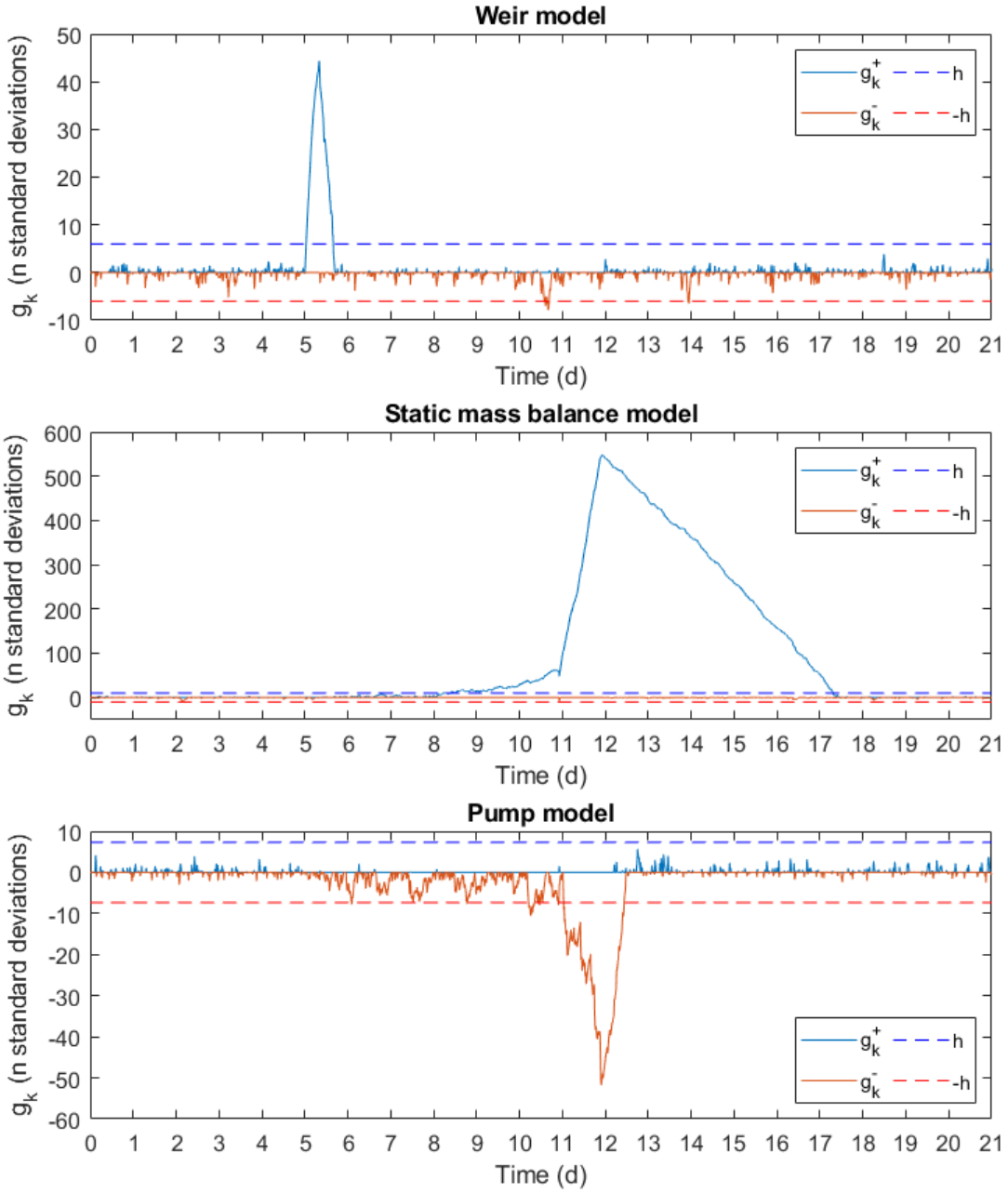

Figure 17. The detection function g_k with thresholds h shown for the three flow rate estimates. The fault is detected as g_k^+ for the weir goes exceeds h. The y-axes are scaled based on each methods residuals standard deviation.

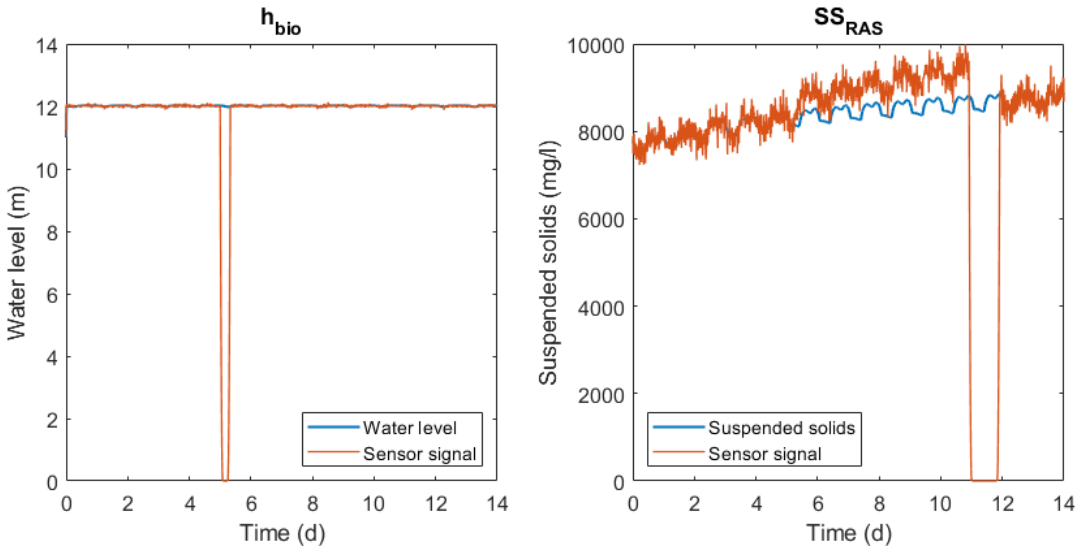

Figure 18. The two faulty sensors h_{bio} and SS_{RAS}. The first fault is detected at $t_{stop} = 8.67$d and the second fault was detected at $t_{stop} \approx 16$d

4 Discussion and conclusion

The CUSUM control chart was implemented and tested on three types of faults in five different sensors. The generated sensor faults were *drift*, *shift*, and *complete failure*. Two cases were investigated where (1) the pump model was assumed to give the true RAS flow rate, and (2) where the median of all four estimates was assumed to give the true RAS flow rate. The aim was to evaluate if it would be possible to detect faults in sensors based on the residuals from the different RAS flow rate estimates.

The algorithm was successful in detecting faults in all of the five sensors used, but for some of them the detection delay was quite long. Although not included in this study, it is possible that the detection delay is smaller when using the flow rate estimates to detect changes and faults instead of the sensor value itself. The faults in the sensors multiply when used in the soft sensor, as is quite obvious when visually examining e.g. Figures 7 and 8. The shift was detected very fast in both cases ($<$ 1 d for most sensors) even though the residuals were only about 1-5% of the simulated true water level and SS_{bio} concentration. The SS_{RAS} measurement has less influence on the flow rate estimates from the mass balance model than the SS_{bio} sensor, which results in a longer detection delay for faults in the SS_{RAS} sensor. One could decrease the detection delay by setting a stricter threshold h. However, since the detection time was very short for faults in the SS_{bio} sensor, it might be unwise to do so as it possibly could give many false alarms. One improvement could be to lower the threshold and at the same time formulate a criterion that says either that the fault detector must detect a given number of consecutive faults before sending an alarm, and possibly also that faults must be detected from more than one flow rate estimate before the alarm.

The method allowed to visually determine, both from the CUSUM plots and the flow rates, which sensor was failing when analysing data with unknown faults. Based on the

conclusions drawn in Chapter 3.1.4 (Table 5) it might be possible to further develop the fault detector so that it automatically can differentiate between type of fault and in which sensor. The CUSUM method is suitable for univariate faults, i.e. to detect one fault in one sensor at a time. To be able to differentiate between types of faults one would need to make several comparisons at once, a different method than the CUSUM algorithm should then be used. A machine learning method suitable for classification problems, like k-Nearest Neighbour or a tree-based method, could be a good start.

Case 1, where the pump model was used as the reference RAS flow rate, had in general longer detection delay than what was achieved in Case 2. In Case 2, the soft sensor models are compared to the median of the three model outputs, and thus, the reference RAS flow rate is dependent on the model outputs. This can trigger alarms from several models which in turn will make the fault detector faster, but at the same time makes it difficult to isolate the faults. An alternative way to proceed could be to create an algorithm that compares all flow rate estimates to each other and thereby not deciding which flow rate is true and still being able to automatically determine which sensor is faulty. The algorithm could be designed so that it detects faults in each set of residuals, compares the times of fault detection (an interval would be needed), and make decisions based on which set or sets of residuals are deviating.

An interesting extension would be to apply this method to detect faults in historic plant data as a first step towards implementing it at a real plant. As discussed previously, it is probably valuable to test other fault detection methods prior to such an extension.

References

Alex, J., Rieger, L., Winkler, S., and Siegrist, H. (2003). Progress in sensor technology – progress in process control? Part II: Results from a simulation benchmark study. *Water Science and Technology*, vol. 47(2):pp. 113–120.

Basseville, M. (1985). *On-line detection of jumps in mean*. In: Basseville M., Benveniste A. (eds) Detection of Abrupt Changes in Signals and Dynamical Systems. Lecture Notes in Control and Information Sciences, vol. 77. Springer, Berlin, Heidelberg.

Basseville, M. and Nikiforov, I. (1993). *Changes in the Scalar Parameter of an Independent Sequence*. In Detection of Abrupt Changes: Theory and Application. Prentice-Hall, Englewood Cliffs, N.J., USA. ISBN: 0-13-126780-9.

Blomstrand, P. and Jemander, R. (2017). *Systemteknisk studie av pumpstyrning på Henriksdals nya reningsverk*, Uppsala university. Department of Information Technology. Accessible: http://www.w-program.nu/filer/exjobb/Patrik_Blomstrand_och_Rasmus_Jemander.pdf.

Hammer, M. J. S. and Hammer, M. J. J. (2014). *Water and Wastewater Technology*. Pearson Education Limited, Essex, 7th edition.

Karassik, I., Messina, J., Cooper, P., and Heald, C. (2007). *Pump Handbook*. McGraw-Hill Education, New York, USA.

Mandipoor, B., Majd, M., Sheikhalishahi, S., Modena, C., and Osmani, V. (2020). Monitoring and detecting faults in wastewater treatment plants using deep learning. *Environmental Monitoring Assessement*, vol. 192(148).

MathWorks (no date). *cusum*, Accessible: https://se.mathworks.com/help/signal/ref/cusum.html [2021-10-06].

Newhart, K. B., Holloway, R. W., Hering, A. S., and Cath, T. Y. (2019). Data-driven performance analyses of wastewater treatment plants: A review. *Water Research*, vol. 157:pp. 498–513.

Page, E. (1954). Continuous Inspection Schemes. *Biometrika*, vol. 41(1):pp. 100–115.

Persson, J., Fridell, K., Gustafsson, E.-L., and Englund, J.-E. (2014). *Att räkna på vatten - en formelsamling för landskapsingenjörer*. (Report 2014:17). Alnarp, Sweden: Swedish University of Agricultural Sciences.

Rieger, L., Alex, J., Winkler, S., Boehler, M., Thomann, M., and Siegrist, H. (2003). Progress in sensor technology – progress in process control? Part I: Sensor property investigation and classification. *Water Science and Technology*, vol. 47(2):pp. 103–112.

Rosén, C., Rieger, L., Jeppsson, U., and Vanrolleghem, P. (2008). Adding realism to simulated sensors and actuators. *Water Science and Technology*, vol. 57(3):pp. 337–344.

Saagi, R., Flores-Alsina, X., Butler, D., Gernaey, K., and Jeppsson, U. (2016). Catchment & sewer network simulation model to benchmark control strategies within urban wastewater systems. *Enviromental Modelling & Software*, vol. 78:pp. 16–30.

Samuelsson, O. (2021). *Sensor Fault Detection and Process Monitoring in Water Resource Recovery Facilities*. Uppsala Dissertations from the Faculty of Science and Technology, Uppsala: Acta Universitatis Upsaliensis. ISBN 978-91-513-1116-6.

Spindler, A. and Vanrolleghem, P. A. (2012). Dynamic mass balancing for wastewater treatment data quality control using CUSUM charts. *Water Science and Technology*, vol. 65(12):pp. 2148–2153.

Stockholm Vatten och Avfall (2017). *Tekniskt nedslag i Stockholms framtida avloppsrening*, Accessible: http://www.stockholmvattenochavfall.se/globalassets/sfa/pdf/informationsbroschyrer/tekniskt-nedslag-i-stockholms-framtida-avloppsrening.

Zhang, D., Lin, Z., and Gao, Z. (2018). A Novel Fault Detection with Minimizing the Noise-Signal Ratio Using Reinforcement Learning. *Sensors*, vol. 18(3087).