  Water Science & Technology—WST | 57.3 | 2008

Adding realism to simulated sensors and actuators

C. Rosen, L. Rieger, U. Jeppsson and P. A. Vanrolleghem

ABSTRACT

In this paper, we propose a statistical theoretical framework for incorporation of sensor and actuator faults in dynamic simulations of wastewater treatment operation. Sensor and actuator faults and failures are often neglected in simulations for control strategy development and testing, although it is well known that they represent a significant obstacle for realising control at full-scale facilities. The framework for incorporating faults and failures is based on Markov chains and displays the appealing property of easy transition of sensor and actuator history into a model for fault generation. The paper briefly describes Markov theory and how this is used together with models for sensor and actuator dynamics to achieve a realistic simulation of measurements and actuators.

Key words | actuators, benchmark model, Markov chains, modelling, sensors, wastewater

C. Rosen
U. Jeppsson
Department of Industrial Electrical Engineering and Automation,
Lund University,
Box 118, SE-221 00 Lund,
Sweden
E-mail: *christian.rosen@iea.lth.se*

L. Rieger
P. A. Vanrolleghem
model*EAU*, Département de génie civil,
Pavillon Pouliot,
Université Laval,
Québec, QC, G1K 7P4,
Canada

INTRODUCTION

The interest in using dynamic simulation models to test, verify and benchmark control strategies in wastewater treatment operation has put focus on the realism of the models used. Concerns about whether the models represent the true behaviour of the process have been raised as long as there have been models around. However, these concerns have mostly aimed at the process description and the models' ability to correctly describe the physical, chemical and biological mechanisms of the process. When control aspects are studied in a simulation environment, other issues related to realism also become important. In control, good online measurements are vital. In a simulation environment, this does not pose a problem but in real applications, the quality of the online measurement is perhaps the largest obstacle between failure and success. Although most researchers and practitioners agree with this statement, surprisingly little attention has been given the task to describe sensors and actuators in a realistic manner (with some exceptions). Rieger *et al.* (2003) proposed classification of sensors of different types depending on the measurement mechanism and also provided models for the different classes. Some reports on models of actuators exist, mainly focusing on the aeration system (Alex *et al.* 2002; Rieger *et al.* 2006). However, not many reports can be found where models also take faults and failures into account. Faults and failures are very detrimental to performance of a control system and in order to apply control, often much more time has to be spent on the "safety net" around the control system than on the actual control loops and strategies.

In this paper, we propose a framework for incorporating faults and failures of sensors and actuators in the simulation of wastewater treatment plants. The work presented here is a part of the development of BSM1 (Copp 2002) into a new set of benchmark simulation models, the BSM1_LT (Rosen *et al.* 2004) and BSM2 (Jeppsson *et al.* 2006), initiated by the IWA Task Group on Benchmarking of Control Strategies for Wastewater Treatment Plants. The final goal is to have a model that stimulates control engineers to come up with innovative and efficient control strategies for wastewater treatment operation. However, to be successful in reality, control

doi: 10.2166/wst.2008.130

strategies also need to be reliable and robust. By adding more realism to the simulations, the step from simulation to reality ought to be somewhat shorter and easier.

The BSM1 has also been used as a data source. Data produced by the models have been used for various studies. One interesting use of this data is process monitoring. The BSM1_LT is therefore also aiming at providing a platform for benchmarking process monitoring approaches and algorithms. Also for this use, the realism of the simulated measurements is crucial to fully challenge the monitoring approach and to facilitate the move from a research product to real use at the treatment facilities.

SENSOR AND ACTUATOR FAULT MODELLING

The occurrence of a fault in a sensor or actuator is depending on many different factors of which some are deterministic and some are stochastic. It would be too complex to model all factors that influence the time, appearance and magnitude for a fault. A simplistic option to incorporate faults and failures in simulations would be to manually impose faults at desired locations during the simulation period. A problem with this approach, apart from it being quite cumbersome, is the difficulty to obtain a disturbance and fault distribution that truly is behaving according to what is observed in reality. An approach that is often seen in various industrial applications is to treat the occurrence of faults using a model. In most of this contribution, we refer to the sensor as the object of modelling. However, the procedure to model an actuator is identical.

The fault model

Markov chains are often used to model failures in industrial simulations (Olsson & Rosen 2005). A Markov chain contains a number of different states (s_i) between which the system switches according to certain transition probabilities. The transition probability to switch from s_i to s_j at time instance k is:

$$P_{ij} = P[s_j(k)|s_i(k-1)] \tag{1}$$

In the simplest form, a sensor (or actuator) is modelled having two states. One state (s_1) represents a fully functional sensor and the other state (s_2) represents any sensor fault. A Markov chain for this problem is depicted in Figure 1. In this case, the transition probability p_{12} defines the probability for a fault at any given time instance given that the sensor is functioning. The probability p_{11} defines the probability that the sensor will remain functioning as is, naturally, $p_{11} = (1 - p_{12})$. Conversely, p_{21} defines the probability that the sensor will be repaired (or replaced) and p_{22} defines the probability that the sensor will remain broken.

More generally, a Markov chain is described by its probability matrix $\mathbf{P}$, which is written as:

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1m} \\ p_{21} & p_{22} & \cdots & p_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m1} & p_{m2} & \cdots & p_{mm} \end{bmatrix} \tag{2}$$

with the condition that $0 \leq p_{ij} \leq 1$ and that the sum of each row equals 1. If we define a row vector $\mathbf{p}$ for the probability distribution we can write $\mathbf{p}(k+1) = \mathbf{p}(k)\mathbf{P}$, where k is the time instance. A Markov chain is, thus, characterized by having no memory. All the history of the system is stored in the present state. Now, assume an initial condition $\mathbf{p}(0)$ (for instance that the sensor or actuator is functioning). We can then write the probability distribution for any given time instance n as $\mathbf{p}(n) = \mathbf{p}(0)\mathbf{P}^n$. If the Markov chain is said to be ergodic, there exists a stationary solution $\mathbf{p}^*$ independent of the initial conditions. This means that it is possible to find a solution which describes the modelled system *on an average*. This is useful since it is then possible to determine the transition probabilities of $\mathbf{P}$ based on knowledge on the failure history of a certain sensor or sensor type.

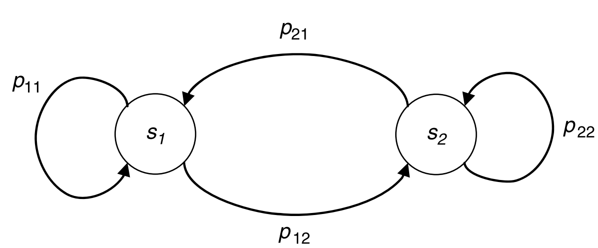

Figure 1 | A Markov chain with two states.

For instance, if on an average the sensor is working properly for 95% of the time, failing for 2% and is in calibration for the remaining 3% of the time, we want the model to produce the same result. For a small system like this, trial and error in determining the transition probabilities of **P** would suffice. However, as soon as the modelled system gets only slightly bigger and states have more than one entry path and/or exit path, it is quite difficult to guess the transition probabilities. By setting the desired stationary solution (e.g. $\mathbf{p}^* = [0.95\ 0.02\ 0.03]$ from the example above) and solving the equation:

$$\mathbf{p}^*(\mathbf{I} - \mathbf{P}) = 0 \quad (3)$$

using the additional constraint that the sum of each row equals to 1 it is possible to calculate the transition probabilities of **P**. There is, thus, an advantage of using this type of modelling since there exists a theoretical package to analyse the occurrence of faults and to tune the model to behave in a desired and realistic manner.

SENSOR FAULTS

All sensors are more or less often subject to failure. The failure type, the frequency and the time to repair are dependent on the type of sensor, their locations, maintenance schemes, etc. and will differ highly between WWTP facilities. In the model proposed here, a sensor (or actuator) can only be in one fault state at a time, i.e. multiple faults are neglected. The following sensor fault types are defined:

1. *Operational*. Measurements are only affected by normal noise according to sensor specifications.
2. *Excessive drift*. Most sensors have a tendency to drift due to various reasons. The calibration and maintenance scheme is often set so that the drift is not in a significant way affecting the measurements. However, sometimes excessive drift is observed which will have a large impact on the measurement/control system if not detected. The drift may be both positive and negative. This type of drift will not continue after calibration (per definition connected to a maintenance action in the fault model).
3. *Shift (off-set)*. A shift often occur after a wrong calibration, a change of parts/chemicals or sudden clogging of tubes. This means that the sensor will produce a value and be able to follow variations in the measured variable but with a bias in the value.
4. *Fixed value*. The sensor is stuck and delivers a constant value.
5. *Complete failure*. A completely faulty sensor is characterized by no signal or minimum (or sometimes maximum) signal for simple sensors or no/minimum signal complemented with a failure status for more advanced sensors.
6. *Wrong gain*. When sensors are calibrated, it is quite common that the gain (slope) of the sensor will be erroneous. This will lead to a change in the variability around the calibration point.
7. *Calibration*. Although not a fault, a calibration instance will not produce the correct measurement. The output is simplified to that of a complete failure.

The faults listed above can be represented in a state graph (Figure 2). As can be seen from the graph, the occurrences of the faults are different. Fault 2 – 5 can only occur when the sensor is operating (i.e. in state 1). Faults 4 and 5 return to state 1 with the probability of p_{41} and p_{51}, respectively, whereas faults 2 and 3 can only move to fault 7 (calibration). Fault 6 can only occur as a direct consequence of calibration and is only ended by a new

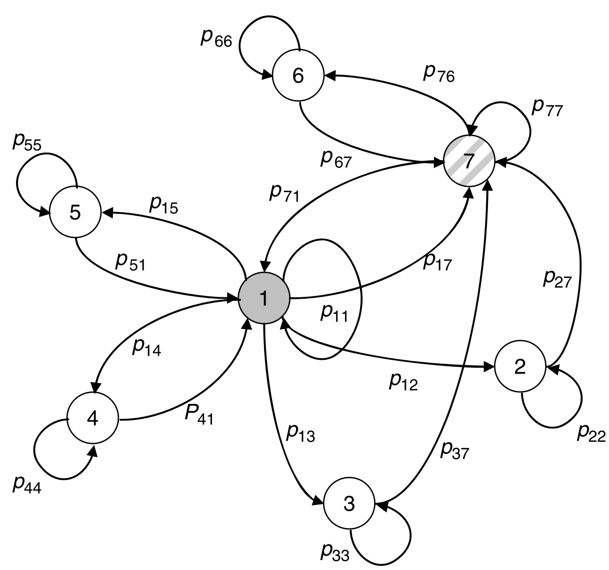

Figure 2 | A Markov chain representation of the sensor fault model.

calibration. The transition probability matrix for the Markov chain representing the 7 different states is:

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} & p_{15} & & p_{17} \\ & p_{22} & & & & & p_{27} \\ & & p_{33} & & & & p_{37} \\ p_{41} & & & p_{44} & & & \\ p_{51} & & & & p_{55} & & \\ & & & & & p_{66} & p_{67} \\ p_{71} & & & & & p_{76} & p_{77} \end{bmatrix} \quad (4)$$

with the empty spaces occupied by zeros. Note that it is the parameters of **P** that need to be determined in order to find the model for the faults in the list above.

Calibration instances

The timing for calibration in the model described above is entirely stochastic. This is in many cases not realistic, since there is normally a calibration and maintenance scheme. The calibration instances can be forced on the system at a certain frequency, for instance every week. This means that the model description must be changed somewhat. To avoid that transition to calibration is done outside the predefined scheme, **P** has to be rewritten so that fault states 2, 3 and 6 are so called absorbing states. An absorbing state means that once in this state, the system will stay in this state indefinitely, unless the system is affected from the outside, e.g. the time schedule for calibration. An absorbing state has $p_{ii} = 1$, which means that p_{22}, p_{33}, p_{77} are all equal to 1 and that $p_{12}, p_{27}, p_{37}, p_{67}$ are all equal to 0.

Some sensors have the capability to autocalibrate. This procedure is normally shorter than the manual calibration and is automatically initiated by the sensor itself. Since the duration is relatively short, autocalibration is not included in the corruption of the measurements. However, the model proposed here can easily include autocalibration as well. The normal output from a sensor during autocalibration is to hold the last known value. This fault type is already included and it is basically only a matter of estimating the transition probabilities for that fault to what is expected from autocalibration.

Actuator faults

Actuator faults are similar to those of sensors. The most obvious malfunction is naturally no actuator capacity at all. However, it is also possible to imagine faults corresponding to drift, shift and fixed value of sensors. An important difference, though, between failures in actuators and sensors is that actuators normally do not provide means to detect the malfunction. Pumps may have indicators whether they are working or not but normally no information is given on the current pump capacity. The same is true for valves, mixers, etc. The actuator faults can be modelled using the model for sensor faults but with slightly different transition probabilities and the states for wrong gain (fault state 6) and calibration (7) removed (Figure 3).

MODEL IMPLEMENTATION

Creating the fault vector

The realization of a Markov model in time is straightforward and can be written as just a few lines of code. For each time step, the transition to a new state is done according to a uniformly distributed random number, which is compared with the transition probabilities of that particular state. So far, only the timing of the fault has been discussed. However, for fault type 2 (drift), 3 (shift) and 6 (wrong gain after calibration) additional information is needed. For drift fault the rate (f_r) at which the sensor/actuator is drifting has to be defined. For the shift fault the bias (f_b)

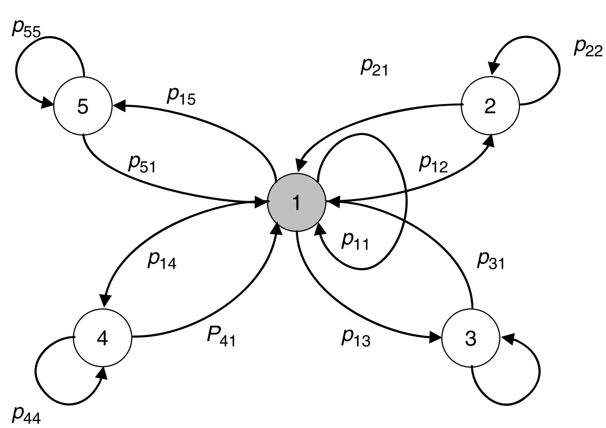

Figure 3 | A Markov chain representation of the actuator fault model.

Table 1 | Fault types and their corresponding fault vector

Fault type	Fault vector
1. Fully functional	[1 0 1 0]
2. Excessive drift	[1 $(t - t_0) \cdot f_r$ 0 0]
3. Shift	[1 f_b 1 0]
4. Fixed value	[0 0 0 1]
5. Complete failure	[0 0 0 0]
6. Wrong gain	[f_g $(1 - f_g) \cdot c_0$ 0 0]
7. Calibration	[0 0 0 0]

t = current time and t_0 = time for start of drift event.

must be set and for the wrong gain fault the incorrect gain (f_g) must be determined. Also, the calibration point (c_0), i.e. the concentration at which the calibration is incorrect, must be given. By inspecting historical data, reasonable average values for these parameters can be found. Assuming Gaussian distribution for the parameters, these are implemented as normally distributed random variables with the appropriate mean and variance. With information on the current state, a vector of the effect on the measurement is created. In Table 1, this vector is shown for the different fault types. The displayed vector is designed to fit implementation in Matlab/Simulink and is explained further in the following section.

Simulation model

Here, the classification of Rieger *et al.* (2003) and its corresponding models will be used to illustrate how the result of the Markov model is implemented in a simulation environment. In Figure 4, a block diagram of the class C_1 sensor of Rieger *et al.* (2003) is shown. It can be seen that the class C_1 sensor contains a transfer function that gives a dynamic response, a noise source, saturation and a sample and hold for mimicking the discrete output.

In Figure 5, the implementation of the fault vector on the C_1 sensor is shown. The design of the fault vector is explained by how the different faults act in different places in the sensor: a drift is imposed as a linearly increasing or decreasing bias; a shift is simply a bias; a fixed value keeps the last output value; a complete failure sets the signal to minimum output (often zero); the incorrect gain is multiplied with the signal and the calibration point is removed as a bias; and for calibration the output is set to minimum output.

EXAMPLE – SENSOR MODELLING

To illustrate how the model can be tuned to display the behaviour of a specific sensor, online ammonia measurements will be used to exemplify the approach. The measurement location is in the influent stream to a small treatment plant in Sweden, serving approximately 15 000 people (Rosen 1998). The fact that the sensor is located in the influent makes it subject to many disturbances and faults since it is exposed to the raw wastewater. In Figure 6, the ammonia measurements are shown, covering a period of 286 days and sampled every 15 minutes (in total 27,503 samples).

Identifying the Markov model

The first step is to identify the different types of faults present in the data series. By looking at Figure 6, it is clear that there are at least 3 major breakdowns of the sensor: at

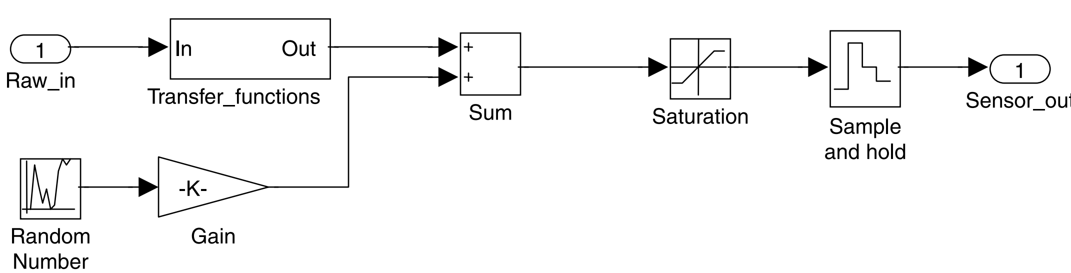

Figure 4 | A class C_1 sensor (Rieger *et al.* 2003).

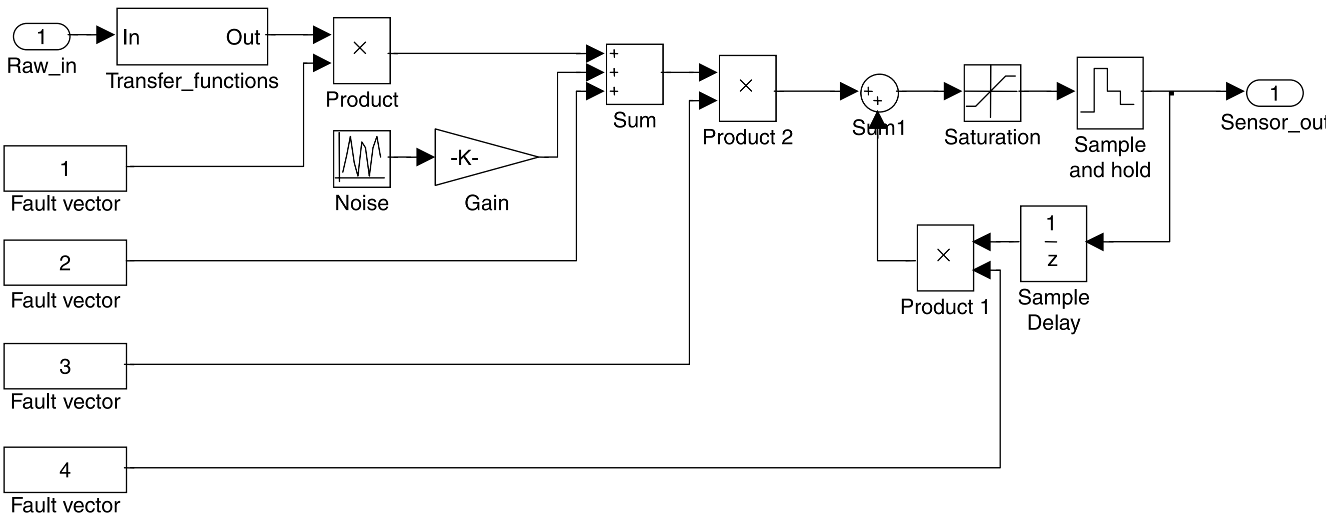

Figure 5 | The implementation of the seven fault types in a class C_1 sensor.

days 22 – 29, 141 – 146 and 271 – 286. These are classified as type 5 faults – *complete failure*. However, a closer inspection reveals another 12 breakdowns. It is also possible to distinguish calibration instances (the sudden drops in the data series), which seems to be rather irregular in time (we will therefore model the calibration instances stochastically). The sensor is calibrated (or maintained, e.g. cleaning) at 102 instances during the period. These instances are consequently classified as type 7 – *calibration*. Further, it is possible to find a number of instances where a faulty calibration results in wrong gain (or slope). These instances can be seen at days 38–42, 85–92 and 130 − 131 and are classified as type 6 – *wrong gain*. Although errors of type 2–4 certainly are present, the great variability of the data series makes it difficult to distinguish these and they are therefore not considered in this example.

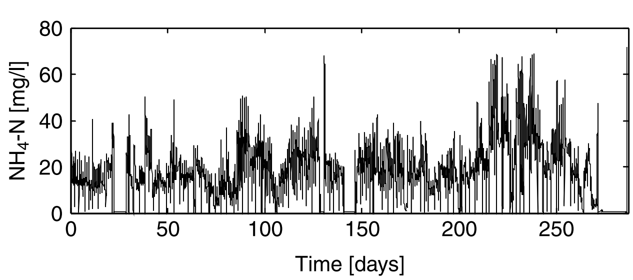

Figure 6 | Ammonia measurements in the influent of a Swedish WWTP.

The second step of characterising the measurement data is to determine the average duration of each fault type. The average duration of the 15 complete breakdowns (type 5) is 251 samples (approx. 2.6 days). For the instances with wrong gain after calibration (type 6), the average duration is 354 samples (approx. 3.7 days) and for the calibration instances (type 7) 3.8 samples (approx. 1 hour). The last piece of information needed to be able to solve Equation 3 is the fraction of the total time each fault occurs. The sensor is in the type 5 fault 14 % of the time (the number of samples in type 5 divided by the total number of samples). The sensor is in the type 6 fault 4 % of the time and in the type 7 fault 1 %. Since we disregard fault type 2 – 4, the sensor is its in normal state the remaining 81% of the time. The desired stationary solution is thus:

$$\mathbf{p}^* = [\,0.81 \quad 0 \quad 0 \quad 0 \quad 0.14 \quad 0.04 \quad 0.01\,]$$

Now, we have the information to start defining the probability matrix $\mathbf{P}$. Referring to Equation 4 there are at start 19 unknown transition probabilities. However, the number of unknowns can be reduced. Since states 2–4 are not considered, transition probabilities associated with these states can simply be set arbitrarily since p_{12}, p_{13} and p_{14} must be set to zero (meaning no probability to go to states 2 – 4). For instance, set p_{22}, p_{33} and p_{44} to one and

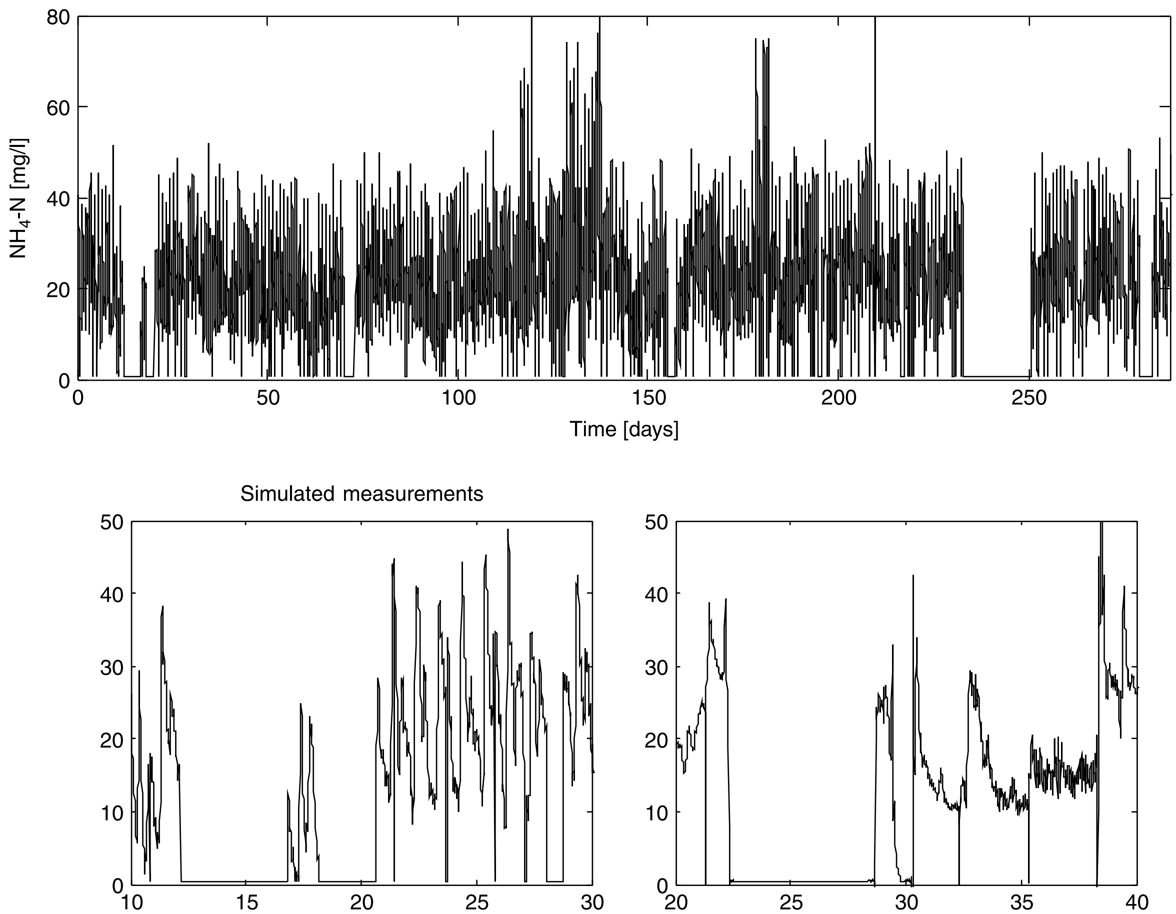

Figure 7 | The simulated ammonia measurements (top). Closer inspection of simulated measurements (bottom left) and real measurements (bottom right).

p_{27}, p_{37} and p_{41} to zero. Now, 10 unknowns remain. The information obtained from the characterisation of the faults will now be used. Since we know the average duration in states 5, 6 and 7, we can assign transition probabilities to p_{55}, p_{66} and p_{77}, which will give us p_{51} and p_{67} since we have the condition that the sum of each row must equal one. The transition probability for p_{55} is simply the 1 minus the inverse of the average duration in that state since it describes the probability to stay in the state: $p_{55} = 1 - 1/251 = 250/251$. Conversely, $p_{66} = 353/354$ and $p_{77} = 3/4$ are the transition probabilities to stay in state 6 and 7, respectively. Now, the number of unknowns has been reduced to five: p_{11}, p_{15}, p_{17}, p_{71} and p_{76}. However, from Equation 3 we only have four equations remaining but we need five to solve the system. We then use the row sum condition and add the equation $p_{71} + p_{76} + p_{77} = 1$. The solution of Equation 3 becomes:

$$\mathbf{P} = \begin{bmatrix} 0.9950 & & & & 0.0007 & & 0.0043 \\ & 1.0 & & & & & \\ & & 1.0 & & & & \\ & & & 1.0 & & & \\ 0.0040 & & & & 0.9960 & & \\ & & & & & 0.9972 & 0.0028 \\ 0.2423 & & & & & 0.0077 & 0.7500 \end{bmatrix}$$

Note that although the sensor is in the fully functional state 81% of the time, the transition probability to stay fully functional is 99.5% (p_{11}). This is perhaps somewhat surprising but nevertheless true due to the high probabilities to remain in states 5 and 6 for long periods of time.

Realization

Referring to the classification of Rieger *et al.* (2003), we use a class C_1 sensor and its corresponding model to implement the ammonia sensor in the simulation environment as shown in Figure 5. The only adjustment compared to Rieger *et al.* (2003) is that the noise level is decreased to 1% of the maximum value since this fits better with the real measurements (the proposed level is 2.5%). Since we disregard fault types 2 and 3, the only remaining parameter to set is the one of fault 6 (wrong gain). Since the real ammonia data displayed an increased gain, this parameter is set to a random variable with an average of 2 and a standard deviation of 0.1 (when the fault occurs the gain is doubled). As input to the sensor model we use the influent ammonia to the BSM2/BSM1_LT (Gernaey *et al.* 2006). It is important to note that no attempts have been made to make the simulated measurement similar to the real measurement in terms of mean value, diurnal variation, etc. Focus is only on the quality aspects of the measurements, that is, faults, failures and noise.

In Figure 7, the simulated sensor is shown (top panel). When comparing this with the real sensor output (Figure 6), they display similar behaviours in terms of faults, calibrations, etc. Also, at a closer investigation of both the simulated and the real measurements, it is not obvious which is real and which is simulated.

CONCLUSIONS

In this paper, an approach for modelling sensor and actuator faults and failures is proposed. The approach is based on the theoretical framework of Markov chains. The approach allows for transferring sensor and actuator history into sensor/actuator models, which produce realistic characteristics. It is shown by an example that the Markov framework is suitable for generating faults and together with models for sensor dynamics will add significantly to the realism of simulated measurements. The added realism can be used to test robustness of different control and monitoring strategies.

ACKNOWLEDGEMENTS

Peter Vanrolleghem holds the Canada Research Chair in Water Quality Modelling.

REFERENCES

Alex, J., To, T. B. & Hartwig, P. 2002 Improved design and optimization of aeration control for WWTPs by dynamic simulation. *Wat. Sci. Tech.* **45**(4–5), 365–372.

Copp, J. B. (Editor) 2002. *The COST simulation benchmark – Description and simulator manual*. COST action 624, Office for official publications of the European Communities, Luxembourg.

Gernaey, K. V., Rosen, C. & Jeppsson, U. 2006 WWTP dynamic disturbance modelling – an essential module for long-term benchmarking development. *Wat. Sci. Technol.* **53**(4–5), 225–234.

Jeppsson, U., Rosen, C., Alex, J., Copp, J., Gernaey, K., Pons, M. N. & Vanrolleghem, P. A. 2006 Towards a benchmark simulation model for plant-wide control strategy performance evaluation of WWTPs. *Wat. Sci. Technol.* **53**(1), 287–295.

Olsson, G. & Rosen, C. 2005 *Industrial Automation – Applications, Structures and Systems*. IEA, Lund University, Sweden.

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

Rieger, L., Alex, J., Gujer, W. & Siegrist, H. 2006 Modelling of aeration systems at wastewater treatment plants. *Wat. Sci. Technol.* **53**(4–5), 439–447.

Rosen, C. 1998 Monitoring wastewater treatment systems. Licentiate's thesis, Lund University, Sweden. ISBN 91–88934–10–1.

Rosen, C., Jeppsson, U. & Vanrolleghem, P. A. 2004 Towards a common benchmark for long-term process control and monitoring performance evaluation. *Wat. Sci. Technol.* **50**(11), 41–49.